48－E002

中級情報処理技術者育成指針

専　　門　　編

昭和49年3月

JIPDEC

財団法人　日本情報処理開発センター

この報告書は，日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて，昭和48年度に実施した「情報処理の普及促進に関する事業」の一環としてとりまとめたものであります。

I

18

東門

序

当財団では，昭和43年以来数年にわたり，情報処理技術者育成事業の一環として，まずはじめに上級技術者を対象とする「上級情報処理技術者育成指針」，次に初級技術者を対象とする「初級情報処理技術者育成指針」を作成し，それぞれのレベルにおける情報処理技術者の教育目標とそれに至るための教育内容，方法を示してきました。

しかし，情報処理の分野では，これらの上級と初級との中間ともいうべき層の情報処理技術者を養成する教育カリキュラムもまたきわめて重要であり，これの完成によって，はじめて初級から上級までの一貫した情報処理教育のカリキュラム体系が完成できるといえましょう。

このような考えにもとづき，当財団では昭和47年より「中級情報処理技術者育成指針」の作成に着手しましたが，幸い各界の多数の専門家のご協力を得てこゝに完成のはこびとなりました。

この事業実施にご尽力いただいた関係各位に心から感謝の意を表しますとともに，この中級育成指針が各方面で利用され，わが国における情報処理技術者育成の一助となりますことを念願する次第であります。

昭和49年3月

財団法人　日本情報処理開発センター
会　長　難　波　捷　吾

中級情報処理技術者育成指針委員会

（敬称略，50音順）

委員長	山内二郎	東京大学
委員	石川利男	日本航空（株）
〃	今村茂雄	（財）情報処理研修センター
〃	魚木五夫	広島修道大学
〃	鵜沢昌和	青山学院大学
〃	木幡靖郎	日本電信電話公社
〃	向阪　浩	通商産業省
〃	小林健二	日本電気（株）
〃	南條　優	日産プリンス自動車販売（株）
〃	間野浩太郎	青山学院大学
〃	篠崎・敬	（財）日本情報処理開発センター
〃	中嶋栄之助	（財）日本情報処理開発センター
〃	甲賀将之	（財）日本情報処理開発センター

中級情報処理技術者育成指針作成小委員会

（敬称略，50音順）

委員長	魚木五夫	広島修道大学
委員	赤木耀介	（株）日立製作所
〃	石井信司	日本電気（株）
〃	石川利男	日本航空（株）
〃	大日方　真	日本コンピュータ（株）
〃	上條史彦	情報処理振興事業協会
〃	杉山　進	日本電信電話公社
〃	南條　優	日産プリンス自動車販売（株）
〃	服部幸英	日本鋼管（株）
〃	原田睦明	（株）ソーシャル・サイエンス・ラボラトリー
〃	平尾信正	東京亘斯（株）
〃	平松啓二	東京電機大学
〃	間野浩太郎	青山学院大学
〃	甲賀将之	（財）日本情報処理開発センター
〃	山本欣子	（財）日本情報処理開発センター

中級情報処理技術者育成指針作成小委員会第１分科会

（敬称略，50音順）

主　査	魚木五夫	広島修道大学
委　員	赤尾嘉治	（財）情報処理研修センター
〃	岡本行二	東京芝浦電気（株）
〃	塩田俊朗	（財）情報処理研修センター
〃	元植郁夫	広島修道大学

中級情報処理技術者育成指針作成小委員会第２分科会

（敬称略，50音順）

主　査	甲賀将之	（財）日本情報処理開発センター
委　員	島野滋雄	トピー工業（株）
〃	高森　寛	青山学院大学
〃	成田誠之助	早稲田大学
〃	東口　寛	東京大学
〃	藤代侑宏	東京理科大学
〃	矢田光治	電子技術総合研究所
〃	若松清司	電子技術総合研究所

中級情報処理技術者育成指針作成小委員会第３分科会

（敬称略，50音順）

主　査	石井信司	日本電気（株）
委　員	石川宏明	日本電気（株）
〃	折出岩男	日本電気（株）
〃	鈴木　蕃	日本電気（株）
〃	田中　隆	日本電気（株）
〃	吉田敬一	日本電気（株）

中級情報処理技術者育成指針作成小委員会第４分科会

（敬称略，50音順）

主　査	大日方　真	日本コンピュータ（株）
委　員	岩城三郎	（株）フジミック
〃	戸田英雄	電子技術総合研究所

中級情報処理技術者育成指針作成小委員会第５分科会

（敬称略，50音順）

主　査	原田睦明	（株）ソーシャル・サイエンス・ラボラトリー
委　員	穂鷹良介	（株）ソーシャル・サイエンス・ラボラトリー
〃	渡辺　功	（株）ソーシャル・サイエンス・ラボラトリー

事務局　（財）日本情報処理開発センター
技術部　教育課

総　目　次

中級情報処理技術者育成指針

総　目　次

第1分冊「基礎編」

Ⅰ．総　論

1. 情報処理技術者の育成…… 1
2. 中級情報処理技術者育成指針の位置づけ…… 10
3. 中級育成指針の構成内容とカリキュラムの編成…… 16
4. 学習指導計画と指導法…… 27

Ⅱ．基　礎　科　目

情報処理概論…… 37

1. 情報処理概論…… 37

コミュニケーション技法…… 47

1. コミュニケーション技法…… 47

コンピュータ・システム…… 55

1. コンピュータ・システムの構成と発展動向…… 56
2. データ通信システムの概要…… 64
3. コンピュータ導入概要…… 70

ハードウェア(Ⅰ)…… 79

1. システム構成…… 80
2. 演算処理装置…… 87
3. 記憶装置…… 93
4. 入出力装置…… 102

ソフトウェア…… 109

1. ソフトウェアの概念…… 110
2. オペレーティング・システムの概念…… 113

3. プログラム言語の概要…… 115
4. COBOL言語の概要 …… 119
5. FORTRAN言語の概要 …… 123
6. ソフトウェアの評価…… 126
プログラミング言語（FORTRAN） …… 129
1. FORTRAN概要…… 130
2. FORTRANプログラム形式 …… 131
3. データ…… 133
4. 式…… 135
5. 実 行 文…… 137
6. 非実行文…… 141
7. 手続きと副プログラム…… 144
8. その他…… 146
9. まとめ…… 147
プログラミング言語（COBOL）…… 149
1. COBOL概論 …… 151
2. 言語の概念…… 152
3. 見出し部と環境部…… 154
4. データ部…… 156
5. 手続き部…… 159
6. 報告書機能…… 164
7. 分類機能…… 165
8. 区 分 化…… 166
9. 登 録 集…… 167
10. 予 約 語…… 168
11. 練習問題…… 169
プログラム設計(I)…… 171
1. プログラム開発の手順…… 172
2. プログラム開発方法…… 176
3. プログラムの構造…… 181
4. ファイル処理…… 190
5. ジョブ制御とオペレータ制御…… 198

6. サービス・プログラムの利用…… 201
7. プログラムの基本設計…… 207
8. プログラム流れ図の作成とコーディング…… 211
9. プログラム・テスト…… 218
10. プログラム設計と文書化…… 222
経営実務 …… 227
1. 企　業…… 228
2. 経　営…… 234
3. 計　画…… 240
4. 組　織…… 243
5. 統制・管理…… 248
6. 事　務…… 256
システム分析・設計(I) …… 261
1. システム・アプローチ…… 262
2. システム分析・設計の手順…… 264
3. システム分析・設計チーム…… 267
4. 業務設計 …… 269
5. コード設計 …… 271
6. 入出力設計 …… 273
7. ファイル設計…… 279
8. プロセス設計…… 285
9. システムの評価…… 287
10. 文書化と標準化 …… 290
コンピュータ室の運用管理…… 295
1. コンピュータ室の運用管理…… 295
ソフトウェアの生産管理…… 307
1. ソフトウェアの生産管理…… 307
情報処理用数学…… 315
1. 集合と論理…… 316
2. ベクトルと行列…… 320
3. 微分と積分…… 323
4. 確率と統計…… 326

5. コンピュータと数値計算…… 330
システム概論…… 333
1. システム工学概論…… 334
2. システムのモデル化…… 338
3. システム・シミュレーション…… 342
4. システムの最適化…… 348
5. システムの信頼性…… 353
経営科学(I)…… 357
1. 経営科学の基礎…… 358
2. 経営科学の諸手法などの紹介…… 368

第2分冊「専門編」

専 目 科 目

数値計算法 …… 1

1. 数値計算と誤差 …… 2
2. 常微分方程式 …… 5
3. 線形および非線形，建立代数方程式と固有値 …… 10
4. 代数方程式 …… 17
5. 補間法と数値微分・数値積分 …… 25

統計解析 …… 29

1. 母集団と試料 …… 30
2. 検定と推定 …… 36
3. 実験計画法 …… 42
4. 統計的解析 …… 53

シミュレーション …… 57

1. コンピュータ・シミュレーション …… 58
2. シミュレーション・モデルの作成 …… 67
3. モデルの運用 …… 75
4. シミュレーション言語 …… 79
5. 演　習 …… 89

コントロール・システム …… 97

1. 制御工学の基礎 …… 98
2. 情報の収集・変換・伝送 …… 104
3. コントロール・システムのモデル化 …… 108
4. 最適化と適応化 …… 110
5. コントロール・システムにおけるコンピュータの機能と構成 …… 114
6. 制御用コンピュータのソフトウェア …… 118
7. コントロール・システムの実例 …… 123
8. コントロール・システムにおけるソフトウェア …… 129

経営科学(II) …… 135
1. 線形計画法 …… 136
2. 経済評価 …… 144
3. ダイナミック・プログラミング(動的計画法) …… 147
4. ネットワーク分析と計画管理 …… 152
5. 在庫管理 …… 161
6. 予　測 …… 167
システム分析・設計(II) …… 171
1. 問題発見の技術 …… 172
2. システム分析の技術 …… 176
3. システムの組立て …… 185
4. 業務システムの設計 …… 194
5. 経営管理システム …… 199
6. システム思考 …… 202
7. オンライン・システムの設計 …… 204
8. システムの評価技法 …… 210
9. 各種システムの具体的設計例 …… 220
10. 演　習 …… 223
コンピュータ導入計画 …… 225
1. コンピュータ導入計画のたて方 …… 226
2. 導入組織と人員計画 …… 231
3. 教育計画 …… 234
4. 機械化計画とプログラム開発計画 …… 236
5. 設備計画 …… 238
6. コンピュータ用品と消耗品 …… 243
7. 機械化移行計画 …… 245
経営情報 …… 247
1. 経営活動と情報システム …… 248
2. 経営情報システム …… 253
3. 経営における情報システムの例 …… 258
4. 経営情報システムの開発(演習) …… 269

データ・ベース …… 283
1. 概　説 …… 284
2. データ・ベースの諸機能 …… 289
3. データ・ベースの構築技術 …… 293
4. データ・ベース管理者機能 …… 299
5. ユーザ・インターフェイス …… 301
6. データ・ベースの標準化 …… 307
7. 具体例 …… 309
8. データ・ベースと情報検索 …… 310
データ通信システム …… 315
1. データ通信システムの概要 …… 316
2. 端末装置 …… 323
3. 通信回線 …… 332
4. センタ・システムのハードウェア …… 337
5. オンライン・システムのソフトウェア …… 343
6. オンライン・システムのファイル …… 351
7. 信頼性 …… 354
ハードウェア(II) …… 363
1. 処理方式 …… 364
2. 性能評価 …… 379
3. 信頼性 …… 388
4. 互換性 …… 392
5. 実　習 …… 396
オペレーティング・システム …… 397
1. オペレーティング・システムの基本 …… 398
2. プログラム管理 …… 401
3. ジョブ管理 …… 404
4. タスク管理 …… 412
5. データ管理 …… 416
6. 通信管理 …… 423
7. 障害管理 …… 433
8. タイムシェアリング・システム …… 435

言語プロセッサ…… 443
1. 言語の記述…… 444
2. 基礎技法…… 449
3. プロセサ…… 452
4. コンパイラの構成…… 455
5. 設計演習…… 467
アセンブリ言語…… 469
1. プログラミングのためのハードウェア…… 471
2. アセンブリ言語の構成…… 475
3. 実行命令およびアセンブリ命令…… 478
4. プログラムの分割と連絡…… 482
5. マクロ言語…… 484
6. 実習のためのオペレーティング・システム…… 489
7. 実　習…… 491
プログラム設計(II)…… 497
1. プログラム構造とモジュール設計…… 498
2. データ構造…… 505
3. 分類と組合せの技法…… 509
4. 操索の技法…… 513
5. ファイル処理…… 515
6. オペレーティング・システムの利用…… 521
7. オンライン・システムのプログラミング…… 526
汎用プログラム・パッケージ…… 529
1. 汎用プログラム・パッケージ…… 530
2. 実　習…… 534

專　門　編

数 値 計 算 法

目 次

第1章　数値計算と誤差 …… 2
1.1　数値計算 …… 2
1.2　誤　差 …… 3
第2章　常微分方程式 …… 5
2.1　微分方程式の分類 …… 5
2.2　オイラー法 …… 6
2.3　ヒューン法 …… 7
2.4　ルンゲ・クッタ法 …… 8
2.5　ミルン法 …… 9
2.6　ハミング法 …… 9
第3章　線形および非線形連立代数方程式と固有値 …… 10
3.1　行列演算の基礎 …… 10
3.2　行列・ベクトルの基本演算プログラム …… 11
3.3　線形連立代数方程式 …… 11
3.4　ガウス・ジョルダンの消去法 …… 12
3.5　ガウス・ザイデルの反復法 …… 12
3.6　ニュートン・ラフソン法 …… 13
3.7　擬似逆行列 …… 13
3.8　大規模な問題に用いられる技法 …… 13
3.9　固有値と固有ベクトル …… 14
第4章　代数方程式 …… 17
4.1　方程式の根 …… 17
4.2　多項式の性質 …… 17
4.3　超越関数の性質 …… 18
4.4　はさみうち法 …… 18
4.5　ニュートン法 …… 20
4.6　ホーナー法 …… 21
4.7　逐次代入法 …… 21
4.8　複素根 …… 22
4.9　ヒッチコック・ベアストウ法 …… 23
4.10　収束と各種手法の比較 …… 23

第5章　補間法と数値微分・数値積分 …… 25
5.1　補間法 …… 25
5.2　グレゴリー・ニュートンの補間式 …… 25
5.3　差　分 …… 26
5.4　ラグランジュの補間法 …… 26
5.5　補間における誤差 …… 26
5.6　数値微分 …… 27
5.7　数値微分の誤差 …… 27
5.8　数値積分 …… 27
5.9　数値積分の誤差 …… 28

数 値 計 算 法

目 次

第1章　数値計算と誤差 …… 2
1.1　数値計算 …… 2
1.2　誤　差 …… 3
第2章　常微分方程式 …… 5
2.1　微分方程式の分類 …… 5
2.2　オイラー法 …… 6
2.3　ヒューン法 …… 7
2.4　ルンゲ・クッタ法 …… 8
2.5　ミルン法 …… 9
2.6　ハミング法 …… 9
第3章　線形および非線形連立代数方程式と固有値 …… 10
3.1　行列演算の基礎 …… 10
3.2　行列・ベクトルの基本演算プログラム …… 11
3.3　線形連立代数方程式 …… 11
3.4　ガウス・ジョルダンの消去法 …… 12
3.5　ガウス・ザイデルの反復法 …… 12
3.6　ニュートン・ラフソン法 …… 13
3.7　擬似逆行列 …… 13
3.8　大規模な問題に用いられる技法 …… 13
3.9　固有値と固有ベクトル …… 14
第4章　代数方程式 …… 17
4.1　方程式の根 …… 17
4.2　多項式の性質 …… 17
4.3　超越関数の性質 …… 18
4.4　はさみうち法 …… 18
4.5　ニュートン法 …… 20
4.6　ホーナー法 …… 21
4.7　逐次代入法 …… 21
4.8　複素根 …… 22
4.9　ヒッチコック・ベアストウ法 …… 23
4.10　収束と各種手法の比較 …… 23

第 5 章　補間法と数値微分・数値積分 …… 25
5.1　補間法 …… 25
5.2　グレゴリー・ニュートンの補間式 …… 25
5.3　差　分 …… 26
5.4　ラグランジュの補間法 …… 26
5.5　補間における誤差 …… 26
5.6　数値微分 …… 27
5.7　数値微分の誤差 …… 27
5.8　数値積分 …… 27
5.9　数値積分の誤差 …… 28

科目「数値計算法」

教育の目標

この科目は科目「情報処理用数学」で学んだ事項を基礎として，コンピュータによる数値計算の特徴をより深く理解させるとともに，各種の基本的な数値計算を自由に処理できるような実力を身につけさせることを目標とする。

コンピュータと数値計算に関する基本的な考え方は科目「情報処理用数学」の第5章でふれてあるが，そこでは比較的易しい問題を扱っており，また比較的あさくふれてあるので，ここでは部分的には重複するが，より深い理解と数値計算のための算法を計画し，コンピュータによる処理を研修生自身で相当範囲までできるようにすることをねらいとしている。

時間配分

章	履習時間（時間）		
	講義	演習	実習
1．　数値計算と誤差	5		
2．　常微分方程式	15		10
3．　線形および非線形 連立代数方程式と固有値	15		10
4．　代数方程式	15		10
5．　補間法と数値微分・数値積分	10		
合　　計	60		30

第１章　数値計算と誤差

用　語　この章ではつぎの用語を教える。

数値解析，反復法，数値法，有限桁数計算，４捨５入誤差

目　標

数値計算に固有な誤差である４捨５入や切り捨て誤差ならびにそれらの累積誤差などについて説明し，数値計算における誤差の概念を理解させる。最後に必要な有効桁数の答を得るためにどのような注意が必要かを具体例をあげて，説明する。

内　容

1.1　数値計算

(1) 数の表現

今日われわれは数の表現に10進法を使っているがコンピュータでは２進法や８進法が使われていることの意味をのべ，それぞれの内容と相互変換がどのように行われるかを具体例をあげて説明し，演習などをさせて理解させる。特に整数の場合は問題がないが，小数の変換の際，切り捨て誤差の生ずることのあることを示しておきたい。例えば0.6を２進法で表わそうとすると，

$$0.6 = (0.100110011001\cdots\cdots)_2$$

と無限に続くが，コンピュータでは有限個の２進数しか扱えないのでここに誤差が生ずる。

続いて各種進法（10進法，２進法，８進法，16進法）における加算，減算，乗算，除算の方法を説明する。演算の仕方は10進法とその他は基本的に同一であることを理解させる。なお，コンピュータでは減算に補数を使うことを説明し，その仕方を理解させる。

最後に数の表現として固定小数点と浮動小数点のあることを示し，固定小数点演算では誤差に対する考慮が計算の設計の段階で必要であることを理解させる。

(2) 数値解析

数値解析についてのべる。物理学上の問題などを解く場合，数値的な答が欲しい場合の多いことを示す。その問題が数式でモデル化された時，数学的な処理により解が代数式，例えば一次式や二次式などで表現できるとき"閉じた形"の解があるといわれる。その時は単なる代数計算だけを行なえばよいが，多くの数字式は閉じた形の解をもたない。その場合の解は試行錯誤で求めることになり，前の計算結果の一つ，あるいは数値を次の近似計算に使用することを**反復法**と呼び，このようにして解くのがコンピュータの妙味になっていることを理解させる。

1次式2次式は単なる**数値法**で求められるが4次式以上になると反復法を適用しないと解けないことを示す。またその時，解の近似式が必要となることを簡単に説明する。

1.2 誤　差

(1) 誤差の発生

まず，数値計算では無限数の演算はできないので，各演算ごとに4捨5入あるいは切り捨ての操作を行なわねばならないことを説明し，これにより結果にある大きさの誤差が生ずること，また無限級数に展開した関数にも項を途中で切り捨てて演算するための誤差が入ってくることを理解させる。

最後にコンピュータで演算するときの誤差を評価する手法とこの誤差を小さくする手法を述べる。

技術上の問題は厳密な解の値を求めることがほとんどできないことを示す。例えば円周を求めるにしても無限数のπが関係してくることなどを例示すればよい。有効な解を得るためにはいつどのような近似が行なわれたかを知っておく必要のあることを説明する。近似は普通

① 数式モデルを作るため単純化が行なわれるとき，

② 測定上データが不正確であるとき，

③ 数学モデルの解を求めるときに近似を行なうとき

に生ずることを説明し，特に②と③がコンピュータによる数値計算で問題になることを明確にさせる。

(2) 誤差の考え方

測定量の精度は測定装置の最小桁の尺度で決るが，計算が意味をもつ有効桁がそれによって支配されることを身近な例を示して理解させる。温度計，電圧電流計などの例をあげることができる。

また得られた解によって物を設計，加工する段階で必要とされる有効桁数もそう多くはないことを明示する。例えば電気回路ではインピーダンスなどの値が，せいぜい3桁程度で与えられ，しかも温度，電流値などによって変動するので，電流を5桁以上有効桁数で計算するのはナンセンスであることなどをよく理解させる。

コンピュータによる計算に**有限桁数計算**では4捨5入がどうしても必要になってくることを説明し，テーラー展開による数値計算では項の切り捨てが必要になり，その場合やはり誤差の入ってくることを明らかにする。

(3) 誤差の表現

誤差の表現法として絶対誤差，相対誤差，パーセント誤差などのあることを述べておく。続いて誤差の伝達，累積について述べる。四則演算の際，誤差がどのように伝達されるかをよく説明しておく。

例えばAが非常に大きくBが非常に小さいような場合，コンピュータでは$A \pm B = A$となることのあることをよく理解させておきたい。

次に**4捨5入誤差**がどのように累積するかを例をあげて説明する。微分方程式，補間計算などで歩幅を極端に小さくしすぎると誤差が大きくなるようなことを説明しておく。

(4) 誤差への対策

以上の誤差の発生の原因を理解した上で，どのようなオーダの誤差がでるか，どのような計算設計をしたら誤差が少なくできるかについて例示しておく。個々の対策は以下の各章でふれたい。

最後により有効桁数の必要なときにはコンピュータで倍精度計算ができることを示し,その際の四則演算はどのような注意が必要かを述べておく。

指導上の留意点

ここでは個々の計算手法に入る前の準備としてディジタル型コンピュータによる数値計算はどのようなものであるか，どのような誤差が入るかを概念的に理解させることに重点をおく。具体的な問題，例えば行列計算，補間，微分方程式などの計算ではそれらに固有な誤差があるので，そこで具体的に理解させたい。

第 2 章　常微分方程式

用　語　この章ではつぎの用語を教える。

線形微分方程式，非線形微分方程式，初期値，境界値，固有値，オイラー法，ルンゲ・クッタ法，予測子・修正子法，ミルン法

目　標

ここでは，微分方程式を解くための，いろいろな数値計算法を紹介し，その特徴を理解させる。各種数値計算法の特徴を明確にするため，まず，微分方程式の分類，すなわち，線形と非線形，一段階法と多段階法などを説明するとともに，数値計算上における問題の位置づけを明確にするために，初期値問題と固有値問題の区別を理解させる。また，各種数値計算法による数値解と，誤差の関連を説明し，所要の精度にあった適切な数値計算法を選択できるような能力を修得させる。

内　容

2.1　微分方程式の分類

時間とともに変動するシステム，たとえば，電気の分野では簡単なL－R－C回路などの物理現象をとりあげて，微分方程式の性格を具体的に説明する。つぎに，微分方程式の分類として，

① 偏微分方程式と常微分方程式

② **線形微分方程式と非線形微分方程式**

③ 高階微分方程式と連立微分方程式

の3種類があることを説明し，問題の微分方程式がどの種類に属するかを明確にする。

つぎに，数値計算法の分類として，一段階型と多段階型の2種類があることを述べ，一段階型に属する数値計算法としては，

① オイラー法

② ヒューン法

③ ルンゲ・クッタ法

多段階型に属する数値計算法としては，

① ミルン法

② アダムス法

③ ハミング法

などがあることを簡単に説明するとよい。また，一段階法と多段階法の違いを明確にするとよい。

さらに，解析しようとする問題は初期条件，境界条件および助変数を微分方程式などによって，

① **初期値**

② **境界値**

③ **固有値**

の3種類の問題に分類できることを簡単に説明する。数値計算を行なう実際問題では境界値問題が多いということを述べ，境界値問題の処理にはかなりの工夫が必要であることを，専門分野ごとに例をあげて説明するとよい。

また，これまで説明した数値計算法は，有限個の助変数をもつ関数族で近似する差分近似によるものだけであるが，他の数値計算法もあるので簡単にふれておくとよい。

一段階型および多段階型に属する数値計算法とは，いずれも $dy/dx = f(x, y)$ で表わされる微分方程式を，$x = x_0$ のとき $y = y_0$ という初期条件のもとで解くことであることを述べ，上記の微分方程式が (x, y) の各点において方向を指定するものであり，曲線 $y = g(x)$ の各点での接線がその方向に一致するとき，これが解となることを説明する。

また一般に，微分方程式 $dy/dx = f(x, y)$ の解 $y(x)$ を初期値 x_0 でテーラー展開（Tayler's expansion）すると，

$$y(x_0 + h) = y(x_0) + hf(x_0, y(x_0)) + \frac{h^2}{2!} f'(x_0, y(x_0)) + \cdots\cdots$$

となることを説明する。

2.2 オイラー法

オイラー法では，微分方程式の解をテーラー展開し，次式のように第2項以下を打ち切った計算式を採用していることを説明し，

$$y(x_0 + h) = y(x_0) + hf(x_0, y(x_0))$$

これを一般化して，

$$y_i + 1 = y_i + hf(x_i, y_i)$$

で表わしていることを述べる。

オイラー法は，誤差が大きいためコンピュータを利用した数値計算では，ほとんど使われていないが，計算が簡単であるため手計算での概略をつかんだりするのに有効な計算法であることを簡単に説明しておく必要がある。

とくに，きざみ幅と誤差との関連を理解させるため，つぎのような数値例をあげるとよい。

与式　$\dfrac{dy}{dx} = x + y$

初期値　$x_0 = 0$　　$y_0 = 0$

積分のきざみ幅を $h = 0.01$，　$h = 0.1$ のそれぞれについて $x = 1$ まで求めてみる。

$h = 0.1$ の場合

x	y		
0.0	0.0		
0.10000	0.0		
0.20000	0.01000		
0.30000	0.03100		
0.40000	0.06410		
0.50000	0.11051		
0.60000	0.17156		
0.70000	0.24872		
0.80000	0.34359		
0.90000	0.45795		
1.00000	0.59374	真値	誤差
		0.71828	0.12454

$h = 0.01$ の場合

x	y		
0.0	0.0		
0.10000	0.00462		
0.20000	0.02019		
0.30000	0.04785		
0.40000	0.08886		
0.50000	0.14463		
0.60000	0.21670		
0.70000	0.30676		
0.80000	0.41672		
0.90000	0.54863		
1.00000	0.70481	真値	誤差
		0.71828	0.01347

2.3 ヒューン法

オイラー法の累積打切り誤差が，ほぼきざみ幅の値hに比例するのに対して，きざみ幅をhにしたときと，きざみ幅の値を$h/2$にしたときの二つのyの値から傾斜決定を行なって精度をあげるようにした計算法がヒューン法である。図表を使って幾何学的意味を説明すれば，より効果的である。
$(y_{n+1} - y_n)$の打切り誤差がオイラー法ではh^2のオーダであるのに対して，ヒューン法ではh^3のオーダであること，および，累積打切り誤差がh^2の定数倍で評価されることを説明に付け加えるとよい。

2.4 ルンゲ・クッタ法

常微分方程式の初期値問題の数値計算法として，もっともよく使われるのがルンゲ・クッタ法である。

その理論的背景を明確にするため，この数値計算法がヒューン法をさらに変形し，傾斜決定の計算回数を4回にふやして，（$y_{n+1} - y_n$）の打ち切り誤差をh^4 の項まで合わせるようにしたものであることを説明する。傾斜決定の計算回数をm回までふやして，（$y_{n+1} - y_n$）の打ち切り誤差をm次のルンゲ・クッタ法(Runge-Kutta method)と呼ぶことを述べ，オイラー法とヒューン法がそれぞれ$m=1,2$の場合にあたることを説明に加えるとよい。一般に使われている$m=4$の，いわゆる4次のルンゲ・クッタ法が，かなり精度の高い解を得ることのできる計算法であることを述べ，5次以上では急に計算の手間がかかるようになることを付け加えることによって，4次のルンゲ・クッタ法がよく使われる理由を十分理解させておくとよい。

ルンゲ・クッタ法の説明にあたっては，まずxの値をhきざみにとって，

$$x_n = x_0 + nh \quad (n=1,2,3\cdots\cdots) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots(1)$$

でのyの値y_n を，次の式で順次に求めてゆくことを説明し，

$$y_{n+1} = y_n + (k_1 + 2k_2 + 2k_3 + k_4)/6 \cdots\cdots\cdots\cdots\cdots(2)$$

ここに，

$$k_1 = hf(x_n,\ y_n),$$
$$k_2 = hf(x_n + h/2,\ y_n + k_1/2),$$
$$k_3 = hf(x_n + h/2,\ y_n + k_2/2),$$
$$k_4 = hf(x_n + h,\ y_n + k_3)\cdots\cdots$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots(3)$$

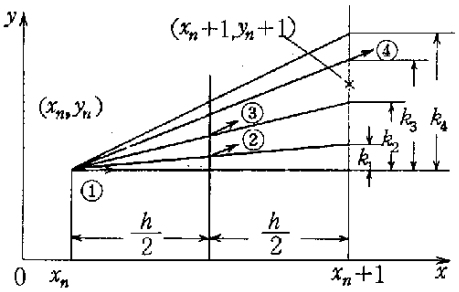


図2－1ルンゲ・クッタ法

つぎに，図2－1を用いて下記のように解法の意味を解説するとよい。すなわち，まずはじめにx_n での解の値y_n は既知として，①の点，すなわち（x_n，y_n）で右辺を計算する。これは方向$f(x_n, y_n)$で水平にhだけゆくと，y座標はk_1 だけあがることを意味する。ここで一度に$x_n + h$までゆかず，この直線上を水平に$h/2$だけいったところ，②の点，すなわち（$x_n + h/2$，$y_n + k_1/2$）で第2回目の傾斜を計算する。この傾斜をもつ直線にそって水平にhだけゆけばk_2 だけあがることを意味する。

さらに，①の点から$h/2$ だけいったところで，第3回の傾斜計算を行ない，③の点，すなわち（$x_n + h/2$，$y_n + k_2/2$）を求める。この傾斜で水平にhだけ進むと，k_3 だけ上がることを意味する。最後に①の点からhだけ進んだときの傾斜計算を行ない，④の点，すなわち（$x_n + h$，$y_n + k_3$）を求める。このように，4回の傾斜計算で，水平にhだけ進む間の上昇量として，k_1，k_2，k_3，k_4 という4個の評価が得られ，これに1：2：2：1 という重みをかけて平均した量だけyを補正してやろうというのが(2)式の意味である。

ここで重みを1：2：2：1 としているのは，$y_{n+1} - y_n$ の打ち切り誤差をh^4 の項までに納め

るようにするためであることを指摘する。

また，ルンゲ・クッタ法を改良したものにルンゲ・クッタ・ギル法がある。これは，ルンゲ・クッタ法に比べて，

① コンピュータを使用するときに，作業用記憶場所が少なくてすむこと，

② 未知関数 y が多いときに有利であること，

③ 丸め誤差をフィードバックして誤差を減らすように工夫されていること

などの特徴をもっていることを説明するとともに，計算手順がやや複雑になるという難点のあることも指摘しておくとよい。

2.5 ミルン法

1段階法が差分法において次段の解 x_{n+1} を求めるのに前段だけの値 x_n を使って計算を進めるのに対して，多段階法は x_{n-1}，x_{n-2}，……における値も利用して計算を進める計算法であることを述べ，多段階法の特徴である予測子・修正子の意味と計算法を説明する。

たとえば，1階常微分方程式 $dy/dx = f(x, y)$ を解くとき，独立変数 x をきざみ幅で分割して，

$$x_n = x_0 + nh \quad (n = 1, 2, \cdots\cdots)$$

における y の値 y_n を順次求めることになるが，y_{n-1} までの値で y_n の値をまず予測し，つぎに $f_{(n)} = f(x_n, y_n)$ の値を補助的に使って修正するという手順を，十分な精度が得られるまで繰返していく方法を**予測子・修正子法**と呼んでいる。その代表的な方法が**ミルン法**である。

多段階法では，x_n での値 y_n を求めるのに，y_{n-1}，y_{n-2}，y_{n-r} を要するわけであるから，初期値 y_0 のほかに出発値として y_1，y_2，……y_{r-1} をまず求めないと y_r が求められない。したがって，出発値の計算には1段階法を補助的に使うことが必要であることを具体例で説明するとよい。

2.6 ハミング法

予測子・修正子法の一つであるミルン法やアダムス法が予測子と修正子の差が所定の許容範囲に納まるまで反復を繰返させるのに対し，ハミング法では計算時間の短縮を考え，反復を行なわせずにまず予測子を求め，これを補正子を用いて補正し，この補正子に対して修正を施し，この修正子と予測子から直接解を得るようになっている。以上のことを簡単に例をあげて説明する。

また，ハミング法もミルン法やアダムス法といった予測子・修正子法と同様に，初期値 y_0 から積分を続けることはできないので，ルンゲ・クッタ法などを用いて予め出発値を求めてやる必要があることを述べておくとよい。

指導上の留意点

ここでは，コンピュータを利用して微分方程式を解くための，もっとも一般的な数値計算法がルンゲ・クッタ法であることを強調し，その内容をよく理解させることに重点をおく。他の計算法については，その背景を理解させる程度にとどめ，手法の長所，短所を比較して，概説する程度でよい。また，実習にはルンゲ・クッタ法とミルン法をとりあげるとよい。

第 3 章　線形および非線形連立代数方程式と固有値

用　語　この章ではつぎの用語を教える。

正則行列，階級，虫くい行列，ガウス・ジョルダンの消去法，枢軸操作，ガウス・ザイデルの反復法，弛緩法，ニュートン・ラフソン法，ヤコビ行列，擬似逆行列，最小2乗解，帯行列，LDU分解法，クラウト法，固有ベクトル，べき乗法，ヤコビ法，ダニレフスキー法

目　標

線形および非線形連立代数方程式の行列解法と固有値の求め方の基礎事項を修得させることが，この章の目標である。

まず基礎となる行列およびベクトルの演算（加・減・乗算）を説明し，これらのサブプログラムを作成させることによって具体的な内容と行列演算に関するプログラミング技法を修得させる。次に線形連立代数方程式の解法としてガウス・ジョルダンの消去法を説明し，枢軸操作を理解させる。引き続いて弛緩法であるガウス・ザイデル法を説明し，プログラミング実習をさせるとともに，その収束性についての認識を与える。非線形連立方程式の解法としては，代表例としてニュートン・ラフソン法を理解させる。更に最小2乗解の意味を回帰分析などを例にとって説明し，その解法として擬似逆行列を理解させる。また実習問題では問題の次元が非常に大規模であることに鑑み，大規模問題の解法に用いられる代表的な技法の大略を理解させる。最後に行列の固有値について，その物理的意味の応用について重点的に説明し，具体的な解法については，べき乗法，ヤコビ法およびダニレフスキー法の大略を理解させるに留める。

内　容

3.1　行列演算の基礎

線形計算の基礎となる行列および行列式の基本的な性質および用語の理解は本章の内容を修得するための必要条件である。本節の基礎的な内容については基礎編の科目「情報処理用数学」第2章ベクトルと行列で解説されているが，ここでは若干の重復は復習という意味で許すことにして次の事項を解説する。

① 行列の定義と行列間の基本的な演算。

行列間の和，差，積および行列とスカラの積，スカラ（パラメータ）による微分，積分を説明する。これにはまず3×3次元程度の実行列をいくつか用意して，基本演算を数値例によって

実際に示すとよい。

② 実行列の転置，逆行列の定義，**正則行列**，対称行列，単位行列，対角行列などの解説。

これらの特徴をもつ行列の定義と例をまず実行例についてあげ，次に複素行列について説明する。

③ 行列式の定義と性質

行列式の定義と性質を簡単な数値例によって説明する。内容としては次にあげる用語の意味が理解される程度とする。

行列式の値，小行列，余因子，**階級**（rank）

3.2 行列，ベクトルの基本演算プログラム

行列およびベクトルの基本的な演算の定義を理解させ，実際の数値計算を行なうサブプログラムを作成させる。具体的には次のようなサブプログラムを作成するよう指導する。

行列として： $\mathbb{A}$ (m×n)， $\mathbb{B}$ (m×n)， $\mathbb{C}$ (m×n)，

$\mathbb{T}$ (m×p)， $\mathbb{U}$ (n×p)

ベクトルとして: $\mathbb{X}$ (n×1)， $\mathbb{Y}$ (n×1)， $\mathbb{Z}$ (m×1)

スカラとして: S

を与え（数値例としては4～6次程度でよいであろう），次のような演算をするサブプログラムを作成させる。

$\mathbb{T} \leftarrow \mathbb{A}\mathbb{U}$	（行列と行列の積）
$\mathbb{C} \leftarrow \mathbb{A}+\mathbb{B}$	（$\mathbb{A}-\mathbb{B}$） （行列の和，差）
$\mathbb{Z} \leftarrow \mathbb{A}\mathbb{X}$	（行列とベクトルの積）
$\mathbb{Y} \leftarrow S\mathbb{X}$	（ベクトルのスカラ倍）
$\mathbb{B} \leftarrow S\mathbb{A}$	（行列のスカラ倍）
$S \leftarrow \mathbb{X}^t\mathbb{Y}$	（スカラ積）
$S \leftarrow \lVert \mathbb{X} \rVert$	（ノルム）
$\mathbb{V} \leftarrow \mathbb{A}^t$	（行列の転置）など

3.3 線形連立代数方程式

線形連立代数方程式は，工学分野のみならず経済，統計など，非常に広範な分野で遭遇するものであることをまず認識させる。次に実際的な規模の問題では，変数が数十から数百になることは稀でないことを示し，手計算で用いられる方法（特に逆行列を求めるクラメルの方法）には限界があり，また必ずしもコンピュータ向きでないことを理解させる。

計算手法としては，

① ガウス・ジョルダン（Gauss－Jordan）の消去法

② ガウス・ザイデル（Gauss－Seidel）の反復法

③ ニュートン・ラフソン（Newton－Raphson）法

などの代表的な手法があり，行列の特殊性（対称行列，**虫くい行列**（ Sparse matrix ）など）を生かした技法が種々開発されているが，これらをすべて網羅的に説明することは，時間的制約からしても不可能であろう．また，実際に計算を行なうときには，十分に専門家によって検討され，完成したプログラムを利用することが多い。しかしながら，手持ちの問題がどのような特徴をもっており，どの方法を利用するのが得策であるかを選択できる能力を養成することは重要なことである。このためにも，前掲した代表的な手法の大略と長所，欠点を知っておくことが必要である。(各手法の内容については3.4以下で説明する。)

線形連立方程式は，ただ1回だけ特定の係数行列と条件（右辺の値）について解くことは稀であり，条件をいろいろと変化して解の値を検討したり，あるいは係数行列の一部が変化した場合を検討することが多い。また係数行列の逆行列が必要であるかないかによっても解法が異なってくる。3.4以後の説明は，常にこのような問題の特殊性を認識させるように努めて進めていく必要がある。

3.4 ガウス・ジョルダンの消去法

ガウス・ジョルダンの消去法を，まず3次元程度の場合について数値例を示して説明する。最初の説明では**枢軸**（ピボット）**操作**を中心にし，次に最大枢軸要素をつくるための行の入れかえの必要性と具体的な方法を説明する。

引き続き，行列Aの逆行列の計算が拡張された行列〔A : I〕（Iは単位行列）に対して枢軸操作を行なうことによってなされることを説明する。この場合も行列Aは3次元程度の簡単な数値例でよい。

ガウス・ジョルダンの消去法は，最大枢軸を選ぶ場合でも，FORTRANで70～100ステートメント程度で書けるので，実際にプログラムを作成する実習を行なうとよい。この場合，研修生のプログラミングの能力に応じて，出来るだけ一般性のあるサブプログラム（例えば可変のディメンジョンを用いたもの）を作成させるようにする。

次にガウス・ジョルダンの消去法では逆行列A^{-1}と同時に行列Aの行列式の値 det〔A〕の値も容易に得られることを説明し，上述のプログラミングの実習の中に含めさせる。また行列式の値が零，あるいは零に非常に近いとき（nearly singular）は逆行列の精度は低下することを，nearly singular な行列をインプット・データとして与えて実際に確かめさせる。（すなわちAA^{-1}の計算を2.2で作成したサブプログラムで計算させ，結果が単位行列Iに一致するかどうかをチェックさせる。）

3.5 ガウス・ザィデルの反復法

3次元程度の単純な数値列（3.4と同一の例題でよい）を用いて，**ガウス・ザイデルの反復法の**アルゴリズムを説明する。次にガウス・ザィデル法は**弛緩法**（relaxation method）の一種であることを説明し，弛緩法の考え方を一般的に理解させる。ガウス・ザイデル法は中間結果を逐次右辺に代入していくだけであり，FORTRANプログラミングは非常に簡単であるので，必ず実習をおこなわせるようにする。次にガウス・ザイデル法の収束性の問題であるが，これを一般的に説明することはかなりの予備知識を必要とするので，対角要素と小さくした数値例をつくって手計算で発散する様子を示し，次に収

束が保証されている場合をいくつか示しておけばよい。 すなわち，①係数行列が正方で対称。②非対角要素の絶対値の行の和が各行とも対角要素より小さい。など代表的なケースを紹介しておく。

3.6 ニュートン・ラフソン法

ニュートン・ラフソン法は第4章「代数方程式」の根を求める方法としても用いられるが，実用面からは連立非線形方程式の解法として有力な手法である。

説明の便宜上，まず1変数の場合についてニュートン法のアルゴリズムを図解により説明し，収束速度と収束しない場合について解説する。次に連立方程式，すなわち，多変数の場合についてのニュートン・ラフソン法のアルゴリズムを説明する。ここでは**ヤコビ行列**（Jacobian）の解説が中心となる。引き続き，2変数程度の数値例（3.5に示した例題でもよい）を用いて収束性がよいこと，しかしながらヤコビ行列の逆行列に計算時間を要することなどを理解させる。ニュートン・ラフソン法のプログラムは，ヤコビ行列（及びその逆行列）が問題によって異なり，一般性がないので，プログラム実習は必要でない。

3.7 擬似逆行列（一般化逆行列）

擬似逆行列とその応用について解説する。具体的には次の順序で説明するとよい。

① 係数行列が正方でない場合，あるいは正則でない場合に普通の意味での逆行列は存在せず，従って解が存在しない（あるいは解が不定である）こと。これには2×3次元，および3×2次元程度の小規模の数値例を示して不定（Under determined）および不能（over determined）の意味を説明するとよい。

② 次に上述のような方程式が実際にはしばしば現われることを示す。これには実験結果の曲線のあてはめ，あるいは回帰分析を例にとって最小2乗解が必要となる理由を示すとよい。

③ **最小2乗解**を求めるために，擬似逆行列（Pseudo inverse）が必要になることを示し，具体的に右方擬似逆行列（right pseudo inverse）および左方擬似逆行列（left pseudo inverse）の定義と計算法を示す。この場合にも，まず簡単な例によって擬似逆行列の幾何学的意味（例えば原点からの2乗距離が最小）を十分理解させるように努め，擬似逆行列が数学的に難解なものではなく，むしろ明解な幾何学的意味をもっていて，非常に実用的なものであることを認識させる必要がある。

3.8 大規模な問題に用いられる技法

コンピュータで解かれる問題の規模は最近非常に大きなものになっており，従って問題の特殊性を利用した技法が必要になってきている。ここではその中で代表的なものの概要を説明し，実際問題に遭遇したとき，どの技法を選択すべきかを決定できる能力を養う。

具体的には次の点に簡単にふれる程度でよい。

① 係数行列の特殊性を利用するものとして

- **帯行列**（主対角要素の近所以外は零）あるいは帯行列に近い場合

・零要素の多い，いわゆる虫くい行列 の場合。

② **LDU分解法**の一つとしての**クラウト法**（Crout's method）

③ 係数行列の要素の一部が変化したときの逆行列の迅速な計算

④ プログラム技術として，演算回数，記憶容量の減少の技術

3.9 固有値と固有ベクトル

固有値や**固有ベクトル**の概念はどちらかといえば理論的なものであるととられやすいので，まず単純な例（簡単な電気回路の振動など）をとって，固有値が振動数（固有振動数）に，また固有ベクトルが固有振動波形に密接な関係があることを理解させる。

次にこの例を一般に拡張する形で固有値および固有ベクトルがどのようなものであるかを説明する。この説明に当っても，単に

$$\mathbb{A}x = \lambda x$$

のxが固有ベクトルであり，λが固有値であるという行き方では理解されにくいので，3次元程度の数値例を通して解説をした方がよい。

固有値および固有ベクトルにはいくつかの性質があるが，代表的なもの（対称行列の固有値は実数である。$\mathbb{A}$と$\mathbb{B}^{-1}\mathbb{A}\mathbb{B}$は同一の固有値をもつ等）は説明をしておく必要がある。（何故ならば，固有値の数値計算法はこれらの性質のいずれかを利用しているからである。）

研修生が工学関係の基礎知識，とりわけ，電気工学あるいは制御工学の予備知識をもっている場合には，特性方程式の根という形で固有値を説明した方が理解されやすい。例えば単純な制御系のステップ入力に対する時間応答や安定性と固有値や固有ベクトルの関係を理解させる方法をとるとよい。

固有値の数値計算法としては，**べき乗法**（power method），**ヤコビ法**（Jacobi's method），**ダニレフスキー法**（Danilevski's method）などいくつかの方法が用いられているが，これらの中にはすべてのタイプの行列には適用されないものもあり，各手法には長所欠点があることを理解させることが重要である。（例えばヤコビ法は対称行列にしか適用できない。べき乗法は最大固有値を求めるのに能率的である。ダニレフスキー法はどのような行列にも利用できる等）。従って具体的な計算アルゴリズムの詳細な解説は不要であり，各方法の大略を2次元程度の簡単な例で数値的に説明する程度で十分である。

指導上の留意点

(1) 3.1の主要な目的は，3.2以後の数値計算のアルゴリズムを理解させるための必要最小限の用語の意味を理解させることである。従ってあまりに網羅的にすぎる説明は避け，もし欠ける場合があったら3.2以下で必要に応じて補うようにする。

(2) 行列の考え方は技術系の知識をもつ研修生には容易に理解されようが，その他の研修生に対してはかなり基礎的な解説から始める必要があろう。

(3) 行列演算をプログラム化するとき，FORTRANで記述するときは必ず配列（array）と次元

（DIMENSION）についての充分な理解が必要になってくるので，FORTRAN言語の説明と十分内容的に協調をとって行なう必要がある。

(4) 行列計算は，プログラム作成の実習では通常2次元配列を用いるが，計算処理効率を高めるため，一次元配列を用いて行なうことが多い。行列の一次元配列を行なう方法はライブラリによって同一ではないので，特に行列の大きさ（ディメンジョン）が固定でなく，変化する場合に使用上，十分の調査と注意が必要であることを認識させる。

(5) ガウスの消去法については，ガウス・ジョルダンの消去法との差異（すなわち前進または後進消去が必要であること）を示すに留める。

(6) プログラミング実習において，最大枢軸を選択する操作が研修生にとって複雑すぎると判断される場合には，この手順を省略してもよい。ただし，枢軸要素の値をスケーリングする手続き（規格化）は必ず行なわせる。

(7) 同じ係数行列$/A$に対して何組かの異なった条件（右辺の値$/B$）を解く場合，逆行列$/A^{-1}$をも求めて$/A^{-1}/B$によって解を求める方法はとらず，右辺の計算だけを個別に処理することが出来ることを説明しておく必要があろう。

(8) ヤコビ法（Jacobi's Method）についてはガウス・ザイデル法との差異（すなわち同一の繰り返しの間は前の繰り返しの値を用いる）を説明するだけでよい。たゞし収束性を議論するにはガウス・ザイデル法よりも数学的に容易であるので，研修生にある程度の行列の知識がある場合には，ヤコビ法の収束条件を説明するとよい。（ガウス・ザイデル法の収束条件もほゞ同一である。）

(9) ガウス・ザイデル法は収束速度が小さく，従って収束性を改善するために加速（acceleration）法が併用される。加速法にはSOR法（Successive Over Relaxation＝逐次過大修正法）やチェビシェフ加速などいろいろな方法が考えられているが，ここでは最も単純な方法（修正量を定数倍する）について触れるに留める。

(10) ガウス・ザイデル法（あるいは一般に弛緩法）は，線形連立方程式よりも非線形の場合に用いられることが多いので，この事を理解させるために簡単な非線形の例について説明するとよい。（例えば次のような例が適当であろう。）

$$\begin{cases} \dfrac{1}{2}\sin(x_1 x_2) - \dfrac{x_2}{4\pi} - \dfrac{x_1}{2} = 0 \\[2ex] \left(1 - \dfrac{1}{4\pi}\right)(e^{2x_1} - e) + \dfrac{e^{x_2}}{\pi} - 2e^{x_1} = 0 \end{cases}$$

（3回程度のくり返しで収束する。）

(11) 非線形方程式をニュートン・ラフソン法で解く場合，初期値によって異なった解が得られることがあることを数値例によって示すとよい。

(12) ヤコビ行列の逆行列の計算が計算時間の大部分を占めることを理解させることが大切である。

(13) 擬似逆行列（あるいは一般化逆行列）という用語を中心に数学的に議論をすすめていくことは理解されにくいので，「最小2乗解」を求めるという問題を前面にだし，その目的のための手法とし

て擬似逆行列の考え方が必要になるという順序で説明することが肝要である。

(14) 擬似逆行列は別に特殊なものでなく，具体的な計算は逆行列と行列の積で得られるので，3.2～3.4までの内容と十分関連を保つことが必要である。

(15) 3.8の①～④までの事項は詳細に説明することは時間的制約から不可能であるので，概略的に述べればよい。（場合によっては，①～④のような場合に遭遇したとき，これを解決する「何らかの手法」がすでに開発されているということを研修生の記憶に留めさすだけでもよい。）しかしながら，3.8の内容を完全に省略することは危険である。

(16) 固有値や固有ベクトルの考え方は，電気，機械，建築，船舶，音響など振動学が用いられる分野には必ず現われるものであり，科学技術の計算センタなどでは，しばしば計算が要請されるが，内容的にはやや高度で専門的であるので，各計算手法のいずれを選択すべきかが判断できる能力を養成することが，特定のアルゴリズムの細部にわたる理解よりもむしろ大切である。

第 4 章　代数方程式

用　語　この章ではつぎの用語を教える。

多項式，超越関数，はさみうち法，ニュートン法，ホーナー法，ヒッチコック・ベアストウ法，不動点

目　標

この章では技術，経済問題に生ずる方程式の根を求めるにあたって必要な知識を身につけさせることを目標とする。まず方程式の性質を多項式，超越関数について理解させ，根を求めるための探索法について全般的に紹介し，はさみうち法，逐次代入法，ニュートン法，ホーナー法などの方法とそれらの特徴を理解させる。複素根を求めるための方法を概観するとともに，一例としてヒッチコック・ベアストウ法について紹介し，プログラムを組ませてみる。このようにして，実根，複素根のいずれの問題も求められるようにする。また，それらの各手法の長所，欠点を幾つかの例をあげて示し，取るべき手法の選択力を身につけさせる。

内　容

4.1　方程式の根

技術，経済の問題を解く時に必要になる方程式の根とは $f(x)=0$ を満たす x のことであるが，$f(x)$ として1次式，2次式，三角関数（例えば $x-\tan x=0$）などを示し，**多項式，超越関数**の区別を明らかにさせる。

関数 $f(x)$ の零点が $f(x)=0$ の根であることを教え，根には実根，複素根のあることを2次式の例で示し，実根の時はグラフ上でx軸との交点の形で根の座標が示されることを理解させる。

実根を求めるための方法としては

① グラフを描いてみること

② 試行錯誤による方法

③ 試行錯誤で出発点をさがし，そこより解析的な方法で根に近づいて行く計算をくり返す

以上の3通りの考え方のあることを示す。②，③ではうまく根に収束しないで発散することもあり得ることを図で理解させる。

4.2　多項式の性質

多項式の定義を示し（$f(x)=a_1x^n+a_2x^{n-1}+\cdots\cdots+a_nx+a_{n+1}$，$a_1\neq 0$），根がその次数（ここではn）と等しい個数あること（等根も含めて）を示す。根には実根，複素根のあることを2次式などで示す。重根（例えば $x^2-1=0$）についても説明しておく。また，$x=a+b_j$，

$y = a - bj$ なる二つの根 x，r に共役複素数と呼ぶことを教え，複素平面上で図示して理解させる。

また多項式の正，負の多数の根を求めるさいデカルトの符号法則を知っておくと便利なので時間と研修生のレベルに応じてふれておくのもよい。

多項式における根と係数の間の重要な関係としてニュートン法の根と係数の関係を理解させたい。

$$(x-r_1)(x-r_2)(x-r_3)(x-r_4) = x^4 - (r_1+r_2+r_3+r_4)x^3$$

$$+ (r_1r_2+r_1r_3+r_1r_4+r_2r_3+r_2r_4+r_3r_4)x^2$$

$$- (r_1r_2r_3+r_1r_2r_4+r_2r_3r_4)x + r_1r_2r_3r_4$$

なる4次式程度までの例から根と係数の関係を理解させるとよい。これから最大根（絶対値）の上限を定めることのできることを示してやる。

すなわち $|r|_{max} \leqq \sqrt{\left(\frac{a_2}{a_1}\right)^2 - 2\left(\frac{a_3}{a_1}\right)}$ である。これは数値例で実際の根を求めて比較してやると理解させやすい。

最後に多項式の根の内1根 r_1 がわかれば次式を $f(x)/x-r_1$ なる操作で下げてやることができることを示し，このような考えで，根を次々と求めて行くことができることを理解させる。

4.3 超越関数の性質

超越関数とは任意の独立変数の有限値の和，差，積，商，根の開放によって定めることのできない関数であるが，たとえば三角関数，対数関数，指数関数，双曲線関数などの例をあげてしたしさを覚えさせる。また超越関数を含んだ方程式を超越方程式と呼ぶが，$x - \cos x = 0$，$x^2 - \sin x = 0$ などの例をあげて根の種類，個数などに注意すべきことを示す。実根についてはグラフによるのが，全般的な問題の理解に重要であることを示す。また数値計算法として多項式による展開もあり得ることも示してやるとよい。

4.4 はさみうち法

方程式の根を決定する第1段階は根の数と実根に近い数値を知ることである。グラフを描けば実根についての個数とおよその値を知ることができる。コンピュータによっても同じような考えが適用できることを図で示す。

最初x軸に沿ってある幅を定め，その幅毎に x を増大させて行き，関数値の符号が逆になる2点を求めることにより根をとらえることができる。幅の選び方は根の上限を係数から求め目安とすることができる。また，グラフから目安を得るのもよい方法である。

根を求めるための代表例として**はさみうち法**（レギュラー・ファルサイ法）を紹介する。この場合，図4－1のように歩幅 Δx のとり方に注意しないと，どうしても必要な根が求められないことがあるのに注意させる。この問題はわざと図4-2のような $f(x)$ を与え，Δx も大きな値を使って，手計算またはプログラミングをさせてみる。発散するのはどうしてかを考えさせて，最後にこの図で図式的に説

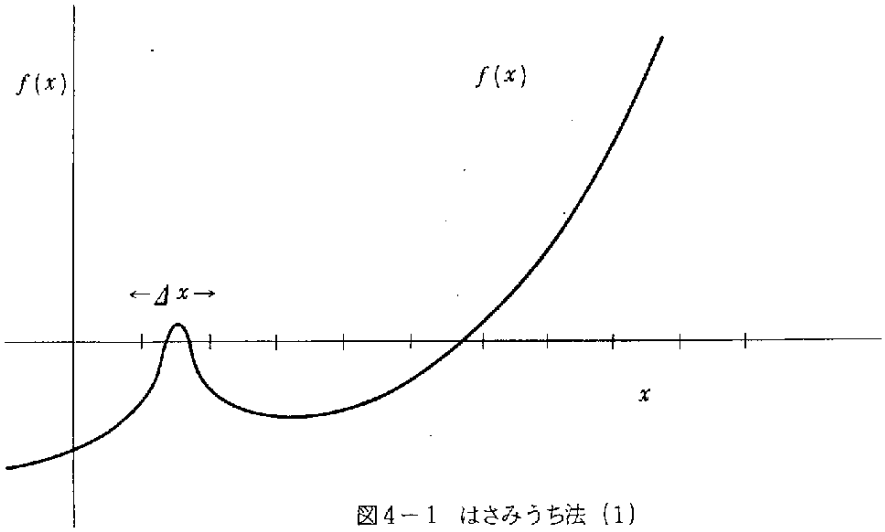


図4－1　はさみうち法（1）

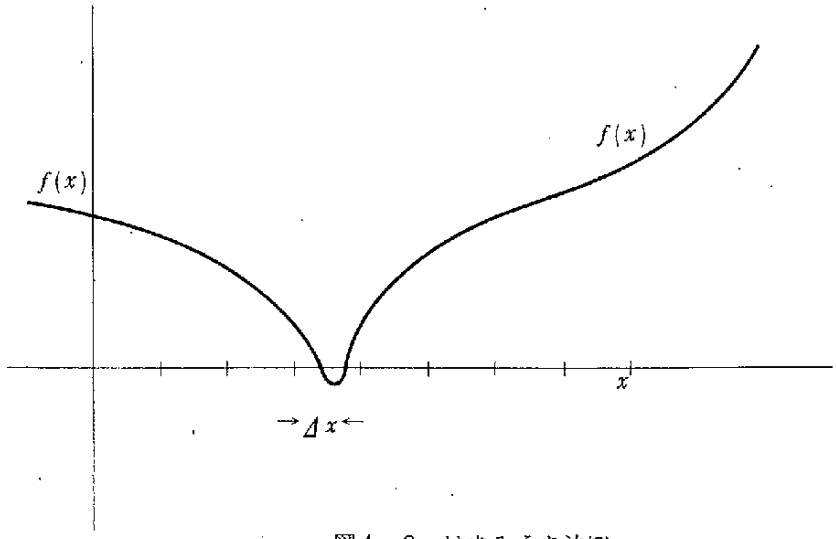


図4－2　はさみうち法(2)

明するとよく理解させることができる。

根が x_n ，x_{n+1} の間にとらえられたら，以後 x_n から歩幅 Δx を細かく（例えば $\frac{1}{2}$ 倍とか $\frac{1}{10}$ 倍）してしらべて行くことにより根の精度をより高くして行くことのできることを示す。またこの方法では一たび根 x がはさみ込まれれば,必ずその根を求めることができることを理解させる。

歩幅を細かくする場合 ½ 倍が平均的には最も有効であることを示す。

また2点間にはさみ込まれたあとのつめとして，補間法を使うのも有効であることを図で示す。実計

算においては精度をどこまでとるべきかにより必要な計算ステップが増加することを実際に示しておきたい。

4.5 ニュートン法

方程式の根を決定するために最も活用されている方法であることを念頭に，近似の式の説明をする。

$$x_2 = x_1 - f(x_1) / f'(x_1)$$

の式はテーラー展開などの式から説明すると式を暗記することを必要とするので好ましくない。
図4－3のような図により説明し，上記の式を自分で導き出せるようにする。

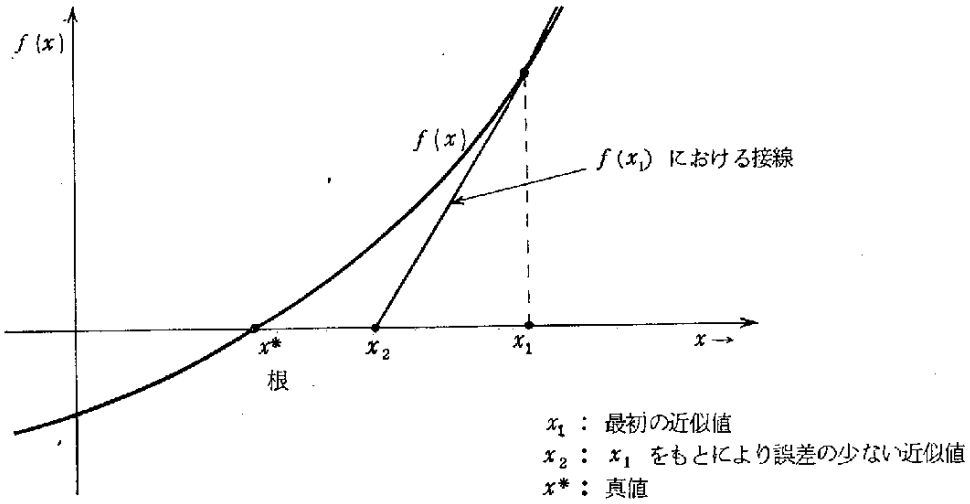


図4－3　ニュートン法

平方根 $\sqrt{a}$ は　$f(x) = x_2 - a = 0$ の根として求められることを示し，**ニュートン法**で解かせてみる，近似式における符号をまちがえる研修生が非常に多いので，手計算の際には必ずチェックすることの必要性を充分強調する。

このような反復計算をどこまでくり返すかは，誤差との関連で重要となることを数値例で示しておきたい。

またこの方法による発散する可能性として次の図を用いて説明する。

発散することはめったにないが，あり得ることを覚えておくことは必要である。時間があったら発散するような問題をやらせてみて $f'(x_1) = 0$ または微少な値となるため x_2 が $+\infty$ または $-\infty$ になることを確かめさせるのもよい。

4.6 ホーナー法

多項式における組立除法をまず実例で示し，これをニュートン法の $f(x_1)$，$f'(x_1)$ を求めるのに都合よく利用できることを示す。組立除法は自分で何回かやってみることが必要で，なるべく自習を多くやらせたい。また，$f'(x_1)$ を求める時の説明は微分についての概念を必要とするので研修生のレベルに応じた補足が必要である。

ホーナー法のプログラム作成程度にやれるようにしたい。

4.7 逐次代入法

$$f(x) = 0$$

を適当に変形して

$$x = F(x)$$

とした上で，ある初期値 x_0 から出発して，

$$x_{k+1} = F(x_k)$$

をくり返し，収束すれば根となることを図によって説明し理解させる。この方法は特別な公式は不要であり，$x = F(x)$ といった単なる元方程式の変形のみでよい点が利点である。

例えば　$x^3 - 23x^2 + 62x - 40 = 0$　を

$$x = 23 - \frac{62}{x} + \frac{40}{x^2} = F(x)$$

とし　$x_1 = 21$ として　$x_2 = 20.1$，　$x_3 = 20.0$（根は 20.0）として計算できることを示せばよい。

ただし，この場合　$x_1 = 21$ としても　$F(x)$ として

$$x = \frac{x^2}{23} + \frac{62}{23} - \frac{40}{23x}$$

とすると，こんどは $x_2 = 21.8$，$x_3 = 23.2$，$x_4 = 26.1$，$x_5 = 32.3$ と発散することを示したい。この場合 $x_1 = 18$ とすると収束はするが，根は 2.00 と別のものになることに注意したい。

この方法は手法として簡便であり，多項式以外にも適用できる便利さがあるが，$F(x)$ の形によっては発散することの多い手法であることをよく注意させる。$|F'(x)| < 1$　が収束の条件であるが，図4－4のような代表的な四つの例をあげて，収束の条件を理解させたい。

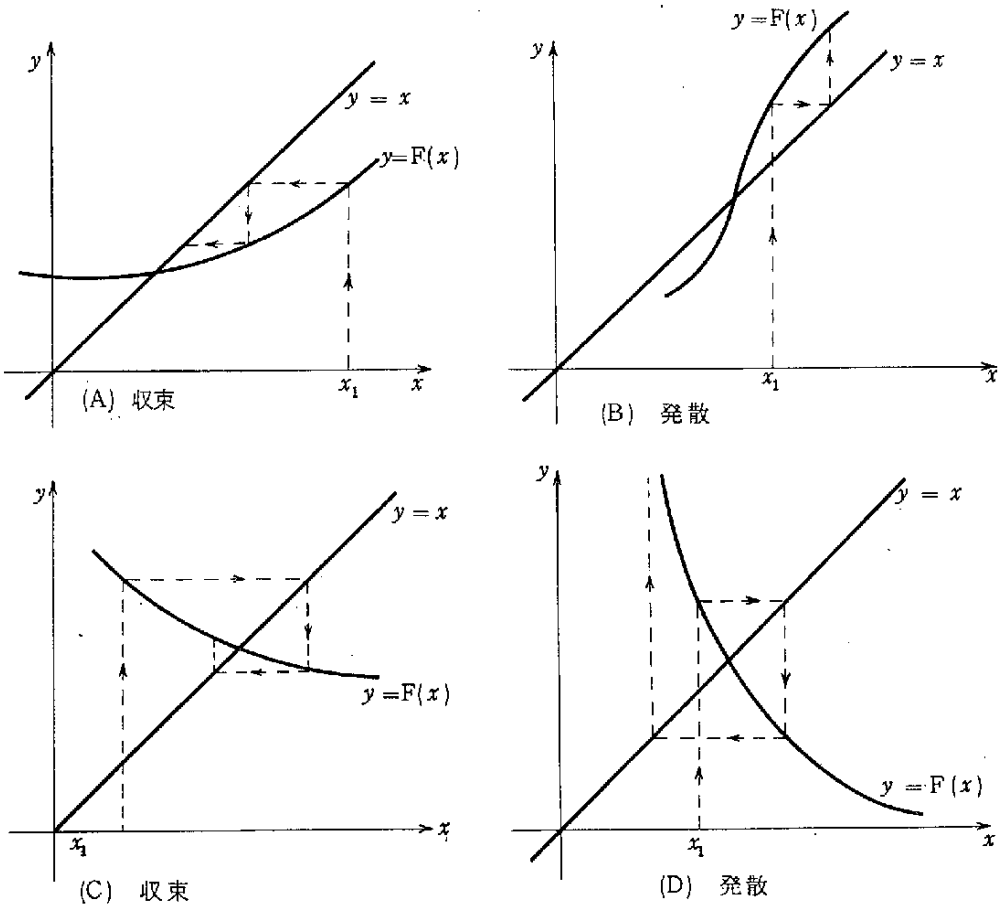


図4－4　収束の条件

4.8 複 素 根

これまでの方法は主に実根を求めるものであるが，多項式の複素根を求める問題を考えさせる。多項式は実係数の1次式と2次式に因数分解できること，2次式の根は共役根となることをまず説明する。

ニュートン法でこの問題を解くことができることを示す。例えば

$$f(x) = x^4 - 6x^3 + 18x^2 - 24x + 16 = 0$$

で初期値として $x_1 = 3 + 3i$ を使うと

$$x_2 = 2.58 + 2.45\,i$$
$$x_3 = 2.27 + 2.13\,i$$
$$x_4 = 2.07 + 2.00\,i$$
$$x_5 = 2.00 + 2.00\,i$$

となり $2+2i$ と $2-2i$ の2根を知ることができる。ニュートン法でこのような計算をする場合近似値 x_2 の実数部と虚数部のそれぞれの相対誤差をチェックすることが必要であることを理解させる。

ここで組立て除法を使って解く方法として次の項の**ヒッチコック・ベアストウ法**を紹介する。

4.9 ヒッチコック・ベアストウ法

高次の多項式の複素根は1対の共役根として存在することを利用し，2次の因子を探して行く方法が考えられること，そこに組立て除法を適用することが，この方法の原理である点を念頭において説明する。

この手法はかなり原理的にめんどうであり理解しにくいので，研修生のレベル，その他に応じて原理はカットしてもかまわない。考え方の基本を理解させれば充分であろう。

与えられたプログラムでの数値計算の演習は経験させておきたい。

4.10 収束と各種手法の比較

以上にのべた多くの反復式は

$$x_{k+1} = \varphi(x_k)$$

という形をしていたことを示し，φ がどのような形であるかを比較してみせる。これにより反復計算の特徴を理解させる。

例えばニュートン法では

$$\varphi(x) = x - \frac{f(x)}{f'(x)}$$

であり逐次代入法では

$$\varphi(x) = F(x)$$

である。

$$w = \varphi(w)$$

となる w を**不動点**であることを説明し，これが $x = \varphi(x)$ の根であることから収束の意味を概念的に理解させる。

一般的な収束の条件（$\varphi'(x_k)$ が連続で，$|\varphi'(x)|<1$）を説明し，ニュートン法，その他の例で数値例をあげる。

続いて反復式の次数を定義し，逐次代入法，ニュートン法などの次数を計算してみる。1回の反復計算で得られる精度にその次数の差が現れることを実例で示し，研修生に理解させる。収束は各種方式の選択に重要な問題であるが，理論的に理解させるのは大変なので，一般的に理解させる程度でよい。

指導上の留意点

(1) 反復計算により根に段々近づけて行くという考え方は自動制御の分野でもみられる身近な現象であるが，数値計算法になれた者には，理解しにくいので，基本的な考えを具体的な数値をあげて充分理解させたい。

実根ではグラフ的な表現を使って理解させやすいが複素根では多少の説明が必要である。

(2) 収束の問題は必要条件のもつ意味を理解させる。数学的にはめんどうな点も多いので，一般的な説明で，数値例を上げて理解させるように努力すべきであろう。

(3) 実習には，ニュートン法とヒッチコック・ベアストウ法をとりあげるとよい。

第5章　補間法と数値微分・数値積分

用　語　この章ではつぎの用語を教える。

グレゴリー・ニュートンの補間式，差分，前進差分，後進差分，中心差分，補間多項式，台形法則，シンプソンの法則

目　標

この章では，補間法について，その必要性と各種手法を理解させるとともに，その基礎となる差分の概念を把握させ，数値微分，数値積分の計算法を各々における精度について理解させることを狙いとしている。

まず，内容を大きく，補間法，数値微分，数値積分に分け，各々について，具体的な手法を説明するとともに，各々の手法における誤差の問題を取り上げ，研修生に各種手法の計算方法とそれぞれの誤差の大きさなどを理解させる。

内　容

5.1　補間法

実験データなどのように表には明記されていない独立変数xの関数値$f(x)$を，表に与えられている関数値を結んだ曲線より求める方法を補間法というわけであるが，いくつかの例をあげて補間が必要となることを実感させる。

その場合補間法を実行するに当ってはグラフによって求めるのが最も簡単であるが，数値的には多項式による方法が使われることを示す。

一例として補間放物線による方法などを示し，具体的な計算例について説明する。それにより，次のグレゴリー・ニュートンの方法等を理解するための基礎を与える。

5.2　グレゴリー・ニュートンの補間式

一般の補間方法は（$n+1$）点を通るn次の多項式$P(n)$をつくることであるが，これは式$P(u)$として以下のように与えられる。

$$
\begin{aligned}
P(u) = f(x_1) &+ \frac{\Delta f(x_1)}{1!}u + \frac{\Delta^2 f(x_1)}{2!}u(u-1) \\
&+ \frac{\Delta^3 f(x_1)}{3!}u(u-1)(u-2) + \cdots\cdots \\
&+ \frac{\Delta^n f(x_1)}{n!}u(u-1)(\cdots\cdots)[u-(n-1)]
\end{aligned}
$$

多項式$P(u)$をこのように表わすのを**グレゴリー・ニュートンの補間式**と呼ぶ。簡単な計算例を示して説明する。

5.3 差　分

補間法，数値微分等で**差分**の概念が必要となるので，ここで全般的に差分の概要をのべる。有限の差分についての概念を図などでまず示し，**前進差分**，**後進差分**，**中心差分**についての相異を理解させる。

続いて差分表により1階差分，2階差分，3階差分……について表で示す。

補間法への適用法として前進差分，後進差分，中心差分の特徴を示す。

また多項式における差分の特徴を示し，その活用の可能性を理解させる。

5.4 ラグランジュの補間法

ラグランジュの**補間多項式**を示し，そのプログラムについても触れる。

等間隔，不等間隔の場合で，いくつかの例をあげて理解させる。

5.5 補間における誤差

補間における誤差はどのようにして生ずるか，またその大きさはどの程度になるか，どのようにしたら収束していくか，誤差の次数などを示しておく。

補間において誤差とも関連して特に注意すべき点として，

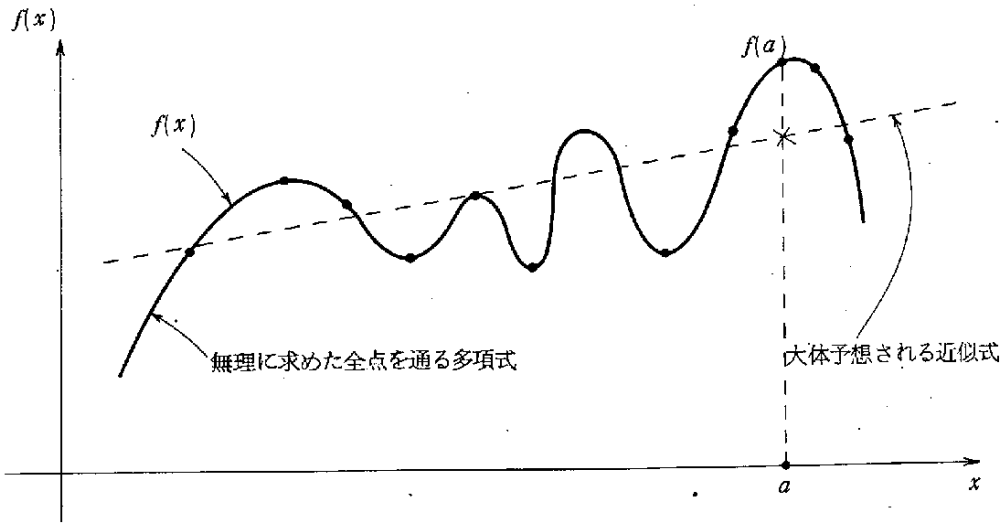


図5－1 補間における誤差

図5－1のような実験データが得られた場合，これを無理に全点を通る多項式で近似すると，aに対し

$f(a)$という値が得られるが，この場合はむしろ図中のx点の方がより近い値として信用できる筈である。

これは補間法そのものの問題にたちもどって考えるべき問題であるが，適用に当って充分注意することの必要性を強調しておきたい。

5.6 数値微分

グラフで微分の概念を復習させ，数値的にはどのようにして求めるかを教える。

多項式で求める方法としてグレゴリー・ニュートンの方法を紹介しておきたい。

5.7 数値微分の誤差

数値微分に含まれる誤差の表示形は，次のように理解させる。すなわち真の関数値$f(x)$と対応する補間値$P(x)$と補間多項式の誤差との間の関係は

$$f(x) = P(x) + e(x)$$

で，微分すると

$$f'(x) = P'(x) + e'(x)$$

ここで$e'(x)$は$f'(x)$の代わりに$P'(x)$を用いたために生じた誤差である。

$e(x)$ の形式，$e'(x)$ の形式から

$$e'(x_1) = \frac{f^{n+1}(\xi)}{(n+1)!}(x_1 - x_2)(x_1 - x_3)(\cdots\cdots)(x_1 - x_{n+1})$$

のような式が得られる。

一般に補間法により得られる場合より誤差が大きいことを示しておきたい。

5.8 数値積分

数値積分の必要性の多いこと，その意味を理解させる。**台形法則，シンプソンの法則**の両者を，数値例をあげて示し，更に3次，4次式などでも近似して積分できることを教える。

係数は次のようである。

次数	式	
1	$(h/2)(y_1 + y_2)$	台形法則
2	$(h/3)(y_1 + 4y_2 + y_3)$	シンプソンの法則
3	$(3h/8)(y_1 + 3y_2 + 3y_3 + y_4)$	
4	$(2h/45)(7y_1 + 32y_2 + 12y_3 + 32y_4 + 7y_5)$	
5	$(5h/288)(19y_1 + 75y_2 + 50y_3 + 50y_4 + 75y_5 + 19y_6)$	

5.9 数値積分の誤差

台形法則の時の誤差は $-\dfrac{h^3 f^{(2)}(x_1+\xi)}{12}$ のような形になることを示し，続いてシンプソンの時は $-\dfrac{h^5}{90} f^{(4)}(\xi_1)$ のオーダになることを示し，いずれも担当精度はよいことを知らせる。

この場合多くの数値例を示して，生ずる誤差のオーダの感覚を身につけさせたい。

指導上の留意点

(1) 補間においては，まず多項式による考えを教えるが，実際のデータはある平均値を中心にばらついているのが多いので，単に機械的に多項式であてはめると著しい誤差のでることを図で示し，よく理解させることが重要である。

(2) 数値微分は使用することが割に少ないので簡単にふれ，その考え方を理解させればよい。

数値積分は微分方程式解法の基礎でもあり十分演習をさせたい。また積分の精度は数式が簡単な割に高いことを示し，エンジニアリングでは簡単な近似式でより高い精度を得ることが重要なことなどにもふれる。

参考文献

(1) 山内二郎共著，「電子計算機のための数値計算法Ⅰ，Ⅱ」，培風館，42年 4版

(2) 一松信著，「数値解析」，至文堂，1963年

(3) 「電子計算機による数値解析」，（財）日本情報処理開発センター

統　計　解　析

目　次

第1章　母集団と試料 …… 30

1.1　度数分布 …… 30

1.2　平均値と標準偏差 …… 31

1.3　確率分布 …… 32

1.4　統計量の分布 …… 33

1.5　演習問題 …… 34

第2章　検定と推定 …… 36

2.1　平均の検定・推定 …… 36

2.2　分散の検定・推定 …… 37

2.3　管理図法 …… 37

2.4　抜取検査 …… 39

2.5　相関と回帰 …… 40

2.6　演習問題 …… 41

第3章　実験計画法 …… 42

3.1　最小2乗法 …… 42

3.2　要因配置と乱塊法 …… 43

3.3　交絡配置 …… 46

3.4　直交配列表 …… 47

3.5　演習問題 …… 50

第4章　統計解析 …… 53

4.1　多変量解析 …… 53

4.2　最適計画 …… 54

4.3　その他の統計解析 …… 56

科目「統計解析」

教育の目標

この科目では，主として，データ解析の立場から，統計解析の適用に関する意味，計算上の注意点，実際適用に当っての手法選択の要点を教育することをねらいとする。

したがって，内容は，主として，推測統計学の範囲に限定し，決定・計画に関する統計解析の分野は，ここでは除外している。教育にあたっては，この統計解析の位置づけを示し，適用を主体に理解，演習を行う旨を述べるようにしたい。ここでは，一般の統計的方法と実験計画法に大別し，各章毎で，目標と要点を，適用本位に教述し，章の中の節では，理論の意味と計算操作上の留意点を強調するよう指導することを期待している。

また，テキストとカリキュラムとは，この育成指針のもとに別にあることが前提とされているので，個別的内容は，教育上の強調点を例示する以外には，とくに触れることはしない。

時間配分

章	履習時間（時間）		
	講義	演習	実習
1. 母集団と試料	15		5
2. 検定と推定	20		10
3. 実験計画法	15		15
4. 統計解析	10		
合計	60		30

第1章　母集団と試料

用　語　この章では次の用語を教える。

離散分布，連続分布，指数分布，Γ型分布，t分布，χ^2分布，F分布，$\overline{X}$分布

目　標

推測統計学においては，母集団をある確率分布とみなし，その分布型は分っているが，その母数値は未知である場合，データを用いて未知母数を推測する問題を扱う。データは，規定した母集団から抽出した無作為標本であると前提する。したがって，

① 母集団規定，つまり対象とする母集団の限定，諸条件の確認など。

② 無作為抽出，無作為化の重要性を例示し，データ解析が，具体的に解釈，行動を可能とさせる根拠となること。

などを理解させる。

この章では，まず母集団を，度数分布図を作成することによって理解させ，実用上あらわれる確率分布について説明し，ついで標本についての統計量で，基礎的なものについて，統計量の計算およびその分布について説明する。

内　容

1.1 度数分布

(1) 適　用

度数分布法は，諸現象や物について，それらのもつ集団的性質を，図形で分り易く表わす方法である。

度数分布のつくり方，計算方法は，比較的簡単であるが，その利用範囲は広く，かつ利用方法をまちがえなければ，大きな効果がえられる。そこで，度数分布の教育にあたっては，適用目的，応用上の注意事項を，特に強調する必要があろう。

たとえば，標識も不明な100人の身長のデータを用いて作られた度数分布を示すことによって，解釈も結論も得られぬことを理解させることも一つの方法であろう。さらに，条件の明らかな若干の度数分布（切削バイトを未修正のままにして作られた切削部品の寸法，条件が一定な工程のモデルとしての正規乱数などによる度数分布）を例示して，母集団の性質を知るためには，母集団の条件を明らかにしておくことが大切であることを理解させる。

分布の形と母集団の条件との関係とを理解させたならば，このことを基礎として，度数分布の利用法として，市場調査，工程能力調査，待ち行列の分布，ロットの検査，実験などへの応用例を示すことによって，統計解析の必要性を認識させる。

なお，統計解析が使われるためには，分布が安定し，特に正規分布が前提となっていることをのべ，えられた度数分布の形が，解釈できないようなときには，母集団の条件規定を完全におこなうことが大切である点を強調し，統計解析を利用する立場を十分理解させる。

(2) 作 り 方

ここでは，度数分布の分布型をクラスわけさせること，検算方式を明確にさせることの２点に着目して，度数分布のつくり方を演習し，理解させる。そして，母集団のパラメータを推測するためには，いかに母集団の条件を明らかにすることが大切であるかを強調する必要があろう。

工場における特性値は，大部分度数分布を作成してみると正規分布またはこれに近い分布をする。ところが正規分布をしない分布がある。この場合，技術的，統計的に注意をはらわねばならない。たとえば，切削部品の寸法をデータとして，度数分布を作成したところ，図１－１のような分布を示した。

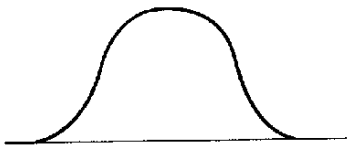


図１－１　度数分布の例

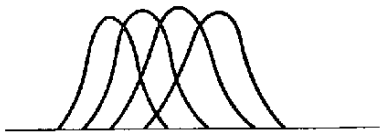


図１－２　正規分布の変動状態

これは切削バイトの調整が一定でないため図１－２のように正規分布が変動していることが推測される。したがって，切削バイトを未修正にすればよいことがうかがわれる。

以下同様に分布型をクラス分けして（正規型，歪型，Ｊ型……等），母集団の型と母集団の条件との関連の理解を深めるよう指導する。

度数分布は，大量データを扱うためにどうしても計算の誤りをおかしやすい。しかし，計算ちがいをおかしていたのでは，次にアクションをとる場合，判断の誤りをする。そこで，度数分布の計算シートを作成し，検算ができる欄をもうけ，計算ちがいをおかさないような計算手順を身につけさせるよう指導する。

1.2 平均値と標準偏差

母集団の分布の特性を表わすものとして母数があり，特に，正規型母集団の母係数は母平均と母分散である。統計解析の目的が母集団の母数を知ることにあり，われわれが取扱うことができるのは，母集団の標本であることから，その標本について，母数になぞらえてまとめた統計量すなわち平均値および標準偏差が母数の推定値の基本として重要であることを強調する。

母数と統計量との関係を明確に理解させるために，正規母集団を例示し，この母集団から抽出された標本を用いて平均値と標準偏差を求める計算手順を演習させ，母集団の一部分である少数標本から求めた平均値および標準偏差は見かけの値であって，必ずしも母数とは一致しないことを説明する。

(1) サンプリングと乱数

統計解析は標本という母集団の一部分をもってその全体を知ろうとするものであり，母集団から抽出された標本が母集団を代表するためには確率化が必要であり，そのために乱数が用いられることを説明する。例えば，特別な意味のある標本だけを抽出したのでは標本から得られる情報が母集団につ

いてどれだけ代表性をもつかについて客観性が得られないことを強調し，実際の母集団から得られた試料が，統計的に処理可能であるためには，母集団からの抽出標本が無作為化されたものであることが必要となることを強調する。

無作為抽出を行なう方法として，ランダム操作を行なうことによって得られた乱数の利用がある。

例えば，電話帳の電話番号を使って乱数表を作成させることによって乱数の概念を説明し，実際に乱数表を使っての演習をすることによりその使用方法を理解させる。また，コンピュータの発展に伴って数式を用いての大規模な乱数発生が容易になり，モンテカルロ法などの統計実験が有力な手法となっていることを説明する。

(2) 計数値と計量値

データは，計数値と計量値に区別される。この区別は，確率分布の型を選択することに関係がある。

計量値とは時間，長さ，温度のような連続的に測ることのできる値をいい，計数値は欠点数とか不良個数といった自然数として数えることのできる値である。母集団のモデルを考えるとき，確率分布はその確率変数が計数値に対応するか，計量値に対応するかによって**離散分布**か**連続分布**かが選択される。このことによって推測を行なうための基礎である確率分布の特性が異なり，そのために，統計量の求め方ないし，検定や推定の方法が異なってくることを説明する。

また，連続量といっても，実際には，例えば測定精度の限界があるため，データを整理するにあたって，ある桁以下は考えないことになるので，理論的には連続ではない。しかし，実際上の問題がないと認められるときは計量値として扱う。

1.3 確 率 分 布

確率分布は，統計解析で取扱う推測の対象とする母集団のモデルであり，数学的な取扱いが可能となるように，数学的に定義された全く抽象的な概念であるが，統計解析における基礎として重要であることを解説する。

よく使われる分布としては，特性値が計量値である場合の正規分布と計数値の場合の二項分布およびポアソン分布があり，母集団から無作為に抽出された標本を，このような分布に従う確率変数の実現値とみなすことによって，確率論を用いて標本変動を求めてゆくことを説明する。現実の問題を取り扱う際に，正規分布が用いられることが多い。現実の現象に対して，正規分布を近似させてもよいか否かは，基本的には実質科学の問題である。

(1) 離散分布

単位体やスコアなどの計数値に対応する分布は離散分布である。一般に，母集団の分布が連続分布の場合の方が離散分布の場合より，推測の精度や効率はよい。

適用目的によってたとえば，測定が簡単な場合，離散分布で扱うと有利な場合があることを例示する。たとえば，ある丸棒の寸法について，それが良品か不良品かを判定することを目的とする場合，測定の容易な止り，通りゲージをもちいることがある。

しかし，丸棒の直径を直接測定する場合と比較すると，同じ測定回数では検査判定の精度は悪く，また，直径の直接のデータは得られないのは当然である。

二項分布，ポアソン分布を主として説明する。

(2) 連続分布

測定精度さえあげれば，連続量としてとらえることのできる値，つまり計量値は連続分布に対応する。推測の対象とする母数の物理的特性が連続的な計量値であるときは，実際上の目的にもとづいて，母集団は連続分布が仮定されよう。分布の型は，この物理的性質から選定される。離散分布にくらべて，高精度で物理的性質が得られるのは利点であるが，その反面，測定，解析に時間，費用がかかるという欠点もある。実際的に，連続量の分布として正規分布が用いられることが多い。

寿命試験，信頼性などでは，母集団の分布として，**指数分布**，**Γ型分布**が使われる。

1.4 統計量の分布

推測の問題は，想定した母集団の確率分布について，その母数は未知であるとして，試料を用いて，この未知母数について推測することである。この推測のため，試料を母集団分布からの確率変数の実現値とみなして，母数推測に適した統計量を用いる。

ここでは，基本的な推測方式に関連した各種統計量とその分布を示す。母集団分布として正規分布を想定する場合，正規母集団という。 実際に，正規母集団にもとづく各種統計量が推測に関して用いられる。その主なものは，平均値の分布，**t分布**，**χ^2分布**，**F分布**である。

(1) 平均値の分布

平均値の分布は，平均の検定，推定の基本となるものである。実際応用面においては，工程平均の管理，実験における分散分析，ロット平均の検査など広くつかわれている。

安定した母集団（正規母集団N（μ，σ^2））での長期的な平均とみなせる統計量 $\overline{X}$ は，N（μ，σ^2/n）の確率分布に従う。また，母平均が母分散σの$1/\sqrt{n}$ （nは標本数）で推定できるという特徴があることを，チップ実験で示し，その **$\overline{X}$－分布**が母平均の検定・推定に利用されることを示す。正規分布におけるパーセンタイルは，実例について演習させるのがよい。

(2) t 分 布

平均値の分布として，母集団分布N（μ，σ^2）においてσ既知の場合は正規分布が与えられるが，σ未知の場合のt分布が導入されている。σ既知の場合と未知の場合の両者における二つの平均値の差の検定を例示して，t－表の使用について演習させること。

(3) χ^2 分 布

正規分布からχ^2 分布を導き，平方和，分散の分布を示す。分散の検定，推定を例示し，χ^2 －表を使えるようにする。

(4) F 分 布

正規分布にもとづくχ^2 分布の分散比としてF－分布が導かれることを示す。

分散分布の例を示して，分散比の分布の必要性を示唆する。F－分布が分散比の分布であることを示して，分散分析への応用に触れる。

このとき，他の分布，二項分布，正規分布，χ^2 －分布，t－分布などとの関係をF－表の上で示して，F－表が自由に使えるようにする。

(5) その他の分布

ここでは，推測の適用される実際面，X－R管理図法，二項分布のZ変換としてのΓ型分布による寿命検査などにおける統計量の性質を述べる。

さらに，低効率統計量として，順序統計量をとりあげ，ノンパラメトリックの方法を例示する。

1.5 演 習 問 題

演習問題としては，つぎに示す程度の問題のなかから適当なものをとりあげ，さらにプログラミングの実習をするとよい。

問1　5枚の硬貨を同時に投げるとき，表がでる枚数を確率変数Xにとる。
Xの分布函数，平均値，標準偏差を求めよ。

問2　甲，乙，丙3人で碁の勝抜き試合を行ない，早く2回勝った人が賞品を貰う。最初は甲，乙が手合せをするものとして，甲，乙，丙3人の賞品を貰う確率をそれぞれ計算せよ。なお3人の技量は同等とする。

問3　甲，乙両人がサイコロを投げ，甲は出た目の数に等しい金額を受け，乙は出た目の数の2乗に等しい金額を受けるという。両人の受ける金額の期待値をそれぞれ求めよ。

問4　ある大学で，入学者選抜試験の成績は平均値 600，標準偏差 100の正規分布をしていたという。志願者の何%が次の得点を得たか。

i) 800 点を超えるもの。

ii) 500 点に達しないもの。

iii) 500 ～ 800 点の範囲内にあるもの。

iv) 600 点から150点以上はなれているもの。

問5　丸棒の直径は標準偏差 0.5 mm の正規分布にしたがうとき，これを4本並列したとき，幅の標準偏差は何程か。

問6　毎日の生産量が平均値10トン，標準偏差0.6トンの正規分布にしたがうとき；1週間（操業日数6日）の生産高が65トンを超える確率を求めよ。

指導上の留意点

管理，調査，検査，実験などにおいて，対象とする試料とその母集団について例示し，統計的認識への導入とする。

母集団を母集団分布として認識させるため，工程とロットの場合を比較し，管理と検査との区別を討論させることは一つの方法である。

標本の抽出方法，測定方法いかんによって，推定する母集団の性質が変ってしまうことを例示し，規定された母集団から標本における無作為化の意義を理解させる。特に，バイアスと精度について強調しておく。データをとる目的に対して，母集団が規定されていないとか，抽出方法が目的に合ったやり方ではない場合を例示し，このような誤ったデータで判断しても，まちがった結論しか得られないことを示し，母集団という考え方が統計的方法を適用する基礎であることを強調する。

第2章　検定と推定

用　語　この章では次の用語を教える。

管理図

目　標

まず，母平均の検定，推定の問題を示し，その手法を説明する。推定論に関しては，誤差が正規分布の仮定に従う場合，完備充足統計量が成り立ち，最良不偏推定量が得られることを示して，点推定の評価と構成を説明する。ここでは，一般的な統計量や推定量についての評価には触れない方がよいが，最尤推定量は，最良不偏推定量に変換できることを述べ，フィッシャー（R．A．Fisher）とネィマン（J．Neyman）の立場について触れておく。

検定についても，正規分布の前提のもとで，最良不偏検定方式が成り立つことを示し，この場合の，有意水準，仮説の意味を解説する。推定方式は，一つの母平均の検定，推定についてσ既知片側の場合，σ未知両側の場合，二つの平均の差の場合へと，より簡単な場合から説明を加えてゆく。

検定と推定との応用として，管理図法，抜取検査について解説する。

内　容

2.1　平均の検定・推定

(1) 平均の検定

一つの母平均について，誤差がσ既知の正規分布に従う場合の検定について，仮説検定の問題を例示する。この問題について，検定の問題を形式化して，最良不偏推定方式を導く，この方式を，実例について計算演習する。ここで，有意水準，検定力関数，仮説についての理論的，実際的意味を理解させることを重点とする。つぎに平均の差の問題をとりあげ，前の例と異なる点を要点として，説明を加える。

このようなステップで，σ未知の場合までを，重点的に説明し，演習する。このような教え方は，研修生にとって，理解と適用を深める一つのうまい方法であろう。さらに，正規線型モデルの問題を提起しておく。

(2) 平均の推定

まず，一つの母平均についての最良推定を求める問題から出発する。正規分布の仮定のもとで，最良不偏推定量を導く方法をとれば，容易であろう。ここで推定の実際手法の演習までを行ない，このテーマを完結させておくこと。ついで，二つの平均の差の推定にふれ，いわゆる構造モデルの形を示しておく。

最尤推定は，正規分布の場合，さきに導いた推定量と一致することを確かめて，ここで，統計量の諸性質，評価基準について触れておく。漸近的性質は特に強調するに及ばないが，フィッシャーの情報量の考え方は強調しておいた方がよいであろう。また不偏推定に関連して，逆にＮＳＥの評価を示しておく。

2.2 分散の検定・推定

正規分布を前提として，一つの分散および二つの分散比に関する検定の問題である。ここで，同一分布に従う分散が，χ^2分布，同一分布に従う分散比がＦ分布になることを復習し，仮説を導入する。

平均の検定の場合，有意水準と検定力関数を導いたが，同様に，ここで，非心χ^2分布，非心Ｆ分布を示し，帰無仮説が成り立たない場合の検定方式の評価が満たされることを理解させる。特に注意することは，平均値の場合を対比するとき起こる片側，両側の解釈の混乱があることである。

また，ここで，母数モデルに対応して非心Ｆ分布，変量モデルに対応する検出力を区別し，その実際面，実験の構造モデルにおけるちがいを区別させること。さらに，ここで非心ｔ分布を示して，平均の検定の場合と比較させることで，理解の助けとする。

2.3 管理図法

管理図（Control Chart）とは，工程が統計的に安定状態にあるかどうかを，工程の特性値(データ）により統計的な管理限界を用いて判断するためのグラフである。

工程が統計的安定状態にあるということは，工程の母数（平均，バラツキ）が一定の状態に保たれていることである。管理図上では，25点中23～25点がプラスマイナス２σの範囲に，ゆったりとクセのない状態でプロットされる。たとえ３σ管理限界内に点がプロットされていても，プラスマイナス１σ内に点が収まっていたり，管理限界内一杯に点がプロットされるのは，安定状態にあるとはいえないことを理解させる。前者の場合には，サンプリング，測定に問題があり，後者の場合は，管理されてない要因が存在するといえる。

これらは，チップ実験，乱数などの模擬実験で，実際に，目で確かめさせるとよい。次に，管理図は，正しく描かれていたとしても，データが工程を正しく反映していなければ，管理図の役目を果さない。データが工程（平均，バラツキ）の動きを示すためには，標準化された工程を管理単位（群）とする必要がある。たとえば午前と午後では，気温の変化があって平均の変化するような工程で１日を管理単位（群）とすれば，データは，工程の動きを表わさない。また，サンプルは群を代表するようにとられなければならない。周期的に平均が変化するような工程で，継続的にサンプルをとる場合を考えると，サンプルは，周期的な変化のみを反映し，群全体を代表しえない。図を用いて例を示すとよい。

最後に，管理図と抜取検査（Sampling test）は，どちらも，統計的には仮説検定方式であるために混同されやすいので，実際目的における違いをはっきり理解させる。この二つは，次表のように要約される。

表2-1　管理図と抜取検査の比較

	管　理　図	抜　取　検　査
目　　的	工程を安定状態に保つ	ロットの品質を保証する
デ　ー　タ	工程群からのランダム・サンプル（無限母集団）	ロットからのランダム・サンプル（有限母集団）
判　　断 ① 管理限界内または合格	管理状態 工程安定状態にある	合　格 ロットの品質は，要求された基準をみたしている
② 管理限界外または，不合格	アウト・オブ・コントロール 工程は安定状態にない。変化した要因を捜し処置をとる。	不合格 ロットの品質は悪い。選別等を行なう。

(1) X－R管理図

工程の特性値が，長さ，目方，強度，純度，時間，収率などの計量値の場合に工程の平均の変化を見るためのX管理図（X Chart）と，バラツキの変化を見るためのR管理図（R Chart）を併用したものがX－R管理図（X－R Chart）である。

管理図の用法として，工程解析用の管理図と工程管理用管理図が知られている。解析用管理図とは，工程が管理図による管理を行なうほど安定状態にあるかないかを解析するために用いられる。データを収集し，層別によって，工程に変化を与える有意な原因を発見し，その原因の水準を一定に保つように標準化を行なう。その後に，合理的な群を設定する。いいかえれば，母集団を設定するための管理図である。このように安定状態になって初めて，管理図による工程の管理が行なえる。この場合が，管理用管理図である。

したがって，工程が安定状態にあっても，作業条件の変化，材料の変化など，工程を変化させる原因が生じた場合には，解析用管理図でもって，工程の解析を行なう。そして，標準化，群の設定等を経て，工程は安定状態へ導かれる。この意味で，管理図の2種類の用法は互いに関連していることを理解させる。次に，管理図の見方に関しては，特に，管理状態のパターンを，模擬実験より実際に管理図を描かせることにより目で覚えさせ，現実の管理状態とは異なるパターンを示した場合に疑問を持つよう教えるべきである。

特に点が管理限界内にプロットされても，それが1σ内にバラツいたり，2σを越えてプロットされる場合には，管理状態にないことを強調し，例を示す必要がある。

(2) 不良および欠点の管理図

ここでとりあげる管理図は，P管理図，Pn管理図，C管理図，U管理図である。計数値の分布に関して復習をする。2項分布のnとP，ポアソン分布のi（＝Pn）の大きさと，分布の形の関係を図示する。（例，P＝1％，5％，$n=\frac{1}{P}$，$\frac{5}{P}$の2項分布，$i=1,2,5$のポアソン分布）

PまたはPn管理図を用いる場合，不良項目として，電球口金のはめあい不良とか，電話機の外観不良などのように，計数値としてのデータしか得られないものがあるが，ネジの長さ不良のように，計量値としても表わせるものもある。特性値を計数値とするか，計量値とするかは，測定しやすさ，処置のとりやすさ，重要度によって決められる。これを例として，具体的な項目を列挙してみるとよい。

C管理図の項目としては，たとえば，織物の一定面積中のキズ数，エナメル線の一定長さ中のピンホールの数，プリント基板上のハンダ付不良箇所の数などがある。U管理図の項目としては，面積や長さが一定でない試料にあらわれる欠点数，例えば，織物の織ムラの数，エナメル線のピンホールの数がある。これらの欠点に関して，他の例をあげて，C管理図が適当かU管理図が適当か考えさせるとよい。

2.4 抜取検査

(1) 必要性と特徴

抜取検査の必要性およびその特徴について解説する。

まず，検査の目的およびその種類を述べることによって抜取検査の必要性を説明し，さらに，抜取検査の発展の歴史について述べ各種の抜取検査方法が開発されてきたことを説明する。統計的抜取検査においては，有限母集団であるロットの品質を保証することが目的であることから，ランダム・サンプリングが必要であることを述べ，乱数表を用いてのサンプリング方法を説明する。抜取検査法，有意差検定法および管理図法は統計解析における基礎的な手法であり，これらは，いずれも検定，推定の応用であることから，抜取検査法だけを独立して説明するよりも，研修生の統計的方法に対する体系的な理解を深めるために，他の手法と関連させて，対象としている母集団は何か，仮説は何かといった項目について比較しながら説明することが望ましい。例えば，仮説検定における母平均の検定，抜取検査法におけるロット平均の保証，管理図法における$\bar{X}$管理図などの手法を互に比較し説明する。

(2) 抜取検査の種類

抜取検査法は，検査単位である品質の表わし方が寸法，重量，燃量といった特性値であるか，不良個数とか欠点で表わされるかによって計量型と計数型に大別される。次の項目について計量抜取検査と計数抜取検査の特徴を理解させる。

① 検査方法

② 適用に際しての理論上の制約

③ ロットの良し悪しを正しく判断する能力と検査個数の関係

④ 検査記録の利用

⑤ 適用して有利な場合

例えば②について述べれば，計量抜取検査は試料を抜取する場合にランダム性が保たれればよく，抜取検査を適用する条件が一般に容易に満たされるが，計数抜取検査は試料を抜取る場合にランダム性が要求されるのは同じであるが，さらにその適用範囲が正規分布をする場合，あるいはその他の特殊な場合に限られるといったことである。

また，抜取検査の型として，規準型，調整型，選別型，連続生産型に分類され，さらに形式として，1回抜取，2回抜取，多回抜取，逐次抜取検査がある。このように一つの異なった属性によって分類され，例えば，計数規準型1回抜取検査といった名称で呼ばれる。このような数多くの抜取検査方式について個別に説明するよりも，属性による特徴について説明し，これらを表にすることによって全体的な関係を明らかにする。また，資材の受入検査，工程検査，製品検査といった抜取検査の適用例を用いて，日本工業規格およびアメリカ軍用規格について説明する。

2.5 相関と回帰

相関と回帰（Correlation and regression）は，変数間の関係を統計的にあらわしたものである。ここでは，2組のデータにもとづいて，両者の母数間の関係を推測するための統計的方法を扱う。実際には，主として正規線型モデルがあり，相関，回帰の問題は2変数正規線型モデルの場合である。

ここでは，まず，相関と回帰とのちがいを注意させ，関係の統計的把握を導入する。

つまり相関は，正規分布に従う二つの変量x，y，（たとえば，一つの製品について二つの特性値で測定した値x，y）の同時分布について，両者が独立である場合の2次元正規分布を基準として独立でない場合を区別するという意味で両者の関係を判定する。これに対して直接回帰は，原因となる指定数（説明変数X）と結果となる従属変数（目的変数Y）の関数関係を研究対象とする。

たとえば，金属の硬さは，原料加工などいろいろな原因によって左右されるが焼純温度によって，その硬さが大半きまる場合，金属の硬さのバラツキと焼純温度のバラツキとの関係を比較させるなどによって，相関と回帰とのちがいを理解させるのも一方法であろう。

(1) 散 布 図

散布図は，相関関係や回帰関係が成り立たないような場合でも，二つの変量間の関係を直観的に与える簡便な方法である。変数の値を，いろいろ与えた若干の散布図を与えて，2変数の関係を概括的に認識させることは，相関や回帰を扱う基礎となろう。

2次元正規分布の場合は，多変量解析の基礎であるから，2次元正規分布に従う場合と従わない場合の二つの場合について，演習をかねて，作図，相関係数の計算を行ない，散布図の見方を訓練しておくこと。さらに，回帰分布の準備として，直線，2次曲線について，データを与え，作図，相関係数，回帰係数を計算させて，散布図の見方を習得させることは効果的であろう。

(2) 回 帰 直 線

散布図の2次元正規分布と直線回帰モデルとの場合を示して，2変数の関係をモデル上で説明する。さらに，回帰モデルにおける説明変数と従属変数の例をあげて，実際利用上の理解を与える。直線回帰モデルについて，統計量，推定，検定の手順を理論的に解説し，後の線型回帰モデルおよび最小2

乗法で，くわしく扱うことを注意しておく。

ここでは，準備的な段階として，共分散の意味や実例について計算演習することによって，回帰モデルになれるようにすればよいとする。

2.6 演習問題

次に示す程度の問題をとり上げ，プログラミングの実習も行なうとよい。

問1. あるトランスの抵抗について，平均の直流抵抗値は，$\mu = 2.8\ \Omega$，標準偏差は$\sigma = 0.2\ \Omega$と分っている。いま製造工程の一部でこれまでと違った方法を採用して生産を行ない，9個のトランスを製造したところ，その標本平均は，3.0 Ωであった。この新しい工程から製造されたトランスの抵抗は増したといえるか。5%の危険率で検定を行なえ。

もし，抵抗が増したとしたら，95%の信頼係数でその抵抗の区間推定を行なえ。

問2. 硫安中の窒素の含有量を分析するとき，その分析値の標準偏差は約0.15%となることが経験からわかっている。信頼係数95%の信頼区間を求め，0.2%位の幅で推定したい。試料を幾つ分析したらよいか。

問3. ある知能テストの全国平均は67.5点，標準偏差13点である。ある地域で，ランダムにえらんだ169人の生徒に課したところ，平均点は69.8点であった。この地域の成績は全国平均より優秀と考えてよいか。

指導上の留意点

一つの未知母数を推測の対象とした検定と推定の考え方と方法を述べ未知母数の多い場合つまり，線型モデル，多変量への準備とする。

推定方式としては，最良方式を与える基準について，正規分布を前提とする場合をとりあげて説明すれば，理解と実用性が得られるから，一般理論にこだわらず方式適用の理解を与えるよう留意すべきであろう。検定方式についても，正規分布にもとづく統計量について，最良不偏推定が成り立つことを述べて，その根拠を理解させれば，十分であろう。なお，仮説検定における仮説の物理的意味については，従来，適用上の誤解が初心者に見られるので，具体的に理解させる必要があろう。

今後の問題として，実用的には，正規線型モデルと最小2乗法とが，この拡張として適用できる旨を，理解させるよう注意を与えておく必要がある。

第 3 章　実 験 計 画 法

用　語　この章では次の用語を教える。

最小2乗法，回帰分析，要因配置，乱塊法，交絡法，分割法，直交配列表，一部実施法

目　標

この章では，つぎのように限定したデータ解析を扱う。

① 要因効果について最良不偏推定が得られるための実験配置の若干，主として，直交計画の理論と使い方。

② 要因効果についての最良不偏推定方式の若干，主として正規線型モデルにおける最小2乗法，分散分析法の理論と使い方。

決定，設計，制御について関数情報が未知な場合，その近似関数形を推測するのは，実験データ解析の問題である。この章で扱う方法は，このような場合の多くをカバーすることができる。

内　容

3.1　最小2乗法

最小2乗推定値を求めるための計算手順を説明する。

前提からは未知母数の数には関係ないから，計算式の意味を理解させるためにまず，二つの母数すなわち，直線モデルについて説明する。

最小2乗推定値は残差平方和を最小にする推定値である。残差平方和についての偏導関数をゼロとおいた正規方程式を解くことによって最小2乗推定値が得られることを説明する。正規方程式を解くことは逆行列を計算することであり，未知母数が増加すれば計算量が急激に増大するが，コンピュータを用いて容易に求められる。計算手順を明確にするために，データの構造を示すX表，正規方程式を示すA表の作成手順を説明し，A表を解いた結果が右辺の係数としてのC表にまとめられることを示し，未知母数の推定値の係数がC表に示されていることを強調する。また，具体的な問題を用いて，正規方程式を解く計算手順を説明し，**最小2乗法**が曲線のあてはめに応用されることを示す。

さらに，誤差について，独立性，不偏性，等分散性といった三つの仮定を設けた場合およびさらに正規性の仮定を加えた場合の最小2乗推定の性質を説明する。すなわち，データと推定値の期待値，分散，共分散および残差平方和の期待値を計算し，その性質を説明する。

(1) 回　帰　分　析

回帰分析の考え方および計算手順について解説する。

回帰分析は，研究の対象とする関数型について線型を仮定する。そのパラメータの推定が目的であ

ることを強調する。ここではデータの構造として説明変数には誤差がなく，目的変数が説明変数の1次式で仮定されるような回帰モデルの最も簡単な場合を用いて説明する。すなわち，回帰直線の全変動が回帰の変動と回帰からの変動に分けられることを，計算式と図を用いて説明する。

回帰係数および切片の計算式を示し，式のもつ意味を理解させる。また検定・推定の方法を説明するとき，回帰モデルにおける誤差に対して，独立性，共分散性，不偏性，正規性といった四つの条件を仮定していることを強調する。検定，推定の考え方は要因分析においても，回帰分析においても同じであることを説明し，目的の違いを比較することによってその意味を理解させる。

回帰分析の考え方が理解されたならば，回帰モデルが曲線である場合について直交多項式として線型化と回帰分析ができることを理解させる。

(2) 正規線型モデルにおける検定・推定

1母数指数型の例として正規モデルの検定・推定を，最良不偏推定として理解した。

ここでは，線型モデルの定義を加え，これまでの方式と"よさ"の点で留意すべき事項を説明し，適用の立場を理解させる。

実際の適用については，研究対象とする現象の数学モデルと線型式との関連とを，たとえば，近似関数多項式によって説明することによって理解させる。

正規線型回帰モデルを適用することが，実際に，調査のデータ解析では多いことを注意し，直線回帰，曲線回帰の直交多項式への変換および要因配置における線型関数のモデルの推定は，共通な推定方式をもつことを比較し，理解させる。

3.2 要因配置と乱塊法

要因実験と1因子実験とを比較することによって，要因実験の性質を明らかにする。ここに，要因実験とは，要因配置による実験である。たとえば，切削工程におけるバイトの摩耗を調べるために，特性値として，クレータ摩耗量y，因子として，バイトのすくい角α_1，α_2，主軸の回転数r_1，r_2，をとりあげたとする。この実験を，1因子実験と要因実験とで実施することを想定し，1因子実験では，交互作用に関する情報が欠如していること，要因実験よりも効率の悪いことを，たとえばくり返しを入れて等しい実験数N＝12としても，1因子実験では，効果推定の分散はσ^2／8であるのに対して，要因実験では，σ^2／12となることなどを示して比較する。

次に，上の例で，実験が午前と午後にわたったとし，かつ午前と午後では，気温や機械の温度が異なるために，結果に差が出るかもしれないとする。午前と午後を同一条件にすることはできないので，これはブロックである。（他にも，ブロックの例は，多くある。農場における実験区画，炉の内部の温度の不均一に由来するもの，異なった原料，日，週等）。ブロックの影響を逃れるために，一揃いの実験を同一ブロックで行ない，ブロックの中ではランダムに行なう方法が乱塊法である。この場合，午前に8回，午後に8回する実験の計画を組み，かつ，擬似データを作って解析してみると，この関係ははっきりしよう。

(1) 1 元 配 置

1元配置（one way layout）は，要因配置のうちの1因子の配置であり，要因効果（factor-

ial effect）の線型モデルによる推測のための実験の計画である。

1元配置による実験と，古典的な1因子実験とは，みかけは似ているが区別されねばならない。1元配置では，線型モデルであり，効果（effect）の差を推測するのを目的とするが，古典的な1因子実験では，必ずしも線型モデルが使われるとは限らない。複雑な関数がモデルとして取りあげられることも多く，未知母数の推定が目的とされる。誤差に関しては，前者においては，実験全体を完全にランダマイズして，データに特定の偏りが入らないよう計画されるが，後者においては必ずしもその配慮はなされない。そのために，推測の最良規準は満たされないのが普通である。

実験実施においては，実験順序のランダマイゼイション(randomization) は当然として，とりあげた以外の要因の水準は一定に保たれねばならない。たとえば，メッキ工程において，作業者が仕上り具合をみて温度の上げ下げをしているような工程では，水準が一定に保たれているとはいえないことは強調すべき点である。

結果の判断についても，次の二つの場合には，判断が異なる。たとえば，ある工程において，従来の材料より価格の安い材料に変化させても，結果（特性値）に影響があるのかどうかを調べたい場合には，これは材料という要因において，二つの材料には差のないことを確かめるための実験である，次に，上の場合に，価格のより高い上等な材料を用いる場合には，逆に，特性値が望ましい値に変化することを確かめるための実験である。

最後に，平方和，効果の推定等の計算法は，すべての要因配置，直交配列表による実験の基礎であるので，手順を理解させ，テキストなしで計算できるようにする。F表が正しく引け，F検定の統計的意味も理解させる。

(2) 2 元 配 置

2元配置（two －way layout）は，2因子の要因配置である。データの構造モデル（model）は，二つの主効果（maineffect）と，交互作用（interaction）と誤差（error）の線型結合として表わされる。2元配置では，データが加法的でない場合（すなわち，交互作用の存在する場合）にも推測が可能である。このことは，要因配置の基礎でもある。

交互作用は，わかりにくいので，グラフ等によって図式的に説明するとよい。 交互作用の存在する場合としない場合とを，最適組み合せを求めるという形で比較すると理解を助ける。

2元配置は，似た形式の実験がいろいろあるので，それらを区別するためのよい例が作られる。たとえば，一つの因子に相当する部分がブロックである場合がある。この場合には，1元配置の乱塊法である。また，一つの処理組み合せにつきr個のデータが得られている場合に，もし，全実験$r \cdot N$がランダムに行なわれたならば，くり返しのある2元配置であり，要因配置の実験である。次に，この実験がN通りの処理組み合せの実験をr回反復したのならば，2元配置（完備ブロック）の乱塊法である。さらに，N通りの処理組み合せの各々で，r個ずつのサンプルがとられた場合には，分割実験である。実験順序，構造モデルを示して，十分に理解させる必要がある。

解析法に関しては，交互作用に関する平方和，効果の計算が重要である。高波交互作用の計算の基礎ともなるので，テキストなしで計算できるようにさせるとよい。自由度の計算に関しても同様である。

(3) 要 因 配 置

要因配置（factorial design）の特徴は，効率のよいこと，すなわち，すべての因子の組み合せによる情報が得られることにある。

たとえば，5因子2水準の要因配置についていえば，実験回数$N = 2^5 = 32$，得られる効果は，

平均＝0因子交互作用効果	${}_5C_0$	＝	1
主効果＝1因子交互作用効果	${}_5C_1$	＝	5
2因子交互作用効果	${}_5C_2$	＝	10
3因子交互作用効果	${}_5C_3$	＝	10
4因子交互作用効果	${}_5C_4$	＝	5
5因子交互作用効果	${}_5C_5$	＝	1
	計		32

となる。これを例として，3水準の場合もやらせるとよい。

次に要因配置は，交絡することなく，各効果が独立に推定，検定ができることを2元配置を例にして示す。

例　2元配置

$$y_{ijk} = \mu + \alpha_i + \beta_i + \alpha\beta_{ij} + \xi_{ijk}$$

$$i = 1, 2, 3 \qquad j = 1, 2, 3, 4 \qquad k = 1, 2$$

これは，X表，A表，C表，分散分析表を作ることにより，容易に理解されよう。

なお，対比の直交性を強調しておくことは，後の準備として有用である。

要因配置は，すべての組み合せを行なうのであるから，因子が増すにつれて急激に実験回数は増す。このことは，現実的には実験ができなくなることを意味する。

そのために，交絡法，一部実施法（直交配列表による実験）が生まれてくることを示しておくとよい。

(4) 乱 塊 法

たとえば，ある切削工程においで，前加工されたものを加工するのに，バイトの送り速度を3水準V_1，V_2 およびV_3 にとって実験したい。実験は，試料を9個とって行なうとする。この場合，試料を時間的に三つに層別したり，三つの異なるロットから3個ずつとられたものであったり，さらに試料は均一な母集団からとられたものであるとし，加工自体を3日間にわけて，各々3水準ずつ行なったりすれば，これらは，すべて，乱塊法（random block design）である。

この例を，1元配置の形式に書き表わして比較させるとよい。さらに，この例について，擬似データを作成し，乱塊法を完全ランダマイズの1元配置として，誤った解析を行なった場合と，正しく乱塊法として解析された場合の比較をして，ブロックを見逃すことによる検出力や推定の精度に対する影響を示すとよい。

ブロックとは，上の例のように，ランダマイズできないもの（日，時間による層別），あるいは，

無視することによって，実験誤差を大きくするもの（異なるロットを混合するもの）である。

一般的に，上の例のほかに農場実験における仕切，温室内における場所，何台もの機械，炉，作業員の交代などがある。これを例として，さらにとり上げさせ検討するとよい。

3.3 交絡配置

要因配置では，因子の数が増すと実験回数は急激に増す。5因子2水準の場合でも$N=2^5=32$回である。

現実には，バイトの摩耗を調べるような簡単な実験でも，数日は必要とする。日によって，作業者が代ったり，機械の調子も異なるので，この実験を完全ランダム実験として実施すると，誤差を大きくする。誤差を小さくするために，ブロックを高次交互作用効果と交絡させる方法が**交絡法**（confounding）である。（完全交絡）

上の例のような機械加工においては，3因子以上の交互作用は無視してもよい。線型モデルは，データを大きな効果，すなわち，主効果，2因子交互作用，3因子交互作用の順で説明するものであるから，高次交互作用は一般に小さいと考えてよい。

しかし，交互作用が無視されえない場合には，要因配置を反復させ，反復ごとに交絡させる相手をかえることにより，交互作用の情報を残すことができる(部分交絡法)。さらに，ブロックを組み分けすることにより2重交絡となる。

ブロックを主効果あるいは，低次の分割に交絡させるのが分割法である。

たとえば，因子としてメッキ液槽の温度Tを2水準とする。実験は2日間を要し，温度変更は1日1回より出来ないとする。この場合，ブロックRと因子Tは，1次単位とするのが普通である。

(1) 交 絡 法

交絡法は，ブロックと交絡させた交互作用に関する情報を犠牲にして，主効果の精度を上げる方法である。

完全交絡が基本であるので，2^3 型要因配置と比較する。因子をA，B，Cとすると，3因子交互作用ABCと交絡したブロック（2日間）は，大きさが4であり，ブロック内で，残りの対比A，B，C，AB，BC，AC，が得られることを示す。

ブロックがさらに組み分けされると（この場合午前，午後など）ブロックの大きさは2となる。

ブロック内で得られる対比は，A，B，C，ABCになる（2重交絡）。

完全交絡，2重交絡は，いずれも，交絡された交互作用は得られない。交互作用を無視できない場合には，完全交絡自体を反復し，反復毎に交絡する交互作用をかえればよい。

この場合に，3反復したとすれば，交絡された交互作用は⅔の精度を持つ。

(2) 分 割 法

分割法（split plot design）は，重要度の小さい主効果とブロックを交絡させる，あるいは，分割が多数にわたる場合には，低次の分割についてブロックを入れる方法である。

ブロックと交絡した主効果の精度は小さくなるが，残りの主効果交互作用の精度はよくなる。

分割法は，主として，現実の要求から用いられることが多いことを示す。

たとえば，炉の温度とか，メッキ槽の組成等は，変更困難であるために，ブロックと交絡される。

また，ある前加工された製品を，さらに，いろいろな条件で加工するような場合には，初めから無作為化は不可能である。このような現実の場を想定して実験がどのような分割になるかを題材とすることも解析練習とは別に重要である。

解析的には，誤差が，1次，2次，……などと，いくつにも分解されるので，その意味を理解させることが重要である。そのためには，データの構造モデルを各次数の構造を説明する。

また不偏分散の期待値E(V)の構造から，検定，推定の方法を説明するなどの工夫が必要である。

3.4 直交配列表

直交配列表（orthogonal array）による実験は，基本的には一部実施法であり，列記号や線点図を利用することにより，交絡法，分割法の実験が容易にできる。すなわち，現実に必要とされる実験のほとんどすべてを，直交表を用いることにより実行できる。

直交表の利用は，実験に有用ではあるが，十分に使いこなすには，強さtの直交行列Oi（N，n，p，t）を理解しておくことが大切である。P＝2の場合には，2^2 型要因配置と 2^2 型直交配列表との対比を理解することは，直接比較することにより容易であろう。さらに，水準を（－1，1）とし，かつ，単位列ベクトルをつけ加えることにより，回帰モデルとの対応も容易である。

P＝3の場合には，3^2 型要因配置と，3^2 型直交配列を比較し，交互作用が2列に現われることを理解させる。これらの例は，多水準合成の規則の理解も助ける。

直交配列表の有用性を理解する別の方法としては，すでに学んだ要因配置，交絡配置を，直交配列表にわりつけてみることである。特に，2^3 型の交絡配置の例は，そのまま直交表にあてはまるので，とり扱い易い。分割実験については，直交表を用いると，普通の方法では計画が困難なものも，きわめて容易にわりつけられる。

(1) 一部実施法

要因配置においては，因子の数が増すにつれて，実験数Nも増す。交絡法においても，必要としない効果とブロックを交絡させているが，**一部実施法**は，高次交互作用効果と主効果を交絡させて，実験数Nを節約する方法である。たとえば，2^4 型要因配置において，因子Dを3因子交互作用とABCと交絡させることによって，実験数は16回から8回に減る。

多くの実験では，3因子交互作用は必要としないので，これは有用な方法である。2^5，2^6 型についても，½，¼，⅛…実施の計画をたてさせてみるとよい。

一部実施における交絡関係を示す目安として，定義対比（defining contrast）が用いられる。

2^n 型要因配列において，定義対比が1≡ABCDということは，主効果Aと3因子交互作用BCDが交絡していることを示す。このことはまた，2因子交互作用ABとCDが交絡していることも示す。ここで，AとBCD，ABとCDは，区別のできないという意味で，同じ効果を示しているので，それらは互に別名（aliases）と呼ばれる。上の例でもって，定義対比，別名を同時に考えるとよい。

(2) 直交配列表

直交行列Oi（N，n，p，t）の定義を理解することが，第一の目的である。強さtとは，n列

のうち，どのt列をとっても，同じ行列は（Nλ，t）行列の中には二つとはないので，各列は，互に直交対比をなしている。さらに，（Nλ，n）行列のうち，任意のt′（<t）列からは，他の列を作りだすことはできない。P＝2，t＝3の場合を考えると，t′＝1すなわち，主効果は，他の主効果とは交絡しない。またt′＝2すなわち，2因子交互作用もまた他の主効果とは交絡しないことを示す。

応用としては，交絡法において，2因子交互作用と主効果の交絡を防ぐためには，t＝3の行列を利用する必要があることを示している。

実験に用いられる直交配列表は，t＝2である。故に，主効果と2因子交互作用は交絡し得ることを例として示す。

強さt=2の直交配列表の作成には，相合式による方法(i)列＋(j)列≡(k)列（mod P）が定義通りで理解しやすいが，実際には，機械的な方法を用いるのがよい。列記号と，相合式の対応を示しておく。

2^n型，3^n型要因配置は，2^n型，3^n型直交配列と対応するので，n＝2，3に関しては，対応させてみると，要因実験と直交行列の関係が理解されよう。

(3) 2水準系のわりつけ

わりつけは，列記号による方法を用いる。列記号によれば，相合式は，掛け算の規則$P^2 \equiv 1$で表わされる。2^2型直交配列と比較して，主効果A，B交互作用ABが，列記号でP，Q，PQに対応することを示す。さらに，AをPQ，BをPに対応させると，ABはQに対応することも同時に注意する。一般の2^n型直交配列では，3因子の交互作用も，それぞれの列記号の掛算となることを，2^3型などを例にして示す。

2^3型における一部実施は簡単にできることを列記号で示す。AをP列，BをQ列，CをS列にわりつける。Dを3因子交互作用と交絡させるならば，定義対比はABCDである。定義対比は，列記号によれば，恒等的に1であるので，ABC，PQS列にDを割りつければよい。同様にして，もし，DをABと交絡させるならば，定義対比はABC，Dを割りつけるのはPQ列となる。交絡法の場合には，上の例で，因子Dのかわりに，ブロックRをわりつければよい。

2^n型（強さ2）の直交配列から，強さ2の任意の3列をとりだして，4水準を合成できる。

同様にして，強さ3の4列（強さ2の直交配列では必然的に7列）を用いて，8水準の合成もできる。必要な列数については，自由度とも関係させた方がわかりやすい。

分割法に関しては，規則に従えばよいが，高次どうしの交互作用が低次に入れることは，わかりにくいので，例を上げて説明するとよい。たとえば，Aが1次単位，B，Cが2次，BC，ABが必要，ブロックRが存在する場合には，交互作用BCは4次単位にあらわれる。

直交配列によれば，形式的にわりつけができるので，交絡，分割などの意味を忘れがちであるのでできれば前の結果と比較するとよい。

(4) 2水準系の例

2水準系の直交配列表の理解のために，わりつけの演習，応用例をあげて説明する。

① 2^n型要因配置実験を直交配列表にわりつける。たとえば，n＝3，4の計画で，すべての交互

作用を求めるわりつけを行なう。この実験を，完全ランダマイズしてくり返した場合には，要因配置実験，完備ブロックとして反復を実施した場合には，乱塊法となる。

② 前の例において，必要としない3因子交互作用列に，新しい因子をわりつけると，一部実施になる。定義対比と別名を演習する。さらに，もう一つの因子を必要としない交互作用列にわりつけると，2重定義対比となることを示す。別名も2組存在する。

③ 交絡法に関しては，完全交絡，2重交絡は，一部実施と比較することにより理解されやすい。

部分交絡法は，直交配列表でなければ，計画が困難であることを強調する。たとえば，4因子A，B，C，Dと4組の2因子交互作用を，2^3 型直交配列表を用いて反復2回でわりつけてみるとよい。

④ 分割実験の例としては，次のような例を示す。要因とブロックが交絡する例として，要因A，B，C，D。Aは，水準変更が困難なので，変更は1度にとどめたい。交互作用BC，CDが必要，日程は，2日間の場合，4日間の場合で交絡関係は変る。

次に，1次単位にブロックの入る例として，要因A，B，C，D。実験は2日間を要し，Aは，1日に1度より水準変更はできない。交互作用BC，CDが必要。Aの水準変更を1度よりしなければ，上の例と同じである。

検定方法，推定の精度に関して比較する。

(5) 3水準系のわりつけ

3水準直交配列によるわりつけは，基本的には2水準系と同じである

ただし，列記号は，相合式に対応して，$P^3 \equiv 4$ なる関係を持つ。2因子交互作用が，2列，$P\bar{Q}$ と PQ^2 （それぞれI系，J系）となるので，この点に注意する。例えば，3因子A，B，Cを 3^2 型直交配列にわりつけると，AをP列，BをQ列として，定義対比をABCとするならば，CはPQ列定義対比を AB^2C とすれば，CはPQ^2 列である。

また，3因子交互作用は4列にあらわれる。Finney，Yates，交絡法による記号が用いられるが，どれでも扱えるようにするとよい

強さ2の4列を用いて，9水準系を作る手順は，2水準の場合と同様である。自由度と対応させて理解させるとよい。

交絡法，分割法に関しても2水準の場合と同様である。。

(6) 3水準系の例

3水準系の直交配列表の理解のために，わりつけの演習を応用例をあげて説明する。

2水準系にならって，3^n 型要因配置の直交配列表による実験，一部実施法，完全交絡，部分交絡，二重交絡，分割実験などの例を演習する。2水準系とは違って，交互作用列が2因子交互作用の場合で，2列，3因子交互作用は4列に表われるので，注意する。自由度と対応させて説明すると理解させやすい。また，多水準のブロックの構成のためには，4列必要なことも例をあげて説明するとわかりやすい。例を以下に示す。

1° 3因子A，B，C（3水準），すべての交互作用が必要。

2° 3因子A，B，C（3水準），2因子交互作用必要，実験には，3日間を要する。

・3° 3因子A，B，C，D，F（3水準），主効果のみ必要，9日間を必要とする。

・4° 3因子A，B，C（3水準），すべての交互作用が必要。実験には9日間を要する。

・5° 8因子A，B，C，D，F，G，H，I（各3水準）

上の8因子の組み合せごとに，試験片を作り，その試験片の前端と後端をおのおのの3回ずつ測定する。必要とする効果は主効果の他にGH，GIである。

3.5 演習問題

演習問題として，次に示す程度の問題をとり上げ，さらにプログラミングの実習をするとよい。

問1. ある反応工程の収率を上昇させる目的で，触媒の添加量を 1.0％，1.5％，2.0％，2.5％と変えて，各5回ずつ実験を行なった。その結果は下表のとおりであった。ただし，5×4＝20回の実験は，ランダムな順序で行なった。

（単位％）

A_1（1.0％）	A_2（1.5％）	A_3（2.0％）	A_4（2.5％）
83.3	87.3	89.5	91.0
84.4	85.8	88.4	93.3
83.5	86.2	89.9	92.0
84.9	85.6	88.7	92.6
84.3	86.7	89.3	92.9

1）分散分析を行なえ。

2）各水準の母平均を推定せよ。

3）直交分解せよ。

4）回帰式を推定せよ。

問2. 糊の配合の方法Aを4水準（A_1，A_2，A_3，A_4），助剤の量Bを5水準（B_1＝1.0％，B_2＝1.3％，B_3＝1.6％，B_4＝1.9％，B_5＝2.2％）にとって，二元配置で糊の皮膜を作ってその伸度（％）を測定した結果，次の値が得られた。

	B_1	B_2	B_3	B_4	B_5
A_1	32	24	52	45	57
A_2	20	36	51	55	58
A_3	17	30	41	47	48
A_4	23	41	42	37	59

1）分散分析を行なえ。

2）Bが有意ならばBの各水準の母平均の確率95％の信頼限界を求めよ。

3） Bについて直交分解せよ。

4） A_2B_3 の組合せの母平均の確率95％の信頼限界を求めよ。

問3. ある反応工程の収率を上げる目的で，反応時間（A），反応温度（B），成分（C）の3つの因子を取り上げて要因実験を行なった。3因子の水準数をそれぞれ2水準ずつとったので，因子の水準の組合せの数は8とおりあるが，1日に8回の実験は行なえなかったので，3因子交互作用$A\times B\times C$を日と交絡させた交絡法の割付けで，2回反復の実験を行なった。実験データは下表のとおりである。これを解析せよ。

〔因子とその水準〕

反応時間	A：	$A_1=4$，	$A_2=5$（時間）
反応温度	B：	$B_1=250$，	$B_2=300$（℃）
成分Cの量	C：	$C_1=2.0$，	$C_2=2.4$（g）

原　表（単位％）

1日 R_1		2日 R_2		3日 R_3		4日 R_4	
a	83.3	ab	86.7	ab	88.2	a	84.2
b	86.8	ac	84.0	ac	84.8	b	84.7
c	84.1	bc	86.5	bc	85.6	c	83.7
abc	86.3	(1)	83.1	(1)	82.5	abc	86.4

$A\times B\times C$ を交絡　　$A\times B\times C$ を交絡

指導上の留意点

実験計画法の問題は，要因の水準に対応する観測結果について，推測を行なうのであるが，そのため誤差に関して，正規モデルを前提し，要因の結果を母数に対応させることによって，母数の推測方式を可能ならしめる。ただし，要因および水準数が多くなると特別な条件を設けて推測を進めることになる。つまり

① 要因の結果の母数について線型化する。

② 正規線型モデルによって最良不偏な推測が得られる。

①を実現するものとして直交配列表が利用され，②を実行するため最小2乗法が用いられる。

要因が連続的な説明変数で，結果に対する近似関数が多項式の場合，直交化によって，最小2乗法が求解される。これは，正規線型回帰モデルである。多要因のすべての組合せは線型モデルを与えるが，これは，要因配置法をもたらす。さらに，必要な要因の効果が直交するように配置する工夫がなされた場合として直交配列表によるわりつけがある。直交配列表と用いて，古典的な配置を計画することもできる。交絡配置と直交配列表との比較をしてみることも，配置計画の理解に役立つであろう。

直交行列の作成，最小2乗法のための行列計算などは，コンピュータのよい材料であろう。

第 4 章　統計解析

用　語　この章では次の用語を教える。

多変量解析，重回帰分析，最適計画，複合計画，相関係数，主成分分析

目　標

本章では，統計解析の問題を多変量の問題および時間的過程に関する問題へ拡張する。実際に用いる解析上の手法は，正規性線型仮定を前提とするので，本章までの比較的簡単なモデルの場合が基本になる。

多変量解析の手法に習熟し，その応用に対する理解を深めること，さらに，多変量解析および時系列に関連して，統計的最適計画について述べ，逐次接近の場合と全体についての解析の場合についての解析を理解させる。

内　容

4.1　多変量解析

多変量解析の手法として，以下について説明する。

(1) 重回帰分析

重回帰分析（multiple regression）とは，P個の説明変数により，目的変数 y に関する情報を得る方法である。

y の関数形としては，$y = f(x)$ のテイラー展開の1次の項まで用いた線形重回帰モデルが用いられる。説明変数は一般に直交しない（相関を持つ）ので，分散分析は，回帰成分全体を残差と比較するという形をとる。偏回帰係数の推定値は，最小二重推定値であるので，最良不偏推定値となっている。

重回帰分析においては，説明変数が直交しないために，個々の偏回帰係数は，その特性の影響を100％説明はしない。複数個の偏回帰係数がまとまって，はじめて，目的変数を説明する。

例として，P＝2とし，項のモデルとしては，$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12}$，$X_1 X_2$ を用いる。（X_1，X_2）の値は，$X_1 = X_2$ の直線上か，それに近い点を数点とって，擬似データを作成し，解析することにより，上に述べたことは理解されよう。。

同様の理由で，P＝3のデータから，P＝2だけとりだして推定した偏回帰係数と，P＝3のまま推定した偏回帰係数とは別のものである。重回帰分析を解釈するのには重要なことである。

(2) 判別関数

（2群）判別の問題とは，たとえば，人間の採用に際して，その人間を事務職につけるか，作業者にするのか決めることである。今，過去のデータより，事務についた人と，作業者になった人の，入社試験時のデータがあるとする。そのデータより，判別関数とよばれる線形式の係数を求め，新規採用者の試験データを判別関数に代入して，その値により，事務につけるか，作業者とするかを決める問題である。

判別関数（discriminant function）$Z=\sum A_i X_i$ の係数 A_i は群間の変動の，群内の変動に対する比を最大にするように定められる。

P＝2の場合のマハラノビスの凡距離を図示するとこの関係は理解しやすい。

計算は，手計算では無理なので，コンピュータにやらせ，その結果について議論した方がよい。

(3) 主成分分析

たとえば，既製服を作る問題を考える。人の体の特性値としては，体重，身長，座高，胸囲，10数種の測定項目がある。これをもし，おのおのの特性値のすべての組み合せを用意するとなると，それは不可能である。そればかりでなく，その必要もない。体重80kgで，身長160cmの人は，まずいないからである。上にあげた人の体の特性値は，互いに相関関係があるのが普通であり，もっと少ない成分で人の体型を表わすことが可能であると予想される。細長の人であるとか，猫背の人であるとかである。この操作を統計的に行ない，数個の成分で説明する方法が，主成分分析（principal component analysis）である。

主成分分析は，このようにして得られた成分が，互いに無相関であり，かつ，要約された成分だけで，大部分を説明する方法であるといえる。

主成分分析を直観的に示すには，P＝2の場合を図示するのがよい。さらに回帰直線とも比較すると，その差異を理解する助けになろう。

実際の計算は，コンピュータによる。

4.2 最適計画

一般に，未知の応答面 $y=f(x,\beta)$ の最適点 y_{max} を見出す実験的方法を**最適計画**（optimal design）という。

ここでは，1次計画として，1因子の方法（UV法，univariate method）と最大傾斜法（SA法，steepest ascent method）をとりあげる。2次計画としては，複合計画を扱う。

1次計画とは，モデルとして1次回帰式 $y=\beta_0+\sum_i \beta_1 X_1$ を用い，逐次的に最適点を探索する方法である。UV法では，応答面に交互作用があると，途中で停留してしまう。SA法では，交互作用が存在しても，最適点まで到達できる。しかし，実験回数はUV法より増す。これらは，2度数の場合を例にして，図で示すことによって理解させるとよい。

ここで問題となることは，スケーリングの問題である。スケーリングの問題とは，応答曲面に対する実験単位の大きさによる回帰係数の推測に与える偏りの問題である。われわれは応答曲面を知ることは

できないのであるから，合理的な基準をどのようにして定めるかは問題である。

2次計画は，モデルとして2次回帰モデル $y=\beta_0 \sum_{ij} \beta_{ij} X_i X_j$ を用いて，応答曲面を2次曲面で近似して，最適点を捜す方法である。計画は，UV計画とSA計画を組み合せた複合計画（composit design）が用いられる。この計画は，3^k 型要因配置計画と較べると，2次係数の推定精度の点ですぐれている。演習として，k＝3，4，5程度を課してみるとよい。

複合計画の種類としては，直交型，等分散型，回転型がある。また，原点で，くり返し数を多くして，実験誤差の大きさを推測できる。原点でのくり返しは，回帰係数の推定精度に影響を与えないので，lack of fit の検定には，有効な方法である。これらは，比較すると，理解を深めるのに役に立つ。

2次計画における問題は，応答面が，実際2次曲面で近似できるかどうか（lack of fit）の問題である。応答面が2次でない場合には，最適点を誤る。2変数の場合の例を図示して，この問題を認識させる必要がある。

(1) UV法・SA法

UV法・SA法はいずれも極値探索を，逐次的に行なうための1次計画である。UV法は，1因子実験による極値探索法である。UV法では，交互作用が存在した場合には，交互作用は1次回帰係数に交絡するために，最適点を見出せない。逆に，交互作用が存在しなければ，（スケーリングの問題を除いて）必ず最適点に到達する。

SA法は，2^k 型要因配置またはその一部実施計画によって，1次回帰係数を推定して，推定された平面の最大傾斜の方向に登って行く方法である。交互作用は分離されるために，最適点の近くまで到達することができる。

推定精度の面では，SA法はUV法より優れている。誤差のある場合と，誤差を0とした場合を比較させるとよい。

実験単位の大きさを決める問題は，スケーリングの問題として知られている。1次回帰係数の推定値には，実験単位の大きさによって，高次の奇数項が交絡してくる。また，推定の精度も変化する。さらに，実際単位においては，一つの因子（変数）が圧力（kg／cm²）であり，他の一つの因子が，温度（℃）である場合に，交絡する項の大きさや，推定の精度を比較できない。圧力の単位をkg／cm²から，ポンド／cm²に変え，温度の単位をCからFに変化させた場合を考えると，回帰係数そのものが相対的なものであることを充分納得させることができる。この点に注意を喚起しておく必要がある。

(2) 複合計画

複合計画（composit design）は，応答曲面を2次の回帰モデルで近似して，最適点を求める方法である。計画は，2^k 型計画（またはその一部実施）と，原点対象な1因子実験の組み合せで行なわれる。原点での実験のくり返しは，回帰係数の推定精度に影響を与えないことに注意し，実際に確かめさせるとよい。この性質は，lack of fit の検定に利用される。3^k 型要因配置と比較すると，複合計画は，2次係数の推定の点で優れている。その点が計算の面倒な複合計画の使われる一つの理由になっている。効率の計算を演習させると理解されよう。

複合計画には，目的によって，直交複合計画（orthogonal composit design），等分

散複合計画（equi－variate composit design）が使われる。直交複合計画は，解析が容易であるが，効率はよいとはいえない。等分散複合計画は，2次係数の推定精度が等しくとられている。推定された2次式を用いて極値を推測するのにはよい計画である。回転型複合計画は，実験点のどこでも（原点を除いて），yの推定精度が等しくなる。応答面の主軸の方向が未知の場合によい計画といえよう。これらの違いを説明し演習させるのがよい。

応答面が2次曲面でない場合には，スケーリングによって，lack of fit が変化する。すると，lack of fit の検定は，応答面に対する知識がない場合には，意味のないことになる。すなわち，複合計画そのものの適用の意味がなくなることを示す。2度数で，3次曲面における複合計画の例を作って吟味させるとよい。

4.3 その他の統計解析

簡便法の目的は，母平均の推定，検定や分散の違いの検定，ランダムネスの検定などを，簡単に手早く計算することにある。また，厳密な方法を用いて結論をだす必要のある場合でも，だいたい目安をつけるためにも簡便法は用いられる。

簡便法としては，2項確率紙による平均値，分散，不良率，符号検定，大波・小波検定と代用R検定，時系列におけるランダムネスの検定のための連の数，連の長さ，上り下りの数による検定などを扱う。いずれも，厳密な方法と比較して，その簡便法を確かめる。また，簡便法の使える範囲を吟味することが望ましい。

時系列に関しては，管理図などにおける時間を要素とした母数からのデータのランダムの検定を扱うために，オートレグレシブ・プロセス（autoregressive process）モデルを扱い，系列共分散（serial covariance），系列相関係数（serial correlations），コレログラム（correlogram）などを説明すると理解しやすい。

(1) 簡 便 法

簡便法が自由に使えるように，簡便法の使用例を多く演習させる。

① 2項確率紙

2項確率紙による母平均の推定・検定，不良率に関する推定・検定，ランダムネスの検定，符号検定，大波・小波検定などを，厳密な方法と比較しながら演習させる。同時に，2項確率紙が近似的に正規分布になっていること，比率の場合の正規近似が np 25 以上でなければ成立しないことを理解させる。

② 代用R検定

母集団のバラツキを示す統計量（データ）として，範囲Rが用いられることを，1.4の(5)を復習しながら理解させる。

検定は，F検定に対応している。目的によって，すなわち，2組の母集団の間に差があるかないかを見るかによって，検定の性能が異なることを注意する。

③ 系列のランダムネスの検定

管理図上にプロットされた点のように，時系列を考えられるデータがランダムかどうかを検定す

るのに，連が用いられる。連による検定には，連の長さによる検定，連の数による検定，上り下りの連の長さによる検定，上り下りの連の数による検定などがある。厳密な検定に比べると，点の数を数えればよいので，非常に簡単である。系列相関係数を計算してみると理解できる。

(2) 確 率 過 程

管理図上にプロットされた点のように，母集団の時間的変化を反映するデータ（統計量）は，確率過程（stochastic process）をなすと考えられる。実現値の系列は，時系列（time series）と呼ばれる。

時系列が相関を持つということは，時刻 t における実現値 X（t）に対して，ある時間間隔 τ だけ隔たったデータ X（$t=\tau$）が影響していることである。X（t）に対するその他の影響は，正規分布に従うので，この時刻系列はマルコフ過程をなす。相関が0な場合は，X（t）は，誤差のみの影響を持つ。すなわち，ランダムである。

時系列の**相関係数**は，τ の値によって，変化する。τ の値を横軸にとって表わしたものが，コレログラムである。

相関係数が0ならば，その時系列は，ランダムであるといえる。時系列でない相関と比較すると，時系列における相関のゼロ仮説の検定は厳密にはむずかしいことがわかる。簡便法の利益がよくわかる。

指導上の留意点

多変量解析（multi variate analysis）とは，独立または互いに相関のある複数個のデータを統計的に評価する方法である。

われわれの扱う現実の調査・研究の対象は，多様な特性を持っており，多変量モデルとしてデータを整理した方が対象を扱いやすい場合がある。たとえば，腕時計の性能をあらわす特性をとりあげると，石の数，ゼンマイのトルクの強さ，歯車の伝達トルクの値，歩度，日差，姿勢差，実用試験における歩み遅れなど，多くの項目があり，一体腕時計の性能を何で表わしたらよいのか判断に迷う程である。試験の成績のように，各データの合計または平均も一つの多変量解析であるが，現実にはいろいろな手法が知られ，用いられている。

多変量解析の主な手法としては，重回帰分析，重相関分析，判別関数，主成分分析，因子分析と重相関分析は互いに関係の深い手法であって，前者は，独立変数と従属変数からなり，独立変数に対する従属変数の線形回帰式を推測するための手法であるのに対して，後者は，変数間の相関を求めることにより，変数間の関係の程度を見ることを目的とする手法である。これらは，多変量解析のうち最もよく知られた方法であり，実験とも関係が深い。（相関と回帰，最小2乗法，回帰分析の節を参照する。）次に，判別関数は，過去のデータの解析結果をもとにして，提出されたある試料の多変量データを用いて，その試料がどの群に属するかを判別する手法である。臨床診断，適性判断などに応用されている。最後に，主成分分析，因子分析は，現象を2～3個の要素で説明したり，現象の構造を分析するための方法である。前者には，人間の体型の分類のための応用，後者には心理学における応用がある。

多変量解析のいろいろな手法の共通の問題としては，一般に提出されるデータが，実験の場合のように，計画され管理されたデータではないことである。そのために，多変量解析によって得られる結論は，

得られたデータにのみ通用するにすぎない。また，データ内の相関は通常0でないので，たとえば重回帰分析の場合には，偏回帰係数間に加法性が成立しない。判別関数の場合にも，変数が1個増減した場合でも，そのままでは，判別の役にたたない。

最後に，計算は，コンピュータを抜きにしては考えられない。有用なプログラムの利用が必要である。

なお，この章は，時間に余裕のある場合や必要がある場合以外は，簡単な概説にとどめておいたほうがよい。

参考文献

(1) 藤代　侑宏「ステップ式による統計的方法」，日科技連

(2) 依田　浩「技術者の統計学」，宝文館

(3) 依田　浩「初等統計学問題演習」，宝文館

(4) 森口　繁一「初等数理統計学」，培風館

(1), (2), (3), (4) は統計解析全体にわたって利用できる。

(5) 石川　馨　共著「初等実験計画法テキスト」　日科技連

実験計画法を教えるときに最適

シミュレーション

目　　次

第1章　コンピュータ・シミュレーション …… 60

1.1　コンピュータ・シミュレーションの使命 …… 60

1.2　シミュレーションの実施効果 …… 62

1.3　シミュレーションの利用限界 …… 63

1.4　シミュレーション実施手順 …… 64

1.5　シミュレーションの分類 …… 67

第2章　シミュレーション・モデルの作成 …… 70

2.1　モデルの構築 …… 70

2.2　統計収集と出力 …… 74

2.3　乱数発生ルーチン …… 75

第3章　モデルの運用 …… 78

3.1　初期設定とパラメータの変更 …… 78

3.2　モデルの検証 …… 79

3.3　シミュレーションの結果の分析 …… 80

3.4　シミュレーションの最適化 …… 81

第4章　シミュレーション言語 …… 82

4.1　GPSS (General Purpose System Simulator) …… 82

4.2　SIMSCRIPT(Simulation Scriptor) …… 84

4.3　DYNAMO (Dynamic Models) …… 86

4.4　CSMP (Continuous System Modelling Program) …… 88

4.5　言語の選択 …… 90

第5章　実　習 …… 92

科目「シミュレーション」

教育の目標

この科目では，企業における予測，計画，評価などの諸問題を解決するためのシミュレーション技法を理解させ，シミュレーション・モデルの作成法から，シミュレーションの結果を解析するまでの能力を身につけさせる。また，現存するシミュレーション言語の中から，研修生に適したものを選んで実習を行ない，業務に役立つプログラムの作成能力を身につけさせるとともに，ケース・スタディをとおしてシミュレーションによる問題解決の技術を習得させる。

時間配分

章	履習時間（時間）		
	講義	演習	実習
1. コンピュータ・シミュレーション	6		
2. シミュレーション・モデルの作成	15		
3. モデルの運用	15		
4. シミュレーション言語	24		
5. 実習			30
合計	60		30

第1章　コンピュータ・シミュレーション

用　語　この章ではつぎの用語を教える。

コンピュータ・シミュレーション，ハイブリッド・シミュレーション，離散形，連続形，混合形

目　標

まずコンピュータ・シミュレーションとは何か，他の手法との関連はどうかということを述べるとともに，コンピュータ・シミュレーションがどういう状況で利用される手法であり，その使命・適用限界・実施効果が何であるかを具体例や歴史的背景をもとに明確にする。

つぎに，シミュレーション研究の体系的アプローチ，すなわち，問題の定式化，情報収集，モデル化，解析，解析結果の評価の各段階とORとの関連を述べ，シミュレーションの位置づけを明らかにするとともに，各実施手順の説明を通じシミュレーションの全体像を理解させる。最後に，第2章以降で扱うディジタル・シミュレーションの位置づけを明確にするためにシミュレーションの分類を述べアナログ・シミュレーションやハイブリッド・シミュレーションとの差異を理解させるとともに，この章の内容に関する包括的な説明を行い，シミュレーション全体にわたって留意すべき点を理解させる。

内　容

1.1　コンピュータ・シミュレーションの使命

コンピュータ・シミュレーションについて説明する場合，ただちにモデル作りの説明から始めてしまうと，シミュレーションを行なう目的とか前提などが不明瞭になってしまう恐れがある。このため，なぜシミュレーションをしなければならないか，シミュレーションをするとどのような結果が得られるのかといったことを具体的な例をあげて説明する。また，LP（Linear Programming），DP(Dynamic Programming）など解析的に解いて得られた結果と，シミュレーションで得られた結果とを比較して，それぞれの手法の差異を明確にすることによって，シミュレーションの特徴を説明する。

まず，システムの設計や運用にあたってシミュレーションによって各種のテストを数値的に行い，実測データの代りとして利用することによりシステム特性を定量的に把握できること，システムの特性を短時間に，しかも経済的に求められること，各種の試案の比較も簡単に行え，設計の最適化や新しい概念の導入も可能となること，実際のシステムでは実験できないような異常状態もシミュレーションなら簡単に調べられることなどを説明する。このような利用の具体例として超高層ビルの設計時にシミュレーションを利用して地震が発生したときのビルの揺れについて検討したことなどを話すとよい。また，企業において経営者，管理者が意思決定をする場合，シミュレーションを利用することにより，円滑

な経営を行っているといったことを話して，企業経営でもシミュレーションが必要であるということを説明する。このとき，在庫問題のシミュレーションなどの具体例を示したほうがよい。

つぎに，シミュレーションの発展過程にそって，どのようなことが行われ，どのような結果が得られてきたかについて説明する。すなわち，初期におけるシミュレーションの利用として，Buffon の針の問題，円周率πの近似値，定積分の求積などについて説明する。さらに，1950年代中期以降の企業の経営活動や経済分析におけるシミュレーションの利用について説明する。たとえば，初歩的な在庫問題などについて具体的に説明する。社会科学におけるシミュレーションの利用として，心理テストの結果に関する解釈や評価および言葉による説明などで利用した例や，国家的な緊急時における国家の意思決定者相互間の関連性を研究した例などをあげる。また，化学プラントや宇宙飛行の問題を解決するために利用された例などを列挙する。このようにして，シミュレーションの実績を述べ，シミュレーションが実際に利用されていることを理解させる。このとき注意することは，シミュレーションでなんでも出来てしまうという過信をもたせないことである。

さらにシミュレーションで求めた解とＬＰ，ＤＰなど狭い意味でのＯＲ的手法の解（解析的に求めた解）とを比較することによって，シミュレーションの特徴を説明する。例として，在庫管理の問題などをあげて，ＯＲ的手法ではコストが最小になる解（発注点）を求めるためには，コスト関数 f (x) を定義し，f (x) を微分してその微係数が0 となるx の値をみつけてこのx を最小コストの発注点とするが，シミュレーションでは，いくつかの発注点についてそれぞれある期間におけるコストを求め，そのなかでコストが一番小さい発注点をとりあげるといった手法を対比することによってシミュレーションの特徴を説明する。このとき，ＯＲ手法では，解析可能な関数（上記の問題では微分可能な関数）を導入しなければならない。さらに現実のシステムではほとんどの変数間の関係は非線形でしかも複雑であるにもかかわらず，モデルを非常に単純化してしまい，現実のシステムとは非常にかけ離れたモデルでシステムを評価してしまう危険が大きいが，シミュレーションを利用すれば，変数間の複雑な相互関係をかなり忠実に表現することができ上記の危険も減少されるということを明らかにしながら，シミュレーションの特徴を解説する。

また，シミュレーション・モデルには確率的要素が含まれており，分布の考え方が重要であることもあわせて述べる。

さらに，待ち行列の適当な例題，たとえば，窓口が1ケ所の食券売り場のモデルなどを例にあげて，待ち行列の理論で解析的に解く平均待ち時間とシミュレーションで求める。平均待ち時間の求め方の違いを明らかにしながら，それぞれで得られた結果の差異を説明する。シミュレーションでは，時間とともにモデルが変化する状態，とくに過渡状態のモデルの動きが明確になるが，解析的に解く場合は,定常状態しか明らかにならないということを強調する。

以上のような説明にもとづいて，シミュレーションとは変化する環境のもとでシステムが時間の経過にしたがって，どのような振舞いをするかをモデルとして再現し，モデルに対してさまざまな考察をすることであるということを理解させる。さらにシミュレーションとは，システムを設計したり，検討したりするときに利用する手段であり，シミュレーションができるから，システム設計や分析ができるということではないということを強調しなければならない。また，このようなシミュレーションを実施す

るにあたっては，コンピュータが不可欠であるということを述べ，さらに，確率，統計の知識も必要であるので，従来の知識を整理しておかなければならないということも述べる。

1.2 シミュレーション実施効果

情報の集中化を可能にし，実験提起の方針を与え，かつ実験の一般化の法則性を示す基盤となるべきシミュレーション理論やモデル化理論は，非常に異った領域における複雑な問題を解決していくうえで極めて効果的な手法であることを説明する。ここではシミュレーションがどのような状況において使用され，期待される効果が何であるかを具体的に説明する。目的が多様であることから実施効果の説明については，つぎのような主要なもののみに留める。

① 実際に動いているシステムのうえで実験する場合に，莫大な費用，危険あるいは時間がかかるので，実在あるいは計画中の複雑なシステムをモデルを用いて容易に実験出来る。

たとえば，アポロ計画において途中下車の出来ない宇宙軌道や月面状態を推測して地球上で各種の模擬実験を行い，着陸に成功した。

② 代替案を単に比べるだけでなく，当初の見通しや仮定が狂ったときどうなるか，そしてそのような場合にどういう手が打てるかを事前に研究出来る。

たとえば多品種少量生産の工場における生産計画の代替案を，見込み違いや特注品の飛込み，部品，原材料の納期遅れなどを考慮して評価する。

③ シミュレーション・モデルを使用して仮説を定式化し，テストすることが出来る。いろいろな決定規則や選択対象を評価したり，比較出来る。たとえば，人口集中・交通問題・住宅問題・土地問題などの複雑な要素を含む都市問題を解決するために何を対象にどのような施策を行なうかを評価・検討する。

④ 変化しつつある条件のもとで，システム構成要素の動きを観察し，システム構成要素相互間の関連性を見極め，重要な変数を明らかにすることが出来る。

たとえば企業において，販売戦略に対する競争者の反作用，販売計画に対する顧客の反応，顧客の行動パターン，売上の予測方法などの相互関係を分析し，効果的な因子に着目してマーケティング活動を行なう。

⑤ 教育・訓練用のシミュレーションを用いて思考法則並びに手法を具体的かつ有効な教訓として体得させることが出来る。

たとえば，工程の管理，解析，検査機能を有する製造工程における品質管理の教育・訓練のために，シミュレーションを用いる。このようにシミュレーションには非常にみのりある効果が期待されるわけであるが，あらかじめ目的をはっきり定めておかなければ単なるシミュレーションの遊戯に終り，貴重な時間と多額の金銭とを労費し逆効果をも生じかねないので目的と適用限界をよく理解したうえでシミュレーションを行うべきであることをあわせて理解させる。

1.3. シミュレーションの利用限界

シミュレーションを利用していろいろな考察を進めるとき，シミュレーションは万能でないということを明確にするために，シミューションの利用限界を説明する。とくにアナログ・シミュレーションとディジタル・シミュレーションを対比することによって，ディジタル・シミュレーションの限界を明らかにする。このため，図1－1のような自動車のサスペンション系のシミュレーションをアナログ型コンピュータとディジタル型コンピュータによって行なう場合を考えてみる。

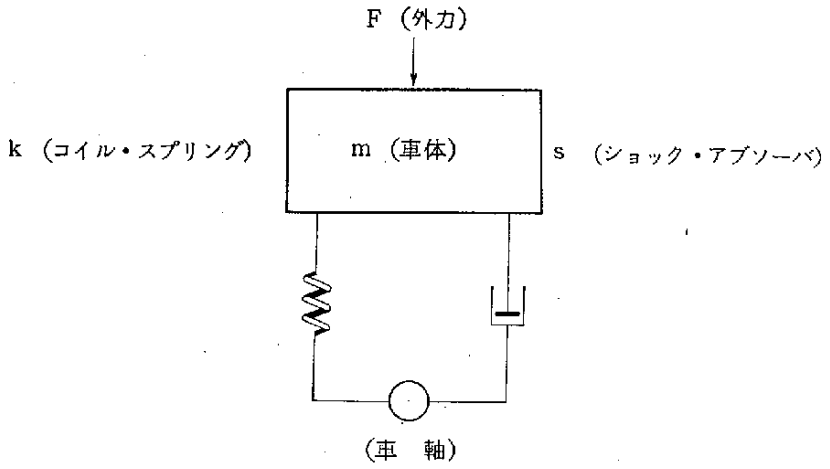


図1－1　自動車のサスペンション系

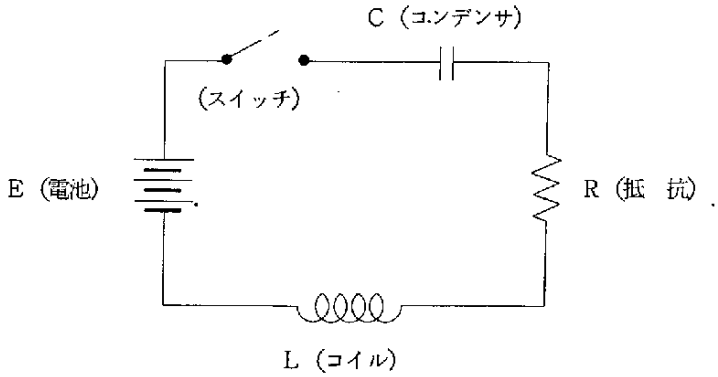


図1－2　電気的に相似させたサスペンション系

アナログ型コンピュータでシミュレーションを行なう場合は，自動車のサスペンション系を図1－2のように電気的に相似させて，シミュレーションを行なう。このように電気的に相似させた場合，電圧は外力を表わし，キャパシタンスの逆数はスプリングのスティフネスに相当し，抵抗は緩衝器のダンピング効果を示し，コイルのインダクタンスはスプリングによって支えられている車体の質量に等しい。時

間とともに変化するサスペンション系の垂直速度は電流によって表わされている。このような電気回路のモデルを構築してさまざまなシミュレーションを行ない，時間とともに変化するサスペンション系の特性をすみやかにしかも，安価にシミュレーションすることができるということを述べる，そして，ディジタル型コンピュータでシミュレーションする場合には，

$$F - k\int_0^t V\,dt - SV - m\frac{dv}{dt} = 0$$

といった微分方程式を，ルンゲ・クッタ法などの積分ルーチンを用いて解析的に解いて，サスペンション系の特性を求めるということを述べる。そして，ディジタル型コンピュータによってシミュレーションする場合は，アナログ型コンピュータでするように時間を連続量として扱うことはできず，あくまで時間間隔（微分方程式を解くキザミ）が存在していることを明らかにする。また，ディジタル型コンピュータによるシミュレーションにおいて，より連続形に近づけるため時間間隔を短かくすると，シミュレーションに費す時間がかかるばかりでなく，微分方程式が解けなくなってしまうことを述べて，ディジタル型コンピュータにおけるシミュレーションはあくまで時間に対して離散的であるということを説明する。

したがって，ディジタル型コンピュータでシミュレーションするときにとった時間間隔において，システムの変化を無視してもよいという仮定にもとづいて連続系のシミュレーションが行われていることを明確にしておかなければならない。

また，シミュレーションはあくまで，数学モデルにおいて実験をするので，現実のシステムを完璧に模擬することは不可能であり，かならず，仮定や前提が存在していることを明確にしなければならない。すなわち，この節ではディジタル・シミュレーションは，あくまでモデルの近似であるということを強調しておかなければならない。

1.4 シミュレーション実施手順

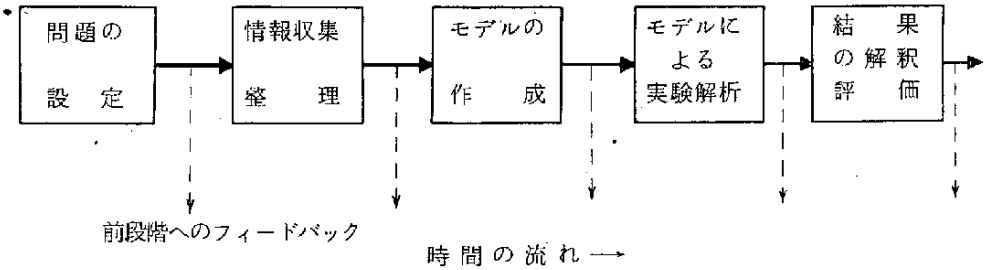


図１－３　シミュレーションの実施手順

まず図１－３のシミュレーション実施手順の実線部をシミュレーション研究の体系的アプローチでの論理上の順序として説明する。この場合，「問題の設定」，「情報収集」，「モデル化」，「解析」，「解析結果の評価」というＯＲ体系の一つの手法として，モデルによる実験という意味も含めて「解析」の部分を受けもつのがシミュレーションであることをあわせて説明する。これによりシミュレーションの位置づけを明らかにするとともに各実施手順を説明することによりシミュレーションの全体像を理解させる。

(1) 問題の設定・定式化

(a) 目的の明確化

まず研究の目的を明確かつ完全に定義する。これが不完全だと外部要因との関連が無視される危険性がある。研究結果の測定基準を設定し何を問題にしているかを明らかにする必要性を強調する。

(b) システムの記述

なるべく一般性のあるような形でシステムを記述する。すべてのシステムに適用出来るようなシステム理論はまだ存在しないが，「構成要素」，「パラメータ」，「相互関係」といった共通な特性により記述出来ることを説明する。

(c) 仮定事項の確認

結果の解釈は問題の定式化の際にどのような仮定を行なったかにより変ってくることを強調する。多くの場合，モデルの単純化のための仮定であり，重要でないと思われる特性変数を無視したり，非線型の関係を線型で近似したり，連続的な確率変数を離散値で近似したりすることが多く，こうした仮定事項の全体をよく認識していることが必要である。

(2) 情報収集・整理

システムのパラメータ（確率変数の母集団パラメータ，倉庫の収容能力など），システムの変数の動的特性（確率変数のパラメータ値の変動など），システム構成要素間の関係（構成要素の相互依存性，外部障害の影響など）といったシミュレーション研究に必要な資料的データを収集し，それを研究の目的に都合のよい形に変換することは非常にたいへんな作業であるが，同時に非常に重要なステップであることをとくに強調する。十分なデータが入手出来ない場合，実験を行なううえで重大な障害となる場合もあるし，データ収集中に対象システムに関する理解を深めることが出来る場合が多い。データの収集におけるサンプリング上の考慮，適合度検定などによる確率変数の分布型の決定，平均，分散，範囲といった分布のパラメータの決定，多重回帰方程式などによるシステム構成要素相互関係の決定などといったデータの整理のし方については，専門編「統計解析」で詳細に説明されているのでここではその必要性を述べるにとどめる。

(3) モデルの作成

シミュレーション・モデルは，現象のはたらきを研究するために，現象の本質を比較的簡単な数式や記号により表現したものである。したがってモデル作成の段階では，現象の本質をつかむことがもっとも大切である。ゆえに簡略化しすぎてモデルが実際の状況をほとんど模写していなかったり，あまりにも多くの特徴をモデルに持ち込みすぎて，モデルが手に負えない程面倒なものになってしまったのではシミュレーションを行う意味がないことを理解させる。その意味においても，出来れば単純化したモデルから次第に複雑なモデルへと順次改善して，所望のモデルを得ることが時間的にもまた経済的にも許されるならば，現象の本質を理解するうえで効果的であることを説明する。

「焦点となる問題に集約してモデルを設計する」ということが，この段階での主命題である。さらに，モデル化に関する一般理論といったようなものはまだ存在せず，多くの経験と勘に頼らざるを得ない部分であるので，常日頃から身のまわりに起こる現象を分析し，モデルとして描いてみるとか，

大いに好奇心を発揮し，現象の解明を試みる，あるいは，解明するためには何を知らなければならないかといったことに関心を持たせることも必要となろう。コンピュータ言語を用いてモデルを作成する場合には，シミュレーション用の言語を用いるか（GPSS，CSMPなど），汎用言語を用いるか（FORTRAN，PL／Iなど）によりモデル化の難易に差がでてくる。たとえば，FORTRANのような汎用言語をシミュレーションのために用いた場合には，時間軸の進行制御ルーチンや統計収集，出力ルーチンなどもすべてプログラム作成者が作成しなければならない。しかしながら連続形，離散形を問わず適用出来，必要な機能も手間さえ気にしなければ追加出来るといった利点もあるので，目的に応じて使いわけることが必要である。ただしプログラム作成者の多くの無駄と手間を省くためにシミュレーション言語が開発されているので，それを利用したほうが望ましいことを強調する。

(4) モデルによる実験と解析

このステップで行なうべきこととしてつぎのものを概説する。

(a) モデルの検定

一つは，モデルが論理的に正しくプログラミングされたか否かの検定，もう一つは現象を正しく表現しているモデルであるか否かの妥当性の検定

(b) 初期条件と過渡状態の除去

開始条件はモデルから得られる結果に影響を及ぼす。ピーク時の状態に注目している場合には空の状態を初期条件として与えることは誤まりとなる。またダイナミックス・モデルのように微分方程式でモデルを表現した場合，過渡状態（均衡点にいたるまでのゆらぎ）を生じる場合もあり，モデルをある程度ダミーで動作させてから実際の研究に用いるほうが好ましいこともあるといったことを説明する。そしてシミュレーションが過渡状態をも含めた動作特性を研究するためなのか，あるいは安定状態での値を求めるためなのかにより，初期設定が異なり，後者の場合には，均衡点に近い状態を初期設定し，過渡状態を取り除くことが必要となることを説明しておかなければならない。

(c) シミュレーション・モデルからのサンプリング

モデルの実行は，モデルで代表される現象の観測を目的として行なわれ，その結果モデルの運用時特性の予測を可能にする。したがってモデルから得られるデータに基づきモデルの状態を予測することの良し悪しは，基本的にはサンプリングの問題と見なし得る。ここでは一般的に知られている四つのサンプル採取法を説明するに留め，適当なサンプル・サイズとシミュレーション結果の精度の問題は，サンプリングの方法，モデルの形式および何がサンプリングされているかに関係しているということを注意事項として説明する。

(d) シミュレーション・モデルによる実験

シミュレーションでは，モデルの相異なる決定規則のテスト，係数を変化させた場合の行動変化の効果判断，代替的なモデル構造の評価といったものを一括して行う。このステップは実行時間を非常に費す部分であるので，計算時間を節約し，さらにシミュレーションから得られるデータの解釈のしかたを改善するという意味でも統計的実験計画法の必要性を説明する。その際に開始条件の

影響やモデル内の不規則な振舞いから生じる特性値の不規則変動を統制するように実験を計画することが重要となる。実験計画においては，とくにつぎの三点を強調する。

① どのように変動させる因子を選択（従属変数に影響を与える独立変数の選択）するか。

② 各因子の水準をどのように選定するか。(因子が「固定」しているか「ランダム」であるかを明確にする。因子の取り得るすべての水準を考慮しなければならないか否かを左右するので。)

③ どのように分析方法を決定するか。（分散分析法や多重回帰法，スペクトル分析などいろいろな手法があるが実験の計画を行う時点で分析法を決めておくべきことを説明する。

最適化手法に関しては，探索法といわれるような技法があるが，その適用には慎重を期することが必要であるといった程度の説明に留めたほうがよい。

(5) 結果の解釈・評価

モデルが信頼出来るか否かにより正しく評価出来るか否かが決まるのでモデルの妥当性を完全にチェックしておくことが絶対必要な要件となる。また最適解を得ることは不可能なので，どの程度の信頼度で結論を下したか，それに関連して実験条件，実験方法などを明確に記述しないと意味がないことを説明する

個々の説明が終わったならば，各ステップは一直線に行われるものではなく，行きつ，戻りつ，する多重のフィードバック・ループを繰り返すことにより次第に良いモデルが出来上っていくことを図1－3によって説明すればよい。

1.5 シミュレーションの分類

シミュレーションには，いろいろな分類法があるが，ここではコンピュータによる分類に関して下記分類にもとづいて説明し，この科目で扱うディジタル・シミュレーションの位置づけを理解させる。

使用するコンピュータによる分類

ディジタル・シミュレーション …… 実時間軸は離散的に扱う

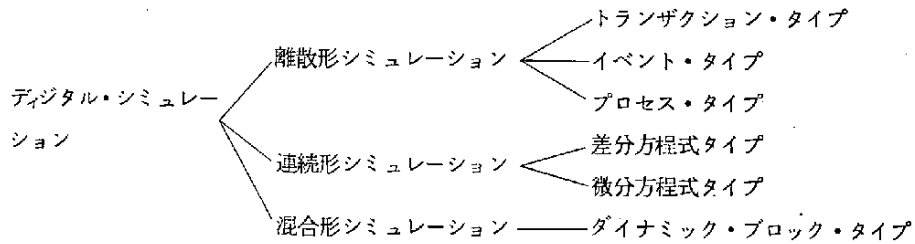
アナログ・シミュレーション …… 実時間軸は連続的に扱う

ハイブリッド・シミュレーション…… 実時間軸は離散的かつ連続的に扱う

ここでは，ディジタル・シミュレーションが離散信号を扱うものであることを説明する。

ディジタル・シミュレーションに関してもモデルの時間進行方式により**離散形**シミュレーション，**連続形**シミュレーション，**混合形**シミュレーションに区分され，さらに事象変化のとらえ方により，各種タイプに分類され，表1－1のようになされていることを説明する。

表1－1　シミュレーションの分類



つぎに，ディジタル・シミュレーションとアナログ・シミュレーションの特性比較を表1－2を用いて説明し，両者の差異を明確にする。

表1－2　ディジタル・シミュレーションと
アナログ・シミュレーションの差異

比較項目	ディジタル・シミュレーション	アナログ・シミュレーション
変数	離散数値表現	連続数値表現
要素の操作	順次操作	並行操作
実行速度	低速度	高速度
対象とする問題の拡大に対する追従性	記憶容量の増加と解答時間の増加。	大型コンピュータが必要となるが解答時間は増加しない。
精度と動作域	必要なだけ拡張できるが，計算時間の増加をもたらす。	物理的測定能力により制約を受ける。
積分能力	有限微分法により，積分しなければならない。	連続積分が可能
離散データの扱いと保存	容易	困難
実際のシステムとのコミュニケーション	アナログ・ディジタル・コンバータと実時間の計算速度が必要になり困難。	実際のシステムをシミュレーションの中に包含する事が容易。
実行時の操作性	解を得ているときには，通常，出力データを得ることは不可能 ただし，対話型のシミュレーションではこれが可能。	シミュレーション実行中にシステムの特性を簡単に見ることができる。

指導上の留意点

この章はシミュレーションに関する導入の意味を持っており，全体像，すなわち，シミュレーションの位置づけ，意義・効果・限界を正しく認識させるよう配慮しなければならない。そのためには歴史的な背景や具体例をふまえた説明が必要となるので，「参考文献」として列挙したなかから適宜選択してより具体的な解説を試みる必要がある。またシミュレーションの実施手順を説明する部分では，各ステップでの留意点を簡単に説明する程度で，詳細な説明は，第2章以降のモデルの作成，モデルの運用の章で随時行うようにしたほうがよい。

第2章　シミュレーション・モデルの作成

用　語　この章ではつぎの用語を教える。

制御可能変数，制御不能変数，パフォーマンス測定基準，目的関数，フロー・アプローチ，ファンクショナル・アプローチ，スティト・チェンジ・アプローチ，固定増分法，可変増分法，同期，サンプリング，擬似乱数，乗算合同法。

目　標

実際にシミュレーション・モデルを作成していくうえで考慮しなければならないことや，モデル化を可能にする各種技法 ならびに考え方，などを説明することによって，シミュレーション・モデルを作成する能力を養う。モデル化の技法を説明するときには出来る限り具体的な適用場面を提示しながら講義を進めたほうが効果的である。またシミュレーション・モデルをコンピュータ言語とくにシミュレーション言語を用いて表現する際に利用出来る有効なモデル化の諸概念を平易に説明し，基本的な考え方と表現能力を養うこともこの章の目的とする。

内　容

2.1　モデルの構築

(1) モデル化の考え方

シミュレーション・モデルを構築するときには，まずモデルを連続的に扱うのか，離散的に扱うか，あるいは両者を混在した形で扱うかを決定しなければならない。どういう形でモデルを表現すべきかを決定する厳密な選択基準を与えることは出来ないし，同一の対象を連続的にも離散的にも表現出来る場合もあるので，すでに微分方程式で表現されているモデルとか，類似モデルがあるとかいう具合に，モデルを問題なく構築出来る場合は別として，モデルの構築がむずかしい場合には，シミュレーションによって何を得ようとしているのかという問題を明確にとらえてモデルを構築することを説明する。

モデルを構築するときの基本的な考え方としてつぎのようなことを述べる。モデルとして表現する現象のなかには，現象を情報あるいはある種の物体の連続的な流れとして全体的に把握するほうが表現しやすくなる場合と離散形の変化が有限個集まって全体を構成すると考えてもさしつかえない現象，むしろそのほうが現象の本質を見極めるのに都合がよいものとがある。通常前者の現象に対しては，連続形モデルを，後者の現象に対しては離散形モデルを構築してシミュレーションを行なう。

ここで連続的表現のマクロ性（一群として扱ってしまう）とミクロ性（過渡現象解析），離散的表現のマクロ性（平均と分散で特性値を代表させてしまう）とミクロ性（個々の要素の一つ一つの行動を追跡する）といったことをあわせて説明する。

つぎにどんなモデルを構築する場合でも必要となるつぎの諸点に関する説明をまず概括的に行ない，さらにそれを解決するための方法に関して実例をとおして理解させる。

① 問題を理解し得る形に定式化する。

② 関連する変数やサブシステム間の相互作用を識別する。

①の問題の定式化に関しては，**制御可能変数**(C)，**制御不能変数**(U)，**パフォーマンス測定基準**(P)，**目的関数**(V) の決定が重要であり，Vi = g (Pi)　Pi = f (Ci, U) の形に問題を整理すると問題定式化の見通しが得やすい。

ただし，関数 f や g が一意に定まる場合は稀であり，いくつかの代替案を考えておくことが必要となる。

②の変数，サブシステムの同定に関しては，**フロー・アプローチ**，**ファンクショナル・アプローチ**，**スティト・チェンジ・アプローチ**の方法を説明する。

フロー・アプローチに関しては生産ラインのような物の流れのある例を引用し，まず具体的な例にもとづいて変数間にどのような流れが存在し，時間的にどう遷移してゆくかを考えさせ，一連の流れをサブシステムとしてとらえると便利であることを理解させる。

ファンクショナル・アプローチの説明においては，対象システムがいかなる機能を持ち，それに関連するインプットとアウトプットがなんであるかを考えさせ，それと他の機能とのかかわりあいや，機能のグループ化，階層化を通じてサブシステムを理解させる。

スティト・チェンジ・アプローチの説明においては，システムにどのような状態が存在し，状態変化を引き起こすものがなんであるかを考えさせ，状態変化を因果関係としてとらえさせ，状態変化と事象，事象のグループ化（アクティビティ）を理解させる。

(2) モデルの表現

モデルは，①構成要素，②変数，③パラメータ，④相互関係といったもので記述されることを説明する。構成要素とは，システムを構成する事物のなかで，それぞれ独立した単位と見なすことが出来るものであり，研究目的により，あるシステムでは，構成要素であるものが別のシステムでは外生変数になることもあり得るといったことを理解させる。また構成要素は，通常，静的構成要素と動的構成要素とに大別されることを述べる。

この分類は効率的にモデルを作成したり，あるいはモデル全体の見通しをよくするための便法に過ぎないが，SIMSCRIPTのような言語においては，パーマネント・エンティティ，テンポラリ・エンティティとしてはっきりとモデル化の概念が作られていることを述べ，エンティティの数がシミュレーション実行中不変のものは，パーマネント・エンティティとして定義しているといった説明をしておいたほうがよい。たとえば道路の交通モデルでは道路，交差点，信号はパーマネント・エンティティであり，自動車は，シミュレーション実行中にシステムに出入りしたりあるいは発生・消滅したりするテンポラリ・エンティティであることを説明する。変数は，条件あるいは，システム状態が変化すると値が変化するような属性であり，エンティティの属性を表現するものとしてはアトリビュートとシミュレーション言語において共通の一般的な用語となっていることを述べる。ここでは①独立変数と従属変数　②制御可能変数と制御不能変数　③内生変数と外生変数　④インプット変数とア

ウトプット変数の定義と相互関係を理解させる。（表2－1参照）

またコンピュータのうえで変数を定義する場合には，データの参照形態にしたがって①ローカル変数（一つのプログラムの単位で固有な変数），②グローバル変数（どのプログラム単位からも参照出

表2－1　　システム変数の交差分類

	独立	従属	制御可能	制御不能	内生	外生	インプット	アウトプット
独立			×			×	×	
従属				×	×			×
制御可能	×				×	×	×	
制御不能		×			×	×	×	×
内生		×	×	×				×
外生	×		×	×			×	
インプット	×		×	×		×		
アウトプット		×		×	×			

来る変数），③エクスターナル変数（本来一つのプログラム単位に固有なデータを他のプログラム単位からも参照出来る変数）といった変数の分類を行い，この分類がプログラム単位間でのデータの占有と共有という意味で重要となることを理解させる。また，パラメータとは，シミュレーション実行中の事象生起により影響されないようなシステム属性であり，各種分布の平均，分散，範囲といった属性やシステムのパフォーマンスに影響を及ぼすようなものであることを説明する。たとえば機械と修理工のモデルにおいては1時間あたりの故障件数の分布とか修理時間の分布の平均や分散や分布の形とか機械故障コストが修理工の1時間あたりのコストなどがパラメータとして選ばれるといったことを述べるとよい。パラメータの変化に対するシステム・パフォーマンスの感度を調べることが多くのシミュレーション研究において，重要な目的となっているため，シミュレーションの目的によってなにをパラメータにすべきかがおのずと決まってくることを理解させる。

つぎに相互関係について説明するわけであるが，この関係は①構造的関係　②関数的関係　③順序的関係などがあることを理解させる必要がある。構造的関係とは，構成要素とその属性とを結合する

ような関係，たとえば同一属性を持つ構成要素の集合（セット），あるいは階層を表現していることを述べる。関数的関係とは，構成要素の行動を全構成要素の状態や外生変数，制御可能変数，独立変数などの値の関数として規定することができる関係であり，たとえば，機械と修理工のモデルでは未修理時間の時間数を，それまでの作業時間中におきた故障の数，それらの故障の修理時間，修理工の数の関数として表現するといったことを述べる。順序的関係とは，位置と時間における関係であることを説明する。位置における関係としてはセットの表現において，属性値の昇順あるいは降順に構成要素を並べる場合などを説明する。時間における関係としてはシステムの運行中，時間的な従属関係によってのみ関連し合っているような事象の表現，たとえば修理が行なわれるのは故障が発生してから，ある時刻経過してからであるといったような関係を表現していることを説明する。

(3) タイミング・コントロール

シミュレーション・モデルにおける時間の進行方式として**固定増分法**と**可変増分法**を説明する。

コンピュータによってシミュレーションを行う場合には，なぜ，時間のタイミングが問題になるかというと，それは実際のシステムでは構成要素が同時に作動するのに，シミュレーションの場合は構成要素が逐次的にしか作動しないからであることを説明する。このため，シミュレーション・モデルにおける時間の進行機構は，モデルの構成要素の作動が実際のシステムと**同期**するような機能を発揮しなければならないので，時間の進行機構では，同一時刻に生起する事象に処理優先権を与えて競合状態を回避させているが，これはあくまで擬似並行動作であるから，この問題がディジタル・シミュレーションの一つの限界となっていることを理解させる必要がある。離散形モデルの場合，どのような時間の進行機構を使用すべきかということは研究対象とするシステムのプロセス性質とモデルの定式化のしかたにより決まるといったことを述べる。一般的な考え方としてつぎのことを説明するとよい。固定増分法は，事象が規則的に起るようなシステムとか確率変数間の関係が明確に規定されていない大規模システムとか確率変数間に既知の関係がないシステムなどのシミュレーションを行う場合に用いられる。可変増分法は，確率変数間の関係が明確に規定されているシステムとか，事象が既知の確率変数の関数として記述出来るシステムなどのシミュレーションを行う場合に用いられる。離散形のシミュレーションでは計算時間が短縮されるという点で可変増分法が明らかに優れているので，通常のシミュレーションでは可変増分法を採用しているということを説明する。

連続形モデルでは，差分方程式として表現するDYNAMOなどでは固定増分法を採用しており微分方程式の積分表現をしているCSMPなどでは，積分を計算する方式により，固定増分法や可変増分法のどちらかいっぽうを採用していることを説明する。連続形モデルの場合，可変増分法の採用は精度を向上させるためであり，固定増分法の1増分区間を所定の誤差範囲内におさまるまで細分割するので，離散形の場合にあげた特徴と異なり，計算時間の点で必ずしも有利とはいえない。

1増分区間内で微係数の符号変化があるような場合に固定増分法を用いると細部を無視する恐れがあるが，増分単位を小さくしてこれをカバーしようとするとフラットな特性の部分で無駄な計算時間を費やすことになる。このような場合に可変増分法が有効であるが，可変増分法の場合には，最小分割単位を与えないと不連続点で無駄な計算時間を費やすことになるので，求めたい精度に合わせて適当な最小時間を設定すべきことをとくに注意しなければならない。

つぎに，実際のシミュレーション言語で用いているタイム・テーブル(時刻表）の操作に関する説明を行い，タイム・テーブルも，時間という共通の属性を持つエンティティの集合（セット）として表現されていることを理解させる。テーブル操作として登録（スケジュール），再登録（リスケジュール），抹消（キャンセル），割込み，割込み解除，延期といった事柄を説明する。登録や再登録に関しては，単に時間属性によるものだけでなく，別の属性を利用して行う場合，たとえば，Aという事象名の前とか後とかいったことによっても出来ることを説明する。これは本質的にタイム・テーブルも一つのセットにすぎないからであるが，ＳＩＭＳＣＲＩＰＴでは可能だがＧＰＳＳでは不可能であるようにシミュレーション言語によってこれが不可能なものもあるので時間軸を様々な属性で制御しなければならない場合にはシミュレーション言語の選択が必要となる。

2.2 統計収集と出力

シミュレーションの実施に従ってシミュレーション・モデルから発生するデータをうまく収集し，見やすい形式で出力することによって正しくモデルの状態を観測しなければならない。このとき，コンピュータによるシミュレーションが終了したあと，手作業によるデータの整理をできるだけ少なくするようにプログラムを作っておかなければならない。シミュレーション・モデルから発生するデータにもとづいて，正しくモデルの状態を予測することは，基本的に，データの**サンプリング**の問題である。しかし，どのようにデータをサンプリングするかは，すべてのシミュレーション・モデルについて同じではない。シミュレーション・モデルの構造によりサンプリング形式は異なるし，サンプリング形式が異なれば，得られたデータの持つ意味も異なってくる。そこでシミュレーションにおけるつぎにあげるいくつかの基本的なサンプリングの考え方や問題点を説明する。

① シミュレーションの実行時間が一定ならば，サンプル・サイズを n とすることは，シミュレーション・モデルの相異なる実行を n 回するということになる。

② シミュレーション・モデルをきめ細く解析するためにはサンプル・サイズを大きくすればよいということではなく，モデルの動特性にあったサンプル・サイズを決めなければならない。このことは，シミュレーションを実行するときの時間間隔を決めることとも関係している。というのは，シミュレーションの実行時間を一定にしておけば，時間間隔 Δt を短かくすると，サンプル・サイズ n は大きくなるからである。

つぎにサンプリングの方法として，決定論的なモデルと確率論的なモデルにおける方法を説明する。

円周率πを求めるような決定論的なシミュレーション・モデルでは，有限の繰返し実行で，ある誤差以内の解を求められるため，サンプリングの問題ははとんどない。このようなシミュレーションにおいては，サンプリングの誤差と無関係に１回の実行で完全な情報を入手することができるといったことを述べる。

有限の繰返しで解を求めることが困難な決定論的モデルのシミュレーション，たとえば，在庫費用を最少にする発注量と発注点を求める在庫問題のシミュレーションでは，正確な解を得るために，在庫費用がある値に収斂するまで十分に繰返しシミュレーションを実行する必要があるといったことを強調する。表２－２は，シミュレーションの繰返し回数と求めた解（コスト）の関係を示した

ものである。表2－2によれば，250回以上繰返すと，ほとんど解が同じになっており，このモデルでは，シミュレーションの繰返し数，すなわち，サンプル・サイズは250回が適当であるということを示している。

表2－2　決定論モデルの結果におよぼすシミュレーションの繰返し数

シミュレートの繰返し数	シミュレーション結果		理論値
	トータル・コスト	コスト／年	コスト／年
50	$ 72	$ 361	$ 316
100	122	304	316
150	196	327	316
200	264	330	316
250	320	320	316
500	639	320	316
750	958	319	316
1000	1277	319	316

（250期間を年に換算して求めた。）

確率論的モデルのシミュレーションではサンプリングの問題が決定論的モデルのシミュレーションに比べて複雑であることを強調して，どのような情報を収集したらよいかを説明する。確率論的モデルでは，時間の経過とともに変化するモデルの状態や，モデル内を流れる個々の要素に関連する情報の信頼度を高めるため，複数回のシミュレージョンを実行したほうがよいということを述べる。このときのサンプル・サイズと情報の信頼度の関係を簡単に説明するとよい。

シミュレーションを実行して，どのような情報を，どれだけ収集したらよいかということは，最終的にはシミュレーションの目的，すなわち，なんのためにシミュレーションをするのかといったことにより決定されることであるということを強調する。

2.3 乱数発生ルーチン

シミュレーション・モデルを構築するにあたっては，モデルの構成要素のなかで，動作特性が確率変数で表わされるようなものは，各確率変数を規定する確率密度関数を求め，この確率密度数に合致した乱数を発生させなければならない。モデルの動作特性がどのような確率密度関数，すなわち，どのような確率分布の確率変数であるかを決めることは，かなりの経験が必要となるが，自然界に生ずる多くの確率分布は，正規分布，指数分布，二項分布，ポアソン分布などといった，従来から研究が積みかさねられている非常に少ない確率分布で近似することができることを説明する。

そこで，まず，母集団と標本の概念を明確にする必要がある。ここでは，母集団をある一つの特性の測定値を表わす数値の全体と考え，試料を母集団の部分集合と考える。つぎに，確率分布について説明する。確率分布は確率変数が離散的な集合に含まれることができるような離散形分布と，確率変数がとる値はある区間上で連続集合である連続形分布にわけて説明する。離散形と連続形の違いについて，ここであらためて解説する必要がある。離散形分布では，ベルヌイ分布，二項分布，超幾何分布，ポアソン分布，一様分布などについて，それぞれの確率密度関数，平均値，分散，さらに，各分布はどのような現象を近似することができるかなどについて説明する。たとえば，ポアソン分布については，確率密度関数は

$$P x = \frac{\lambda^{x}}{X!} e^{-\lambda} \qquad (\lambda > 0,\ x = 0,\ 1,\ 2,\ \cdots\cdots)$$

で，平均値，分散がともに λ であり，ある種の事象，たとえば「欠損」とか「変化」といった事象が時間的に，または距離的に，地域的に，数量的にランダムに生ずるような多くの場合に適用されることを説明し，つぎにあげるようないくつかの具体的な事象をあげるとよい。

① 単位時間あたりにサービス窓口に到着する人数

② トタン板に存在するキズの数

③ 1日あたり，交通事故で死亡する人数

④ 速記者が1ページあたりに犯す誤字の数

以上のような内容でいろいろな確率分布を説明するわけであるが，とくに，正規分布，指数分布，ポアソン分布については十分説明しなければならない。

シミュレーションを実行するにあたって上記の各確率分布における確率変数のランダムな値を選びだすために乱数を利用しなければならない。そこで，コンピュータにおいて乱数を発生させる方法をいくつか説明し，それぞれの乱数の発生方法，性質について解説する。シミュレーションを実行するときに使用する乱数が備えていなければならない性質，たとえば，周期が長いとか，再現性があるとか，統計的性質が十分でなければならないといったことを述べて，この条件をほぼ満足させ得るプログラムで発生させる**擬似乱数**の代表的な例を説明する。

平方採中法のような初期に開発された方法については，名前を列挙する程度にとどめ，ここでは，**乗算合同法**について説明する。とくに，擬似乱数系列を発生させる漸化式のパラメータと，乱数の周期の関係を明確に解説しなければならない。そして，ここで説明した方法にもとづいて，FORTRAN によって乱数発生ルーチンのプログラムを作成させるとよい。

ここで得られた乱数は，いづれも，ノン・ランダムな方法で発生した，一見ランダムに見える乱数であるから，異なる目的にこの乱数を利用するとき，乱数の異なる側面が影響して，結果を支配してしまうこともあるということを強調しなければならない。このためシミュレーションにおいて使用する乱数については，それぞれ，要求される性質を満足しているか検査しなければならない。そこで，乱数の統計的な検定が必要になるのでつぎにあげる検定について説明する。

① 頻度の検定（χ^2 検定）

② 系列相関の検定

③ 組合せの検定

④ ギャップ・テスト

⑤ 連の検定

⑥ 独立性の検定

いままでは，一様乱数を主として説明しできたわけであるが，さらに，一様乱数をもとにして任意の確率分布をもつ乱数をもとにして任意の確率分布をもつ乱数を発生する方法について説明する。まず一般の乱数発生アルゴリズムについて説明し，ついで指数分布，正規分布などの特定の分布，さらに，ポアソン分布と指数分布の関係などを解説する。一般の乱数発生アルゴリズムとしては，一様乱数を使って累積密度関数から任意の確率分布の乱数を求める方法や，フォン・ノイマンによって提案された棄却法や，合成法，合成棄却法などについてアルゴリズムを中心に説明する。

また，正規分布の乱数を中心極限定理の応用した方法で発生させるアルゴリズムについて説明する。代表的な例として，〔0, 1〕区間の一様乱数を12個発生させ，それらを加え合せたものから6.0を引いた値が，平均値 0, 標準偏差1の正規乱数になることを説明する。また，中心極限定理についても，簡単に説明したほうがよい。

また，窓口に到着する人の時間間隔と，単位時間あたりに窓口に到着する人数の関係を示しながら指数乱数からポアソン乱数を導き出す方法を説明する。この説明によって連続値と離散値の区別が明確になるであろう。

指導上の留意点

概念を説明するだけでは，何らモデル作成能力の向上を期待出来ないので，つぎの事項を実際に行なわせるとよい。

① 問題の定式化をさせる。

② フロー，機能，状態を抽出させる。

③ 構成要素，変数，パラメータ，相互関係を抽出・整理させる。

④ 事象生起の様子を時間軸上に表現させる。

⑤ セットに関する基本的な操作（挿入・削除）を実現出来るポインタ処理のフローチャートを書かせる。理解の程度により適宜ＦＩＦＯ，ＬＩＦＯ，RANKEDを扱わせてもよい。

⑥ 確率分布を一様乱数を用いて表現させる。離散形分布と連続形分布の累積分布を求めさせてから，前者に対しては，離散値の属性とリスト属性を求めさせ後者に対しては線形補間を行なわせる。

第3章　モデルの運用

用　語　この章ではつぎの用語を教える。

過渡状態，定常状態，初期条件，探索法，シンプレクス法

目　標

第2章において，シミュレーション・モデルの設計と構成に関する概念・技法に関して述べたので，この章では，モデルが作成された後，シミュレーションの実施段階で有効となる概念，各種技法に関する説明を行う。モデルを妥当性あるものにするにはどうしたらよいか，初期条件や過渡状態の影響が無視し得ない場合はどうするか，モデルからのサンプリングをいかに行うべきか，代替的構造の評価や特性変数変化の効果判断，モデルの相異なる決定規則のテストといったモデルによる実験をどう計画し，実行すべきか，実験を効果的に行うには最適化技法をどのように適用すべきかという基本的な考え方を理解させ，その手法を修得させる。

内　容

3.1　初期設定とパラメータの変更

初期状態，**過渡状態**，**定常状態**の説明を行い，モデルによっては平常の稼動特性に達するまでに何サイクルかのウォーミングアップが必要であることを説明する。平常時の稼動特性を解析するのに空の状態からシステム・パフォーマンスの測定を始めたのでは，統計的に意味がない場合もあることを指摘する。過渡状態にあるシステムを観測する場合には，均衡問題には関心がなく，**初期条件**の与え方が重要となり，また連続形モデルでよく見られるような境界値問題に関しても，初期値の設定が重要となることを注意する。定常状態におけるシステムの稼動特性を解析する場合には，システムが極限分布に従って動作する時点を決めることが問題になる。均衡状態を作り出すには ①はじめのある期間に生成されたデータを無視する ②システムの安定状態に近いような初期条件を選ぶ ③適当な初期条件を設定してシミュレーションを一定期間行い，その最終状態を実際のシミュレーションの初期条件とするといった方法があるが，適切な初期条件を作る規則はないので，研究の目的，必要な精度，システム構成要素間の関係についての知識，経験の豊かさなどによりどのような方法を使用すべきかを決定しなければならないことを理解させる。

つぎにGPSSとSIMSCRIPTの初期設定の方法に関する説明を行なうことによって，一般的な初期設定の方法を理解させる。GPSSの場合初期設定用のダミー，Xactを作動させてSAVEVALUEやPARAMETERを初期設定する方法と，INITIAL命令により初期設定する方法を説明する。

ＳＩＭＳＣＲＩＰＴの場合，パーマネント・アトリビュートの初期値とテーブルの設定を System Specification Card と Initialization Card で指定するが Initialization Form ではアレイ番号順に記入しなければならないこと，ランダム・ルックアップ・テーブルにより確率分布を初期条件デックの一部として与えることを説明する。

3.2 モデルの検証

シミュレーションを実施していくうえで，モデルが，対象とするシステムを十分に表現しているかいないかは重要な問題である。このため，まず数学モデルの段階でモデルが妥当であるかを検証し，さらに，コンピュータでシミュレーションをするためのモデル，すなわちプログラムの検証をしなければならないことを説明する。そして，モデルをどのような項目について，どのように検証するか解説する。

まず，シミュレーションを行うためのプログラムを作るまえの段階では，モデルが研究の対象とするシステムを十分に表現するように定式化されていることを確認しなければならない。このとき，モデルが実際のシステムを厳格に表現していなければならないということではなく，システムの重要な特性がモデルにとりいれられ，システムの数学的，論理的な因果関係によるシステム構成要素の真の行動がモデルに十分に表現されているならば満足しなければならないということを強調しなければならない。モデルを理論的に完全に検証することは非常に難しいが，モデルを検証するつぎのような方法を説明するとよい。

① モデルの分布やパラメータや因果関係を決めるために使用したデータが適切であったかを確認するため，モデルで使用した分布の極値，分布の形，平均値などと実際のデータを比較する。

② いろいろな手計算を行い，計算結果として得られる行動の特性を調べることによってモデルの正しさを検証することができる。たとえば，ある結果は決して負になってはいけないとか，ある結果は当然，時間的に周期的な性質をもっていなければならないといったことを検証する。また，モデルを作成するときに決めた仮定が正しいか検証する。たとえば，独立性，数値の丸め方，区間の幅，線形成などに関する仮定の検証を行なう。

③ モデルのなかで，なにか重要な因子が省略されていないかどうかを検討しなければならない。モデルの最終的な価値は，モデルが研究対象のシステムやプロセスの行動をどれくらい正しく予測できるかによって決まる。間違った予測をする可能性をできるだけ小さくするためには，システムの構成要素間の関係が正しく規定されていることを確かめることが重要であり，データが正確でなければならない。また，重要な変数に関しては感度分析を行うべきであるといったことを丁寧に説明しなければならない。

モデルの検証が完全に行なわれるならコンピュータで実行するシミュレーション・モデルすなわちプログラムに対しては単にプログラムのデバグだけでよいと思われるが，実際には，さらに，この段階でもモデルの検証が必要であることを強調しなければならない。コンピュータでモデルを動かしてみてはじめて，数学モデルの欠陥が表われてくることがしばしばある。また，手計算でモデルを検討しても，それは，単に「ウォーミングアップ」の状態の検証にしかならず，均衡状態の検証はやはり実際にコンピュータでプログラムを実行してはじめて可能になってくる。

さらに，相互関連性のある多くの構成要素からなる複雑なモデルでは，各要素の同期化が大きな問題である。この同期化の問題を検証するには，やはりプログラムを実際に実行してみなければならない。このような理由を述べて，プログラムの段階での検証の必要性を説明する。そして，この検証のやり方としてつぎのような方法を述べる。

① 過去のデータを用いてプログラムを実行し，それによって得た結果と過去の既知の結果とを比較することによって，数学モデルまたはプログラムを検証する。

② 確率論的モデルから不規則な要素を取り除いて，決定論的モデルとしてモデルのシミュレーションを行ない，モデルやプログラムの検証を行なう。

③ 複雑な確率分布を初歩的なものと取りかえて，モデルのシミュレーションを行ない，結果を容易な形で実証することによって，モデルやプログラムの検証を行なう。

④ 単純なテスト環境を作り，モデルの周囲条件の多数の組合せと同じだけシミュレーションを行なうことによって，モデルやプログラムの検証を行なう。

シミュレーションを行って得られた結果と現実の既知の値を比較するためには，一般的な統計的方法，すなわち，平均値の差の検定や χ^2 検定などを使うことができるといったことを説明しなければならない。モデルの検証ということは，非常にやっかいな問題であって，まだ多分に技法的なものであるから，相当な経験を必要とするもので，ここでは，なるべく，具体的な例をあげて説明したほうがよい。

3.3 シミュレーションの結果の分析

シミュレーションを実行して得られた結果をどのように分析し，どのように解釈したらよいかを定量的に行う手法を説明する。しかし，シミュレーションの種類によっては，一般的な統計的方法による分析を行わずに，結論が引き出されることも可能な場合があるということを注意しておく必要がある。たとえば，インダストリアル・ダイナミックスのモデルでは，変化する環境のもとで，対象システムがどのように変化するか，すなわち，システムの安定度と長期的な傾向がどうかといった二つの点で評価されるので，この評価は，シミュレーションによって得られたアウトプットを観察して容易に得られる。このため，統計的方法が，実験の結果を評価するのに常に必要であるというわけでもないことを強調しなければならない。

そこでモデルから得られた結果を統計的方法で評価するときの注意点を説明する。シミュレーションを行なって得られた結果の差異は，つぎの理由によるものと考えられる。

① 初期条件による影響

② モデル内の不規則過程によって表われてきた結果の不規則変動

③ シミュレーションによって表われた変化

シミュレーションでは，①，②の影響は押えて，③の変化だけを抽出し，解析しなければならないということを説明する。この結果の分析にまず分散分析法を用いる方法を述べ，この場合は，運用規則の各組み合わせに対して別の乱数列を使用しなければならず，シミュレーション特有の利点，すなわちまったく同じ条件のもとでいろいろな対立案の実験が可能であるということが損なわれてしまうということを注意しなければならない。シミュレーションの結果を分析する方法として，結果を順位法あるい

は多重対照法によって比較する方法も説明する。さらにもう1つの方法として，スペクトル分布を使用する方法について説明する。この方法は，自己相関をもつシミュレーションの結果を解析するときに有力な手段であるということを強調しなければならない。

シミュレーションの結果を解析するには，なるべく定量的な判断を下す必要があるため，上述の方法以外にも，さまざまな統計的方法を使うことも説明したほうがよい。

3.4 シミュレーションの最適化

種々の初期条件とパラメータを動かしながら，シミュレーションを繰りかえし最適な初期条件とパラメータをみつける方法について説明する。むやみやたらと，初期条件やパラメータを変更して，多数回のシミュレーションを繰りかえすことは，能率的でなく，シミュレーションのコストが増大してしまうので，シミュレーションは効率よく行なわなければならないということを注意する必要がある。

シミュレーションを効率よく行う方法として最適値**探索法**を説明する。その一つである**シンプレクス法**の説明をするために，簡単な在庫問題のシミュレーションなどを例にあげるとよい。シンプレクス法は在庫コストが一番安くなる発注点と発注量の組合せを，あたかも山を登っていくように求める方法であるということを説明する。シンプレクス法の詳細は参考文献(2)を参照するとよい。

シミュレーションのプログラムを作成するにあたっては，このような最適値探索プログラムを付加しておき，自動的に特定のパラメータの最適組合せを探索するようにしたらよいことを説明するとよい。この自動最適値探索プログラムは，完了したシミュレーションの結果を分析し，次回のシミュレーションの実行に使うパラメータを決めているにすきず，シミュレーションの実行に人間が結果の分析者として完全に排除されてしまうということを強調しなければならない。人間は，シミュレーションを開始するにあたって，初期条件や開始時点および探索係数をプログラムに与えなければならないし，いくつかの判断をプログラムの探索終了後に行なわなければならないことを十分注意しなければならない。

指導上の留意点

この章で解説する概念・技法は，実際問題としては，モデルが運用されるまでは利用されるものではないが，モデルを構築する際に十分考慮しなければならないものであり，運用計画をふまえてモデル化がなされていないと，実験が出来なくなる場合もあるので，この点を強調しておく必要がある。

第4章　シミュレーション言語

用　語　この章ではつぎの用語を教える。

ブロック・ダイアグラム，ブロック・トランザクション，属性，事象，エンティティ，ストレイジ，ファシリティ，セット，レベル，レイト

目　標

この章では，各種のシミュレーション言語がもっている機能を，使用例とともに説明することにより，第3章の"モデルの作成"で述べられた事項が，シミュレーション言語でどのように扱われているかを示し，いくつかの基本的な現象を言語を用いて記述する演習もあわせて行なう。ここではとくに4種類のシミュレーション言語をとりあげ，その特徴，使用法を理解させる。細部の記述法の話にまでは立入らずに，基本的なシミュレーション言語の性格と機能，表現法に関する一応の理解が得られることを目標とする。最後にシミュレーション言語の選択基準を述べ，4通りのシミュレーション言語の特徴比較をまとめ，なにをよりどころに選択すべきかを理解させる。

内　容

4.1 GPSS (General Purpose System Simulator)

(1) GPSSのシステム構成

まず適当な例題の**ブロック・ダイアグラム**，コーディング例を示し，これにそって主要命令を説明する。実際のプログラム・リストと出力リストを示すことにより，直観的にどういうものであるかを理解させてしまったほうが個々の説明に対する理解を早め得る。例題はあくまでも導入の意味で用い，最初の数**ブロック**と主要ブロックの説明にとどめる。この例題でGENERATE ブロックで発生され個々のブロックを移動した"もの"がGPSSのモデル化概念である**トランザクション**であることを指摘し，流れを基調としてモデルを構成する2.1で述べたフロー・アプローチの一つとなっていることを理解させる。またデータの内部構成を示してトランザクションという概念を説明してしまったほうが混乱が少なくてよい。トランザクションのパラメータにより**属性**が表現される。たとえば，属性を変えれば銀行窓口でサービスを受ける客とか，コンピュータ・ネットワークを流れるジョブ，製造工程を流れる部品などがトランザクションとして表現されるが，必ずしも具体的なものである必要はなく，仮想のもの，たとえば，ランダムに異常事態を発生させるための制御用トランザクションなども考えられることを説明する。GPSSによるシミュレーションとは，トランザクションが正しい時間の順序に従ってブロックからブロックへと動くことにより，**事象**を起こさせるので，このトランザクションの進行は，ブロック命令の実行にほかならないことを指摘する。このトランザクションの進行を制御するタイミング・ルーチン

はオーバー・オール・スキャンと呼ばれGPSSの中枢部を形成するものである。ここでGPSSのタイム・テーブルであるイベント・チェイン（Current Event Chain, Future Event Chain）をオーバー・オール・スキャンが走査し，時間を進行させる様子をGPSSのシステム構成図とともに説明し，全体像を理解させるとよい。GPSSの主なシステム構成は次のようなものがある。

① GPSSの制御

シミュレーションに関するすべての仕事（アセンブル，モデル入力，シミュレーション実行）を総合的に管理する。

② GPSSのアセンブラ

③ GPSSの入力

モデルの入力とブロック命令群の設定

④ GPSSのオーバー・オール・スキャン

・時刻の更新

・CECのトランザクションのスキャン

・トランザクションを次のブロックへ進める。

⑤ GPSSの出力

(2) エンティティ・エラー出力と統計情報の出力

つぎにGPSSでモデル表現に用いられる諸要素を抽象化した**エンティティ**に関してグループ単位に説明する。エンティティには，ある時刻におけるその要素の状態を記述する属性（アトリビュート）が定義されており，数値的なもの（Standard Numerical Attribute）と論理的なもの（Standard Logical Attribute）があり，記号を用いて外部より参照出来る。これらに関しては代表的なものを二,三（たとえばStorage に関する Rj とか，乱数RN′など）説明するにとどめ，一覧表を示せばよいであろう。エンティティに関しては，

① 基本エンティティ

Block

Xact

② 装置エンティティ

Facility

Storage

Logic Switch

③ 計算エンティティ

Arithmetic Variable

Function

④ 統計エンティティ

Queue

Frequency Table

⑤ 参照エンティティ
Save Value

⑥ チェイン・エンティティ
User Chain

に分けて説明するとよい。

各ブロック命令は，エンティティに関連づけて説明すればよい。すべての命令を説明することは時間的に不可能であり，研修生を混乱させるだけなので，各種エンティティに関連して特徴的な命令を重点的に説明し，それ以外は，実際にモデルを作成させる実習の部で補足すればよいであろう。

(3) GPSSの特徴

最後に離散形シミュレーション言語では，現在もっともポピュラーな待ち行列のモデル向きの言語であるGPSSの特徴として以下の点を説明し言語選択の一つの基準を与える。

① トランザクション中心の見方を基本としており，現象の記述が直接的であり簡便かつ具体的である。

② システムの構成要素の表現が簡明であり，特性に応じて**ストレイジ**（Storage），**ファシリティ**（facility），ロジック・スイッチ（Logic Switch）を記述することが出来実際的である。

③ 基本命令がBlock と称するマクロ・コマンド形態をとっているので，はめ絵的にプログラムを構成出来，作成し易いとともにデバグも簡単である。

④ 実用性に富む反面，システムとしての融通性，実行速度（インタプリティブ形式），GPSSが使用する記憶容量に問題がある。

4.2 SIMSCRIPT（Simulation Scriptor）

SIMSCRIPTによるモデル作成の基本概念として，エンティティ，アトリビュート，**セット**，ステイト，イベントを説明する。（表4−1参照）

表4−1 SIMSCRIPTの構成分類表

- モデル
  - ステイト（状態）
    - エンティティ（構成要素）
      - パーマネント（恒久的）
      - テンポラリ（一時的）
    - アトリビュート（属性）
    - セット（集合）
  - イベント（事象）
    - イクソジナス（外生）
    - エンドジナス（内生）

さらに表4−1の分類を示し，用語を理解させる。

任意時点におけるシステムの状態は，そのときに存在する個々のもの（エンティティ）のすべての属性の現在値と，そのエンティティがどのセットに属しているかにより表現されることを理解させる。

状態の変化を記述することによりシステムの動きが表現される。状態の変化を表現するには，エンティティを発生・消滅させるとか，属性値を更新したり，セットへエンティティを出入りさせるといっ

た操作を記述すればよいわけで，状態の記述と状態変化の記述がSIMSCRIPTによるプログラミングの一番重要なことであるということを理解させる。エンティティに関してはFORTRANにはないテンポラリ・エンティティという概念を，ダイナミック・アロケーションによる主記憶装置の有効利用という意味も含めて説明する。テンポラリ・エンティティに関連して，イベント・ノーティスという普通のテンポラリ・エンティティ機能に自動的なタイミング機能を付加したような特殊なテンポラリ・エンティティを説明しイベントとの関連を明らかにする。セットに関しては，サブスクリプトによる包含関係，すなわちサブスクリプトなしのセットはシステム全体に属し，1次元のセットは1種類のエンティティに属し，2次元のセットは2種類のエンティティに属し，オウナ・エンティティとメンバ・エンティティにより所属関係が規定され，複雑なデータ構造を簡便に表現出来ることを示す。またセットに関する操作を効果的に行うためのポインタを利用したリスト構造とFIFO，LIFO，RANKEDの関係を説明する。イベントに関しては内生事象と外生事象の相異，すなわち，事象生起時点の指定の仕方の違いを説明し，内生事象の生起時刻を指定する時点がいつになるのかとかシミュレーション実行前にその時刻を決定出来ず時間間隔に基づいてスケジューリングを行なうべきことを理解させる。SIMSCRIPTは汎用プログラミング言語としての性格が強く，FORTRANプラスアルファの機能を有しているため，FORTRANを理解している研修生にとって理解し易い反面，意識の変革を要求されるのでそれを理解させるのはむずかしい。この理由のもう一つとしてイベントの概念を認識させることの難しさがある。

というのはイベントの概念というのが，あまりにも一般的であり過ぎ，しかも具体性や実用性に欠けるきらいがある。また幾多の同時並行現象，関連事象を体系化し，相互作用を矛盾なく把握し，論理的に整合した事象ルーチン群を構成することの難しさをあげることができる。したがって簡単な例題に基づきイベントの概念を容易に理解することを中心にしてSIMSCRIPT言語によるプログラミングを説明する。そのための例題としては窓口モデルについていえば，単一窓口であり，客の到着時間と基本サービス時間が決まっており，実際のサービス時間が一様乱数を用いた1次式で表現され，セット操作の説明のためにサービス優先順位を到着順序に従って行うといった簡単なものを選択すれば十分である。

この例題にそくしてプログラミングの過程を説明する。

(a) 状態のプログラミング

SIMSCRIPT特有のディフィニィション・フォームを示し，テンポラリ・エンティティ（客）の属性として，S“セット客”（後続客），オウナ・エンティティに対してF“セット客”（先頭客），L“セット客”（最後層客）がポインタを収容するために必要となることを説明する。同時に内生事象を生起させるために必要な特殊なテンポラリ・エンティティであるイベント・ノーティスの説明も行なう。

(b) 状態変化のプログラミング

① イベント・リスト

外生事象は，名前に対応する識別番号を必要とし，外生事象カードで外生事象の生起すべき時刻を指定するときに利用されることを説明する。

② イベント・ルーチン

イベントを内生・外生とに明確に区別していることの利点として，システム・モデルを作成する

場合の境界の設定という問題に触れ，境界における外部からの影響を，時間管理された入力として簡便に受け入れられることを説明する。

③ システム・スペシフィケーション，イニシャリゼーション，イクソジナス・イベントなどの各ヤードの簡単な説明を行ない，どういう順序で何を用意すべきかを理解させる。

(c) シミュレーションの実行

まずSIMSCRIPTにおいての準備段階として行なう事項，すなわちディフィニション・フォームとイニシャリゼーション・カードとの内容によるパーマネント・アトリビュートの初期値設定，イクソジナス・イベント・カードは内容によるイベント・チェインの構成を説明する。イベント・チェインの内容を時間軸にそって図式化するとよい。タイミング・ルーチンがこのイベント・チェインの内容を更新する様子を例題にそくして解説し，タイミング・ルーチンの機能としてつぎのことを理解させる。

①イベント・チェイン上で時刻の一番早いものを探す。②時刻をその事象の生起時刻に更新する。③制御をその事象ルーチンに渡す。④制御が戻って来たら事象をイベント・チェインから削除し再び①に戻る。

イベント・ルーチンのサブルーチンとの差異を理解させることが重要である。

最後にSIMSCRIPTの利点と欠点とを整理して説明するとよい。

4.3 DYNAMO（Dynamic Models）

DYNAMOは，インダストリアル・ダイナミックス（Industrial Dynamics）によって分析され，構築されたモデルのシミュレーションを行うために開発された言語であるから，インダストリアル・ダイナミックスの考え方がある程度わかっていなければならない。そこで，つぎのような簡単な例題をあげて，モデルを作り，DYNAMOによるコーディング例を示すとよい。

アパート建設業の意思決定問題で，アパート需要に対処したアパート建設個数の決定をどのようにしたらよいかを解析する問題である。仮定は，つぎのとおりである。

① アパート着工から完成までは3年かかる。

② 正常なアパート空室数は 1000 ユニットとする。

③ 初年度の実際空室数は正常な空室数1000ユニットと等しい。

④ 初年度の新規貸室数は2000ユニット／年でこの値は，新設着手室数および完成室数と等しい。

⑤ 2年目からアパート需要が急増して 2,200 ユニット／年となり，以後その水準に安定する。

このとき，アパート新設数の意思決定ルールを，「その前年のアパート新規貸室数に正常空室数と実際の空室数の差の 1/2 を加えたもの」とすると，アパートの空室数などがどのように変化するかを解析する。このモデルをDYNAMOでコーディングするとつぎのようになる。

表4－2　DYNAMOのプログラム・リスト

SPEC	DT = 1/LENGTH = 20/PRTPER = 1/PLTPER =1
NOTE	MODEL OF APARTMENT INDUSTRY
1L	AVN. K=AVN. J+ (DT) (CN. JK−LN. JK)
40 R	NCN. KL = LN JK + (½) (NVN−AVN. K)
37 B	CNB = BOXLIN (5,1)
36 N	CNB = BOXLCAD (0,0)
1 L	CNB＊1.K = CNB＊1.K + (DT) (NCN. JK+0)
6 R	CN. KL = CNB＊3K
45 A	LNG. KL = STEP (200, 2)
7 R	LN. KL = 2000 + LNG. K
6 N	AVN = 1000
6 N	LN = 2000
6 N	NCN = 2000
C	NVN = 1000
PRINT	1) AVN/ 2) NCN/ 3) CN/ 4) LN
PLOT	AVN = NCN = N, CN = C, LN = L
NVN	正常空室数　（ユニット）　C
AVN	実際空室数　（ユニット）　L
CN	完成室数　（ユニット/年）R
LN	新規貸室数　（ユニット/年）R
NCN	新規着手室数（ユニット/年）R

この例を使いながら，DYNAMOの使い方，機能，構造などを説明するとよい。まず，DYNAMOで使用する変数や定数のあらわし方や変数の種類，すなわち，**レベル**，**レイト**の変数や補助変数，定数などについて概説する。

つぎに，プログラムの中で使う方程式を，例題のプログラム・リストにそいながら解説する。このとき，モデルの時間表示をどのように行っているかについて，プログラム・リストと表4－3を使って説明する。

表4－3　時間表示表

方程式の左辺にある変数（定数を含む）の種類	左辺の変数（定数を含む）の時間表示	右辺の変数（定数を含む）の時間表示					
		レベル L	補助変数 A	レイト R	サプリメント S	定数 C	イニシャル・バリュー計算される定数 N
L レベル	K	J	J	JK	—	なし	なし
A 補助変数	K	K	K	JK	—	〃	〃
R レイト	KL	K	K	JK	—	〃	〃
S サプリメント	K	K	K	JK	K	〃	〃
C 定数	なし	—	—	—	—	—	—
N イニシャル・バリュー計算される定数	〃	—	—	—	—	なし	なし

さらに，DYNAMOの利点として，つぎのような項目を簡単に説明する。

① 方程式の形式がきまっているので，プログラミングが簡単で理解しやすい。

② 表とグラフの形で出てくるアウトプットの指定が簡単である。プログラム・リストのPLOTとPRINTのステートメントについて説明するとよい。

③ モデルを構成する方程式をならべる順序は任意でよい。

また，DYNAMOの問題点や使用上の注意点などをよく説明しなければならない。ここではとくに，DYNAMOで行うシミュレーションがアナログ・シミュレーションのような機能を発揮するような解釈をされる危険性を有するので，この点を注意する。

すなわち，シミュレーションを進めていくためには，かならず時間間隔DTが存在し，この単位時間のなかでは，モデルの動作がスタティックで差分方程式の形式で表現されているということを説明しなければならない。また，シミュレーションを実行中にシミュレーションのプロセスを変更することは困難であるとか，プログラムのなかで表現できる式の形式がきまっているとかいった問題点や，DYNAMOを使うことができるコンピュータは，ごく限られたものしかないといったことも注意すべきである。

ここで説明したDYNAMOはIBMの704, 709, 7090, 360や，HITAC 5020, FACOM 230−50／60などに開発されたものであるが，このDYNAMOのいくつかの制約条件を除いたDYNAMOⅡもすでに開発されているということも最後に付け加えたほうがよい。たとえば，DYNAMOⅡでは，方程式を制限なしに，代数の規則に従って作成することができるとか，DYNAMOにおいては，モデルの実行を停止するような誤りでも，DYNAMOⅡではシステム自身がその修正を試みてくれるといった点などについて説明したほうがよい。

4.4 CSMP (Continuous System Modeling Program)

CSMPモデルの構成とは，ある時刻において系を構成する状態変数間の相互関係を数学的または数値的に表現するということであることを簡単な例題（たとえば，ダンパがあるバネの振動現象）をもとに説明する。(図4−1参照) ここではつぎの事項を説明するとよい。

① モデルの数学的記述はどうなるかたとえば，ダンパが速度の2乗に比例して作用するものとすれば，

$$\begin{cases} m\ddot{x} + c\dot{x}^2 = -kx. \\ x(0) = a\ ,\ \dot{x}(0) = 0 \end{cases}$$

② 系の状態変数間の関係はどうなるのか

$$\begin{cases} \ddot{x} = -\dfrac{1}{m}(Cx^2 + kx) \\ \dot{x} = \int_0^t \ddot{x}dt + x(0) \\ x = \int_0^t \dot{x}dt + x(0) \end{cases}$$

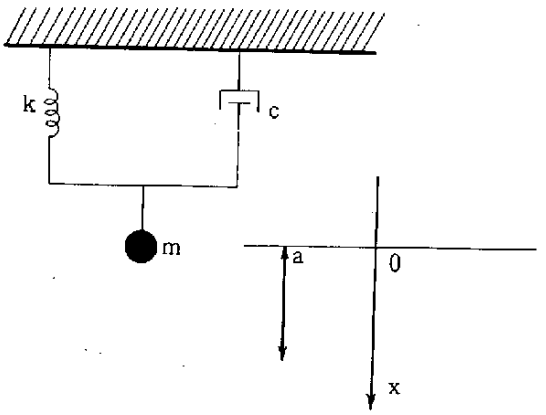


図 4−1 バネの振動

この時系を構成する状態変数（$\ddot{x}, \dot{x}, x$），系を動かすパラメータ（k, c, m），外部から与える初条件（$\dot{x}(0), x(0)$）の説明が必要であろう。

③ ブロック・ダイヤグラムで表現するとどうなるか。

まずアナログ型コンピュータ用に使われるブロック・ダイヤグラムの考え方を説明し，CSMPとの関連を明らかにする。（図4-2,4-3参照）

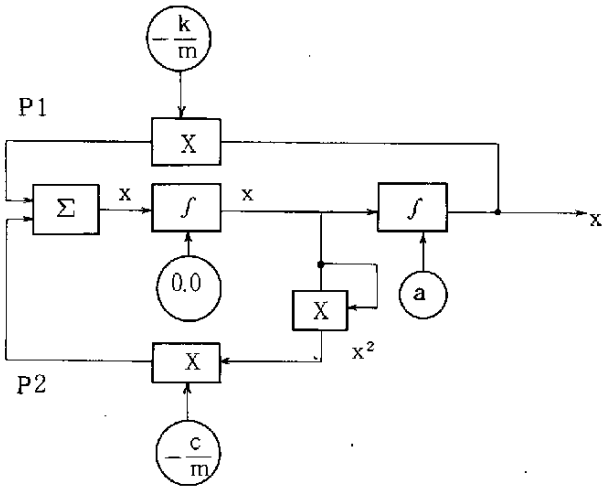


図4－2　ブロック・ダイヤグラム

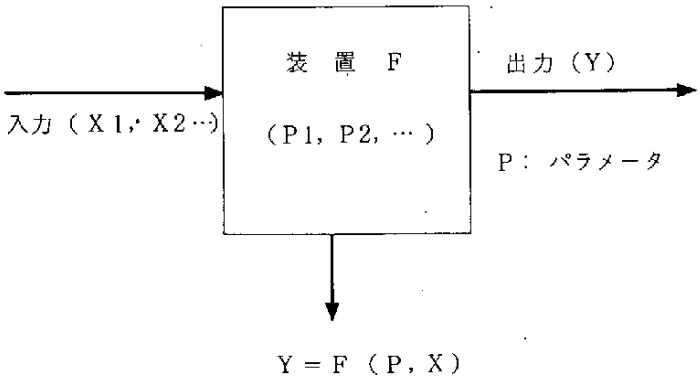


図4－3　CSMPの考え方

CSMPに用意されている，加算，乗算，積分などの関数ブロックを適当な順序で組み合わせることによりブロック・ダイヤグラムが作成されることを示す。

④ CSMPにより表現するとどうなるか。

ブロック・ダイヤグラムが作成出来れば，これをCSMPの言語に翻訳するのは非常に簡単な作業であり，ブロック単位にY＝F（P，X）を与えればよいことを説明する。（下記プログラム・リスト参照）

CSMPのプログラム・リスト

```
X2DOT  =  P1+P2
P1     =  X*(-K/M)
P2     =  XDOT**2*(-C/M)
XDOT   =  INTGRL (0,X2DOT)
X      =  INTGRL (A, XDOT).
```

CSMPによるモデル化に関する説明のあとで，CSMPによりシミュレーションを行なう場合の手順として，

① 系をモデル化する。
（ストラクチャ・ステートメント）

② モデルをシミュレートする際のパラメータやデータ値の決定（データ・ステートメント）

③ シミュレーションを制御する命令を与えるコントロール・ステートメントの説明を行う。

ストラクチャ・ステートメントの関数ブロックに関しては記述法（積分ならばY＝INTGRL（IC，X）），機能（$Y=\int_0^t X\,dt+IC$），等価なラプラス変数（1／S）等を説明する。

コントロール・ステートメントに関しては，トランスレーション・コントロール，実行コントロール，アウトプット・コントロールに分けて説明する。トランスレーション・コントロールのSORT，NOSORTに関しては，ソートされない区間でのブランクを伴うFORTRANステートメントの使用が可能であることを説明する。実行コントロール・ステートメントでは終了条件の指定法について説明する。

最後にCSMPの利点と欠点について説明する。

利点として，

- 技術者向きの問題向け言語
- 自動ソート機能を備えている
- FORTRANを併用出来る
- 多くの関数ブロックを備えたブロック中心言語である
- 繰返し計算やパラメータの最適化が簡単に行える。
- 計算時間と精度を考慮して積分法を選択出来る

欠点として，

- 計算時間がかかる
- 診断ルーチンが不備である

などをあげておけばよい。

4.5 言語の選択

構築したモデルを実際にシュレーションする場合，どのような言語を使ってモデルをプログラミン

グするかは重要な問題であることを述べる。

今日まで，GPSSとかSIMSCRIPTとかDYNAMOとかいったさまざまなシミュレーション言語が開発されているが，それぞれの言語は，それぞれプログラミングするのに得意なモデルの形態があり，どの言語が一番良いといったことは問題ではないといったことを強調しなければならない。

モデルの形態に適したシミュレーション言語を選ぶ必要がある。たとえば，待ち行列モデルを解析するためには，GPSSが適しており，このモデルをDYNAMOでプログラミングすることは適当ではないといったことを説明する。どの言語が，どのようなモデルに適しているかについてはすでに説明してあるので，ここでは，モデルを構築したあと，どのようなシミュレーション言語を選んだらよいかといったことが重要な問題であるということを強調すればよい。また，シミュレーション言語によらず，汎用の科学技術計算用言語，たとえばFORTRAN，などによってプログラミングすることも，めんどうではあるが，必要な場合もあるといったことを述べる必要もある。たとえば，モデルにおいて，意思決定機構が非常に複雑，たとえば，最適計算を十分に行なおうとする場合には，どうしても，その部分はFORTRANなどで行なわなければならなくなることがあるといったことを説明する。

指導上の留意点

各言語に関する詳細な説明はマニュアルや参考書類が豊富であるのでそれらに譲り，ここでは，よく知られた4種類のシミュレーション言語の重要な性能に関して，種々の可能性と特性の概要を明確にし，利用可能な技法を深く検討し得る能力を養うという方向で説明する。したがって，シミュレーション言語の多種な個々の命令を詳しく説明することは避け，共通概念でグループ化し，そのなかでも重要と思われるもののみを説明すべきである。そのグループとしてつぎの事項に関して説明するのも一つの方法である。

① モデル化の概念と表現法

② 状態の記述

要素の表現，分布の与え方，など

③ 状態変化の記述

時間軸制御メカニズム，グループ操作，など

④ 実行時の自由度

初期値の再設定，パラメータ変更，シミュレーション結果の解析と次のランへの反映，など

⑤ 特　性

計算時間，記憶容量，適用分野，など

第5章　実　　習

目　標

モデルを作成する能力を養うことは極めて困難なことであり，多分に直感的な要素が強く，モデル作成の一群の規則を作っても，最善の場合でも非常に限定された有効性しかなく，最悪の場合には必要な直感の開発を妨害することにもなりかねないので，ここではシミュレーション実施手順に基づき実際にモデルを作成する過程を通して，第4章までに述べた事項の実質的な意味を理解させるとともに，問題に対する有用なモデルを得る鍵を発見する能力を養うこと，モデルを表現する能力，運用する能力を養うことを目標とする。

内　容

インダストリアル・ダイナミックスの考え方により　モデルを構築し，DYNAMOを使ってシミュレーションを実施する手順は，すでに述べてあるので，ここでは，インダストリアル・ダイナミックスによる独特なモデルの作り方やDYNAMOによってコーディングするときの注意点などに重点を置いて説明したほうがよい。このため，例題をあげて，その問題を解析するために実際にモデルを作り，DYNAMOをコーディングするといったことを実施したほうがよい。では，どのような問題をとりあげて演習を進めたらよいかというと，やはり，企業の経営における意思決定問題をとり扱ったほうがよい。たとえば生産活動，在庫量，作業者の雇用に関係した問題を分析する例題や，研究開発に関するマネジメントの効率向上のポイントを発見するシステム分析の例題や，新製品の発売に際して，どのようなタイミングで，どのような広告を出したらよいかを分析する例題などをあげるとよいのではないか。問題の設定の段階では，特別な制約はないので，一般的な問題設定の方法を解説すればよい。ただ，とくに対象とする問題の境界を明確に決めなければならないことについて注意する必要がある。ここでは，一応問題の設定が行われた例題について，どのようにモデルを作っていったらよいかについて重点を置いて演習を進めていく。そこで一応問題の設定がすんだ例題として，つぎのような問題を例示してみる。

(1) 問題の設定

家庭用電気機械器具の生産工場で，生産している製品の種類は1種類とする。生産工場における生産は見込み生産であり，顧客の注文は，最終的には生産水準に反映されるけれども，短期的にみれば，注文を受けてから生産するのではない。そして，生産量は，直接的に現在使用している労働者数によってのみ決まり，労働者の数は，マネジメントの意思決定により，弾力的に増減できるものと仮定する。

このとき，顧客からの注文が変動すると，顧客への製品引渡しも変動して，毎日の生産引渡量の工場の生産台数が一致せず，倉庫の製品在庫量が変動する。顧客からの注文の変動に対して，在庫量をのぞましい水準に保つためには，工場の生産調整をどのようにしたらよいかがマネジメントの問題である。この問題を解析するために実際にモデルを構築し，シミュレーションを実施する演習を行なってみる。

(2) モデルの構築

この生産工場のなかで，生産活動が停止しているときもある値が存在する変数と，値が存在しない変数とがとり出させる。たとえば，前者の変数としては，在庫量とか労働者数とか仕掛品などであり，これからは，レベル変数であるということを理解させなければならない。また，後者の変数は，1日あたりの受注量，あるいは，製品引渡し量などであり，これらは，レイト変数であることを理解させなければならない。このとき，レベルとレイトの性格をあらためて説明し，さらに，レベルとレイトの関係を説明し，意思決定機構の働きを理解させる。すなわち，レベルに流れ込むフロー・レイトは，意思決定機構の操作によって調節されるということで，意思決定機構とは，パイプの中を流れる水の流量を調節するバルブのようなものであるということを説明するとよい。

つぎに生産工場において，物の流れ，注文の流れ，金の流れ，労働者の流れ，機械設備の流れ，情報の流れがどのように結びついているかを図に書かせてみる。最初は図5－1のようなもっとも簡単な流れを，インダストリアル・ダイナミックスで決められた記号で記述させればよい。

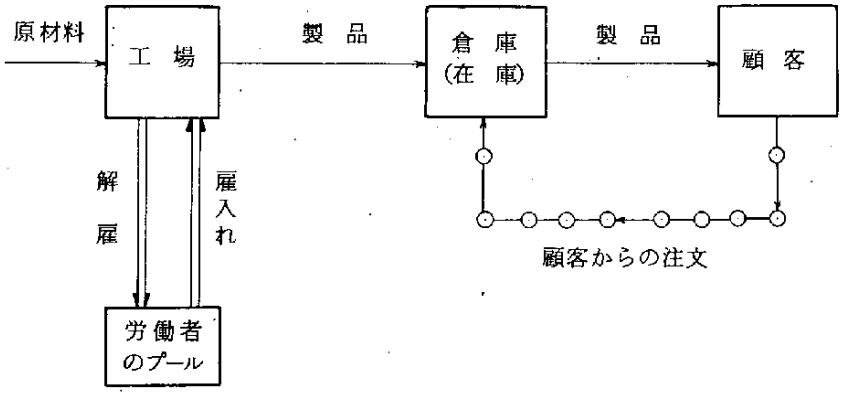


図5－1　電気機械器具の生産工場における
フローのブロック線図

図5－1のブロック線図をもとに，レベル変数間の因果関係を抽出させる。たとえば，仕掛品の量と在庫量の関係や労働者数と仕掛品の量の関係などを考察させる。このとき，レベル変数間の関係が正の関係，負の関係のいずれであるかを考察させる必要がある。レベル変数間の因果関係ループが規定されたら因果関係ループ間の関連についても考えさせ，モデル全体としてループが正になっているか負になっているかを調べさせる。

つぎに各レベル変数を形成するフローはどのようなものがあるかを抽出させる。すなわち，レベル変数がどのようなレイト変数で構成されているかを決めさせるのである。たとえば，在庫量（レベル変数）は，生産の完了した製品の量（レイト変数）と製品引渡し量（レイト変数）に関していることなどを例にあげるとよい。このとき，時間の経過による各変数の変化について十分注意する必要があ

る。そして，インダストリアル・ダイナミックスのモデルにおいては，時間の流れを図5－2のように表わすということを説明する。

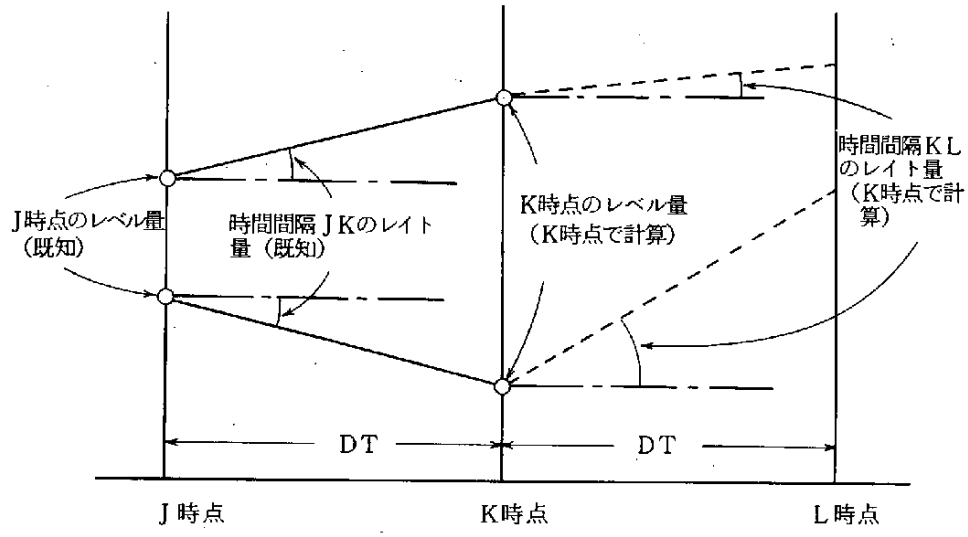


図5－2　時間の流れ

抽出したレイト変数のなかから，レベル変数をコントロールするレイト変数をみつけさせる。そして，レベル変数間の因果関係を考慮しながら，レベル変数をコントロールするレイト変数の値がどのような値によって，どのようなステップで決まってくるのかを決めさせる。このとき，補助変数について説明するとともに，このステップのなかに，生産工場を管理していくうえでの重大な意思決定機構が含まれてくることに注意しなければならない。意思決定機構の存在が確認されたら，つぎに意思決定機構についての考察を加えさせる。この意思決定機構が，人間の判断を必要とする意思決定機構なのか，あるいは，モデル内の意思決定機構自体で処理できるものなのかを区別させる。もし，前者の意思決定機構であれば，どのようなポリシーで意思決定をするのかは人間が指示しなければならないことを強調する必要がある。

レベル変数の因果関係ループやレイト変数の値を決めるアルゴリズムや意思決定機構などが決ったら，つぎには，モデルの各パラメータを定義しモデル内の時間をどのように進めるかを考えなければならないことを説明する。パラメータについては，定数なのかあるいは時間とともに変化する値で，テーブル関数などのような時間の関数としておかなければならないかをチェックさせ，実際にテーブル関数などを作らせる。

インダストリアル・ダイナミックスのモデルにおいても重要な要素である時間の進め方については，解析の対象としている現実のシステムに合致させて進めるようにしなければならないことを強調する。しかし，モデルでは，現実のシステムと異なり，1単位の時間（DT）の間は，モデルの状態は動かないということをとくに注意しなければならない。このため，単位時間間隔（DT）は，現実のシス

テムが停止してもよいと判断できる時間間隔で決めなければならないことを強調する必要がある。そこで，レベル変数間あるいは，レベル変数とレイト変数の間では，どれだけの時間遅れがあるかを調べさせる。たとえば，労働者を雇い入れるとき，雇い入れることの意思決定をしてから，実際に労働者数になるまでには，1次指数遅れになるということを説明するとよい。

このようにして出来上ったモデルを，インダストリアル・ダイナミックスで決められた記号，すなわち，


：レベル変数


：レイト変数


：補助変数

などを使ってフロー・ダイアグラムを書かせる。各自がそれぞれフロー・ダイアグラムを書き終ったら，図5－3のような標準的なフロー・ダイアグラムを示すとよい。

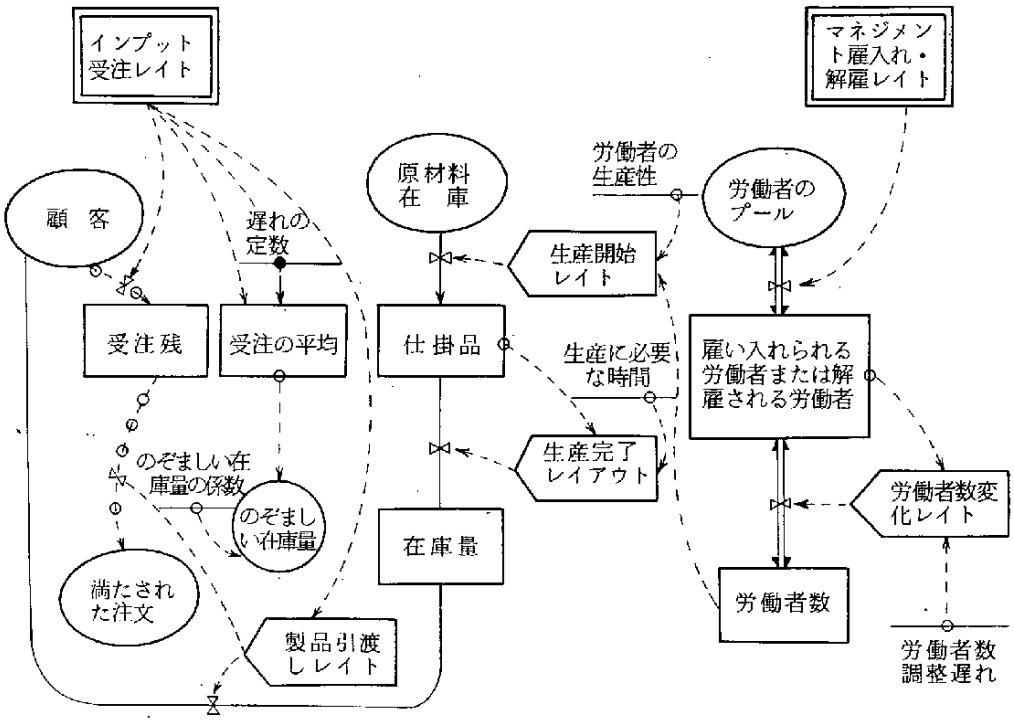


図5－3　電気機械器具の生産工場におけるフロー・ダイアグラム

(3) システム方程式の作成

図5－3のフロー・ダイアグラムにもとづいてシステム方程式を作っていかなければならないが，このとき，まず，システム方程式では時間の表示をどのようにするのかについて説明する必要がある。たとえば，在庫量に関するシステム方程式をとりあげてみると，

ＩＡ　：在庫量〔個〕（レベル変数）

ＰＦＲ：生産が完了した製品の量〔個／日〕（レイト変数）

ＳＳＲ：製品引き渡し量〔個／日〕（レイト変数）

ＩＡ．Ｋ＝ＩＡ．Ｊ＋（ＤＴ）（ＰＦＲ．ＪＫ－ＳＳＲ．ＪＫ）

のようになる。この式は，現時点（Ｋ時点）の在庫量が，前時点（Ｊ時点）の在庫量に，前時点から現時点の間（ＪＫ時間）で，在庫として入庫してきた製品の量（ＰＦＲ）と出庫していった製品の量（ＳＳＰ）の差を加えた値であることを示していることを説明する。この例にしたがって，一つ一つシステム方程式を作らせる。モデルをＤＹＮＡＭＯでコーディングするとき，このシステム方程式がそのまま利用できるので，システム方程式の表現は，ＤＹＮＡＭＯをコーディングする規則に従って書かせたほうがよい。また，シミュレーションを効率よく実施するため，モデルの初期値や定数を一つの表にまとめておいたほうが便利であるということを注意しておく。システム方式が出来上ったら，一応手計算でモデルを確認させる。この手順を落してしまうと，ＤＹＮＡＭＯでコーディングしたあとのデバグの段階で多大な労働を費やしてしまう危険があるということを注意する。

(4) ＤＹＮＡＭＯによるコーディング

システム方程式をＤＹＮＡＭＯでコーデイングするときには，まず各コンピュータ・メーカが発行しているマニュアルにそって説明すればよい。ＤＹＮＡＭＯの文法についてはメーカによる差異はないので，適当なメーカのマニュアルを使って授業を進めればよい。また現在，ＤＹＮＡＭＯⅡというレベルの高い言語が開発され，一部のメーカのコンピュータで使用することができるようになっているが，ＤＹＮＡＭＯの文法を知っていれば，ＤＹＮＡＭＯⅡの文法も理解できるので，ここではＤＹＮＡＭＯの文法にそってコーディングを進める。簡単なコーディング例を説明して，あとは各自にコーディングさせるとよい。ＤＹＮＡＭＯでコーディングするときは，とくに，式のうえでの時間の表示に注意しなければならないことをとくに説明しておかなければならない。

(5) プログラムの実行および結果の解析

コーディングが終ったプログラムは実際にコンピュータにかけさせ，各自にデバグさせる。シミュレーションの結果がでたら，結果の検証，解析を行わせ，もし必要ならばモデルの変更をさせる。さらに，意思決定機構の判断基準を数通り変えてシミュレーションをさせてみて，最適解を求めさせ，あらためて，シミュレーションの有効性について理解させる。

指意上の留意点

シミュレーションを実施するにあたり，もっとも重要なステップは，モデル作成のステップである。モデルのよし悪しがシミュレーション研究の成功失敗につながる。このため，モデル作りの演習にもっとも時間を費やす必要がある。ところが，モデル作りにはこれといった特定なルール，理論が確立され

ていないので，実際の問題を与えて，経験的にモデル作成の能力をつけてやらなければならない。この実習では，インダストリアル・ダイナミックスによるモデル作りを説明しているが，研修生がどのようなシミュレーションに興味をもっているかによって例題をかえればよい。研修生がシステムのダイナミックな特性の解析に興味のある場合は，DYNAMOやCSMPなどによってモデルの演習をやり，待ち行列のような事象中心の問題に興味のある場合は，GPSSやSIMSCRIPTなどによってモデル作りをさせるとよい。

参考文献

(1) J.H. Mize／J.G. Cox 共著 小笠原 暁 他共訳，「シミュレーションの基礎（Essentials of Simulation)」 45年4月第2版，培風館，¥1,100

本著は，ここで述べたすべての章にわたって使用できる内容を網羅しており，研修生にテキストとして与えるには絶好の教材である。

(2) R.C. Meier et al.，「Simulation in Business and Economics」，Prentice Hall 1969

実務家が企業における諸問題をシミュレーションを通して解析する立場から編集されており，第2章，第3章の授業などに並用するとよい。

(3) Thomas H. Naylor et al.，John 「Computer Simulation Techniques」 Wiley & Sons

第2章の内容の補足のために参考にするとよい。

(4) Dowson 他共著 清水幾太郎 編訳，「社会科学におけるシミュレーション」，日本評論社

(5) B.A. Behnkob 永村純一訳，「シミュレーションとモデル化の理論」 学献社，¥3,000

(6) 野本 明／寺野寿郎 編，「システム工学の手法（システム工学講座2）」，日刊工業新聞社

(4)，(5)，(6)は，第1章の内容を補足するために使用するとよい。

(7) 渡辺一司著，「インダストリアル・ダイナミックスの基礎理論」，東洋経済新報社，¥1,800

(8) 中西俊男著，「システム・シミュレータ」，産業図書 ¥1,600

(9) 内海武士著，「CSMPのSocial Systems Simulation への適用」，数理科学1972，12 ～ 1973，3

(7)，(8)，(9)はDYNAMO，GPSS，SIMSCRIPT，CSMPの各種言語の説明を補足するために第4章で用いるとよい。

(10) Henry Kleine，「A Survey of user's view of discrete simulation languages」，Simulation 1970，May

(11) 松浦卓文著，「シミュレーション言語の比較とモデル」，ソフトウェア科学 1971，8

10)，11) は第4章の言語の選択に関連して参考にするとよい。

(12) William T. Morris，「On the Art of Modeling」，Management Science Vol. 13，No12

(13)　坂倉省吾著，「インダストリアル・ダイナミックスの応用」，東洋経済新報社，¥ 1,600

(12)，(13)は第5章の実習において参考にするとよい。とくに(12)は，モデル化能力の向上を考えるうえで参考となろう。

コントロール・システム

目　　次

第1章　制御工学の基礎 …… 98

1.1　自動制御システム …… 98

1.2　自動制御システムの分類 …… 99

1.3　ブロック線図と伝達関数の概念 …… 100

1.4　ラプラス変換とその応用 …… 100

1.5　基本的な伝達要素の種類と特性 …… 101

1.6　周波数領域での解析 …… 101

1.7　システムの応答特性 …… 101

1.8　システムの安定性 …… 102

1.9　状態空間におけるコントロール・システムの解析 …… 102

第2章　情報の収集，変換，伝送 …… 104

2.1　計測器からの信号 …… 104

2.2　信号の伝送 …… 105

2.3　信号変換と標準化 …… 105

2.4　マルチプレクサ（入力走査装置） …… 106

2.5　A／D変換 …… 106

2.6　アナログ出力装置 …… 107

2.7　ディジタル入力装置 …… 107

第3章　コントロール・システムのモデル化 …… 108

3.1　モデル化の目的とモデルの選択 …… 108

3.2　モデルの形式 …… 108

3.3　プロセス同定とパラメータ推定 …… 109

第4章　最適化と適応化 …… 110

4.1　プロセス外乱 …… 110

4.2　定常状態における最適化 …… 111

4.3　動的最適化 …… 112

第5章　コントロール・システムにおけるコンピュータの機能と構成 …… 114
5.1　コンピュータの有する基本的システム機能 …… 114
5.2　基本的なシステム機能の複合 …… 115
5.3　コンピュータ・コントロール・システムの構成 …… 115
5.4　制御用コンピュータの基本構成 …… 116
5.5　多機構成の目的と種類 …… 116
第6章　制御用コンピュータのソフトウェア …… 118
6.1　オペレーティング・システムの概要 …… 118
6.2　コア・エリアの管理（コア・システムとコア・マッピング） …… 119
6.3　プログラム実行の制御と割込み処理 …… 120
6.4　制御用コンピュータのプログラミング …… 121
第7章　コントロール・システムの実例 …… 123
7.1　コントロール・システムの概要 …… 123
7.2　コントロール・システムの目的 …… 124
7.3　コントロール・システムの構成 …… 125
7.4　データの収集，伝送，変換 …… 125
7.5　コントロール・システムにおけるソフトウェア …… 126
7.6　制御機構とコンピュータの処理内容 …… 127
7.7　コントロール・システムにおける人間の役割 …… 127
第8章　シーケンス制御 …… 129
8.1　シーケンス制御の概要 …… 129
8.2　シーケンス制御システムの基本構成要素 …… 130
8.3　シーケンス制御システムのソフトウェア …… 131
8.4　コンピュータによるシーケンス制御 …… 131
8.5　シーケンス制御の応用例 …… 132
8.6　数値制御機械の概要 …… 132

科目「コントロール・システム」

教育の目標

この科目では，コンピュータを使って生産工程や工作機械などをコントロールするための基本的な技術を理解させ，コントロール・システムを構築できる能力を身につけさせる。

このため，コントロール・システムを構築し，運用するにあたり，必要な技術，たとえば，情報の収集，変換，伝送技術などを重点的に教え一般の制御理論については簡単に説明して概略を理解させる。

また，プロセス制御用コンピュータ，ソフトウェアなどを説明し，コントロール・システムにおけるコンピュータの機能と構成を理解させる。

時間配分

章	履習時間（時間）		
	講　議	演　習	実　習
1.　制御工学の基礎	10		
2.　情報の収集・変換・伝送	5		
3.　コントロール・システムのモデル化	3		
4.　最適化と適応化	6		
5.　コントロール・システムにおけるコンピュータの機能と構成	9		
6.　制御用コンピュータのソフトウェア	8		
7.　コントロール・システムの実例	10		
8.　コントロール・システムにおけるソフトウェア	9		
合　計	60		

第１章　制御工学の基礎

用　語　この章では次の用語を教える。

オートメーション，フィードバック制御，フィードバック，シーケンス制御，目標値，基準入力，制御偏差，定値制御，追値制御，プログラム制御，追従制御，比率制御，レギュレータ，サーボ機構，プロセス制御，ＰＩＤ調節計，直接ディジタル制御（ＤＤＣ），連続制御システム，不連続制御システム，ブロック線図，伝達関数，操作量，シグナルフロー図，ラプラス変換，インディシャル応答インパルス応答（重み関数），比例ゲイン，１次おくれ，２次おくれ，周波数伝達関数，ボード線図，定常応答，過渡応答，安定性，特性方程式，状態空間，状態変数，推移行列，多変数制御，無干渉制御，モード制御

目　標

古典的な制御理論から現代制御理論の初歩までを概説することによって，制御工学の理論面を紹介することを目的とする。コンピュータは主としてプロセス・システムに利用されていることが多いので，プロセス制御の基本となる制御理論の説明を中心として，コントロール・システムの基本的事項を理解させる。特に制御工学の基本的用語のもつ意味を十分に理解させることが肝要であり，周波数領域や時間領域におけるシンセシス問題の技術的な詳細は省略する。またコンピュータ制御は既存のＰＩＤ調節計の上位レベルに位置することが多いので，これらの工業計器の特性を理解させる。更に最適化制御や多変数制御では，状態空間における時間領域での応答を解析するアプローチがとられるので，現代制御理論の初歩の理解は不可欠である。

内　容

1.1　自動制御システム

自動制御システムの動作を単純なフィードバック・システムを例にとって説明し，併せて制御工学で用いられる最も基本的な用語の意味を解説する。具体的には以下の項目について順を追って説明する。

(a) **オートメーション**という言葉で代表される自動制御システムの組み込まれたプロセスを，化学工学石油精製，機械工業等にそれぞれ例をとって自動化の必要性，自動化によって期待される利得を概説する。これには実際のプラントにおけるコントロール・システムの動作を説明するスライド等を用いると効果的である。

(b) オートメーションを実現するにあたって用いられる自動制御のしくみを，フィードバック自動制御とシーケンス自動制御に分け，両者の基本的な相違点を説明する。これには，両者の制御方式の最も

基本的でかつ単純な例を用いて，まず相違点の大要を理解させ，つぎに，より厳密な定義（たとえばＪＩＳによる定義）を研修生に与えて，上記の理解を確認させる。フィードバック制御とシーケンス制御のＪＩＳの定義はつぎの通りである。

フィードバック制御「**フィードバック**によって制御量の値を目標値と比較し，それらを一致させるように訂正動作を行なう制御」

シーケンス制御「あらかじめ定められた順序にしたがって制御の各段階を進めてゆく制御」

コンピュータがどのような形で自動制御システムに利用されているかという点から考えると，シーケンス制御システムはフィードバック制御システムと同等あるいはそれ以上に重要である。しかしながら制御工学理論の立場からこの両者を眺めると，同等に取り扱いにくい点が多いので，シーケンス制御システムについての詳細は別に第８章で説明する。

従来，フィードバックを有さないコントロール・システムは自動制御システムではない（たとえば電気学会・日本機械学会合同の自動制御標準用語（案）昭和29年）として，シーケンス制御システムを自動制御システムから除外していたが，現在ではフィードバックが必ずしも必要条件ではないように修正されている。またシーケンス制御システムの多くは，あらかじめ定められた動作の実行の確認情報がフィードバックされて次の動作が行なわれるので，前記分類でのフィードバックは「**目標値**によってきまる**基準入力**と比較して**制御偏差**をつくるための後向き径路（フィードバック径路）の有無」という狭義の定義によることに注意を要する。

1.2 自動制御システムの分類

自動制御システムを①目標値の時間的性質，②応用形態，③制御動作の連続性によって分類して説明し，あわせてそれぞれに関連する用語の意味を理解させる。すなわち，

(1) 目標値の時間的性質による分類

① **定値制御** 目標値が時間的にある限られた期間では一定である制御で，自動電圧調整器などの例を説明するとよい。

② **追値制御** 目標値が時間的に変化する制御で，これをさらにつぎの３つの場合について例をあげて説明する。

- **プログラム制御** 目標値があらかじめ定められた変化をするとき（熱処理炉の温度制御など）
- **追従制御** 目標値が任意の時間的変化をする制御で，たとえば，ならい旋盤のサーボ機構などを説明するとよい。
- **比率制御** 目標値が他の量と一定の比率関係を保って変化する制御で，プロセス制御の一例（燃焼ガスの混合比の制御など）を説明する。

(2) 応用形態による分類

① 自動調整（**レギュレータ**）定値制御システムの応用形態による分類であることを説明する。

② **サーボ機構**（サーボメカニズム，物体の位置，方位，姿勢などを制御量とする制御で，応用例（航空機の自動操縦，自動平衡形工業計器など）とその特徴（制御量が機械的変位，入出力間の前向きのゲイン大，遠隔制御が多い，など）を説明する。

③ **プロセス制御**工業プロセスで温度，流量，圧力，レベル，濃度など主として連続量を制御量とするもので，定値制御が多いが，比率制御やプログラム制御の形をとることも多く，またこれを実現するためのＰＩＤ調節計について説明する。

ＰＩＤ調節計は，コンピュータ制御システムの直接制御レベルで**直接ディジタル制御（ＤＤＣ）**でない限り不可欠のものであるので，その重要性を強調する必要がある。たゞしＰＩＤ動作の説明は予備知識を必要とするので理論的説明は1.5節で行なう。

(3) 制御動作の連続性による分類

① **連続制御システム**　制御動作が時間的に連続である制御システム

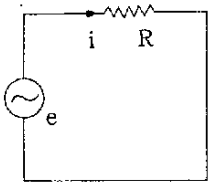
② **不連続制御システム**　制御動作が空間的に（たとえば不感帯 バックラッシュ，飽和，リレーなど）あるいは時間的に（サンプリング制御動作）不連続である制御システム

1.3 ブロック線図と伝達関数の概念

自動制御システムの設計や解析は**ブロック線図**を用いるのが普通であり，ブロックの入出力関係を表わす**伝達関数**の考え方が基本になる。ここでは伝達関数の意味，ブロック線図の作成法，ブロック線図の等価変換による簡略化の方法の概要を説明する。具体的にはつぎの事項について解説する。

(a) 単純なシステムを例に，伝達関数の概念を説明する。これには図１－１に示す抵抗器の回路程度のものでよい。

(b) 次に単純なフィードバック・システムを例に，いくつかの構成要素から成り立つシステムのブロック線図表現と基本的用語（目標値，基準入力，制御偏差，操作量，外乱，フィードバック信号，検出部，操作部，調節部など）との対応を示す。伝達関数は「入力・出力間の関係を表わす関数的表現」と定義されており，一般にはラプラス変換された領域で表現されているが，ラプラス変換の説明と同時におこなうと繁雑になる恐れがあるので，各構成要素が比例ゲインだけで表わされるような単純なシステム，たとえば直流発電機の定電圧制御（レギュレータ）などについて説明するとよい。



$$i = \frac{I}{R} e$$

図１－１　伝達関数の概念

(c) 複雑なブロック線図の等価変換による集約化（簡略化）の方法を説明する。これには等価変換の定則を説明した後，差し引き点，加え合せ点の移動，フィードバック・ループ結合などについて研修生に問題を与えて簡略化する演習問題を解かせる方法をとる。

(d) ブロック線図表現の利点と欠点および**シグナル・フロー図**の概略を説明する。特にブロック間に相互干渉がある場合，シグナル・フロー図の方が都合のよいことを解説する。

1.4 ラプラス変換とその応用

システムの時間領域が周波数領域における解析の基本となる**ラプラス変換**の基礎と伝達関数との関連について解説する。具体的にはつぎの事項を説明する。

(a) 単純なシステム（たとえばＲ－Ｌ回路）について，通常の時間領域の微分方程式をたて，ラプラス

変換およびラプラス逆変換による解法の概要を説明する。

(b) 単位ステップ関数（**インディシャル応答**）および**インパルス応答**（**重み関数**）の意味および伝達関数との関連について説明する。

(c) ラプラス変換の基本的性質および定理については次の事項を説明する。

- 線形性（加法性）
- 時間領域（t ー領域）での微分と積分
- Sー領域での微分と積分
- 推移定理（むだ時間と関連して）
- 結合定理（積分のたゝみ込み）
- 初期値定理，最終値定理
- 展開定理

1.5 基本的な伝達要素の種類と特性

つぎに列挙する基本的な伝達要素の実際例とその特性について説明する。

- 比例要素（**比例ゲイン**）　（カッコ内は関連説明事項）
- **1次おくれ**要素（時定数の定義と物理的意味）
- 1次すゝみ要素
- 積分要素（積分時間）
- 微分要素（雑音が混入した信号の微分の問題）
- **2次おくれ**要素（インディシャル応答についての基本的な用語（減衰率，応答時間など）の説明）
- むだ時間要素（伝達遅れ）

つぎにこれらの基本要素の組み合せ，とくにＰＩＤ要素およびＰＩＤ調節計について概要を説明する。

1.6 周波数領域での解析

周波数応答の意味と利用法，伝達関数との関連を基本的な伝達要素について説明し，あわせて以下の用語の意味を理解させる。

- **周波数伝達関数**（ラプラス変換領域での伝達関数との関連を中心に）
- **ボード線図**（ゲイン曲線，位相曲線，ゲイン余裕，位相余裕）
- 折点周波数，特性周波数
- 減衰係数，固有周波数，特性方程式（2次おくれ要素について）

1.7 システムの応答特性

与えられた入力に対してのシステムの応答特性を**定常応答**と**過渡応答**に分けて概要を説明する。

(a) 定常応答の特性

目標値に関する定常偏差の求め方（最終値定理の用い方）と定常位置偏差（オフセット），定常速

度偏差などの用語の意味を説明する。

(b) 過渡応答の特性

2次おくれシステムを例に，そのインディシャル応答に関連して減衰率固有振動数などを説明する。

1.8 システムの安定性

コントロール・システムの安定性は最も大切な必要条件であることを説明し，**安定性**を決定する要因として，とくに**特性方程式**の根の性質との関連を解説する。具体的な内容としてはつぎの通りである。

① コントロール・システムの特性方程式の意味をたとえば単純なサーボ機構について説明する。

② 安定判別法としては代数的方法，ナイキスト条件，根軌跡など各種の方法があるが，Routh - Hurwitz の方法のみの説明で十分であろう。

1.9 状態空間におけるコントロール・システムの解析

最近のやゝ進んだコンピュータ制御（とくに最適化制御）や多変数制御では1.8までに解説した古典制御理論では限界があり，またコンピュータ制御のアルゴリズムに適していない。ここでは**状態空間**における**状態変数**解析の初歩をつぎの事項について説明する。

① システムの状態方程式を簡単なシステムを例に説明し，状態空間，入力変数，出力変数，状態変数などの用語とその意味を説明する。

② 伝達関数あるいはブロック線図の形でシステムがあらかじめ記述されていることも多いので，これを状態方程式に変換する手法を説明する。

③ ひきつづき，**推移行列**（transition matrix）および状態推移方程式の解の求め方を簡単な2次システム程度を例に数値例をまじえて説明する。

④ コンピュータ制御ではシステムを時間的に離散化して表現する必要があることが多いので，離散的システムについての状態推移方程式とその解法の概略を説明する。

⑤ **多変数制御**については**無干渉制御**と**モード制御**概要を述べるに留める。

指導上の留意点

(1) ラプラス変換そのものについては，厳密には複素関数論についての知識を必要とするが，ここではラプラス変換表を参照しながら順変換，逆変換によって，一応，応答の解析が出来る程度を目標とする。

(2) 伝達関数との関連については，初期条件の扱い方に注意する。

(3) 周波数領域での設計（シンセシス）問題は，ゲイン余裕，位相余裕，ナイキスト線図にもとづいておこなわれるが，シンセシス問題について深入りする必要はない。

(4) 応答特性の改善の問題，すなわち，補償要素のシンセシス問題は，内容がやゝ技術的になりすぎること，コンピュータ制御では必ずしも中心課題ではないこと，等の理由からとくに触れる必要はない。

(5) 3次以上の高次のシステムについても，定常応答，過渡応答特性は勿論重要な問題であるが，とくに2次おくれシステムについて基本事項を十分理解させるように努める。

(6) 非線形システムを含めた一般システムの漸近安定性の問題（たとえばリアプノフ条件など）は，時間が許せばその概念を説明するとよい。ただしリアプノフ関数のつくり方そのものの説明をする必要はない。

(7) 状態変数解析は，統一理論を展開するのに便利ではあるが，古典制御理論のように制御対象や制御機構の物理的イメージとの対応が薄れやすいので，出来るだけ具体的なプロセスを例に説明するようにする。

(8) 離散的システムには，従来，Z変換が用いられているが，その基礎理論の解説にある程度の時間を必要とするので省略してもよい。

(9) 最適化システムの説明は便宜上，第4章で行なうが，状態変数表示が用いられることが多いので，1.9で用いる例題を最適化システムの説明にも共通に使えるようなものを選択するように努める。

(10) 現代制御理論におけるシステム解析は，まずシステムの可制御性と可観測性の検討からはじめられるが，これらの概念を数学的に説明するためには，かなりの予備知識が必要とされるので，単純な例で定性的に説明するとよい。

第２章　情報の収集，変換，伝送

用　語　この章では次の用語を教える。

原始データ，検出，変換，伝送，アナログ信号，ディジタル信号，雑音，電流信号，時分割多重伝送，フィルタ，マルチプレクサ，パーティライン方式，アナログ・ディジタル変換(A/D変換)，アナログ出力装置，ディジタル入力装置

目　標

プロセスの各所に発生する原始データを検出（測定），変換，伝送してコンピュータへ入出力する過程と方法について理解させる。

まずプロセスに発生する原始データの種類と方法について理解させ，電気信号に変換された情報を伝送する方法と，その際混入する雑音に対する補償法を述べる。つぎに電気信号の変換と標準化の方法およびマルチプレクシングについて解説する。また，コンピュータへ入力する際のインターフェイスであるプロセス入出力装置について，入力と出力の場合およびアナログ量とディジタル量の各組合せについて動作原理と利用法について理解させる。

内　容

2.1　計測器からの信号

制御対象から発生した**原始データ**をコンピュータへ入力するためには一般に図２－１に示すような信号の**検出，変換，伝送**が必要となる。ここでは，図２－１に基づいて以下の事項を説明する。

(a)　図２－１にもとづいて制御対象の物理量（圧力，流量，変位，速度，温度）性質，状態などが，どのように検出，変換，伝送され，コンピュータの入力チャネルに入力されるかを，特定の物理量（たとえば温度）を例に具体的に説明する。

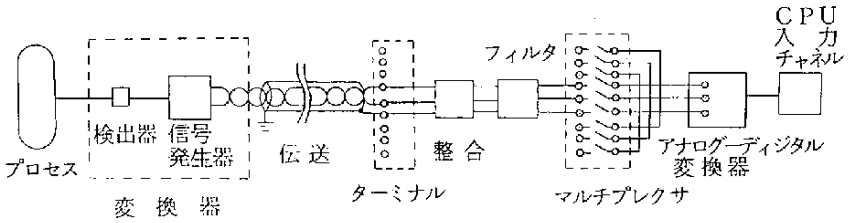


図２－１　制御用計算機入力へのプロセス計測信号

(b) 典型的な物理量（温度，流量，圧力，トルク，電力，速度，回転数など）を電気量（ミリボルト度の直流電圧，パルス列など）に変換するための代表的な変換器（トランスデューサ＝Transducer）を簡単に説明する。たゞし動作原理についての詳細な説明は必要としない。変換器の出力信号としては，低レベル（1ボルト以下）および高レベル（1ボルト以上）の**アナログ信号**および高レベルの接点信号（オンオフ信号），並列形式で表わされた2進ないし2進化10進の**ディジタル信号**など各種の場合があるが，とくに低レベルのアナログ信号をある一定距離以上伝送するには雑音の問題から増幅器を必要とすること，また時分割多重伝送方式で伝送された直列パルス信号を受信側で直列－並列変換する必要性などを中心に説明する。

(c) プロセス制御では，コンピュータ制御を実施する前に，既存の記録計器や指示計器がすでに用いられていることが多く，この場合には特別の変換器を用いなくても指示計器そのものに用いられている変換器にたとえばスライド・ワイヤを付加して制御用コンピュータの入力として用いられることが多いことを説明する。

2.2 信号の伝送

変換器の出力信号を伝送する方式についてアナログ信号とディジタル信号に分けて説明する。とくにアナログ信号の伝送については，伝送中の**雑音**の性質とその対策を中心に説明する。具体的には以下の事項を中心に解説する。

(a) 雑音の定義，雑音の種類（ピックアップ，コモンモード雑音，ノーマルモード雑音，伝導性雑音，静電的雑音，電磁的雑音など）とその除去対策。

(b) アナログ信号の伝送（搬送波によらない場合）は伝送線路へインピーダンスに影響されないように**電流信号**として送られることが多い理由。

(c) ディジタル信号の伝送は短距離の場合には高レベルの信号が用いられることが多いので誘導の問題はそれほどないこと。

(d) 比較的長距離に多数のディジタル情報を伝送するときに用いられる**時分割多重伝送方式**の原理およびこの方式を選択すべきかどうかの基準（入力点数や伝送距離との関連）。

2.3 信号変換と標準化

伝送された信号をA／D変換器のマルチプレクサへ入力するための信号の変換と標準化について次の事項につき説明する。

① 信号の変換

- 電流信号から電圧信号への変換
- 交流信号から直流信号への変換（整流回路とフィルタ）
- 標準入力信号範囲への変換
- 信号レベルの偏倚（バイアス）

② 信号源，伝送線路とのインピーダンス整合の問題

③ フィルタ回路

ＲＣ平衡フィルタの使用目的と応答時間の関連，ＲＬフィルタの使用，ディジタル信号中のインパルス雑音除去，ディジタル（数値）フィルタの必要性

2.4 マルチプレクサ（入力走査装置）

マルチプレクサの機能，種類，制御用コンピュータからの入力信号の選択の方式などにつき，つぎの事項を説明する。

① マルチプレクサの必要性とその機能

② 切換方式の種類（入力のサンプリング速度によって，リード・リレー，水銀リレー，トランジスタ・リレーなどが用いられるが，その選択の基準）

③ 入力アナログ信号の選択方式（シーケンシャルあるいはランダムな選択方式について）

最近は図２－２に示すような信号源を並列に並べてマルチプレクサを通してコンピュータに入力する方法に代って，図２－３に示すいわゆる**パーティライン方式**が大幅に採用されつつある。またミニコンピュータなどに最近多く見られるコモンバス方式についても適宜説明を追加することが望ましい。

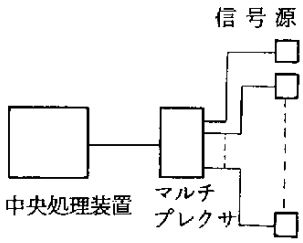


図２－２　マルチプレクシング方式

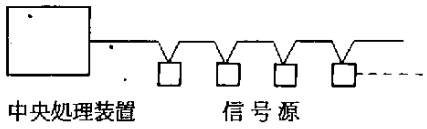


図２－３　パーティライン方式

2.5 Ａ／Ｄ変換

アナログ信号を量子化してディジタル信号に変換するための**アナログ・ディジタル変換器**（Ａ／Ｄ変換器）について次の事項を説明する。

① Ａ／Ｄ変換器の種類

いわゆる，積分型と逐次比較型について動作原理を簡単に説明する。

② 雑音の影響

コモンモード雑音およびランダム雑音に対してこの雑音除去率を上記の二つの方式について比較説明する。

③ 変換精度と速度

変換精度（８～12ビット）と速度の範囲を説明する。

2.6 アナログ出力装置

制御用コンピュータで処理された情報をプロセス，あるいはプロセスに直結する制御器へ伝送して閉ループを形成するとき，アナログ信号の形で情報が必要な場合にディジタル信号をアナログ信号あるいはパルス列に変換する必要があることを次の事項につき説明する。

① 出力信号の形式

- 梯形抵抗回路による電圧あるいは電流信号
- パルス列信号
- パルス幅信号

② 閉ループ制御におけるパルス列信号の利用

設定値に比例した数のパルス列を出力し，それによってステッピング・モータを駆動して位置制御やアナログ調節計に入力する例を説明する。

2.7 ディジタル入力装置

プロセスにおける各種の接点信号（スイッチの開閉など）や符号化されたディジタル信号（たとえばBCDコード）をどのようにして制御用コンピュータに入力するかに関してつぎの事項を説明する。

① ディジタル入力信号の種類

プロセスからのディジタル信号として，リミット・スイッチ，圧力スイッチ，しゃ断器の補助接点，コンソールからの10進スイッチ，カウンタ値などがあることを説明する。

② 接点入力制御回路

プロセスからの接点信号が信号調整回路を経てコアに読みこまれる様子を説明する。

③ 中央演算処理装置の負担を軽減するために**ディジタル入力装置**に備えられている機能として次の二つを説明する。

- 状態変化検出（チェンジ・ディテクション）
- パルス・カウンタ

指導上の留意点

(1) A／D変換には各種の方式が用いられており，またA／D変換器の性能（変換精度，変換速度）も改善されつつあるので，最新の状況を調査して説明する必要がある。

(2) プロセス入出力装置の最近の傾向として，信号のアナログ・ディジタルの区別によって別個の名称（たとえばディジタル入力装置）で呼ぶよりも，ディジタル入出力（2進入出力），表示出力，アログ入出力をすべておこなう多重入出力装置が用いられており，いわゆるプロセスI／O装置はれをさす。勿論内容的には，たとえばディジタル出力は16点を1グループ，表示出力は4点[illegible]プというようにグループ化されている。

(3) プロセスI／Oに遠隔点で発生した情報を入力するには150 ～ 1800 Baud 程度の非同[illegible]式，2400 Baud 程度の同期式の伝送路を通しておこなわれ，このために通信結合（制御）装[illegible]用いられているのでその概要を必要に応じて説明する。

第3章　コントロール・システムのモデル化

用　語　この章では次の用語を教える。

システム・モデル，論理モデル，計算処理モデル，推定モデル，予測モデル，プロセス同定，パラメータ推定

目　標

主としてプロセス・システムを対象に，モデリングの方法を習得させる。まずモデリングの目的を具体例を用いて説明し，合目的的なモデルを選択，作成できる能力を養なう。つぎに，モデルの形式を論理モデルと計算処理モデルの2種類に大別して，プロセス・システムにどのようなモデルがどのような目的で用いられているかをやや具体的に説明する。さらにモデルの構造およびパラメータを同定，推定する方法を理解させる。

内　容

3.1　モデル化の目的とモデルの選択

システムの制御を行なうにあたっては，制御対象の構造および機能を知る必要があり，特にコンピュータ制御を実施するためには定量的モデル，すなわち，数式モデルを必要とすることを説明する。とくに以下の項目について出来るだけ具体例にもとづいた解説が必要である。

① モデル化の目的がシステムの概略設計のためなのか，あるいは詳細な制御アルゴリズムの決定のためなのか。

② それによって，システムのどの範囲までをモデル化すべきか，すなわち，モデルの境界はどこまでか。

③ どの程度，詳細なモデルをつくるべきか。

④ モデル化に必要な情報は実際上，得られる可能性があるかどうか。

⑤ 変数のうち，独立変数は何であり，従属変数は何か。

⑥ 独立変数のうち，制御可能な変数（すなわち操作変数）は何であり，制御不能な変数（外乱，雑音）は何か，また制御不能な変数は何に基因するか。（プロセス制御を考えれば，原料のバラツキ，周囲条件の変化，装置自身の特性の変化，経済的要因など）

⑦ 拘束条件や上下限の制限はどのようなものがあるか。

3.2　モデルの形式

システム・モデルを線形性，確率的変動要素の有無，時間的推移に対する考慮などから分類して説明す

ることは一般に行なわれていることであるが，ここでは，モデルの目的（モデルの利用され方）から，とくにプロセス・システムを主体として次の事項を説明する。

(a) **論理モデル**

標準的なプロセスの運転手順や一般にシーケンス制御に用いられるモデルについて説明する。

(b) **計算処理モデル**

プロセス変数に関して演算処理を必要とするモデルで次の二つに分けて説明する。

① **推定モデル**

直接測定できない変数，あるいは観測できても観測値に外乱や雑音が含まれていて，測定値そのものを真値として取り扱うことのできない場合に用いられる推定モデルの概要を説明する。

② **予測モデル**

コントロール・システムの機能として予測機能が必要とされる理由を説明し，予測モデルの例を述べる。（たとえばあるプロセスの出口温度が流量変化によって予測されるとき，出口温度を測定することにより入口の流量制御を行なうことが出来る例などを説明するとよい。）またこの問題の数学的基礎として，各種のフィルタ（算術平均，移動平均，一次おくれフィルタ）について説明する。

3.3 プロセス同定とパラメータ推定

プロセス・システムでは入出力変数が多変数であること，非線形性，むだ時間，外乱などの存在によって，プロセスの構造を数学的に記述すること（すなわちプロセス方程式の作成）が困難であることが多いが，単純なプロセス・システムを例に，理論的分析（すなわちプロセスを支配している物理・化学の法則にもとづくもの），経験的分析によるプロセス方程式の立て方を解説する。つぎに一応決定されたモデルが，実システムをどの程度忠実に表現しているか，すなわちモデルの検証に関して，シミュレーション，あるいはプラントの実データによるチェックの方法とモデルの改善法について説明する。

パラメータの推定については，モデルを線形化して回帰分析をおこなう基礎的な理論を説明する。

指導上の留意点

(1) モデル化の過程や目的に合致したモデルの選択法にきまった法則といったものはなく，したがって教師がその専門分野を通して精通しているシステムのモデル化の例を説明して，その過程で問題点となったものを紹介するのがもっとも効果的であろう。

(2) 構造モデルの決定やパラメータ推定には回帰分析のほか，相互相関やスペクトル解析，さらには実験計画法にもとづいた高度な方法があるが，あまり深入りした説明の必要はない。

(3) モデル化の方法やモデルの種類については基礎編「システム概論」第2章，また統計的手法の応用については専門編「統計解析」との関連に留意する。

第4章　最適化と適応化

用　語　この章では次の用語を教える。

プロセス外乱，構造的外乱，パラメータ外乱，直接制御，最適化制御，適応制御，直接最適化，頂点保持制御，静的最適化，動的最適化，可制御性，可観測性

目　標

プロセスに対する各種の外乱に対する補償をおこなうことによってプロセスの運用を最適におこなうための最適化制御の概要を理解させる。

まずプロセスへの外乱を構造的外乱，パラメータ外乱，運用上の外乱，プロセス変数の外乱，雑音に変動周期の長短によって分類して説明し，各々にどのような制御が必要かを制御のレベルという観点から直接制御，最適化制御，適応制御について解説し，具体例を通して理解させる。

次に最適化の手法を時間断面における静的最適化と時間関数としての動的最適化の各手法について紹介し，さらに現代制御理論（状態空間解析）がどのように最適化制御と関連するかを理解させる。なお，最適化手法の具体的なアルゴリズムについては簡単にふれるに留め，各手法の利害得失と利用の限界を強調することが肝要である。

最後に適応制御の考え方の初歩を習得させる。

内　容

4.1　プロセス外乱

プロセスには各種の外乱が入り，そのためプロセスの運転状態が所望の状態からはずれて最適性が失なわれる。ここでは**プロセス外乱**の種類とその影響を抑えるための制御の種類（レベル）との関連をつぎの順序で説明する。

(1) プロセス外乱の種類

プロセスには種々の外乱が入るが，これを外乱発生の頻度（周波数）の点からつぎのように分類して，各外乱がプロセスのどの部分の状態を乱すかを周期の長い順に適当なプロセス・システムを例に説明する。

① **構造的外乱**

経済状態の変化（市場性，一般経済状態）

長期計画の変更

構成要素の変化（新しい要素の追加，一部の撤去）

など。

② **パラメータ外乱**

構成要素の劣化，特性の時間的変化，摩耗

原料の成分変化，バラツキ

など。

③ 運用上の外乱

運転点のバラツキ

負荷変動

設定値の変動

など。

④ プロセス変数の外乱

圧力変動

成分変動

温度変動

など。

⑤ 雑　音

測定器の雑音，データ変換，伝送に伴なう誤差

など。

(2) 各種外乱の影響と制御のレベル

(1)で列挙した各種の外乱に対する補償をおこなうため，プロセス制御では制御を次の三つのレベルに分けておこなっていることを説明する。

(a) **直接制御**

プロセス変数に対する外乱を補償して設定値に一致するようなコントロールでＰＩＤ調節器，DDCなどで行なわれていることを説明する。（第1章の定値制御，追従制御との関連に留意する。）

(b) **最適化制御**

プロセス・パラメータおよび制御機構が与えられたという条件でプロセス変数（制御のおこなわれる段階ではその設定値となる）の関数として与えられる評価関数を最適化（最大化，最小化）する制御で，ＤＤＣおよびディジタル・シーケンサを除く，コンピュータ制御の主な適用分野であることを説明する。（システムの有効さや評価関数の決定法については，基礎編「システム概論」第4章との関連に留意する。）

(c) **適応制御**

対象プロセスの入出力信号などから得られるプロセスの状態およびプロセスのおかれている環境の変化に対応して，制御アルゴリズムそのものを変更したり，評価関数をきりかえたり，そのパラメータを変更する機構であることを説明する。

4.2　定常状態における最適化

比較的周期の長い外乱に対しては，ある時間断面でのプロセスの静的な状態のみにもとづいて最適化

が実行できる場合が多くあること，また具体的な最適化手法として数式モデルを用いない**直接最適化**（いわゆる**頂点保持制御**）および数式モデルを用いた線形計画法，グラディエント法，さらに**静的最適化**の限界と**動的最適化**の必要となる場合について説明する。

具体的には次の事項を説明する。

① 外乱変数と操作変数をグラフの2軸にとってプロセスの評価関数（たとえば利益あるいは効率）が一般に等高線であらわされることを説明し，この等高線の頂点（最大利益，最高効率）を保持するためには，外乱に応じて操作変数を変更していく必要があることを理解させる。

② 化学プロセスなどでは評価関数の数式モデルの作成が困難なため，現在の運転点から出発して直接試行錯誤的に頂点に達するような方法が必要であり，その例としてＥＶＯＰなどを説明する。また操作変数の値を座標軸にとると，外乱によって頂点が移動すること，およびこの頂点につねにプロセス状態を追従させるための頂点保持制御の例を解説する。

③ 最適化に関与するプロセス変数が多数の場合，線形の数式モデルを用いて最適化をおこなう例として線形計画法の原理およびオンライン制御での実施例を説明する。

④ 数式モデルが非線形である場合の最適化手法の例として，グラディエント法および，いわゆるプローブ法（Probe Method）の原理を説明する。

⑤ プロセスの運用にはプロセス変数に関する上下限，あるいは等号，不等号であらわされる制約条件があるが，最適化にあたってこれらの制約条件を取り入れる方法として罰金法（ペナルティ法）の考え方およびその代表例としてＣＲＥＳＴ法を説明する。

⑥ 静的最適化では時間的な要素が含まれないが，過渡的なプロセスの変動が重要な要素になる場合をプロセスの応答時間との関連で説明する。

4.3 動的最適化

動的最適化は一般に，システムを状態間で表現し，制御の目的によって最小時間制御，積分形式の汎関数であらわされる場合などについて動的計画法あるいは最大原理で代表される変分学的手法が用いられるが，内容的に高度なものである割に，実施例は必ずしも多くない。したがって中級情報処理技術者としては下の項目に限定してよいであろう。

① 最適化問題の記述の形式として

$$\dot{x} = Ax + Bu + Cd$$

の標準形式の表現が用いられること。

② 評価関数として

$$T = \int_{t_1}^{t_2} p(x, u, d, t)\,dt$$

の汎関数表現が採用されること。

③ システム方程式が線形で評価関数が２次形式のとき，$U = -Kx$ という状態フィードバック

の形で最適制御がきまること。

④ 最適化手法として，最大原理，動的計画法が用いられること。

⑤ システムの**可制御性**，**可観測性**の意味と単純な例題による説明。

指導上の留意点

(1) 4.1の三つのレベルより更に上位の機能として，自己組織化機能が考えられるが，これはシステム（プラント）の設計の問題と考えられるので，とくに制御のレベルとして説明する必要はないであろう。

(2) コンピュータ制御の数学的理論としては，最大原理やダイナミック・プログラミングが詳細に紹介されていることが多いが，現状から考えると，その必要はないと考えられる。実際によく使われる手法として，デシジョン・テーブルを利用することがあるので，デシジョン・テーブルについて簡単に触れておくとよい。

(3) 数学的な厳密さは必要でないが，システムの可制御性，可観測性は現代制御理論の基本的な前提となるので，その概念は十分理解させるようにする。

第5章　コントロール・システムにおける コンピュータの機能と構成

目　標

コントロール・システムにおいてコンピュータのもつどのような機能がどのような形で利用されているか，また応答性が信頼性の面で制御用コンピュータが汎用コンピュータに対してどのような特徴をもっているか，また処理能率が信頼性向上のためにどのようなコンピュータ・システムの構成が用いられているかを理解させる。

まずコンピュータの有する基本的な機能（データ収集，変換，記憶，転送，計算，指示，表示）について説明し，次にこれらの基本的なシステム機能をどのように組合わせて，コントロール・システムに必要とされる機能（計測とデータ処理，最適化適応化機能，調節機能など）を達成するかを理解させる。

次に制御対象（プロセス）とコンピュータの間の情報の授受の方法，速度および人間の介在の有無によって，オフライン，インライン，オンラインの各方式が考えられることを説明する。

ひきつづいて制御用コンピュータの特徴的な機能として割り込みと各種のチャネルについて説明し，その必要性と利用方法の大要を把握させる。

さらに対象システムの規模の拡大，複雑化あるいはコントロール・システム全体の信頼性の向上のために，複数台のコンピュータを効率的に配置した多機構成の例を紹介し，実際例を通して各々の特徴を理解させる。

内　容

5.1　コンピュータの有する基本的なシステム

コントロール・システムにコンピュータのどのようなシステム機能が利用されているかを説明する。具体的には次の機能について，各機能が第1章で説明したコントロール・システムにどのように用いられるか，またその限界（経済的，技術的）は何かを説明する。

① データ収集

対象システムから原データを収集する。

② 変　　換

他のシステム機能に都合のよい形にデータの形式を変換する。

③ 記　　憶

データやプログラム（制御手順やデータの処理形式）を必要な期間，記憶する。

④ 転　　送

データやプログラムを必要とされる場所に転送する。

⑤ 計　　算

四則，微積分などの演算，大小比較などの論理判断を行なう。

⑥ 指　　示

対象システムに，制御量が目標値に合致するよう最終制御要素（直接制御要素）を通して指示する。

⑦ 表　　示

システムの運転員や管理者が理解されやすい形に情報を変換して指示する。

5.2 基本的なシステム機能の複合

コントロール・システムでは，5.1で列挙した基本的なシステム機能がいくつか組合わされて利用されていることを次の三つの場合に分類して説明する。

(a) 計測とデータ処理の機能

実例として，原データからの雑音除去（フィルタリング），予測，システム状態の推定などのコンピュータに入力されるまでの機能および表示（監視制御における警報など），記録（日報・月報作成，データロギング），効率計算などのオペレータや管理者に対する情報の提供について説明する。またシミュレーションもこの中に含まれる。

(b) 最適化と適応化の機能

直接最適化（試行錯誤法や頂点保持制御），数式モデルによる最適化（ＬＰ，グラディエント法，最大原理など）などの最適化機能とシステムの管理的な面からスケジューリング，配分，割当ての計算，さらに学習機能について説明する。

(c) 調節機能

ＰＩＤ制御やＤＤＣ計算，フィードホワード制御計算，多変数システムのモード制御，プログラム制御，シーケンス制御などの機能を説明する。

5.3 コンピュータ・コントロール・システムの構成

制御対象（プロセス）と制御用コンピュータとの間の情報伝送の方法および人間による情報収集，処理，制御対象に対する操作の有無によってコンピュータ・コントロール・システムの構成を分類して説明する。具体的には次の事項について解説する。

(a) オフライン方式

「オフライン」の定義をおこない，この方式では集積されたデータがある時間遅れをもってバッチ方式で処理されること，またこのような時間遅れが許容されるのはどのような場合かを説明する。

（たとえば設計のための予備的段階での検討，定期的に行なわれる生産報告，制御方式の検討など）

(b) インライン方式

「インライン」の定義をおこない，プロセスの状況を運転員が迅速にコンピュータに入力し，この情報に基づいてコンピュータが処理した内容を運転員に指示する例などを説明する。

(c) オンライン方式

プロセスで発生した情報を人間を介さずに直接コンピュータへ入力する方式で，コンピュータで処

理され，コンピュータから出力される情報がプロセスに直接与えられるか，あるいは人間を通してプロセスにフィードバックされる場合があることを説明する。

5.4 制御用コンピュータの基本構成

制御用コンピュータはリアルタイムで情報処理をする要求から汎用コンピュータとは異なったいくつかの特徴をもっていることを最も基本的な構成について説明する。具体的には次の項目のもつ内容の概要を解説する。

① 単純なプロセス制御を例に，プロセスからのデータの取り込み，割り込み，バルクメモリへのデータの転送，プロセスへの指示情報などを中心に情報の流れと経由するチャネルの関係を説明する。

② チャネルの概念と各チャネルの用途（相違点）などを次の項目につき説明する。

- マルチプレクサ・チャネル
- セレクタ・チャネル
- DMA（ダイレクト・メモリ・アクセス）チャネル
- 割込みチャネル
- 各チャネルの転送速度の範囲

5.5 多機構成の目的と種類

5.4で説明した単機システム（Stand－alone System）は多くのコントロール・システムで用いられているが，システム規模の拡大，信頼性の向上，処理能率の向上などのために多機構成のシステムが用いられる理由を次の多機構成について説明する。

(a) 直列システム

それぞれに割当てられた機能をもつ複数台のコンピュータが串形に結合されている構成でプロセス入出力やマンマシン・インターフェイスの処理を前段のコンピュータで処理する場合，制御用コンピュータが不得意とするファイル処理を後段のファイル処理用コンピュータに委ねる場合などを説明する。

(b) 並列冗長システム（Dual System）

完全な並列照合をおこなうシステムは経済的にも，また技術的にも問題点が多いが，一応，次の待機切換システムと対比する意味で概要を説明する。

(c) 待機切換システム（Duplex System）

待機切換システムはバックアップの程度によって各種の構成が考えられるが，電源部，I／Oチャネル，記憶装置，中央演算処理装置をすべて並列にする完全な形とI／Oチャネルを切換えて扱う方式（クロスコール方式）など代表例を説明する。特に待機切換システムでは切換器の信頼性設計の重要性を強調する必要がある。

(d) マルチ・プロセッサ・システム

制御用コンピュータ・システムとしては特に大規模なシステムにしか用いられていないが，メモリバス，I／Oバスを中心とした中央演算処理装置群，記憶装置群の幾何学的構成，およびこの構成によ

る利点（たとえば一部の故障時にフォールバック運転が可能であること）などを説明する。

(e) マルチ・コンピュータ・システム

複数台のコンピュータが入出力チャネルを経由して（共通母線で直接結合されていない）情報交換をしながら動作するシステム構成であること，またこの構成の処理能率，信頼性などから見た特徴を説明する。

(f) ハイアラーキ・システム

複数台のコンピュータからなるシステム構成を層別化の方法から，Multi－Strata，Multi-Layer，Multi－Echelon の三つに分類してその考え方の概要と実際例を説明する。

指導上の留意点

(1) 5.2での説明は，各機能の分類に含まれる項目の内容の詳細な説明ではなく，5.1で列挙したコンピュータの基本的なシステム機能がどのように組合わされているかを理解させることにある。

(2) 5.5の（a）から（f）までの多機構成システムの分類は，システムを機能，幾何学的構成，層別化の方法など，異なった面から分類したものであるから各分類はオーバーラップする。

第6章　制御用コンピュータのソフトウェア

用　語　この章では次の用語を教える。

ワーキング・エリア，優先順位，パーマネント・コア・エリア，コア・システム，コア・マッピング，システム・オーバーロード，固定コア・システム，浮動コア・システム，区画コア・システム，プロセス割込み，システム割込み，クロック割込み

目　標

制御用コンピュータのソフトウェアが一般の事務・科学用のものとどのように異なっているかを理解させるのが，この章の主要な目標である。まず制御用コンピュータに要求される機能を述べ，この要求を満足するようなオペレーティング・システムの必要性を理解させる。次にオペレーティング・システムの主要部分の構成と考え方を比喩的に説明する。ひきつづいて，制御用コンピュータの限られたコアで多数の機能プログラムを処理するための方式（コア・システム）の概要を理解させる。またプログラムの実行の制御が割込みによってなされることとプロセス割込み，システム割込みおよびクロック割込みについて説明する。最後にコントロール・システムのプログラミングの基本的なパターンとその他の考慮すべき要素が何であるかを理解させる。

6.1　オペレーティング・システムの概要

(1) オペレーティング・システムの特徴

制御用コンピュータは，事務・科学計算用の一般のコンピュータと異なり，本質的にオンライン・リアルタイム制御を目的としており，そのソフトウェア，とくにオペレーティング・システムは次のような特徴をもっていることを説明する。

(a) コアに常駐するプログラムを出来るだけ少なくして，作業エリア（**ワーキング・エリア**）を出来るだけ多くとるように留意する。とくにシステム・コンポーネントとして制御用コンピュータが用いられるときは，コンピュータの価格を出来るだけ安価にすることが必要とされ，したがって限られたコアをいかに利用するかについての工夫が必要である。

(b) プロセスに発生する事象におくれなしに対応していくために割込み処理（Interrupt）が大きな要素になる。

(c) 入出力機器やチャネルを効率的に動作させるように制御する必要がある。

(d) 多重処理制御をおこなうときのタスクの**優先順位**（Priority）の処理を効率よく行なう必要がある。

(e) システム・タイマを設けて，定まった時刻，あるいは時間間隔でなすべき仕事を管理して処理する必要がある。

(2) オペレーティング・システムの基本的な考え方

以上のような必要性を満足するために，オペレーティング・システムが用いられているが，つぎにオペレーティング・システムの基本的な考え方を次の順序で説明する。

(a) （制御用コンピュータに限らず）コンピュータによる処理は基本的には直列的であること，すなわち，同一時点で複数の仕事を並行して処理することは出来ないこと。

(b) したがって，ある仕事を処理しているとき，より優先度（あるいは緊急度）の高い仕事の処理が必要とされたときは，現在の処理を中断して，優先度の高い仕事の処理を開始しなければならないこと。

(c) このためには，現在の処理内容や中間結果を一時的に適当な場所に待避させる必要があること。またコアの大きさは限られているので，多くの場合，プログラムはコア以外のメモリ（バルク・メモリ）に格納されており要請があった時に容易にかつ迅速にコアにロードされる必要があること。（これに関連した事項，たとえばリロケータブル・プログラム，コア・システム等の説明は後にまわす。）

(d) コアを効率的に使用するため，用途に応じてコア・メモリをワーキング・コア・エリアと**パーマネント・コア・エリア**に分け，さらに後者は三つのエリアに分けられていることを説明する。

具体的には次の事項を説明する。

- ワーキング・コア……バルク・メモリにあるファンクション・プログラムによって共同利用されるコアのエリア
- コモン・データ・ストレージ・エリア……プログラム間で共用され，しかも迅速に使用できる必要のあるデータを格納するエリア
- モニタの主要部分……モニタの主要部分，例えばエグゼクティブ・コントロール・プログラム（Executive Control Program = ECP）などを常駐させておくエリア。ECPについてはまだ説明していないので，ここでは，オペレーティング・システムの中枢部分ということで理解させておけばよい。
- モニタ中のファンクション・プログラム……使用ひん度の高いサブルーチンやサブプログラム・システムを常駐させておく。
- 前述のコアエリアの管理と同時に，時間的な管理が必要であることを説明する。これには単純なリアルタイム・コントロール・システムを例に，ある割込みが発生したときの割込み処理の内容をとくにタイミングという点に注目して説明する。

6.2 コア・エリアの管理（コア・システムとコア・マッピング）

制御用コンピュータでは限られたコア容量を有効に用いて処理を行なわなければならず，また仕事の内容も一般のシステムでは20～200程度の機能プログラムを必要に応じて処理する必要がある。このため，オペレーティング・システムの一つの大きな目的は大部分の機能プログラムが格納されているバルク・メモリを如何にしてコアにロードして処理するかであることをまず説明し，つぎに各種の**コア・システム**について解説する。

(1) コア・システム

(a) 固定コア・システム

各機能プログラムにはきめられたコア・エリアが割りあてられており，リロケーションされることのない方式（通常コア・イメージ方式と呼ばれる）で，小規模なコントロール・システムに用いられること。

(b) 浮動コア・システム

一部の機能プログラムやモニタを除き，すべてのプログラムは必要に応じてバルク・メモリから呼び出されてつねに異なったコア・エリアに転送されて実行される方式をとっている。プログラムがリロケータブルであり，また空いているコア・エリアがどこにあるかをつねに管理する機能が必要となり，一般に用いられている方式であること。

(c) 多重プログラミング・システム

コアに複数の実行中のプログラムをおき，これを少しずつ優先度の高い順から何回にもわけて処理していく方式で，プログラムが細分化され，実行中のプログラムが割り込まれたときになすべき処理方法がきめられている必要があることなどを説明する。

(2) コア・マッピング

つぎに浮動コア・システムについて，**コア・マッピング**の考え方（ワーキング・コア・エリアの区画化（セグメント，セグメントの使用状態（ステータス，ステータス・ビット），**システム・オーバーロード**（あいているコアがなくなったとき）との関連などを説明する。さらに**固定コア・システム**と**浮動コア・システム**の中間的なものとして**区画コア・システム**（Fixed Partitioned Core System）を説明するとよい。

6.3 プログラム実行の制御と割込み処理

制御用コンピュータによるプログラムの実行の制御は，通常の事務・科学計算の場合と異なって割込み（Interrupt）によってなされる。割込みとは現在実行中のプログラムを途中で中断してプログラム制御を記憶装置内のあらかじめ定められたロケーションへ移す現象を起こす機能であることを説明し，プログラム割込みの発生条件として次の事項を具体例を混じえて解説する。

(1) プロセス割込み（Process Interrupt）

プロセス割込みとは，制御用コンピュータ・システムの外部に発生した事象あるいは手動によって割込む場合で，つぎのような例を説明する。

- 温度が限界を超えたとき
- 荷箱が定位置に入り積込み準備が完了したとき
- オペレータがデータの書込みを要請したとき，あるいは日報作成を要求したとき（オペレータ・インタラプト）

(2) システム割込み（System Interrupt）

システム割込みとは，コンピュータ・システムの稼動状態の変化，ハードウェアの故障，演算処理中の異常（オーバーフローなど）などによるプロセスの状態変化やオペレータの操作によらない割込みで，次のような例を説明する。

・A／D変換終了

・データ転送中にパリティ・チェックにかかったとき

・オーバーフロー，アンダーフロー

(3) クロック割込み（Clock Interrupt）

クロック割込みとは，制御の開始時刻がきめられている仕事を開始するための割込みであることを次のような例について説明する。

・一定周期でプロセス変数を読み込んで効率を計算して作表する

・指定された時刻に積算値を上位コンピュータに伝送する。

つぎに割込みレベルと割込み優先順位について，まず単純な1レベルの割込み機能をもつ場合について，ひきつづいて多重レベル割込みの場合について説明し，タスク管理の考え方を理解させる。

6.4 制御用コンピュータのプログラミング

コントロール・システムで用いられる制御用コンピュータのプログラミングは，たとえば技術計算のプログラミングの場合と異なって，タスク（あるいは機能プログラム）を処理するときの順序関係とタイミングが重要であることをまず説明する。たとえば技術計算ではプログラム中でデータが必要になる前ならいかなる時点でデータ・カードを読んでもよく，また処理結果を直ちに出力してもよいし，格納しておいて後で出力してもよいが，コントロール・システムでは入力，出力のタイミングとタスクの処理順序が大切であることを理解させる。

つぎにプログラミングの基本的なパターンとして次の三つを説明し，複雑なプロセスの制御プログラムも基本的にはこれらの組合せで解決されることを理解させる。

① 割込み動作……割込みレベルの上下関係でタスクの処理順序をきめる。

② クロック動作……タスクの開始時刻を中心に処理を進める。

③ シーケンス動作……処理順序があらかじめ与えられている場合である。

ひきつづいて，プログラムを構成する主要な要素およびその内容の概要をつぎの各項について説明する。

① 制御機能……制御プログラムの中心となるもので，プロセス入力の走査，状態の計算，特性の計算，作表，警報などのサブプログラムである。

② 割込み処理……6.3の内容をプログラミングの面からやや具体的に説明する。

③ 入出力処理……多数の入出力装置の動作（非同期的な入出力要求）の管理について概要を説明する。

④ バルク・メモリの利用と情報の転送

⑤ コンピュータ故障時の対応策

指導上の留意点

従来，オペレーティング・システムの説明は，IBM 1800のMPXなどを例にそのマニュアルの内容に従ってページを追って解説していく方法がとられているが，全体像を理解させるためには，より直観的な理解力にうったえるような説明の仕方が望ましいであろう。たとえば，人間の思考過程や動作

などについてのアナロジーを用いて，思考や動作が並行しておこなわれにくいこと（ラジオの音楽をきゝながら複雑な数学の問題は解けない），数学の問題をといているとき，たとえば電話のベルが鳴ってその応対をしなければならないときは，中間結果を脳の記憶（ワーキング・コア・エリア）からメモ用紙（バルク・メモリ）に待避させて記憶しておかなければならないこと，ひんぱんに用いられるデータ（たとえば円周率$\pi = 3.14$）は脳の記憶（コモン・データ・ストレージ）に留めておいた方が便利であること，数学の問題を解いているとき，突発的な事象（割込み），たとえば地震がおこったときには，机の下にもぐるという動作が優先すること，午後10時になったら大学受験の英語講座のラジオを聴くこと（定時処理）などのように説明していくとよい。この際，注意すべきことは，このような人間の思考や動作過程のアナロジーの個々の例の説明に留まらず，それらの相互関係，とくにどの機能（サブプログラム）がどの機能によって管理されているか，すなわち機能の上下関係をはっきりと理解させることである。

第7章　コントロール・システムの実例

目　標

第1章から第6章までの内容を総合的に理解させるため，具体的なコントロール・システムをとりあげて，システムの構築目的，機能，構成などを説明するとともに，システムの計画段階から運用段階までの間に生じた問題点を指摘して，その解決方法を理解させる。

とくに，この章では制御の対象となる現場からどのようにデータを収集してコンピュータに入力しているか，あるいはコンピュータの出力をどのような手段で現場の機器に伝えているかなどを中心に説明することによって，現場の機器とコンピュータとのインターフェイスをよく理解させる必要がある。

また，制御の機能をハードウェアにうまく分担させると，ソフトウェアにかかる負担が非常に軽減できることを具体的な例で説明する。これによって，コントロール・システムを設計するときには，ソフトウェアとハードウェアの機能分担をうまく行なわなければならないことを認識させる。

コントロール・システムを構築するにあたっては，情報処理技術者がハードウェアの技術者（とくに計装あるいは生産担当の技術者）とよく協調をとりながら作業を進めていく必要があることを理解させることも，この章の目的である。

なお，ここで記述することは，あくまで講義の進め方を示すためであり，実例としてとりあげたコントロール・システムの内容を説明するためではない。

内　容

この章を講義するにあたっては，教師ならびに研修生の専門分野，あるいは実際に関係している生産工程などをとりあげて説明するほうが効果的である。

そこでコントロール・システムの具体的な実例として，ここでは生糸の繰糸工程をとりあげ，どのようなことを説明をしていけばよいかを記述する。

7.1　コントロール・システムの概要

実際の繰糸工程においては，どのような作業が行なわれているかを簡単に概説するとともに，コントロール・システムの機能を解説する。ここでは，生産工程の詳細を説明する必要はなくむしろ研修生に新しい用語をなじませる程度に簡単に説明するほうが望ましい。たとえば，繰糸工程の例では，自動繰糸機の働きが中心となっており，糸故障発生回数，接緒回数，給繭補充回数，小枠回転数，繊度調節量盛数といったデータが自動操糸機の制御に重要な意味を持っていることを説明する。

つぎに，コントロール・システムを計画するまでには，どのような開発の過程を歩むかについて説明する。すなわち，繰糸工程の例では，第1段階において，繰糸工程の状態を把握するために必要なデータをコントロール・センタで収集し，そのデータに基づいて，作業員が小枠回転数や繊度調節量盛数の

変更をコントロール・センタで集中操作できる機構を開発したという実績などを説明する。ついで第2段階では，コントロール・センタで収集したデータをコンピュータを使って集計し，自動的に記録をとることができるようなハードウェアとソフトウェアの開発を行なうとともに，第3段階で，コンピュータに集まったデータを利用し，作業員が行なっていた定常的な意思決定をコンピュータに代行させ，自動繰糸機の小枠回転数や繊度調節目盛数を制御するソフトウェアを開発したという事例を解説する。

このような説明によって，コントロール・システムの開発には十分な年月が必要であり，かつまた1歩1歩可能な部分から構築していかなければならないことや，現在運用されているシステムでも，順次改良を加えられることによって効果的なコントロール・システムが完成されたことなどをよく認識させるとよい。

7.2 コントロール・システムの目的

コントロール・システムを設計するにあたっては，まず第一に，コントロール・システムの目的を明確にする必要があり，もし，コントロール・システムの目的を不明確にしたまま，システム設計に着手してしまうと，たとえそのシステムを構築できたとしても，コントロール・システムの動作がまったく役に立たないものとなる恐れのあることを強調する。また，システムの目的をどのように設定するかによって，コントロール・システムの機能自体がまったく異なってしまうことも説明する。たとえば，繰糸工程のコントロール・システム（図7－1参照）では，与えられた人数の作業員で均質の生糸をできるだけ多く生産することが，その目的である。このような目的を設定すると，コントロール・システムは単位時間あたりの糸故障発生個数を，所定の範囲内に抑えるように自動操作機の小枠回転数を調節するが，もし生糸の生産量を減らしてもよいから，商品質の生糸を生産させようとするときには，単位時間あたりの糸故障発生個数をできるだけ少くするように自動操糸機の小枠回転数を調節することになる。

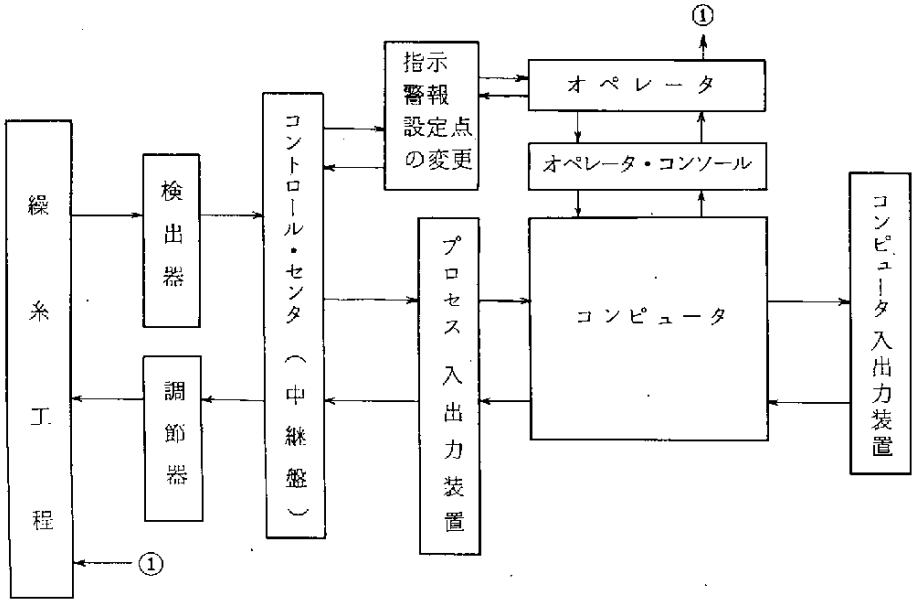


図7－1　コントロール・システムの構成

7.3 コントロール・システムの構成

ここでは，コントロール・システムの機器構成を中心に話しを進める。とくに，コンピュータと工程間のインターフェイスについては十分に時間をかけて説明したほうがよい。

まず，図7－1のようなコントロール・システムの構成図を利用して，各機器の説明を行なう。

(1) コンピュータ

コントロール・システムに使用するコンピュータの機能や特徴，ならびに規模や価格などを説明する。繰糸工程の例では，記憶容量4Kワードのミニコンピュータを採用しているが，どのような根拠で4Kワードのミニコンピュータを採用したかといったことを説明しながら，コントロール・システムに使用するコンピュータの決め方を解説すると効果的である。

(2) コンピュータ入力装置

紙テープ読み取り機や紙テープ穿孔機などの入力装置を簡単に説明する。

(3) オペレータ・コンソール

オペレータがコンピュータを介してプロセスとの間でデータや指示のやりとりを行なうときに使用する機器の概要を解説する。オペレータ・コンソールは，現場の作業員が使用するので，素人でも操作を間違わないように工夫されていることなどを指摘しておくとよい。また，プロセスのモニタリングとか警報を出す場合には，タイプライタで印字するだけではなく，音を出したり，画像で示したりすると，出力情報の価値が高まることも説明するとよい。

(4) プロセス入出力装置

プロセスの計測装置で検出したデータをコンピュータに入力する部分と，コンピュータから処理したデータを出力する部分を図示して説明する。データの収集，伝送，変換については7.4項で説明するのでここではコントロール・システムにおけるプロセス入出力装置の位置づけを明確にすればよい。

7.4 データの収集，伝送，変換

プロセスから，どのような手段でデータを収集し，コンピュータまで伝送するかといったことを概説するとともに，データを収集する計測装置の機能とか，たとえば，ノイズの対策などデータ伝送において考慮する必要がある対策を説明する。繰糸工程の自動繰糸機からコントロール・センタに収集されるデータは，糸故障発生回数，接緒回数，給繭補充回数，小枠回転数，繊度調整目盛の5種類であることを述べそれぞれの計測装置について説明する。たとえば，糸故障発生回数については，糸故障発生を機械的に検知しマイクロスイッチを駆動させて電気信号に変えディジタル信号として検出する。また小枠回転数については，小枠回転軸に取り付けたタコジェネレータの発生電圧によりアナログ信号として検出する。すべてのデータはいったん，コントロール・センタにある中継盤に集められ，ディジタル信号とアナログ信号にそれぞれ分類されて，コンピュータに入力される。繰糸工程から，コントロール・センタにデータを伝送するケーブルには，ノイズが混入するのを防ぐため，シールド・ケーブルを採用していることを指摘しハードウェアの面でノイズの対策をどのようにしているか理解させる。

図7－1の例では，データ伝送の途中に，コントロール・センタの中継盤が設置されており，このコントロール・センタにおいても，データを把握できるようになっている。これは，繰糸工程におけるコントロール・システムの開発が段階的に行なわれたことから生じた特徴であることを注釈しておくとよい。

また，データの変換については，プロセス入出力装置の機能とあわせて解説するとよい。ここでは，とくに具体的な数値を対応させてA－D変換，D－A変換の原理を説明する。

さらに，データを収集するための計測装置や検出器の精度についてもその重要性を研修生に認識させる必要がある。コントロールに必要なデータの精度に比較して極端に高い精度の計測装置や検出器を設置しても無益であり，逆に，必要な精度のデータを収集できない精度の計測装置や検出器を備えても無意味であることを述べる。

しかしながら，データの収集，伝送，変換の各段階で発生する問題点をすべてハードウェアで解決するといったことよりも，ソフトウェアで解決できるものは，できるだけこれを採用するのが得策であることを理解させるため，たとえば，ノイズについては，サムチェックなどを適用することによって，ソフトウェアでも対策を講じることができることなどを説明する。

なお，ここでは，データの収集，変換，伝送などの方式を解説的に述べるよりも，それぞれの段階で生じた問題点の解決方法を中心に説明するほうがよい。

7.5 コントロール・システムにおけるソフトウェア

ここでは，一般の参考書などで説明しているコントロール・システムのソフトウェア，すなわち，オンライン・リアルタイム処理でコンピュータを動かす汎用のシステム・プログラムや，プロセスをコントロールする汎用のアプリケーション・プログラムの説明などは，できるだけ簡単に行なったほうがよい。むしろ，ある特定のプロセスをコントロールするために必要なシステム・プログラムあるいはアプリケーション・プログラムについて説明するほうが効果的である。とくに，システム・プログラムなどは，機能を絞れば非常に簡単になることを述べ，むしろ汎用のシステム・プログラムをコントロール・システムに適用すると問題のあることを指摘するとよい。さらに，コントロール・システムでは，ソフトウェアとハードウェアがお互いに助け合うことによって，それぞれの負担が著しく軽減されることを強調する必要がある。たとえば，ある時点の小枠回転数を検知するとき，ただ1回の検出で小枠回転数を決定しようとすると，精度のよいハードウェアが必要でこれに多額の費用がかかってしまうが，ハードウェア自体の精度がある程度よくなくとも，小枠回転数を何回か検出し，その平均値をその時点の小枠回転数とするといった方法を採用すれば前者と同じ検出精度を確保できるので費用を軽減することになる。

また，7.7で述べるようにコントロール・システムに人間がうまく介在するソフトウェアが非常に軽減されることも指摘しておく必要がある。

7.6 制御機構とコンピュータの役割

制御機構は，プロセスやシステムの果すべき目的によってまったく異なる性格を持つので，一般的な概要を説明するだけでよい。ただ，プロセスの制御機構については，どのようなデータをプロセスから収集し，コンピュータの出力でプロセスのどこを動かせばよいのか，という点を十分に理解させる必要がある。たとえば，繰糸工程の場合，非常に多くの要因があるにもかかわらず，なぜ，糸故障回数と接緒回数と給繭補充回数という，三つのデータを収集し，コンピュータの出力で小枠回転数と繊度調節目盛という，二つの調節点だけを制御するのかについて解説する。

さらに，コンピュータの役割を明確に理解させるため制御機構のコンピュータが処理する仕事の内容を説明する。すなわち，繰糸工程の側では，糸故障発生回数，接緒回数，給繭補充回数の三つのデータを集計する機能（ロギング，集計機能），集計したデータにもとづいて，小枠回転数などの最適値を決定し，指令する機能（制御機能），単位時間あたりの糸故障発生回数が規定より多くなると赤ランプを点灯したり，コンピュータの電源が切れるとブザーを鳴らしたりする機能（警報機能）などを説明するとよい。

7.7 コントロール・システムにおける人間の役割

この章の締めくくりとしてコントロール・システムにおいては，人間がどのような役割を果しているか，あるいはどのような役割を受持つべきかについて説明する。ここでは，“コンピュータを導入したコントロール・システムでは人間がまったく不必要となる”といった観念の誤りを指摘し，コントロール・システムの部分的機能を人間に分担させることがコンピュータへの負担を軽減でき，望ましいコントロール・システムが構成されることを具体的に説明すべきである。コントロール・システムに人間が介在すると，制御機能の柔軟性や信頼性が高まるばかりでなく，コントロール・システムの建設や運用費が格段に安くなるといった事例を解説するとよい。たとえば，繰糸工程の例ではすべての制御機能をコンピュータに持たせておくと，単にコントロール・システムの建設費が高くなるというばかりでなくコンピュータにトラブルが発生して生産が停止してしまう恐れが多くなるので，糸故障回数，接緒回数，給繭補充回数といったデータ以外の要因については人間が判断して制御することにしたり，あるいはコンピュータにトラブルが発生すると，ただちに手動に切りかえて人間が工程を操作するなどの方法を採用して，費用の軽減を図っていることを説明する。

指導上の留意点

この章では，多くの分野にわたる実例をとりあげて表面的な説明をするのではなく，研修生の専門分野における一例だけをとりあげてコントロール・システムのあり方を理解させたほうがよい。

また，コントロール・システムが完成したときに，そのシステムが目的にかなった働きをするかどうかを，適確に評価する方法や，コントロール・システムをうまく保守する方法などについても，簡単に触れておくほうがよい。

さらに，実務経験のある研修生には，コントロール・システムの基本設計くらいまで行なわせると効

果的である。

なお，この章で記述した要点を一般的な言葉で列挙するとつぎのようになる。

7.1 ・コントロール・システムの概要

・コントロール対象のシステムを概説する。

・コントロール・システムの開発過程を説明する。

7.2 コントロール・システムの目的

・コントロール・システムの目的を明確にすることを強調する。

・コントロール・システムの目的と機能の関連を解説する。

7.3 コントロール・システムの構成

・コントロール・システムのハードウェア構成を説明する。

7.4 データの収集，伝送，変換

・データの収集，伝送，変換に必要なハードウェア，ソフトウェアを説明する。

・ノイズの対策を説明する。

・データの経路について説明する。

7.5 コントロール・システムにおけるソフトウェア

・システム・プログラムを説明する。

・アプリケーション・プログラムを説明する。

・ハードウェアとソフトウェアの機能分担を説明する。

7.6 コントロール機構とコンピュータの役割

・コントロールの方式やモデルを説明する。

・コンピュータの処理内容を説明する。

7.7 コントロール・システムにおける人間の役割

・人間と機械の共存方法について説明する。

・人間によるバックアップ体制の効果を説明する。

第8章　シーケンス制御

用　語　この章では次の用語を教える。

検出スイッチ，論理回路素子，限時要素，論理代数，交換則，組合せ則，配分則，ド・モルガンの定理，回路の簡略化，ワイヤード・ロジック方式，数値制御機械，数値制御，NCテープ，APTシステム，群管理システム，APT言語，パート・プログラム

目　標

シーケンス制御の大略を理解させるのが本章の目的である。まず，フィードバック制御方式との対比においてシーケンス制御の特徴を説明し，引き続き，シーケンス制御の基本構成要素について簡単な解説をする。次にシーケンス制御のソフトウェアとして，論理代数の基礎とシーケンス制御との関連を具体例をまじえて説明する。さらにコンピュータ（シーケンスを含めて）によるシーケンス制御の方式として，ワイヤード・ロジック方式のソフト化という行き方を中心に解説する。また，シーケンス制御の実際面からの説明として，実用されているシステム例を用いてシーケンス制御の全体像をより具体的に把握させる。最後に数値制御についての概略的な解説をおこなう。

内　容

8.1　シーケンス制御の概要

シーケンス制御（Sequential Control）は，フィードバック制御と並んで，コントロール・システムを実現するための基本的な方式の一つである。「シーケンス制御とは何か」という命題に対して，一つの定義文で説明することはなかなか難かしいが，一つの方法としてフィードバック制御の定義と対比して説明すると理解されやすいと思われる。すなわち，

① シーケンス制御の定義

「シーケンスとは，現象の起こる順序をいい，シーケンス制御とは，あらかじめ定められた順序，または一定の論理によって定められる順序に従って，制御の各段階を逐次進めていく制御をいう」

（JIS　C0401−1970：シーケンス制御記号による）

② フィードバック制御の定義

「フィードバックによって制御量の値を目標値と比較し，それらを一致させるように訂正動作を行なう制御」

この二つの定義文において，次の二つの事項をさらに説明することによって両者の制御方式の差異を明確に理解させるとよい。すなわち，

① 定められた順序

これにはシーケンス制御は「将棋倒しの制御」であるという例をとってみるとわかりやすいであろう。すなわち，将棋の駒の並べ方が定められた順序であり。（ただし駒を必ずしも一例に並べる必要はなく，途中で分岐がある場合も考えられる）。各駒がそれぞれ制御の各段階をあらわすとし，一番端の駒を押し倒すことによって，シーケンス制御が開始され,次々に隣りの駒を倒していくことにより制御のステップがすゝめられていくことを比喩的に説明するとよい。

② フィードバックによる訂正動作

ここでいうフィードバックとは，システムの現在時点における状態を制御動作（訂正動作）に反映させることであり，この意味で制御動作の結果（出力）を常に（サンプル値制御では一定間隔で）監視しているのがフィードバック制御である。これに対してシーケンス制御は，制御の各段階が終了した時点では，所期の動作が成功裡に成し遂げられたかをチェックするが，（そしてその動作結果によって，一定の論理に従って引きつづく制御動作の順序を定めるが），制御の各段階では全くの「行きっぱなしの制御」であることをはっきり理解させる必要がある。

つぎにシーケンス制御システムで取り扱う情報（すなわち，コントローラの入出力信号）が，「開・閉」，「入・切」，「昇・降」，「増・減」，「始動・停止」というような「2値情報」であることを，簡単なシーケンス制御システム，たとえば，エレベータなどを例に説明するとよい。（ただし，実際のシステムの構成要素の動作は連続的なものであり，その動きをリミット・スイッチ，継電器によって2値情報に変換していること，またシーケンス制御システムの個々の制御には，フィードバック制御が用いられることも多いことを理解させる必要かある）。

8.2 シーケンス制御システムの基本構成要素

シーケンス制御システムで一般に用いられている基本構成要素，すなわち，ハードウェアを次のように分類し，各分類に属する代表的なものの動作原理と目的を理解させる。

(a) 検出スイッチ

検出スイッチとは，人為的でなく，自動的に判断できる能力をもち，特定の物理量の大小の判別ができる能力をもつ機器であり，温度スイッチ，圧力スイッチ，レベル・スイッチ，リミット・スイッチなどの例を説明する。

(b) 操作スイッチ

人間の意志により決定された情報内容を2値信号の形で伝達するための機器にあることを理解させれば十分で，個々の操作スイッチの説明は必要ではない。

(c) 論理回路素子

論理回路素子は，各種リレー（コントロール・リレー，無接点リレー，水銀リレーなど），半導体素子などの組合せにより，特定の論理判断，記憶，計数などの機能をもつものであることを理解させる。

ただし，この説明には論理代数の基礎知識を必要とするので，この時点での説明は，NOR素子，NAND素子などの単純な素子の説明に留め，これらの組合せとしてのやや高度なものは8.3以後で解説するようにする。

(d) 限時要素

予定の時間遅れをもって応動することを目的とする**限時要素**いわゆるタイマーが必要となることを説明する。

8.3 シーケンス制御システムのソフトウェア

8.3で説明した各種の構成要素（ハードウェア）をどのように統合することによって，所望のシーケンス・ロシックを実現できるかを，**論理代数**と論理素子との対比という形で説明する。

(1) 論理代数の基礎

まず並列接点回路と直列接点回路の論理表現という意味で，AND，OR，NOT，NAND，NORなどを説明する。次に論理代数の一般定理とその意味（回路上での実現方法）について解説する。

これは，**交換則**，**組合せ則**，**配分則**，**ド・モルガンの定理**など最も基本となるものを説明すれば十分である。

(2) シーケンス制御における応用

次に単純なシーケンスを論理代数を用いて回路的に実現する方法を説明する。たとえば，次のような単純なシーケンスを用いて説明するとよい。

「A点でキーを押してから，B点のランプをつける。B点でキーを押すとランプは消える。かつA点でキーを押してから復帰する前にB点でキーが復帰するときはランプは消えたまゝである」これは，ランプをL，二次リレーをXとすると

$$L = A \cdot \overline{B} \cdot \overline{X}$$

$$X = (A \cdot B) + (A \cdot X) = A(B + X)$$

という表現であらわされることを説明し，これを実現するための回路を示す。

さらにやや進んだ例として2進カウンターや**回路の簡略化**（特にDon't Care Condition について）基本的な考え方を理解させる。

8.4 コンピュータによるシーケンス制御

ディジタル・シーケンサおよびコンピュータによるシーケンス制御の方法についての大略を説明する。

まず，8.2で説明した論理回路素子を配線して実現する従来の方式（すなわち，**ワイヤード・ロジック方式**）の欠点（製作上のはく雑さ，信頼性の限界，論理変更のやりにくさなど）を述べ，それを解決するための一つの最近の傾向として，シーケンサ（プログラマブル・ロジック・コントローラ）について説明する。これにはコンピュータに近い方式（リード・オンリー・メモリ，リード・ライト・メモリを用いたストアード・プログラム方式）から簡単なものまで種々開発されているが，ここでは，ワイヤード・ロジックのワイヤリングをソフト化した単純なものに重点をおいて説明する。

つぎに，従来のワイヤード・ロジック方式のソフト化とはまったく無関係に，シーケンスを流れ図の形で表現し，それにもとづいて，アセンブリ言語やその他の特殊言語を用いてシーケンス制御を実現する方式について説明する。この方式では，通常のコンピュータ制御との制御差異は判然としなくなるが，限時要素とシステム・タイマーとの関連，および流れ図がほとんどすべて判断ブロックになること

を強調する。例として，大きな慣性を軸負荷としてもつ電動機の起動から一定速度まで上昇させるときのシーケンス制御をコンピュータで行なわせたとして（勿論，このような単純な例はワイヤード・ロジック方式が適しているが），そのときのシーケンスを流れ図の形で表現し，プログラミングする手順を説明するとよい。

この場合には，問題を出来るだけ単純化して，たとえば始動指令→補機運転→始動準備完了→しゃ断器投入（投入条件として始動抵抗最大，切換開閉器運転側，励磁開閉器，始動抵抗器，短絡用開閉器"始動位置"などをチェック）→始動抵抗器減少→始動抵抗器短絡→始動完了というメインとなるシーケンスのみについて説明すればよい。

8.5 シーケンス制御の応用例

シーケンス制御は産業界のほとんどすべての分野で用いられている制御方式であり，実際面から考えるとフィードバック制御方式よりもはるかに応用面が広いが，そのすべての利用形態について説明することは不可能であるので，エレベータ，コンベア，火力プラント，化学工業，セメント，工作機械などの中から適当なものを選んで，更にそれをモデル化して，8.4まで説明した内容を整理して解説する。

この場合，シーケンスの個々のステップを詳細に説明しても意味がなく，たとえば，図8－1のブロックに従って，それぞれのブロックに，とりあげたモデルのシステムのどの機能が対応するかを明確化するように努める。また信頼性，汎用性，環境の良否，価格などの点からの考察もあわせて行なうとよい。

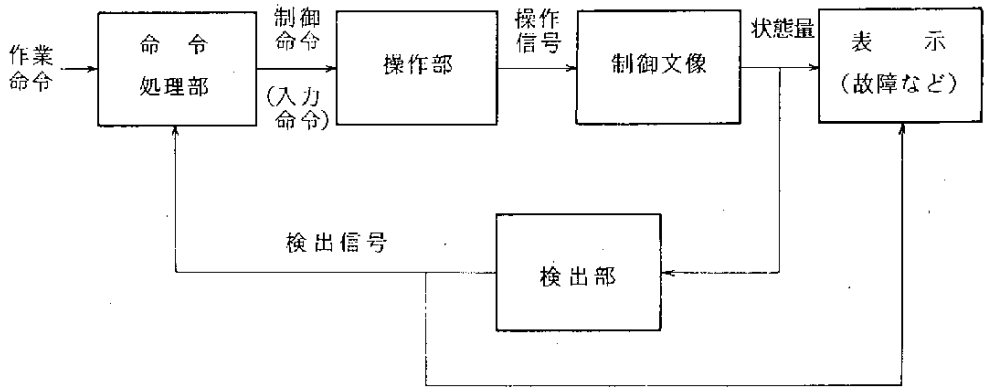


図8－1　シーケンス制御のブロック

8.6 数値制御機械の概要

ここでは，**数値制御機械**そのものを説明するまえに**数値制御**（Numerical Control）という概念を明確にする必要がある。すなわち，8.1で定義した自動制御の概念をもう一度復習し，さらに，数値制御では，**NCテープ**の指令どおりに工作機械を動かすだけで，自動制御におけるフィードバック機構がないということを説明する。これによって，自動制御の概念と数値制御の概念の差異を明確化す

るとともに，数値制御の内容をシーケンス制御の章に包合した観点を明確するとよい。

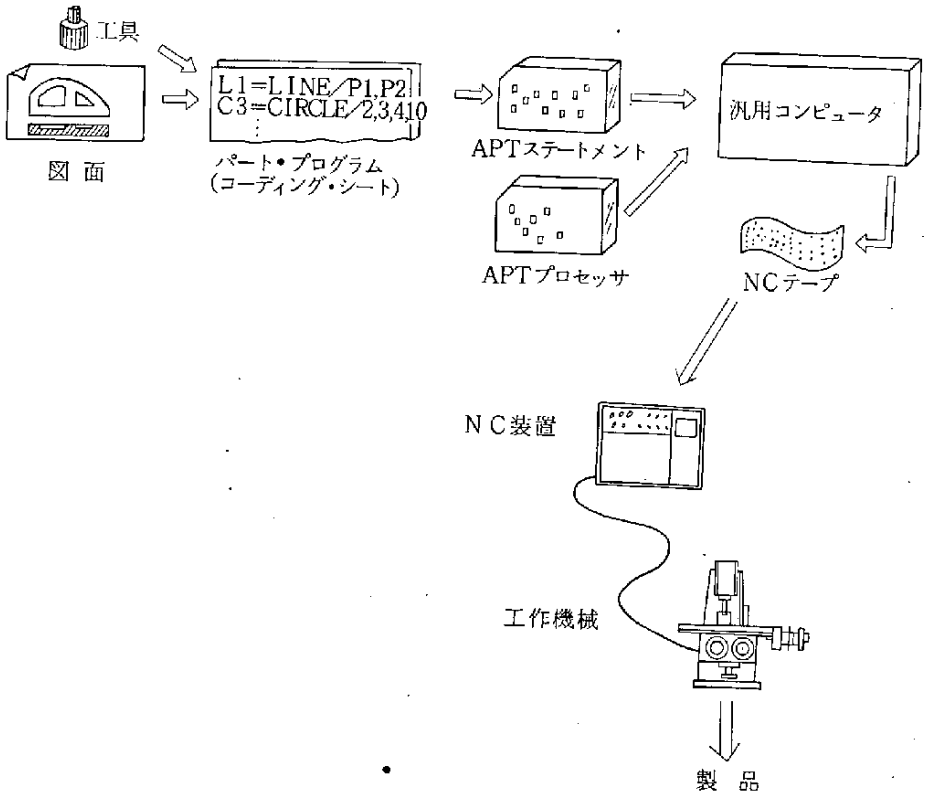


図8－2　数値制御機械の構成

加工したい工作物の図面が出てから，どのような手順で工作機械が動きだすのかといつた作業のフローを説明するには，図8－2のような数値制御機械の構成図を利用するとよい。このとき，NCテープの作り方として，手作業で行なう方法と，**APTシステム**（Automatically Programmed Tools）を利用して，汎用コンピュータが自動的にNCテープを作り出す方法の二通りについて解説を加えたほうがよい。さらに，NC装置と工作機械の関連については，ハードウェアの説明はなるべく避け，NC装置がどのような形態で工作機械をコントロールしているかについて説明したほうがよい。たとえば，NC装置がNCテープから指令数値（Δx，Δy）を読み込み，この数値によって工作機械の動きをコントロールするような機構を具体的に解説するとよい。

また，数台の工作機械を同時に制御する**群管理システム**についても簡単に解説するとよい。このとき，NC装置のかわりに，制御用コンピュータが使われる形態などについても説明を加えると効果的である。

さらに，数値制御装置の利点，欠点について説明することによって，数値制御機械を導入する場合の判断基準をある程度養成しておいたほうがよい。つぎに，NCテープを自動的に作成する代表的なシステムである。APTについて説明し，APT言語によるパート・プログラムの練習問題などをやるとよ

い。APTシステムについては，APT言語で書かれたパート・プログラムから，NC装置にかけられるNCテープを作成するシステムであるといったことを説明し，パート・プログラムの練習問題* に入るとよい。

なお，APT言語は，NCテープを作成する代表例としてあげたものであるから，メーカによってそれぞれ独自の言語を開発してあり，講義を進めていくにあたっては，実際に使われそうな言語を使って説明をしたほうがよい。

指導上の留意点

論理回路素子および限時要素は，コンピュータ（ディジタル・シーケンサ等も含めて）の論理判断機能およびシステム・タイマによって置き換えることが，コンピュータを中核としたシーケンス 制御システムの行き方であり，この場合には，コンピュータのもっソフトウェア機能によってプログラム化される訳であるが，ここでの説明は，適当は具体的シーケンスを例によって，即物的に各種の論理素子がどのような目的のために必要となるかに重点を置く必要がある。

参考文献

(1) 成田誠之助 「システム工学の手法」 コロナ社，45年8月初版 ¥ 1900

コントロール・システム全体における。システム問題解決の基礎理論を解説している。

(2) 野坂康雄編 「鉄鋼業のコンピュータ・コントロール」 産業図書，45年3月初版 ¥ 1800

(3) 宮崎誠一 「化学プロセスの制御」 産業図書，46年1月初版 ¥ 1700

(4) 高松武一編 「プロセス・システム」 日刊工業社，47年3月初版 ¥ 2400

(5) 佐田登志夫編 「生産システム」 日刊工業新聞社，48年1月初版

(6) 研野和人 「数値制御のソフトウェア」 産業図書，45年2月初版 ¥ 1600

参考文献の(2)から(6)は各アプリケーションごとに，それぞれのコントロール・システムを解説している。

(7) ハーコン・マクダニIW 「デシジョン・テーブル入門」 岸田孝一訳，松平誠解説，日本経営出版会 ¥ 1200

* 練習問題としては，参考文献(6)の37ページより38ページに適当なものがのっているので，利用するとよい。

経　営　科　学 (II)

目　　次

第1章　線形計画法 …… 136
　1.1　線形計画モデル …… 136
　1.2　シンプレクス法 …… 137
　1.3　双対理論と感度分析 …… 139
　1.4　各種タイプの線形計画 …… 139
第2章　経 済 評 価 …… 144
　2.1　経済比較における費用 …… 144
　2.2　賃金の流れ（Cash Flow）の時間的価値評価 …… 145
　2.3　代替案の経済比較 …… 145
第3章　ダイナミック・プログラミング（動的計画法） …… 147
　3.1　ダイナミック・システムと多段決定過程 …… 147
　3.2　ダイナミック・プログラミングの例題 …… 148
　3.3　ダイナミック・プログラミング …… 149
第4章　ネットワーク分析と計画管理 …… 152
　4.1　PERT，CPMの歴史と応用分野 …… 152
　4.2　プロジェクトのアローダイヤグラム …… 153
　4.3　PERT計算 …… 155
　4.4　作業時間の不確定性の取扱い …… 158
　4.5　必要資源の見積りと配分 …… 158
　4.6　CPMについて …… 159
第5章　在 庫 管 理 …… 161
　5.1　経営発注モデル …… 161
　5.2　一般化された経済発注モデル …… 163
　5.3　確率的モデル …… 164
　5.4　オペレーショナル・ゲーム …… 165
第6章　予　　測 …… 167
　6.1　時系列分析 …… 167
　6.2　単一方程式モデル分析（回帰分析） …… 168
　6.3　計量モデル（連立方程式モデル） …… 168
　6.4　産業連関分析 …… 169

科目「経営科学（II）」

教育の目標

経営管理におけるコンピュータ利用の効果をさらに高めるために，各種の科学的手法を用いて予測や計画,管理が行なわれるようになってきた。この科目では，経営科学の種々の手法を使って，①具体的に問題を解き，②プログラムを開発し，③システムのメンテナンスを行ない，④コンピュータの計算結果を正しく解釈することのできる者を育成することを目標とする。

このような能力をそなえるためには，各手法が基づいている基礎的理論を理解し，少なくとも基本的アルゴリズムの大筋は習得しなければならない。したがって，かなりの比重を問題定式化や計算手順の演習におく必要がある。そこでまず数理計画，ネットワーク技法をはじめとする各種最適化手法を身につけ，それらの手法を適宜使って問題を解くことができるようにする。また，手法や技法のみにとどまらず，企業の意思決定や計画の具体的問題，また社会・経済事象を分析し，問題の定式化，モデルの構築をも，上級情報処理技術者とともに，その一部を分担し，業務を遂行できる能力を養う。

時間配分

章	履習時間（時間）		
	講　義	演　習	実　習
1.　線形計画法	14	2	
2.　経済評価	5	2	
3.　ダイナミック・プログラミング（動的計画法）	5	2	
4.　ネットワーク分析と計画管理	10		
5.　在庫管理	14	2	
6.　予　測	12	2	
合　計	60	10	

第 1 章　線形計画法

用　語　この章では次の用語を教える。

凸集合，アクティビティ•アナリシス，シンプレクス法，基底解，人工変数，可能解，双対定理，緩急相補性，シャドウ・プライス，輸送問題，割り当て問題，ネットワーク問題

目　標

線形計画モデルの構成と線形計画法（ＬＰ）の基礎理論を理解させる。現実世界での資源配分や計画・管理に関する諸問題を観察し，そこに線形モデルとしての構造を抽出し，定式化する能力を習得させる。すなわち，数学的手法としてのＬＰ理論の基礎知識を得るだけにとどまらずに，現実の問題の本質的構造を洞察して，モデル化する訓練をすることも大切である。例えば，物質の輸送とは何の関係もない生産計画に関する問題であっても，定式化の設定如何によっては，輸送型問題として定式化されることもある。

ＬＰの理論としては，ＬＰモデルの幾何学的特徴，シンプレクス法，双対定理を理解させれば，十分である。ＬＰ計算の結果を正しく理解・解釈でき，その情報を実際の計画・管理に活用できるようにすることを目標とする。その意味において，双対理論を基礎としたシャドウ・プライスの概念の把握は重要である。シャドウ・プライスをつかってＬＰの解の感度分析ができるようにする。

２変数・３不等式程度の簡単なＬＰ問題は，シンプレクス法を用い手計算で解けるようにする。シンプレクス法を使わない特殊タイプのＬＰである輸送問題，割り当て問題，ネットワーク最短経路問題などは，実用規模のかなり大きい問題でも，手計算で解けるようにする。その際のアルゴリズムの基礎として双対理論，そのなかでも緩急相補定理が活用されていることを理解させる。

内　容

1.1　線形計画モデル

(1)　線形計画の定義と特徴

まず，簡単な線形計画（ＬＰ）問題の例を提示し，線形計画の一般形式を整理して示す。制約式は線形不等式群で表わされ，どの変数も非負であるという条件を満す点の集り，すなわち可能解集合の性質を理解させる。そのために２変数，４不等式ぐらいの線形計画の問題をつかって可能解集合を図示する。その点集合の特徴のひとつとして，凸集合であることを指摘するため，**凸集合**の概念を簡単に言葉で説明する。２次元空間（X_1，X_2）内で目的関数 $Z=c_1x_1+c_2x_2$ に特定の関数値 Z_1 を与える点の集まりは直線になること，したがって Zの値を $Z_1<Z_2<Z_3<\cdots\cdots$ と変化させるとこの直線が平行移動することを指摘する。ここまで説明すれば，２変数の線形計画であれば，図で解くことができるようになる。

この時点で，線形計画の可能解集合の特異な点として端点というものがあることを解説し，最適解のなかには少くともひとつ端点があることを指摘することは，後のシンプレクス法を修得させるための前準備になるであろう。しかし凸集合やその端点に関しては，厳密に数学的な定義を与える必要は，この段階ではない。

(2) 典型的線形計画の例

いくつかのよく知られている代表的線形計画問題をひろって，線形計画の構造をもった具体的な問題になじませる。例えば

栄養問題（混合問題の一種） 原料a，bに単価c_1，c_2 が与えられている。それぞれの原料は単位当り，ビタミンA，B，C，Dをそれぞれある一定量含有していることが知られている。ビタミンA，B，C，Dがそれぞれある一定量ずつ要求されるとき，それを最小費用で満足するのに，原料a，bはどれだけづつ購入すべきか。

その他，配分問題，**アクティビティ・アナリシス**や，特種構造をもった線形計画として，輸送問題などを例として示しつつ，線形計画問題になじませるとともに，かなり普遍性をもった応用範囲の広い問題タイプであることを理解させる。

1.2 シンプレクス法

ここでは，**シンプレクス法** で一般の線形計画問題を解けるようにするためその基礎理論を理解させる。

(1) 標準形に直すことができるようにし，基底解の概念を理解させる。

線形不等式は，スラック変数を導入すればすべて，線形方程式に変換できる。したがって線形計画の制約は，スラック変数も含めて，一般にn 個の変数，m個の方程式，（$m \leq n$），そしてすべての変数は非負という形で表現できるが，これを標準形と呼ぶ。

基底解：n 個の変数のうち，（$n-m$）個をゼロとおけば，m個の変数についてのm個の方程式ができるので，これについて解いて得られる解のことを基底解と呼ぶ。またそれらm個の変数のことを基底変数と呼ぶ。ただしこれらm個の変数の係数の行列式はゼロでないものとする。ある基底解のどの変数も非負であれば，この基底解は与えられた線形計画の可能解でもあり，これを可能基底解と呼ぶ。

1.1でつかった2変数，4不等式の線形計画をそのままつかって標準形に直し，可能基底解をすべて示すとよい。例えば，

制約： $X_1 \leq 20$， $X_2 \leq 10$， $X_2 \leq 1+\frac{2}{3}X_1$ ，

$X_1 + X_2 \leq 28$， X_1 ，$X_2 \geq 0$ のもとで

$Z = X_1 + 3X_2$ を最大化せよ。

この例題では，可能基底解は六個ある。そして可能解集合を図示して，それぞれの可能基底解が凸集合のどれかの端点に対応することを指摘する。

1.1の(1)で指摘されたように，少くともどれかの端点が目的関数値を最大化（あるいは最小化）していることから，代数的にも，最適解の探索は可能基底解のなかに限定できることを指摘する。

(2) ひとつの基底解から別の基底解を得るためのピボット操作の習得

例題としては，スラック変数がそのまま基底変数となるようなものをえらぶのがよい。このようにして最初の可能基底解が得られる。

(n－m) 個のゼロ変数のどれかを基底変数にしたいときは，m個の基底変数のどれかがゼロ変数にすることにより，別の基底解を得ることができる。そのための代数的操作は，連立方程式を解く際のガウス消去法と類似のもので，その基底変数としたい変数をどれかの方程式だけに残して他の方程式から消去すればよい。これをピボット操作と呼ぶ。しかし，このようなやり方では，新しく得られた基底解が可能解であるかどうか，また目的関数値は前の可能基底解よりも大きくなっているかどうかの保証がないことを指摘する。したがってどのゼロ変数を基底変数にすれば，目的関数値が増大（減少）するか，またどの基底変数をゼロ変数にすれば新しい基底解が可能解であるかを確かめながら基底変換を行なうのがシンプレクス法であることを説明する。

シンプレクス・ダブローの形と意味を理解させるのに，いま，X_1 ，X_m を基底変数とすると，与えられたＬＰをピボット操作で変換して，

$$
\begin{array}{ll}
X_1 & +\bar{a}_{1,m+1}X_{m+1}+\cdots+\bar{a}_{1n}X_n=\bar{b}_1 \\
\quad X_2 & +\bar{a}_{2,m+1}X_{m+1}+\cdots+\bar{a}_{2n}X_n=\bar{b}_2 \\
\quad\quad \ddots & \vdots \\
\quad\quad\quad X_m & +\bar{a}_{m,m+1}X_{m+1}+\cdots+\bar{a}_{mn}X_n=\bar{b}_n \\
Z & +\bar{c}_{m+1}X_{m+1}\cdots\cdots+\bar{c}_n X_n=\bar{b}_U
\end{array}
$$

の形に整理されることを理解させる。現在その値がゼロである X_{m+1} ，X_{m+2} ，---- X_n の変数のうち，どれを正にすれば，目的関数値が増大（減少）するかを調べるには，目的関数におけるこれら変数の係数 $\bar{C}_{m+1}$ ，$\bar{C}_{m+2}$ ，……$\bar{C}_n$ の符号を調べればよいことを理解させる。例えば，$\bar{C}_K<0$ であれば，X_K をゼロから正にすることにより，Zは増大する。このようにして，X_K を基底に導入することが決定される。

基底に導入する変数 X_K が決定されたら，それをどれだけ正の値として大きくできるかという問題が生じ，X_K を増大することにより他の基底変数の値が負とならない制約内でぎりぎりまで X_K を増大させるのがシンプレクス法であることを理解させ，さらに，X_K をぎりぎりまで大きくすることによりいままで正の値をとっていた基底変数のどれかがゼロの値になる。すなわち，基底変数からゼロ変数となるものが定まる。これによりピボットの行がきまることを説明する。

シンプレスク法の計算手順をいくつかの簡単な例題をつかって演習させたあと，最初の可能基底解をどうして見つけるかという問題が残ったことを指摘する。そしてそのためには何も新しいアルゴリズムを覚える必要はなく，**人工変数**なるものを導入し，それを最初の基底変数として計算を開始する。その際の目的関数は人工変数の和としてそれを最小化すれば，最適解が得られたとき**可能解**が存在すれば，人工変数は基底変数としては残らないから，ともかく与えられた問題のある可能基底解が得られることになる。この段階をシンプレクス法の第１段階と呼ぶ。

ここまででシンプレクス法の説明は完了してよいが，与えられた問題に可能解が存在しない場合は，

第1段階の終了とともに判明することを指摘すること。また，与えられた問題に最適解が存在しない場合，すなわち，目的関数値を無限に小さく（大きく）できる場合も，基底から追い出すべき変数を探す際に，ピボット列のどの係数もゼロか負であることにより判明することを理解させる。

これらいろいろなケースについて，簡単な例題について，手計算をやらせて演習させる。

1.3 双対理論と感度分析

与えられた線形計画問題の双対問題の作り方の説明をし，演習させる。でき上った双対問題の具体的な物理的解釈は通常困難なことが多いから，必ずしも双対問題を物理的に解釈させようと努める必要は最初はない。ある程度機械的に双対問題の作り方を定義してよい。しかし，でき上った双対問題のさらに双対問題を作るともとの主問題にもどってしまうことを理解させ，双対構造の対称性を理解させる。主問題の制約式と双対問題の変数の間に対応関係があり，主問題の変数と双対問題の制約式の間に対応関係がある。

双対定理に関しては，いくつかの異る表現があるが，重要なことは，主問題か双対問題のどちらかに可能解が存在するときは必ず他方にも可能解が存在することである。主問題が最小化問題であれば，主問題可能解の目的関数値は双対問題可能解の目的関数値よりも小さくなることはないということ，しかも，目的関数値が等しくなるような1組の主問題可能解と双対問題可能が必ず存在することを理解させる。次に，双対理論で，応用上とくに重要な定理は**緩急相補性**（complementary slackness）定理である。この定理を理解させる最もよい方法は，簡単なＬＰ問題の最適解を与えて，緩急相補定理をつかってそれが最適解であることを証明できるようにすることである。すなわち，与えられた主問題最適解に対して緩急相補条件を満す双対可能解を求める演習をさせる。輸送問題，割当て問題，ネットワーク・フローなど特殊タイプ線形計画問題はシンプレクス法を用いず，加減算だけで解けるアルゴリズムがあるが，それらのアルゴリズムの理論的基礎となっているのは双対理論，とくに緩急相補定理である。

双対変数の別名を**シャドウ・プライス**（shadow price）といゝ，線形計画の感度分析に有益な情報を提供する。たとえば，資源配分，アクティビティ・アナリシスにおいて，最適解が得られたとき，第 i 番目の資源制約不等式の右側の定数（利用可能の資源量をあらわしている）が1単位だけ緩和されれば，目的関数値はどれだけ改良（増加あるいは減小）されるかを示しているのがシャドウ・プライスであることを説明する。

1.4 各種タイプの線形計画

(1) 輸　送　問　題

輸送問題は応用分野が広範であり，その解法も理解しやすいので各自手計算で解けるように指導する。まず，m個の発送地（供給地）とn個の目的地（需要地）からなる輸送問題を定義し，その構造的特徴を説明する。発送地 i から目的地 j への輸送量を示す変数 X_{ij} を定義し，それらが $m \times n$ の配列（行列）に並べられることを示し，輸送問題の制約式はこの配列の第 i 行の変数 X_{ij} の総和が発送地 i の供給量 a_i に第 j 列の変数 X_{ij} の総和が目的地 j の需要量 b_j に等しく

ならなければならないものであることを示す。また，i からj への単位輸送費C_{ij} も$m \times n$ の配列に並べることができる。

輸送問題を線形計画として定式化したらどのような形になるかを示し，また，その双対問題をも書いて見せる。双対問題では各発送地 i に変数U_i が対応し，目的地 j に変数V_j が対応すること，また主問題のX_{ij} 配列の各マス目（i ,j ）に対応して双対制約式$U_i + V_j \leqq C_{ij}$ があることを理解させる。

X_{ij} の配列のマス目は全部で$m \times n$ 個あるが，そのうちせいぜい$m+n-1$個だけのマス目に正の配分（$X_{ij} > 0$）を与えることにより，常に需要・供給の条件だけは満すことができる。もし，この正の配分のマス目が互いに独立な位置にあれば，すなわち，正の配分のマス目でだけ直角に方向を加えることができるようにしてあとは直進する経路で閉じたループが存在しなければ，この可能解のことを与えられた輸送問題の可能基底解と呼ぶ。

ある与えられた可能基底解が最適解であるかどうかをテストするにはすでに1.3 で習得した方法で，緩急相補条件を満す双対解を求めさせる。このとき，せいぜい$m+n-1$個の正のマス目に対応した$m+n-1$個の方程式$U_i + V_j = C_{ij}$ を満す双対変数は容易に見つけることができる。これらの双対変数が配分がゼロのマス目に対してすべて$U_i + V_j \leqq C_{ij}$ を満足すれば，現在の可能基底解は最適解である。しかし，　$C_{ij} - U_i - V_j < 0$　であるようなゼロ配分マス目があれば，このゼロ配分マス目（i ，j ）に正の配分をするように解を修正して総輸送費をさらに小さくできることを説明する。その修正方法は，いま見つけたゼロ配分から正配分にすることに決めたマス目からスタートして直進し，他の正配分において直角に方向を変えてまたスタートしたマス目にもどってくるような閉じたループに沿って行なえばよいことを理解させる。以上を理解すれば，輸送問題を手計算で解くことができる。

輸送型の問題に定式化する演習としては，たとえば，次のような生産計画問題が適当であろう。

「いまから，向う4期間，製品に対する需要が与えられていて，それぞれ r_1 ，r_2 ，r_3 ，r_4 単位とする。1期間中の生産容量は a 単位までであり，各期間における生産費は単位当り，それぞれ d_1 ，d_2 ，d_3 ，d_4 円とする。また，製品をある期間から次の期間まで在庫として保管するときの在庫保持費は製品単位当り1期間b 円とする。これら条件のもとで，生産および在庫費を最小で，各期の需要を満たすには各期においてどれだけずつ生産すべきであるか。」

(2) 割り当て問題（Assignment Problem）

割り当て問題には，通常の割り当て問題と単純割り当て問題とがあるが，ほとんど区別しないで考えられることが多い。しかし，それぞれを定義して，線形計画として定式化してみせるようにする。これらが輸送問題の特種な場合であることを指摘する。しかし14(1)で習得させた輸送問題を解くアルゴリズムを割り当て問題に適用しても退化の問題が生じ，うまく行かない。割り当て問題を解くうまい方法がいろいろ考案されているが，そこまで学ばせる必要はない。こゝでは，機械の割り当てや，スケジューリングの問題を割り当て問題に定式化する指導をすればよい。

(3) ネットワーク問題

輸送問題を一般化すると**ネットワーク問題**になることを説明する。最大フロー問題，最小費用フロー問題をそれぞれ線形計画で定式化して説明する。そして，輸送問題，割り当て問題はこのネットワーク・フロー問題の特種ケースとして理解させる。

ネットワーク・フロー問題では，もしアーク容量b_{ij}が整数であれば，どのアーク・フローf_{ij}も整数であるような最適解が存在することを説明する。

広い範囲の応用問題がネットワーク・フローで定式化できるが，適当な応用例としては，二点間の最短経路問題をネットワーク・フロー問題として定式化してみせるのがよいであろう。この場合，ソースからシンクへのフロー値が1単位であるようなフローのなかで総費用 $\sum_{i,j} C_{ij} f_{ij}$ が最小となるようなフローを求めるのである。この最小費用フロー問題を解けば，最適解は整数であるから各アーク・フローf_{ij}は，0か1の値をとることになり，$f_{ij}=1$のアークをソースからたどって行けばシンクに到達する。この経路がソースからシンクへの最短経路である。

最短経路を求めるアルゴリズムは，この最小費用フロー問題を線形計画として定式化し，その双対問題を考え，緩急相補正定理を適用することにより，きわめて簡単なラベリングの方法で解くことができる。このアルゴリズムをつかって，最短経路問題を手計算で解く演習を指導する。

最短経路問題として定式化できる応用問題の例は非常に多い。特に，ダイナミック・プログラミングで解ける決定論問題の多くは，最短経路問題としても定式化されるから，その定式化の演習をさせる。

(4) 用語定義

① 割り当て問題（Assignment Problem）

n個の職があり，これに対してm人の応募者がいるとする。しかし，各応募者はn個の職のうちのいくつかの限られた職にだけ適していて，適していない職のために採用することができない。このとき，どの候補者をどの職に採用するようにすれば，最も多くの候補者を採用することになるかというのが単純割り当て問題（Simple Assignment Problem）と呼ばれるものである。これに対し，候補者iを職jに割り当てたときの便益C_{ij}（あるいは費用）が与えられているとき，総便益（あるいは総費用）を最大（最小）にするには，どの候補者をどの職に割り当てるべきかというのが割り当て問題である。

② ネットワーク（Network）

ネットワークあるいはグラフ（Graph）は，ノード（Node）と呼ばれる対象の集りNと二つのノード，たとえばxとyを結ぶ線（x，y）の集合Aとから成り立っている。ノード間を結ぶ線のことをアークとも呼ぶ。たとえば，交通システムをネットワークで表現すると，交差点がノードであり，交差点間を結ぶ道路がアークである。アークには方向が与えられている場合と，与えられていない場合とがある。

③ ネットワーク・フロー（Network Flow）

ネットワーク・フローとは，ネットワークの各アーク（i，j）に付与された非負の実数値f_{ij}

で，次のような各ノードで保存条件を満足するものである。あるノードi に向ってアークで結ばれているノードの集合をA_i とし，ノードi から他のノードへ向ってアークで結ばれているノードの集合を B_i とすると，保存条件とは，

$$\sum_{k \in A_i} f_{ki} - \sum_{j \in A_i} f_{ij} = 0 , \quad i = 1, 2, \cdots\cdots\cdots n$$

であり，この保存条件が各ノードで満足されているような f_{ij} の集合のことを与えられたネットワーク内のフロー（flow）と呼ぶ。

ときに，ネットワーク内の二つの特別のノードに対して，保存条件を満たさないことを認める場合がある。その一つはソース（source）と呼ばれるノードs で，このノードに関しては，

$$V = \sum_{j \in B_s} f_{sj} - \sum_{k \in A_s} f_{ks} > 0$$

が成立し，もうひとつのシンク（sink）と呼ばれるノードt においては，

$$-V = \sum_{j \in B_t} f_{tj} - \sum_{k \in A_t} f_{kt} > 0$$

が成立する。このVの値のことをソースs からシンクt へのフロー値と呼ぶ。通常，各アーク(i,j) のフローf_{ij} には上限 b_{ij} が与えられて，f_{ij} は$0 \leqq f_{ij} \leqq b_{ij}$ の範囲に制約される。この上限定数 b_{ij} はアーク（i ，j ）の容量と呼ばれ，このアーク内を流れうるフローの限度を示している。最大フロー問題（maximal flow problem）とは，各アークの容量が与えられているとき，ソースからシンクへのフロー値vを最大とするフロー f_{ij} を求める問題のことである。

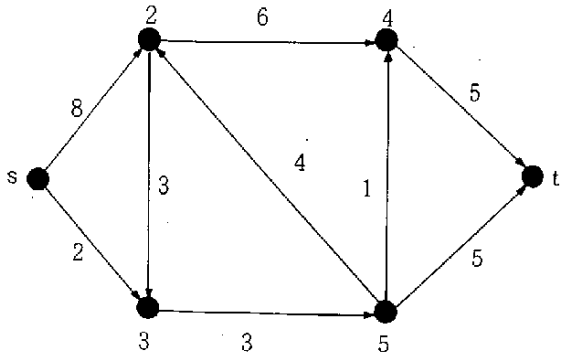


図1－1　ネットワークとアーク容量の例

図1－1にネットワークの例を示す。各アークのわきに示した数がそのアークに与えられた容量である。また　図1－2にこのネットワーク内のフローの例を示している。このフローの値はV＝7

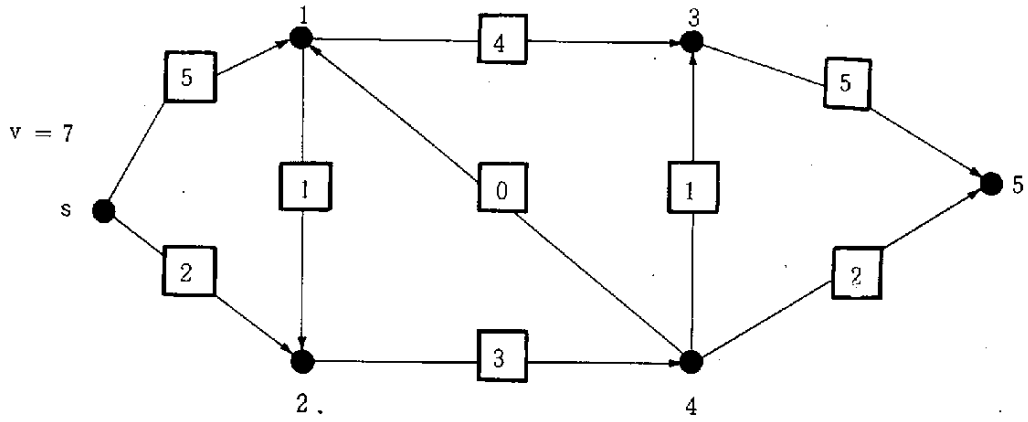


図1－2　ネットワーク・フローの例

でこれはまだ最大のｖの値ではない。

④　最小費用フロー時間

アーク（ｉ，ｊ）内を流れるフロー f_{ij} に対して，単位当り費用 C_{ij} が与えられたとき，ソースs からシンクt へのフロー値v が特定の値ｒ になるようなフローのうちで，その総費用 $\sum_{i,j} C_{ij}\ f_{ij}$ を最小とするようなフローを求めよというのが最小費用フロー問題である。

指導上の留意点

(1)　線形計画のプログラムは，現在はほとんど改訂シンプレクス法がつかわれている。しかし，ここでは標準のシンプレクス法だけ教えればよい。ここで線形計画を解くアルゴリズムを教える目的は，線形計画における重要な概念と原理を理解させ，コンピュータ•アウトプットを正しく解釈できるようにすることである。この点が習得されてあれば，行列演習に慣れていれば改訂シンプレクス法は独自で学ぶことがきる。標準形シンプレクス法も改訂シンプレクス法も基本的概念・原理は同じで，計算効率が違うだけである。

(2)　線形計画を意思決定の道具として，正しく解釈し，感度分析を行なうなどの柔軟な使い方が習得されるためには，双対理論の理解とシャドウ・プライスの概念が非常に重要であるから，十分に演習させること。

(3)　文章で記述された応用問題を読みながら，その問題のＬＰ定式化を行なう演習を十分に行なうこと。

(4)　ＬＰ計算のプログラム・パッケージがあれば，それを使って各自が定式化した問題のインプット・データを用意させ，解かせてみる。計算結果の解釈，感度分析をさせレポートを書かせる。

(5)　特種タイプのＬＰについては，研修生の能力をみた上で，輸送問題だけやって，あとは省略してもよいであろう。その場合は，双対理論の応用演習としても，輸送問題は，じっくり時間をかけて学ばせるのが望ましい。問題定式化の演習も大切である。

第2章　経済評価

用　語　この章では次の用語を教える。

機会損失費用，埋設費用，減価償却，現価比較法

目　標

設備投資や代替的加工法の選択などの意思決定における経済的評価の基礎として，まず，費用の考え方を理解させる。費用を考えるにあたり，意思決定の代替案を評価するのに考慮するべき費用部分と，当面関係のない費用部分を明確に区別できる能力を習得させる。

各代替案について，それらを採択したことに伴う費用や便益の実現が時間に依存する場合，どのようにして同じ基準にたってそれらを評価するかを説明する。そのために，減価償却，資本回収，資本コスト，現価比較，終価比較などの考え方を身につけさせる。そして，これらの概念を基礎として，設備投資や意思決定代替案を比較評価することができるようにする。

内　容

2.1 経済比較における費用

経済的選択の意思決定における費用の概念には混乱が起りやすいから，明確な考え方を身につけさせる必要がある。

ある代替案Aを選択するとき，それにより手持ちの資源がその代替案実施のために使われ，他の用途に使う機会が失なわれてしまう。この資源をA以外の最もよい用途に用いたときの便益が失われることになる。この失われた便益が代替案Aを選択することの**機会損失費用**，あるいは機会費用（opportunity cost）と呼ばれるものである。わかりやすい例としては，「ある土地の所有者がその土地を空地で1年間遊ばせたとして，その費用はゼロであろうか」という問題提起を行なうとよい。親からただでもらった土地をそのまま放置する場合でも，もしそれを人に貸して年間30万円の収入が考えられるときは，その30万円というかせぎ損った便益はその土地を放置することの費用であると考えなければならない。たまったお金を銀行に預けないで，タンスの中にしまっておくことにも，同様に機会費用が考えられると説明する。

次に，いくつかの代替案を経済的に比較するには，それらの間の相対的損得だけを考えればよいという考え方を身につけさせる。たとえば，A案とB案を比較するには両案に共通な費用や利益は考察から除外してよいということである。このことを相対費用の原理という。特に，過去において支出してしまった経費は，代替案のどれを採択するかでとり返しがきかない（回収不能）ものは当面の意思決定には無関係であり，無視してよい。これを**埋没費用**（sunk cost）という。これを説明する例としては，次のようなものがよいであろう。ある店Aで，機械を100万円で買ってもうすでに割賦払いで15万円払ってしまった時点で，他の店Bで同じ機械を80万円で売っているのに気がついたとする。このとき，店Aにはあと85万

円支払えばよいが，もし店Aの契約をとり消して15万円を没収された上で店Bの機械（80万円）を買えば，その費用は没収費15万円プラス機械の値段80万円で計95万円だから，この契約取り消しの方が高くつくと考えてよいかというように問題を提起するとよい。ここでは，この意思決定がどう決まろうと，すでに支払った15万円は没収されているからこれは埋没費用である。

2.2 資金の流れ（cash flow）の時間的価値評価

金銭の価値は時とともに変化するという原理に基いて，具体的な評価法を身につけさせる。まず，導入として，「いまから10年後に百万円返すから，いまお金を貸してくれといわれたとして，あなたならいまいくらまでなら貸してくれますか」と聞いてみる。もうすこし難しくして，「いまから10年間，毎年10万円ずつ返しますからお金を貸してくださいといわれたとして，いくらまでいま貸してくれますか」という問題を考えさせる。これらの問題を正しい筋道で考える基礎として，自分が持っているお金を人に貸すという意思決定の費用について正しい認識が必要であることを指摘する。こゝでも機会費用の概念が有効である。自分のお金を人に貸すということ以外の代案として一番よいものが銀行に定期で預金することであれば定期の利子率だけ，機会費用として，人に貸すという代案にも課せばよい。したがって，その費用は自分がお金を使う他の代案としてどんなものをもっているかに依存する。

利子率の値が適当に定まったら，上の問題は現価係数，資本回収係数をかけることにより解答がきまってくる。現価係数，資本回収係数の導き方も説明する。このほか年金終価係数，年金現価係数，減債基金係数について説明し，その用途を理解させる。これらの係数については，年数nと利子率iの関数として数値表があるから，その数値表をつかう演習もさせる。

減債基金係数をつかって毎年の**減価償却費**を算出する方法も説明する。

2.3 代替案の経済比較

いくつかの代替案を経済的に評価・比較するには，すべて同一の基準にたって比較しなければならない。各案の毎年の収益，操業費を評価し，純利益（償却前利益）の毎年の流れを現在価値，あるいは最終年価値に換算して比較するのが普通である，もちろん，初期投資も同じ基準で考察する。機械や設備に関する投資代替案を比較する例題をつかって考え方を説明する，各代案の初期投資，固定費，操業費，設備の終年価値などがすべて異る場合でも，系統的に比較する能力を習得させる。**現価比較法**，終価比較法のほかに，代案の利益率を算出して比較した方が容易な場合もあるから，例題をつかって解説する。

収益や収益率を算出するときは，減価償却費を差し引いてはならないことを強調する。減価償却費は税金計算のための利益算出には重要な問題であるが，投資の意思決定のためには考える必要はない。

指導上の留意点

(1) この章は，手法とか計算手順などは別に難しいものはなにもない。経済比較における費用の考え方，キャッシュ・フローの時間的評価が正しく理解され，代案比較の経済的考え方がわかればよい。

(2) 個人の生活レベルにおける身近かな例をつかうのがよいであろう。たとえば，次のようなものがよいであろう。

(a) 10年後に家を建てる資金として500万円を目標として積み立てるには毎年いくら積み立てればよいか。

(b) 毎月2万円ずつ20年間返済していく能力が自分にあるとして，利率7%なら，自分はいくら借金することができるか。

(c) 100万円借りて，毎月1万円ずつ11年間返済するよう要求されたとして，これは利率がいくらの計算であるか。

(3) 投資や取替の意思決定において，重要な判断要因は経済性だけではないことを指摘しておく必要がある。資金調達の可能性，成長性，融通性，危険性，その他不確定性などが総合判断されて決定がなされる。その場合でも，経済評価だけでも正しく数量的になされていれば，総合判断はより正確にされやすくなることを強調する。

第3章　ダイナミック・プログラミング（動的計画法）

用　語　この章では次の用語を教える。

ダイナミック，多段決定過程，状態変数，最適政策，再帰関係式，最適性の原理

目　標

時間変数を含むシステムの挙動をある期間にわたって最適化したいとき，ある一時点の1回限りの決定ではなく，その期間にわたって一連の決定をつぎつぎと行なってシステム挙動を制御しなければならない。まずそのようなシステムを記述し，理解させる。多段決定過程の構造を解説し，例題をつかいながら，ダイナミック・プログラミングの考え方を把握させる。

ダイナミック・プログラミング（DP）は，時間変数を含むという狭い意味の決定問題に限らず，多種多様の最適化問題に対するひとつの解決手段であることを理解させる。ダイナミック・プログラミング的発想の特徴として，多次元の困難な最適化問題をより低い次元の解決しやすいいくつかの小問題（サブ問題）に分割して解こうとする点があげられるが，複雑大規模な問題に対するこのような接近法を身につけさせるのがこの章の重要な目標である。複雑大規模な問題に対してそれを一気に解決しようとする古典的方法は無力に近いことが多いが，これをいくつかの小問題に分解してしまえば比較的取り扱いやすくなるものである。しかも，それら分解された小問題は一つ一つが独立した小問題ではなく，これらは一連の段階の系列をなし，ひとつの小問題を解いたあとそれを包含するさらに大きな小問題を解くというぐあいに，次ぎつぎと小問題の系列が最適化され，ついに大域的最適解にいたるのである。このような階層構造で問題解決をはかる手法として，ダイナミック・プログラミングを理解させる。

ダイナミック・プログラミングの基礎として最適性の原理を十分に理解させる。

内　容

3.1　ダイナミック・システムと多段決定過程

ダイナミック・プログラミングの計算手法を説明する前の準備として，時間の経過とともに展開するダイナミック・システムの性質をよく理解させる。そのよい例として，次のような生産・在庫システムを記述するのがよいであろう。N ケ月間の期間にわたって，各月の需要 r_1，r_2，……，r_N，が与えられている。第 i 月の生産量を X_i 単位として，第 i 月の生産費は $g_i(X_i)$ で与えられている。また，第 i − 1 月から第 i 月へ在庫として持ち越す量を I_i とそれに対して在庫費 $h(I_i)$ がかかる。このシステムでは初期の在庫 I_1 でスタートして，1月に X_1 生産したとすると，2月初め（1月末）の在庫量 I_2 は，$I_2 = I_1 + X_1 - r_1$ で表現される。したがって，

$$I_{i+1} = I_i + X_i - r_i, \quad i = 1, 2, \cdots\cdots, N-1$$

こゝで，X_i の系列X_1，X_2，……，X_N が意思決定変数であり，これら一連の X_i の値が定まれば変数I_2，I_3 ……，I_N の値も定まり，Nケ月間の在庫量の動きが確定する。このシステム の動きは，I_1 からスタートしてX_1 を決めるとI_2 がきまり，さらに X_2 を決めるとI_3 がきまるというように展開して行く。このように決定をつぎつぎに行なうことによって，状態がつぎつぎと変化して行く過程として，**多段決定過程** (multi stage decision process) を理解させる。また，決定変数X_i の値の系列$\{X_1, X_2, \cdots\cdots X_N\}$ を政策（policy）として定義する。各期の 在庫量I_i はこの系の状態の動きを記述する変数であることからI_i は**状態変数** (state variable) と呼ばれることをも理解させる。第 i 期始めにおける系の状態がI_i であったとき，決定 X_i を行なうと次 の第 i ＋1 期の系の状態がI_{i+1} に推移するが，その新しい状態 I_{i+1} の決り方が，I_i と X_i のみに依存し，系がどのようにして I_i に推移してきたかという過去の履歴によらないならば，I_{i+1} はI_i と X_i のみの関数という意味で

$$I_{i+1} = T(I_i, X_i), \quad i = 1, 2, \cdots\cdots, \quad N$$

と書くことができる。こゝで例として示した生産・在庫システムにおいては

$$T(I_i, X_i) = I_i + X_i - r_i$$ であることを理解させる。

さて，システムの挙動が望ましい動きになるように，決定変数X_1，X_2，……X_N の値を決めたいのであるから，その望ましさの尺度が必要であり，それを$\{I_i\}$と$\{X_i\}$の関数で示したものを基準関数と呼ぶ。上の生産・在庫システムの例では，Nケ月間の総生産費と在庫保持費の和

$$\sum_{i=1}^{N} (g_i(X_i) + \sum_{i=1}^{N} h(I_i) = \sum_{i=1}^{N} g_i(X_i) + h(I_i)$$

が最小となるように $\{X_i\}$ の系列を決めたいのであることを説明する。そして，基準関数の値を最小とする決定変数の系列$\{X_1^o, X_2^o, \cdots\cdots X_N^o\}$ のことを**最適政策** (optical policy) として定義する。

3.2 ダイナミック・プログラミングの例題

ダイナミック・プログラミングの基本原理を一般論として説明する前に，まず簡単な数値例を解いて見せながら，この手法の考え方と特徴を説明するのが効果的である。たとえば次のような例題を解いてみせるのが適当であろう。

〔例題〕「ある時点から，3期間にわたって合計100単位を生産したい。第 i 期において X_i 単位生産する費用を$g_i(X_i)$ とする。こゝで $g_i(X_i)$ はX_i を第3期末まで保持する在庫費をも 含むものとする。いま，$g_1(X_1) = 40X_1$，$g_2(X_2) = 2X_2^2$，$g_3(X_3) = X_3^2$ で与えられたとき，総費用を最小とするには各期にどれだけずつ生産すればよいか。」

まず，この問題を 3.1 で理解させたダイナミック・システムの枠組で理解することから説明する。決定変数X_1，X_2，X_3 であることは，すぐわかるはずであるが，状態変数はどのような量である かを考えさせる。そして，次のような状態変数I_i を定義してみようと提案する。すなわち，I_i は生産すべき 100 単位のうち第 i 期の始めにおいて，これから先の残りの期間において生産すべき量として残っ

ている部分を示す。第1期においてはもちろん$I_i = 100$であり，$I_{i+1} = T(I_i, X_i) = I_i - X_i$ の関係がある。

第3期初めにおいて，未生産部分がI_3であれば，これが最終期であるから$X_3^* = I_3$ でなければならない。そしてその費用は

$$f_1(I_3) = X_3^2 = I_3^2 \text{ である。}$$

第2期初めにおいて，未生産部分が仮りにI_2であったとして，X_2はどのぐらいかを考える。そのためには，次の問題を解く。

$0 \leqq X_2 \leqq I_2$ の制約で，

$$2X_2^2 + f_1(I_3) = 2X_2^2 + (I_3)^2 = 2X_2^2 + (I_2 - X_2)^2$$

を最小化する。この最小化された値はI_2に依存するので$f_2(I_2)$と記す。すなわち

$$f_2(I_2) = \min_{0 \leqq X_2 \leqq I_2} \left\{ 2X_2^2 + (I_2 - X_2)^2 \right\}$$

これを解くと，$X_2^* = \frac{1}{3} I_2$ でなければならないことがわかる。そのとき $f_2(I_2) = \frac{2}{3} I_2^2$ である。同様に考えて

$$f_3(I_1) = \min_{0 \leqq X_1 \leqq I_1} \left\{ 40X_1 + f_2(I_1 - X_1) \right\}$$

$$= \min_{0 \leqq X_1 \leqq I_1} \left\{ 40X_1 + \frac{2}{3}(I_1 - X_1)^2 \right\}$$

これを解くと，$X_1^* = I_1 - 30$，　$f_3(I_1) = 40I_1 - 600$　である。この例題では$I_1 = 100$ であるから

$$X_1^* = 70,\ X_2^* = \frac{1}{3} I_2 = \frac{1}{3}(30) = 10,\ X_3^* = I_2 - X_2 = 20$$

そして最小化された総費用は$f_3(100) = 3400$　上の計算における$f_j(I_i)$のjは残りの計画期間数をあらわす。

3.3 ダイナミック・プログラミング

上の例題を解いたプロセスの特徴を次のように整理して説明する。

(1) 三つの決定変数X_1，X_2，X_3を求めるために，問題を三つの段階に分割し，おのおのの段階では一つの変数についての最小化問題を解いている。そして，各段階での問題はすべて同じタイプの方程式

$$f_j(I) = \min_{0 \leqq X \leqq I} \left\{ g_i(X) + f_{j-1}(I - X) \right\}$$

ただし，　　$f_1(I) = I^2$

からなり，三つの問題は $j=1$ からつぎつぎに代入して得られる。この方程式は**再帰関係式**と呼ばれる。

(2) 決定変数を直接求めようとはしない。むしろ，関数 $f_j(I)$ を $j=1,2,$ の方向，すなわち時間経過と逆の方向に，求めて行く。関数 $f_j(I)$ を求めるには，どうしてもその前に関数 $f_{j-1}(I)$ が求められていなければならない。関数の系列 $\{f_j(I)\}$ が求められてから，その結果を利用してこんどは時間の経過にそって決定変数の値が求められる。

(3) 各段階でまだ生産されずに残っている量を I と仮定して，関数 $f_j(I)$ を求める。したがって段数が異る場合に対し，また異る I の値に対しても問題の解が得られていることになる。たとえば，上の例題では，第2期と第3期の2期間に60単位生産しなければならないときの最適解は

$$X_2 = \frac{1}{3} I_2 = \frac{1}{3}(60) = 30$$

$$f_2(I_2) = \frac{2}{3}(I_2)^2 = \frac{2}{3}(60)^2 = 2400$$

とわかっている。

以上のような多段階法がうまく行くのは，たとえば，第1期の最適な決定変数値 X_1^* がまだわからないうちに，第2期の決定変数の最適値 X_2^* は，第2期の系の状態値 I_2 のみに依存して，決めることができるからである。すなわち，第 i 期の系の状態 I_i が与えられれば，それ以降の最適決定は，第 i 期以前にどのような決定をして系が I_i の状態になったかには無関係に，決めることができる。このようなことがいえるときに，問題を解いたときの最適政策の重要な性質として次のことがいえる。

「初期状態と初期決定がなんであろうとも，以後の決定は初期決定から結果する状態に対し最適政策をつくっている」これは**最適性の原理**と呼ばれるもので，このことをよく理解させる必要がある。

3.1で説明したように，システムの推移が $I_{i+1} = T(I_i, X_i)$，$i=1, 2, \cdots\cdots, N$ で記述され，第 i 段のシステム評価関数が $g_i(I_i, X_i)$ であるとして，N期間中に最小化（あるいは最大化）したい目的関数Fが

$$F = g_1(I_1, X_1) + g_2(I_2, X_2) + \cdots\cdots + g_N(I_N, X_N)$$

のように，各段階の状態と決定の関数の和で表現されるか，あるいは適当な変換によってこの形になる場合には，その最適政策は，最適性原理に記述された性質をもち，ダイナミック・プログラミングの方法で解くことができる。そして，その場合の一般的，再帰関係式を次のように示す。

$$f_j(I_i) = \min_{x_i} \left\{ g_i(I_i, X_i) + f_{j-1}(I_{i+1}) \right\}$$

$$f_1(I_N) = g_N(I_N, X_N)$$

たゞし　　$I_{i+1} = T(I_i, X_i)$，$j = N - i + 1$

$f_j(I_i)$　システムの状態 I_i が与えられたとき，残りの j 個の期間を最適決定で経過したときの最適目的関数値。

ダイナミック・プログラミングは，必ずしも時間変数を含み時間とともに状態推移するダイナミックなシステムの最適化問題だけに限って有効な手法なのではないことを指摘しておくべきである。多変数の最適化問題で，最適性の原理が成立する多段決定過程に定式化，あるいは変換できれば，ダイナミック・プログラミングは常につかえることを説明する。例えば，次のような数理計画法の例題を与えるとよい。

〔例題〕「いまS円の資金があり，これをN個の投資活動に配分したい。第i 番目の活動に X_i 円投資したときの便益は $g_i(X_i)$ である。総便益を最大にするには，資金S円を各活動にどれだけずつ配分すべきか。」

この問題は要するに，

$$X_1 + X_2 + X_3 + \cdots\cdots + X_N \leqq S$$

$$X_i \geqq 0 \qquad i = 1, 2, \cdots\cdots, N$$

の制約のもとに $\sum_i g_i(X_i)$ を最大化せよ。

という資源配分問題である。この問題の簡単なケース，N＝4, についてダイナミック・プログラミングで解く演習をさせるとよい。

指導上の留意点

(1) ダイナミック・プログラミングは例題を豊富に与えて，考え方に慣れさせながら教えるのが効果的である。

(2) 決定論的問題（deterministic problem）でダイナミック・プログラミングの手法が十分に身についたら，確率的問題（probabilistic problem）を与えるとよいであろう。しかし，原理は全く同じであることを理解させる。

(3) 最適性の原理を理解させるには，最適性原理が成立しないような問題を上げて，その違いを理解させるとよい。

(4) ダイナミック・プログラミングは定式化における基本的態度の一つとでもいうべきもので，線形計画などのようにはっきりしたアルゴリズムをもつわけではない。決定変数の個数を少なくした小問題の系列を順ぐりに解くことを繰り返し，ついに大域的最適解（global optimal solution）を得るという接近法である。このような接近法を身につけさせる。線形計画のような一般的コンピュータ・プログラムなどは用意されてないので，一つ一つの問題についてプログラムを書かなくてはならないが，そのプログラムは比較的単純なフローから成り立っている。

(5) 最短経路問題なども，ダイナミック・プログラミングで解かせてみるとよい。

第4章　ネットワーク分析と計画管理

用　語　この章では次の用語を教える。

PERT，アロー・ダイアグラム，最早イベント時刻，早遅イベント時刻，作業の時間的余裕，クリティカル・パス，3点見積り，CPM

目　標

PERT，CPM技法の歴史の概要を理解させ，これらがプロジェクトの計画・管理者にどのようにして用いられ，どんなことが可能になり，どんな情報を提供するかを理解させる。また，これらの手法の応用分野についての認識をもたせる。

プロジェクトを構成する作業間の先行関係から，アロー・ダイアグラムをかけるようにし，それをもとにして，クリティカル・パス，最早ノード時刻，最遅ノード時刻，各作業の余裕を計算できるようにする。

各作業の持続時間について，最可能値，楽観値，悲観値が与えられたとき，それをもとにして，プロジェクトの期待完了時刻，また完了時刻が特定日数よりも遅れる確率を求めることができるようにする。

各作業が必要とする各資源量がわかっているとき，ネットワークから，プロジェクトの期間中において各資源利用がピークになったりする時期を発見できるようにする。その場合，資源調整や作業時間の調整法を理解させる。

CPMによるプロジェクト完了時刻と費用の関係の分析の概要を理解させる。

内　容

4.1　PERT，CPMの歴史と応用分野

PERT (Program Evaluation and Review Technique) の起源については，アメリカ合衆国海軍の艦隊弾道ミサイル計画（ポラリス計画）の計画改善と管理の必要性から創始されたものであり，海軍，ロッキード社とブーズ・アレン・ハミルトン社が協力してPERTを開発し，1958年にはじめてポラリス計画に適用されたことについて触れる。ポラリス・ミサイルをはじめの予定より2年も早く動かせたPERTの功績は有名である。CPMについては，1957年ごろから開発されたもので，デュ・ポン社の新工場の設計と管理のため，W. R. Walker，レミントン・ランド社のJ. E. Kelleyなどが主体となって，最小の増加費用で工期の短縮を図るために開発された。

PERT，CPMの応用分野については，次のような分野で適用例がみられる。

① 研究開発計画

　マーケッティング：販売拡張計画

長 期 計 画：新製品開発，コンピュータ導入，モデル・チェンジ計画

② 土 木 建 設

橋梁，建物，船舶，道路，都市建設

③ 工 場

新規建設，移転

④ サービス関係

ラジオ・テレビ放送番組の編成，映画，演劇のシナリオから完成まで

⑤ 政府・官庁

都市の災害復興，都市計画，経済開発

⑥ 軍 事 関 係

新兵器開発，作戦・戦術計画，基地建設，兵力動員計画

4.2 プロジェクトのアロー・ダイアグラム

アロー・ダイアグラムのかき方について以下のように説明する。一つのプロジェクト（Project）はいくつかの作業（アクティビティ，activity）からなりたっており，それらのアクティビティを列挙して各々のアクティビティについて，それに先行する作業とそれに後続する作業をリストした表をつくることからはじめる。たとえば表4－1のようなものを示す。

この作業リスト表からネットワークを描くことを教える。個々の作業（アクティビティ）をアロー（矢線，arrow）で表現し，このアローでイベント（event）と呼ばれる二つの○印の結合点の間を結ぶ。あるイベントiから他へ向って出ているアローで表わされる作業は，このイベントiに向って結ばれているアローの作業がすべて終了するまでは，開始することができないという先行関係（precedence relation）が表わされていることを説明する。イベントには番号をふるが，PERT計算手順を簡単にするためには，任意のアローについて，始点となっているイベントの番号の方が，そのアローの終点となっているイベントの番号よりも小さくなるように，番号づけをした方がよい。

表4－1 リ ス ト

作業名	先行作業	後続作業
A	ナ シ	D,E,F
B	ナ シ	E,F
C	ナ シ	H
D	A	G
E	A，B	G
F	A，B	H
G	D，E	ナ シ
H	F，C	ナ シ

任意の二つのイベント間は，ひとつ以上のアローで結ぶことができないように約束されている。したがって，平行して進行できる二つの作業を表現するには，ダミー作業が必要である。ダミー作業が必要になるもう一つの場合は，図4－1(a)のように，DはA，Bの両方が終了しないとスタートできないが，CはAだけ終了すればスタートできるような先行関係を表現したいときである。このようなときは，図4－1(b)のようにダミー作業を設けてうまく表わせることを説明する。この場合，図4－1(c)のようにしてはいけないことをよく理解させる。与えられた作業リストと先行関係からネットワークが描けるように演習で指導する。

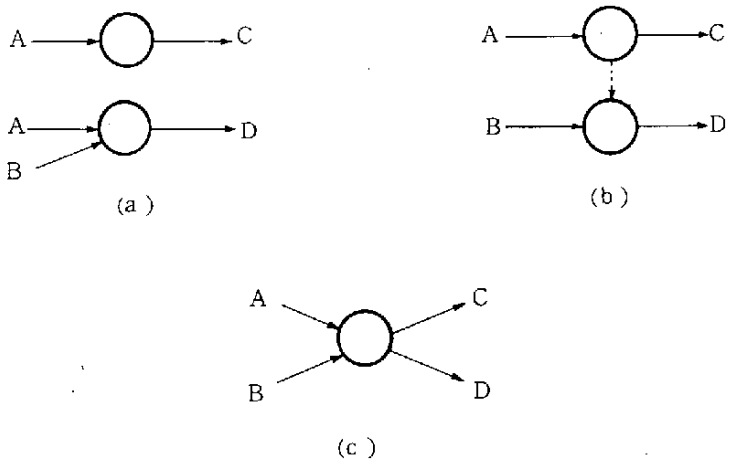


図４－１　ダミー作業の使い方

その他アロー・ダイアグラムをかくに際し，知っておくと便利なことを記す。

① 作業の分割

図４－2(a)のように，先行作業の半分が終れば後続作業はスタートしてよい場合がある。このようなときは，その先行作業を分割して別個の作業として取扱えばよい。そして図４－2(b)のように書けばよい。

② 作業の行なわれないアロー

材料を発注してそれが到着するまでの待ち時間や，コンクリートが固まるまでの待ち時間は実際に作業は行なわれないが，アローで表わしておく。

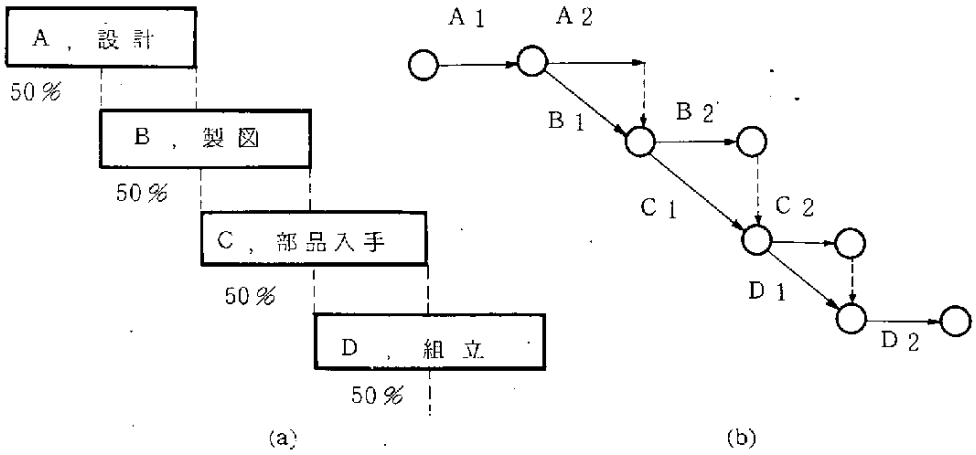


図４－２　作業分割の例

③ 時間に関する制約

作業間先行関係とは独立して，プロジェクト開始後 m 日以降でないとある作業 x がスタートできないとか，ある作業Yが終了して n 日経過しないとプロジェクトが完了しないような場合も，それらはアローで表現する。（図4－3参照）

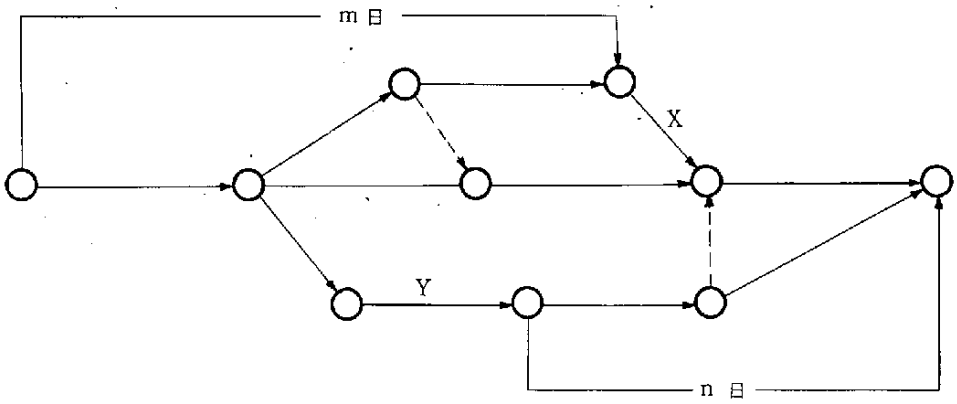


図4－3　時間に関する制約

4.3 PERT計算

各イベントｉに関して，**最早イベント時刻**（earliest event time）t_i^e と**最遅イベント時刻**（latest event time）t_i^l，という概念を明確に理解させる。まず，最早イベント時刻 t_i^e は，イベントｉを始点としているアローの作業（ｉ，ｊ）が開始できる最も早い時刻である。t_i^e の計算手順はプロジェクトの開始するイベント1について，$t_i^e = 0$ とする。作業（ｉ，ｊ）の所有時間を D_{ij} とすると，イベントｊの最早イベント時刻 t_j^e を計算するには，このイベントｊに向かって結ばれている作業（ｉ，ｊ）のすべてについて，$t_i^e + D_{ij}$ を求め，そのうちの一番大きいものをえらぶ，すなわち，$t_j^e = \max_{\bullet N} (t_i^e + D_{ij})$ を計算すればよい。

この計算を，ｊ＝1,2,……，Nについて繰り返す。最終イベントNの t_N^e がプロジェクトの完了時刻，すなわち完了時刻の可能最早のものである。

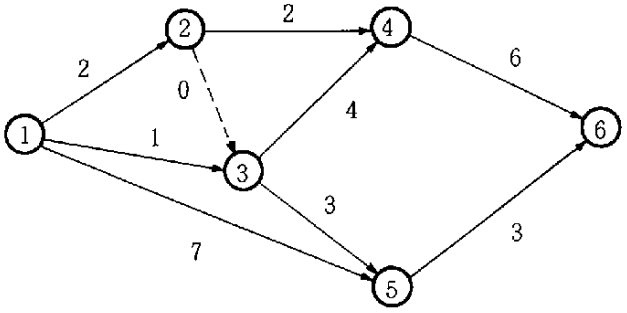
プロジェクトの最早完了時刻 t_N^e が求まったら，各イベントｉの最遅イベント時刻 t_i^l を計算することができる。これは，プロジェクトの完了を t_N^e よりも遅らせないために，イベントｉを始点としているアロー（ｉ，ｊ）の作業が開始しなければならない最も遅い時刻のことである。t_i^l の計算手順としては，まず，$t_N^l = t_N^e$ とおく。イベントｉの t_i^l を求めるには，ｉを始点として結ばれているアロー（ｉ，ｊ）のすべてについて $t_j^l - D_{ij}$ を求め，そのうちの一番小さいものをえらび t_i^l とする。

すなわち

$t_i^l = \min_j (t_j^l - D_{ij})$ である。この計算を $i = N, N-1, N-2, \cdots, 2, 1$ の順番に進めて行く。

図4－4(a)のネットワーク t_i^e, t_i^l を計算するには，同図(b)のようなマトリックスのワーク・シートで行なわせる。

各イベントについてこれらの時刻の概念と求め方が身についたら，こんどは各作業（i，j）について，最早開始時刻（earliest starting time），ES_{ij}，最早終了時刻（earliest finishing time） EF_{ij}，最遅終了時刻（latest finishing time） LF_{ij}，最遅開始時刻（latest starting time） LS_{ij}，などを説明する。このうち，イベント時刻と一致するものは，$ES_{ij} = t_i^e$, $LF_{ij} = t_j^l$ の2つである。また $EF_{ij} = ES_{ij} + D_{ij} = t_i^e + D_{ij}$ であり，$LS_{ij} = LF_{ij} - D_{ij} = t_j^l - D_{ij}$ である。



(a)

t_i^e	i \ j	1	2	3	4	5	6	
	1		2	1		7		
	2			0	2			
	3				4	3		
	4						6	
	5						3	
	6							
(b)								t_j^l

図4－4　t_i^e と t_j^l の計算のためのワーク・シート

作業（i，j） についてこれらの時刻が求まると，**作業の時間的余裕**を求めることができる。まず，作業（i，j）の全余裕（total float） TF_{ij} は $TF_{ij} = LS_{ij} - ES_{ij} = t_j^l - D_{ij} - t_i^e$ で求めることができる。これは，作業（i，j）を最早時刻 t_i^e でスタートした場合はこの作業の終了はこの全余裕分 TF_{ij} まで遅れても，後続の作業で予定通り進行すればプロジェクト完了時刻に影響がないことを示す。つぎに，自由余裕（free float） FF_{ij} は

$$FF_{ij} = t_j^e - EF_{ij} = t_j^e - t_i^e - D_{ij}$$

で求める。自由余裕分以上作業（i，j）が遅れると後続の作業（j，i） は最早開始時刻にスタートできなくなる。これらの用語の理解に混乱を起さないように図4－5を示すとよいであろう。

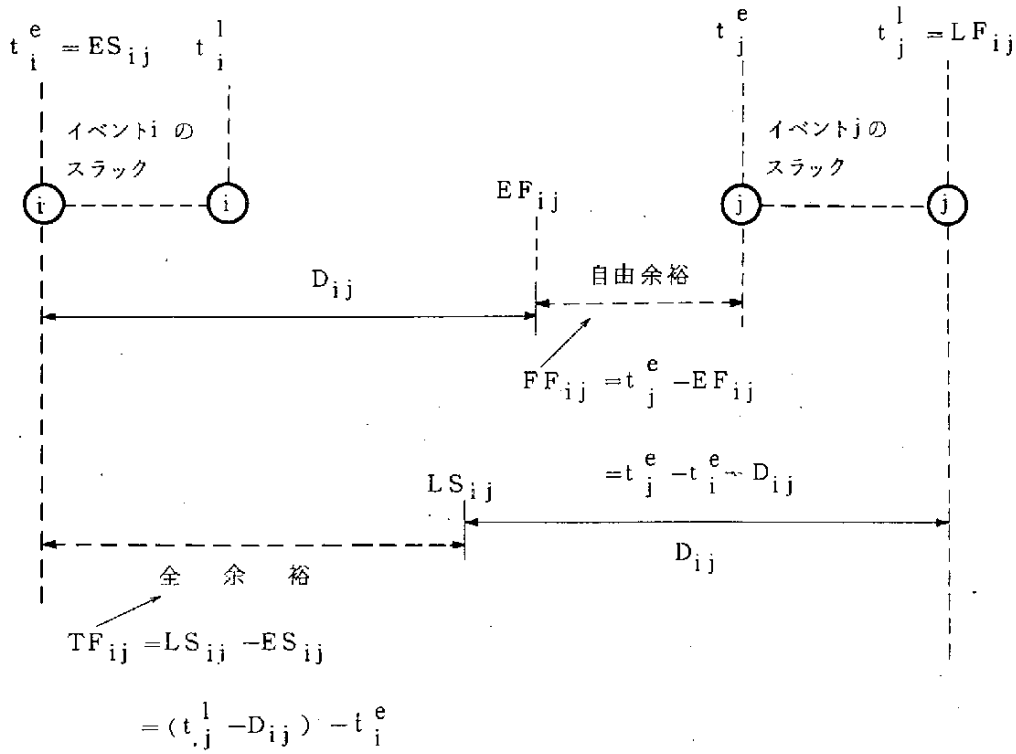


図4－5　　イベント時刻，作業時刻，作業余裕の間の関係

これらの時間を計算したあと，各作業ごとに D_{ij} ，ES，EF，LS，LF，TF，FFをリストした表をつくって書き込ませる。自由余裕のない作業をプロジェクト開始イベントからたどって行くと，自由余裕ゼロの作業の系列でプロジェクト終了イベントまで通っている。この作業の系列を**クリティカル・パス**（critical path）と呼んでいる。また，クリティカル・パスが通るイベントにおいては，最早イベント時刻と最遅イベント時刻が等しい，すなわち，イベント・スラックはゼロである。クリティ

カル・パスの意味はよく説明する必要がある。

4.4 作業時間の不確定性の取扱い

各作業（i, j）の遂行所要時間の推定として，楽観値 a_{ij} と呼ばれる最もうまくいった場合の所要時間，その反対に最も悪い結果を想定した悲観値，b_{ij}，いちばん可能性が高そうな最可能値，m_{ij}，の3種類の推定値をつかった**3点見積り法**を習得させる

これらの推定値から，次式によって，所要時間の期待値 D_{ij} と分散 σ^2_{ij} を推定する。

$$D_{ij} = \frac{1}{6}(4m_{ij} + a_{ij} + b_{ij})$$

$$\sigma^2_{ij} = [\frac{1}{6}(b_{ij} - a_{ij})]^2$$

これらの推定式は，どうやって導かれるかについては，くわしく説明する必要はない。期待値 D_{ij} をつかって，PERT計算を行なえば，プロジェクト完了時刻の期待値 T_e が求まる。これはクリティカル・パス上の作業（i, j）^{c} の期待値の和である。すなわち，

$T_e = \sum_{(i,j)^c} D_{ij}$ D^c_{ij} である。プロジェクト完了時刻の分散 σ^2 は，クリティカル・パス上の作業の分散 σ^2_{ij} の和である。

$$\sigma^2 = \sum_{(i,j)^c} D_{ij} \ \sigma^2_{ij}$$

これらは確率論の定理で，独立な確率変数の和の期待値はそれぞれの変数期待値の和に等しく，分散もそれぞれの分数の和に等しいということをつかっている。しかも，中心極限定理により，プロジェクト完了時間は，期待値が $T_e = \sum_{(i,j)^c} D_{ij}$ D_{ij} で分散が $\sigma^2 = \sum_{(i,j)^c} D_{ij}$ σ^2_{ij} の正現分布をなすとみなしてよいことを理解させる。このことから，プロジェクト完了時間が特定の日数以上になる確率は正規分布表から求めることができる。このような方法を3点見積り法と呼び，PERTが1958年のポラリス・ミサイル開発計画に使われたときはこの方法が用いられていた。これは単純な不確実性の取り扱い方であるが，実用上役立つから十分習得させること。

4.5 必要資源の見積りと配分

プロジェクトを構成する各作業は，いくつかの資源をその所要時間にわたって必要とする。たとえば，資金，各種労働力，機械，設備などである。各作業について各種資源の必要単位数がわかっていれば，PERTにより各作業の最早開始時刻がわかるので，これらを基にして，各資源ごとに，月単位あるいは週単位で資源必要量の見積り表を作製することができる。このように各作業が最早時開始時刻にスタートすると仮定して，その日程に基づいて，各資源がどの期間どれだけの単位必要とするかを積み上げて行くと，資源必要量の時間的変動パターンが階段状の起伏をもったグラフで表現される。このような操作を山積みと呼んでいる。

研修生には，PERT計算のあと，山積みにより，月単位，あるいは週単位の資源必要量の見積り報告書を作製させ，その検討を指導する。

資源見積り表の検討には，各資源ごとに，どの時点，期間において必要量が手持ちの資源量を越えるかを観察させる。そして，クリティカル・パス以外の作業を最早開始時刻から早遅開始時刻の間で移動させ，必要量が制限量を越えないように変更できるかどうかを机上で検討させる。このような操作により必要量の山積みのピークをくずして，平滑化することを山くずしと呼んでいる。

山くずしにより，必要量を平滑化した結果，すべての資源必要量が制限量以内におさまることもあるし，どれかの資源必要量がどうしても，制限を越えてしまうこともある。しかし，このようにして最終的に決った必要量は，管理者が，日々進行する作業の必要量として，検討して行く基準となり，ときには超過勤務を課員にさせるとか，新社員を採用するなど，資源獲得の準備・計画をする重要な資料となることを説明する。また，企業の進行状態の遅滞の予想とその対策の資料にもなるのである。

4.6 CPMについて

CPM (critical path method) でつかわれる正確なアルゴリズムを習得させるには，相当の時間がかかり，つかわれる変数の数も多く繁雑であるから，PERT，CPM関係の仕事に現に従事している研修生層でないかぎり，避けるべきであろう。ただ，CPMではどんな問題を取り扱い，どのような結果を与えてくれるとかについては理解させておく必要があろう。

PERTとCPMの大きな相異点は，CPMでは各作業（アクティビティ）の持続時間は変数X_{ij}として取り扱われ，作業（i，j）の持続時間をある限度内で1単位短縮すると，追加費用がC_{ij}だけかかると仮定されている。作業（i，j）の所要時間は，標準時間がb_{ij}であるが，緊急で作業すれば，緊急時間a_{ij}まで短縮できる。すなわち，$a_{ij} \leq X_{ij} \leq b_{ij}$，この範囲内で作業（i，j）を$X_{ij}$の期間で終了したときの費用は$K_{ij} - C_{ij} \cdot X_{ij}$として表わすことができる。CPMでは，どの作業も標準時間b_{ij}だけかけるとしたときのプロジェクト完了時刻$|\lambda_M|$をまず計算する。つぎにどの作業も緊急時間a_{ij}でおえたときのプロジェクト完了時刻λ_0を計算してくれる。プロジェクトの実際の完了時刻λは$\lambda_0 \leq \lambda \leq \lambda_M$の範囲の値をとれるわけであるが，$\lambda$をこの範囲内の特定の値$\lambda'$にしたいとき，それを最小の追加費用が達成するには各作業の持続時間X_{ij}はどんな値にすべきか，また，作業の開始時間はいつか，そしてそのときの最小費用などの値を求めるのがCPMである。プロジェクト時間を短縮するためには，クリティカル・パス上の作業を短縮しなければならないし，そのなかでも追加費用の小さいものから短縮して行く。しかしクリティカル・パスの時間をある限度以上短縮すると，別の作業の系列がクリティカル・パスになってくる。このようにプロジェクト時間を，次第に短縮して行くにつれて，クリティカル・パスがいろいろと変ってくるので計算が繁雑になることが理解されるであろう。

指導上の留意点

(1) プロジェクトの初期計画の立案・評価におけるPERT，CPMの手法を習得させると同時に，経営管理者に提供する管理情報としてのPERT，CPMの内容・質，その効果と役割などについて適切な認識を与えるよう心がけるべきである。ことに以下のような点を指摘する。

① 経営管理・進行状態報告の改善：将来起こりそうな事態，トラブルに対しても，タイムリーに報告を提供し，適切な手段を講じることを可能にする。計画通りにプロジェクトが進行しているかど

うか，計画からはずれているならば，どのような影響が各方面に波及するかを事前に知らせる。

② 問題所在の確認：手持ちのどの資源がいつ緊迫するか，どの作業がネックの原因か，など管理者が焦点をあわせるべき問題領域をつきとめる。

③ 連絡の改善：一つのプロジェクトに関係している多くの管理者が同一チャート・グラフ類を手にして，計画全体に関する理解・認識を深めることができる。責任の所在，問題点の解決法についても客観的な分析とコミュニケーションを可能にする。

④ 資源のよりよい管理：作業間・部門間の資源の合理的な配分と融通を可能にする。

⑤ 意思決定の結果の評価：いくつかの可能なプロジェクト実行についての代替案をネットワークで評価し，その代替案の資源的，時間的な意味を調べることを可能にする。プロジェクトの修正・改良についても，その意味を検討できる。

⑥ 時間の節約

(2) ＰＥＲＴ，ＣＰＭのプログラム・パッケージは，優れたものが利用できるから，それらをつかって，入力データを準備させ，結果を解釈する演習をさせるのが望ましい。

第5章　在　庫　管　理

用　語　この章では次の用語を教える。

保管費，発注サイクル，リード・タイム，発注点法，安全在庫，定期発注法，（s，S）方式

目　標

在庫管理問題は，どの企業においてもいたるところに存在し，企業組織のみならず，個人生活のレベルでも，公共組織においても存在する普遍的な問題であることを理解させる。そのために，いくつかの基本的在庫モデルを，決定論的な場合と，不確実性の介在する場合とについて，理解させる。それらをつかって，最適な発注量や発注のタイミングの求め方を学習させ，最適解の導き方を身につけさせる。在庫管理問題の基本的要素とその構造を理解させ，在庫問題への接近法，考え方を身につけさせる。費用パラメータの推定誤差などに対する感度分析，すなわちモデルから導かれた解と実際問題に対する真の解との間の費用差の検討なども考え方を学習させる。

在庫問題の本質や考察されるべき要因についての理解を深めるために，研修生ひとりひとりが在庫管理者の役割を演ずるオペレーショナル・ゲーミングをクラスで行なう。そして費用構造や需要の不確実性，リード・タイムなどの大小に依存して，在庫管理ポリシーはどのように調節されるべきかについての考え方を身につけさせる。

内　容

5.1　経済発注モデル（economic lot size model）

このモデルは単位時間当りの所要量（需要）が近似的に一定の値 r であると考えてよい場合に適用され，在庫量の時間的変動パターンが図5－1に示すようなのこぎり状になる。このモデルでは，需要

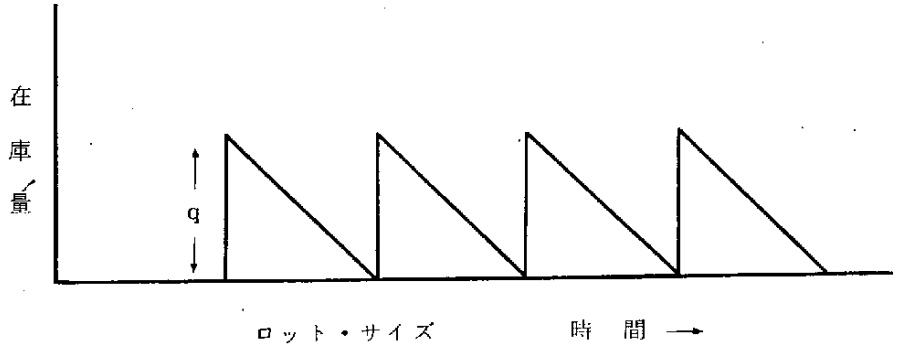


図5－1　経済発注モデルにおける在庫変動パターン

が決定論的にわかっているから在庫量がゼロになったときに，ロットサイズqだけ注文する。在庫管理において考慮されるべき主要な費用は3種類あり，

① **保管費**（holding cost）C_1（円／単位／単位時間）

これは在庫に積まれる資金の機会費用，在庫スペースのレンタル賃，事務，保険，税金などを含む費用で，在庫量や在庫時間に依存する（通常は比例する）費用である。

② 品切れ費（shortage cost）C_2（円／単位／単位時間）

これは需要があるにもかかわらず，在庫がないため，需要発生時に出荷できないときのペナルティ費用である。需要発生時に出荷できなくても，遅れて需要を満たすことがゆるされる場合には，このペナルティ費用は品不足量とその持続（需要発生からそれが満たされるまでの）時間に依存（比例）する。

③ 発注費（あるいは生産準備費）（set up cost），C_3（円／回）

これは注文の量や一回当りの生産量に依存せず，単に注文ごとに，あるいは生産準備をするごとにかゝる一定の費用である。

在庫問題に関連したこれらの費用については，くわしく説明したほうがよい。

最もよく用いられる経済発注モデルでは，図5－1に示されているように，品切れになることを認めない場合である。注問量qが在庫にあったとして，それがなくなるまでの時間はq／rである。この時点でまたqだけのロットが入荷し，単位時間当りrだけ在庫が減って行くというパターンを繰り返す。このように繰り返される変動パターンの周期のことをすなわちq／r のことを**発注サイクル**という。（発注されてから次の発注がなされるまでの時間）

1サイクル当りの総費用は，$\frac{1}{2}qC_1(\frac{q}{r})+C_3$ であり，これを単位時間当り費用に換算すると

$$K(q)=\frac{1}{2}C_1q+\frac{C_3\cdot r}{q}$$

となり，このK(q)を最小とする。発注量は

$$q=\sqrt{2r\frac{C_3}{C_1}}$$

であることを導くことができる。

このモデルが適用できる身近な例として，毎月の生活費が7万円で銀行へ預金をおろしに行くわずらわしさを1000円と考えたとき，銀行へ行ったときはいくらぐらい預金を引き出すべきかという問題を考えさせるとよい。すなわち，r＝¥70,000，C_3＝¥1000がわかっている。保管費C_1については，第3章で学習した機会損失費という概念の応用として，お金100円をタンスの中にしまっておくことの費用を考えさせるとよい。タンスの中にしまっておかないで投資したとして，5％の年利益率の機会を失ったと考えれば，月当り換算して $C_1=0.05\times\frac{1}{12}$ である。

したがって

$$q = \sqrt{2\,(70000)\,\frac{1000}{0.05/12}} \fallingdotseq 183000 \quad (円)$$

そしてこれにしたがうと年間銀行に何回行く必要があるかを計算すると

$$\frac{(70000)\times(12)}{183000} = 4.59 \quad 回である。$$

したがって，4～5回銀行へ行って預金を引き出せばよいというのが最適解である。研修生に，この解と各自実際に何回ぐらい行っているかの行動を比較させる。またC_1 やC_3 が 倍になったり，半分の費用であったとき，解はどのようにかわるか，各自の行動はC_1 と C_3 がどのぐらいの値のときの行動に相当するかなどを検討させる。

5.2 一般化された経済発注モデル

これは図5－2に示すように，発注量qが一度に入荷するのではなく，単位当りk単位のレイトで生産する場合である。また，需要が遅れて満たされることも認める。

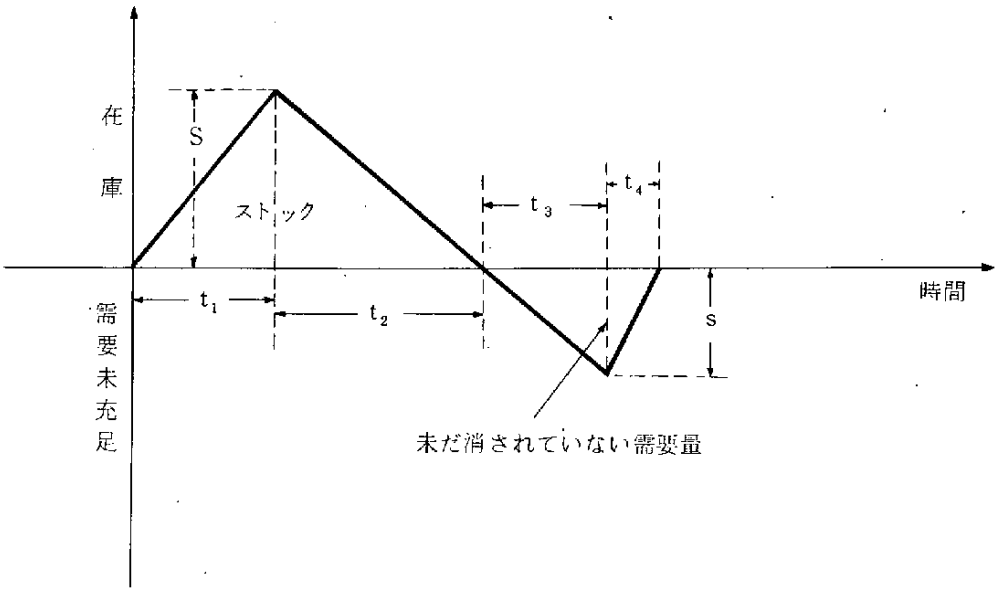


図5－2　一般化された経済発注モデル

この場合は，単位時間当り費用は，

$$K = \frac{\frac{1}{2}[C_1 S(t_1+t_2) + C_2 s (t_3+t_4)] + C_3}{t_1+t_2+t_3+t_4}$$

しかし，$S = t_1 (k-r) = t_2 r$

$s = t_3 r = t_4 (k-r)$

の関係を用いると，

$$K = \frac{\frac{1}{2} k r.(C_1 t_2^2 + C_2 t_3^2) + C_3 (k-r)}{k (t_2+t_3)}$$

となり，Kを最小とする t_2 と t_3 の値は，

$$t_2 = \sqrt{\frac{2 C_2 C_3 (1-r/k)}{r (C_1+C_2) C_1}}$$

$$t_3 = \sqrt{\frac{2 C_1 C_3 (1-r/k)}{r (C_1+C_2) C_2}}$$

これをつかうと，1サイクル当りの生産量（発注量） q は，

$$q = r(t_1+t_2+t_3+t_4) = \sqrt{\left(\frac{2rC_3}{C_1}\right)\left(\frac{1}{1-r/k}\right)\left(\frac{C_1+C_3}{C_2}\right)}$$

である。

このモデルは，かなり応用の広いものであるから，よく理解させる。

このほか，決定論的モデルでは，購入単価（あるいは生産単価）が発注量（生産量） q に依存して（値引きなどで），変化する場合がある。また，単一品目の在庫問題として取り扱うことができないで，多種品目の在庫問題で，しかも，在庫資金の総額に上限制約がある場合の問題がある。これらの問題状況は在庫管理では，往々にしてみられるので，解説することが望ましい。Ackoff と Sasieni による文献(1)を参照するのが適当であろう。

5.3 確率的モデル

需要の不確実性を無視できない場合は，確率的モデルを採用しなければならない。発注してから入荷するまでの**リード・タイム**（lead time）の間の需要を確率変数として取り扱い，その分布を考える。近似的に正規分布を考えてよいことが多い。在庫がある一定量までさがったら，一定量 q を発注する方式を**発注点法**という。ここで，**安全在庫**，安全係数の概念を説明する。このほか，**定期発注法（s—S）方式**についても学習させるのが望ましい。

5.4 オペレーショナル・ゲーム

次のような在庫管理の状況で，研修生に在庫管理者の役割を演じさせ，ゲームを行なう。

(1) 変数の定義

D（I）：第I週目の期間中の需要

P（I）：　〃　　〃　生産量

S（I）：第I週の末（第I＋1週の始め）の在庫量

これらの間には，

$$S(I)=S(I-1)+P(I)-D(I)$$

の関係がある。

(2) 費　用

(a) 在庫保管費C_1　(円／単位／週)

(b) 品切れ費C_2　(円／単位／週)

(c) 生産変更費C_3　生産レベルを週から週へ変更するとき，すなわちP（I＋1）≠P（I）ならば，C_3 円 だけ生産レベル変更準備としてかかる。

(d) 有料倉庫使用料：C_4　(円／単位／週)，会社内敷地での在庫保管容量をLとすると，S（I）がLを超えたときだけ製品1単位に対して1週間C_4 円の有料倉庫使用料がかかる。

(3) 生産レベル変更のためのリード・タイム

ある週まで保っていた生産レベルを変更しようという意思決定をしてから，それが実施に移されるまでに2週間の準備が必要であるとする。したがって，生産レベル P（I）をきめる意思決定は毎週行なうが，それは2週間先の週の生産レベルをきめていることになる。また，第I週から第I＋1週への生産レベルの変更はb単位以上はできないものとする。すなわち

$|P(I+1)-P(I)| \leq b$　とする。

(4) 需要D（I）について

各週の需要D（I）は平均値μ，標準偏差σの正規分布からサンプリングして発生させて行く。ゲームの進行中，1～2回程度，μあるいはσの値を変更する。研修生（ゲーム・プレーヤー）には，新しいμの値は知らせる。これは，季節によって需要のパラメータが変わることに相当する。

(5) オペレーショナル・ゲーミングの進め方

1年ないし2年分の在庫管理ゲームを1時間～2時間で行なうことができる。第I週末の在庫S(I)や，需要D（I）の傾向，その時点の生産レベルP（I）およびもうすでに予定されている生産レベルP（I＋1），P（I＋2）などを観察考慮して，各プレーヤーは2週先の生産レベルP（I＋3）について意思決定を行なう。そのあと，第I＋1週における需要D（I）が正規分布からのサンプリングで発生され，アナウンスされる。各プレーヤーはそれにもとずいて，S（I＋1）＝S（I）＋P（I）－D（I）を計算し，それまでに積算された総費用を計算表に記入し，またP（I＋4）を決定する。そのあと，D（I＋1）がアナウンスされる。このようなことをくり返して，在庫管理の意思決定者の役割を演じさせながらゲームを進行する。また，第27週と第28週は夏休みで生産活

動はしない。すなわち，P（27）＝P（28）＝0，をあらかじめ定めておく。在庫管理者は，第27～第28週になるまえに，ある程度の在庫を準備しておくように，配慮しなければならない。

(6) 学習検討させるべき事項

(a) 毎週末の意思決定においては，どのようなポリシーで望むべきか

(b) 費用パラメータ，C_1，C_2，C_3，C_4，やパラメータL，b，などが変更されたとき，そのポリシーはどのように調整されるべきか。（最初はbの値は10ぐらいの小さい値を採用した方がよい）

(c) 需要分布N（μ，σ）のμやσなどの変化などの予測にもとずいて，在庫管理ポリシーはどうあるべきか。（最初は$\mu=100$，$\sigma=20$ぐらい）

(d) 手計算によるゲームを一回行なったあとは，このゲーム・シミュレーションを計算機で行なうプログラムを作らせ，ポリシーの部分をサブルーチンとして，種々のポリシーをシミュレーションで検討させる。

(e) このゲームについて，各自がポリシーを確定したあと，クラス全体でコンテストを行なう。

(f) 以上の学習・検討の結果，レポートを書かせる。

指導上の留意点

(1) 在庫管理モデルは無数にあるが，ここでは，在庫管理問題の本質的構造を理解させるための代表的モデルを説明する。

(2) 解析的に分析して理解させるのは，なるべく決定論的モデルにとどめ，確率的，すなわち需要の不確実性を無視できない場合の在庫問題は，解析によるよりも，オペレーショナル・ゲームによった方が教育効果があると思われる。モデルから最適解を解析的に導くことよりも，問題そのものについての洞察を深めることが望ましい。

(3) 在庫問題を理解させるとともに，シミュレーションあるいはオペレーショナル・ゲームが経営科学の重要な手法のひとつであること，その活用の仕方などについて理解を深めさせるよう指導すること。

第 6 章　予　　測

用　語　この章では次の用語を教える。

正規方程式，時系列分析，指数平滑法，計量モデル分析，産業連関分析

目　標

経済・社会現象に関する予測手法を習得させる。データに基づく数量的手法，とくに時系列分析，単一方程式モデル，計量モデル分析（連立方程式モデル），産業連関分析がどのような状況のもとで，どんな仮定をおいて，予測につかわれるかを理解させる。これら各予測手法を比較検討させ，各手法の特徴を理解させる。

これら予測手法の原理を理解させ，与えられたデータを用いて，予測に必要な計算ができるようにする。計算結果の解釈，結果の信頼性の評価についての基礎知識を習得させる。

内　容

6.1　時系列分析

(1) 1次式による時系列モデル

最も簡単な1次式による時系列モデルの説明からはじめる。Yを時系列変数，tを時間として，

$$Y = a + bt + u$$

であらわされるのが1次式モデルである。こゝでa，bはパラメータ，uは誤差である。

そこで，最小2乗法（the least square method）の説明からはじめねばならない。そして，**正規方程式**（normal equation）が導かれる経過を説明する。例題をあたえて係数パラメータ，a，bを推定させる。

(2) 季節変動と循環変動分析

季節変動や循環変動を考慮した**時系列分析**のモデルは大別して

$$Y = T + S + C + I$$

$$Y = T \times S \times C \times I$$

の2種類がある。ここに

T：長期傾向，　S：季節変動，　C：循環変動，

I：誤差項　である。

これらのモデルのもとに，過去のデータ系列をつかってTとSを抽出する方法がいくつかある。普通は，まず長期傾向Tの要素を1次式モデルあるいは移動平均法（moving average）でとらえ，これの成分をYのデータから除去することを行なうのでこれを習得させる。

対移動平均比率法（ratio to moving average method）では移動平均における平均項数を12ケ月にとる。そのため，季節変動がとり除かれ，不規則変動も移動平均の過程でとり除かれるので，移動平均系列は傾向変動と循環変動の合成とみなされる。したがって

$$\frac{\text{原データ系列}}{\text{移動平均系列}} = \frac{T \times S \times C \times I}{T \times C} = S \times I$$

の比の系列は季節変動と不規則変動で合成されたものとみなされる。これから不規則変動を除去して，季節変動が得られる。

これらは，手計算で演習させて理解させる。循環変動については解析が最も困難であるから，省略すべきである。

(3) 指数平滑法による線形モデル

指数平滑法は，ウインター流とブラウン流があるが，季節変動のない系列について用いられるブラウン流がよく用いられるので，ここではブラウン流について説明するとよい。過去のデータでも，現時点に近い新しいデータをより高く評価するという指数平滑法の基本的考え方をよく理解させ，その原理にもとづいた計算手順とで，モデルのパラメータの推定ができるようにする。1次平滑法，2次平滑法，3次平滑法のモデルのちがいを説明する。平滑定数を調節して，過去のデータに与える重みづけをできることも理解させる。

6.2 単一方程式モデル分析（回帰分析）

重回帰分析の基礎的知識のある研修生であるかどうか確認する必要がある。重回帰分析の知識がある研修生であれば，ここでは別に新しいことを教えるものはない。

統計解析の知識が十分でない研修生であるならば，最小2乗法，正規方程式，決定係数，重相関係数，誤差項の標準偏差などの概念をよく理解させる必要があろう。また，重回帰モデルにおける基本的仮定やその仮定が成立しないときのパラメータ推定値への影響などは基礎的知識として，身につけていなければならない。誤差項の系列相関の有無についての，ダービン・ワトソン比検定などを解説する。

6.3 計量モデル（連立方程式モデル）

いくつかの変数が相互関係をもちながら，展開していくシステム・モデルである。**計量モデル分析**における基本作業を大別すると，

① モデルの作製

② モデルの係数の推定

③ 推定モデルをつかって予測を行なう。

これら三つの段階についてそれぞれ説明し理解させる。モデルについては，因果関係図を用いて例題をつかうとよい。この図をつかいながら，内生変数，先決変数，タイム・ラグつき内生変数，外生変数の概念を理解させる。

モデルの係数の推定については，あまり難しい議論はしない方がよい。重回帰分析における基本的仮

定が成立しないときでも，うまく係数を推定するための独自の統計理論があるが，研修生の能力をみて，二段階最小2乗法を教えても，よいであろう。

推定されたモデルの利用については，まず誘導形に変換の仕方とその解釈利用方法を説明する。シミュレーションによる政策のテスト・検討などの可能性について十分理解させる。

6.4 産業連関分析

産業連関分析とは経済における諸産業セクター間の依存構造をとらえて，最終需要を予測し，その波及需要の大きさから全体の生産に対する需要を予測する手法であることを理解させる。

産業部門間の取引量から技術係数を推定する経過を解説する。さらに，このモデルをつかって，予測にどのようにつかうか。種々の政策のテストや，隘路をつきとめるのにつかわれることを理解させる。

指導上の留意点

(1) 予測は難しいテーマであるが，いくつかの代表的手法を選択して，教えた方がよい。

(2) 抽象論を避けるため，やさしい数値例をたくさんつかうこと。

(3) この章で列挙した数量的分析手法のほかに，技術予測，デルファイ法，シナリオ・ライティング，関連樹木法，クロスインパクト・マトリクス法など定性的直感的予測法など最近重要を増していることも指摘されるべきである。

(4) システム・ダイナミックスやその他のシミュレーションも計量モデル・シミュレーションと同様に，予測や政策テストに用いられることも説明すること。

(5) この章の内容は，研修生の統計知識の程度をみたうえで，かなりその難易を調節して，教える必要があろう。

参考文献

(1) Ackoff , R. L. and M. W. Sasieni , 「Fundamentals of Operations Research」 , Wiley, 1968
(邦訳：松田武彦，西田俊夫「現代ORの方法」日本経営出版会)

(2) Thierauf , R.J. and R.A. Grosse, 「Decision Making Through Operations Research」 , Wiley, 1970
(邦訳：斉藤嘉博訳「オペレーションズ・リサーチ概論」，日科技連)
各章末に演習問題あり。

(3) サシーニ，ヤスパン，フリードマン，（森口繁一監訳）「オペレーションズ・リサーチ・手法と例題」，紀伊国屋書店

(4) 近藤次郎，「オペレーションズ・リサーチの手法」，日科技連，1973

(5) 佐治信男，白根礼吉，横井満，大前義次「オペレーションズ・リサーチ，理論と実際」，培風館，1963

(6) 日科技連OR演習部会編，「初等ORテキスト」，日科技連，1972

(7) Wagner, H.W., 「Principles of Operations Research With Applications to Managerial Decisions」 Prentice Hall, 1969

(8) チャーチマン,エイコー,アーノフ,森口繁一監訳「オペレーションズ・リサーチ入門」(上,下) 紀伊国屋

(9) 春日井博「ORの基礎と技法」,税務経理協会,昭和46年(1971)

(10) 守谷栄一,「詳解演習:オペレーションズ・リサーチ」 日本理工出版,昭和48年(1973)

(11) 森口繁一,「線型計画法入門」,日科技連,1957

(12) ガス,S.I.,小山昭雄訳「線型計画法」,第三版,好学社,1969

(13) 千住鎮雄,関根智明「やさしいOR教室」,日本生産性本部,1962の第13,14章

(14) タイル,ブート,クルック,関根智明訳「ORと計量経済学」,好学社,昭和43年の第3,4,5章

(15) 宮川公男,「計量経済学入門」,日経文庫,昭和41年

(16) 円山由次郎,「需要予測と時系列分析」日本生産性本部,昭和38年

システム分析・設計 (II)

目　　次

第1章　問題発見の技術 …… 172

1.1　面接調査 …… 172

1.2　資料収集と観察 …… 174

1.3　問題点発見のための分析 …… 174

第2章　システム分析の技術 …… 176

2.1　システム分析の手法 …… 176

2.2　ランド式のシステム分析 …… 178

2.3　SOP式のシステム分析 …… 178

2.4　事務分析式のシステム分析 …… 182

第3章　システムの組立て …… 185

3.1　処理内容の確認 …… 185

3.2　処理量の予測 …… 185

3.3　ターンアラウンド・タイムの確認 …… 187

3.4　処理方式の選定 …… 187

3.5　コンピュータ・システムの組立て …… 188

3.6　入力データ作成システムの組立て …… 189

3.7　出力データ配布システムの組立て …… 189

3.8　情報処理システムを組立てるさいの基準と
組立てたシステムの検討と確認 …… 191

3.9　業務の外注 …… 191

第4章　業務システムの設計 …… 194

4.1　業務処理手順の標準化 …… 194

4.2　基本手順の設計 …… 195

4.3　例外手順の設計 …… 195

4.4　移行手順の設計 …… 195

第5章　経営管理システム …… 199

5.1　経営管理システムの階層構造 …… 199

5.2　運用レベルのシステム …… 200
5.3　管理レベルのシステム …… 200
5.4　計画レベルのシステム …… 200
5.5　設計のアプローチ …… 200
第6章　システム思考 …… 202
6.1　目標システムの機能展開 …… 202
6.2　内部機能への変換 …… 202
6.3　新システムの合成 …… 203
6.4　最適システムの選定 …… 203
第7章　オンライン・システム設計 …… 204
7.1　導入準備段階 …… 204
7.2　端末装置の決め方 …… 205
7.3　機末装置の台数と回線の決め方 …… 206
7.4　ファイルの決め方 …… 207
7.5　CPUの決め方 …… 207
7.6　機種選定のしかた …… 208
第8章　システムの評価技法 …… 210
8.1　効率評価技法 …… 210
8.2　信頼性評価技法 …… 214
8.3　コスト評価技法 …… 218
第9章　各種システムの具体的設計例 …… 220
9.1　オンライン・システムの設計例 …… 220
9.2　バッチ・システムの設計例 …… 222
第10章　演　習 …… 223
10.1　グループの編成 …… 223
10.2　運用上の注意 …… 223
10.3　演習の問題 …… 223

科目「システム分析・設計(II)」

教育の目標

この科目では，研修生が情報処理システムの分析および基本設計を担当するために必要な知識を与え，またそのための能力を身につけさせることをねらいとする。

科目「システム分析設計(I)」では分析・設計のための基本的概念を教え，さらに詳細設計の方法について学ばせたのに対し，本科目では分析と概要設計についての方法を学ばせる点がちがっている。

そのため，本科目では分析の方法として問題発見の技術，およびシステム分析の技術について学ばせる。また概要設計の方法として，情報処理システムの組立て，最適業務システム，経営管理システム，およびオンライン・システムの設計について教える。さらに分析と概要設計に必要なシステム思考とシステムの評価技法についても理解させる。

システム分析と概要設計の技術は単に講義を受けたり練習問題を行なってみるだけでは完全に身につくものではなく，実際の仕事の上での経験を積むことが必要なのはいうまでもない。しかし本科目を学ぶことによって研修生はその基本的方法を理解し，独力で実際の仕事の上で分析と概要設計を進めながらその技術を身につけていくことができるようになることが期待される。

またこの科目では知識を与えるよりも，むしろ実務的な技術を身につけさせるのがねらいなので，指導に際しては理論的な体系を整然と数えることよりも，実例を多くあげたり，ディスカッションを行なったり，あるいは演習問題を多数行なわせたりするように配慮する。

またとくに分析では，創造性がきわめて大切なのと，研修生のそれを尊重し，研修生がOJTによってみずからそれを育成できるようになるためのきっかけを与えてやるようにしなければならない（演習問題には，定量的に答が明確になるものがあるが，それらについては基礎技術としてマスタさせるのは当然としても，定量的な答が明確に出ないものについては演習問題の指導上，注意しなければならない）。

時間配分

章	Aコース		Bコース	
	講義	実習	講義	実習
1. 問題発見の技術	4 時間	時間	10 時間	時間
2. システム分析の技術	8		10	
3. システムの組立て	6		6	
4. 業務システムの設計	3		3	
5. 経営管理システム	6		3	
6. システム思考	8		8	
7. オンライン・システムの設計	9		8	
8. システムの評価技法	10		6	
9. 各種システムの具体的設計例	6		6	
10. 演習		30		50
合計	60	30	60	50

第1章　問題発見の技術

用　語　この章では次の用語を教える。

面接調査，時間分析，コスト分析，クレームの分析

目　標

システム分析は現状における問題点の発見と認識から始まる。

問題点発見の方法として，面接調査と資料収集の二つの方法について学ばせる。これによって，研修生が現実に問題点の発見が行なえるようになることを目標とする。

内　容

1.1　面接調査

現状における問題点を発見する方法としては，現在のシステムに関係ある人たちと直接会って話をきくのがもっとも直接的であり，効果がある。これが**面接調査**（interview）である。

問題発見のための面接の技術として必要な四つの項目，面接計画，面接の相手，質問項目，および面接のしかたについて，次のことを説明する。

(1)　面接計画

面接を行なうに際しては，かならずその動機，ねらい，または問題意識があるはずである。さいしょにそれを確認する。

次に，その動機に関連して，だれとだれに，何を，どんなスケジュールで面接するかの計画を立てる。

スケジュールとしては，面接の期間はあまり長期間にわたって行なうよりは，短期間に集中して行なった方がよい。これは，規模にもよるが1週間から3週間程度が適当であろう。また，面接調査のほかに，資料・データ収集やそれらの総合的分析の過程があることも考慮する。

(2)　面接の相手

面接の相手としては，検討するシステムに関連ある各部門において，もっとも関係が深く，かつ大きな役割を果していると思われる人を選ぶ。地位としては，各レベルの人を含ませた方がよい。

人選にあたっては，見方が特定の立場に偏った人ばかりを選ばず，たとえば現行システムに批判的な人と好意的な人をまぜるようにする。また実際の人選には事情のくわしい人の助言をあおぐ。

(3) 面接の項目

面接調査の項目は次のとおりである。

① 背 景

その人の業務経験，専門分野，現在および以前における業務上の位置，現在の業務での位置，担当期間など。

② 以前とくらべて良くなった点

その人の判断で，現システム以前とくらべ，良くなったと思う点とその理由は何と何かを具体的にきく。

③ 以前とくらべて悪くなった点

その人の判断で，現システム以前とくらべ，悪くなったと思う点とその理由を具体的にきく。

④ 現在の問題点

その人の関係している部分での，現在問題点とその理由を具体的にきく。

⑤ 他への要望

他部門や上層部・下層部への要望事項としてどんなことがあるかを具体的に理由とともにきく。

⑥ 提 案

何か積極的な提案を持っていれば，それを具体的に理由とともにきく。

(4) 面接のしかた

面接のしかたとしては次の点に注意をする。

① 信頼感を与えること

面接のしかたでもっとも重要なことは，相手に信頼感を持たせることである。そのためには確実な紹介者またはトップの支援があるとよい。質問に際しては特定の意見はいわないようにし，良ききき手となっていろいろのことをきき出さなければならない。

② 主旨を明確にすること

面接の主旨を相手にはっきり理解させ，不要な不安感を持たせないようにする。また質問は具体的に明確に行なう。またそこで出た要望は実現されるという約束や，それに近い印象を与えてはならない。

③ 事実と意見を区別してきくこと

きき出す際に事実と意見をはっきり区別しなければならない。答は断片的なものであることが多いから，それらをつなぐために必要なこともきくようにする。そのためには単に答をきくのではなく，相手の立場になっていっしょに考える努力が必要である。とくに口数の少ない人と多い人に対しては，具体的な質問を順番に行なわなければならない。

④ 具体的な内容をきくこと

要約的，概要的なことよりも，具体的なことをきく。"たとえば？"というようにきき出すのもよい。

⑤ メモをとること

ききだしたことはできるだけくわしいメモをとり，あとで整理するために役立てる。

1.2 資料収集と観察

現行システムの状況に関する資料の収集に関し，次の点について説明する。

(1) 収集すべき資料

現行システムに直接または間接的に関係する次の資料を収集する。

- 組織図
- 業務マニュアル（作成時期を確認する）
- 手続書（作成時期を確認する）
- 帳票サンプル（記入されたもの）
- 各種統計資料
- その他現行システムの理解に役立つ資料

(2) 現行システムの理解

これは資料収集の基礎となるもので，現行システムの目標，構成，機能，データの発生から処理が終るまでの流れについて理解しておく必要がある。

(3) 観察調査

データ量および処理時間について次を調べる。

- いつ，どこで，どのくらいの入力データが発生するか。
- どんな帳票が使われるか。
- いつ，どのくらいの入力がどこからどんなかたちで入力されるか。
- それぞれの処理にはどのくらいの時間がかかるか。
- 出力は，どんなものが，いつ，どこでどのくらい出されるか。
- 入力や出力に対する時間的制約はどうなっているか。

(4) 出力の活用のしかたに関する調査

出力がどのように活用されているかについて調査する。

- 出力の種類ごとに，それがどこでどう使われているか。
- とくに重要な出力についてはくわしく調べる。
- ほとんど活用されていない出力についてはその理由を調べる。

(5) 工数と直接コストの調査

現行システムでの工数と直接コストについて次の点を調査する。

- 入力のためにかかっている工数と直接コスト
- 処理のためにかかっている工数と直接コスト
- ソフトウェアの保守のためにかかっている工数と直接コスト

1.3 問題点発見のための分析

問題点の発見こそは，すべての改善の基礎である。そのためには，改善のための強力な目的意識と改善への熱意が前提となることを理解させる。なお，ここでいう目的意識とは，あくまで現実におけるニーズから発生するものであって，アナリストのひとりよがりなものであってはならない。問題点発見のた

めの分析として次の項目を説明する。

(1) 時間分析

時間分析（time study）ではまず入力から出力までの時間を体系的に記述する。データ量の少ないとき，多いとき，および標準のときに分けてそれらをパターン化する。

システムの運用上要求されている時間的スケジュール（目標）と現実のものとの対比を行なう。差異が発見された場合，それが現実に支障がないかどうかを調べる。支障があれば，それは問題点である。

(2) コスト分析

コスト分析（cost analysis）は次のように分けて行なう。

① 開発費

現行システムを開発するのに要した総コストと，そのうち，そのシステムで負担すべきだと考えられる部分を概算する。

② 運用工数

運用のために費されている工数と，それを金額に換算したものを求める。

③ その他の直接費

運用のために必要とされるその他の直接費の概算を求める。

④ 間接費

間接費の概算を求める。

⑤ 本システムの利益

直接的利益および間接的利益を金額換算したものの概算を求める。

⑥ 利益とコストの対比

利益とコストの対比を行ない，当初の計画と比較する。

(3) トラブルとクレームの分析

① 事故とトラブル

どのような事故やトラブルがどのくらいの頻度で発生しているか，また今後の発生可能性について分析する。

② **クレームの分析**

どのような部門・地位の者からどのようなクレームが発生しているか，そのうちで本質的なものは何かを分析する。またそれらのうち，容易に解決できないものは何かを明らかにする。

(4) ま と め

以上を整理して，本質的な問題点と小さな問題点とに分けてまとめる。

指導上の留意点

(1) 演習問題として，研修生の所属会社での現行システムを適宜選ばせて，問題点を検討させるとよい。

(2) 本章はとくに実例やケースを多くし，研修生みずからに考えさせるように配慮して教えることが必要である。そのためには，研修生にまず明確な問題意識を持たせることを配慮しなければならない。

第２章　システム分析の技術

用　語　この章では次の用語を教える。

システム分析，ＳＯＰ，事務分析，代替案，費用対効果分析

目　標

代表的なシステム分析の手法とその特徴を学ばせ，それぞれにおける長所を十分に理解させて，研修生がそれらを活用し，システム分析を行なえるようにする。

内　容

2.1　システム分析の手法

システム分析の技術としてもっとも重要な次の三つについて学ばせる。

- ランド式のシステム分析
- ＳＯＰ式のシステム分析
- 事務分析式のシステム分析

これらはその対象，考え方および主なねらいが表２－１のとおり違っていることを理解させる。しかしこれらはいずれも無関係なものではなく，とくにEDP化のために今日使われているシステム分析手法は，ＳＯＰ式の流れを汲むものとはいえ，他の二つの手法の長所を取入れているので，その意味での理解をさせるように配慮する。これらについてはたとえば次のようなことを教える。

表２－１　三つのシステムの分析手法の比較

	対　　象	考　え　方	主なねらい
ランド式のシステム分析	国家・地方自治団体の政治・経済・軍事社会問題・企業の経営上の問題	真の問題の追及，それを解決するための代替案の選定，モデルを使った費用対効果分析を重視する	政策決定
SOP式のシステム分析	企業などの業務処理システム	現状分析を厳密に行なうことにより，ＥＤＰ化などの範囲と方法を明確にしていく	ＥＤＰ化
事務分析式のシステム分析	事務手順の流れ	現状分析を精密に行なうことにより，事務手順の流れの改善を行なう	事務改善

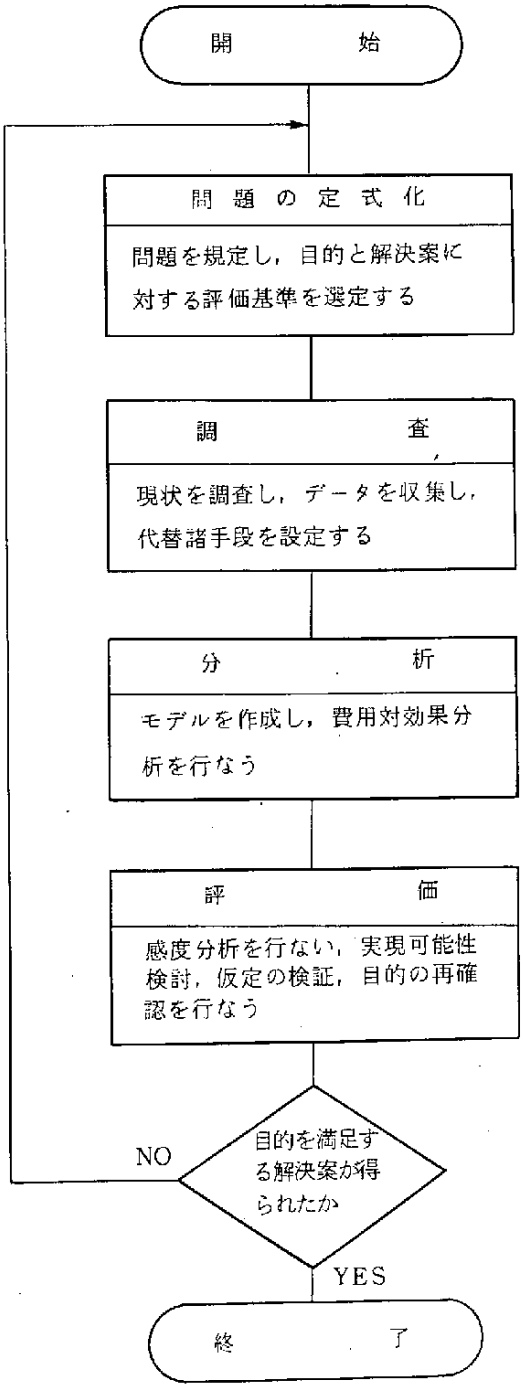


図２－１　ランド式システム分析の手順

2.2 ランド式のシステム分析

システム分析 (Systems analysis) の基本的考え方はランド (RAND Corp.) で開発されたシステム分析において確立されたと考えられる。

ランド式のシステム分析の手順は図2－1に示すとおりである。

(a) 問題の定式化

問題を明確に規定し，進まなければならない目的を明らかにし，また解決案に対する評価基準を定める。

(b) 調 査

現実を調査し，必要なデータを収集する。また問題解決のための解決案をつくり上げる。これらは**代替案** (alternative plan) として具体的にまとめる。

(c) 分 析

分析のためのモデルを作成し，**費用対効果分析** (cost －effectiveness analysis) を行なう。ここでのモデルは数学的またはコンピュータ・プログラムによるシミュレーション・モデルでなくても，もっと簡単なものでもよい。

(d) 評 価

各代替案において，パラメータ的要因が変化した場合の状況とを調べる。

各代替案の技術的可能性の程度，リスク（失敗の危険性）の検討を行なう。

こうして選んだもっとも望まし解決案に対し，それが前提としている仮定の検証を行ない，また目的の再確認を行ない，この代替案が満足であれば終了し，満足でなければ最初にもどって分析をくりかえす。ランド式のシステム分析は国家・地方自治団体の政治・経済・社会・軍事問題や，企業の経営上の問題を主対象とする政策決定を主なねらいとしている。

2.3 SOP式のシステム分析

SOP式のシステム分析は，現状調査，問題点の明確化，新システムの提案というステップで業務のEDP化を行なっていくためのシステム分析手法である。この種の分析手法はEDP化のために今日広く利用されている。

SOP (Study Organization Plan) は1956年ごろにIBMで開発されたもので，文書化のための標準用紙を準備することにより，利用者が分析を容易に行なえるように工夫されている。SOPは次の三つの段階から構成されている。

- 段階Ⅰ　分析と設計
- 段階Ⅱ　現実化作業
- 段階Ⅲ　運　用

このうち，もっとも注目すべきものは段階Ⅰて，これはさらに次の三つの局面に分けられている（図2－2）。

・局面Ⅰ　現行システムの理解

・局面Ⅱ　システム必要条件の決定

・局面Ⅲ　新システムの設計

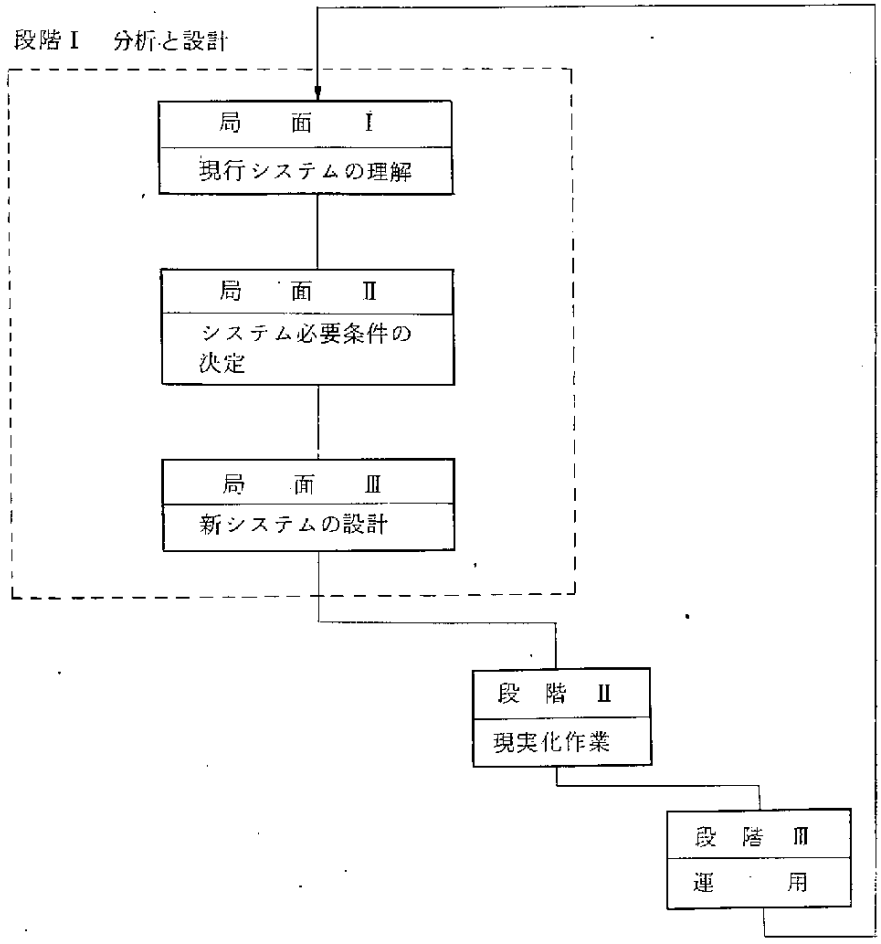


図２－２　SOPにおける手順の流れ

最初の局面Ⅰ現行システムの理解では次の三つの報告用紙が使われる。

- リソース利用状況報告用紙

  予算，マンパワーなどの利用状況を計数的に表わしたレポート

- アクティビティ報告用紙
- 仕事の処理のしかたを流れ図で表わすレポート
- オペレーション報告用紙

アクティビティ報告用紙の流れ図の各部分について，それぞれ何を入力し，何を用いて何を行ない，何を出力するかを記述するレポート

これらは図2－3に示すような相互の関係にある。

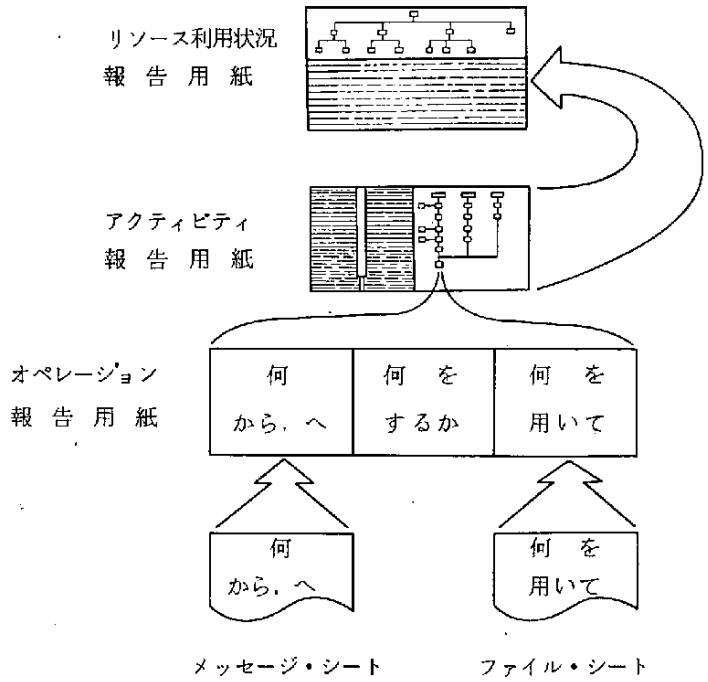


図2－3　SOPにおける三つの文書間の関係

SOPの新システムの設計の段階において用いる各種文書の関係を図2－4に示す。

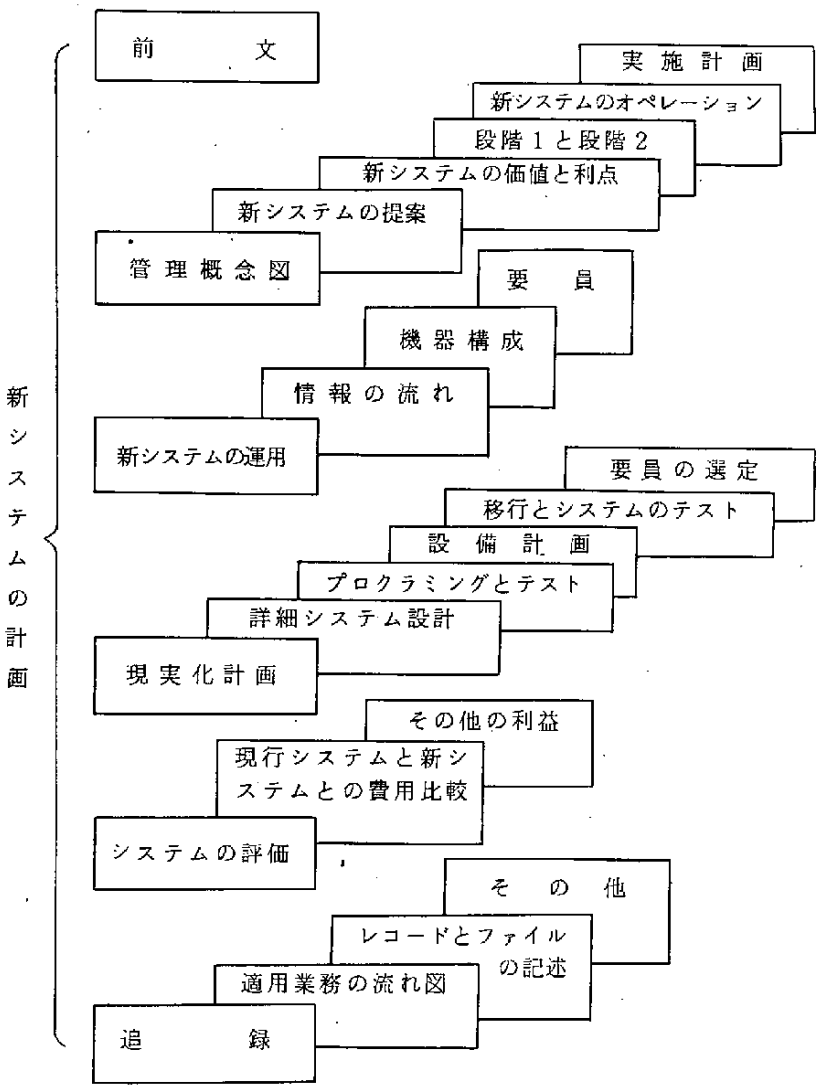


図2－4　SOPの新システムの設計の段階における各種文書の関係

2.4 事務分析式のシステム分析

事務分析はコンピュータの利用以前から，事務改善の方法を検討するために行なわれてきた（図2－5）。事務分析では事務のすすめ方を分析し，それを専用の記号を使って流れ図に書く。流れ図記号にはいろいろのものがあるが，NOMA式，日能式，および能大式が有名である。

図2－6はNOMA式の事務分析流れ図の例である。図2－7，図2－8は主な事務分析記号を示したものである。

事務分析については，それを詳細に解読したり，各方式の違いを厳密に識別したり，あるいは流れ図を完全に書けるようにする必要はない。むしろ事務分析のもっとも適した利用分野がどの範囲であるか，（すなわち位置づけ）を理解させるとともに，事務分析流れ図を見ても，記号説明を見れば容易に理解でき，拒絶反応を起こさせないように指導する。演習問題として，簡単な問題を与えて事務分析流れ図を書かせてみるとよい。

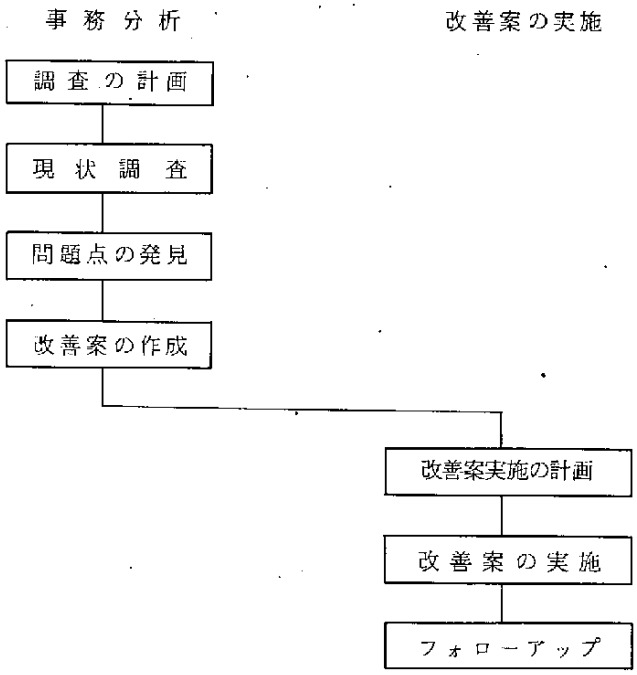


図2－5　事務改善のステップ

指導上の留意点

各手法の基本思想と，優れている点を明確に理解させるように配慮する。研修生にそれらを吸収させることにより，形式的な分析手順としてのみではなく，システム分析の思想を含めたかたちでシステム分析を身につけさせるようにする。そのために実例やケース・スタディを行なわせるのもよい。

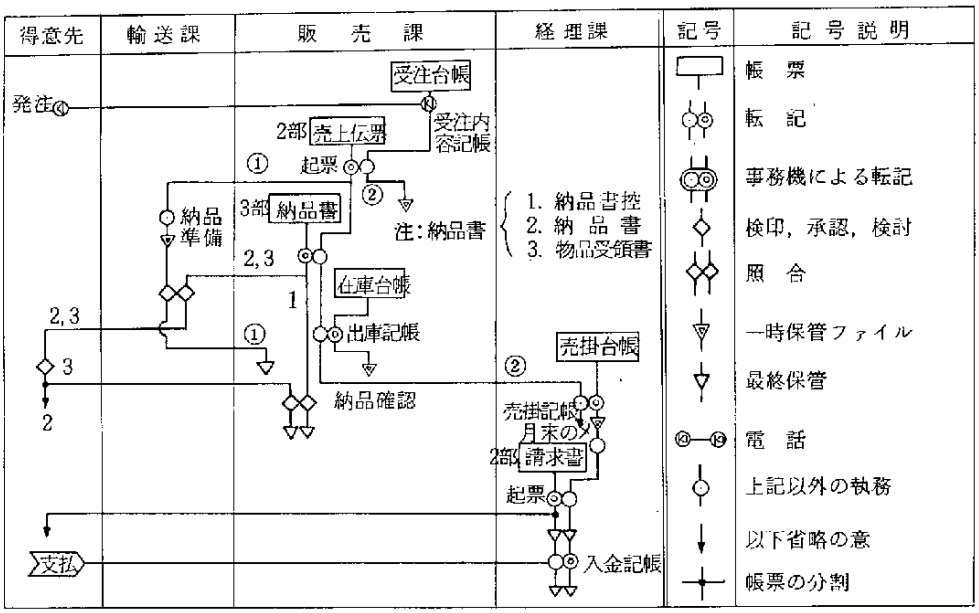


図２－６　NOMA方式の事務分析フローチャートの例

○ 作業	○ はこび	△ とめおき	□ 検査
Ⓟ 新規帳票の準備	Ⓣ 郵送する	△ 受けとる	◇ 質の検査
Ⓕ 記入する	Ⓟ 各自が運ぶ	▽ 一時保管	□ 量の検査
⊗ 集める	Ⓜ 社内便で運ぶ	▽ 永久保管	◯ 承認，決裁，押印
⊗ 化分け，分類		▽ 一時保管から取出す	── 帳票の動きの線
⊗ 2課，係以上に分けて渡す		▽ 永久保管から取出す	▬▬ 物の動きの線
○ 連絡会議			
計算する			
Ⓔ 省略			

図２－７　能大式事務分析記号

分類	記　号	記　号　説　明
帳票	複票 単票	伝票，複写枚数だけ右上にかさねて書く。索引番号，名称を枠の中に記入する。
		台帳，番号，名称を書く。
		報告書，集計表，計算用紙，番号，名称を枠の中に書く。複写する場合は，伝票と同じく右上に枚数だけかさねて書く。
現物		現品 現金，小切手，手形
フロー		帳票の流れ
		現物の流れ
		配布・分割，分岐点に黒点を書き入れる
		統合，まとめる
		流れの交差
検査		検査，内容検討・判断，検印，承認
		照合，帳票を突き合わせる
停滞		一時保管，帳票が一時停滞の状態
		最終保管，保存，処理の終了
		破棄
作業		転記，○から◎に転記する
		電話
		その他すべての作業
省略		

図2－8　いろいろな事務分析記号の例

第 3 章　システムの組立て

用　語　この章では次の用語を教える。

集中型処理，分散型処理，階層構造型処理，入力データ作成システム

目　標

情報処理システムをどのような手順で組立てるか，また各段階でどのような選択可能な方法があるかを図３－１の手順に従い理解させる。最後に開発段階での一部作業の外注の考え方についても学ばせる。これによって，研修生が簡単なシステムの選択と組立てができるようにすることを目標とする。

内　容

3.1　処理内容の確認

情報処理システムの組立てを行なうには，まず処理の目標，および処理の内容と範囲についての確認を行なわなければならない。

目標や内容の認識が誤っていては，たとえどのように優れたシステムを組立てても意味がない。一般に分析の担当者と設計の担当者は異なっており，また各種情勢の変化などで設計段階に入るときにはこれらが微妙に違ったり，あいまいになってきていることがある。そこで設計の最初のこの段階でこれらを確認し，確定することは意味がある。確認のためには，設計担当者みずからが自分の理解していることを文書化し，責任者によってそれを確認し，確定してもらうのがよい。担当者だけの勝手な解釈はもっとも危険なものである。責任者による確認は，それによって責任者自身の管理上の確認にもなり，これから始まる設計作業をスムーズに行なう上でも大きなプラスになる。

3.2　処理量の予測

処理内容と範囲が確定すれば，処理量の予測はこの段階ではむずかしいことではない。分析段階のデータを使えば，容易に行なうことができる。

処理量の予測は，入力データ量，出力データ量および処理の量について，平均時とピーク時に対して処理サイクルを考慮して行ない，またそれらの将来の増加率を考慮して推定する。処理の量については，仮に設定した特定の機種のコンピュータについての処理時間（またはCPU時間）を用いると便利である（これは後に機種が確定したときには変換する）。同様に入力データはカードに換算し，出力データはラインプリンタの印刷行数（またはページ数）に換算して行なうのが便利である。

この段階での推定値には不確定な部分があるのがふつうである。その場合には不確定さの程度に応じた安全率（ふつうは 20％またはそれ以上）を考慮して多めに算定しておく必要がある。

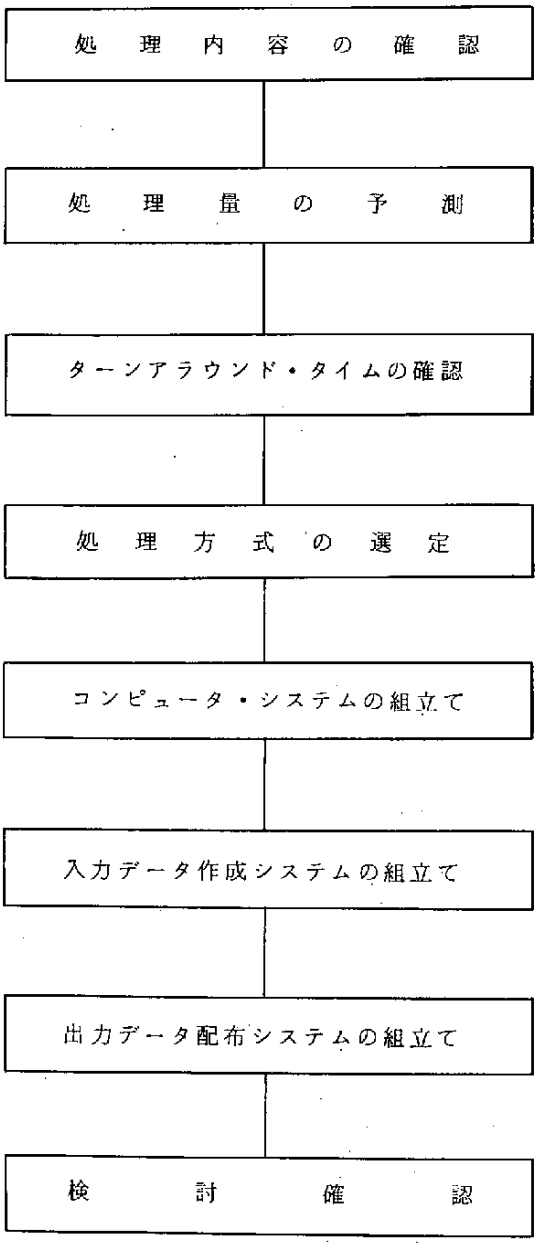


図3－1　情報処理システムを組立てる手順

3.3 ターンアラウンド・タイムの確認

ターンアラウンド・タイムは，一般に短かければ短かいほどよいのはいうまでもないが，コスト負担を考えるとほとんどの業務では結局あまり短かくすることは適当でないことがわかる。コストとの関連で考えたターンアラウンド・タイムに対する要求の程度は，基本的にはシステム分析の段階で行なわれているはずなので，ここではそれの確認を行なうことになる。これは業務処理ごとにデータ発生から処理結果受領までの時間が次のいずれなのかを明らかにすることである。

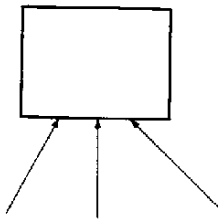
- 1分以内か
- 1時間以内か
- 半日（6時間）以内か
- 3日以内か
- 1週間以内か

またデータ発生地点と処理地点（処理予定地）とどのくらい離れているかも明らかにしておく。

3.4 処理方式の選定

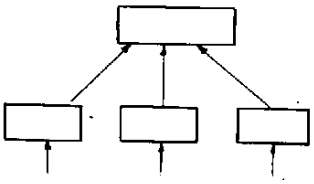
処理方式の選定とは，処理を集中処理で行なうのか分散処理で行なうのか，あるいはバッチ処理，オンライン・リアルタイム処理，またはリモート・バッチ処理のいずれで行なうのかなどを決定することである。

一般に社内の管理体制が中央集中型か分散型かによって**集中型処理**体系 がよいか，**分散型処理**体系がよいかが決まる。さらに分散型において処理の種類が多く，処理量も多くて，かつ統一管理が望まれる場合には**階層構造型処理**体系が適している（図3－2）。



(a) 中央集中型処理体系（中央集中型管理体制の場合に適している）

(b) 分散型処理体系（分散型管理体制の場合に適している）



(c) 階層構造型処理体系（分散型で，とくに処理の種類と処理量が多く，かつ統一管理が望まれる場合に適している）

図3－2　処理体系の選択

バッチ処理か，オンライン処理かなどの選択は，一般にデータ移送の方法がデータ伝送によるかどうか，および処理のしかたがバッチ処理か即時処理かなどに応じて表３－１のように行なわれる。なお，この表に関連して，各処理方式の概念を理解させる。

表３－１　バッチ処理かオンライン処理かなどの選択

データ移送の方法		処理			処理方式の選択
伝送によらない場合	データ伝送による場合	バッチ処理	即時処理		
			プログラムは共通	プログラムはそれぞれ別	
○		○			バッチ処理
○			○		インライン処理
	○	○			リモート・バッチ処理
	○		○		オンライン・リアルタイム処理
	○			○	タイムシェアリング処理

3.5　コンピュータ・システムの組立て

コンピュータ・システムの組立てについては，処理の内容と処理量により，次の各項目について決定する。

(a)　中央処理装置の速さ

処理内容が事務処理中心か，技術計算中心か，また処理量（処理時間で換算して評価する）がどのくらい必要かを定める。

(b)　主記憶容量

処理の内容と処理量，および最低必要とするソフトウェアからの条件により，必要とする主記憶容量を決定する。

(c)　直接アクセス記憶装置

処理内容とソフトウェアからの条件により，必要とする直接アクセス記憶装置の種類と容量を決定する。

(d)　磁気テープ装置

処理内容により，必要とする磁気テープ装置の種類（記録密度，速度など）および台数を決定する。

(e)　入力装置

処理内容と入力データ作成システムの種類により，入力装置の種類と台数を決定する。

(f)　出力装置

処理内容と出力配布システムの種類により，出力装置の種類と台数を決定する。

(g)　チャネルと制御装置など

補助記憶装置，入力装置および出力装置の種類と台数により，チャネルと制御装置などの種類と

数を決定する。

(h) ソフトウェア

処理内容により，使用するソフトウェアの選定を行なう。

(i) 移行のしやすさとコストなど

移行のしやすさとコストなどにより，機種選定を行なって，データ処理システムを決定する。

3.6 入力データ作成システムの組立て

入力データ作成システムとしては，次のいろいろな方式がある。

- 伝票からカードパンチを行なう方式
- 伝票からマークする方式
- 伝票から紙テープを作成する方式
- 伝票作成と同時に紙テープを作成する方式
- 伝票を直接読取る方式（OCR，MICR）
- 現場で直接マークまたは数字等の記入を行なう方式
- ターンアラウンド方式
- 伝票からキーツウテープで磁気テープを作成する方式
- 伝票からキーツウディスクで入力を行なう方式
- 媒体作成を行なってからオンラインでデータを収集する方式
- 媒体作成を行なわず直接オンラインでデータを収集する方式

これらについて，それぞれがどのようなものであり，どんな利害得失があり，またどのような場合に適するかを理解させる。

3.7 出力データ配布システムの組立て

いろいろの出力データのあらわし方および配布のしかたについて，その概念と利害得失を教え，選択についての考え方を学ばせる。

(1) 出力データのあらわし方

出力データのあらわし方としては，次の四つがある。

- 紙への印刷による出力
- 映像表示による出力
- フィルムへの記録による出力
- 機械での読取りが可能な媒体への出力

(a) 紙への印刷による出力

これはラインプリンタまたはタイプライタによる出力である。文字の種類としては，英字，数字，カナ，特殊文字がふつうであるが，漢字を用いることもある。

用紙としては，横ケイだけの入った汎用紙を用いる場合と，あらかじめ特別の様式を印刷した専用紙を用いる場合とがある。

写しを作成する方法としては次の3つの方法が考えられる。

① 重ね印刷

ワンタイムカーボンまたはノーカーボン用紙を必要な枚数（最高6枚程度まで）だけ重ねて印刷する方法。

② くりかえし印刷

出力データを磁気テープ等に入れておいて，必要な枚数だけ印刷する方法。

③ 複写機でコピーを作成する方法。

複写機で必要枚数だけのコピーを作成する。

(b) 映像表示による出力

映像表示装置によるCRTへの出力である。次の2種類の装置がある。

① 図形表示

任意の図形や文字を表示する。

② 文字表示

所定の行数，桁数の文字を表示する。文字の種類は英字，数字，カナ，および特殊文字である。

なお，これらは別の画面複写装置を用いてコピーを取ることができるのもある。

また，映像表示装置はそのまま入力装置としても使用できる。

(c) フィルムへの記録による出力

これには次の2種類がある。

① COM

Computer Output Microfilm，すなわちマイクロフィルムへの直接出力で，文字，点，線などの合成したものを高速で出力できる。

② ソフトプリンタ

漢字，英字，数字，カナ，特殊文字などの文字を肉太のかたちで高精度で巾の広いフィルムに出力し，印刷用の版として直接使えるようにするものである。

(2) 出力データの配布のしかた

出力データを配布する方法としては，次の3種類の方法がある。

① メールによる方法

社内便や郵便などにより，宛先人に送付する。

② ファックスによる方法

紙のかたちでの出力をファクシミリを用いてアナログ伝送する。この場合には映像情報でもよい。

③ データ伝送による方法

データ伝送により，宛先での端末機に送信する。

出力データのあらわし方および配布の方法の選択と組立てに際しては，その量およびそれらの実験のために要するコスト，所要時間，確実さなどを考慮して決定する。

3.8 情報処理システムを組立てるさいの基準と組立てたシステムの検討と確認

情報処理システムを組立てるさいの基準と，組立てたシステムの検討の確認の方法についての考え方を理解させる。

組立ての基準は次の四つである。

(a) 必要最小限の機能を満たしているかどうか。

必要最小限の機能をシステムが持たなければならないのはいうまでもないが，それ以上の機能はむしろコスト増の原因となるので，慎重に考えるべきである。

(b) 必要な効率を発揮できるかどうか。

必要な効率を発揮できるシステムでなければならないが，それ以上の効率はやはりコスト増の原因となる。

(c) 使いやすさ，すなわち移行のしやすさ，運用のしやすさ，および維持のしやすさも重要な要素である。また将来業務の拡張などにともなってシステムの拡張を行なわなければならないとき，それが容易に実現できることが必要である。

(d) コスト

以上の基準を満足するいくつかの案が考えられるさい，そのなかのどれを選択することがもっともコスト的に有利かを検討し，結論を出す。

3.9 業務の外注

データ処理に関連した業務の一部を外注することがある。それは次の二つの場合である。

- 社外の優れた専門的能力を利用したい場合
- ある機能を社内に固定的に保有することを避けたい場合

これらの場合に行なう外注について，主な注意事項を学ばせる。

(a) 外注の範囲と相手先

多く行なわれている外注の範囲と相手先は表3－2に示すとおりである。

(b) 外注すべきかどうかの判断基準

ある業務を外注すべきかどうかの基準について，たとえば次のようなことを説明する。

- 社内での工数が確保できるかどうか。
- 社内での専門的能力が十分かどうか。
- 社内処理よりもコスト的に有利かどうか。
- 適切な外注先があるかどうか。
- 適切な外注管理ができるかどうか。

(c) 外注に際しての注意

外注に際しては，事前に次のことを明確にしておく。

- 発注金額（または単価と計算法）
- 納　期

表3－2　主な外注の範囲と相手先

段階	作業項目	外注を行なうケース										
		1	2	3	4	5	6	7	8	9	10	11
調査分析	調査の企画											
	調査の設計	○										
	調査の実施	○	○									
	調査の集計分析	○	○	○						○	○	
システム設計	基本設計									○	○	
	詳細設計				○					○	○	
プログラム設計	内部仕様の決定				○	○				○	○	
プログラミング	コーディング				○	○	○			○	○	
	デバッギング				○	○	○			○	○	
システムテスト	システムテスト				○	○	○			○	○	
プログラミング運用	パンチ				○	○		○		○	○	
運用処理	オペレーション								○	○	○	
	計算処理										○	○
代表的な外注先		シンクタンク、調査会社	調査会社	計算センタ、ソフトウェア会社	ソフトウェア会社	ソフトウェア会社	ソフトウェア会社	パンチ会社、計算センタ	ソフトウェア会社	ソフトウェア会社	計算センタ	計算センタ

・業務範囲（明確に限定する）

・その他（作業のしかた，納入物件，機密保持責任，検収方法，延滞責任，保守期間など）

これらは契約書として文書化することが望ましい。

プログラムの外注の場合には，検収を確実に行なうべきである。

指導上の留意点

たんなる知識として教えるのではなく，手法として教えることが必要で，そのためには例やケースを多くして，研修生自身に考えさせながら教えるように配慮する。

第４章　業務システムの設計

目　標

業務システムの設計について，まず業務処理手順の標準化の重要性を学ばせる。次に基本手順の設計，例外手順の設計，および移行手順の設計のしかたについて理解させる。これによって，研修生が業務システムの設計ができるようにすることを目標とする。

内　容

4.1　業務処理手順の標準化

ＥＤＰ化以前の段階では，業務処理手順は一般にほとんど標準化されていない。手作業中心で情報の加工処理を行なっているときは，使用する情報の項目の種類，形式，精度，入手時期などが不統一で，それらに対する加工処理の内容や出力のしかたもほとんど統一されていない。入手作業の場合にはそれでも大きな支障がないからである。

それに対して，業務をＥＤＰ化するさいには処理手順の標準化が前提となる。この標準化の努力なしに行なうＥＤＰ化は成功しない。

このことの重要性を研修生に十分理解させる。できればそのような具体的事例をあげて説明する。

＜説明する具体的事例の例＞

「ある小さな会社で製品の在庫管理の機械化をしたが，期末の棚卸しの際の現物との差異が絶えず，社内で処理の信用が失われて十分な成果が得られなかった。原因は営業担当者の伝票記入の不徹底，すなわち納入商品交換，緊急納入，返品などの際の特別な場合の伝票発行が正確に行なわれていないことだった。」

データ処理の前提となる原始帳票作成段階における業務処理手順の標準化に関し，徹底しなければならないこととして，次を説明する。

- 原始帳票作成基準のルール化
  どういう場合にどの帳票をいつ作成するかをルール化し，厳守する。
- 原始帳票記入法のルール化
  記入項目ごとにどのように記入するかを徹底させる。また省略してよい項目と，してはならない項目をはっきりさせる。
- 使用するコードの徹底
  使用するコードを周知させ，正しく記入が行なわれるようにする。
- 読みやすいように記入すること。
  記入は明確に，読みやすいように行なうようにする。訂正方法も明確にする。

・原始伝票の回送順序のルール化

原始伝票の回送順序，期限を明確にし，守るようにする。

・例外事項の扱い方

標準的でないケースにおいて，原始伝票をどうするかをルール化しておく。

以上の標準化により，新人でも短期間の訓練でまちがいなく業務が行なえるようにする。

4.2 基本手順の設計

次にデータ処理の内容としての基本手順の設計について説明する。

またふつうの事務処理の標準的流れ図は図4－2のようになる。処理手順の設計に際しては，プログラム設計段階でのモジュール化による設計をも考慮して，制御の流れと各処理の内容を分けて考えるように指導する。

一般にデータ処理の内容としては，図4－1の各要素の組合わせになると考えられる。

またふつうの事務処理の標準的流れ図は図4－2のようになる。処理手順の設計に際しては，プログラム設計段階でのモジュール化による設計をも考慮して，制御の流れと各処理の内容を分けて考えるように指導する。

4.3 例外手順の設計

基本手順の設計の概要が終ったなら，それの終了する前に例外手順の設計を開始する。

例外手順の設計に関連しては，次の項目について説明し，理解させる。

(a) 特殊なケースの処理の設計

標準的でない特殊なケースの場合の手順の設計を行なう。特殊なケースとしては，厳密に考えるときりがないので，モジュール化の思想を取入れて体系的に整理し，あまり数が増えないように工夫する。

(b) バックアップ処理の設計

ハードウェア，ファイルまたはプログラムの一部に故障が生じた場合の処置を設計する。バッチ処理ではふつうチェックポイント・リスタートの方法を設計すればよい。

(c) 事前・事後処理の設計

本来の処理に先だって行なわなければならない処理，および事後に行なわなければならない処理についての設計を行なう。

4.4 移行手順の設計

移行はやりなおしが許されない。したがって移行手順は確実な方法を定めなければならないことを理解させる。

移行手順の設計に関して考えなければならないこととして，次を学ばせる。

(a) 新システムの確認

プログラムが完了してシステムが支障なく動くこと，運用マニュアルが完備していること，オペレー

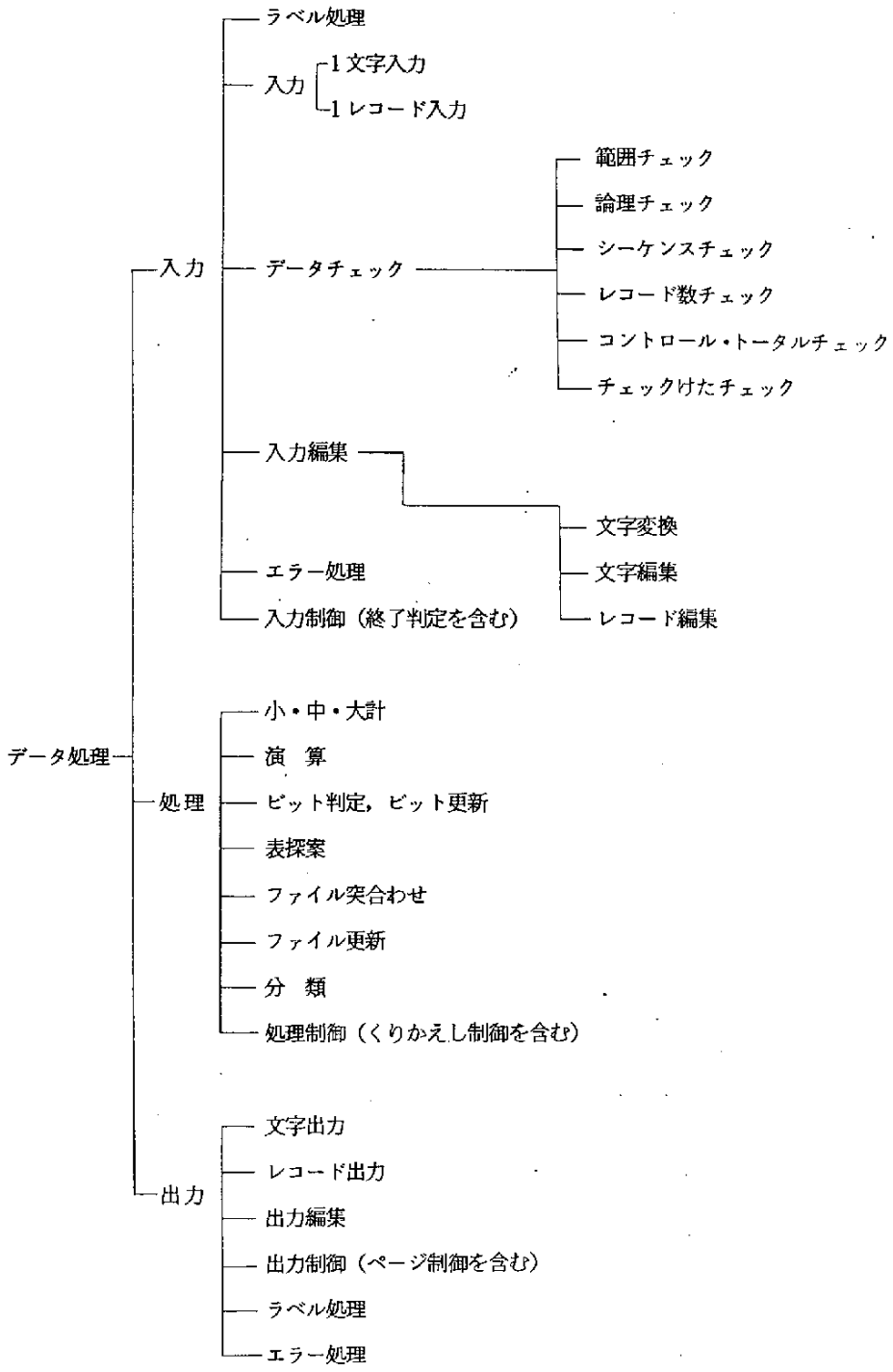


図4－1　データ処理の内容の構成要素

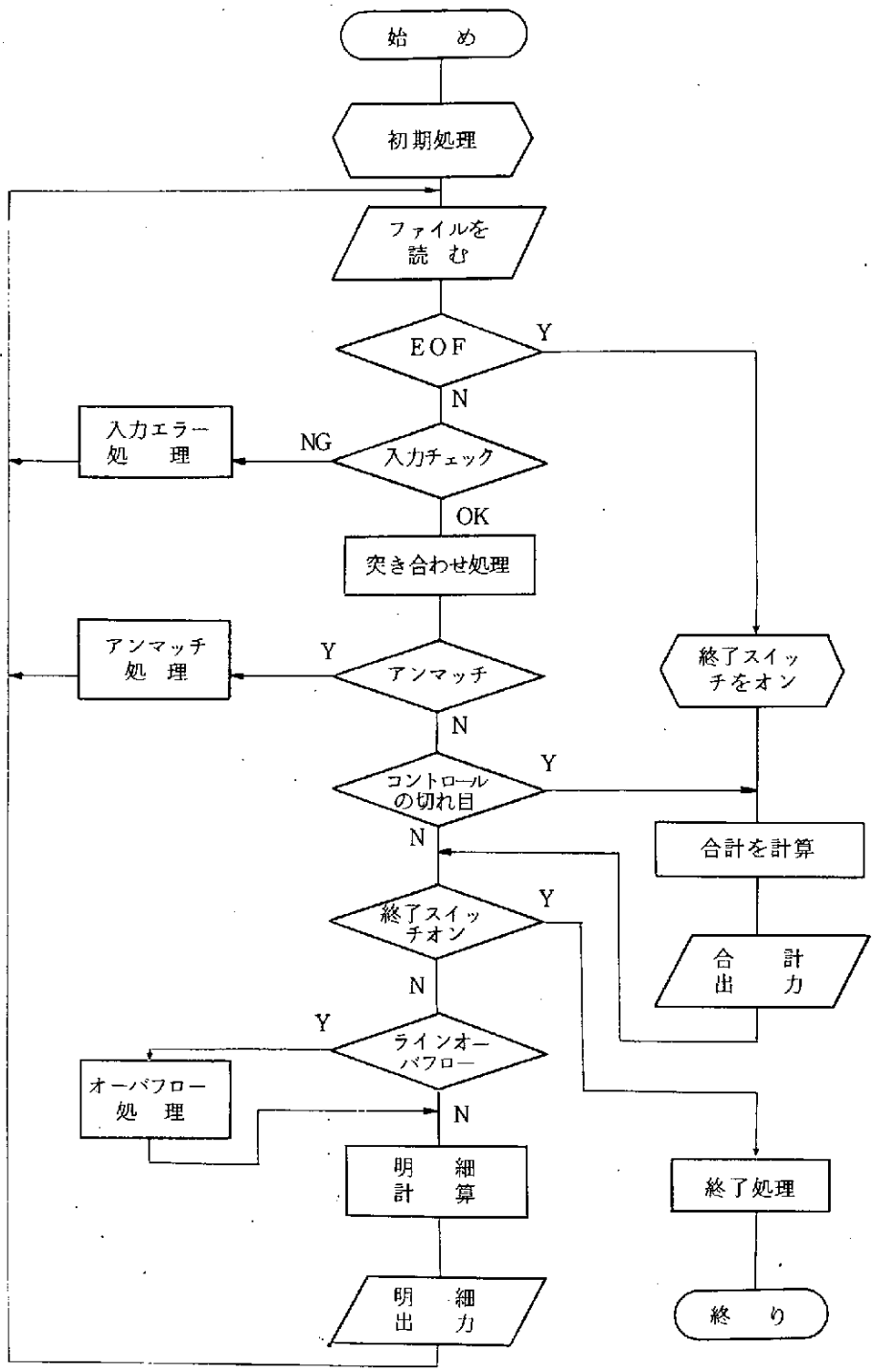


図4－2　事務計算のあらい流れ図

タなど関係者の新システムへの教育訓練がスケジュールどおりに完了することの確認を行なう。またそれらの手順を定めておく。これらのために必要な要員の条件と工数も見積る。

(b) マスタファイルの作成

新システムで使用するマスタファイルの作成はかなりの手間のかかる作業である。パンチが必要な場合にはその量と所要期間，必要なマシン使用時間，工数などを含めて，マスタファイルの作成手順を定める。

(c) 並行ランの手順とスケジュール

並行ランのための二系列の入力データをどう準備するか，二系列の作業手順はどうするか，スケジュールはどうするかなどを定める。並行ランは新システムの試行によるチェックであると同時にリハーサルという本作業による訓練という二つの目的を持っていることを忘れてはならない。

(d) 新システムのチェックとトラブル時の処置

新システムが正しく動いているかどうかをどうやって速やかに確実にチェックするか（旧新システムとどこでどう比較するか），また問題点が発見されたときどう処置をするか（修正法と作業の復元法）を定める。

(e) 新システムへの切換え

いつ，どういう手順で切換えるかを定める。

(f) 切換え後のトラブルの処置

切換え後もしばらくの間はトラブルの発生は避けられないことを想定し，それへの対処のしかたを明確にする。

指導上の留意点

知識として教えるのみでは意味がないので，具体的事例などを使いながら学ばせるように工夫する。

第５章　経営管理システム

用　語　この章では次の用語を教える。

定常業務処理，判断業務処理，意思決定システム

目　標

経営管理システムの構造とレベルについて理解させる。それによって，研修生が将来実際に担当するシステムの位置づけと，発展の方向を認識できるようにする。

内　容

5.1　経営管理システム階層構造

経営管理システムは，運用レベル，管理レベルおよび計画レベルの三つのレベルからの階層構造を持つものとして理解できる（図５－１）。これらについて，それぞれの特質を学ばせる。

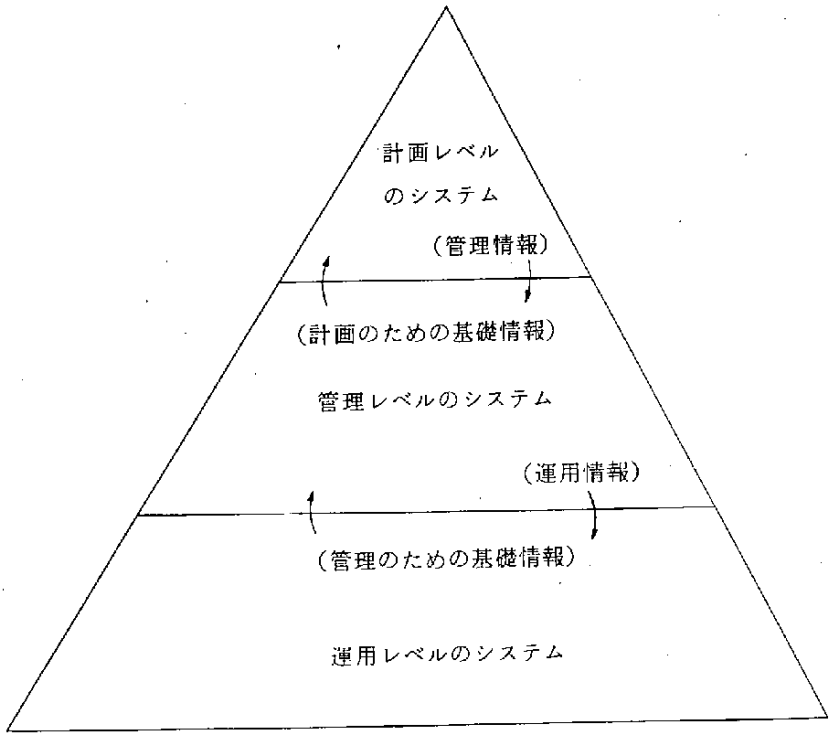


図５－１　経営管理システムの階層構造

5.2 運用レベルのシステム

運用レベルのシステムの基本的目標は，作業レベルの定常あつかい処理をいかにして低コストで，確実な業務遂行運用を行なうかにある。

運用レベルのシステムは，基礎処理または作業管理といった**定常業務処理**を行なうもので，管理または計画のために必要な情報を外部（管理システムまたは計画システム）へ報告すると同時に，このシステムの管理上または計画上の情報を外部から受取り，それにしたがって運用を行なう。

5.3 管理レベルのシステム

管理レベルのシステムの基本的目標は，いかにして想定され管理対象を効果的に管理し，短期的な利益を得るか（すなわち売上増大またはコスト低減）という内部管理の向上であり，**判断業務処理**を行なうことである。

管理レベルのシステムは管理のために必要な情報を外部（運用レベルのシステム）から受領し，また運用のために必要な管理情報を与える。さらに計画のために必要な情報を外部（計画レベルのシステム）へ報告し，また管理のために必要な計画情報を受けとる。

5.4 計画レベルのシステム

計画レベルのシステムの基本的目標は，想定された計画目標をいかにして効果的に計画し，長期的および短期的な利益を得るかということにあり，戦略的な**意思決定システム**を構成する。

計画段階のシステムが計画のために必要とする情報は外部（管理システム）から得るし，また管理のために必要な計画情報を与える。

運用レベル，管理レベルおよび計画レベルでの代表的な経営管理システムの例を表５－１にあげる。

5.5 設計のアプローチ

設計のアプローチとしては，まず運用レベルを完成させ，次に管理レベルをそして最後に計画レベルを完成させるのがよい。なぜなら，上位のシステムが効果的に動くためには，下位レベルからの基礎情報がなければならないからである。そして全レベルのシステムが完成したとき，経営情報システム（ＭＩＳ）が築きあげられる。

指導上の留意点

経営管理システムを構成している個々のレベルについて詳細に教える必要はなく，それぞれの位置づけと今後の発展の方向を与える程度で十分である。特にＢコースについては，専門編において別に用意されている科目「経営情報」で詳細に学習することになっている。

表5－1　代表的な経営管理システムの例

分　野	運用レベル	管理レベル	計画レベル
営業販売	売上伝票作成 請求書発行 問合わせへの解答	販売管理 得意先管理 売掛金管理	販売計画 製品計画
配　　送	出庫・在庫処理 配送指示，記録 問合わせへの回答	在庫管理 配送管理	
資材購買	在庫記録 発注処理 検　収	発注先管理 在庫管理 納期品質管理 買掛金管理 コスト管理	
生　　産	作業進捗管理 工数記録	工程計画 部品展開 工数集計 原価管理	生産計画
設　　備	保　守	設備計画	設備計画
経営財務	会計処理	貸借計画 損益計画 原価計算 減価償却計算 予算編成	利益計画 資金計画 損益分析
人事労務	給与計算	労務諸統計	人事計画

第6章　システム思考

用　語　この章では次の用語を教える。

機能分析

目　標

システム思考の具体的な進め方の一つとして，機能分析によるシステム設計の方法がある。これは，次の三つのステップから成立っている。

・目標システムの機能展開

・内部機能への変換

・新システムの合成

ここでは，この機能分析の考え方を理解させ研修生が具体的問題に関し，システム思考を行なえるようにすることを目標とする。

内　容

6.1　目標システムの機能展開

機能分析によるシステム設計の第1のステップが目標システムの機能展開である。最初にまず考えているシステムの基本目標が何であるかを明確にする。この目標は，あくまで機能的な観点から明確化されなければならない。基本目標は単一目標であるとはかぎらない。いくつかの異質な内容から成る複合目標のこともある。基本目標はニーズに基づいて，ユニークに設定されなければならない。

次に基本目標から始って，それを達成するために必要な副次的機能というように，機能面にのみ着目して次々に詳細機能へと展開していく。これは最終的には図6－1(a)のような木構造による機能展開図となる。

6.2　内部機能への変換

機能展開図は，ほんらい外部機能で表現されているので，これを内部機能に変換しなければならない。内部機能への変換は一義的に決まるのではなく，いろいろの代替案がありうる。これらについては，実現可能性，確実性，コストなどの要因が選択の基準となる。

この変換を図で示すと，図6－1(b)のようになる。すなわち，図6－1(a)のもっとも下位の部分に着目し，それらの集合を内部機能のもっとも下位の集合へと変換するわけである。いうまでもないことだが，この変換は，両集合要素間の1対1対応にはならない。

6.3 新システムの合成

変換された内部機能の集合を階層構造的に集約していき，全体としての新システムに対する代替案を合成する（図6－1(c)）。

6.4 最適システムの選定

このようにして合成されたいくつかの代替案のうち，全体としての実現可能性，確実性およびコストなどの面からの評価を行なって，最終的にひとつの最適システムを選定する。

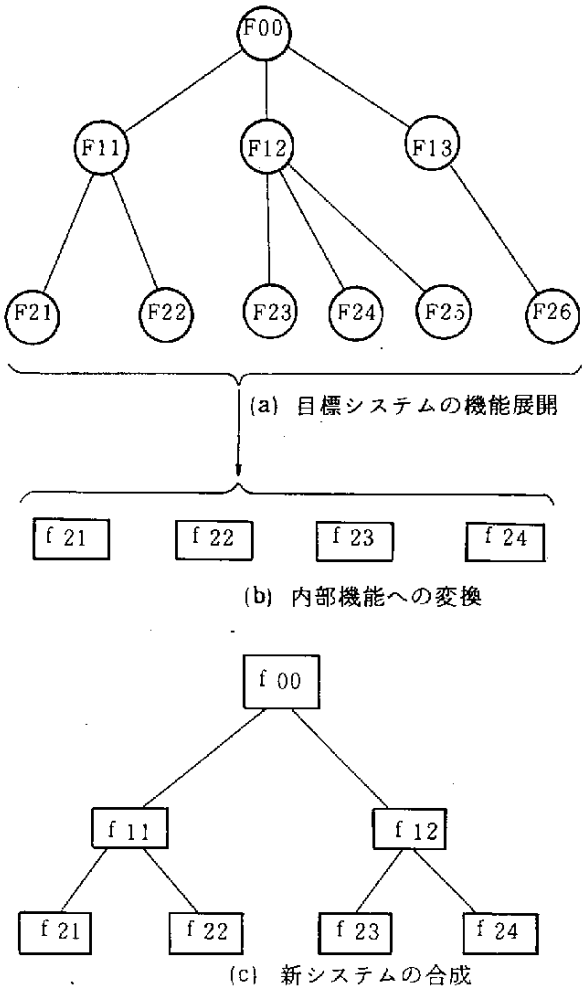


図6－1　機能分析によるシステム設計

指導上の留意点

具体的な例を使いながら学ばせることが必要である。抽象的な理論として理解させてはならない。

第7章　オンライン・システムの設計

用　語　この章では次の用語を教える。

網設計

目　標

オンライン・システムの設計のしかたについて学ばせる。そのためには業務の分析，データ量の分析の重要なことを認識させる。これによって研修生が簡単な指導のもとにオンライン・システムの設計ができるようにすることを目標とする。

内　容

7.1 導入準備段階

(1) 適用業務の範囲の決定

オンライン・システムに用いる機器を決定するには，まず適用業務の範囲を決めなければならない。

どの業務のどの範囲までをオンライン化するかである。すでにバッチ処理の行なわれているものの一部をオンライン化したり，オンライン業務のパターンがほぼ明確になっているときは，これはそれほど困難な仕事ではない。後者の例としては，標準的なオンライン・バンキング，座席予約，問合せ業務，データ収集およびメッセージ・スイッチングなどがある。

しかしながら，他に例のない新しい業務であったり，自社内でも全く新しい業務手順としていろいろな変革を行なわなければならないようなときは，準備を慎重にし，技術的にも深い分析を行なったうえで適用業務の範囲を決めなければならない。このような場合には，オンライン化の効果，技術的な成功可能性，必要なコストなどには事前に明確に予想できない要素が少なくないので企業のトップ層の理解と推進へのバックアップが不可欠である。

オンライン化の準備作業がかなり進んだあとで，業務範囲がふえたり，コスト面での制約が強く出てきたりしたのでは，そのプロジェクトは成功しなくなる危険がある。さらに技術者どうしの間で，考え方の相違を起こし不要なロスを生ずる。オンライン化をすすめるにあたっては，システム分析を十分行なって適用業務の範囲を明確にし，その予想される効果，技術的可能性，コスト，期間について文書化を行ない，トップ層の完全な認識とそれに基づく意志決定によりGOの指示を明確に出してもらうことが必要である。

オンライン化に際しては，段階的アプローチをとることが多い。これは技術，コストおよび効果という点でリスク（危険負担）を少なくすることができるので，すぐれたアプローチである。この方法をとる場合には，段階ごとに業務範囲，スケジュール，予算などを明確に改定しなければならない。

(2) 堅実な手段を選ぶこと

システム分析の段階では，基本的にできるだけ先進的で柔軟な手段をさがすことが重要だが，システム設計および導入準備段階では逆にできるだけ保守的で堅実な手段を選ぶようにすることが必要である。システム設計段階以降では，与えられたスケジュール内に，与えられたコストで仕上げなければならず，失敗はスケジュールとコストにきびしい損失を与えるからである。

そのため，機器・ソフトウェア・プログラム・運用手順のいずれをとっても，できるだけ確実なものを選ぶように心がけなければならない。

機能にあいまいさのある新型の機器,ソフトウェア・サポートのない機器,保守性や信頼性に疑問のある機器の選択に際しては，徹底的にきびしい態度をとらなければならない。そのためにはメーカに対しても，カタログや口先ではなく実績で評価しなければならない。

システムや機器の機能に対しては，何んでもできるという総花的な能力よりも，むしろ目的とする必要な機能を完全に備えたものの方を選ばなければならない。

7.2 端末装置の決め方

オンラインシステムにおける機器の決定は，端末装置から始めてCPUへというように外から内へ向かって順番に決めていかなければならない。それはオンライン・システムにおいては，センタの基本的機能は端末装置に対するサービスを行なうことにあり，センタ機器の必要能力は入ってくるメッセージの量に強く依存するからである。

端末装置を決めるに際しては,どんな機種にするかということと，どこに何台置くかという二つのことを決定しなければならない。

どんな機種にするかということについては，考えている業務においてどのように使われるのか，すなわち

- 印刷が必要か
- 紙テープ，カードなど特別な媒体を端末装置が扱う必要があるか
- オペレータは専任か
- 使用環境はどんなところか
- 1件当たりの入力メッセージの内容と量はどのくらいか
- メッセージの種類はどのくらいか
- その他特に注意すべきことは何か

などにより，たとえば表7－1のような考え方で端末装置の種類候補をしぼることができる。

使用できそうな機種候補をしぼることができたら，回線との関係，使用台数，コスト，信頼性，納期，プログラム・サポートなどの総合的判断により最も最適なものを選定する。

なお，プッシュホン電話器を使用する場合には，音声応答装置をセンタ側に備える必要がある。

オンライン・バンキングや座席予約などでは，その適用業務のために特に設計された専用端末装置を使用することが多い。メーカがそのようなものをすでに開発して持っている場合にはよいが，そうでないときには，開発コストに対する負担が大きくなってしまうため，一般用の端末装置の使用もやむを得

ない。タイプライタや文字ディスプレイ装置などは，かなりの汎用性を持っている。

適用業務によっては，1種類の端末装置を使用するよりは，2種類（以上）の端末装置を使用した方がよいこともある。特に，いくつかの業務のオンライン処理を同時に行なうときはそのようなことが多くなる。これはプログラム・サポートとコスト面で支障がなければ，別に問題はない。

表7－1　端末装置の条件

入力条件			出力条件			操作環境			
文字入力必要	特別な媒体必要	入力は多いか	印刷必要	特別な媒体必要	出力は多いか	操作員は専任か	受信管理きびしいか	環境条件きびしいか	
N	N	N	N	N	N	N	N	N	プッシュホン電話器
Y	N	N	Y	N	N,Y	N	N	N	タイプライタ
Y	N	Y	Y	N	N,Y	Y	N	N	タイプライタ
Y	N	N,Y	N	N	N,Y	N,Y	N	N	文字ディスプレイ装置
N,Y	N	N,Y	N	N	N	N	N	N	マークリーダ
N	N	N,Y	Y	N	N,Y	Y	Y	N	銀行用専用端末機など
N	N	N,Y	N,Y	N	N,Y	Y	N	Y	工場用専用端末機など
N	Y	Y	N	N	Y	Y	N	N	カードまたは紙テープ読取機とラインプリンタ

7.3 端末装置の台数と回線の決め方

端末装置の台数と回線の決定は，定量的な調査と分析に基づいて決定されなければならない。

端末装置設置場所ごとのピーク時におけるトランザクション種別ごとの，現在のトランザクション発生量を正確に調査する。次にそれらの将来の延び率を考慮し，オンライン・システムの稼動開始時およびその後の量を予測する。また入力トランザクション1件ごとのメッセージの長さ，入力所要時間，出力メッセージの長さ，出力時間などを調べる。また回線制御時間や端末装置のドキュメント・セッティング，客との応待時間など付随する必要時間も予想する。さらにレスポンス・タイム（メッセージを端末装置で入力してから出力されるまでの時間）は，どのくらいまで許されるかを決める。

こうして選定された端末装置1台当たりの単位時間（たとえば1時間当たり）のメッセージ処理能力がわかる。端末装置に対するバックアップと回線との関係から，端末装置場所ごとに必要な端末装置の台数がこうして求められる。

回線は伝送速度，使用率，レスポンス・タイム，分岐の有無などを考慮し，交換回線か専用回線かを決め，バックアップを含めてコスト的に最も有利なネットワークを決定する。これを**網設計**（net work design）という。

もし，集配線装置を使うことができ，また回線制御装置に選択可能性がある場合にはプログラミング・

サポート，信頼性およびコストなどを考慮し最適な構成を決定する。

7.4 ファイルの決め方

オンライン・システムにおけるファイルとしては，直接アクセス記憶装置が不可欠である。直接アクセス記憶装置は次のような目的で使用される。

- システム・プログラム記憶用として
- オンライン制御用として（入力メッセージの処理前の一時的保存，および出力メッセージの処理後の一時的保存用）
- 適用業務におけるマスタファイル用として

システム・プログラム用の部分はバッチの場合も変わらない。オンライン制御用，およびマスタファイルの構成と大きさを決定し，プログラム的検討とバックアップを考慮した検討とを行なって，直接アクセス記憶装置の台数とその使用割当てを決定する。

もし同時にバッチ業務などの処理を行なう場合には，そのための必要台数をこれに追加して，システム全体としての必要台数が求まる。

磁気テープ装置も，オンライン処理に必要なものと，同時に行なうバッチ処理などで必要なものとから必要台数を決定する。

最後に，オンライン時以外のときに必要とするこれら装置の台数が，これを上まわらないことを確認する。

7.5 CPUの決め方

CPUとしては，まず主記憶の大きさを決定する。主記憶の大きさは，次の要素を合計したものである（図7－1）。

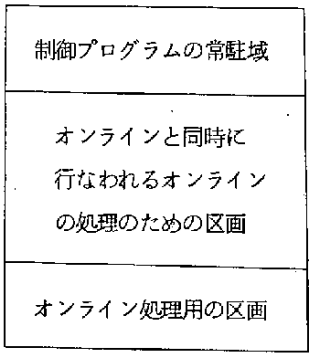


図7－1　オンライン・システムにおける主記憶の割当て

- 制御プログラムの常駐域
- オンライン処理用の区画
- オンラインと同時に行なわれるオンライン以外の処理のための区画

さらにオンライン処理用の区画としては，次の要素が考えられる（図7－2）。

- 回線制御プログラムのための領域
- 回線入出力のためのバッファ領域
- メッセージ処理プログラムのための領域

これらは機器発注の段階で概略決定しておかなければならない。オンライン・システムでは，開発がすすむにつれ，必要な主記憶の大きさがふえる傾向がある。これに対処するため，事前に予備として20%程度の大きさの領域を考えておくとよい。

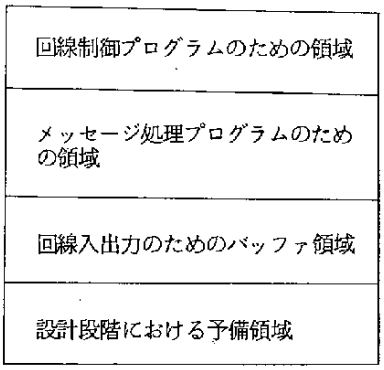


図 7－2　オンライン処理用の主記憶の区画のうちわけ

制御プログラム（スーパバイザ）に種類がいくつかあるときには，そのうちのどれを選ぶかは重要な技術的決定項目である。機能的に優れたものがよいにこしたことはないにしても，コスト面からの制約，適用業務，ハードウェアとのバランスなど考えると，一般に必要最小限のものを選ぶのがよい。

最後に，オンライン時以外の時点で必要な主記憶の大きさをチェックし，その機種でそれに最も近い上位の選択可能サイズを選定する。

次にCPUのスピードを選定する。主記憶の大きさが決まると，選択できるCPUスピードはあまりない。

CPUのスピードを決めるには，オンライン処理のために単位時間に処理しなければならないメッセージ数と，各メッセージを処理するのに要する時間を推定する。これには必要なプログラムの実行ステップ，ファイルのアクセス回数，スーパバイザのオーバヘッド時間なども調べなくてはならない。さらに行なわれるバッチなどの処理からの影響も考慮する。

オンライン・システムでは，当初の計画段階よりも処理が複雑になったり，規模が大きくなったりすることが多いので，安全率を多めにとって決めるのがよい。

7.6 機種選定のしかた

機種選定は，端末装置からCPUまでの必要条件をすべて満たすものをまず選び，さらにそのなかから最適なものを選定する。なお，端末機はCPUと必ずしも同一メーカのものである必要はないが，その場合にはインターフェイス，プログラミング・サポート，保守体制等に関し，慎重にチェックする必要がある。

最適な機種の条件としては，

- 確実に仕様どおりに動くこと
- 信頼性の高いこと
- ソフトウェアが完備していること

が不可欠である。これらを考慮したあとで，コスト面での評価を行なうべきである。オンライン・システムでは，バッチシステムとこの点が異なる。

上記3点からみても問題のある機種は，結局高くつくことになってしまうことを忘れてはならない。

指導上の留意点

(1) オンライン・システムの設計にあたっては，オンライン・システムに関する知識が不可欠となるので，基礎編の科目「コンピュータ・システム」および専門編の科目「データ通信システム」を十分習得しておく必要がある。

(2) 折にふれ，バッチ・システムとの違いに留意させながら指導する。

(3) 具体的実例を示しながら理解させるように工夫する。

第8章　システムの評価技法

用　語　この章では次の用語を教える。

ギブソンミックス法，アウエルバッハ法，GPSS，SCERT，デュアル・システム，デュプレックス・システム，負荷配分システム，時間稼動率，MTBF，MTR

目　標

定量的なシステム評価の技法として，効率，信頼性，コストの三つの方向からのアプローチが考えられる。こゝでは，このような各種のシステム評価技法について等ばせる。それによって研修生がシステムの評価を行なうためにはどんなデータを収集することが必要であり，どうすれば評価が行なえるかを身につけさせ，簡単な評価を行なえるようにすることを目標にする。

内　容

8.1　効率評価技法

効率評価技法には次のものがある。

- 統計法
- ベンチマーク法
- ハードウェア計測法
- ソフトウェア計測法
- シミュレーション法

統計法とは，そのシステムについてのデータを収集し，それらを統計的に調べて評価するものである。

ベンチマーク法とは，特定の標準問題をシステムに行なわせ，その処理時間によって評価する方法である。

ハードウェア計測法とは，ハードウェアで作成された特別の計測装置を用い，ハードウェア的な事象，たとえば入出力の回数と時間，割込みの回数などを調べてシステムの評価を行なう方法である。

ソフトウェア計測とは，ソフトウェアを使ってハードウェア計測法と同じようにいろいろの計測を行なって評価を行なうものである。

シミュレーション法とは，いろいろのシミュレーション・モデルを使い，ソフトウェアでシミュレーションを行なって評価する方法である。

(1) 統　計　法

統計法のうち，もっとも簡単なものは**ギブソンミックス法**である。これは入出力を除いたCPUの

内部演算につき，標準的な技術計算，事務計算におけるいろいろの命令比率を調べておき，今考えている機種がその比率で命令を混合したプログラムに対して，平均命令実行時間がどのくらいになるかを計算することにより，スピードを評価するものである。表8－1にギブソンミックスを示す。

表8－1　ギブソンミックス

事務用		技術用	
命令の区分	パーセント	命令の区分	パーセント
加減算	9	加減算	33.0
比較	24	乗算	0.6
移送	25	除算	0.2
分岐	31	条件つき分岐	6.5
編集	4	比較	4.0
入出力開始	7	無条件分岐	17.5
		シフト	4.6
		論理演算	1.7
		インデックス	19.0
		浮動小数点加減算	7.3
		浮動小数点乗算	4.0
		浮動小数点除算	1.6

ギブソンミックス法の第一の欠点は，機種によって命令セットがちがうので，標準の命令構成比率（すなわち表8－1）が定義しにくいし，もしこれが定義できても，特定の機種でそれをどう考えたらよいかがむずかしいことである。これらについて例示などを行ない，具体的に理解させる。

第二の問題は，事務計算あるいは技術計算といっても，個々の適用業務によって命令の構成が変るし，同一業務でも機種のくせによって比率は変わるはずである。しかし，これらは同一命令セットを持った同一ファミリ内の機種の相対評価を行なうときは支障はない。

第三はもっとも重大な問題点だが，入出力時間に対する評価がないことである。

(2) ベンチマーク法

これは科学技術計算や事務計算の典型的な問題をいくつか考えて，これを処理するために必要とされる時間を調べて比較する方法である（これは必ずしも実際に行なわせなくてもよい）。

この方法は入出力時間を考慮しているので，ギブソンミックス法よりも優れている。標準問題法でよく知られているものに**アウエルバッハ法**がある（アメリカのコンピュータ調査会社アウエルバッハ社が開発）。そこでは次の四つの標準問題を考えている。

① ファイル更新問題

マスタファイルをトランザクション・ファイルによって更新し，明細印刷を行なう。パラメータとして，カード・ベースか磁気テープ・ベースか，マスタのアクティビティ，レコードの大きさ，および

ブロッキングファクタなどを考える。 マスタのレコード数を10,000とし，全所要時間とCPU時間を考える。

② 分類問題

4本の磁気テープによる2ウェイ・マージと，6本の磁気テープによる3ウェイ・マージを考え，レコード数が10,000件の場合と1,000件の場合を考える。

③ マトリックス変換

逆行列の計算において，次数をnとすると，$(n+1)^3$ 回の浮動小数点演算時間を考える。

④ 多項の計算

データを読んで多項式の和および平方根を計算し，答をアウトプットする時間を調べる。

(3) ハードウェア計測法

ハードウェア計測法とは，CPUに特別の計測装置を接続することにより，

- CPUのラン状態
- ウェイト状態
- スーパバイザ状態
- 各チャネルの使用時間

などをわかるようにした方法である。 測定する項目は，スイッチを切換えたり，接続法を変えたりすることにより，いろいろ変更できる。カウンタには電子的なものと機械的なものとがある。記録は磁気テープや紙テープにとったり，写真にとったりする。

(4) ソフトウェア計測法

ソフトウェア計測法は，ハードウェアで行なったと同じような機能をソフトウェアで行なうもので，オペレーティング・システムのスーパバイザに出口を設けておいて，いろいろの事象を時間経過の順に磁気テープなどに記録する方法である。

これによってたとえば，

- ジョブやジョブステップの開始・終了
- ファイルのオープン・クローズ
- ファイルへの参照
- 各タスクのウェイト

についての統計データを収集するものである。

ハードウェア計測法やソフトウェア計測法で収集したこれらのデータは，ユーザが自分のジョブを効率よく動かすために機器構成を改善したり，オペレーティング・システムの使い方（主記憶，ファイルの割当て，ジョブのスケジューリングなど）を改良するのに用いることができるし，別に作成するシミュレーション・モデルの入力データとして使用することができる。またオペレーティング・システムやハードウェアの技術者が設計や改善のために用いることもできる。

(5) シミュレーション法

シミュレーション法は，ソフトウェアでシミュレーション・モデルを使い，システムを評価する方法である。

システム評価を行なうためのシミュレーションのための言語としては，**GPSS** (General Purpose Simulation System) が有名である。これは乱数発生ルーチンを使い，指定した時間分布にしたがってトランザクションを発生させ，それを待合わせ行列を持つサービス場所を合成して作ったモデルのなかを通過させ，定常状態の待合わせの数や待合わせ時間などを求めようというものである。GPSSの使用に際しては，知ろうとする目的に沿ったモデルをいかに作るかがかぎになる。

シミュレーション法の具体的例としてたとえば SCERTなどのシステム詳細プログラムについて説明した方がよい。

COMRESS社のシステム評価プログラム，**SCERT** (Systems and Computers Evaluation and Review Technique) は，コンピュータに対する本格的なシステム評価を行なうためのプログラム・パッケージとして有名である。SCERTを使用する場合には，ユーザは処理したい業務と開発業務の合計の仕事量，および想定しているコンピュータの機器構成を所定の用紙に詳細に記入する。出力としては，ランの分析，各装置の使用状況，機器のコスト，マルチプログラムラミングのためのラン・スケジュールなどが出力される。

(6) マルチプログラミングの評価法

マルチプログラミングは，主記憶装置やファイルのスペースに余裕があり，しかもマルチプログラミングをしようとしているタスクのうち，1つを除いてはCPUをあまり使いすぎないような場合で，かつハードウェアやオペレーティング・システムがそのような機能を持っている場合，有効である。

コンパイル・実行のようなジョブは，CPU，主記憶，チャンネル，ファイルをかなりはげしく使うのがふつうなので，大型機（IBMシステム360モデル65クラス）でも，3つ同時に走らせることは実用的ではない。中型機（IBMシステム360モデル40クラス）では一つが限界である。

単独で走らせた場合，CPUを 33％使用し，残りは入出力であるような二つのタスクをマルチプログラミングで走らせたとする。このときの両タスクの実行時間の遅れは，数学的なモデルを使って解くことができる。結果だけを示すと，CPUの利用については優先順位によって決まってしまうので，入出力の競合により，表8－2のようになる。これは実験的にも確かめられている。

表8－2　マルチプログラミングによる実行時間の遅れ

	入出力の競合の程度		
	なし	中程度	きわめて強い
優先順位の高いタスク	1.0	1.2	1.4
優先順位の低いタスク	1.2	1.4	1.7

8.2 信頼性評価技法

(1) フェイル・セイフとフェイル・ソフト

システムの障害に対する考え方としてのフェイル・セイフとフェイル・ソフトの考え方を説明する。

フェイル・セイフ（Fail Safe）とは，システムの一部が故障してもシステムの機能はなんら低下しないで持続するようにした方式である。これに対し，フェイル・ソフト（Fail Soft）とは，システムの一部が故障すると，機能が一部低下するが，しかし完全に止まることはないようにする方式である。当然のことながら，フェイル・セイフの方が望ましいが，コストがかかるのでやたらには使えない。

信頼性を高めるためのシステムの構成法としてのデュアル・システム，デュプレックス・システムおよび負荷配分方式の三つについて説明する。

バックアップのまったくないシステムをシンプレックス・システム（Simplex system）という。シンプレックス・システムは，一つのサブシステムそのままでシステムを構成する方法である（図8－1(a)）。シンプレックス・システムは，バックアップを必要とするほどの高い信頼性が要求されないときに使用される。

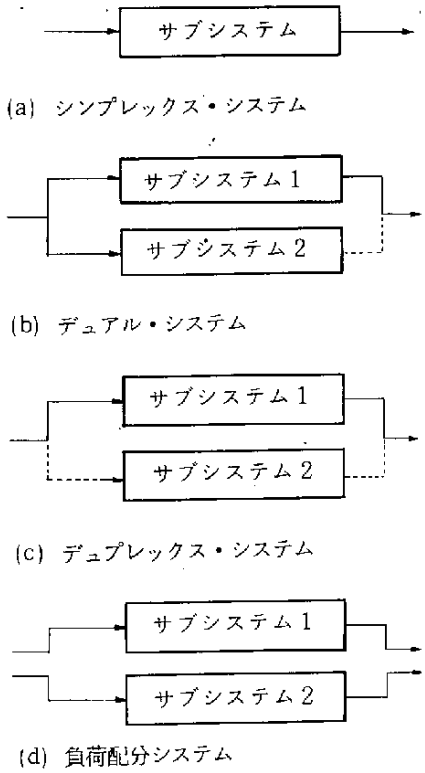


図8－1　サブシステムからシステムを構成する方法

デュアル・システム(Dual system)とは，二つのサブシステムで同じことを同時に実行させる方式である（図8－1(b)）。この場合には，一方が故障しても他方のみによって完全に処理が継続できるので，フェイルセイフ・システムである。

デュプレックス・システム(Duplex system) とは，まったく同じようなサブシステムを二つ並列に置いておき，平常時はどちらか一方によって処理を行なっておき，故障時に切換えを行なう方式である（図8－3(c)）。切換えが行なわれないときは，待機サブシステムは別のことをしていてもよいが，切換え時には中断できなければならない。これもフェイルセイフ・システムであるが，切換え時に処理不能のある短い時間がある。

負荷配分システム（Loadsharing system）とは，ふだんは二つのサブシステムの間で負荷を配分しておき，一方が故障した場合その部分はそのままにして作業を続行するという，フェイルソフト方式のシステムである（図8－1(d)）。

(2) MTBFと時間稼動率

システムの信頼性を定量的に評価するための尺度としてのMTBF時間稼動率について説明する。

MTBF（Mean Time Between Failure）とは，システムが平均何時間ごとに故障するかという統計的な時間間隔のことである。

時間稼動率（Availability）とは，そのシステムが使用可能であった時間を，使用予定時間で割ったものを十分長い時間について考えたものである。すなわち，時間稼動率Aは次式で表わされる。

$$A = \frac{T_U}{T_S} = \frac{T_S - T_D}{T_S}$$

$$= \frac{MTBF - MTR}{MTBF}$$

$$= \frac{MTF}{MTBF}$$

$$= \frac{MTF}{MTF + MTR}$$

ここで，T_S はシステムが稼動を予定された時間

T_U は T_S のうちで正常に稼動した時間

T_D は T_S のうちで正常に稼動しなかった時間

MTBFはシステムが次に故障するまでの時間間隔の平均

MTF（Mean Time to Failure）は稼動していた時間の平均

MTR（Mean Time to Repair）は修理に要した時間の平均

である。T_S，T_U，T_D がある所定期間（たとえば1年）を考えるのに対し，MTBF，MTF，MTRは1回故障するまでを考える点が違っているが，状態の変らないシステムについて，十分長い時間を考えれば両者は一致する（図8－2）。

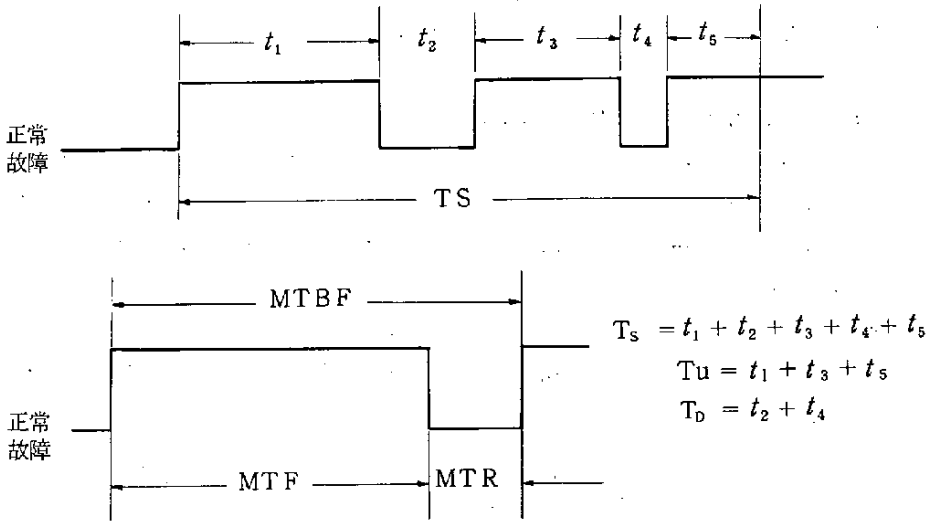


図８－２　MTBF，MTFおよびMTR

＜例　１＞

たとえば，１年間のうち，システムの稼動を予定した7,200時間（T_S）のうち，6,840時間だけ正常稼働（T_U）したとすると，

$$T = \frac{T_U}{T_S} = \frac{6,840}{7,200} = 0.95$$

となる。また平均190時間ごとに故障し，修理に平均10時間かかったとすると，

$$T = \frac{MTF}{MTF+MTR} = \frac{190}{200} = 0.95$$

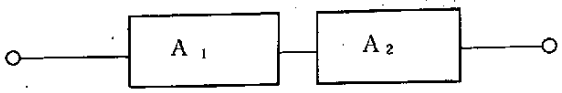
となる。

時間稼動率Aの意味は，このシステムの稼動予定時間については，Aという割合だけ正常に稼動するということである。

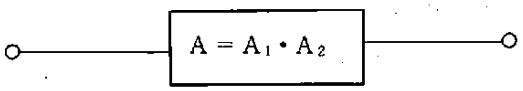
さて，つぎに二つのサブシステムを結合したシステムの時間稼動率を考えてみよう。まず直列システムすなわち，双方とも正常に稼動しているときのみ全体が稼動する場合（図８－3(a)）では，両サブシステムの時間稼動率をA_1，A_2とすると，全体の時間稼動率Aは，

$$A = A_1 \cdot A_2$$

となる（図８－３(b)）。



(a) 直列システム



(b) 等価なシステム

図８－３　直列システムの時間稼動率

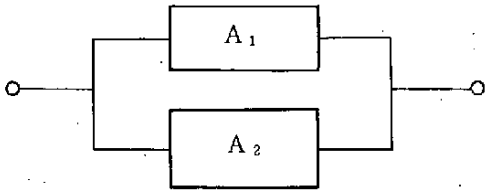
＜例　２＞

たとえばＣＰＵ，ファイル装置の時間稼動率がそれぞれ0.９５であるような場合には，両者を合わせたシステムでは，

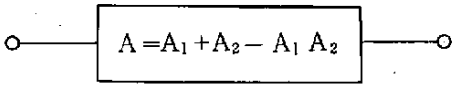
$$A = A1 \cdot A2 = 0.95^2 = 0.9025$$

である。

つぎに並列システム，すなわちどちらか一方が稼動していれば全体が稼動するようなシステムについて考えよう（図８－4(a)）。この場合には，



(a) 並列システム



(b) 等価なシステム

図８－４　並列システムの時間稼動率

各サブシステムが故障したときのみ全体が故障することから，

$$1-A=(1-A1)\cdot(1-A2)$$

すなわち

$$A-A1+A2-A1A2$$

となる（図8－4(b)）。

＜例　3＞

たとえば，A1＝A2＝0.95であるような二つのファイルからなる並列システムでは，

$$A=0.95+0.95-0.9025=0.9975$$

となる。

なお，ここでいう直列・並列は実際の接続法とは関係ない，システム全体が稼動するために，いずれも必要かそれともどちらかだけでよいか，で定まる点に注意しなければならない。

以上は二つのサブシステムからなる場合であるが，これらは容易にn個のサブシステムからなるシステム，および直列・並列の組合わされたシステムに拡張して考えることは容易である。

＜例　4＞

m個の同一サブシステムを並列にしたものをさらにn個直列にした場合の時間稼動率を表わす式を考えよ。

時間稼動率を構成するMTRは，主として人間系によって定まることは注目すべきことである。また実際に故障から回復するには，ハードウェアのMTRのほかにソフトウェアやアプリケーション側での，リカバリーが必要である。

高信頼性を要求するときは時間稼動率だけでなく，故障の回数そのものが問題になることがある。この場合にはMTBFも評価の対象となる。

8.3 コスト評価技法

システム分析では，コストをある目的達成のため，別の目的達成に使えなくなるような資源の量であると定義している。この立場では，いろいろの機器のうち，量に制約のあるもの，たとえばコンピュータ利用時間，人材，目標達成までの時間などもコストと考えることができる。これらは，最終的には金額に換算することができる。

コスト評価を行なうための，開発に要したコストを償却するためのコストと，運用コストとの両方について理解させる。

開発費の償却には，初期開発費用とレベルアップの費用とがある。定額償却の場合には，システム・ライフ期間でこれに金利を加えたものを割り，期間当りの償却費用が得られる。

開発コストの構成要素として大きいものは，システム設計者，プログラムなどの開発要員の人件費およびプログラム開発のための機械使用料である。一般にオンライン・システム機能が複雑で制約条件が多いため，技術的にもむずかしいし，期間も余計にかかり，バッチ処理に比べてはるかに多くの開発

コストがかかる。

運用コストには，コンピュータ・レンタル，回線使用料，端末装置レンタル（または償却費），センタおよび端末装置の操作員，人件費，電力，空調，部屋代などがある。

コスト評価の方法としては，まずある業務当たりのコストを考えることができるが，これはその業務を以前他の方法で行なっていたなどの理由で，比較の基準をもっている場合に使えるものである。

オンライン・システムでは，1メッセージ当たり，その処理にいくらかかるかということでコスト評価を行なうことができる。これはバッチシステムでも，計算センタなどで，たとえば月次給与計算を1人150円で行なうなどというように使われることがある。

メッセージ1件当たりいくらかかるかは，1カ月に平均何件のメッセージを処理し，1カ月にとのくらいのコストがかかるかがわかれば，割算をすることによって容易に算出することができる。たとえば，償却費と月間運用維持費の合計が1,000万円の中型システムが，月間100万件のメッセージを処理するとすれば，1メッセージ当たりは10円ということになる。もし，同時にバッチ処理などの別業務を行なうさいは，コストからその分を除外しなければならない。

オンライン・システムでは，その運用時間はメッセージが入力されてもされなくても，主記憶やディスクなどは占有されてしまうので，たとえばコンピュータ利用時間が空いていても，それを100％有効に活用することはできない。いいかえれば，かなりの固定費がかかってしまう。したがって処理するメッセージの総量が少ないと，メッセージ当たりのコストはそれだけ増えてしまう。これは待機コストがきわめて大きいということである。

オンライン・システムでは，単位時間あたりの入力メッセージの量の変動が激しいので，ピーク時を想定してシステムを設計すると，驚くほど大きなコストがかかってしまう。

もし，オンライン処理のコストがメッセージ量によってのみ決まる場合，すなわち外部のタイムシェアリング・センタを使って従量制の料金(固定費なし）でオンライン処理を行なう場合には，たとえメッセージ当たりの単価がある程度高くても，メッセージ量がある値以下では安くなる。

指導上の留意点

定量的な評価を行なうためには，定量的なデータの収集が不可欠である。これについても説明を加えて研修生が自分で評価が行なえるように工夫する。

第９章　各種システムの具体的設計例

目　標

システム設計を行なう場合，これと類似の他のシステムについての実例が役立つことが多い。そこで，こゝでは，各種のシステムについて具体的な設計例を紹介し，研修生がその考え方や具体的な方法等についての知識を習得して実際にこれらを応用できるようにさせる。

内　容

9.1　オンライン・システムの設計例

最初にまず適用業務の範囲であるが，これは現実の分析が中心である。ここではこの部分は省略して，結論だけを述べておく。オンライン化の業務範囲として，販売管理・在庫管理を取り扱うことにする。

各営業所では売上げを端末装置から入力し，在庫状況を調べて倉庫から出庫するか，あるいは製造命令を出すかする。製造には平均５日程度しかかからない。在庫品については倉庫に出庫指命が出され，また倉庫からの出庫・発送入力により在庫マスタ が更新される。

確実性と開発コストの点を考慮し，プログラミング労力がわずかで確実なソフトウェアとハードウェアを持つ機種を選ぶことを決めた。

端末装置は，業務の性質からハード・コピーがとれ，キーボードを持つものということで，タイプライタ形式のものを採用する。

入力メッセージの量は，今後１年間次のように予測されている。

営業所ａ	１時間に最高	150 件（市内）
営業所ｂ		80 件（市外）
営業所ｃ		40 件（市外）
倉　庫ｘ		200 件（市内）
倉　庫ｙ		200 件（市外）

営業所からの入力は，１件当たり30秒以下，倉庫からのものは15秒以下ですべて終了する（関連する出力を含めて）。

必要な端末機の台数は，以上のことから当初は営業所ａは２台，他はすべて１台ずつでよい。またセンタ端末装置を１台センタ内に置く。なお２年目以降のデータの増加量はそれほど急激ではない（20%以下）と予想されている。回線は 200 bps のものを使用する。

ファイルは磁気ディスク・パックを２台使用する。

オンライン処理と同時に単純な事務計算が行なえることが望ましい。この場合，主記憶容量40Kバイト，磁気テープ２台，カードリーダおよびプリンタは必要とする。オンライン処理のためにはＣＰＵはあまり使われない。

機種はソフトウェア・ハードウェア，信頼性，サービス，価格などを総合評価してA社のモデルBと決めた。

必要な主記憶の大きさは，オンラインのユーザエリアは計算の結果30Kバイト使うことが明らかになった。スーパバイザのためのエリアは，オンライン制御を含め，20Kバイト必要である。

以上から，結局最終機器構成は（図9－1）のようになる。

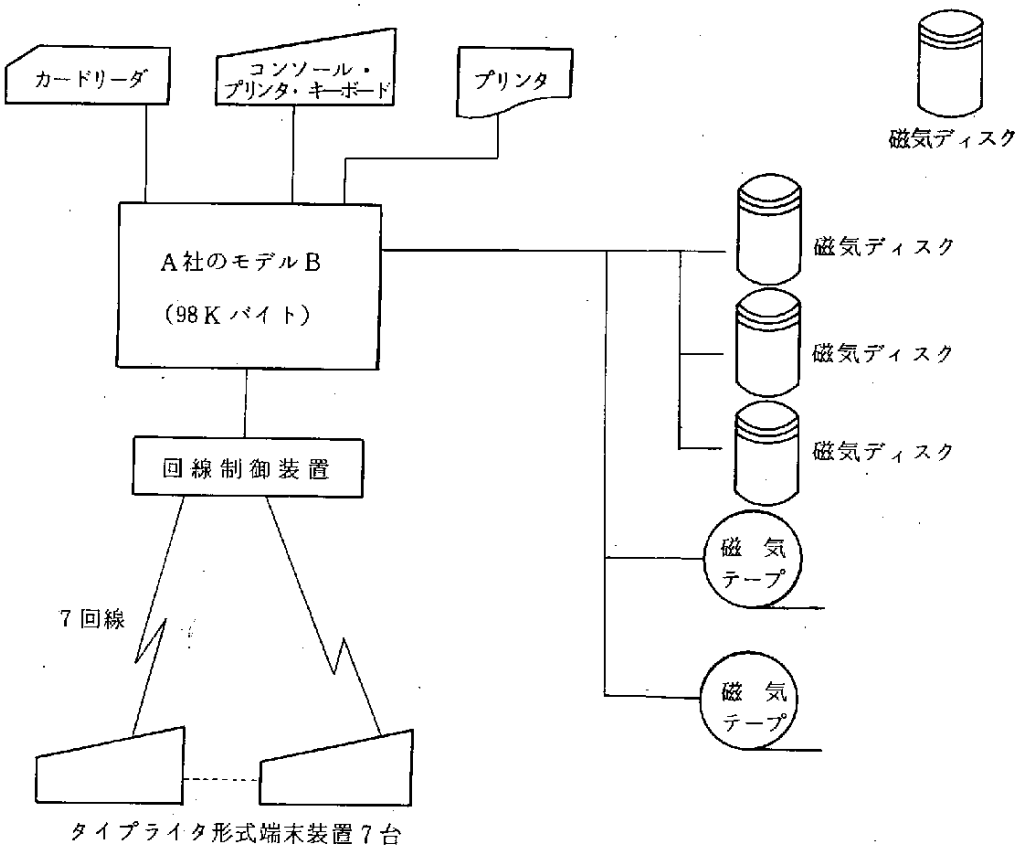


図9－1　設計されたシステムの機器構成

なお，ここでは回線の分岐は行なわないと考えている。また機器の故障時にはオンライン処理を停止し，回復してから再開することにしてある。

ソフトウェアの設計は回線制御プログラムおよびメッセージ処理プログラムについて行なう。回線制御プログラムは，使用するOS，機器構成，メッセージの入出力制御のしかたなどによって設計するが，詳細はここでは省略する。

このほか，国鉄のMARS，NHKのTOPICS，日本航空のJALCOMⅡなどのシステムについて紹介をしてもよい。ただし，これらは広く浅く行なうよりも，むしろ狭く深く行なうのがよい。

9.2 バッチ・システムの設計例

（省略）。

指導上の留意点

(1) ここでは現実的な具体的な条件を与えてシステムの設計を示すことが望ましい。十分な理解をさせるには，具体的なデータがたくさんあればあるほどよいが，しかしあまり細かいデータを与えすぎると焦点がぼけるので注意しなければならない。

ここでオンライン・システムの設計例を示したが，更にバッチシステムの設計例も示すとよい。

(2) コースの違い（AまたはB）によって取上げる事例を考慮すること。

第10章　演　　習

目　標

演習を行なうことによって，本科目で履習した内容の定着化をはかることと，これらを基に実際の問題解決のための応用力を身につけさせることを目標とする。

内　容

次のような問題を作成して研修生に与え，演習させる。演習はグループ方式で行ない，最初に十分なオリエンテーションを行なってから，途中は自主的に行なわせ，最後に発表会を行なって，全員でディスカッションさせるのがよい。

10.1　グループの編成

1チーム3～5名とする。メンバーはなるべくランダムとする（同一のところからの出席者や同年令・同学歴のもの，仲のよいものや反目し合うものはなるべくグループを分ける）。各メンバーの能力はほぼ均等になるようにする。まとめ役を置き教師との連絡役をさせる。

10.2　運用上の注意

当初のオリエンテーションで問題を与え，目標（課題）を明確に理解させる。

作業中は，作業のための部屋を1チーム一つづつ用意する。時間外の作業も自由にさせる。

作業グループによるプロジェクト作業の演習のつもりで指導する。作業内容については教師は途中ではなるべく意見をはさまない。進行状況だけを報告させ，チェックする。特定の者だけが作業をして他が遊ぶということがないように十分注意する。

最後には提案書（など）を提出させ，その概要を大きな紙を使って発表させる。時間は1チーム30分程度。

発表会では目標と矛盾しないかぎり，自由に発言させる。個人プレイや感情的な発言や行動はさせないようにする。

最後に教師からの評価を行ない，論評を行なう。教師が2人以上いるときは，発表会には全員出席した方がよい。

10.3　演習の問題

次の項目を含む具体的な条件とデータを与え，提案書を作成させる。

会社の概要

現在の問題点と目標

対象業務

現行システム

各種データ量

チーム数が多い場合は2～3チーム程度ごとに異なる問題を与えるようにする。

指導上の留意点

(1) あまり大きな問題を与えすぎないように工夫するが，逆に単純すぎる問題では効果が期待できないので困る。

(2) コースの違い（AまたはB）によって，取上げるテーマを考慮すること。

参 考 文 献

(1) Quade, E.S他,「Systems Analysis and Policy Planning」1968, American Elsevier

(2) 福島康人「システムズ・アナリシス」昭和45年，タイヤモンド社

(3) グランズ他著，鵜沢監訳,「マネジメント・システム」昭和45年，日本経営出版社

(4) 吉原賢治「インプット・システム設計改善のポイント」事務管理，1972年8月号，日刊工業新聞社

(5) 情報処理学会編,「情報処理ハンドブック」昭和47年，9,000円

(6) 三浦大亮編「システムの分析と設計(1), (2)」昭和48年，オーム社, 1,600円，1,400円

(7) 岩城三郎「コンピュータシステム概説」昭和47年，オーム社 1,500円

コンピュータ導入計画

目　　次

第1章　コンピュータ導入計画のたて方 …… 226
1.1　計画段階 …… 226
1.2　実施段階 …… 229
1.3　評価段階 …… 230
第2章　導入組織と人員計画 …… 231
2.1　導入組織の種類 …… 231
2.2　導入組織の位置づけ …… 232
2.3　人員計画 …… 232
第3章　教育計画 …… 234
3.1　教育の必要性と目的 …… 234
3.2　教育の性格 …… 234
3.3　教育の対象 …… 234
3.4　コースの選定 …… 235
3.5　教育体制の確立 …… 235
第4章　機械化計画とプログラム開発計画 …… 236
4.1　計画のたて方と作業管理 …… 236
第5章　設備計画 …… 238
5.1　コンピュータ室の設備計画 …… 238
5.2　電源設備 …… 239
5.3　空調設備 …… 240
5.4　コンピュータ用品および消耗品のための室 …… 241
5.5　その他の設備 …… 241
第6章　コンピュータ用品と消耗品 …… 243
6.1　コンピュータ用品と消耗品の種類 …… 243
6.2　導入計画 …… 243
6.3　コンピュータ用品と消耗品の管理 …… 243
第7章　機械化移行計画 …… 245
7.1　機械化移行にともなう作業の種類 …… 245
7.2　機械化移行計画のたて方 …… 245

科目「コンピュータ導入計画」

教育の目標

コンピュータの普及にともない，多くの業務がコンピュータを中心としたシステムのもとで処理されるようになった。コンピュータを業務に利用するには，コンピュータを導入し，システムの設計を行ない，プログラムを開発して，コンピュータを運営していかなければならない。しかし，コンピュータは一般に高価で，その役割は多く，十分な導入計画を立案して導入しないと，有効な利用は期待できない。

こうした背景をふまえて，コンピュータ導入に関する知識を身に付け，コンピュータ導入計画を立案しコンピュータ導入までの作業，日程表（たとえば，ガントチャート，PERT，CPMなど）などを作成できるように指導する。そのためには具体例を提示することは理解の手助けとなる。

時間配分

項　　目	履習時間（時間）
1. コンピュータ導入計画のたて方	10
2. 導入組織と人員計画	5
3. 教育計画	10
4. 機械化計画とプログラム開発計画	10
5. 設備計画	10
6. コンピュータ用品と消耗品	8
7. 機械化移行計画	7
合　　計	60

第１章　コンピュータ導入計画のたて方

用　語　この章では次の用語を教える。

機種選定

目　標

コンピュータは一般に高価であり普通の物品を購入するようにはいかない。

またコンピュータのはたす役割は大きく，さまざまな分野にわたり，使い方も多種多様である。しかし反面技術的な面，経済的な面，組織の動機づけの面などに解決しなければならない問題をいろいろかかえることが多い。

十分な導入計画をたてて計画にもとずいてコンピュータを導入すればこうした問題を未然に防ぐことができる。

ここではコンピュータ導入計画の手順を説明して概略を理解させる。

内　容

1.1 計画段階

計画段階では次のような作業が行なわれ，この段階はコンピュータ利用のよしあしが決まる重要な段階であることを理解させる。

- 導入方針の確立
- 導入体制の確立
- 調査分析
- 概要設計
- 機種選定

(1) 導入方針の確立

コンピュータ導入の着想はいろいろな部門から提案される。いずれの部門からの着想であっても，導入方針を明確にして導入を推進していかないと，せっかくの導入もむだになる要素が多い。

導入方針の確立要素として次の点を説明する。

① 導入目的

事務合理化，ＭＩＳの確立，サービス向上，機機や装置のオートメーション化，研究開発の促進など

② 期待効果

直接効果，間接効果など

③ 適用分野

事務，経営管理，科学技術計算，プロセス制御，数値制御，情報検索，教育，サービス分野など

④ 導入時期

⑤ 導入資金

導入経費，経常費など

(2) 導入体制の確立

導入方針が確立したら，導入実現をめざして導入体制を確立する必要がある。

導入体制については，次の点を説明する。これらは1ステップずつ進めるのではなく，並列的に推進させることを理解させる。

① 導入組織，人員の確立

② コンピュータに関連する設備の準備

③ コンピュータ設計要領の制定

④ コンピュータ導入資金の確保

(3) 調査分析

調査分析の目的，重要性，方法について理解させる。

次の点があげられる。

① 目的および重要性

・現状と問題点の把握

・コンピュータ・システムの検討

・導入目的の効果的な達成

② 方 法

・面接法

・観察法

・資料法

(4) 概要設計

調査分析から得られた資料にもとずいて，コンピュータを必要とする業務，コンピュータを利用する業務を明らかにし，システムを設定するために必要な概要設計について説明する。次のような作業が含まれる。

① 適要業務範囲内でのシステム条件の設定

② コンピュータによる処理の概要の設定

③ システムの評価ポイントの設定

・ドキュメントの標準化

(5) 機種選定

機種選定は

・導入方針に一致するか

・概要設計を満足するシステムか

という前提をもってなされることを理解させ，機種選定の方法について，次の点を説明する。

- 機種選定の手順
- 機種選定の要素
- 機種選定の評価要素

(a) 機種選定の手順

次の手順を説明する。

① コンピュータ導入組織（たとえばコンピュータ導入委員会など）の発足

② コンピュータ・システムの選定項目，条件の設定

③ 導入機種候補の選定

④ コンピュータ・メーカへの問いあわせや依頼

⑤ コンピュータ・メーカからの提案書

⑥ コンピュータ評価

⑦ 導入機種決定

(b) 機種選定の要素

次のような項目を説明する

① 処理方式

② ハードウェア

③ ソフトウェア

④ 運用方式

⑤ サービス

⑥ 価　格

(c) 機種選定の評価要素

次のような評価要素を説明する。評価要素の選定は導入する組織によって異なることも理解させる。評価要素を次に示す。

① ハードウェア

処理速度，信頼性，保守の容易さなど

② ソフトウェア

制御プログラム，言語処理プログラム，応用プログラム，各種ユーティリティなど

③ 人間に関する要素

プログラムの生産性，操作性，使いやすさなど

④ 総　合　性

コスト・パフォーマンス，適応性など評価方法の例として次のような方法を紹介する。

- カタログ・データによる比較方法
- テスト・プログラムによる評価（ベンチマーク・テスト）
- モデルによる評価
- チェック・リストによる評価

・総合採点表による比較方法

1.2 実施段階

実施段階では次のようなことが行なわれることを説明する。

・教　育

・設備工事

・詳細設計

・プログラム作成

・業務の移行

これらの内容については別の科目や章で詳しく説明するので簡単に説明する程度にとどめる。

(1) 教　育

次の点を説明する。

① 全社的な教育

経営管理者，コンピュータ導入関連部門，関係者など

② 基礎教育

コンピュータ部門に対して，コンピュータの基礎知識を身につけさせる。

③ 専門教育

コンピュータに関する専門知識，技術の習得を目的とするもの

(2) 設備工事

設備工事に関連して次の点を説明する。

① 建　物

② 空　調

③ 電　源

(3) 詳細設計

次の点を説明する。

① プログラム作成条件の設定

出力仕様，入力仕様，コード仕様，ファイル仕様，処理手順の決定，チェック・システムなどの作業による。

② 人間が行なう処理手続きの設定

事務処理手続きの設定，データ収集と事故対策，秘密保全要領の設定など

(4) プログラム作成

次のような作業について簡単に説明する。

① 詳細設計書の理解と確認

② プログラム開発計画の設定

③ プログラム仕様書の作成

④ プログラム流れ図の作成

⑤ コーディング

⑥ プログラムのデバグ

⑦ プログラム個別テスト

⑧ プログラム総合テスト

⑨ 操作仕様書の作成

(5) 業務の移行

従来の方式で行なっていた業務をコンピュータによる処理に移行させるために必要な作業を理解させる。

次の点を説明する。

- 新しい業務のテスト
- 新しい業務のための訓練
- 新しいシステムの文書化

1.3 評価段階

コンピュータにより処理が実施してからも，最初の導入方針が貫ぬかれているか，期待どうりの成果をあげているか，プログラムや処理の仕方に誤まりないか，環境がどう変化したかなどについて，新システムを維持しながら，評価していく必要があることを理解させる。評価について次の点を説明する。

- 評価要素
- 対　策

(1) 評価要素

次の項目について理解させる。

- 合目的性
- 信頼性
- 安定性
- 適応性
- 経済性

(2) 対　策

次の対策を説明する。

- プログラムの修正
- システムの変更

指導上の留意点

コンピュータの導入は経営組織の規模，導入目的，経営方針などが，経営組織によって異なるため，多くの方法，ケースが存在する。

ここではコンピュータを初めて導入する，中規模程度の例を中心に説明してある。できればすでに導入を行なった例を具体的に示してから教えた方が理解しやすい。

第2章　導入組織と人員計画

目　標

コンピュータの導入を推進するために，コンピュータを導入するための特別な組織を設けることが多いことを述べ，経営の規模や導入目的により組織形態が異なることを理解させる。

内　容

2.1　導入組織の種類

導入組織にはいろいろな形態があるので，それぞれの教育の場にふさわしい形態をとりあげて説明する。次の例は導入組織の1例である。

(1) コンピュータ導入組織

コンピュータ導入委員会，導入班など委員会は次のようなスタッフで構成されることが多い。

- 重役クラス
- 各部門統括責任者
- コンピュータ部門統括責任者

導入組織では次のような作業が行なわれる。

- 導入方針に基づく調査
- 導入計画進捗管理
- 経営者への助言

(2) 専門委員会

専門委員会では次のような作業が行なわれる。

- コンピュータ導入組織の補佐
- コンピュータの調査や分析
- コンピュータ導入段階の管理

(3) コンピュータ運営委員会

運営委員会では次のようなことが行なわれる。

- コンピュータ運営の方向づけ
- コンピュータ運営のための予算の決定
- 実施段階における組織

(4) コンピュータ運営専門委員会

次のような作業を行なう。

- コンピュータ運営委員会の補佐
- コンピュータ運営に関して専門的立場からの検討

(5) コンピュータ担当部門

次のような作業を担当する。

- 適用業務をコンピュータで処理できるように準備する
- 各部門からの専門家で構成されることが多い

2.2 導入組織の位置づけ

コンピュータを導入していくための組織は経営者の考え方，導入規模や導入目的によってその位置づけが異なる。

例として次のような位置づけを説明する。

- ゼネラル・スタッフとして位置づける，全社的な規模で運用する場合に多い。
- ライン部門と同列に位置づける
- 特定部門に直属させる

導入目的が限定されており，限られた範囲でコンピュータを活用する場合に多い。

2.3 人員計画

(1) 要員の種類

次のような要員について説明する。

- コンピュータ部門管理者
- システム・アナリスト
- システムズ・エンジニア
- プログラマ
- オペレータ
- キーパンチャ
- 事務担当
- チェッカ
- ライブラリアン
- スケジューラ
- 保守員

(2) 要員管理

次の項目について説明する。

- 勤務体制
- 勤怠管理と健康管理
- チーム
- 要員教育
- 職務の転換（ジョブ・ローテーション）

(a) 勤務体制

コンピュータはほとんどの場合，通常の勤務時間外でも使用されることが多い。そのため，オペレータや，プログラマなどの勤務が不規則になったり，負担がかかりすぎたりして問題が生じることが多いことを説明し，次の点に触れる。

・交替勤務

・時間帯

またキーパンチャの問題に関連して，次の点に触れるとよい。

・拘束就業時間と実労時間

・休憩時間と休憩室

(b) 勤怠管理と健康管理

次の点に関する管理を説明し，作業管理と健康管理は互に深い関係があることも理解させる。

・職業病

・病状の早期発見

・モラル管理

例として勤怠表による管理を説明し，病状の早期発見法，本人の自覚，定期健康診断による場合の対処方法について説明するとよい。

(c) チーム

コンピュータ・システムの運用は個人で行なうより，チームによって行なわれることが多い。これはプロジェクト・チームともよばれ，目的達成のため各構成員が機能を分担している。次の点について説明する。

・チームの目的と期間について

・チームの編成と人員

・リーダ

・構成員の身分

・討論方法

・チーム組織のあり方

(d) 職務の変更（ジョブ・ローテーション）

次の点について説明する。

① 目　的　　向上への努力，幅広い人間形成，モラル向上など

② 問題点　　職務年限，順序，頻度たとえば学力レベル，経験年数，年令などによるちがいなどの具体例を取上げ，対策を検討させるとよい。

指導上の留意点

第1章でも述べた通り，ここでは中規模程度の例を中心に説明してある。したがって導入組織は一例であり，また位置づけについても導入規模や導入目的によって異なるので一例として述べてあることに注意する。

第3章　教　育　計　画

目　標

教育計画は，コンピュータ運営上の上でも，導入計画を推進していく上でも，要員の確保，ＰＲなどの面からも重要であることを理解させる。

内　容

3.1　教育の必要性と目的

次の点を説明する。

- 要員の確保
- 要員技術の向上
- コンピュータ導入関係部門へのＰＲ
- コンピュータ導入に対する組織の思想統一と組織環境の統一
- 要員間のコミュニケーション

これらは運営規模，組入方針により異なるが，情報処理技術者育成指針についても触れるとよい。

3.2　教育の性格

次の点を説明する。

(1) コンピュータ・システム概略教育

コンピュータ導入関係部門，経営管理者に対してのＰＲ，コンピュータ・システムの概略などの教育

(2) 専門教育

コンピュータ要員に対する専門知識の学習など。

(3) 実務教育

要員に対する実務の訓練など。

3.3　教育の対象

次のような教育の対象者について述べる。

- 経営管理者
- コンピュータ導入組織の委員
- 専門委員会の委員
- コンピュータ担当部門
- コンピュータ導入関連部門
- コンピュータ導入関連外部門
- 他社からの要員（ファシリティ管理など）

3.4 コースの選定

(1) 教育機関

次の教育機関について説明する。

- 公共団体
- コンピュータ・メーカ
- 専門学校
- コンサルタント会社やソフトウェア会社
- 通信教育
- 自社

(2) 教育方法

次の方法を説明する。

- 専任教師による方法
- 自　習
- 視聴覚教具や教材の活用

(3) コース選定の基準

次のような点を述べる。

- 教育目的
- 教育対象
- 教育水準
- 教育期間
- 費　　用
- 人　　数
- カリキュラムと教育内容
- 使用する教材
- 教育機関

3.5 教育体制の確立

教育体制を確立することはコンピュータ導入後，コンピュータをよりよく運用していくために必要であることを理解させ，次の点について述べる。

- 教育体制確立の必要性
- 教育体制のあり方

教育目的，教育方法，教育対象，カリキュラム，日程，教育の機関，教材，教育成果の確認，研修など。

指導上の留意点

実例を示して理解させるとよいがあまり深く立入るべきでない。

教育の対象では図などを使用して説明すると有効的である。

第4章　機械化計画とプログラム開発計画

目　標

機械化計画とプログラム開発計画は，導入方針を確実に実施させる段階であり，適用業務の決定，コンピュータの設置，システム設計，プログラム作成から成るもので，最少の費用と期間が要求され，着実な管理が望まれることを理解させ，次の点について説明する。

内　容

4.1　計画のたて方と作業管理

次の点について述べる。

- 作業の種類
- 期間の設定
- 経費の算出
- 作業体制の決定
- 外注の管理

(1) 作業の種類

次のような作業がある。

- システム分析
- システム設計
- プログラミング
- デバグ
- システム・テスト
- コンピュータ室の工事
- 空調工事
- コンピュータ搬入
- 空調搬入
- 配　　線
- 現　　調
- ハードウェア・テスト
- 空調テスト
- 立合検査

(2) 期間の設定

次の点を説明する。

・担当者の能力

・契約上の期間

・搬入時期

・外注の能力

(3) 経費の算出

次の点を説明する。

・内部作業費（1人あたりのコスト，工程数による。）

・外注作業費

・コンピュータ用品，消耗品費

・コンピュータ・マニュアルの購入費

・各種レポート費

・材　料　費

・各種搬入費

・雑　用　費

(4) 作業体制の決定

次の点を説明する。

・組　織

・リーダ

・作業の分担と権限と責任範囲の決定

・作業報告の標準化と文書化の標準化

・外注要員

(5) 外注の管理

コンピュータ室の工事やプログラミングについて，作業の専門化，多量化に伴い外注に作業を委託することが多い。作業の外注に関連して，次の点を説明する。

① 外注の方法

外注への委託方法，契約，打合せなど。

② 外注の管理

工程管理，品質管理，作業管理，コスト管理，評価など。

③ 作業完了時の問題

検収，文書化，納入書，完了通知書など。

指導上の留意点

この段階では計画を明確にし，運用しやすくするため，多くの表が用いられる。これらの表について，作成方法を説明し，実際に書かせると理解しやすい。（PERT，CPM，ガントチャートなど）

第5章　設　備　計　画

用　語　この章では次の用語を教える。

機械室のレイアウト，フリー・アクセス，電源装置，ＡＶＲ，定周波定電圧装置，無停電装置，空調装置

目　標

コンピュータを十分活用していくためには，コンピュータ室や，関連設備を十分に検討し，コンピュータの能力を十分発揮できるように管理していくことが大切であることを理解させるとともに設備計画についても説明する。

内　容

5.1　コンピュータ室の設備計画

次の点を説明する。

- コンピュータの環境
- 機械室のレイアウト
- 床　方　式

(1)　コンピュータの環境

次の点を説明する。

①　入力電源

電圧，周波数，相数，波形ひずみなど。

②　室内温湿度

温度，湿度，温度こう配など。

③　室内じんあい

室内じんあい，腐食性ガスに対する対策など。

④　設置面振動

⑤　室内磁気および電界

磁気記憶に対する問題など。

⑥　アース

⑦　床

コンピュータの荷重対策，配線への影響，空調方式への影響，塵埃の問題，静電気の影響，絶縁性の問題，床の面積，配線への影響など。

⑧ 壁および窓

(2) 機械室のレイアウト

コンピュータ・オペレーションや，メンテナンスが最も効率的に行なえるよう**機械室のレイアウト**を行なう必要がある。

① コンピュータからのレイアウト

- 各装置の外形，寸法
- 各装置の据付け条件
- 配線ケーブル
- 各装置の発熱量

② 保守性

- 保守エリアの確保

③ 操作性

- コンソールの位置と作業領域
- 室の出入口
- 付属機器，用品，消耗品などの取扱い

④ その他の要件から見たレイアウト

- 室の広さ，高さ
- 柱の位置
- 窓の位置
- 照明の位置
- 配置上の美観
- 将来の問題
- 空調方式

(3) 床方式

床方式は一般に次の方式があることを説明する。

- **フリーアクセス**方式
- レースウェイ方式
- 床上ダクト方式

床方式は，コンピュータの温湿度設定条件により，空調方式，冷却方式によって影響を受けることにも触れる。

5.2 電源設備

電源装置の設置環境を理解させ，電源装置の種類について説明する。

(1) 電源装置の環境

次の点を説明する。

- コンピュータの入力規格

・停電への対策

・アース

・落　電

・設置場所

・電源の大きさ，形

(2) 電源装置の種類

次のような装置について，用途，特徴などを説明する。

・変圧装置

・ＡＶＲ（自動電圧調整装置）

静止型自動電圧調整器，誘導型自動変圧調整器，電動発電器などがある。

・定周波定電圧装置（ＣＶＣＦ）

回転型，静止型がある。

・無停電装置

5.3 空調設備

コンピュータは，一般に多くの熱を発生する。また，ハードウェアの特性上，一定温湿度で運転しなければならない。じんあいも嫌う。このため空調装置を必要とすることを理解させ，空調装置の設置環境，空調方式の種類を説明する。

(1) 空調装置の環境

次の点を説明する。

・コンピュータの温湿度，じんあい条件

・空調装置の許容量

コンピュータの総発熱量，室内にある他装置の総発熱量，人間の発熱量，照明発熱量，外部からの侵入熱量，コンピュータ室の広さなどが検討される。

・コンピュータの配置と高さ

・空気取入口

・温湿度自動制御の検討

・塵埃除去装置

空気濾過器，空気清浄装置，集塵機などがある。

(2) 空調方式の種類

次の方式について説明する。

・パッケージ空調方式

・ダクト空調方式

・床下空調方式

また水冷による冷却方式についても触れるとよい。

5.4 コンピュータ用品および消耗品のための室

コンピュータ用品および消耗品を管理し保存するために，特別な室が用いられることが多い。次の点を説明する。

- 管理，保存する用品，消耗品の決定
- 量，大きさ
- 温湿度条件

5.5 その他の設備

次の設備について説明する。

- データ・エントリ装置の設置
- 作業室とその他の室

(1) データ・エントリ装置の設置

コンピュータを導入し，運営していく上で，データ・エントリ装置が必要となることを理解させ，次の点を説明する。

- データ・エントリ装置の種類
- データ・エントリ装置の設置環境

(a) データ・エントリ装置の種類

次の装置について説明する。

- カードせん孔装置
- 紙テープせん孔装置
- キーツーディスク装置
- キーツーテープ装置
- キーツーカセット装置
- 端 末 装 置

(b) データ・エントリ装置の環境

次の点を説明する。

- 温 湿 度
- 入 力 電 源
- 作 業 領 域
- 台　　数

(2) 作業室とその他の室

次のような室について説明する。

- 居　室

・保　守　室

・データせん孔室

防音，照明，色彩，温湿度などが問題となる。

・休息室と仮眠室

指導上の留意点

機会があれば，実際に機械室を見学させ，実際の設備などを学ばせることも，理解を早める。

床方式は，図を使用することにより，その各方式の違いがわかり理解の助けとなるだろう。

第6章　コンピュータ用品と消耗品

目　標

コンピュータ導入計画の一環として，コンピュータ用品や消耗品の購入計画について理解させる。

内　容

6.1　コンピュータ用品と消耗品の種類

(1) コンピュータ用品

磁気テープ・ロッカ，磁気ディスク・ロッカなどのコンピュータ用品について次の点を説明する。

- コンピュータ用品の名称
- 用　途
- 特　徴

(2) 消　耗　品

ライン・プリンタ用紙，紙カード，リボンなどの消耗品について，次の点を説明する。

- 消耗品の名称
- 用　途
- 特　徴

コンピュータ用品，消耗品はメーカにより名称，特徴などが異なるので，なるべく多くのメーカ品について，具体的な物を提示して，説明すると理解しやすい。またできるなら，市場価格についても説明するとよい。

6.2　導　入　計　画

次の点について述べる。

- 設備計画との関連
- 導入コンピュータの確認と作業量の推測と用品，消耗品の関連
- 価　格
- 大きさ，重量，色
- 調達時期
- 取扱い

6.3　コンピュータ用品と消耗品の管理

(1) コンピュータ用品

次の点を説明する。

(a) 設置条件

特にファイルを保管するコンピュータ用品は，磁気，温度，湿度，日あたりなどが問題となる。

(b) 取扱い

長期間，ファイル等を保管する用品は，ファイル等の破損，盗難などが問題となる。

(c) 安全対策

秘密保持をするファイルのための用品管理，火災に対する対策など

(d) 責任者

(2) 消耗品

消耗品は，量や大きさに種々のものがあり，頻繁に入出庫することを理解させ，次の点について説明する。

(a) 保存条件

温度，湿度，日あたりなど

(b) 量の適性な把握

入出庫の状況や，発注のための棚卸など

指導上の留意点

コンピュータの用品，消耗品はメーカーにより名称，特徴などが異なるので，多くのメーカ品について，具体的な物を提示するか，写真などをできるだけ多く用意して利用すると有効である。

またこの章は，第5章で機械室を見学させたとき，ついでに磁気テープ・ロッカ，磁気ディスク・ロッカの説明がなされていると，理解を早める。

第７章　機械化移行計画

目　標

従来のシステムから新しく導入したコンピュータを中心としたシステムへスムーズに移行していく段階であることを理解させ，次の点を述べる。

内　容

7.1　機械化移行にともなう作業の種類

次の作業について説明する。

- 新システム実施要領の作成および標準化
- 新システムのＰＲ
- 新システムの訓練
- 新システムのテスト
- コンピュータ用品と消耗品の調達

7.2　機械化移行計画のたて方

次の点を説明する。

- プログラム完了予定日の把握
- 基本ファイル作成完了予定日の把握
- 新システム実施要領完了予定日の把握
- 新システムＰＲ期日の決定
- 新システム訓練期間の決定
- 新システム訓練者の決定
- 新システム・テスト期間，対象，開始予定日の決定
- コンピュータ用品および消耗品調達期間の決定

計画の変更にともなう計画修正についても説明する。

指導上の留意点

計画については，実際の日程表（ガントチャート，ＰＥＲＴ，ＣＰＭなど）や，計画書を実際に作成させたほうが理解しやすい。

参考文献

(1)　土岐秀雄著，「電子計算機／機械室管理」，日本経営出版会，昭和42年，1,500円，全般的に参

考になる。

(2) 南条優，保坂岩男共著，「システムの開発と運用」，オーム社，昭和47年，1,800円，全般的に参考になり，具体例がよく示してある。

(3) 社団法人日本能率協会出版事業部，「EDPスペシャル・リポート⑩」，昭和43年，実費，機械化移行計画，教育計画，機種選定が参考になる。

(4) 吉谷竜一著，「日程の計画と管理」，日刊工業新聞社，昭和45年，800円，全般的に参考になる。

(5) ジェームズ・H・グリーン著，高井英造訳，「オペレーションの計画と管理」，東洋経済新報社，昭和44年，850円，オペレーションの計画に参考になる。

(6) 「組織の研究と文書類作成の手法解説書」，日本アイ・ビー・エム株式会社，実費

(7) 「OUK 9200／9300設備計画書」，沖電気工業株式会社，日本ユニバック株式会社

(8) 「TOBAC－5600導入設備計画と設置工事」，東京芝浦電気株式会社，実費

(9) 「EDPシステム設計」，日本電気株式会社，実費

(10) 「プログラミング作業の標準化について」，日本電気株式会社，実費

(11) 「HITAC 80000設備計画資料」，株式会社日立製作所，実費

(12) 「HITACEDPSシステム設計」，株式会社日立製作所

(13) 「プログラミング計画と標準化」，富士通株式会社，実費

(14) 「FACOM 230－25／35／45 設置計画手引書」，富士通株式会社，実費

(15) 「導入解説書」，三菱電機株式会社，実費

経　営　情　報

目　　次

第１章　経営活動と情報システム …… 248

1.1　企業経営階層 …… 248

1.2　企業における情報 …… 250

1.3　意思決定と経営情報 …… 252

第２章　経営情報システム …… 253

2.1　経営情報システムとは …… 253

2.2　計画情報システム …… 254

2.3　管理情報システム …… 254

2.4　作業情報システム …… 255

2.5　経営情報システムの分類 …… 255

第３章　経営における情報システムの例 …… 258

3.1　高度な受注管理 …… 258

3.2　生産管理システム …… 259

3.3　在庫管理システム …… 262

3.4　販売情報システム …… 264

3.5　人事情報システム …… 266

3.6　経営計画補助システム …… 267

第４章　経営情報システムの開発（演習） …… 269

4.1　演習の手順 …… 269

4.2　例　題 …… 271

科目「経営情報」

教育の目標

コンピュータの経営・事務面への利用を担当する情報処理技術者の育成に際し，教育方法としてはＯＪＴによりあるシステムを担当させ経験によって逐次知識を得て行く方法もあるが，本科目では，まず「経営活動の場で情報と情報システムがどのような働きとどんな位置を保っているか」を解説し，経営においてコンピュータがどのような役割を果すかを概念的に知らしめ，ついで経営情報システムと称されている経営・事務面でのコンピュータ利用の本質を教える。

以上の基礎的考え方を第1,2章で学ばせ，ついで情報システムの諸例を第3章でとりあげる。ＯＪＴでは一つのシステムを修得するには数年の経験が必要だが，諸システムはそれぞれ固有の特徴を持っているので，出来るだけ多くのシステムのもつ特徴を短時間の間に修得させることを目的としている。

講義だけでは概念的にしか「経営情報」を修得させたことにならないので，かなりの時間を演習に使用し，具体的に「経営情報」を掴ませる努力をすることが望ましい。ここではＯＪＴの長所を出来るだけ取り入れてあり，そのため演習の範囲は企業全体を包括するようになっている。

時間配分

章	履習時間（時間）		
	講　義	演　習	実　習
1.　経営活動と情報システム	8		
2.　経営情報システム	18		
3.　経営における情報システムの例	21		
4.　経営情報システムの開発（演習）		33	
合　　計	47	33	

第１章　経営活動と情報システム

用　語　この章では次の用語を教える。

経営のピラミッド，ライン，スタフ，企業内情報，企業外情報，定型的情報，非定型的情報，経営科学

目　標

情報処理技術者は情報システム確立の業務に従事しているが，この場合広い視野から情報システムを観察しうる能力を養成する必要がある。これは単にシステムが目的にそって完成すればよいと言うのではなく，経営管理の中にその情報システムがどう機能するかが情報システムの価値を左右するのである。このために以下の事項を修得させる。

- コンピュータ化の対象になっている情報システムが企業経営のどの部分を占めているかを知るため経営管理の本質とその構造を知ること。
- 企業における情報の重要性と価値を知り，経営管理と情報システムの関係を把握すること。
- ひいてはコンピュータが企業内で占める重要性を知り情報処理によってコンピュータが造り出したものが経営管理にどれだけの利益をもたらすかを相対的に知ること。

内　容

1.1　企業経営階層

企業を構成する経営階層は経営学によると経営意思決定層（経営者層…トップ・マネジメント層）管理者層（ミドル・マネジメント層），（作業者層（オペレーション層）の三つがあり図１－１のように経営ピラミッド(1)を形成していることを解説する。

ついで経営の中には幾つかの基本的な職能があり（表１－１参照），これら

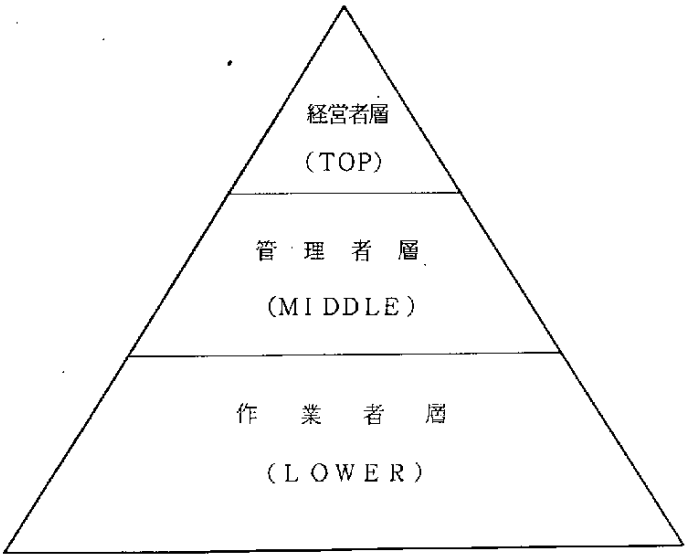


図１－１　経営のピラミッド (1)

の職能の組合せは業態の相違によって異なり**経営のピラミッド**の中にそれぞれ機能していることを説明する。できればそれぞれの職能の基本的機能を簡単に解説しておく。

それぞれの職能は経営のピラミッドの上で経営階層を同じ関係でもっていることを解説する。図1－2経営のピラミッド(2)を参考にするとよい。

表1－1　　基本的職能

	主要な基本的職能										
	経理・会計	人事・労務	販売管理	在庫管理	生産管理	品質管理	技術管理	新製品開発	購買管理	顧客サービス	その他
製造業	○	○	○	○	○	○	○	○	○		○
商業	○	○	○	○					○	○	○
金融業	○	○								○	○
不動産業	○	○		○						○	○
運輸業	○	○		○			○			○	○
・											
・											
・											
・											

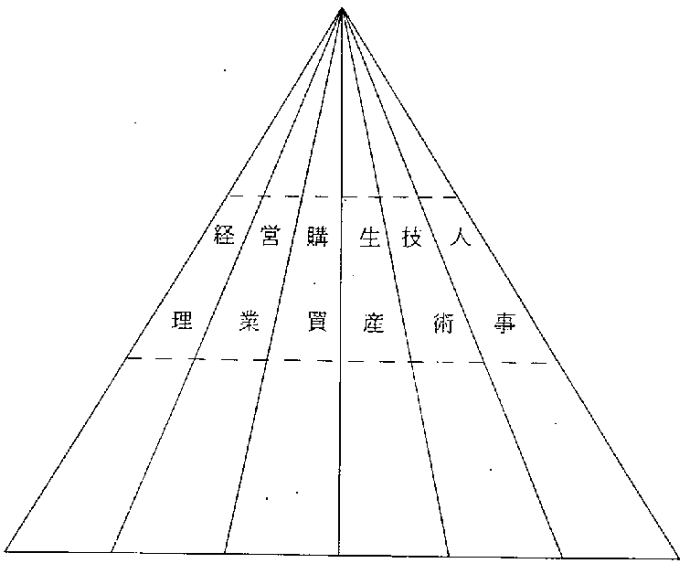


図1－2　経営のピラミッド　(2)

図1－2で図示されるピラミッド形の経営組織の見方を変えると **ライン** と **スタフ** があり，それぞれ企業内に重要な役割を果しているがライン，スタフについてそれぞれの役割を解説する。この場合前の経営階層と関係づけて説明するとよい。

ライン，スタフの説明においてぜひ付加したいのは，前者が定形的業務をその職としているのに対し，後者は非定形的な業務が多いことであり，情報システムの形成は前者の方がより容易であることを解説したい。

1.2 企業における情報

企業の情報の存在とその種類を分析し，情報システムの対象となる情報がどのような形で存在するか解説する。

それにはまず次の概念を身につけさせる。

(1) 企業内情報と企業外情報

企業内情報として一般に企業内の事務処理（事務計算）で扱われる情報，例えば人事情報・経理情情・技術情報・生産情報・在庫情報などや，コンピュータ化の対象にならない情報（非定型的情報といった方がよい）があり，これらはそれぞれ数値的情報・非数値的情報があることを具体的に解説する。

この他に企業外の情報があり，企業経営にとって非常に重要な働きをすることを説明する。

企業外情報としては

- 業界情報
- 関係官庁諸統計
- 主要企業財務諸表（有価証券報告書）
- 株式情報
- 諸外国諸統計
- 新聞情報
- 金融情報

などがあり，それらの情報のうちコンピュータの処理可能なファイルとなっているもの，いないものなどについて説明する。

(2) コンピュータで処理可能な情報

企業内外の情報が，企業経営にとって役に立つ情報と役に立たない情報とに分けるニーズのふるいがあり（図1－3），その目をくぐって来た情報（企業にとって有効であるという意味で「有効情報」と呼ぼう）にはコンピュータで処理可能な定型情報と一般には普通のファイリング・キャビネットに入る非定型情報とが存在することを解説し，定型情報か非定型情報かは機械化をしようという意志と同種の情報の量の大小によることを説明する。

磁気ファイルになるか書類ファイルになるかは以上のように量とデータ処理の経済性によってきまることを教える。

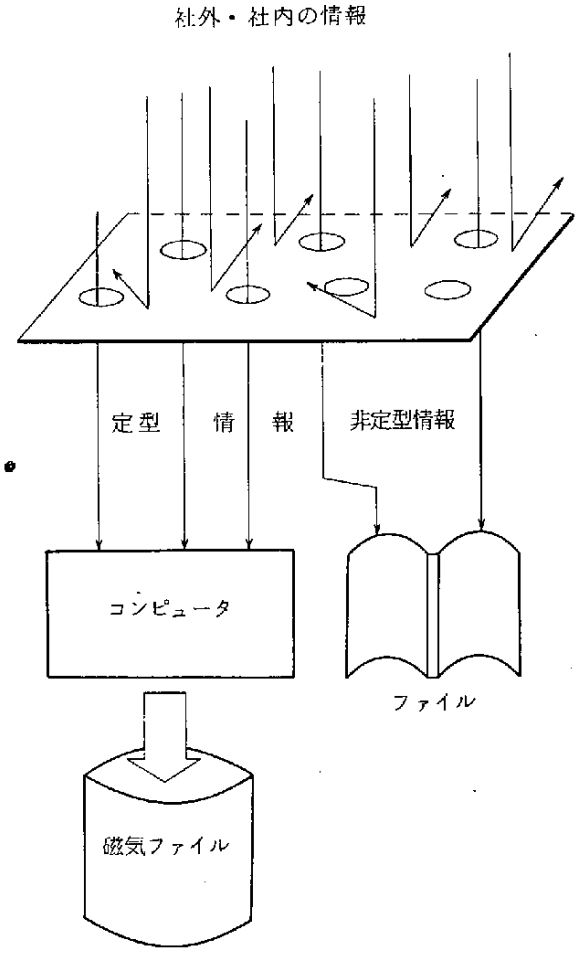


図1－3　情報の整理

(3) 情報の収集

コンピュータに情報を収集する方法について基本的なパターンを解説する。例えば

① バッチ処理へのオフライン・インプット……コンピュータの入力媒体に変換しうるデータの収集，

・原始伝票をキーパンチにより紙カードに変換し，まとめてインプットする。

・原始伝票をキー・エントリーして磁気テープに変換する。

・光学文字読取機（OCR）や光学マーク読取機（OMR）用の原始帳票を作成してインプットする。

・帳票を作ると同時にコンピュータに直接インプットできうる紙テープなどの媒体を作り，それをインプットする。

② リモート・バッチ処理へのオンライン・インプット……上記バッチ処理の変形

③ リアルタイム処理へのオンライン・インプット……ランダムに発生したデータを発生都度インプットする，即ちリモート端末（タイプライタ，CRTディスプレイ装置，インテリジェント・ターミナルなど）からその都度インプットする。

④ ミニコンを用いたデータ収集……例えばPOS（point of sales）レジスタを動かすと同時にデータを収集する。

1.3 意思決定と経営情報

意思決定に必要な経営情報とはどんなものがあるか考えさせる。意思決定過程とはどんなものであるか理解するのは情報処理技術者が一番苦手とするもので特に企業経営のトップ層の意思決定についてはまったく理解できないと思われるから経営情報の側からアプローチする。

(1) 定型的情報と非定型的情報

情報処理に適するのは，パターン化された情報で，まったく予測しえない情報は情報処理に適さないことを理解させる。意思決定は，このような**定型的情報**をもとに行なわれるケースはごく少なく，突然表われた変化に対して，右にするか左にするかを決める必要に立たせられる。つまり**非定型的情報**に基いて行動する場合が多い。

(2) 経営科学の有効性

意思決定はこのように特に定まっていない，どちらかと言えば整理されていない事象に基いて意思決定まで導く過程が重要であり，このために**経営科学**は方法論として有効であることを解説する。

問題を数式化できる場合には確率統計や，数理計画などのORの手法が有効で，数式化できない場合はシミュレーションやモデリングなどの手法が有効であることを解説する。

この解説に際して一，二の簡単な例題を用いることが望しい。

指導上の留意点

この章は経営情報システムの概要を教えるにあたり，その基礎的概念として経営活動の場に情報システムがどのように役立つかを知るためにもうけてある。そのためカリキュラムの作り方は概念的な部分が多く，このまゝ指導すると，いわゆるお話しになってしまうので出来るだけ具体性を持たせるような話し方及びテーマを与えて討論させるよう指導して欲しい。

また経営学の分野を扱っている部分があるので基礎編の「経営実務」を復習する。

第 2 章　経営情報システム

用　語　この章では次の用語を教える。

経営情報システム，計画情報システム，管理情報システム，作業情報システム

目　標

この章では，企業における経営情報システムの実体を認識させることを目標とする。それにはまず今迄知識として持っていたいろいろの事務処理システムが経営情報システムの中でどのような位置に存在するかを知らせ，ひいてはどんな形態の経営情報システムがあるかを分類することにより経営情報システムの概念を把握させる。

結果としてある企業のシステム作成の場に当った時に企業全体の経営情報システムと個々のシステムの間の関係を正確に把握できるような能力を養成する。

内　容

2.1　経営情報システムとは

「**経営情報システム**とは何か」という疑問に対して正確ではないにしても正鵠を得た解説をしておいた方がよい。

特に個々のサブシステムの経営情報システムとの関係を知っておくことが大切である，このために第2－1図程度の解説を試みておこう。（この図は図1－1と照らし合わせて見ると分かりやすい。）

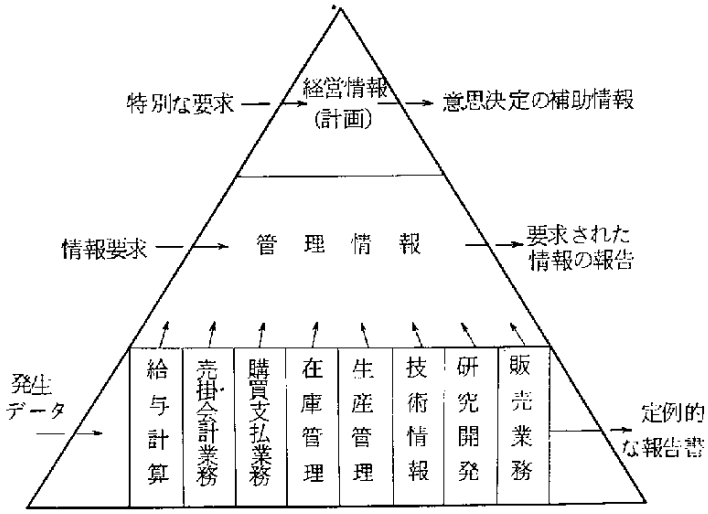


図2－1　経営情報システム

図の説明上注意したいことは個々のサブシステムの位置が経営情報システムの作業レベルにおける一部分のように見られることだが，この不足分は，図2－2で補ってほしい。

経営情報システムを計画情報システム，管理情報システムおよび作業情報システムの三つの階層に対応する情報システムに概念的に分類して指導すると分かりやすい。

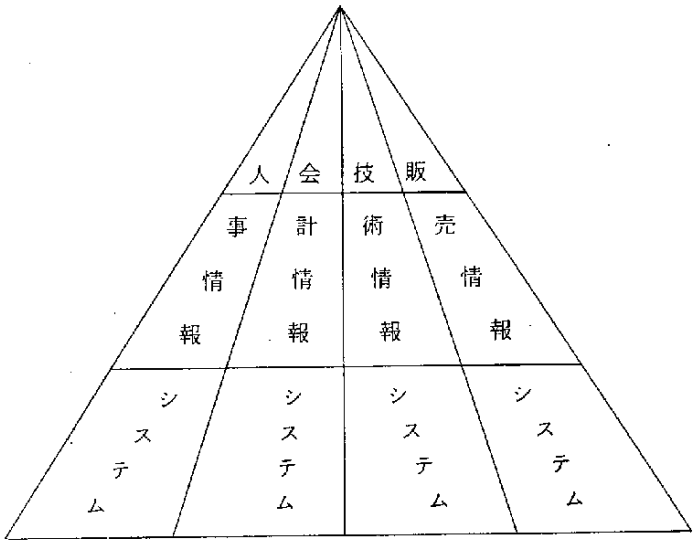


図2－2　経営情報システムの分野別系統

2.2 計画情報システム

トップ•マネジメントおよびトップ•マネジメントを補佐するスタフ(ゼネラル•スタフ)に必要な情報システムを解説する。特にこのレベルにおいて意思決定が必要な場は定例的な情報処理の結果について言々している場ではなく，まったく今まで予期し得なかった場合か，またはあらかじめ完全な情報を収集し得ない場合における意思決定が多いので，一般にはコンピュータを手段とする情報システムを形成しにくい。そのため**計画情報システム**の普及は現在でも相当遅れているので，事例をあげて説明しにくい。従って下記のヒントにしたがい説明する。

① トップ・マネジメントはどんな情報を要求するか

② トップ・マネジメントを補佐するためにどんな手法が有効か

- 経営科学の諸手法…シミュレーション，モデル作り，最適化技法，線形計画，投入産出分析，統計，システム・ダイナミックス，他
- データベースの作成と検索…トップ・マネジメントに必要なデータベース
- システムズ・アプローチの考え方…問題解決のためには十分利用できる。

③ コンピュータ，情報処理技術者の時間および金を計画情報システムにどの位流用できるか。

④ デマンド処理（要求があった場合にただちに実施する情報処理）に対応するソフトウェア，ハードウェアの可能性

2.3 管理情報システム

計画情報システムが経営をオーバーオールした考え方で作られているのに対して，管理情報システムは主として中堅管理者が機能別に分かれた職務（経理・人事・生産・販売など）において，人と物と金と情報を管理するために必要な情報システムによって構成される。

また，**管理情報システム**は作業情報システムと計画情報システムの中間に存在しているので計画

情報システムのような完全な非定型的情報処理とはいえず，また作業情報システムのように定例的な情報処理でもない，しかしどちらかといえば定例的な情報処理に例外管理や異常値管理を加えた，ややデマンド処理型の情報システムと考えてよい。

管理情報システムは以上のような位置に属するかと思うが表現がデリケート過ぎはっきりしないので次のように割り切って指導したらよいだろう。

管理情報システムは中級管理者向きにアレンジした作業情報処理でルーチン的に実施されるとは必ずしも限らない。原則として必要なときに必要なデータがデータベースからアウトプットされればよい。

また，管理情報システムは部長・課長向きデータ処理システムと考えてもよい。その例として人事情報システムなどを採りあげるとよい。

管理情報システム用のデータベースについては次のように考えよう。作業情報システムのデータベースは原単位のデータがなまのまま貯えられているからデマンド処理型のデータ処理（特に情報検索など）ではそのままの型で取扱いにくい。従って編集し直してからアフトプットされる方式がとられる。このことが前から予測し得るならば管理情報システム用のデータベースを作業情報用データベースから編集し直して作っておいた方が便利かもしれない。従って管理情報システム用データベースは現在のハードウェアの能力を条件と考えて，あらかじめ専用のものとして持っていた方が得という結論になるだろう。

管理情報システム用データベースはデマンド処理志向型のデータ構造をもっている必要があるのでこの辺りを十分解説しておく。

2.4 作業情報システム

今日行なわれているコンピュータ利用の大部分がこの**作業情報システム**の範ちゆうに属するもので，情報処理技術者の殆んどは作業情報システムの設計・開発にあたっている現実を説明し，作業情報システムとは図2－1でみるようにトランザクション・データ（発生データ）を受け入れてあるきまった時間または日の単位で定例報告書をアウトプットするシステムであることを解説する。

作業情報システムを説明することは比較的容易であり，具体的には第3章で実例について解説して行けばよいので，ここでは作業情報システムの経営情報システムの中における位置およびそのデータベースが上のレベル（管理・計画層）のデータベースの基礎である事実を中心に説明すること。

2.5 経営情報システムの分類

ＭＩＳ（Management Information System; 経営情報システム）は企業の形態，企業内の業務の種類，コンピュータの利用方式，ユーザの経営階層のレベル，利用目的の五つの角度から考え，それぞれの分類の中がどのように分かれているかを理解させ，ＭＩＳを構成するサブシステムを理解させる。

(1) 経営機能別分類

経営を構成している機能（企業内の業務の種類）に対応する情報と情報システムを表2－1に従って解説する。

表２－１　主たる経営機能別分類

販　売　情　報	販売・在庫情報，得意先情報，需要予測，アフターサービス情報競争他社情報など
会　計　情　報	売掛金請求・回収情報，買掛金支払情報，財務会計情報，原価計算など
生　産　情　報	生産計画，部品構成・部品展開情報，日程・進捗情報，工数情報設備情報など
技　術　情　報	技術情報，特許情報，設計情報，新製品開発に関する情報，技術管理情報など
人　事　情　報	従業員に関する記述情報，職歴情報，給与情報，個人能力情報など

(2) 業種別分類

業種の相違によって情報システムは異なるので業種別分類について解説をする。例えば以下の各業種について，出来るだけ数多く広く紹介しておきたい。

① 金融・保険業

銀行の普通預金や為替などのバンキング・システム，損保や生保の契約ファイル・システムなど

② 商　業

契約や取引などの商社総合システムやメッセージ・スイッチング，セールスマン管理システム，百貨店での商品大量仕入販売システム，クレジットの管理システム

③ 製造業

生産から販売までの諸システム，製造自動化システム，設計自動化システム，自動制御などのオートメーションなど

④ 運輸業

座席予約システム，運行計画システム，機械管理システムなど

⑤ その他

広域職業紹介システム，市場調査，病院管理システム，不動産紹介システムなど

(3) 機械化方式別分類

コンピュータの処理方式，コンピュータ・システムの構成，規模など，通信回線との接続方式などの相違による分類や処理のサイクル（業務処理の頻度）の相違による分類なども考慮に入れておく必要があるので説明しておきたい。この分類を歴史的発生順に挙げてみよう。

① ローカル・バッチ処理方式

従来からのバッチ方式でその処理の頻度によって，

・月次処理　(monthly)

・週毎の処理（weekly）

・毎日の処理（daily）

② オンラインによるデータ伝送方式

次にデータ収集だけでも通信回線を利用しようと考える。この方式は次のように発展する。

- point to point： データの送付に通信回線を利用したもの。コンピュータはデータの送付過程に
- 上記の発展型：データ・センタ側はコンピュータに直接インプットする。しかし，この場合データを集めるだけにしかコンピュータは使われていない。

③ オンライン・リアルタイム方式

データ通信の発達でデータ発生時点でそのデータを処理してしまう方式として最近大いに利用されるようになった。

④ タイムシェアリング方式

コンピュータを通信回線により複数のユーザが同時に利用する方式。端末はタイプライタから映像装置付のものに発展した。

⑤ リモート・バッチ処理方式

タイムシェアリングの端末が，タイプライタではスピード的にも量的にも制限があるので小型コンピュータまたは小型コンピュータなみの入出力装置を接続したもの。

以上のほか，これらの混合型や発展方式があるが基本的に簡単な例により説明し，対面するシステムについていろいろな方策を考えうる実力を養成する。

(4) 管理レベル別分類

それぞれのアプリケーションがどんな階層に役に立つかを例を挙げて解説する。

① 作業レベル

一般の定例的なデータ処理は99％作業レベルであるので特に例を挙げて説明する必要はない。

② 中堅管理レベル

定例的なデータ処理の中で特に管理者が必要とする例外的事項とか異常の検出とかの情報処理および定例的なデータ処理により作られた結果から，管理者の目的に応じて非定例的な情報処理要求にもとづいてアウトプットを出すなど。

③ 経営者レベル

特に経営科学として展開されるシミュレーションやモデル作り，最適化技法や線形計画などの手法とか，作業レベルで作られた情報を検索するとかがある。管理レベル別分類については説明しにくいがやはり何かモデル的なシステムによりレベルの相違をはっきりさせなければならない。

指導上の留意点

経営情報システムを説明するとき兎角落ち入り勝ちな観念論にならないように注意する。 そのためには出来るだけ豊富なシステムの実例を説明することである。実例は教師の経験により多少分野のかたよりが出てくることは止むを得ないが，出来るだけそのかたよりをなくすために実例報告集（例えば各コンピュータ・メーカのユーザ研究会議事録や実例集）などにより勉強させるようにする。

この科目では実習を行なうことになっているのでそのような場合に補足しておくのもよい。

第３章　経営における情報システムの例

目　標

全体的なＭＩＳの概念ができたところでＭＩＳを構成する諸サブシステムについて主だったシステムを開発しうる知識と能力を養成するためにこの章と次章がもうけられている。

そのため第３章は代表的な単語を解説するということではなく代表的な諸サブシステムを解説する方向をとってある。

ここで取り上げたシステムは製造販売業種のシステムであるが情報処理技術者の所属する企業によっては，この例では不適当な場合もあるので，その場合は教師がサブシステムを再編成して指導する。履習時間については第３章の履習時間を確保するように注意すること。

3.1では高度の受注管理はリアルタイム・システム指向になることを教え，データ収集とデータベースの確立について教育する。

3.2はデータ構造の特徴を修得することを目標とし，さらにデータの量と処理の頻度との関係 からデータ構造を研究させる。

3.3は他のシステムとのインターフェイスを重要視し，システム化させることを狙いとしている。

3.4販売情報システムではデータの正確性の維持について解説する。3.5人事情報システムはどちらかというと管理情報システムの分野に属する情報システムで3.6は計画情報システムの分野の諸例の一つとして修得させる。

内　容

3.1　高度な受注管理

企業が特に努力をする所は注文を受けて，注文内容に従ってサービスを行なう受注管理（オーダー・エントリー）の分野である。顧客の要求を人間が処理をすることが最近までたて前になっていたので機械化システムとしてオーダー・エントリー・システムは余り一般的ではなかったと言える。しかし，製造販売業においては今後特に重要になると思われるので本サブシステムについて解説する。

(1)　オーダー・エントリー・システムの難しさ

オーダー・エントリーは販売する商品の量と性質によってシステムの形態は変ってくるので対象商品のタイプを定める（表３－１）。

オーダー・エントリー・システムの必要度はＡが一番多くＢ，Ｃ，Ｄ，Ｅの順序で弱くなってくる。従ってＡのタイプでは作る製品がオーダー毎に異なるのでそのシステム化は困難になる。

表3－1　商品の種類

商品のタイプ	価格・数量	例
A. 顧客の注文によってタイプがいろいろで定っていない	高価・少量または　1個	住宅・発電機プラント
B. メーカのカタログによっていろいろ組合せる	高価・少量または　1個	コンピュータ
C. メーカのカタログに従うが多少の改造可能	価格中程度，少量	自動車
D. 多種大量生産	中価　大量	家電製品
E. 少種大量生産	安価　大量	食品，化粧品

(2) オーダー・エントリー・システムの目のつけ所

顧客に対する応答が速くなければならないのでオンライン・リアルタイム・システムの採用が望ましいが，オーダーの数が極めて少ない時にはリアルタイム・システムにかかる経費が非常に重荷になる。従って採算性を重要視する。

(3) 設計・製造分野と販売分野とのインターフェイス

工場と営業とはまったく異なった分野・環境を持ち合わせているので両方からデータを集めるシステムは非常に難しい。

(4) マン・マシン・インターフェイス

インプット，アウトプットの関連を十分に考慮に入れる必要がある。特に営業とコンピュータのインターフェイス，つまり端末機器のインプット方式について研究する。

(5) データベースの確立

生産の進捗状況，工場の稼働状況，見積りデータ，技術的情報，顧客先の情報，オーダの進捗状況，販売情報などができる限り統一化されたデータベースになっていることが望ましいので，

- データの収集システム
- データ構造
- ファイル・マネジメント
- 統一データベースの問題点

などを解説する必要がある。このような具体的な例を挙げることが知識となる近道である。

3.2 生産管理システム

オーダー・エントリー・システムの隣のサブシステムは生産管理システムであるが，生産管理システムが身につくには生産現場の経験が必要で机上の研究ではとても無理だと思われる。

従ってここでは生産管理システムを構成するいくつかのサブシステムのうち，次のことについて解説する。

・部品展開システム

・山積み，再山積み（ローディング・システム）

(1) 部品展開システム

部品展開システムにおいては，製品を第1次部品，第2次部品……第n次部品および材料に分解した場合これらが木構造になり，従って部品展開に必要なデータのファイル構造は木構造になる(図3－1）。

いくつかの製品について，これを構成する部品や材料が共通に使える場合が多いが，それらを別々のものとして製品ごとに木構造上定義すると莫大なファイルが必要となる場合がある。そこで，それらの共通なものをまとめて一つのものとして定義できるとファイル•スペースは節約できる。しかし，そうするとファイル構造は単純な木構造では不可能になり，複雑な構造を考えねばならず，プログラム・ステップ数が増える。このようなことが部品展開システムの中の一番難しい所であることを説明する（図3－1，3－2参照)。

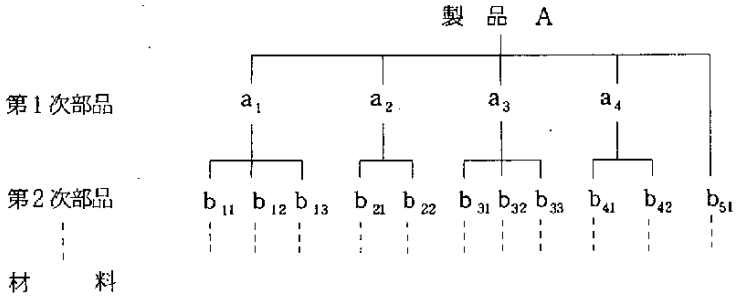


図3－1　部品展開……(1)

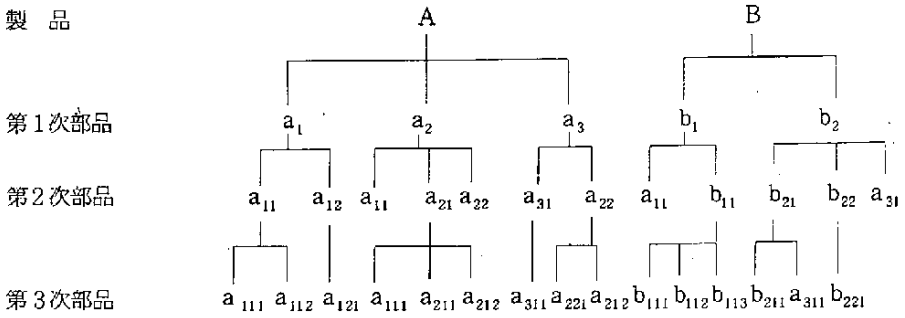


図3－2　部品展開……(2)

次に部品展開ファイルを作り上げるタイミングとインプットの難しさである。製品のオーダーを受けた時，設計図面が完成していなければ部品展開ファイルはできない。設計図面が完成し製造にかか

る間にファイルをまとめなければならないし，場合によってはコーディングも行なわねばならないのでインプット技術とタイミングが大変難しいことを解説する。

(2) 山積み，再山積み（ローディング・システム）

各製造機械や製造グループ毎に仕事をあてはめる計画は仕事の数と製造機械や製造グループの数によって大変複雑になり，この計画を上手に行なうことが生産を合理的に実施する秘訣である。そのためローディング・システムにコンピュータを利用することが企業運営の効果をあげることになるし，ローディング・システム自体の特徴も著るしいので是非説明しておきたいシステムである。

(a) ローディング・システムの問題点

ローディング・システムは生産計画の基礎であるので生産計画のもつ最も重要な問題点，困難な点がローディング・システムにつきまとう。

① 変更・改定の頻発

せっかく出来上ったローディング・チャート（アウトプット）が出来上った時点ですでに古くなってしまうということがありうる。生産管理を経験した人なら誰でも一度は味わうこの問題を説明し，その解決の方法が変更をなくすなどより処理を速くすることおよび処理の頻度を出来るだけ多くすることにあることを説明する。

② 高速・大容量の補助記憶装置の必要性

例えば仕事の数が月100，各仕事が平均20の機械や製造グループの手にかかる，3ケ月間の製造計画を時間刻みで行うとすれば，これに必要な記憶容量は2232000×n だけ必要となる。

$$\text{必要なマス目の数（n 字分のメモリー）} = 100 \times 20 \times 3 \times 31 \times (\underbrace{8}_{\text{一日の作業時間}} + \underbrace{4}_{\text{残業時間}})$$
$$= 2232000$$

n が100とすると2億字分のメモリーを要することになるなどを数字をもって説明する。ちなみに2億字は集団ディスク1台分であるし世の中にはすぐこの位の計画が必要となる。また，これらのデータの取り出しスピードが速くないと処理に大変な時間がかゝることは推察できよう。これも正確な数で示した方がよい。

③ ローディング処理の時間

変更が頻発すると殆んど毎日のようにローディングをし直さなければならないがその処理時間が数時間も必要だとすると他の業務処理に影響を与える。

(b) ローディング・システムとは

ローディング・システムの説明にあたっては図3－3のようなチャートを書いて概要と目的を説明した方が分かり易い。

その場合機械（マシンまたはライン）を中心にしてチャートを作る方法と仕事を中心にしてチャー

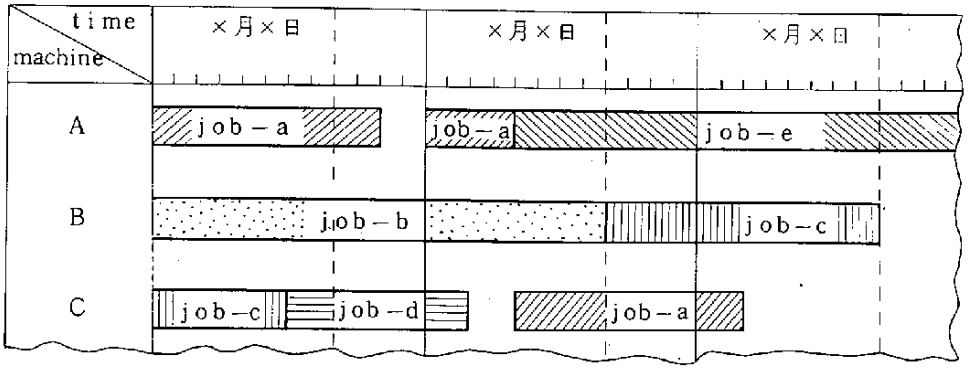


図3－3 生 産 計 画 表

トを作る方法とがあるが前者の方が多少見易いかも知れない。

(3) チャートからデータ・ファイル

チャートは何によって作るかをまず説明しよう。部品展開のとき図3－3にあてはまるような仕事の手順が当然教えられてなければならない，つまり

job—aについてみるとmachine A（14 hrs）—machine C（10hrs）の順で仕事が行なわれ，所要時間は段取時間＋単位当り所要時間×数量ということになる。このような所が解説のポイントになるだろう。

次にデータ・ファイルの上に計画図表を写し換える仕事だがこんな所を無雑作に行なうとデータ量がやたらに多くなり，処理時間もかかるようになるので注意したい。

(4) データ構造の確立

莫大な量のファイルを取扱う必要が生じることから，データ構造の問題，特にインプットにおいてはデイリーまたはそれ以上の間隔で行なわれるが，アウトプットは即時性を要求されるようなデータ・ファイルにとって効率的なデータ構造の探求の重要性を解説する。

この場合ファイルの検索速度にとって都合のよい方法，キーのつけ方，キーと物理的な番地との関係，ポインターなど，プログラミング技術的に興味深い所も説明しておくとよい。

3.3 在庫管理システム

在庫管理システムは在庫をする製品や部品の種類によっていろいろな特徴をもっている。また，その範囲も自動倉庫における在庫管理システムなど，マテリアル・ハンドリングと結合した高レベルのシステムから単なる物の出し入れ管理という単純なシステムまできわめて広い。

(1) 物の出し入れ管理

在庫管理システムの基本的な所は物の出し入れの情報をタイミングよく行なう所であることをまず解説する。物の出し入れとは

当残 ＝ 前残 ＋ 倉入 － 出荷（1品別，場所別）

の方程式で表わされることは簡単で誰でも理解しているが，預り品や引当品の処理をどうしたらよいかというと戸迷う人が多いだろう。しかし物の出し入れ管理というとこのような日常茶飯事な異常情報も取扱わないと現実に存在する物の量を管理することが出来ない。この上厄介なことにはその企業の資産になっていない場合だってある。このような複雑な情報の処理があることを説明しておこう。

(2) 標準発注量，適正在庫量計算

ここでは以下について解説する。

① 単なるデータ処理から科学的手法への転進がこの標準発注量・適正在庫量計算である。

② 倉庫スペースを出来るだけ少なく済ます。

③ 棚卸在庫量をなくすることによって資産を減少し負担金利を少なくする。

④ その上生産や販売に支障を来たさないような在庫量・発注量は，常にどのような状態で製品が作られたり，販売されたりしているかを正確に把握し，それにＯＲ的解析を試みることによって数値的に計算しうる。

(3) 在庫量の把握

データ処理が高度化されると随時に在庫量を知りうるシステム（オンライン・リアルタイム・システム）の作成も可能である。オンライン•バンキング などはこの応用のシステムであるといえる。また商品についてもリアルタイムを志向するシステムが増加しているので在庫管理システムの応用範囲は単に製品在庫の静休的状態の把握ばかりでなく金や人，（人の場合は Skills Inventory など）の在庫管理をダイナミックに行なう所まで範囲を広げて考え解説すると効果がある。

(4) 他のシステムとのインターフェイス

在庫管理システムはそれを単独なシステムとして処理したのでは不経済な場合がある。むしろ他のシステムと連動してこそ効果のあることを下の例により説明する。

① 商品在庫管理システムと他のシステムとのインターフェイス

商品在庫のトランザクション•データである倉入は商品購入システムまたは生産管理システムのアウトプット•データであり，出荷は販売情報システムからのインプット• データだという所に目をつける。即ち図３－４のような関係を実例として紹介する。

② 座席予約システムと他のシステムとのインターフェイス

座席の予約問合せシステムは一種の在庫管理システムであるがこれがやはり座席販売システムや運輸機関の場合には運行システムなどとのインターフェイスのあることを解説してみよう。

③ 銀行普通預金オンライン・システムと支店現金残高把握システム

このような関係を持つ在庫管理応用システムと他のシステムとのインターフェイスの研究も指導テーマとして役に立つ。

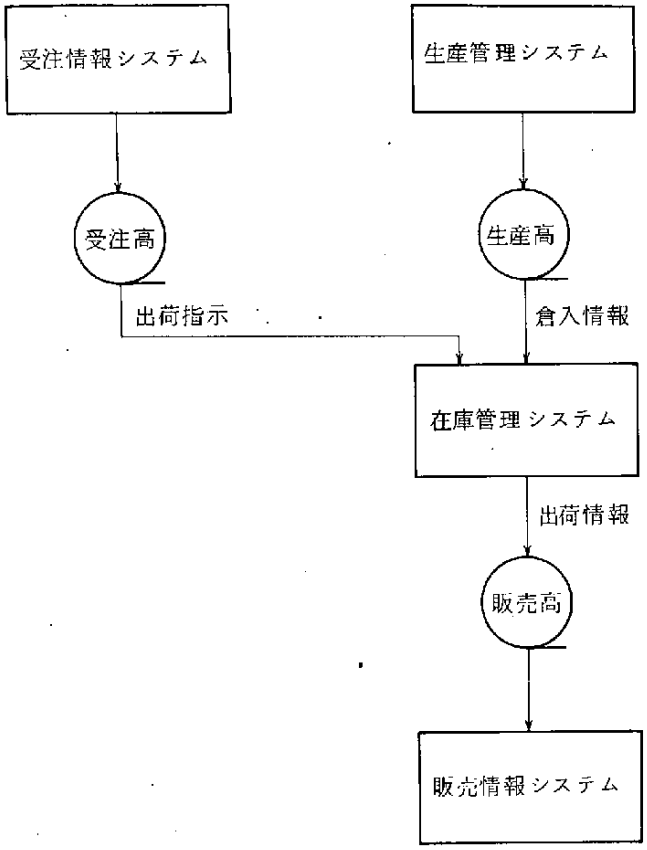


図3－4　在庫管理システムと他のシステムとのインターフェイス

3.4 販売情報システム

単なる販売情報処理システムは，初歩的なアプリケーションとして広く実施されているので，中級情報処理技術者の指導テーマとしては必ずしも適当でないが，その内容を幅広く考えたマーケティング情報システムとして，また他のシステムとのつながりにおいて勉強させることはシステム構成を考える上で役に立つ。

(1) 需要予測へのアプローチ

販売情報を分析し，それに市場の傾向を付加して先々どの位需要があるか，商品別あるいはもっと細分化した形で解析し，明日のデータとする方法を解説する。単なる平均値計算または指数平滑平均計算，少し高級になると関数のあてはめ（最小2乗法の適用）など，そのやり方は種々考えられる。基本的には販売情報処理によって蓄積したデータと，統計あるいは計量経済学的手法との混合の上に成り立つ所であるので，場合によっては専門編「統計解析」「経営科学(II)」などの科目と組合わせるとよい。

(2) 販売活動の分析

販売といってもセールスマンが売りまわる販売，店頭で不特定多数の客を相手に販売するもの，一種類または少数の限られた品種や非常に多数の品種を売るもの，継続的に売れるものそうでないものなど色々な形態がある。

これらの形態の違いはあるが販売活動が何らかの形で伴うのでその活動分析などを取り上げ解説することは異なった面から販売を見るという意味で役に立つ。例えば，

- セールスマンの行動分析（販売能力分析）
- 顧客の嗜好の推移分析
- 店員の接客態度と販売高との関連

などが挙げられるがシステムとしては販売伝票の中に一つのデータ基本項目を追加したり，アンケートをとったり，サーベイをしたり，そのために一つのシステムを作つたり色々できることがあるのでこれらを主に指導する。

(3) 販売情報システムと他のシステムとの関連

販売情報システムは企業の業種の相違によっていろいろなシステムと関連することになるので，教師の関係深い分野について解説するとよい。

会計情報システムは一番関係深いシステムであり，また業種を問わずに必らず関係するシステムであるのでここでは会計情報システムとの接点を特に詳細に解説する。以下の実例を挙げて具体的に解説することが望ましい。

① 情報の一貫性

販売情報システムで出来たデータベースを会計情報システムが100％利用できる。

② タイミング

販売情報システムで処理される時点で会計情報システムがその情報を利用できる。

③ 情報処理の重複の回避

販売情報システムでいったん処理されたものが会計情報システムで再び同じ処理がほどこされない。

(4) 販売情報システムの技術的問題点

販売情報システムでは，情報の収集がもっとも問題になる。即ち，

① 営業・販売店という情報処理をもっとも苦手とする人々にデータを作成させねばならない。

② 営業・販売店は企業の末端であり数が多い。

③ 情報の正確性の維持はまったく困難。

④ 情報収集の速度を早くするためにデータ伝送などが要求される。

⑤ 情報作成者（インプット帳票作成またはインプット端末操作の担当）の教育は非常に根気を要する。（一般に女子の勤続年数は非常に短い）

などの問題解決をしないとシステム開発のメドが全然立たない。そこで，ここではこれらの問題点がよく熟知できるような具体的な解説を行なう。

3.5 人事情報システム

人事情報システムは業種の相違や企業の性格に関連なくもっとも一般的な情報システムの一つであるが，従業員の数や職種などによって影響されることが大である。

必要性の有無に拘らずスキルズ・インベントリなどは情報処理技術者教育のテーマとしては手順なものと思われる。

システム全体の構成としてファイル作成段階（input phase），ファイル更新段階（transaction processing phase），ファイルに対するデータ要求段階（demand processing phase），検索段階（retrieval phase），アウトプット段階（out put phase）の五つの段階に分けて説明するとよいだろう。

(1) ファイル作成段階

従業員に関するどんな情報を集めたらよいかをまず検討する。これにはどんなアウトプットが要求されるかをあらかじめ調べておくと役に立つ。勿論すべて必要な情報が最初から全部わかっていることは考えられないからファイルのデータ構造には柔軟性をもたせるように指導する。

人事情報の性質は，

① 不動項目として，生年月日，番号，氏名，性別，その他のように最初から最後まで変らない項目

② 変更の殆んどない項目として，住所，職種技能，など

③ 数年に一度は必らず変わる項目として，所属部課社内資格など

④ 毎年定期的に変る項目として，給料，考課点，勤務率など

⑤ その他として，性質，希望する仕事や勤務地などのように5種類位のデータ群に分かれるがいずれもせいぜい月単位で情報が正確に分かっていればよいのでファイルは一番最初作成する段階に最も手間がかかるというような所に目をつけさせて説明する。

(2) ファイル更新段階

ファイルの信頼性について，人事情報ファイルはもっとも信頼性を確保し難いファイルの一つである。即ちこれが間違いであると最終的に決定し得るものが非常に少ないという不思議な情報処理をしなければならない。そこでどのような手段でファイルを更新したら正確なファイルに近づくかを例を上げて説明する。

インプット方式をいくら精度を上げてもそのファイルが利用されなければダメなのだが，ファイルが効率的に利用されているとしてインプット時点におけるファイル更新方式によって精度を高めるように工夫させる。

コストを度外視するとインテリジェント・ターミナルのようなものでファイルを更新すれば精度はよくなると思われるが高々万のオーダーの従業員管理をするために余りぜいたくは許されないだろう。そこでコンピュータと従業員との間にターンアラウンド方式を採用すれば精度が上るが，この方式だとコンピュータの使用度合はどうなるかも検討させておきたい。

(3) ファイルに対するデータ要求段階

どの位のファイルができたらどんなアウトプットが要求されるか前もって判っているケースは余り

多くはないが必要な時に必要なデータが早く得られることが要求される，このような要求に対してはコンピュータ・システムは余り強力だとはいえない。工夫をこらさなければならないのはこの辺りであるのでいくつかのアウトプットの例とファイルを挙げて説明する。

この問題の解決方法はいくつかあるが，

① ファイル・マネジメント・システムにおける工夫

② アウトプットが容易であるようなプログラムを準備する（準備の仕方はいろいろあるだろう）

などを挙げて解説する。

次に要求方法の研究をしなければならない。即ち端末機器から要求をインプットするのと依頼書などに要求を書いてコンピュータ室に持込むのとでは本質的に違うだろう。

(4) 検索段階

要求がコンピュータに伝えられると次に検索のソフトウェアが動く。この段階の説明は情報検索の一般的な方法を基礎に指導したらよい。

(5) アウトプット段階

一般には情報要求 → 検索 → アウトプットを情報検索と言っているが，情報システムはそのプロダクトとしてアウトプットを重要視したいのでここでアウトプットの段階を特に取り上げた。

・データの変換

・編集

の二つがアウトプットでは問題になるが，特に最近色々なアウトプット装置が出て来たので効果的な編集をし効果的なアウトプットが出来上るような解説が望ましい。

3.6 経営計画補助システム

経営計画補助として

・資金分配

・予算管理

・製品計画（プロダクト・ミックス）

・キャッシュ・フロー

・人員計画とシミューレーション

・企業モデル

・財務諸表分析

などの中から一，二の実例をとり上げ，データの集め方，データベースの作り方，モデルまたはシステムの作り方，アウトプットの作り方，利用のさせ方，ＰＲの方法などにポイントをおき解説する。

上の一つ一つを正確に理解させるにはそれぞれ相当の時間をさかないと不可能だし，それぞれ得意・不得意の分野もあろうと思われるので，内容の詳細は特に説明をしなくてよいだろう。

指導上の留意点

経営情報システムを知るためには出来るだけ多くの構成システムの内容を知ることが一番近道であり，そのためにできるだけ多くの色々なシステムを本章で説明してほしい。この指導にあたって教師は必ずしも1名である必要はなく個々のシステムの内容を熟知し，できるだけ具体的にくだいて話のできる数名の教師に分担してやってもらうのも一計である。

ただ前後に関連がなくバラバラに話をされたり，一つ一つ業種が異なってくると研修生は戸惑うことになるので教師間の十分な打合せが必要である。

また教師が自分の知り得ない範ちゅうのシステムを担当した場合は適当な参考書で研修生と一緒に勉強するという方法も面白いかと思う。

サブシステムの加工方法については教師に委ねるが，目標にあるように経営情報システム全体との関係において三つの各レベル情報システムと結びつけながら解説してほしい。

第４章　経営情報システム開発（演習）

目　標

本章は第３章迄に講義により勉強した経営情報システムをテーマによる演習で実地にそくして指導しようというのがねらいであり，

① 実際のシステム設計の能力

② コミュニケーションの能力

③ 問題の発見とそれをまとめる能力

を同時に養成し，経営全体にどのように諸サブシステムが作用するかを把握させることを目標としている。

内　容

4.1 演習の手順

経営情報システムの演習は単なるシステム分析設計の演習ではいけない。確かに形態内容はシステム分析設計と似ているが，より混とんとした現象の中から経営に役立つシステムを造り出して行くという点で異なっていなければならない。

演習の手順については次の項目の示す所から一つないし二つの例題をあげてやらせて見ることが好ましい。

① 状況の整理

② システム全体の把握

③ 問題点の発見

④ 情報システム体系の作成

⑤ 可能性の検討

⑥ システム設計

⑦ プロセスの検討

⑧ システム評価

(1) 状況の整理

情報処理技術者が経営情報システムの設計に直面する場合，そんなにきれいな形で問題が整理されている場合は「絶対に」と断言し得る程存在しない。目に写るのはいろいろな問題が雑然と存在している状況で，それをまず整理するという所から始めなければならない。そこで，まず一体どんな状態になっているのか状況や諸意見をまとめておくことが必要となる。

そのために，対人接待には基礎編の「コミュニケーション技法」を使つたり，アイデア発掘にブレーン・ストーミング，ＫＪ法，ワークデザインなどの手法を利用する。ユーザまたは社内のユーザ部門の各々が持ついろいろな考え方をまとめるについては，一定のルールなどないので教師が各自持ち合

わせている手法をこの用途の為役立たせてくれればよい。

(2) システム全体の把握

コンピュータに処理させるということを念頭に置かせながら，今ここで持ち込まれたシステムの端から端まではどうなっているのだろうか，どこまでを考えたら一応満足の行く境界になるだろうかをスケッチさせる。ところで現行システムのスケッチはそんなに容易に書けるものではない。そのための調査だけでも尨大な作業量になるに違いない。しかし演習は残念ながら実際の問題ではないので相手がいない。そのためこの辺りの補足は教師が代理を勤め十分いじめておく必要がある。

ここでの注意は他のシステムとの関連，どのような範囲のシステムと考えているか，などが注目すべき点となろう。

(3) 問題点の発見

このようにして大きいワクは作ったにしろユーザは真実どんな問題点を解決してほしいと思っているのか，身を以って知る必要がある。これは例えば「問題点は最初から挙げてあるではないか」などという不満が研修生から出るかも知れないが，経験をもった情報処理技術者ならすぐ判断できるように，実務の世界では，最初から問題だと言われているのが本当に唯一の問題だというケースよりも，裏に隠されたもっと大きい問題がぞろぞろ出てくる可能性が多く，本当の問題は裏に隠されている問題だということが珍らしくない。

問題点の発見にはユーザ担当とのきん密なコンタクトとシステムの整理とブレーン・ストーミングのような手法による以外にはない。

(4) 情報システム体系の作成

問題点が完全と言えない迄も一応把握でき，システムの範囲と他のシステムとの接合点とがはっきりして来て初めてどの程度の情報システムができ上がるかの体系作りを実施させる。情報システム体系作成にあたってはコンピュータを中心とした情報システム作成の可能性があり，それ自体が問題解決に対して効果があるかどうかを検討させながら，一体どんな情報システムにしたらよいか，その流れはどうなっていたらよいかの概略図を書かせる。この概略図では他の情報システムとの接合点を記入しておく必要がある。

(5) 可能性の検討

概略図ができ上がると次に情報システム確立の実行可能性が次に挙げる諸点から考えられるかどうかを十分に検討させなければならない。「絵に書いた餅」ではシステムを作り上げる価値はない。

① 情報処理技術的に可能か

ソフトウェア的に可能か，一体そんなプログラムはできるものであろうか，などが十分検討されねばならない。

② コンピュータとその周辺機器の能力の上で可能か

一体それが現有の諸機器で処理可能なのか，現在その仕事にあてることができる能力とキャパシティーの上から可能かどうか。

③ コンピュータ要員の数と能力の上で可能か

システムズ・エンジニアの頭の中で考えられたシステムがプログラマやオペレータの数と能力お

よびモラルの面で可能かどうかを検討する。1シフトの作業でやっとと言う状態で，3シフトの作業を行なわなければならなくなるなどのシステムは勝手に作られるべきではない。

④ 経済的に可能か

例えば，月商10億円の会社がいくらよいシステムを開発したからといって月1億をそのためにとって行かれるなどは経済的に不可能と考えられる。話がこうはっきりしている場合はともかく，一般に情報処理技術者は特に経済面を忘れたシステムを考えようとする傾向があるのでこの点注意が必要である。

⑤ ユーザが親しみ易くなっているか

システムは作成し得たがそのシステムにデータをインプットする人や，そのシステムを利用する人がそっぽを向くようでは成功はおぼつかない。

(6) システム設計

すべてプロジェクトが確立した後の手法でこの項はここでは余り深く考える必要はない。（略）

(7) 移行プロセスの検討

システム完成に当り従来のシステムから新しいシステムに変る際の移行プロセス，および新らしいシステムが実際どのように動き，以前働いていた人がどのように仕事を変えればよいのか，など移行に当って必要なプロセスの検討を行なう。

(8) システム評価

演習ではシステム評価を研修生にやらせることは実際難しいので教師による評価に変えて実施することが望ましい。

(9) 演習に有効な手法

この演習にあたって有効となる手法には以下のようなものがある。これらは，他の科目たとえば，「コミュニケーション技法」,「システム分析・設計(I),(II)」などで紹介されているので，適宜利用してほしい。

- 「コミュニケーション技法」
- 「ブレーン・ストーミング」
- 「KJ法」
- 「ワークデザイン」
- 「会議の進め方」
- 「インタビュー技法」
- 「事務分析技法」
- システム分析の諸技法

4.2 例　題

(1) 与えられた環境

(a) 会社概況

・社　名：JIP精工株式会社

・資本金：40億円

・設　立：昭和25年

・所在地：本社：川崎

　工　場：川崎工場，岐阜工場

　営業所：東京，大阪，名古屋，福岡，広島

・業　種：ベアリング製造業

・売上高：年商360億円

・従業員：4,200名

・工場別概況：

川崎工場：従業員2,500名，月産14億，主要製品ボールおよびローラー・ベアリングなど

岐阜工場：従業員800名，月産7億，主要製品特別注文ベアリング各種

・組　織：図4－1，図4－2参照

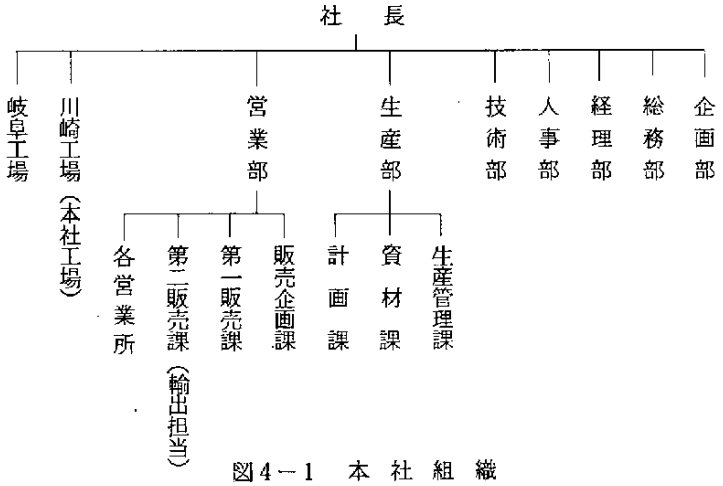


図4－1　本社組織

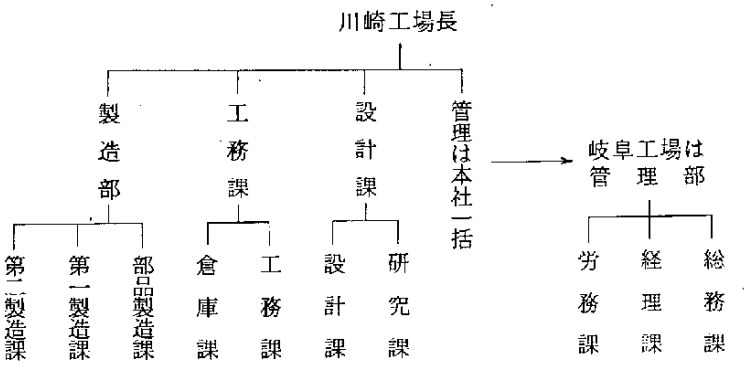


図4－2　工場組織

(b) 当社の情報処理概況

ＪＩＰ精工（株）は，昭和25年に会社設立以来，町工場的な会社の状況から今日ベアリングにおいては，業界5指に入る大メーカになったが，やっとこの数年来何とか会社の形態ができて来たといって差支えない。従ってこれまでの情報処理形態は大変貧弱であった。

一方，会社のこの７～８年の発展は売上，従業員数，組織などの面で爆発的であったので，コンピュータを導入せねば処理できない状態である。勿論経理部，人事部は会計処理，給与計算などを社内外の計算サービス会社に委託している。

(c) 当社の情報処理の流れ

① 受注の形態

当社の販売は全国各地の営業所または本社の販売課から製品の使用得意先へ直接販売される，いわゆる直売と，営業所から代理店経由で販売されるものがある。代理店経由で販売しているものも特約店を経由するものと，そうでないものがある。（図４－３参照）

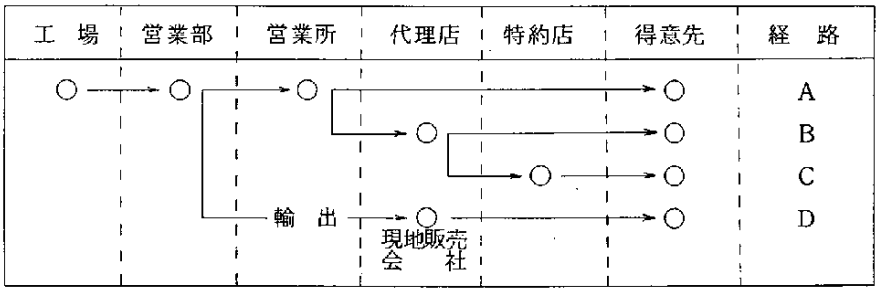


図４－３　販　売　経　路

直売による得意先の数は約６００社，代理店は40社，特約店は50社である。輸出は現地販売会社（20社）を通じて販売している。図４－３は販売経路を図示したものである。ベアリングは国際的に寸法精度の仕様が統一されているものが多く，これを標準品と呼んでいる。特定国の国家規格品や特定メーカのカタログ品で国際的に普及しているものがあり，これを準標準品と呼んでいる。この二者以外のものは特殊品と呼んでおり，この三者の比率は当社では７：１：２位である。受注生産と見込生産は混在していて大体３：７の割合である。生産量の構成比率は表4－1に示されている。

表４－１　販売経路と受注形式

受注形式＼販売経路		営業部直売（図４－３のA）	代理店経由（図４－３のB,C,D）
ライン物	――――	見込生産（40％）	見込生産（10％）
バラ売用	品名ごとに取まとめ可能	――――	見込生産（20％）
そのつど受注		受注生産（25％）	受注生産（５％）

② 受注業務の流れ

・見込生産

各営業所は代理店や大口購入得意先から得た情報および独自の需要予測（営業部のベテラン部員が今まで得た経験を基にして作り出した数字）に基づいて四半期前に販売見込をたて営業部の販売企画課で集計し，品種毎の生産依頼を生産部に提出する。この際，在庫量とにらみ合わせることは当然であるが，この数字を工場の生産能力と調整して計画課は製造命令書を作成する。

・受注生産

得意先から代理店経由または直接営業所または営業部に注文がくる特注品であるが，そのうち自動車メーカなどの大口得意先のものは特注品（受注生産の範囲）に入れず，ライン物と称して見込生産の分野に入れてしまう。従って，受注数量は少なくて，単価は高く，在庫をしていないという範囲のものを称している。

この受注業務は営業が技術部や生産部の協力の元に原価見積，仕様書を客先に提出し受注したら受注票を生産部にまわし製造命令書を作ってしまう。図４－４は，受注から製造命令書が工場に到達するまでの流れを示している。

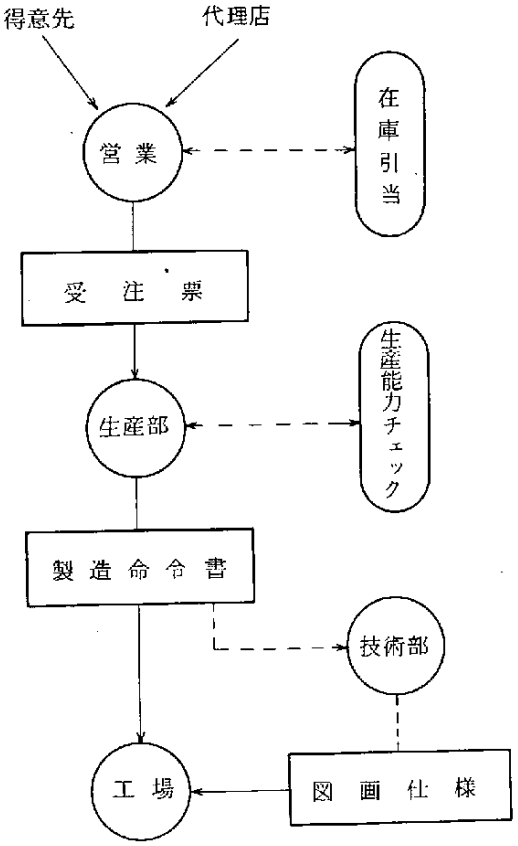


図４－４　受注から工場まで

③ 販売業務の流れ

・営業在庫引当

営業が在庫をもっていてすぐ販売してしまうライン物やバラ売りの場合には営業が納品書（納品書控，請求書，受領書がワンライティングになっている）を発行し，製品の送付と共に請求も出来てしまう。この際，納品書には経理部の出荷承認が必要で，経理部では得意先の支払状況その他を見合わせて出荷承認する。

・注文生産または工場出荷

この場合は，工場倉庫または現場から製品が出て行くので工場の生産高の集計もできなくてはならない（工場にある間は生産されたということにならない，工場棚卸資産の形になっている）。従って工場から出される伝票は営業に対する出荷案内書（納品書・納品書控・請求書・受領書がワンライティングになっている）となっている。

④ 販売情報の処理

・台帳の事務処理

納品書控によって営業部の中の記録担当が得意先別の台帳と日計帳と品種別売上台帳の三種類の台帳に出荷日付，品名，数量，単価，金額，割戻し金額などを記入する。一方，受領書が客先から返ったときには得意先別台帳の受領日付を埋めることによって納品が完了する。日計帳は日別，週別および月別に集計し売上成績の記録とする。得意先別台帳は売掛金請求用として用い，月に1回締めて得意先別の請求書を改めて作成する。品種別売上台帳はマーケティングの資料および生産計画に反映させる資料として用いる。これら三種類の台帳の事務処理が販売情報の処理の主たるものである。

・会計処理

上記得意先別請求書作成は会計処理の一部でもある。経理部では請求と入金とを得意先別台帳上に記録し売掛金勘定としている一方，その消込用にも用いている。得意先別台帳の請求分の内，消込まれていないものは売掛残として残り，あまり支払いがないと改めて未決済処理分として別の台帳に転記させる。入金した中には受取手形が多くを占めている。受取手形は期日別に手形の記録台帳を付けており，しかも取扱い銀行別になっている。当社は企業規模としては中の部類に属し，手形が満期になるまで置いておく程資金繰りが楽ではないので有利な手形から割引く必要があり，手形管理は一つの仕事になっている。他の会計処理として支払いがある。支払い業務は次に述べる買掛金の決済としての支払いと，その他の現金支払い（当社では現金支払と言う）がある。支払いは小口現金，銀行振込，支払手形の三通りの方法により行なわれる。現金支払は全て小口現金か，または銀行振込で行なわれるが，買掛金の決済は買掛残として残す。 支払い方法はその時の資金繰りによって定める。現金支払の手続きは現金支払伝票（3枚組）を発行し，支払照査を受けて出納から現金を受け取ることができるようになっており，その後伝票は会計処理にまわされる。振込でもよい場合は出納窓口から直接銀行に振込まれる。会計処理には非常に多くの伝票が使われており，その仕訳作業はかなり大変な仕事となっている。

⑤ 購買事務の流れ

材料，部品，機械装置，その他物品，用役などの購買は全て本社の生産部資材課で行なっている。

工場や本社の各部課が物品の購入をしたい場合には，購買請求書を切って資材課に回付する。資材課は業者との間で見積りを完了して，単価，金額，納期などを決定して注文書を発行し，注文台帳に記帳する。物品その他が納入されると注文台帳を消込むと同時に取引先別資掛金台帳に記入し，物品購入伝票（会計処理用）を作成する。取引先別買掛金台帳は後刻経理部で買掛金処理に使われ，支払に結びつくが，物品購入伝票は資産勘定処理に用いられる。当社はあらゆる点で伝票会計と大福帳処理とが二本立てのような形になっているので大変複雑な事務処理である。しかし納期遅延は取引先別買掛金台帳を見ればすぐ分るし，処理は面倒だが管理には具合がよい。

(d) 情報の種類と量

① 見積要求票

営業発行，件数　月間　2,000枚，様式は図4－5参照。

見積要求票

見積要求票　　A営業所控

品名		見積計算		
		得意先名		
個数	原価	希望納期	回答納期	営業所
主要寸法				扱店
備考				予定売価
				承認

図4－5　見積要求票の例

② 受 注 票

営業発行，件数　月間　1,500枚，様式は図4－6参照。

受 注 票（需要票もこれに準ずる）

受 注 票　　A営業所控

品名		工場	用途	発行日付
品名コード	グリース	得意先名		得意先コード
個数	金額	納期	分納完納	営業所
製造番号				扱店
				送先
				担当

図4－6　受注票の例

③ 納品書（控・請求書・受領書）

営業発行（これに出荷案内書の付加したものは工場発行），件数　月間　2,500枚，

様式は図4－7参照。

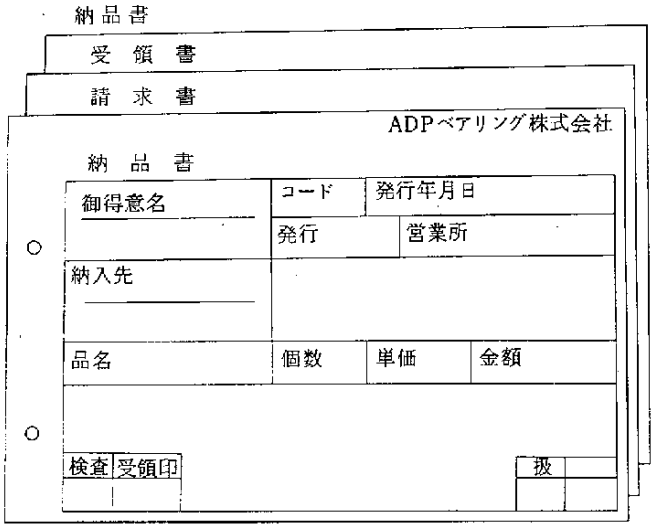


図4－7　納品書の例

④ 製造命令書

生産部発行，件数　月間　800枚，様式は図4－8参照。

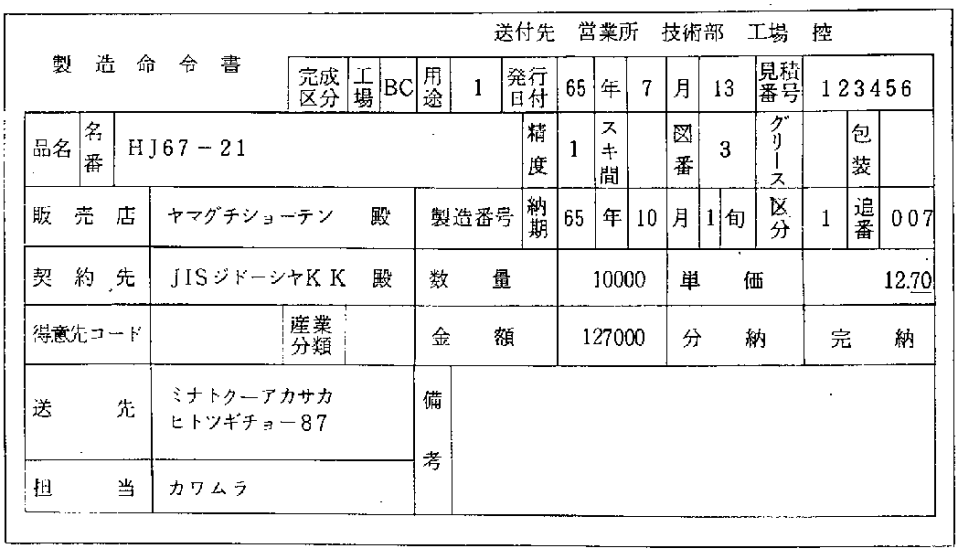


図4－8　製造命令書の例

(2) 当社の方針・施策と問題点

(a) 方針・施策

最近における企業環境は益々激化する企業競争，高騰する人件費，縮少される労働市場，高騰する資材費，この所激しくなってきた円の変動などに起因する販売価格の変化などきわめて厳しくなってきている。先に述べた当社の状態では，これらの諸問題に対処し切れなくなってきていることは明らかである。そこで，このように判断した経営トップは常務会においてコンピュータを導入し，経営・事務をシステム化する以外に道はないと判断した。

しかし残念ながらこの会社ではコンピュータやシステム化について経験をもったものがまったくいないので，企画部長に調査を命じた。企画部長は信頼がおけるＤＩＣコンサルタント（株）にこの問題を全面的に依頼し，当社の経営の体質改善をするべく新しい経営情報システム確立の計画作りを依頼した。ＤＩＣコンサルタント（株）は優秀な経営コンサルタント並びにＥＤＰコンサルタントを派遣し種々の調査を行なった結果，次の諸点をまとめた。

① 機械化要員が皆無なので，すぐ高性能のコンピュータを社内に設置することは難しい。

② 全社を対象にするのはいっきょには困難，まず本社を対象に第１次経営情報システムを確立すべきである。

③ 当社に許される機械化経費は月間３０００万円を限度とする。従ってコンピュータの賃借料は１０００万円前後。この程度を目標に社外の計算センタに処理を外注する。

④ システムズ・エンジニアを数名雇用する。

⑤ 機械化を進めるにあたりＤＩＣは当初極力応援するが，２年後にはＪＩＰ精工が１人歩きできるようにする。

⑥ 第１次経営情報システムの目標は２年後とする。

⑦ ＪＩＰ精工の問題点は次項に述べるが，まったく整理ができない状態にあるので本計画を進める組織作りにはＪＩＰ精工の優秀な人材を集め，会社百年の計を作るべく活躍してほしい。

⑧ 本計画の推進委員会の委員長は社長または副社長でなければならない。

以上のコンサルティング報告は常務会において全面的に採用され，社長を長とし企画部長を副とした経営情報システム推進委員会を設立した。

(b) 問題点

① 標準品では数多い在庫品の管理（約２３００品種）が十分でなく，特にある営業所では在庫切れになっていても他の営業所には山積みされている。このような状態で注文を逃がしたりするケースを解消したい。また，売れていないものを沢山製造して在庫が多く，一方在庫の不足しているものを作らなかったりすることも同時に解消したい。

② 特注品の場合には以下のような問題がある。

工場の生産日程状況が不明なことが多いので納期を確認できない。

・見積価格算出に時間がかかり過ぎるため入札などの場合利益が確保できるかどうか不明のことが多い。

・工場の生産過程のバランシングがうまく行かない。暇なときと忙しいときとの差が著しい。

③ 生産工程の進捗状況が営業・生産両部で適確に把握できない。

④ どうも棚卸資産（製品・材料共）が多過ぎ，そのため余計な資金を寝かすことになる。

⑤ 販売損益の算出に時間がかかり，儲っているかどうかが不明のままビジネスをする破目になる。そこで，できれば販売損益は月末後5日以内に知りたい（現在は25日頃）。また，売上高だけは手計算で月末後6日頃判明しているがこれもできるだけ早く知りたい。

⑥ 請求書作りもさることながら，入金して来たものが一体何の請求に対して入金したのか不明のものがあり，売掛金管理が十分には行なわれていない。これはぜひ厳密に管理したい。そうでないと当社の管理の「ずさんさ」から不良債権ができてしまうという得意先には申訳のないことが絶えない。

⑦ 買掛管理もずさんである。大体当社では購買請求を出さなければ物を買えない規定なのにもかかわらず，物が納品されてから請求書が切られるというずさんなことが多いのが現実である。

⑧ 納期管理はまったくできていない，というよりどれが納期遅れでどれが納期遅れでないのかも不明である。従って現場では常に余計なものをもっていて生産に対処しているので棚卸資産がかさむ。

⑨ 購入品の単価の管理もずさんである。当社が購入している主なものは鉄，特殊鋼材，（外輪，内輪，コロ，間座，つば輪などの材料）半製品として転動体，保持器，リベット，シール，ケース，止め輪（図4－9参照）など品種はそう多くない，にもかかわらずこのような状態では困る。

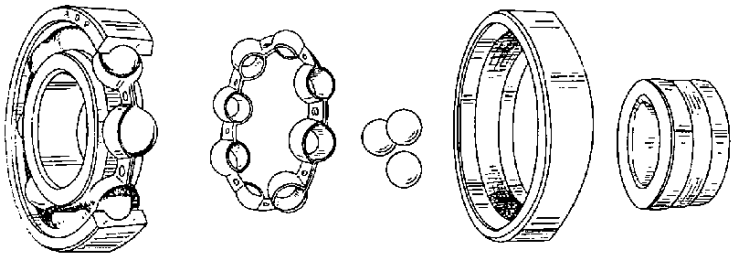


図4－9　ボールベアリング概略図

⑩ 製品の種類が多くどんどん増加する傾向にあるが，このままでは管理限界を過ぎてしまうし原価も上るだろう。

⑪ マーケット・リサーチを行ないたい。特にベアリングは他産業の発展とかなりの相関関係がある筈だからやりよい筈だが，現在では手がつかない。また当社の業界における地位を上げねばならない。

⑫ 工場，営業間の情報交換はきわめて不十分である。

この他にも多くの問題点がある。特に工場の生産に関する問題点は後々の話の都合上省略した。

(3) 当社の機械化プラン

以上の背景をもとに委員会は本格的に活動を開始し，次のような機械化プランをたてた。即ち，

① 本社事務の機械化を中心に進め関連ある所については工場の情報処理も進めるが，当社の能力で

はいっぺんにはできかねるから，できるだけ第2次計画にまわす。

② コンピュータは本社工場の近くにある計算センタの大型機を借り当社に端末用小型機を導入してリモートバッチ処理により行なう。

③ センタの大型機の能力は，当社にとって無限と考えてもよい。時間当り利用料金は当社の場合特に安く200千円となっている。

④ 当社設置のリモートバッチ端末仕様は

ライン・プリンタ	300行／分
カードリーダ	300枚／分
カードパンチ	100枚／分
OCRリーダ	200レコード／分

⑤ 通信回線は計算センターとの間に4800bps 1回線を専用で借り各地方営業所との間はテレックスと結び，岐阜工場との間は200bps の特定回線で結ぶ。

⑥ 情報処理要員については次のようにする（表4－2参照）。

表4－2　人員計画表

	採用／転用（第1年度）	第2年度	第3年度
管理者	転用1		
S・E	採用5（経験3年）	採用2（新人）	採用2（新人）
プログラマ	採用5（経験2年） 転用3	採用3（新人）	採用3（新人）
その他	転用2（女子）	採用2（女子）	採用1（女子）
計	16名	23名	28名（1名退職）

⑦ 経費については，表4－3の通りである。

表4－3　年度別月額経費　（単位：千円）

月額経費	第1年度	第2年度	第3年度
人件費	1920	3220千円	4360千円
計算センタ使用料	4000	5200	8000
通信回線借用費	450	500	500
消耗品費	500	700	1000
データ・エントリー費	500	1200	2000
その他経費	300	400	500
計	7670	11220	16360

⑧ 設備費用については計算センターを借入れるためほとんど不要である。

⑨ 第2次計画において第1次計画を反省し，その効果を概算する。コンピュータを設置する，しない，は第5次計画において定める。

⑩ 第1次計画においては本社の経営情報システム化を目標とし，できるだけトータル化を目指して機械化をする。目標2年間。

以上の計画は常務会において可決されＤＩＣコンサルタントの手で必要なＳ・Ｅ，プログラマを集めてこの計画の実行が開始された。

指導上の留意点

(1) ＪＩＰ精工（株）の経営情報システムの概略図をＤＩＣコンサルタント会社の経営情報システム専門コンサルタントになったつもりで作成する。

その場合，ＪＩＰ精工（株）へのリコメンデーションを書くつもりで作成するが，その要件としてＪＩＰ精工には専門家が1人もいないということを念頭におく。

(2) コミュニケーション技法を用いて設計したシステムを発表させる。この際，本システムの効果についてはできるだけ具体的にさせる。

(3) 問題点がどう解決されるのかについても発表させる。この際，ＪＩＰ精工の問題点がどのように整理されているかを述べさせることも演習の効果の一つである。

(4) ついで本社業務を中心としたサブシステムの詳細設計にあたらせる。これらのサブシステムはシステム全体の概略図に明示されているものが中心となるが，詳細システムの設計にあたって多少の追加システムがあってもかまわない。演習時間が十分にとれるならばすぐプログラミングができるほど精密にできていることが望ましい。

(5) サブシステムについては現場の担当者がインプットする際のマニュアル，結果のアウトプットの見方のマニュアルが完備していることが望ましい。

(6) システム間のインターフェイスがきちんとできているか詳細にチェックする。あるサブシステムからアウトプットとして他のサブシステムに渡したものが受取られていなかったり，またその逆では困る。できるだけ合理性に富んだシステムが望ましい。

(7) あらゆる点から実行可能性の検討がなされていなければならない。要員，経費，コンピュータ，資源などの面から検討されている必要がある。

(8) 移行する場合の問題点も十分に考えられている必要がある。

全体としてシステムが古い体系から新しい体系にどう変って行くかが明示されているのも重要なポイントである。

参考文献

(1) 吉谷竜一著，「システム設計」日経文庫。ワークデザインについての参考文献。

(2) 川喜田二郎著，「発想法」，「発想法（続）」中公新書，ＫＪ法の参考書。

(3) オズボーン著「独創力を伸ばせ」，上野一郎訳，1958，ダイヤモンド社，ブレーン・ストーミングの参考書。

(4) 「事務管理ハンドブック」技報堂，事務分析技法に役立つ。

(5) 日本アイ・ビー・エム「Study Organization Plan」，システム分析技法に役立つ。

(6) 日本事務能率協会偏「MISハンドブック」経営出版会

(7) 日本アイ・ビー・エム「経営情報システム」

データ・ベース

目　次

第1章　概　説 …… 284

1.1　定　義 …… 284

1.2　発展の歴史とデータ・ベース・システムの実例 …… 285

1.3　データ・ベースによる処理と今までの処理，データ・ベースの導入 …… 285

第2章　データ・ベースの諸機能 …… 289

2.1　データ・ベースの生成 …… 289

2.2　データ・ベースの更新 …… 288

2.3　データ・ベースの検索 …… 288

2.4　データ／装置独立性 …… 289

2.5　プログラミング言語とのインターフェイス …… 290

2.6　データ・ベースの保護と信頼性 …… 290

第3章　データ・ベースの構築技術 …… 293

3.1　順編成ファイル …… 293

3.2　ランダム・ファイル …… 294

3.3　オペレーティング・システムとの関係 …… 297

第4章　データ・ベース管理者機能 …… 299

4.1　設計，開発機能 …… 299

4.2　データ・ベースの管理 …… 299

4.3　オペレーション …… 300

4.4　モニタリング …… 300

4.5　システム改良 …… 300

4.6　演習問題 …… 300

第5章　ユーザ・インターフェイス …… 301

5.1　ユーザの種類 …… 301

5.2　ホスト言語と自立型言語 …… 301

5.4　基本データ操作機能 …… 304
5.5　障害対策 …… 306
第6章　データ・ベースの標準化 …… 307
第7章　具体例 …… 309
第8章　データ・ベースと情報検索 …… 310
8.1　概　要 …… 310
8.2　情報の蓄積 …… 311
8.3　情報の検索 …… 312
8.4　具体例 …… 312

科目「データ・ベース」

教育の目標

データ・ベースによる処理でも，従来のデータ処理による方法でも外観上著しく異なる処理を行なう訳ではない。しかし，そこでなされている作業は，従来の処理と質的に異なるものであり，内部的にそれをサポートしているソフトウェア・システムも異なった環境の下に置かれている。

本科目では，まず第1章で実際の必要性からなぜデータ・ベースという概念を持ち出す必要があるのかをまず理解させ，次に第2，第3章でその目的のために技術的にどういった類の考慮をする必要がるのかを説明する。また，データ・ベースを実際に運用するときに必要とされる諸配慮について第4章で触れ，次いで第5章で実際にユーザがデータ・ベースを使う方法について説明する。

第6章では現在，データ・ベースをかってのCOBOLのように，標準化しようとしている国際的な動きについて説明を行ない，将来の展望を行なう。

第7章は既存のデータ・ベース・システムの研究，あるいは著名なアプリケーションについてのケース・スタディである。

本科目は，中級情報処理技術者のコースB及びコースCに該当する人のための育成指針であり，説明項目に対しての時間の配分などに関して一応標準案を提示している。

しかし，実際の指導に当っては，このコースに必ずしも拘わることなく，目的に応じて最適なトピックを選択することが望ましい。

時間配分

章	履習時間（時間）			
	Bコース		Cコース	
	講義	演習	講義	演習
1. 概説	4	4	4	4
2. データ・ベースの諸機能	6	0.5	6	0.5
3. データ・ベースの構築技術	10	7	15.5	8
4. データ・ベース管理者機能	2	0.5	1.5	0.5
5. ユーザ・インターフェイス	6	4	8	4.5
6. データ・ベースの標準化	5	2.5	5	2.5
7. 具体例	10	6.5		
8. データ・ベースと情報検索	17	5		
合計	60	30	40	20

第 1 章　概　説

用　語　この章では次の用語を教える。

データ・ベース，データの非冗長性

目　標

まず，データ・ベースの定義を与えるとともに，データ・ベースによる処理と今までの処理，データ・ベースの導入で実務上，何故従来のファイル処理では足りなくて，データ・ベース的なアプローチが必要になったかを，具体的な例を通じて理解させる。

また，データ・ベースの発展の歴史，汎用データ・ベース・システムの実例を通じて，内外で開発されている著名なソフトウェア・パッケージについて理解させるとともに，実習を行なうことによって，研修生が身近なものとして，データ・ベース・システムの種々のvariation を知り，共通の特徴を抽出できるように仕向ける。

さらに，データ・ベース的なアプローチが必要とされる具体的なアプリケーションについて説明し，従来の方法で同じ目的を達成するのにどのような不便があるかを理解させる。

内　容

1.1　定　義

データ・ベース（data base）の定義について万人が認める明確な定義というものは恐らく存在しないであろう。しかし，ここでは一般にデータ・ベースというものに情報処理技術者がどのような機能を期待しているかを何項目か列挙して，以下の議論では，そのような条件を満足するデータ処理システムをデータ・ベースと定義することにする。

ここでは，データ・ベースの機能としては次のものを考える。

① **データの非冗長性**（data non-redundancy）

② データ間の関連性，整合性等の一元的管理の容易さ

③ データ処理の安価かつ容易な標準的な手段の提供

④ データ・ベースを利用するプログラムのデータ独立性

⑤ 蓄積データの完全性（integrity），機密保護，信頼性

教師は，以上の機能が第2章以下の研究項目とどのような関連を持つか概要を説明する。

1.2 発展の歴史とデータ・ベース・システムの実例

(1) 発展の歴史

データ・ベース・システムが実際にどのような必要性の下に開発されて来たか，また，現実にはどのような役割がデータ・ベース・システムに要求されるのかを現実のシステムによって考察する。

CODASYL (The Conference on Data Systems Languages) によるデータ・ベースの標準化の話は，第6章で触れられるので，この節では対象からはずす。

(2) データ・ベース・システムの例

データ・ベース・システムは，汎用のパッケージ化されたものと，個々のユーザの下で稼動されている具体的なシステムとに分類して考察するのが適当であろう。

汎用パッケージは，数多くのデータ・ベースの利用例からそれらに共通する，本質的な機能を抽出してでき上ったもので，たとえば参考文献(1)，(2)などに適切な紹介が載っている。

個々のデータ・ベース・システムの運用例ではそのアプリケーションに特有の性質が強調されるあまり，データ・ベースの本質を掴むのに手間取る可能性があるが，汎用パッケージにはそのおそれがすくないので，まずデータ・ベースの正統的なやり方を理解するという意味で，汎用パッケージを知る必要がある。

IDS (Integrated Data Store) は，最も古くから利用されている汎用パッケージの一つで，その後のデータ・ベース・システムに与えた影響（たとえばCODASYLの標準化案への影響）の大きさから考えても，説明の価値は十分にある。

MARK IV，GIS (Generalized Information System)，IMS (Information Management System)，TDMS (Time-shared Data Managemt System) などを，データ・ベース・システムのいくつかのアプローチの代表例として説明する。これらについては，上述の参考文献(1)，(2)に依るのがよかろう。

以上の汎用パッケージのほかに，実際，データ・ベース・システムを企業その他でどのように実現しているかについて，実例の報告などを基にして説明する。

ここでは具体的な業務がどのようにデータ・ベース・システムに表現されているかそれらの関係について理解させる。

できるならば，上記の汎用パッケージを利用した実例について述べるのが適当である。

(3) 演習問題

(a) 参考文献の(1)あるいは(2)を参考にして，上記汎用パッケージの概要を報告させる。

(b) 身近なコンピュータで利用できるデータ・ベース・システムあるいはファイル操作用のシステムを調べ，その設計思想の概略について，本文であげた汎用パッケージと比較させる。

1.3 データ・ベースによる処理と今までの処理，データ・ベースの導入

データ・ベースがなければ，データ処理が一切行なわれないということはない。今までにも，データ・ベースとは無縁に，大規模なデータ処理は行なわれて来ていた。

この節では，どのような必要性から今までのデータ処理と異って，データ・ベース的なアプローチが必要となって来たかを説明する。

本書で考えるデータ・ベースの機能は，先に，1.1定義の項で述べた約5項目のものであるが，これらの機能が，今までの処理ではあまりうまく扱えなくて，データ・ベース的なアプローチを必要とした理由としても考えられることを指摘する。

具体的に1項目ずつ，従来のやり方でうまくないと考えられる点を説明する。

例えば，1.1であげた①の非冗長性（Data non-redundancy）については，従来の方法が順編成ファイルを基盤にしたものであることから，異なる分類キー（SORT KEY）を扱う必要のある場合には，どうしてもデータの冗長性が排除できなかったことを，わかりやすく例を用いて説明する。

1.1の②であげた，データ間の関連性，整合性，一元的管理の容易さについても，従来のシステムでは，データを統括管理するメカニズムが欠如していて，ある限界以上の管理は，人手にまかされている事実を指摘するとよい。これらに対して，データ・ベース・システムが用意する解決策についての詳細については，第2章以下で展開されるわけであるが，ここではその概要について述べ，第2章以下の導入を容易にするように試みる。

1.1で述べた5項目のデータ・ベースの機能は，現在のハードウェア技術，ソフトウェア技術で十分安価に解決されているとは思われない。従って，現実にデータ・ベース・システムを導入する場合には，導入に伴うコストに見合うだけのメリットがなければならないことを注意する。

このような事情をよく踏まえた上で，データ・ベース・システムの導入方法，評価等について理解させる。[3],[4]

指導上の留意点

(1) 第1章は，第2章以下の説明のまとめになっている。教師は，第2章以下の展開を考慮した上で，必要とあればそれらの概要を説明するのが望ましい。

(2) データ・ベースの技術は，現時点ではすべてのデータ処理にとって替る程，強力なあるいは安価なものとはなっていない。データ・ベース・システムの得意な分野，不得意な分野をあらかじめ説明して，誤った先入観念を植えつけないようにすべきである。[4]

(3) (2)とも少し関連するが，データ・ベースは，何の必要性があって生れて来たものかということを理解させる。必要性が大であることを悟らないと，学習意欲は，あがらないものである。

第2章　データ・ベースの諸機能

用　語　この章では次の用語を教える。

データ・ベースの生成，データ・ベースの更新，装置独立性，バインディング，機密保護，パスワード，エラー・リカバリ（障害復旧）

目　標

データ・ベース・システムといわれるものには，共通した諸機能が見られる。これらの基本的な機能について理解することはデータ・ベースをどの応用分野に適用すべきか，或いはあるデータ処理をデータ・ベース手法を用いて解決するときにどのような問題が生じるか等の問に答えるために必要なことである。

データ・ベースには通常のデータ処理の場合と同じく，更新，検索の機能がある他に，これと別に生成の機能がある。また，データ・ベースは大量のデータを長期間使用する必要上，プログラムのデータ或いは物理的な装置からの独立性がうまく保たれるよう工夫されているが，これらについて2.4で説明する。

また理想的には，データ・ベースがどの言語ともインターフェイスを持つべきことを2.5で説明する。これらの機能については，現実のシステムで満足のいく程度の機能を有しているシステムはあまりないから，データ・ベースの持つべき機能として説明する。

2.6のデータの完全性，機密保護，信頼性についても同様に，現状ではかなり不満足なものしかないが，データ・ベースの本来持つべき機能，必要な機能として述べるのがよい。

内　容

2.1　データ・ベースの生成

データ・ベースには普通，データを蓄積する機能が存在する。これを広義のデータ・ベースの更新機能と同一視できない訳でもないが，データ・ベースを創設するときには，そのためのみに特別に設けられた機能によるのが普通である。

これは，データ・ベースと称されるシステムは，普通複雑な物理的構造を持っていて，小規模の変更に対しては適応できるが，データ・ベースそのものの生成というような大量のデータ投入に対しては適応し切れない（できたとしても効率が極端に悪くなる）からである。

たとえば索引付順編成ファイルを例にとって見ても，レコードの追加，変更が度重なると，検索あるいは更新にとって非常に悪影響を持つことが説明できる。

これらの事実をよく理解させ，実際にデータ・ベースを扱うときには，**データ・ベースの生成**あるいはこれに準ずるような大量のデータ投入の際にはデータ・ベースの生成機能をフルに活用する必要性を強調する。

2.2 データ・ベースの更新

データ・ベースに対して何等かの変更を施すことを更新ということにすれば，2.1で述べたデータ・ベースの生成も確かに更新の一種であるが，ここではそれ以外の**データ・ベースの更新**を考える。

この種の更新は次のように分類される。

① 新しいデータを追加するもの

② 既存のデータを削除するもの

③ 既存のデータを新規データで置換するもの。

この③は更に既存データと新規データのサイズを比較して

- 新旧のデータ・サイズが等しいもの
- 新旧のデータ・サイズが等しくないもの

の2種類に分類される。またさらにデータ・ベース中に存在するデータには普通，レコード単位にキーと呼ばれるレコードを識別する情報が含まれているが，このキーの値を

- 変更する，あるいは
- 変更しない

に従って，2種に分類できる。

つまり，③の場合には合計4種類の更新が考えられ，しかも以上のおのおののデータ・ベースに与える影響は微妙に異るから，その差をはっきりと認識させる。

更新の際には，更にデータ・ベースの完全性の問題，あるいは排他制御，それに起因するデッド・ロックの問題があるが，これらについては，再びそれぞれの項で問題にすることを述べ，ここでは単にデータ・ベースの更新には共用の際に難かしい問題が存在することを指摘するだけに留める。

2.3 データ・ベースの検索

データ・ベース・システムが有している検索(*)機能を次の二つに大別して説明する。

① 特別の条件式を与えることができ，その条件に合ったデータをデータ・ベースの中から取出す能力を備えているもの。

② システム側は，基本的な検索機能だけを持っており，より複雑な検索は利用者が自分で行なうもの。

具体的には①の部類に属すシステムとして，MARK Ⅳ，GIS，TDMS，②に属するシステムとしてはIDS，IMSなどをあげ，それらの利害得失を説明する。データ・ベース・システムの現在の技術段階では，あらゆる検索条件を適切に表現できる手段がなく，従って汎用の目的のためには②のシステムが不可欠であることを説明する。

また，表面上は色々の検索条件があっても結局は現在のところ，ハードウェアとソフトウェアの技術上の限界により，第3章で述べるような構築技術が採用されているから，実際の検索を行なう際には，背後で何が行なわれているかを知っておく必要性を強調する。

(*) ここで述べる検索という言葉は第8章でいう情報検索よりせまい意味で用いられている。

検索のために利用されるデータへのアクセス法として

① データ・エントリの物理的な順序によるもの

② 単数あるいは複数個のインデックスを設けて行なうもの

③ ポインタを有したリスト構造あるいはネット・ワーク構造を利用するもの

④ ランダマイジング（第3章2.3で詳細は述べる）などを利用して直接データのありかを知るもの

などがあることを簡単に説明し，誤ったアクセス法を用いぬための注意を与える。

アクセス法と関連して，検索の効率はデータの蓄積(*)方法といかに深い関係があるかを説明する。

特にデータ・ベースのデータは蓄積（生成あるいは更新）に有利になるように設計すると検索に不利となり，逆に検索に有利となるように設計すると蓄積に不利となることが多いとを指摘し，業務の性質がどちらを重視しているかなどの条件もデータ・ベース・システム作成に影響することを述べる。

データ・ベースをどのように設計し，実用化するかについてはシステム自身の開発過程，システムに実際のデータをのせて稼動させる過程のいずれにおいても，多数の相反する要件を考慮して決定しなくてはならないが,その一つの例としてこの蓄積と検索を例にして話してもよい。

2.4 データ／装置独立性

データ／**装置独立性**とは，プログラムがデータ／装置の変化にどれだけ耐えるかということである。

まずプログラムがデータの変化あるいは装置の変化によってどのように影響されやすいかという事実を説明し，次にプログラムの変更ということが如何に高価な犠牲を伴うものであるかということを説明する。データ・ベース・システム以前にもこれらの不便を取除くためにどのような工夫がされて来たかを簡単な次のような事務処理を例にして理解させる。

「①カードを読みこみ，②フィールドA（1～10けた）と③フィールドB（11～20けた）にパンチされている数の和を計算し④プリンタ上に⑤×××・×というフォーマットで出力する」

この例でアンダー・ラインを施した部分はデータあるいは装置が変化することによりいろいろと変りうることを説明し，従来のデータ処理がこれらの問題をいかに解決して来たかを説明する。

データ・ベース・システムではデータそのものが複雑化し，使用装置も大規模化することから，なお一層これらの独立性が要求される必要を指摘し，これらが5.3で述べるデータ・ベースの定義の方法により，プログラムとデータとの**バインディング**（結合）をより間接的にすることにより，一部達成されていることを説明する。また一般的な傾向として，データ・ベースに要求する事柄‘what’を実現する方法‘how’に何種類もの方法があり，それらが効率あるいは参照頻度などの考慮から，しばしば変更されやすいことを例にとって‘what’のみを記述するプログラムは，独立性が高く‘How’を記述するプログラムはこれに反して独立性が低いことを説明する。

＜演習問題＞

(a) 上にあげた簡単な事務処理をアセンブラ，COBOLでそれぞれ書いた場合，変化に対して,どちら

(*) 第8章でいう情報の蓄積よりせまい意味で用いられている。

がどのようにプログラムの手直し部分が少くてすむかを述べさせる。また，装置からの独立に対してオペレーティング・システムの果す役割について議論させる。また複雑なデータ構造を扱うデータ・ベースでは従来のデータ管理，ジョブ管理でどこにどのようにプログラムの独立性から考えて困難があるかを考えさせる。

(b) 簡単な部品管理の例題を与えてそれをIDSあるいはDBTG（第6章参照）で書いた場合と，これを述語論理などで記述した場合を比較させる。[9],[16]

このいずれが'What'を記述したもので，いずれが'How'を記述したものであるかを問う。なお，この問題は，かなり程度が高いので，最初は必ずしもやらなくてよい。

2.5 プログラミング言語とのインターフェイス

いくつかのデータ・ベース・システムは複数個の言語から利用できるようになっている。このような必要性を強調し，そのようなシステムの一般的なやり方をDBTG提案などを例にとって説明する。

2.6 データ・ベースの保護と信頼性

(1) 完全性（integrity）

データ・ベースは更新機能を認める限り，常に不当な動作によって破壊される危険性を持つ。

このような完全性が失われるケースを

① 誤操作によって，データを更新する権利を持つ者がデータ・ベースの内容をこわす。

② システムの完全性のチェックが不備なため，相互干渉などの結果，内容がこわれる。

の二つのケースに分けて説明する。

①は原則として防ぎようがないが，システムとしてそのような誤りが容易に生じる性質のものと，仲々生じにくいものとがあることを説明し，誤りが犯しにくいシステムの作り方，運用方法を説明する。

更新手続きが複雑なシステム，チェックが不備なシステムなどを悪い例として説明する。[13]

②は主にデータ・ベースを共有してオンラインで更新ジョブを流したときに二重更新などの問題で生ずるもので，排他制御あるいは使用規約が必要であることを指摘する。

排他制御については，第3章3.1で詳しく触れるので，ここではDBTGで新しく提案されている使用規約について説明する。

（これは，'71 DBTGのKEEP verbあるいは'73 DBTGのmonitor modeの話である。）

(2) 機密保護

機密保護は，残念ながら現在のデータ・ベース技術では完全に行なわれているとはいえない。

この結果，現段階ではどの程度の制限があるかを教え，データ・ベース・システムの機密保護の限界について説明する。

現在，採用されている機密保護の方法，例えば**パスワード**等により，データの一部あるいは全部のアクセスを制限する方法について説明する。

(3) 障害対策

ここでは，データ・ベース・システム側で用意される**エラー・リカバリ（障害復旧）**の手段につい

て説明をする。データ・ベースの障害対策は，これだけでは十分でなくユーザ側も十分の注意を払わなければならないが，これについては，5.5で再び触れる。

まずはじめに，データ・ベース・システムにおいてどのような障害が生ずるかを具体的な例で説明する。（表2－1参照）

表2－1　障害原因の分類*

1. ハードウェア上の障害
   - CPU
   - 周辺機器
   - 回線関係
   - 入出力媒体
2. ソフトウェア上の障害
   - システム・プログラム・バグ
   - アプリケーション・プログラム・バグ
   - デッド・ロック
3. 運用上の障害
   - ミス・オペレーション
   - 不当データの投入

* 電源，空調等の施設上の障害は含まれていない。

これらの障害対策が不備であれば，データ・ベースの運用にとってどのような重大な事故を引起すかについては，リランが簡単にできないという意味で，オンラインのケースについて説明するのがよかろう。

障害復旧手段については，処理の復旧とデータ・ベースの回復との二つに分けて説明する。

処理の復旧手段としては，チェック・ポイント／リスタート機能の基本的な考え方について説明し，かつ，この機能を使用するときの制約条件を述べる。

データ・ベースの回復手段としては，

① データ・ベース・ダンプからの再構成

② データ・ベース・ダンプと変更履歴からの回復

③ 特定アプリケーション・プログラムの影響の除去（バック・アウト）

などがあることを説明し，これらの手段のために用いる

① データ・ベース・ダンプ

② データ・ベース変更履歴
   - Before Image
   - After Image

③ 変更履歴の集約結果

などの役目について述べる。

また障害復旧は，復旧に要する時間，障害の救済度合によって構成上，経済上大きな差があり得ることを説明する。（図２－１参照）

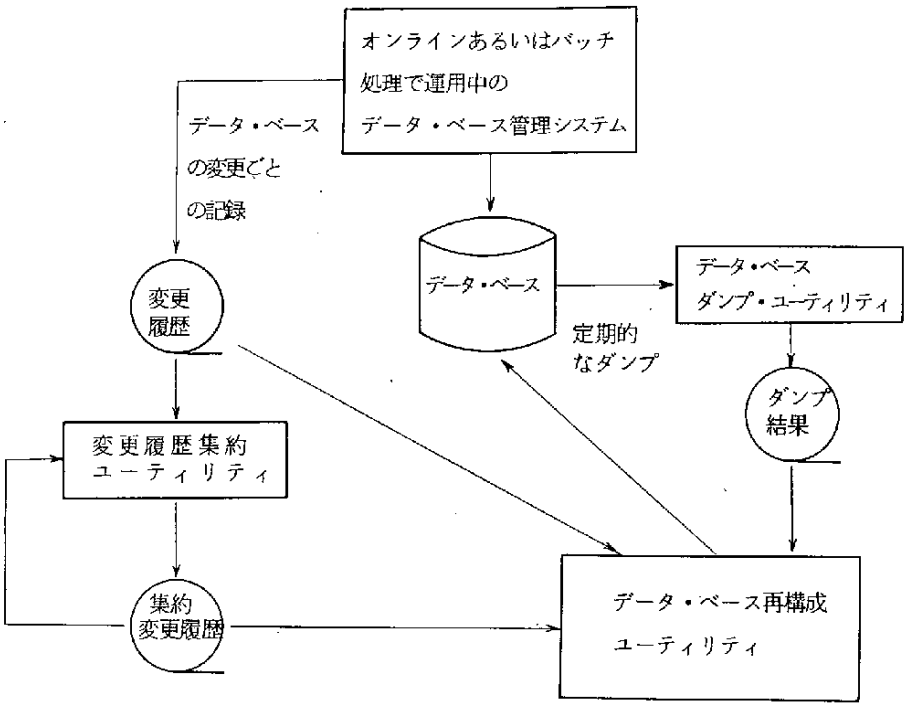


図２－１　データ・ベースの回復手順の例

(4) 演習問題

(a) チェック・ポイント／リスタート機能を用いる場合，制限となる条件とその理由とを研究させる。

(b) 単目的（あるいは固定目的）のデータ・ベースと多目的（あるいは汎用）のデータ・ベースとでは，障害復旧に要する難易度が著しく異るが，何故かを考えさせる。

指導上の留意点

(1) この章は，データ・ベース・システムと称されるものが，どういった便利な機能を持っているかを理解させるのがポイントである。同時に同じデータ・ベースでも利用の仕方によって信頼性などに差があることを強調したい。

(2) これらの機能の追求は，他方において経済的負担の増大或いは効率の減少とも関連することを注意する。

(3) 理想的なあるべき姿と現実の技術水準とのギャップを認識させる。

第3章　データ・ベースの構築技術

用　語　この章では次の用語を教える。

チェンジ・ジャーナル，正当性チェック，サム・チェック，チェック・ディジット，一致チェック，合理性チェック，直接アドレス，辞書索引，計算法（ランダマイジング），2乗採中法，除算法，基数変換法，リスト構造，逆リスト構造，リング構造，木構造，バイナリ・トリー，バランスト・トリー，多重リスト構造，一般ファイル構造，ディレクトリ・デコーディング，固定長打切りキーワード・トリー，一意打切り可変長キーワード・トリー，可変長完全キーワード・トリー，排他制御，LOCK，UNLOCK，デッド・ロック，インタ・ロック，KEEP，FREE，チェック・ポイント／リスタート

目　標

データ独立性などといって，理想的にはユーザが物理的データをなるべく意識しないようにしようというのが現在のデータ・ベース技術の目標ではあるが，実際には，どこかの段階で物理データが操作され，しかもそれが効率に大きく影響して来るのが現状である。

本章ではこれらデータ・ベース・システムの背後にある物理データに密着したデータ処理技術を理解させ，実際の応用上生じた問題に実戦的な解決策を自から導き出す能力をつけさせるのを目的とする。

内　容

3.1　順編成ファイル

(1) 分類（SORT）とマッチング

順編成ファイルを用いての事務処理は，ほとんど例外なく分類（SORT）とマッチングの技法でなされることを説明する。[24]

(2) ファイル・メンテナンス・テクニック

順編成ファイルでよく用いられる ファイルの更新あるいは検索の諸技法を次の項目に沿って説明する。[24],[25]

① **チェンジ・ジャーナル**

② 順編成ファイル内のレコードの移動方法

③ 分類キーの設定方法

必ずしも順編成ファイルでなくても利用できるが，ファイルの完全性を保つために次のような正当性チェックの技法を説明する。

④ **サム・チェック**

⑤ **チェック・ディジット**

⑥ **一致チェック**

⑦ **合理性チェック**

- リミット・チェック
- タイプ・チェック
- 存在〔不存在〕チェック
- 相関チェック

(3) 演習問題

(a) ファイルに対して複数個のキーで検索ないし更新を行なう必要のある例を数種与え，これを順編成ファイルで扱ったとき分類とキーあるいはインデックスとが対応することを考える。

(b) 内容の説明の項 (2)の④から⑦で述べたような正当性のチェックがどのようなときに有効かを考えさせる。チェックの行なわれるタイミング，業務対象について考えさせる。

3.2 ランダム・ファイル

(1) 直接アドレス

直接アドレスとはプログラムが蓄積レコードのアドレスを直接指定する方法である。システム内部ではこの方法が最終的に用いられていることを説明する。

(2) 辞書索引

辞書索引とはレコードを索引するキーとレコードのアドレスとを対にして記憶することにより，キーの内容によるレコードの索引を行なう方法。いわゆるインデックス，あるいはキー・ディレクトリ等の基本となる方法であることを説明する。

(3) 計　算　法

計算法(ランダマイジング)，あるいはハッシング等ともいわれている。

この方法の基本的なロジックを説明し，そのロジックから導かれる次のような性質について説明する。(参考文献 (6) 参照)

① データのスペースと検索あるいは蓄積スピードとがトレード・オフになること。

② 頻繁な更新を行なった後にはアクセスのスピードが落ちる傾向があること。

③ 各レコードに対応して，制御情報を持つ必要があること。

また，キーの値から物理アドレスへの変換方法として代表的な次のような方法について説明する。

④ **2乗採中法**

⑤ **除算法**

⑥ **基数変換法**

(4) リスト構造

データ・ベース構築に用いられる**リスト構造**のうち，次に述べるようなリスト構造について説明

する。[26],[27]

① 単純リスト構造

② **逆リスト構造**

この構造は，頻繁に用いられるものであるから，次の各項目について更に詳述する。

・逆リスト構造を持つファイルの検索方法

・逆リスト構造のファイルの作成方法

・同じく更新方法

③ **リング構造**

この構造を用いれば，原理的にはネット・ワーク構造も表現できることを説明する。この構造の応用範囲としてはBMプロセサの基本データ構造があることを説明する。また歴史的にこの構造が重要な役割を果しているデータ・ベース・システム（例えばＩＤＳなど）の実例などを説明する。

④ 木 構 造

木構造の定義を与え，**バイナリ・トリー**，**バランスト・トリー**などの定義を与える。

⑤ **多重リスト構造**

⑥ **一般ファイル構造**[28]

応用というよりは，むしろ索引順編成，逆リスト構造，多重リスト構造などを総括的に包含する構造として説明する。（図３－１参照）

(5) ディレクトリ・デコーディング

ディレクトリ・デコーディングとは索引付ファイルを構築するときよく用いられるキーからレコードのアドレスを知る（いわゆるデコーディング）方法で，ディレクトリを用いる方法を以下の項目に沿って説明する。

・固定長打切りキーワード・トリー

・一意打切りキーワード・トリー

・可変長完全キーワード・トリー

ディレクトリの構造を説明するとともに，そのデコーディング・アルゴリズムを説明する。

また，これらの方法の長所短所，実際上の応用例を説明する。

(6) 演 習 問 題

(a) ランダマイジングを行なったときの制御情報のおのおのについて，その必要性あるいは，それがない場合の不都合について考えさせる。

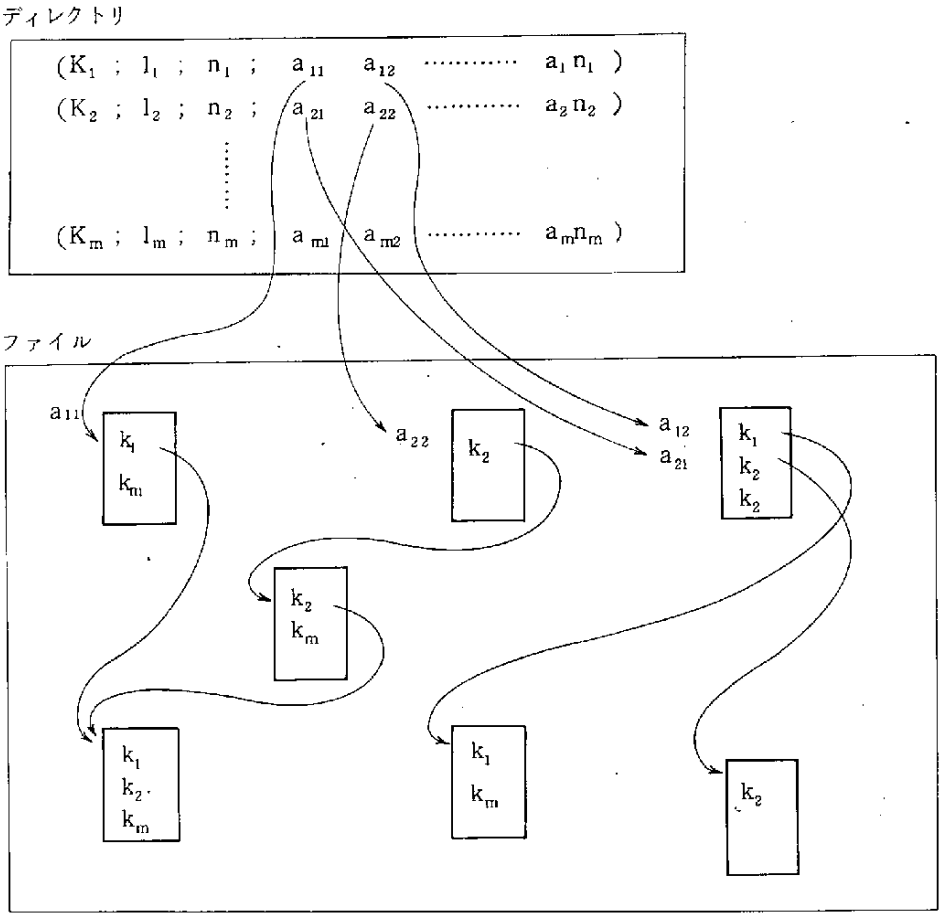
(b) ランダマイジングを行なう場合の種々のファイル構成あるいはデータ構成について，研究させる。

(c) 逆リスト構造で，データを更新するとき，キー部分とそうでない部分の更新には著しい差があるがその理由を考えさせる。

(d) リング構造では，かなり一般的なデータ構造が表現できる。たとえば次のような応用分野ではこの構造をいかに用いるべきかを考えさせる。

① 簿記における仕訳

② BMプロセサの部品展開



$k_1, k_2, \cdots\cdots, k_m$ はキーを表わす。

$l_1, l_2, \cdots\cdots, l_m$ は各キーに対応したレコードのリストの長さを示す

$n_1, n_2, \cdots\cdots, n_m$ はディレクトリから直接ポイントされるレコードの個数を示す

a_{ij} は　レコードの先頭アドレスを示す。

図3－1　一般ファイル構造（generalized File Structure）[28]

③ サブルーチンのcall関係の表現

④ 木構造の表現

⑤ 人とその専攻分野の対応ファイル

(e) 逆リスト・ファイルの例にならって，多重リスト構造を持つファイルの

① 検　索

② 作　成

③ 更　新

をどのように行なうべきかを考えさせる。

(f) ディレクトリ・デコーディングのデコーディング・アルゴリズムの正当性をおのおのの方式について確かめさせる。[26],[27]

この問題は，かなり程度が高いので最初は必ずしもやらなくてよい。

(g) ディレクトリの構成時にキーワードの短縮ができないかどうか考えさせる。

(h) 図3－1の一般ファイル構造の特殊ケースとして索引順編成ファイル，逆インデックス構成，多重リスト構造を考えさせる。

3.3 オペレーティング・システムとの関係

データ・ベース・システムのどのような機能がオペレーティング・システムの機能とどのように関わり合うかを示すのがこの節の目的である。

ユーザ側から見れば同じような機能でもどれだけがデータ・ベース側でサポートされ，どれだけがオペレーティング・システム側の責任なのかという判断ができるようになれることが望ましい。

以上では，主にデータ・ベース側では根本的な解決は難かしくて，オペレーティング・システム側でサポートすべき項目について説明する。

(1) 排 他 制 御

複数個のユーザが独立に共通のデータ・ベースに対して種々のアクセスをするときに**排他制御**が是非とも必要であることを説明する。

排他制御がオペレーティング・システム側でもれなく行なわれていないと，データ・ベースの完全性が保てなくなることを説明する。

また，気安くLOCK，UNLOCKなどの制御命令を認めると**デッド・ロック**の危険性があることを説明し，かつデッド・ロックになったときのユーザ側への影響を説明する。

デッド・ロックにならぬため**インタ・ロック**をかける方法や，ユーザ・プログラム間で使用規約を作成して実用的にこの問題を回避するKEEP，FREE（CODASYL DBTGの提案）等の解決案について説明する。

(2) 共通インターフェイス

データの検索，蓄積に関してオペレーティング・システム側が一本化したインターフェイスを持ち，それ以外のアクセスを禁止する機能が次の二つの点から必要であることを説明する。

① データへのアクセスが何種類かに標準化されることにより，データ・ベースの更新の履歴がとり易くなり，エラー回復等に便利になる。

② 共通インタフェース以外のデータ・アクセスが禁止されることにより，データ・ベース・システム外からのアクセスによるデータの乱れがなくなり，完全性を保つことができる。

(3) チェックポイント／リスタート

オペレーティング・システムの提供する**チェック・ポイント／リスタート**の機能について説明し，データ・ベースの障害回復のためにどのように用いればよいかを説明する。

また汎用目的のチェック・ポイント／リスタート機能でカバーし切れないケースについて説明し，どこに技術的困難さがあるかを指摘する。

(4) 演 習 問 題

(a) デッド・ロック検出のロジックあるいは，デッド・ロックを起こさせないためのインタ•ロック・ロジックを調べさせる。（デッド・ロック問題は，(a) 起こさせぬ方法と(b) 起こすことを許すが，起きたときにうまく処理する方法と(c) 野放しと3種の対応法がある。）

(b) 排他制御にはどれだけの種類のもの（たとえば，書込み禁止とか，排他制御の対象とか）が必要かを考えさせる。

指導上の留意点

(1) データ・ベースの構築技術は，非常に広汎な分野に分散する。そのため，説明が浅くならぬように，通常のデータ管理で述べられる事柄については，できるだけ説明を省き，データ・ベース固有の技術を中心に説明をしぼること。

(2) 第3章の内容は，かなりソフトウェア技術上，細部にわたるものである。単に得られている結論のみを紹介するというのではなく，常に何故そういう結論が得られているのかという疑問を研修生に起こさせ，かつ，それに対する必然的な解答として，結論を説明するようにすること。また，説明は時によっては完全な形で与えず一部を演習問題として研修生自身に考えさせることの方が，教育上の効果が上がる。

(3) データ・ベース構築技術の評価，及び最新の技術については参考文献 (5) ， (6) とその中で更に参照されている資料が参考となろう。

第4章　データ・ベース管理者機能

用　語　この章では次の用語を教える。

データ・ベース管理者，ファイル・バック・アップ，ガーベジ・コレクション，システム編集，ファイル配置，

目　標

データ・ベースは，データを集中管理するため，その維持には従来の個々のアプリケーションごとのファイル・システムの時とは格段の管理が必要となる。

データ・ベース管理者は，そのための高度のデータ処理技術を有した個人ないしは集団で，データ・ベースの運用には欠くことのできない存在とされている。

本章では，このデータ・ベース管理者の必要性を理解させ，彼の果すべき役割について説明する。

内　容

4.1　設計，開発機能

データ・ベース管理者が果すべきこととして次のことがあることを説明する。

① データ・ベースの定義

② 論理データと物理データとの定義，及び両者間のマッピングの定義

③ 物理ファイルのアクセス法の決定

④ エラー回復システムの設計

⑤ 技術上の標準化規準の設定

⑥ 機密保護維持

⑦ 新しい機能の開発，各種データ・ベース手続きの開発

4.2　データ・ベースの管理

これには次の機能がある。

① ユーザとデータ・ベースの間の仲介

② 特権機能の許可

③ データ・ディクショナリあるいはディレクトリの維持

④ データ・ベース・アクセスのための複合コマンドの用意

⑤ 新規あるいは変更されたアプリケーション・プログラムのチェック

4.3 オペレーション

これには次の機能がある。

① 障害復旧オペレーション，**ファイル・バック・アップ**

② データ・ベース生成

③ 前準備

④ データ・ベース再編成

⑤ **ガーベジ・コレクション**

⑥ データ・ベース処理のスケジューリング

4.4 モニタリング

この機能に属すものは以下のようなものである。

① パフォーマンスの監視

② 統計収集

③ 情報消失の危険の検出

④ 機密保護

⑤ データ・ベースの品質保証

⑥ 冗長情報の検出

⑦ 完全性の保証

4.5 システム改良

これには次がある。

① チューン・アップ

② 収集統計データの解析

③ **システム編集**

④ **ファイル配置**

4.6 演習問題

データ・ベースの完全性，信頼性，機密保護のそれぞれを考えるためには，データ・ベース管理者がどの機能をどのように適用すべきであるかを考えさせる。

指導上の留意点

(1) データ・ベースの管理は，従来のデータ処理形態に比して，データ集中度が大であるだけに同じ程度の障害があっても及ぼす影響が大であることに注意させる。

(2) データ・ベース管理者には，データ独立性その他一般ユーザに与えられる便利な機能はあまり与えられなく，物理データと密着した分野での活動が主であることを悟らせる。

第5章　ユーザ・インターフェイス

用　語　この章では次の用語を教える。

ホスト言語，自立型言語，スキーマ，サブスキーマ，セット

目　標

本章では，一般のユーザがデータ・ベースを利用してデータ処理業務を行なう際の諸問題について理解させる。

特に従来のデータ処理と比べて，データ・ベースによる処理がどのような関係になるかを理解させる。

まずデータ・ベースの利用方法としてユーザの要求によって何種類かのレベル分けが可能であることを示し，次に，データ・ベース・システムの方法の二つの大きな考え方について説明する。

さらに，データ・ベースをその処理手順とは別に独立して，別個に定義することの必要性，重要性について理解させ，さらに特別のアプリケーションのためのデータ定義を，CODASYLのデータ・ベース機能を例にとって実際に書けるようにする。

また，データ・ベースにアクセスするときの基本的な操作機能として何があるかを述べるとともに，データ・ベース・システムに通常備わっている障害復旧機能の使用法について理解させる。

内　容

5.1　ユーザの種類

データ・ベースのユーザは，必要とするデータ処理のレベルに応じて次の4段階に分けられることを説明する。[2]

① パラメトリック・ユーザ

② ノンプログラマ

③ アプリケーション・プログラマ

④ データ・ベース管理者

以上のうち④についてはすでに第4章で説明を完了しているので④以外のおのおののユーザがデータ・ベースを利用する場合にどのようなことをしなくてはならぬかを述べる。この説明によって，データ・ベースを取入れる段階にも色々な段階があり，業務ごとあるいはアプリケーションごとに適切な段階を利用することが肝要なことを説明する。

5.2　ホスト言語と自立型言語

データ・ベースのユーザ・インターフェイスには，使用する言語の面から分類して**ホスト言語**（Host

Language）方式と**自立型言語**（Self－Contained Language）方式の2種類が存在する。

(1) ホスト言語方式

これは，データ・ベースとの直接のデータの授受は，データ・ベース機能が行ない，その他ホスト言語にまかすのが適当である処理はホスト言語が分担するやり方である。

この方式による場合ユーザは，データ・ベース相手の処理とそれ以外の処理とを一つのプログラムの中に混ぜて書くことになる。この混ぜ合せは原則としてユーザの自由になるので，この方式は処理の自由度が大きいことを説明する。その反面，データ処理のやり方を細いレベルで指定しなければならないのが普通である。

(2) 自立型言語方式

これに対して自立型言語方式は，データ・ベースを用いてなされる処理の類型パターンを簡単な指定によって使用させようとするものであり，データ処理の表現には自由度が乏しいが，システムが用意するパターンにあてはまる業務は簡潔に記述できる利点があることを説明する。

(3) 演習問題

(a) ホスト言語型方式に分類される次のシステムを調べ，そのホスト言語とのつながり方，システムが用意している基本的な機能を列挙する。

① IMS (Information Management System)

② IDS (Integrated Data Store)

③ CODASYLのデータ・ベース機能

④ その他国産機種のホスト言語型データ・ベース・システム

(b) 自立型言語方式に分類されている次のシステムについて(1)と同様の研究をさせる。

① MARK Ⅳ

② GIS

③ その他国産機種の自立型言語方式データ・ベース・システム

5.3 データ・ベースの定義

(1) 定義の現状

データ・ベースの定義は，データ独立性などを達成するためにデータ・ベース・システムにおいて最も特徴的，かつ重要な位置を占めるが，まだその考え方，定義に用いる言語等に決定版がある訳ではなく，別個のシステムあるいは提案がバラバラに存在しているのが現状である。ここでは，このような現状を説明する。

(2) 定義の説明

各種システムに共通に見られる普遍的な考え方，重要な思想，具体例における例題等を考慮して次のようなテーマに従って説明するのがよかろう。

① データ・ベースの定義には，

・データ・ベース全体を矛盾なく統合した形で定義するいわゆる**スキーマ**と

・そのスキーマの下に，各アプリケーションあるいは，ユーザごとにその処理目的に応じた定義を

するいわゆる**サブスキーマ**

の2種類が存在すること。

② 上記2種類の定義が必要な理由

③ データ・ベースの定義が，ユーザ・プログラムと独立に存在することによって，保たれるデータ独立性の例

④ ③と逆に，たとえデータ定義が別であっても，データ独立性が保てない例

なお，③，④を考えるに当っては，新しいフィールドをレコードに設けるケース，フィールドをレコードから削除するケース，新しいタイプのレコードを導入するケース，あるレコード内のフィールドを別のタイプのレコードに移すケースなど，いろいろのデータ・ベース内に蓄積されるデータの変化を考えるのがよかろう。

⑤ Bachman Notation。 これは，IDS Short Hand とほぼ同内容のもので，CODASYL データ・ベース機能の中の**セット**（SET）を表現する図式と類似のもの。[8]

図5－1，5－2は Bachman Diagramの例である。

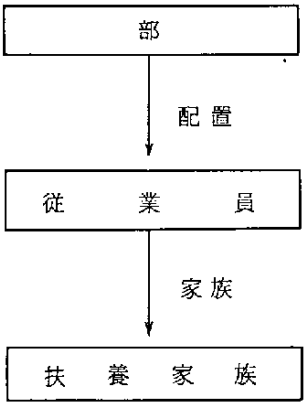


図5－1 Bachman Diagram による 階層構造の表現例

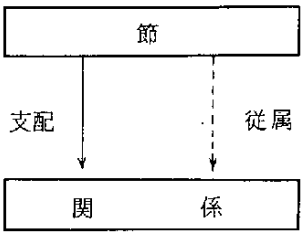


図5－2 Bachman Diagram による ネット・ワークの表現例

⑥ 上記 Bachman Diagram を用いての各種の関係の表現，例えば，

・部品展開での部品マスタと構造レコード

・趣味と専門分野の記録されている人事データ・ベースの定義

・簿記の仕訳と元帳記入の関係

⑦ 論理的な定義とは一応関係ないが，検索あるいは蓄積のスピード・アップのためデータ定義で考慮すべき事柄

・レコードの配置

・レコードのアクセス・パス（Access Path：どのような順番でレコードにアクセスするか）

・ランダマイジングあるいはインデックシング（Indexing）の補助アクセス法

(3) 演習問題

(a) 次の具体的なシステムにおけるデータ定義方法について調べさせる。

・CODASYLデータ・ベース機能

・IMS

(b) (a) で調べた二つのシステムのデータ定義機能を用いて，5.3 (2) ⑥ の例などの表現を考えさせる。

(c) 適当なデータ・ベース・システムのデータ・ベース定義機能について(a)，(b) と同様の研究を行なわせる。

5.4 基本データ操作機能

自立型言語方式を採用しているデータ・ベース・システムでは，基本的なデータ操作機能は表面に現われず，ユーザにはこれらの機能が色々に組合せられたもっとマクロな機能が提供されている。

データ・ベース・システムの能力を正確に理解させるには，これらの基本データ操作機能がユーザ・インターフェイスに現われているホスト言語方式のコマンドあるいは動詞を説明するのがよい。

ホスト言語方式のデータ・ベース・システムの例としては，CODASYLのデータ・ベースとIMSあたりが適当であろう。（表5－1，5－2 参照）

表5－1　CODASYLデータ・ベース Facility における基本動詞

ACCEPT	Currency Indicator あるいは Realm 名を知る
CONNECT	セットの中にメンバ・レコードを挿入する
DISCONNECT	セットの中からメンバ・レコードを取去る
ERASE	データ・ベースの中からレコードを除去する
FIND	ある種の検索条件に従うレコードをカレント・レコードとする
FINISH	Realm の利用を終る
FREE	拡張モニタ・モードにあるレコードを解放する
GET	カレント・レコードをユーザ領域に移す
KEEP	カレント・レコードを拡張モニタ・モードに置く
MODIFY	カレント・レコードの内容を修正する
ORDER	セット中のメンバ・レコードをある順序に従って並べる
READY	Realm を利用可能状態にする
REMONITOR	モニタ・モードあるいは拡張モニタ・モードを一旦打切った後に継続する
STORE	ある種の格納条件に従ってレコードをデータ・ベースに格納する
USE	COBOLのUSE命令とほぼ同じデータ・ベース用の使い方が可能

表5－2　IMSにおける DL／I 呼出し入出力機能

GET UNIQUE	一意に定まるデータ・ベース中のセグメントを検索する
GET NEXT	指定された順番に従って次のセグメントを検索する。
GET NEXT WITHIN PARENT	指定された親セグメントの下に従属するセグメントを検索する
GET HOLD UNIQUE / NEXT / WITHIN PARENT	意味は各々上に説明した三つのGETと同じであるがHOLD 指定があると，その後のコマンドでセグメントの内容の変更あるいは削除をすることを意味する
INSERT	データ・ベースの初期時生成，あるいは新しいセグメントの具体値追加に用いる
DELETE	データ・ベースから指定されたセグメントを削除する
REPLACE	GET HOLD 関係の命令で検索されたセグメントの内容を修正後，置き換える。

＜演習問題＞

(a) CODASYLのデータ・ベース・ファシリティで提案されている動詞を使って簡単なデータ処理プログラムを書いてもらう。

(b) IMSについて上と同じ演習を行なう。

5.5 障害対策

システム側が一般的に提供する障害対策手段については2.6(3) で説明がなされているが，ここでは更に，ユーザ側として考えねばならぬ障害対策について説明する。

はじめに，データ・ベースの安全度あるいは信頼性というものが，ユーザ・プログラムの性質あるいはデータ・ベースの運用環境によって著しく影響を受けるものであるということを説明する。

これを納得させるには，単純にリスタートができないファイルの更新方法であるとか，あるいは，チェック・ポイント／リスタート機能で一般的に設けている制限条件に違反するような操作などを例にあげるとよい。

データ・ベースの障害対策のチェック項目としては次のようなものが考えられる。

① アプリケーション・プログラムにエラーがあった場合のファイルの破壊

② 入力データの正当性

③ ② のチェックを行なったにもかかわらず不当データが入力されたときの処置

④ デッド・ロックを避けるための方策

- 適当な時間待った後の再試行
- 必要資源のロックの順番のとり決め
- 必要資源に余裕を持たせる

⑤ 記憶装置が Read あるいは Write 不能になったときの処置

＜演習問題＞

(a) 同種類のデータの更新でありながら，いくつかの異った方法を何種類かあげさせ，いずれが障害対策上有利であるかを評価させる。

たとえば，あるソース・プログラムの一行を取り去るとき，それをコメント行で置き換える場合と完全に取去った後に行番号を整え直すような更新をする場合の比較などを考えさせる。

(b) 費用の面から次のオプションを比較，評価させる。

- オンライン・データ・ベースで更新を行なう場合と行なわない場合
- 更新を伴なうオンライン・データ・ベースで障害回復を早期に行なわねばならぬ場合と適当に遅延させてもよい場合
- 障害が生じた場合，代替手段が用意されている場合と用意されていない場合

指導上の留意点

(1) データ・ベースの定義方法，基本データ操作機能共に，種々の方法があって一意には定まらない。例にあげたシステムも資料の入手の容易さなどから選ばれたものであるから，指導者は，身近に存在するシステムで例示を行なってよい。

(2) 5.4では，第6章でのデータ・ベースの標準化案の説明に先立って，CODASYLの案を例として用いているので，第6章との説明とよく調和する形で説明してほしい。

第6章　データ・ベースの標準化

目　標

この章の目標は二つある。一つは，標準化の必要性を理解させることであり，もう一つは，現段階での標準化の動きを理解してもらうことである。後者の動きとしては現状ではCODASYLを中心とするものが唯一のものであるので，実際にはその下部委員会での活動を理解させるのが目標である。

内　容

はじめにデータ・ベース処理というものが歴史的には個々の企業ないしは組織で別々に構築され利用されて来たものであるが，最近では，多くの組織体で類似の処理を行なう必要性が見られること，またこの必要性は将来も継続的に存在するであろうことを説明する。

このような事情にあるデータ処理手続きは時間，場所，関連ハードウェア並びにソフトウェアを通じて互換性が必要となることをプログラミング言語の例に照らして説明する。

また，現状ではデータ・ベースについては色々のテクニック，仕様，機能等が個々のシステムの言葉で書かれ，もしくは表現されており，これらの概念整理，技術交流などのためにも標準化が必要であることを強調する。

標準化の具体的な動きについては，参考文献（9），(10)，(11) を参考にするのが適当である。

標準化案の技術的内容については次の順序で説明するのがよかろう。

まず，この案は，データ・ベースのユーザ・インターフェイスとして思想的にホスト言語方式を採用していること，また，データ・ベース・システムとオペレーティング・システム，ユーザとの全体的関連について説明する。

次にデータ・ベースの基本的な概念ないしは機能としてのデータ記述ないしは定義について，スキーマとサブスキーマとを置く思想について説明し，これが,データ・ベースのデータ独立性と異種プログラミング言語とのインターフェイス上果す重要な役割について詳述する。

またこの標準化案で用意している論理的なデータ構造の一つであるセット(SET)についても触れる。

最後に，標準化案のデータ操作機能についてであるが，これについては5.4ですでに例としてとり上げられているから，個々の説明は省略してよかろう。全体との関連で5.4で詳しく説明できなかったものについて説明を補う。

また，これらの標準化の方向がデータ・ベースの進むべきただ一つのものではないことを示すために，参考文献 (12)，(13) でなされているような反論の紹介，参考文献 (14)，(15)，(16) でなされている理論面の最近の動向，ユーザの持っている広汎な機能要求 (17)，(18) についても触れることができれば結構である。

＜演習問題＞

(a) CODASYLの標準化案で，プログラムを書いた場合，どの程度のデータ独立性が保たれるかを吟味する。（参考文献（13）参照）

(b) 同案に盛られているレルム（Realm），セット（Set），レコード，セット・セレクション（Set Selection），モニタ・モード，拡張モニタ・モード，カレンシー・ステータス等の概念を調査させ報告させる。

これらの概念がデータ・ベース処理にとって本質的なものかどうかのディスカッションを行なう。

(c) 同案に対する批判意見（参考文献（12），（13））等を調査し，ディスカッションを行なう。

(d) データ・ベースに対する（現状では過大と思われる）要求（参考文献（14）～（18））から考えると上記標準案は満足できる水準にあるかどうか。

指導上の留意点

(1) CODASYLによる標準化案は，まだデータ・ベース・システムの標準となるには相当の煮詰め，改良が必要と思われる。従って，〝内容〟の最後のパラグラフ，あるいは演習問題で述べたような取扱いをする必要があろう。

(2) それにも拘らず，CODASYL案は現在では標準化という目標には一番近い存在であるし，各国からの改善要求が比較的受入れられやすい柔軟な存在なので将来の発展が期待される。

(3) 色々の困難はあっても標準化は，データ・ベースの発展のためには絶対必要なものであるという教師自身の自覚が必要である。

第7章　具体例

目　標

企業等で導入されたデータ・ベース・システムについて，その導入，設計，利用，評価等，一連の具体例について調査検討する。

内　容

内外の文献で公表されている各種データ・ベース・システムについての報告を基に，データ・ベース・システムの個々の実例を調べさせる。[19],[20],[21]

応用例を一つ取上げ，それを色々な汎用パッケージで記述することを試みさせ，それぞれの利害得失などについて論じさせるのが有益である。

指導上の留意点

(1) この章の学習は，殆どが，演習形式でなされる。研修生は与えられたテーマごとに実際例を調査し，発表する。

(2) 教師は，研究対象となった具体的なデータ・ベース・システムについての大体の知識を研修生に与えておく必要がある。

第8章　データ・ベースと情報検索

用　語　この章では次の用語を教える。

検索，事実検索，文献検索，主題分析，索引，SDI，キーワード，UDC，シソーラス，KWIC，KWOC，自動索引，ストップ・ワード，論理式，インバーテッド・ファイル

目　標

この章の目標とすることは，情報検索システムを構築するためにデータ・ベースがどの程度役立つか，あるいはどのように利用できるかを理解してもらうことである。

まず，情報検索は何かといったことについて概略的に理解させる。情報検索*という言葉は，本来，情報の蓄積*と検索という言葉から出たものであることからも知られる通り，よい情報検索はその背後にしっかりとした情報の蓄積がなされなくてはならない。そこで，このような機能について理解させる。

次に，このようにして蓄積された情報を検索する手法のうち，特にデータ・ベース的なアプローチを取る計算機による検索方法について説明するとともに実際の情報検索システムの例を通して，理解を深める。

内　容

8.1　概　要

この節では，一応データ・ベースとは離れて情報検索一般の概念を説明する。[31] 情報検索が誕生した背景にはその当時（1960年代）扱う情報が飛躍的に増加したという事実があることを説明し，本来情報検索は，大量の情報を収集，整理，利用するという意味で，自動化ないしは機械化と密接な関係にあることを説明する。

この機械化にはコンピュータ以前に，PCS（パンチ・カード・システム），マイクロ・フィルム装置，諸カード・システムがあったが，現在の情報検索にはコンピュータが大きな役割を果していることを説明する。コンピュータ以外の情報処理機器としては，マイクロ・フィルム関係でロール・フィルム，フィルム・ストリップ，マイクロ・フィッシュ，マイクロ・カード，アパチュア・カードなどをIR（情報検索）の常識として説明する。

カード・ベースのものとしては，ハンド・ソート・パンチ・カードなどのへり穴式のものとかピーカ

---

* ここで用いている検索，蓄積という用語は，情報検索特有の意味を持ち，第2章で述べたものよりは，広い内容を持っている。

ブー（あるいはバッテン）式のものに言及する。

ＰＣＳとしては，パンチ，ベリファイア，ソーター，照合機，会計機等があるが，これらについての説明は，簡単でよい。

情報検索の本来の機能の説明に戻って，次は，**事実検索**と**文献検索**という二つの検索に対する考え方を説明する。

情報検索には，データの収集，評価に始まって，分析，索引づけ，蓄積，質問，探索，情報の配布などのフェーズが存在するが，これらの手順に沿って，そのフェーズでの中心的な概念について説明していく。

評価とは，収集したデータが信頼できるものか，あるいは情報検索を行なう使用者にとって必要なものかどうかを評価することである。

分析とは，得られたデータの**主題分析**であり，その結果に基づき，後にさがすときの手がかりとなる**索引**がつくられる。

索引づけとは，蓄積されたぼう大な情報がだれにも利用されないで死んでしまわないように簡単な，索引語によって蓄積データが参照できるようにすることである。

索引づけは，情報の蓄積の主要なテーマである。蓄積情報に対しては種々の質問がなされるが，あらかじめ索引づけられたデータに対して機械的な検索ができるほかは，現状では自然言語の解釈もしくは認識には重大な困難があることを説明する。現状でデータ・ベースに対する質問の中で最も強力なものとして，参考文献(16)あたりで提唱されている検索質問式を説明する。

情報の配布については**SDI**（Selective Dissemination of Information：選定情報提供）を説明する。

8.2 情報の蓄積

情報検索をするには，8.1でも述べたがあらかじめ情報を拾うことができるように情報に索引づけすることが肝要となる。索引語あるいは**キーワード**といわれるものがこれに用いられる。情報検索システムを作って行く上で，索引語の取扱いが最も注意を要することを強調する。

キーワード索引，**UDC**（Universal Decimal Classification：国際10進分類）等の概念を説明する。索引語の扱いに関しては辞書の果す役割を強調し，更に最新脚光を浴びている**シソーラス**について説明する。

索引の作成方法としては**KWIC**（Keyword-in-Context），**KWOC**（Keyword-Out of-Context）などの方法を説明し，**自動索引**の手法について説明する。**ストップ・ワード**（Stop Words：除外語）についてもここで説明する。

索引のデータ・ベース的な構築法としてインバーテッド・リスト・ファイルの役目を3.2(4)に引続いて説明する。

<演習問題>

ＫＷＩＣ索引の作り方の概略フローを考えさせる。

8.3 情報の検索

データ・ベースで取扱う情報検索の範囲を見るには,参考文献(16)で定義しているものを説明するのが適当であろう。この言語は，従来の検索言語が，ファイルに蓄積されたレコードのフィールド間の代数的な大，小，等あるいは不等関係を基本として，それ等をAND，OR，NOTで適当に組合せた**論理式**のレベルのものであったのに対し，新たに∀（全称記号），∃（存在記号）を導入したものである。

普通の情報検索システムの検索能力は，∀あるいは∃の導入以前のものであることを説明する。

インバーテッド・ファイル等のデータ・ベースが用意しているアクセス法とは直接に結びつかないが，質問に適合するデータかどうかを種々の測度を設けて判定する方法 ― 重みづけ等による適合度の設定方法について説明する。検索にはエラーがつきものであるが，ここに統計学の検定の場合と同じ種類の2種のエラー，すなわち第1種の過誤と第2種の過誤があることを説明する。

<演習問題>

データ・ベースに対する情報検索でよく使われる論理式の標準形である選言標準形式（disjunctive normal form）の応用例について調査，報告させる。[28],[16]

8.4 具体例

コンピュータを使用したＩＲシステムの具体例をあげ，説明をするとともに，研修生に実例を研究させ発表させる。内外のＩＲシステムの実例としては次のようなものが適当であろう。[34],[34]

① 日本科学技術情報センターの各種システム

② MEDLARS（Medical Literature Analysis and Retrieval System）米国国立医学図書館の機械検索システム

③ RINGDOC 英国ダーウェント社のサービスする主として薬学関係の情報を扱う。

④ CAS（Chemical Abstracts Service）のＩＲサービス。主として化学関係文献の抄録サービスを行なう。

⑤ INTREX（Information Transfer Experiments）ＭＩＴでサービスしているオンライン文献検索システム

⑥ TIP（Technical Information Program）ＭＩＴで開発された物理学雑誌対象のＩＲシステム

指導上の留意点

(1) 情報検索システムの研究の場合には，単なるデータ・ベース的な見地からの問題点よりも，実際にどうやって価値あるデータを忠実に蓄積，保守を行なっていくかということの方がはるかに重要であることを注意する。

(2) ここで学ばせる情報検索の知識はいずれも，本来の情報検索の分野のほんの一分野にしか過ぎぬことを教え，もっと重要で根深い問題が他にあることを指摘する。

参考文献

(1) CODASYL Systems Committee : A Survey of Generalized Data Management Systems, May 1969.
邦訳 小林功武編著 : データ・ベースの管理, 企画センター, July 1970.

(2) CODASYL Systems Committee : Feature Analysis of Generalized Data Base Management Systems, April 1971.

(3) EDP Analyzer "The Cautious Path to a Data Base" June 1973 Vol. 11, No. 6

(4) 電子協 データ・ベース専門委員会 1973年度 活動報告 1974年出版予定

(5) Alfonso F. Cardenas, "Evaluation and Selection of File Organization— A Model and System CACM, Vol. 6 No. 9 540–548

(6) V.Y.Lum, "General Performance Analysis of Key-to-Address Transformation Methods Using an Abstract File Concept" CACM, Vol. 16 No. 10 603–612

(7) C.J. Date, I. J. Hearth, P. Hopewell, A Collection of Papers on Data Independence, IBM United Kingdon Laboratories Limitted Technical Report TR. 12. 094 March 1972

(8) Charles W. Bachman, Data Structure Diagrams, Data Base Vol. 1 No. 2, 1969, A Quarterly Nowsletter of SIGDP

(9) Data Base Task Group Report to the CODASYL Programming Language Committee April 1971

(10) Proposal DBLTG−73001.00, The COBOL Data Base Facility, DBLTG 1973

(11) DDLC Journal of Development, CODASYL Data Description Language Committee, 1973

(12) T.R. Heywood, The IBM Position on the CODASYL DBTG Proposal in BCS October 71 Conference on April 71 Report

(13) The IBM Position Paper Presented to CODASYL. (12)と同じ論文集に再録

(14) E.F. Codd, A Relational Model of Data for Large Shared Data Banks, CACM Vol. 13 No. 6, 1970 pp 377−387

(15) E.F. Codd, Further Normalization of the Data Base Relational Model, in DATA BASE SYSTEMS Courant Computer Science Symposium, 6 1971 Prentice-Hall

(16) E. F. Codd, Relational Completeness of Data Base Sublanguages; (15)と同じ単行本に集録

(17) Joint GUIDE & SHARE Data Base Requiremens Group, Requirements for a Data Base Management System, Nov. 1970 (12)の集録されている論文集にも集録されている。

(18) 電子協国際動向分科会，ソフトウェア技術に関する動向 — データ・ベースの持つべき機能 — 昭47年3月これは(17)の翻訳である。

(19) 古河建純他 データ・ベースの実際(1) 情報処理，Vol，14 No.7

(20) 石川利久 データ・ベースの実際(2) 情報処理，Vol，14 No.8

(21) 坪井邦夫 データ・ベースの実際(3) 情報処理，Vol，14 No.9

(22) 情報処理学会，情報処理ハンドブック，オーム社，1972

(23) 上條史彦 データ・ベース・システム産業図書

(24) Thomas R. Gildersleeve, Design of Sequential File Systems, Wiley Business Data Pro-cessing Library, Wiley

(25) Martin Larry Young, Design and Implementation of a Generalized User-oriented Input-Data Error Detecting Language, A Dissertation Submitted to the Graduate College of Texas A & M University, Aug. 1972 MIC 73−3584

(26) D. レフコビッツ，オンライン ファイル システム 日本生産性本部，昭和45年

(27) George G. Dodd, Elements of Data Management Systems, Computing Surveys, Vol. 1, No.2, 1969

(28) David Hsias 他，A Formal System for Information Retrieval from Files, CACM Vol. 13, No.2, 1970, pp. 67−73.

(29) 穂鷹良介，データ・ベース，情報処理研修センター，講義資料，P62−30，P62−34，P62−35，P62−38

(30) 小林功武，ファイルの編成と方式，情報処理研修センター，講義資料，S50−12

(31) 河野徳吉編，情報検索の知識，日本経済新聞社，昭和46年

(32) 熊谷東愷，データ・ベースの実際(4)，情報処理 Vol. 15 No.1 pp54〜66

(33) 数理科学，特集 情報検索，No.129, March 1974

(34) 日本情報処理開発センター，新しい検索技術と検索システムに関する基礎理論の体系化，43−S−003，昭和44年3月

(35) 日本情報処理開発センター，技術情報の機械検索についての現状，44−R−004，昭和45年3月

(36) 植村俊亮，オンライン文献検索システム — イントレックスを中心に — 情報管理 Vol. 15 No.8 pp. 559−566

データ通信システム

目　次

第1章　データ通信システムの概要 ……316
1.1　データ伝送とデータ通信 ……316
1.2　データ通信システム構成 ……316
1.3　データ通信システムの例 ……317
1.4　即時処理と一括処理 ……319
1.5　データ通信システムにおけるデータの流れ ……320
1.6　データ通信システムの特徴 ……322
第2章　端末装置 ……323
2.1　端末装置の構成 ……323
2.2　データ伝送に必要な機能 ……324
2.3　端末装置の種類 ……330
2.4　端末装置の代表例 ……331
第3章　通信回線 ……332
3.1　通信速度と回線のしくみ ……332
3.2　変復調方式 ……333
3.3　通信回線 ……333
第4章　センタ・システムのハードウェア ……337
4.1　センタ・システム概要 ……337
4.2　センタ・システムのハードウェア ……338
4.3　機能面から見たシステム構成 ……342
第5章　オンライン・システムのソフトウェア ……343
5.1　オンライン・システム用ソフトウェアの特徴 ……343
5.2　オンライン・システム用ソフトウェアの構成 ……343
5.3　管理プログラム ……344
5.4　処理プログラム ……349
5.5　支援プログラム ……349
第6章　オンライン・システムのファイル ……351
6.1　ファイルに対する基本的要求事項 ……351
6.2　ファイルの種類 ……351
6.3　ファイルと媒体の対応 ……352
6.4　ファイル編成とアクセス法 ……353

第７章　信頼性 ……354
7.1　障害の種類 ……354
7.2　情報に対する信頼性 ……355
7.3　システムの信頼性 ……356
7.4　障害処理 ……357

科目「データ通信システム」

教育の目標

データ通信システムに関する基礎的な知識を習得させるために，データ通信システムの概要，種類，構成について簡単に説明する。引き続いてデータ通信システムを構成する端末，回線，中央のコンピュータ・システムの各ハードウェアの構成およびその具体的な動作ならびにその用途等について詳しく説明する。

一方，データ通信システムの運用を目的通りに遂行していくためには当然ながらオンライン用ソフトウェアを理解することが重要なポイントになる。オンライン・プログラムの構成および特に，オンライン・システム特有な機能を中心に，説明し理解させる。

更に，データ通信システムで使用するファイルが具備すべき条件，ファイルの種類，ファイル編成の他システムの信頼性についても解説し，データ通信システムの基本的な事柄について十分理解させることを目標とする。

時間配分

章	履習時間（時間）		
	講義	演習	実習
1. データ通信システムの概要	5		
2. 端末装置	10		
3. 通信回線	7		
4. センタ・システムのハードウェア	8		
5. オンライン・システムのソフトウェア	15		
6. オンライン・システムのファイル	5		
7. 信頼性	10		
合計	60		

第1章　データ通信システムの概要

用　語　この章では次の用語を教える。

インライン・システム，インハウス・システム，データ収集システム，データ分配システム，問合せシステム，データ交換システム

目　標

データ通信システムのニーズは，最近ますます増大する傾向にある。これらのニーズを実現するための，コンピュータ技術および通信技術の進展も著しいものがある。

そこで本章では，データ通信システムの概要について簡単に説明する。まず，データ通信とは何かを定義し，次にわれわれが無意識のうちに日常生活において，データ通信システムとかかわりあいを持つ身近かな事例を通して，データ通信システムの構成，種類，特徴などについて，ハードウェア，ソフトウェアの両面から簡単に触れ，システム全体の概要を認識させることを目標とする。

内　容

1.1　データ伝送とデータ通信

(1) データ伝送

データ伝送とデータ通信の基本的な相違を認識させるとともにＣＣＩＴＴが定義するデータ伝送の定義についても触れる。

データ伝送とは「機械によって処理される情報の伝送，あるいは処理された情報の伝送」である。

(2) データ通信

データ通信の一般的な定義として「情報処理装置（コンピュータ）と電気通信設備を利用して行なう。情報の伝送および処理」であることに触れ，用語としてのデータ伝送との若干のニュアンスの違いを認識させる。

また，データ伝送技術とコンピュータ技術が一体となったデータ処理システムのことをデータ通信システムということにも触れる。

1.2　データ通信システム構成

ここでは，データ通信システムの構成要素の概要紹介を行ない，各コンポーネントの位置づけ，役割を理解させる。

① 端末装置

② 通信回線

• 変復調装置（モデム）

③ センタ・システム

• ハードウェア（特に通信制御装置，中央処理装置）

• ソフトウェア

④ ファイル

• ランダム・アクセス装置

1.3 データ通信システムの例

ここでは，代表的なシステム例を通して，データ通信システム全体の概要を把握させることに努める。こゝでは，預金業務の例によって説明しておく。

次のようなマンガ的なシステムの構成図を示して，われわれの最も身近な例，預金業務がどのようにデータ通信システムとかかわりを持っているか，またわれわれが無意識のうちに恩恵を受けているデータ通信システムとは何かを考えさせる手引きとする。

次の観点からシステムを把えて説明し問題意識を持たせるとよい。

(a) 端末機のしくみ

① 鍵 盤 部

② ジャーナル印字部

③ 通帳引込口

④ 伝票引込口

(b) 回線構成

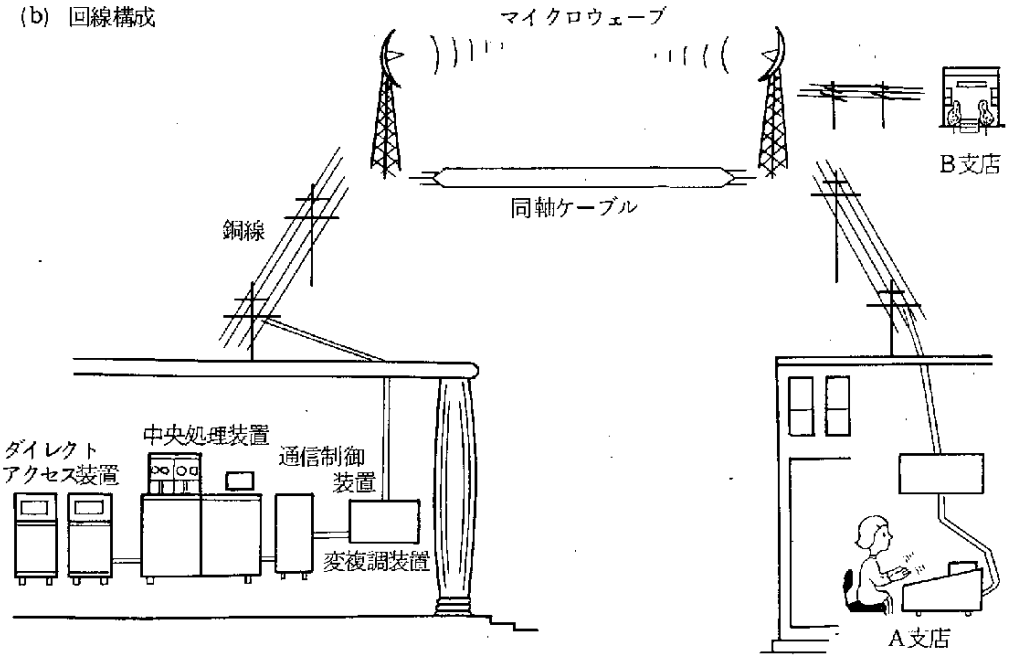


図1－1 預金業務のためのデータ通信システム

(c) メッセージ処理概要

① 端末オペレータとインプット・データ

- 口座番号
- 通帳残高（チェック用）
- 入出金金額

② コンピュータ内での処理

- インプット・データと顧客データが格納されているファイル（口座ファイル）の対応と索引方法
- 必要なチェックと残高更新
- 処理結果の編集と送信

③ 結果のアウトプット

- 預金通帳の印字方法
- 伝送エラーとその対策

④ 信頼性を考慮したシステム構成としてデュプレックス・システムの例を次図をもとに紹介する。

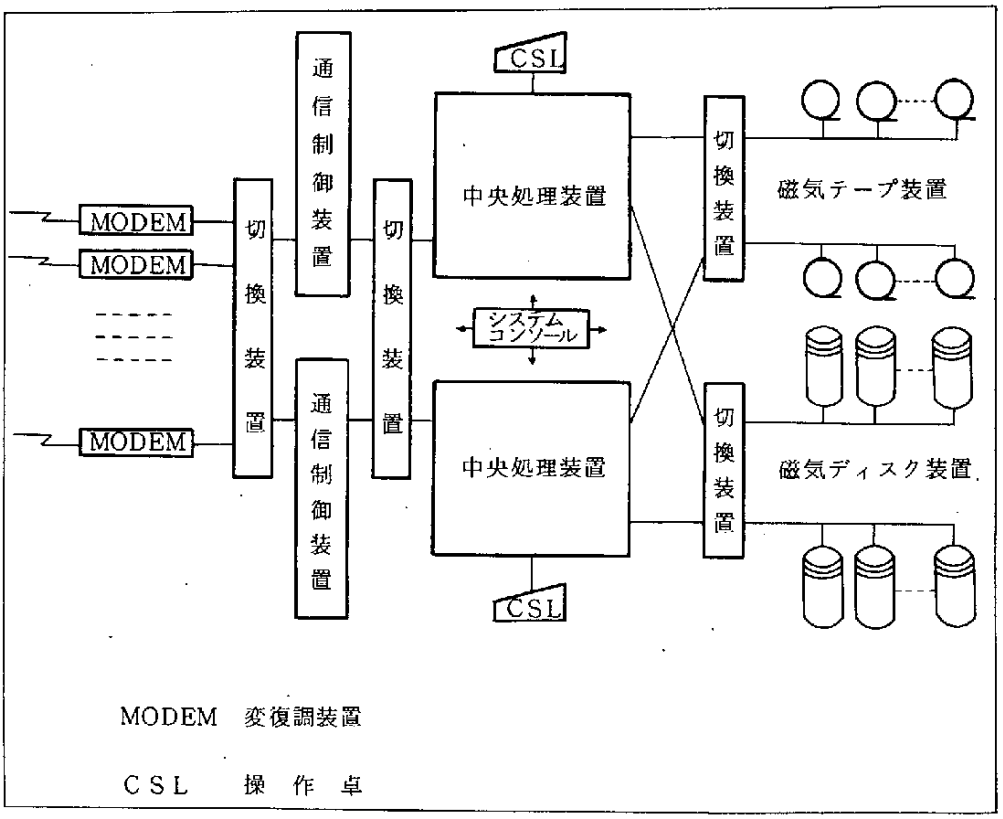


図1－2　センタ・システム

⑤ システムの導入効果

- サービス時間の短縮
- 支店での業務の軽減
- 全店支払いが容易にできる。
- 経営に必要な情報が即座にとれる。

1.4 即時処理と一括処理

ここでは，各用語の意味を理解させることに努める。

(1) オフラインとオンライン

(a) オフライン

これは，遠隔地からのデータをセンタで，一旦紙テープとかカードなどの中間媒体に変換し，オペレータなどの人手を介してコンピュータに入力する形態で，処理方式から見れば従来のバッチ処理と同じである。

(b) オンライン

遠隔地の端末装置は，通信回線でコンピュータ・システムと直結されているので，入力データは発生場所から直接コンピュータにインプットされ，データ処理が行なわれる。データ通信システムにおいてはコンピュータの使用目的から考えてオンライン方式が基本となることを認識させる。

(2) リアルタイム処理とディレードタイム処理

(a) リアルタイム処理

端末からインプットされたデータは，直ちにコンピュータで処理され，その結果を「必要な時間内」に必要な場所に直接アウトプットする処理方式であり，オンライン方式であることが前提となることを認識させる。また，即時処理を行なうシステムには**インライン・システム**とか，**インハウス・システム**と呼ぶ形態があることにも触れるとよい。

(b) ディレード・タイム処理

ディレード・タイム処理はデータ収集をオフラインまたはオンライン方式で行ない，記録媒体にある程度まとまってから一括処理する方法であることを説明する。

(3) タイムシェアリング・システムとリモート・バッチ・システム

(a) タイムシェアリング・システム

コンピュータ・システムを多数の人（企業）が同時に共同で使用するものである。このことによりコンピュータ使用の経済性およびリアルタイム処理方式の利用が可能になることを認識させる。

タイムシェアリング・システムの代表的な例としては，電々公社がサービスしている共同利用システムの各サービス例を紹介するとよい。

- 電話計算サービス（DIALS）
- 技術計算サービス（DEMOS）
- 販売在庫管理サービス（DRESS）

(b) リモート・バッチ・システム

タイムシェアリング・システムでは入出力のデータ量が多い仕事には向かない。そこで端末からジョブ単位の情報をコンピュータへ送ることにより端末側においてセンタ・コンピュータをバッチジョブとして利用できる。前述のディレード・タイム処理方式と関連づけて説明するとよい。

1.5 データ通信システムにおけるデータの流れ

ここでは，システムで取扱うデータ（メッセージ）の種類，データの流れから見たデータ通信システムの種類および端末からインプットされたデータがコンピュータ内でどのようなプログラムの介在によって処理されていくかを簡単に説明する。

(1) システムを流れるデータ（メッセージの種類）

データ通信システムで取扱うデータをメッセージとか，電文とか，トランザクションという表現を使うことがあることに触れる。

(a) 業務メッセージ

システムが目的としている業務処理を行なわせるためのメッセージ（例えば，注文メッセージ・問合わせメッセージなど）

(b) システム運用メッセージ

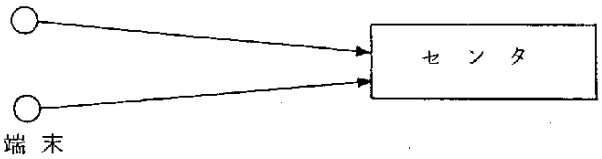
データ通信システムを運用していく上で，必要なメッセージ（例えば，通信開始メッセージ，通信終了メッセージ，テストメッセージなど）

(2) データの流れからみたシステムの種類

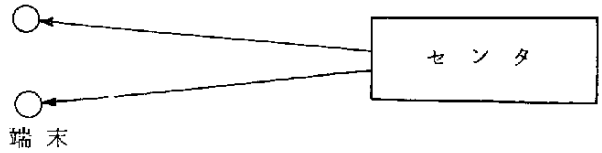
データ通信システムは，主に業務メッセージの流れから見て，次のように分類されている。これらの各システムの特徴について説明する。

- **データ収集システム**
- **データ分配システム**
- **問合せシステム**
- **データ交換システム**

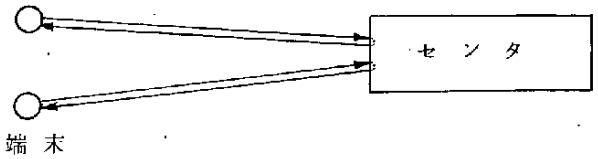
① データ収集システム



② データ分配システム



③ 問合せシステム



④ データ交換システム

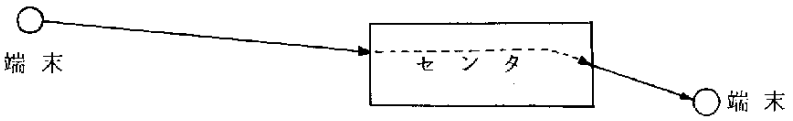


図1－3　データの流れからみたシステムの種類

(3) センタ・システム内のデータの流れ

次図を通してコンピュータ内でデータが処理される過程を説明する。その際，メッセージがインプットされアウトプットされるまで，各プログラムがどのように介在し動作しているかを簡単に説明し，各プログラムの必要性の認識づけを行なう程度にとどめる。詳しくは第5章で説明する。

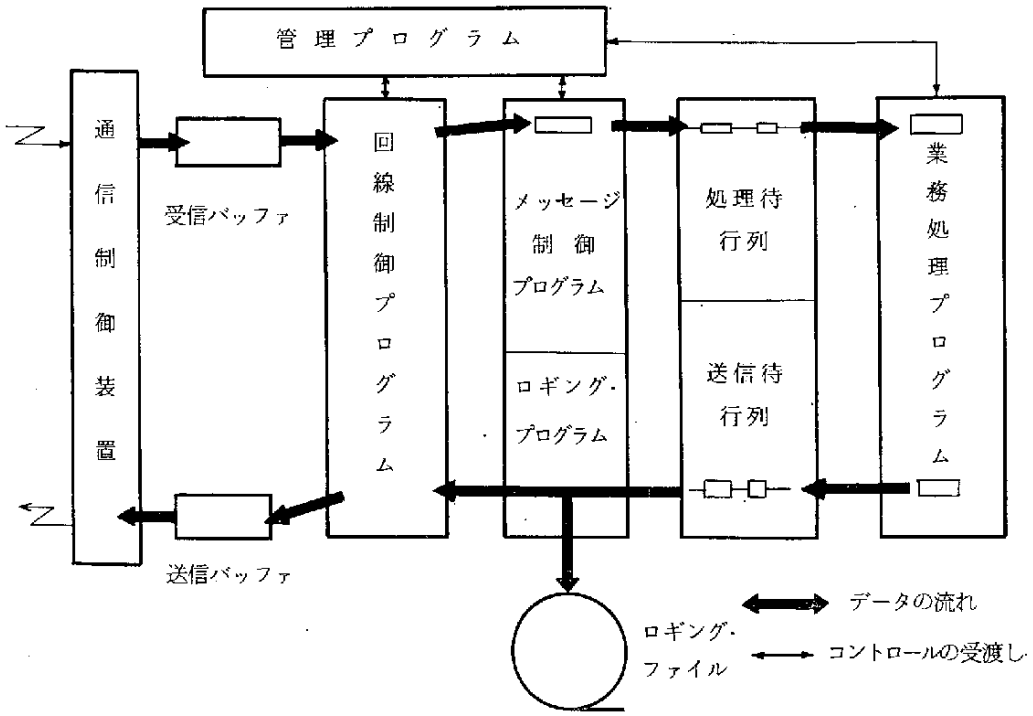


図1－4　データの流れ

1.6 データ通信システムの特徴

データ通信システムを技術面からみた場合，バッチ処理システムと比較して，その特徴は通信回線の存在と，ランダム処理の二点に集約される。そして，その結果派生される数々の技術上の特性が，システムをバッチ処理システムと比較して複雑なものとしている。ここでは実際にどのような点が問題となるかを次の観点から説明する。

(a) データの発生がランダム

(b) データの長さ，種類が一定でない

(c) 応答時間の即時性

(d) 高度の信頼度

(e) 通信制御

(f) 多重並行処理

(g) 割込によるプログラム制御

(h) プログラムのスケジューリングの動的変化

(i) 優先順位

(j) 待ち行列の発生

(k) プログラム・デバグが困難

指導上の留意点

(1) この章の一部は科目「コンピュータ・システム」で説明されていることの復習とする。

(2) データ通信システムに興味をいだかせるように，いろいろな身近な事例をお話し的に取り入れながら話を展開していくことが望ましい。

(3) データ通信システムは，コンピュータの使用目的からして，オンライン•システムであることが望ましい。本科目でもデータ通信システムの狭義の解釈として以降の指導は，オンライン•システムを中心に説明していく。

(4) 各項目については，詳細な説明はなるべくさけ，できるだけ事例をまじえながらデータ通信システムについて大局的に紹介する。

第2章　端末装置

用語　この章では次の用語を教える。

半二重通信方式，前二重通信方式，直列伝送方式，並列伝送方式，調歩同期式，同期式，誤り制御，エレメント誤り率，誤字率，ブロック誤り率，群計数検査，サイクリック・リダンダンシー・チェック，伝送制御方式，伝送制御符号，呼出応答方式，ポーリング／セレクティング方式。

目標

データ通信システムにおける端末装置の位置づけを明示し，端末装置の構成および，通信回線とのインターフェイス特有の機能ならびに，現在使用されている代表的な端末装置の種類について，理解させることを目標とする。

内容

2.1 端末装置の構成

次図を通して端末装置の構成を説明する。端末装置はデータ通信システムにおいて，人間系と機械系の接点となるので，人間工学的な観点から考慮されていなくてはならないことに留意させる。

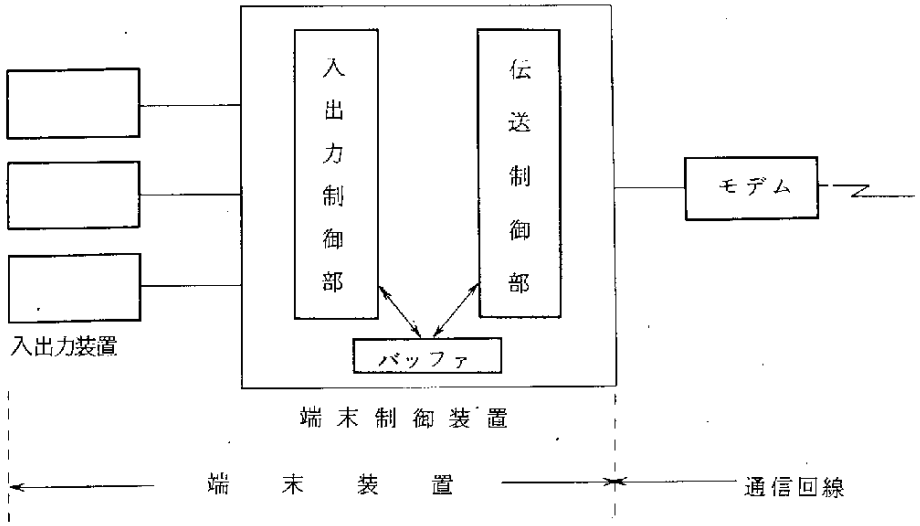


図2－1　端末装置の構成

(a)　入出力装置

従来のバッチ・システムにおけるコンピュータ・センタ内の周辺装置の入出力機器と同等な機能を

もった装置であることを認識させる。

(b) 端末制御装置

別名伝送制御装置とか，ターミナル・コントローラと呼ぶことがある。この装置は，さらに次のものから成り立っていることを説明する。

① 入出力制御部

端末制御部に接続されている入出力機器を制御する部分で，制御機能からみると，次の種類がある。

- 単一機器のみの制御
- 複数機器を制御するが動作は常に一つ
- 複数機器を制御するが動作は入出力各一つ
- 複数機器を同時に動作させる

② バッファ機構

データを一時貯えておく部分で入出力装置の速度と回線速度のアンバランスを調整する。

③ 伝送制御部

回線へデータを送信したり，回線からデータを受信したりするのを制御する部分で，主に回線側とのインターフェイスを受けもっている。次の機能を紹介する。

- 通信路の接続
- 接続相手の確認
- 情報の転送
- 終結
- 通信路の解放

2.2 データ伝送に必要な機能

ここでは，特に端末装置に要求される通信に必要ないろいろな制御機能および用語の意味づけを理解させる。

(1) 通信方式

オンライン・システムにおける端末対コンピュータ間の通信方式を分類すると，次の二つの方式があることを説明する。

(a) 半二重通信方式

伝送方向は，両方向可能であるが，同時には一方向に限られ，伝送方向の切換えにより交互に通信する方法が**半二重通信方式**である。問合せシステムでは，この方式が採用されることを教える。

(b) 全二重通信方式

同時に両方向の通信が可能な方法が**全二重通信方式**である。メッセージ交換システムではこの方式が採用されることを教える。

(2) 伝送方式

データ通信で取扱われる伝送文字は，一般に数ビットから構成されている。これを送る方法として，

時間的に連続して１ビットづつ伝送する**直列伝送方式**と１文字を構成する全ビットを同時に伝送する**並列伝送方式**を紹介する。

(a) 直列伝送方式

直列伝送方式には，次の２種類があることを説明する。

① **調歩同期式**

文字を伝送する場合，その前後にスタート・ビット，ストップ・ビットを付加して１文字単位に同期をとる方法であることを理解させる。

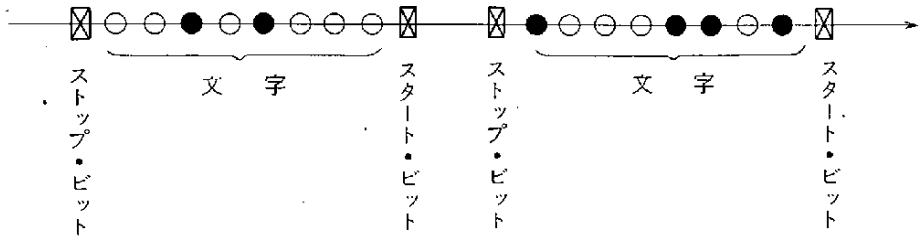


図 2－2　調　歩　式

同期のとり方として，次のことを教える。

・受信側は，スタート・ビットによりクロックをスタートさせる。

・その後クロックによって，各ビットのサンプリングを行なう。

・ストップ・ビットが来ると１文字の受信完了となる。

② **同期式**

文字ごとに同期をとるのではなく伝送のはじめに同期信号を送って送受信の同期をとっておく方法であり，伝送効率がよいことを理解させる。

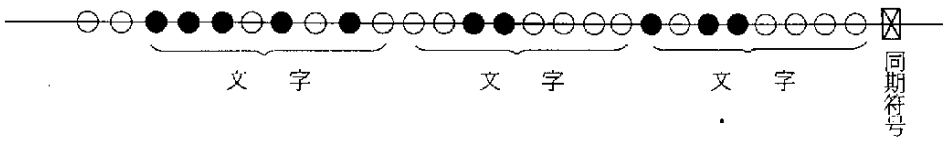


図 2－3　同　期　式

(b) 並列伝送方式

並列伝送のしくみを説明するとともに，オンライン・システムでは経済的に不利のため，ほとんどこの方式は採用されないことにもふれる。

(3) 伝送コード

ここでは，現在の標準コードとしてのＪＩＳコードを提示し，コード表の見方ができるようにさせるとともに，ＪＩＳコードを使用してデータを伝送する場合，伝送上の１文字は垂直パリティ・ビットを考慮すると10ビットから構成されることを理解させる。

表2－1　ＪＩＳ７単位情報交換用符号

					SHIFT IN 側								SHIFT OUT 側			
				b_7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	R＼C	0	1	2	3	4	5	6	7、	8	9	10	11	12	13	14	15
				0	0	0	0	0	NUL	DLE	SPACE	0	@	P	ｐ	p	空白		間隔	ー	タ	ミ		
				0	0	0	1	1	SOH	DC_1	'	1	A	Q	a	q	。		°	ア	チ	ム		
				0	0	1	0	2	STX	DC_2	!!	2	B	R	b	r			「	イ	ツ	メ		・
				0	0	1	1	3	ETX	DC_3	£(#)	3	C	S	c	s			」	ウ	テ	モ		
				0	1	0	0	4	EOT	DC4	$	4	D	T	d	t			、	エ	ト	ヤ		
				0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u			．	オ	ナ	ユ	゛	
				0	1	1	0	6	ACK	SYN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
				0	1	1	1	7	BEL	ETB	'	7	G	W	g	w			ァ	キ	ヌ	ラ		
				1	0	0	0	8	BS	CAN	(	8	H	X	h	x			ィ	ク	ネ	リ		
				1	0	0	1	9	HT	EM	)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
				1	0	1	0	10	LF	SUB	*	:	J	Z	j	z			ェ	コ	ハ	レ		
				1	0	1	1	11	VT	ESC	+	;	K	〔	k	{③			ォ	サ	ヒ	ロ		
				1	1	0	0	12	FF	FS	,	<	L	③	l	−③			ャ	シ	フ	ワ		
				1	1	0	1	13	CR	GS	−	=	M	〕	m	}③			ュ	ス	ヘ	ン		
				1	1	1	0	14	SO	RS	.	>	N	∧	n	|③	SO		ョ	セ	ホ	゛		
				1	1	1	1	15	SI	US	／	?	O	−	o	DEL	SI		ッ	ソ	マ	°		抹消

(4) 誤り制御方式

誤り制御の必要性を認識させるとともに，伝送エラーの検出，訂正のためのいろいろな対策方法を紹介し，それらの種類，特徴を理解させる。

(a) 伝送エラーの原因について次のことを説明する。

- 伝送路を構成する各装置の誤動作
- 伝送路で生ずる歪み
- レベル変動
- 外部雑音（ピーク性雑音，インパルス雑音 等 ）
- 瞬　断

(b) 誤　り　率

通信回線の品質を示す尺度について，次のことを説明する。

- **エレメント誤り率**
- **誤字率**
- **ブロック誤り率**

(c) 誤りチェック方式の種類

- 垂直パリティ・チェック
- 水平パリティ・チェック

・**群計数検査**

・定マーク，定スペース・チェック

・**サイクリック・リダンダンシー・チェック**

・ハミング符号・チェック

・返送方式

・連送方式

(5) 伝送制御方式

端末とコンピュータ間で，データまたはメッセージをやりとりする場合，現在のところ2種類の制御方式が確立されている。これらの制御方式の相違は　端末，コンピュータ間の回線接続形態のちがいに起因するものであることを説明する。同時に**伝送制御方式**を理解する上での代表的な伝送制御符号の意味または，動作ならびにメッセージの形式について説明する。

(a) 伝送制御符号

次の表を使用して**伝送制御符号**の意味および動作について説明する。

表2－2　代表的な伝送制御符号

符号名	符号の意味または動作
SOH	データの始め符号
STX	テキスト（ブロック）の始め符号
ETX	テキスト（ブロック）の終り符号
EOT	データ伝送の終り符号
ETB	ブロックの終り符号
ENQ	相手からの応答を要求する符号
ACK	受信データが正しい事の返送符号
NAK	受信データが誤りであることの返送符号
CAN	無効データを表示するための符号
SYN	送受信間の同期をとるための符号
DLE	この符号の後に制御符号が続くことを示す

(b) メッセージの形式

メッセージの形式について説明する。しかしこれは定型化されたものではなく端末装置の仕様とかシステム運用に深く関係する。そこでここでは，代表的な次の例を紹介する。

(c) 伝送制御方式

呼出応答方式，ポーリング／セレクティング方式について説明する。

その際，メッセージ送受信を制御する回線制御プログラムと関連づけて説明する。

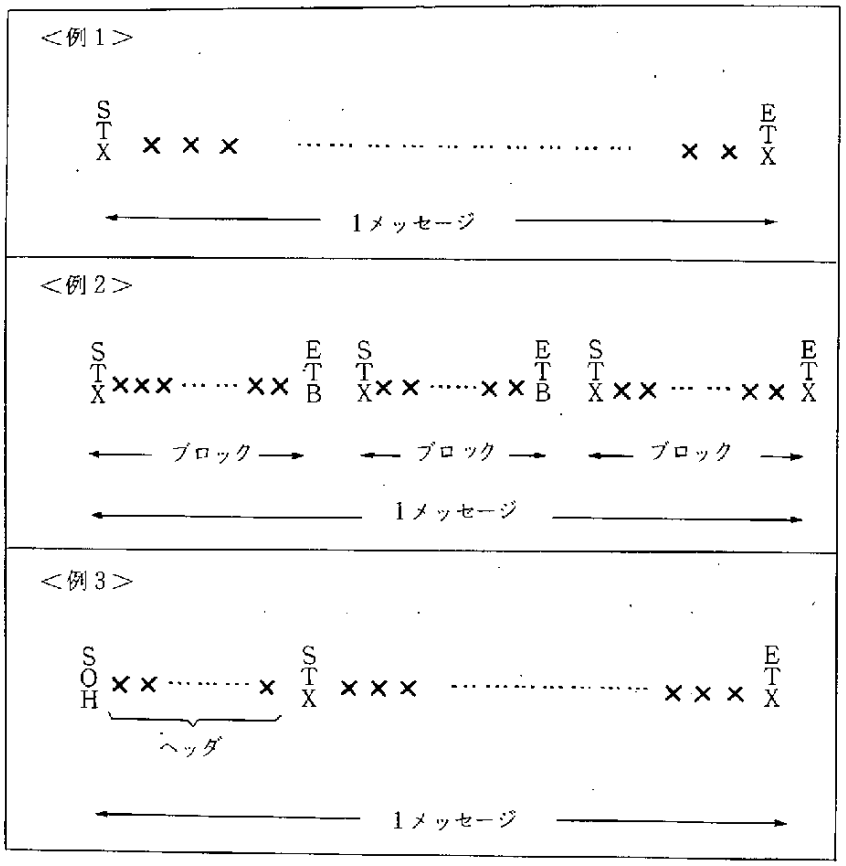


図2－4　メッセージ形式例

① 呼出応答方式

次図の伝送制御手順を通して，呼出応答方式*の制御方法を説明する。

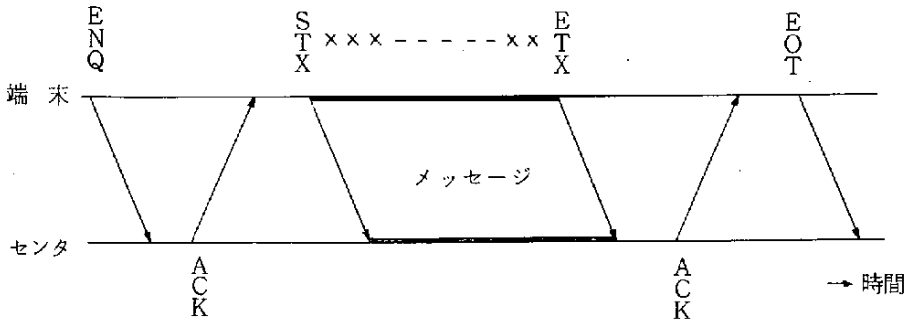


図2－5　端末送信の場合

* コンテンション方式ともいう

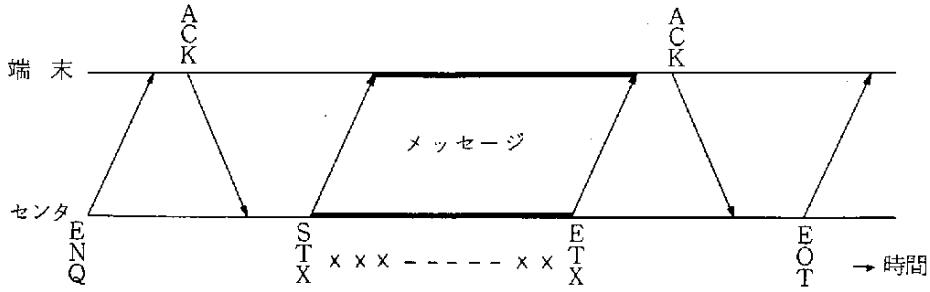


図2－6　センタ送信の場合

② ポーリング／セレクティング方式

次図の伝送制御手順を通して，ポーリング／セレクティング方式の制御方法を説明する。

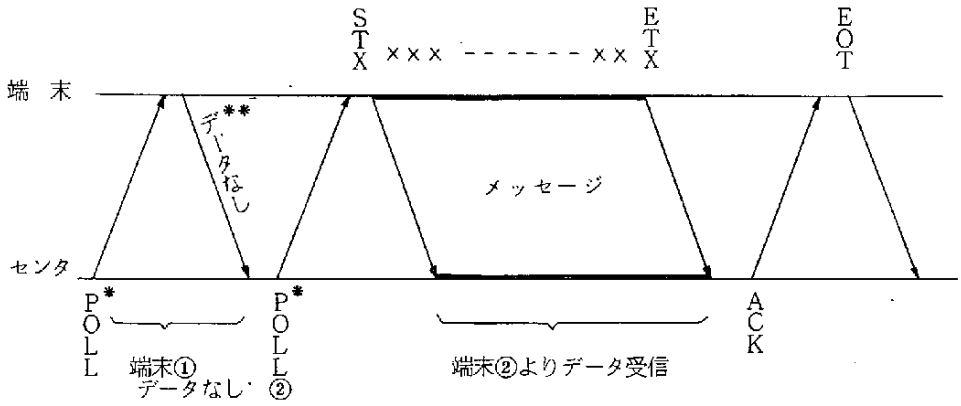


図2－7　ポーリングの場合

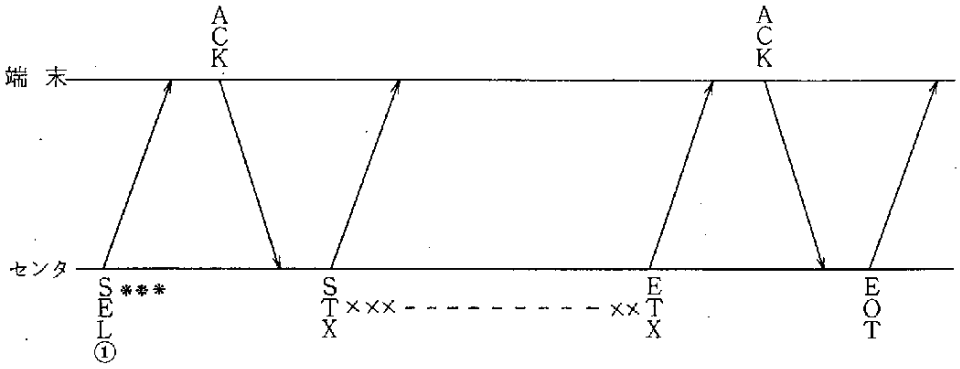


図2－8　セレクティングの場合

* ポーリング・シーケンス符号
** 放棄シーケンス符号またはNo Response
*** セレクティング・シーケンス符号

分岐回線構成の場合には，回線争奪を防止することから，この制御方式がとられる。

2.3 端末装置の種類

公衆通信回線の開放および技術の進展にともない端末装置は多様化していく傾向にある。ここでは，次の観点から端末装置の種類を大別して説明する。

(1) 用途からの分類

① 汎用端末装置

② 専用（特定）端末装置

(2) 入出力方法からの分類

① 記録媒体を用いて入出力するもの

媒　　体	代表的機器名
紙テープ	紙テープ・リーダ
	紙テープ・パンチ
紙カード	カードリーダ
	カードパンチ
シート（帳票）	プリンタ
	マークシート・リーダ（MSR）
	OCR式文字読取装置，MICR式文字読取装置
磁気テープ	磁気テープ装置

② 記録媒体を使用せずに出力するもの

項　　目	代表的機器名
視感（目）	CRT
	表示盤
音　　声	音声応答装置
	プッシュホン（押ボタン式電話機）
手　　指	押しボタン
	キーボード
物理量	テレメータ各種検出装置（車輌，水量，圧力，温度）

2.4 端末装置の代表例

現在，データ通信システムでよく使用されているものを中心に，次の観点から説明する。各装置の構成，動作概要，特徴，適用業務などを紹介する。 その際メーカが準備している具体的装置を設定して装置の性能などについても触れるとよい。

(a) 紙テープ・ターミナル

(b) キーボード・プリンタ

(c) 汎用端末装置（データステーション）

(d) 押ボタン式ターミナル

(e) マークシート／プリンタ・ターミナル

(f) OCRターミナル

(g) CRTディスプレイ装置

(h) その他の端末装置として，次のものを簡単に紹介する。

- プッシュホン
- POS（Point of Sales）ターミナル
- インテリジェンス・ターミナル
- 電光掲示板
- 磁気テープ伝送装置
- データ収集装置
- テレメータ
- 銀行用端末装置
- エージェント装置

導上の留意点

(a) 端末装置は，データ通信システムにおける機械系と，それを使用する人間系の接点として重要な役割をもっている。そのため，端末装置は人間工学的な観点に立って設計作製されていなければならないことを認識させる。

(b) 通信回線の開放によりこれからは，いろいろな種類または機種をもった装置が開発されることが予想されるので，随時それらの新しい技術を持った端末装置の動向などをトピックとして取り入れて説明する。

(c) 端末装置の性能の判断基準として伝送機能に関する用語については十分理解できるように解説する。

第3章　通　信　回　線

用　語　この章では次の用語を教える。

変調速度，信号速度，伝送速度，変復調装置，変調方式，特定通信回線，直結回線方式，分岐回線方式，集線方式，公衆通信回線，網制御装置

目　標

通信回線は遠隔地の端末装置と中央のコンピュータ・システムとを接続し，データの通路としての意味をもつ。一部の私設回線を除いて，企業体がオンライン・システムを導入する際の通信回線は電々公社から借用することになる。

本章では，この事を前提に通信回線を理解する上で必要な基本的事項および現在データ通信システムに提供されている回線の種類，特徴またこれらの回線を使用する際の回線構成の留意点などについて理解させることを目標とする。

内　容

3.1 通信速度と回線のしくみ

(1) 通信速度の表示方法

通信速度の表現方法には大別して次のものがある。これらの特徴および使用のされ方を紹介する。

① **変調速度**

従来から電信の速度表示に使用され，単位はボー（Baud）で表わす。

② **信号速度**

この速度は1秒間に伝送できる総ビットのことであり，単位はビット／秒（BPS）で表わす。現在の通信速度の標準表示方法となっていることを認識させる。

③ **伝送速度**

この速度は1分間に伝送できる文字数によって示される。単位はキャラクタ／分で表わす。

通信速度を決める要因として一般に回線の帯域幅と信号対雑音比に影響されることにも触れるとよい。

(2) 回線のしくみ

通信回線の種類には特定（専用）通信回線と公衆通信回線がある。ここでは2線式と4線式について説明し，それらのちがいは何かの判断ができるようにする。

① 2線式（2W）

伝送路の基本は1対の線路すなわち2線式（2W）で構成されることを理解させる。また1200

bps 以上の通信速度においては全二重通信ができないことに注意させる。

② 4線式（4W）

通信速度に関係なく全二重通信が可能である。2W，4Wの区別は電話品質回線の引込線の構成により大別されることに留意させる。また通信方式の全二重，半二重通信と対比して説明するとよい。

3.2 変復調方式

ここでは**変復調装置**（MODEM: Modulator Demodulator Equipment）の位置づけを明示し，この装置のはたらきとして簡単に，変調／復調とは何かを説明した上で，データ通信に適した**変調方式**の種類と特徴について説明する。

- 振幅変調方式
- 周波数変調方式
- 位相変調方式
- パルス符号変調方式

3.3 通信回線

企業体がデータ通信システムを導入する場合，その企業体自身で保有する通信回線（一般に私設回線という）を利用する方法と電々公社の通信回線を利用する方法とがある。後者の場合については更に二つの方法がある。一つはコンピュータや端末機器を企業体自身が準備し，電々公社から通信回線のみを借用する方法であり，もう一つは，通信回線だけでなく，コンピュータも端末機器も全て電々公社から借りる方法である。

- データ通信回線サービス ―― 特定通信回線
  - 公衆通信回線
- データ通信設備サービス ―― 特定システム
  - 共同利用システム

ここではデータ通信回線サービスを中心に話を進めていくとよい。

(1) 特定通信回線

特定通信回線は，利用者が希望する特定の区間で，利用者が選定し，電々公社が設置する特定の電気通信回線に，自営のコンピュータや端末機器を接続して使用する回線であることを説明する。

(a) 回線の種類

特定通信回線には通信速度のちがいからA規格からJ規格までの種類があることを紹介する。

(b) 回線構成の基本

対象システムの回線網を設計する際の基本となる三つの形態について触れ，それぞれの意図する特徴は何かを理解させる。

① **直結回線方式**

- 端末ごとに1回線づつセンタ直結する形態であること。
- データ量が比較的多く発生し，応答時間の即時性の要求が強い場合に適していること。

・通信コストが割高となる。

② **分岐回線方式**

・センタから同一方向の延長上に点在する回線利用率の低い端末同志を１回線で接続する方法

・通信コストの経済化をはかるのが目的となる。

・直結回線方式より即時性に劣る。

・センタでの回線制御が複雑になるきらいがある。

分岐の方法については次図を用いて簡単に説明する程度でよい。

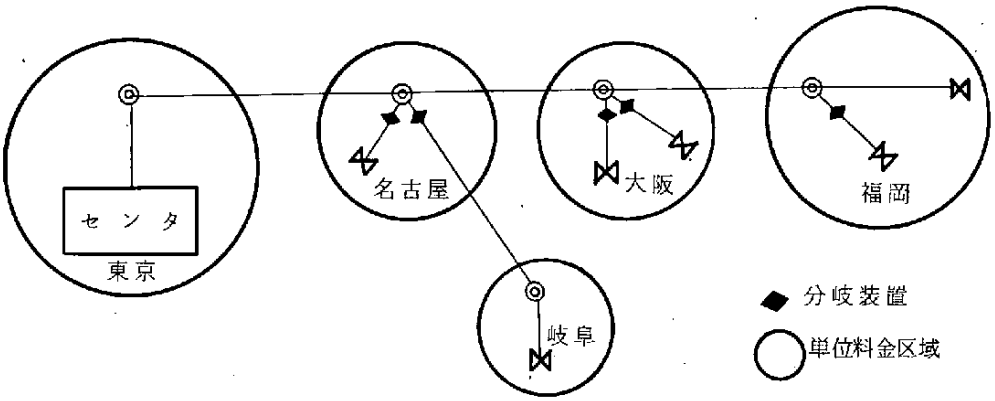


図 3 － 1　分岐回線方式

③ **集線方式**

・端末が同一地域に比較的多く散在する場合，集線装置等で回線を集約し，センタまで少数回線で接続する方法

・回線コストを軽減するのが目的

・集線機能を持つ装置については次のものがある。これらについて簡単に紹介する。

集配線中継装置

集線装置

回線交換装置

小型コンピュータ

(c) 回線料金

回線料金については，いろいろな制度上の問題がある。ここでは，導入時の一時費用と月額使用料があることを認識させ，特に回線使用料を資料として用意し，その料金表の見方を認識させる程度でよいであろう。

(d) 使用契約の種類

特定通信回線サービスを利用する場合，その制度上から見た標準的な使用形態として次のものがあ

る。ここではそれらの各違いを認識させる。

① 単独使用

② 共同使用

③ 他人使用

共同使用，他人使用のシステムでは，メッセージ交換としての使用は許されないことに注意させる。

(2) 公衆通信回線

公衆通信回線は，従来加入電話サービスのために確立してきた加入電話網，またはテレックス・サービスを目的に設備されている加入通信網に利用者が設置するコンピュータや端末装置を接続し，本来の電話，テレックス以外にデータ通信等に利用できることを認識させる。

(a) 回線の種類

公衆通信回線には次の2種があることを説明し，それぞれの回線の特性を理解させる。

① 加入電話回線

加入電話のネットワークを利用するもので，1200 bps 以下の交流符号伝送ができることを説明する。

② 加入電信回線

加入電信（Telex）のネットワークを利用し，50 bps の直流符号伝送ができることを説明する。

(b) 回線接続形態

公衆通信回線を利用して，データ通信を実現するための端末装置およびコンピュータの接続方法について次の図をもとに説明する。

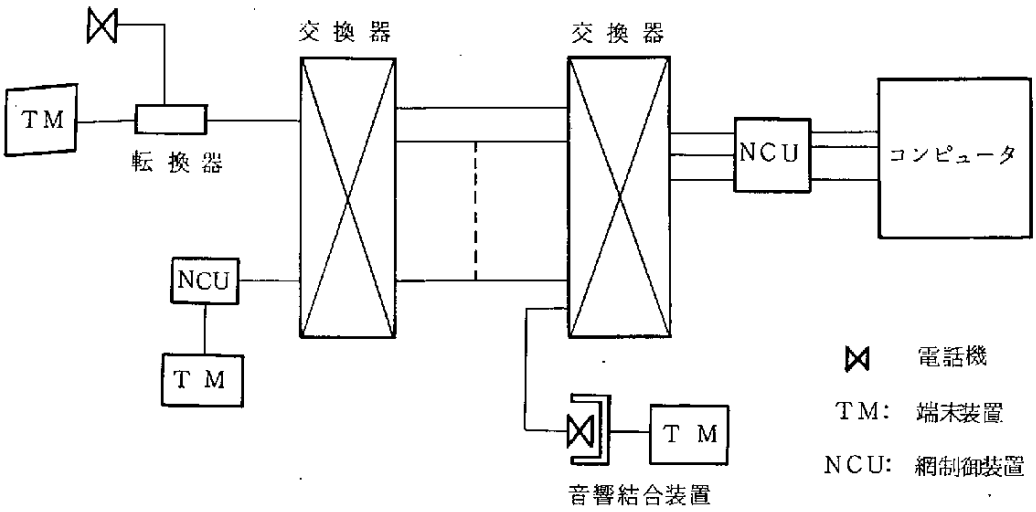


図3－2　加入電話交換網の例

① 公衆通信回線に直接端末装置もしくはコンピュータを接続する場合特定通信回線では必要としなかった**網制御装置**（NCU：Network Control Unit） が必要になることを認識させる。

② 以下に，NCUの機能と種類について紹介する。

主な機能としては

- 交換機の起動
- 相手のダイヤル番号の送出
- 回線接続を認識し，相互間の通信路の接続を確保する。
- 通信終了後，交換機を復旧させ回線を開放する。

(c) 回線使用料

公衆通信回線の料金面での特徴は，通信距離と通信回線の利用時間長に対応した従量制であることを認識させるとともに特定通信回線料金と対比する判断基準ができる程度でよいであろう。

指導上の留意点

(1) 回線網を組む際問題となるのはデータ量から回線利用率の算出を行なう必要がある。これら算出方法についても指導するとよい。

また，利用率の評価として一般によく使用される待ち行列理論から利用率と待ち個数の関係についても説明する。

(2) 特定回線と公衆回線の比較をいろいろな観点から示し，両者を選択する際の判断基準ができる能力を養う。

(3) 通信回線を申請する際のいろいろな手続きとか，利用する際の細かな技術基準については説明する必要はない。

(4) ケースによっては分岐の詳細な規則とか，一つのモデルを設定して回線料金の算定などについても説明してよいだろう。

第4章　センタ・システムのハードウェア

用　語　この章では次の用語を教える。

ライン・アダプタ，局内端末，インターバル・タイマ，時刻装置，システム・コンソール，切換装置，オンライン・アダプタ，単一システム，機能分割システム，業務分割システム，データ量分割システム

目　標

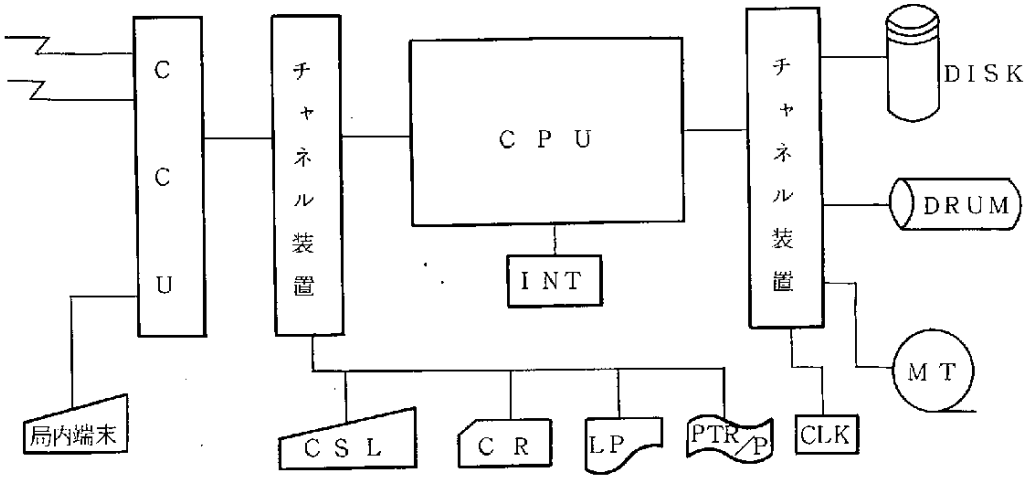
オンライン・システムにおける，センタシステムを構成するハードウェアは，ユーザのシステム規模，業務内容，経済性などによって，各種のシステム構成が考えられる。また大半は，バッチ処理システムの場合と共通するところがある。

ここでは，センタシステムを構成する各装置の機能，構成，および用途について説明し，データ通信システムからみた各装置の位置づけ，および使用のされ方を理解させることを目標とする。

内　容

4.1　センタ・システム概要

ここでは，次図を通して，センタ・システムを構成する各ハードウェアの概要紹介を行なう。



CCU：通信制御装置　　LP　：ラインプリンタ　　DISK：磁気ディスク装置
CPU：中央処理装置　　PTR／P：紙テープリーダ／パンチ　　DRUM：磁気ドラム装置
CSL：コンソール・タイプライタ　　INT：インターバル・タイマ　　M　T：磁気テープ装置
CR　：カードリーダ　　CLK：時刻装置

図4－1　　センタ・システム

(a) 各装置の使用目的，中央処理装置との接続関係を簡単に説明する。

(b) センタ・システムの果すべき役割として，次の事柄について説明する。

① 端末装置および通信回線のコントロール

② コード変換

③ ファイル処理

④ メッセージの処理

⑤ 処理の流れの制御

⑥ マンマシン・コミュニケーション

⑦ 障害処理

4.2 センタ・システムのハードウェア

オンライン・リアルタイム・システムの必要性からみた，各装置の機能，特徴，用途について説明する。

(1) 中央処理装置

コンピュータ・システムの中枢をなすもので，基本的機能については科目「ハードウェア(I)」の復習をする。

(a) 割込機能については，次の観点から説明する。

・割込機能の導入目的

・ハードウェアの割込時の処理

・割込の種類

・割込の原因

(b) 記憶保護機能について説明する。

・この機能の必要性と，プログラムの多重並行処理の関係

・記憶保護エラーと割込の関係

(2) 主記憶装置

ここでは主に，プログラム領域として使用されることを明記する。

(a) プログラム領域としては

・管理プログラム

・処理プログラム

・オーバーレイ・プログラム

(b) データ領域としては

・メッセージの送受信バッファ

・各種テーブル

などに使用されるので，システムの性格により最適な容量を待つ主記憶装置を使用しなければならないことに触れる。

(3) チャネル装置

中央処理装置は，入出力装置に比べて動作時間が格段に速い。 そこでチャネル装置の時分割制御機能による出力装置と中央処理装置の同時並行処理の考え方を説明する。

(a) チャネル装置の主な役割として，次のことを説明する。

① 中央処理装置から入出力命令の指示を受けて，入出力装置を起動させる。

② データ転送の制御を行なう。

③ 入出力動作を終結させる。

④ 入出力装置からの割込を中央処理装置へ知らせる。

(b) チャネル装置には，次の種類があることを説明する。

① マルチプレクサ・チャネル

低速入出力装置の接続を制御する。特に通信制御装置は，このチャネルを使用してメッセージの送受信を効果的に行なう点に触れる。

② セレクタ・チャネル

高速入出力装置とのデータ転送を制御する。

(4) 通信制御装置

通信回線と中央処理装置の間に位置し，メッセージの送受信を混乱なく行なえるような工夫がされている。

ここでは，次の事柄について説明する

(a) 回線との接続関係

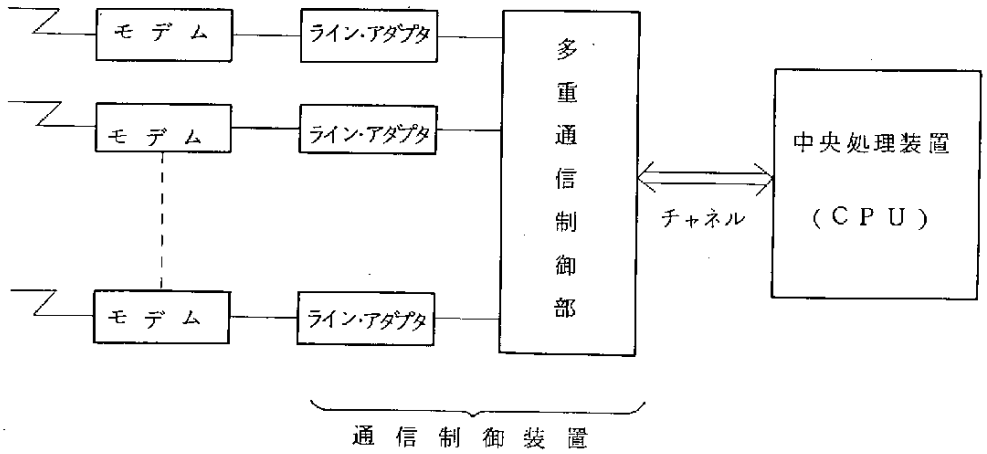


図4－2　　回線との接線関係

① モデムについては，第3章で説明してあるのでここでは，簡単に復習程度にとどめるとよい。

② **ライン・アダプタ**については，回線ごとに必要でありその主な機能はモデムから入力されるビット情報のサンプリングとその逆を行なうことを認識させる。

③ 多重通信制御部（狭義の意味の通信制御装置）は，チャネル装置を介してＣＰＵへ接続されている。

(b) 通信制御装置の機能

システムによって，若干その機能については異なる点もあるが，ここでは次の各機能を紹介して仕組を理解させる。

① 直並列変換機能

回線から送られてくるビット単位の情報を文字に組立てたり，逆に文字をビット単位の情報に分解する。

② バッファリング

回線スピードと中央処理装置の速度は格段の相異があるので，バッファ機構を用いて速度の調整をはかっている。

③ 中央処理装置とのデータ転送制御

受信指令の時には，通信制御装置内のバッファに文字データが貯るとチャネル装置を経由して主記憶装置の受信バッファへ転送される。逆に送信指令の時は主記憶装置の送信バッファから通信制御装置内のバッファが空きになると，データ転送がチャネル装置を経由して行なわれる。

④ 誤り制御機能

伝送エラーを検出するために，受信データに関しては１文字単位に垂直パリティ・チェック およびブロック単位に，水平パリティ・チェックまたは群計数チェック，逆に送信データについてはそれらのビットの付加を行なう。

⑤ 伝送制御符号の検出

たとえば，問合せ符号（ＥＮＱ），始め符号（ＳＯＨ，ＳＴＸ），終り符号（ＥＴＸ，ＥＴＢ，ＥＯＴ），応答符号（ＡＣＫ，ＮＡＫ）などを自動的に検出し，中央処理装置へ割込をかけて通知する。

⑥ 回線状態の監視

モデムとのインターフェイス信号の監視，受信データ途中断の時間監視，各種エラー状態を監視し異常を検出すると中央処理装置へ割込信号を出す。

⑦ 割込機能

伝送制御符号の検出とか送受信バッファの終了の検出または，各種エラー状態検出時に中央処理装置に対して割込信号を発生する。このとき同時にメモリの特定エリアに割込情報を転送することを説明する。

割込情報の一例として次のことを説明する。

・チャネル番号（回線番号）

・割込原因（開始符号，終り符号，伝送エラーなど）

・指　令

・アドレス

(5) 補助記憶装置

ここでは磁気テープ装置，磁気ドラム装置，磁気ディスク装置の基本事項としての記憶媒体の仕組み，特性については，科目「ハードウェア(I)」の復習とする。特に，オンライン・システムにおける使用目的を理解させることに努める。

(a) 各装置の基本事項

・デバイスの物理的構造

・特　性

・性能（容量，アクセスタイム）

・データの記録形式

(b) 各装置の用途として次の事柄を簡単に説明する。詳しくは第6章で説明される。

・マスタ・ファイル

・プログラム・ファイル

・キューイング・ファイル

・ロギング・ファイル

・チェックポイント・ファイル

・各種テーブルの記憶とかワークエリア

(6) その他の装置

ここでは，以下の装置のしくみおよびどのような目的で使用されるかを中心に説明する。

① **局内端末**

システムの信頼性を考え，各端末装置とか，コンソール・タイプライタの障害時のバックアップとして使用する。

② **インターバル・タイマ**

一定時間ごとに，中央処理装置へ割込信号を発生するので，一定間隔ごとにシステム状態が正常か否かをチェックする手段に使われることを説明する。

③ **時刻装置**

データの発生都度この装置から時刻情報を読みとり，データに付加し後で統計資料を作成したり，障害発生時に障害統計として記録し，後でこのデータを分析する。または共同利用システムの場合などは使用時間の測定に使用されることを説明する。

④ コンソール・タイプライタ

オペレータとシステムのコミュニケーションに使用されることに触れる。

⑤ **システム・コンソール**

システムの稼動状況などを表示ならびに制御し，障害時にベルあるいはブザーを鳴らせるとともに，ランプ表示によりオペレータに通知する。

⑥ **切換装置**

待機用システムとか予備機を持つシステムなどで現用と予備機の切換えを行なうときに使用する。切換えは手動で行なうものとプログラムの指定で自動的に行なうものがある。

⑦ **オンライン・アダプタ**

中央処理装置相互間で制御情報とかデータの授受を行なうために使用することを説明する。

4.3 機能面からみたシステム構成

今まで述べてきた各装置を組合せて，システムを構成する場合，その接続方法の組み合せは，事実上無限にあるといえる。これらの組み合せの中から，システムの機能，拡張性，信頼性，経済性など数々の条件を考慮して最適のシステム構成を選択するのは非常な困難な問題である。ここでは，中央処理装置を中心に見た構成を説明する。またこれらの各構成が全てではなく，むしろこれらの構成は基本的なもので，実際のシステムはこれらを応用し自由に，設計することが必要である。ここでは，各システム構成の特徴を対比させながら説明するとよい。

- **単一システム**
- **機能分割システム**
- **業務分割システム**
- **データ量分割システム**

これらのシステム構成は，予備機とか二重化などの信頼性対策に欠ける面があるが，ある程度の信頼性からみた特性も備えていることに留意させる。信頼性からみたシステム構成については第７章で取扱う。

指導上の留意点

(1) この章の内容はかなりの部分が「ハードウェア(I)」編と共通するところがある。これらについては復習程度にとどめて説明するとよい。それらを制御する具体的な命令等を出してもよい。

(2) 通信制御装置の仕組みについては，詳細に説明する。システムによっては，ハードウェアの仕様があるのであるモデルを設定するとよい。その際回線制御プログラムとある程度関連づけて紹介する。

(3) 他のハードウェア構成要素の説明にあたっては，それらがどのようにデータ通信システムの中で，使用されているか，また使用したらよいのかといった観点から紹介する。

第5章　オンライン・システムのソフトウェア

用　語　この章では次の用語を教える。

多重プログラミング制御，入出力制御，通信制御，バッファ制御，ライン・バッファ，システム・バッファ，待ち行列制御，処理待ち行列，送信待ち行列，システム・スタート・プログラム，システム・リスタート・プログラム，ロギング・プログラム，システム・クローズ・プログラム，マンマシン・コミュニケーション・プログラム，システム診断プログラム，リカバリ・プログラム

目　標

オンライン・システムのソフトウェアといっても独立して存在するのではなく，それはオペレーティング・システムの体系に包含されているのが一般的である。ここでは特に，オンライン・システム特有な通信管理機能を中心に説明する。これらの各機能がデータ通信システムから見てどのように生かされているか，また，コンピュータ内でどのようにメッセージが処理されていくか，その際ソフトウェアを構成する各プログラムがどのように介在するかといった観点から把えて，ソフトウェアの構成，機能，特徴について理解させることを目標とする。

内　容

5.1　オンライン・システム用ソフトウェアの特徴

オンライン・リアルタイム・システム のソフトウェアは，一般にバッチ・システムのソフトウェアに比べて複雑で難しいものと受け取られている。その理由としては，データの発生がランダムであり，データ長が不均一であり，多数の端末の同時制御であり，伝送制御の複雑さや，応答時間の要求のきびしさからくるものである。これらの要因をソフトウェアで対処していくために，次のような特徴をオンライン用ソフトウェアが持っていることを説明する。

① プログラムの多重並行処理
② プログラムのスケジューリングの動的な変化
③ メッセージ種別による優先処理
④ 割込みによるプログラム制御
⑤ 待行列制御
⑥ メモリの動的割当

5.2　オンライン・システム用ソフトウェアの構成

オンライン・システムのソフトウェアは，その機能面から分類すると次のようになる。

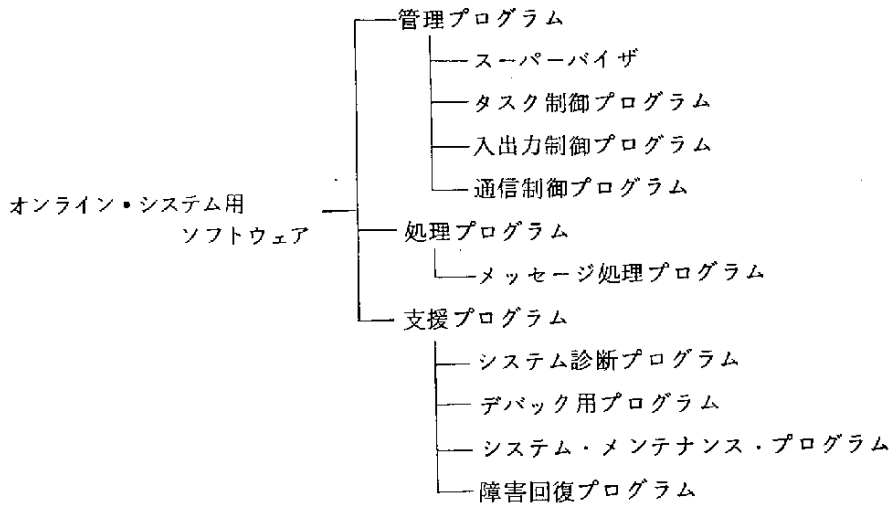


図５－１　オンライン・システム用ソフトウェア

ここでは，各コンポーネントの概要紹介を行なうことにより，今後詳細に説明する個々のプログラムの位置づけを行なう。一般にはメーカが提供するオペレーティング・システムの体系に含まれることに留意する。

5.3 管理プログラム

管理プログラムはシステム全体を統括および管理するプログラム群から構成され，一般にはメーカから提供される。ここでは管理プログラムの機能をオンライン・システムからの必要性という観点から説明する。

(1) 割込制御機能

割込処理は，バッチ処理システムに限らずオンライン・リアルタイム・システムにおいても重要な機能となる。ここでは第4章のハードウェアで説明した中央処理装置の資源である「割込」を，その利用技術としてのソフトウェアから把えて次の事柄について説明する。

① ハードウェアの割込時の処理を復習する。

② 割込のソフトウェア利用技術について，その代表的割込を把えて説明する。

- コンソール割込
- 入出力割込（ＣＣＵを含む）
- 割込命令による割込
- タイマ割込
- プログラム割込

(2) 多重プログラミング制御

プログラムの多重並行処理を管理しているのは，タスク制御プログラムである。ここでは多重プログ

ラミング制御について次の事柄を説明する。

(a) コンピュータ・システムにおけるマルチプログラミングの導入目的について触れる。

(b) タスクとジョブの概念を説明する。

(c) タスク状態とタスク・コントロール・ブロックについて次の項目について説明する。

① TCB（タスク・コントロール・ブロック）

タスク制御プログラムが各タスクの実行を管理する制御テーブルである。

② タスクの状態を次のように分類し管理している。

・停止状態

・動作状態 { RUN（実　態）／READY（実行待状態）／WAIT（事象待状態） }

(d) タスクの優先順位

(c) 次のような場合，実行権は他のタスクへ移ることを以下の例を用いて理解させる。

① 実行すべきタスクが終了し，その旨をタスク制御プログラムへ通知した場合

② 何んらかの事象が完結するまで，タスクの進行を中止せざるを得ない場合

③ 優先順位の高い他のタスクの事象が完結した場合

④ 制御プログラムに入出力動作などを依頼する場合

(3) 入出力制御

オンライン・リアルタイム・システムにおいては，一般に前述したようなマルチプログラミング技法を用いて，複数タスクが同時並行処理されている。それぞれのタスクが勝手に入出力装置をアクセスしたのでは，システムが混乱状態になり効率よい処理は望めない。そこで**入出力制御**プログラムが一括して入出力装置（補助記憶装置を含む）を管理している。ここでは入出力制御プログラムの機能について次の事柄を説明する。

① ボリューム／ファイル・ラベルの処理

② データの読取／書込み

③ データのブロッキング／デブロッキング

④ ファイルの共用制御

⑤ 入出力エラー処理

⑥ 入出力命令の待行列制御

⑦ チャネルのスケジューリング

⑧ 入出力マクロ命令

(4) 通信制御

通信制御とは，通信制御装置を介してシステムに収容されている端末機とメッセージの送受信を行なうことを管理するものでオンライン・システムのソフトウェアにおいて最も重要な機能である。この機能を行なうプログラムが通信制御プログラムである。ここでは，次の事柄について説明する。

(a) 通信制御プログラムの特徴

① 通信制御装置に継がる沢山の端末機を同時制御しなければならない。

② 端末機は遠隔地に散在するため，データ通信に伴うエラーやメッセージの紛失可能性が大きいので十分なチェックおよびエラー処理を考慮しなければならない。

③ 端末機の種類，伝送制御方式，伝送制御符号，通信速度が異なりそれぞれに適切なサービスをしなければならない。

④ 応答時間の即時性が要求されるため，できる限りプログラムの走行ステップが少なく効率のよいものでなければならない。

(b) 通信制御プログラムの構成

通信制御プログラムを機能面から分類すると，次のようになる。

・通信管理プログラム

・回線制御プログラム

・メッセージ・コントロール・プログラム

以下にこれらのプログラムの機能を中心に説明する。

(c) 通信管理プログラム

通信制御の共通制御部分で次の機能を受け持つ。

① 通信制御装置から割込情報を受けとり，割込を起した回線の回線制御プログラムへコントロールを渡す。

② オンライン・リアルタイム・システムで使用するマクロ命令の管理およびサービス

③ メッセージの待行列制御プログラムとのインターフェイス

④ バッファ群の管理

(d) 回線制御プログラム

コンピュータ・システムに接続される遠隔地の複数の端末装置とのメッセージの送受信を制御するプログラムである。このプログラムは端末機の仕様とかシステム運用方法などへの依存が非常に高いので，対象システムごとに作成し管理プログラムへ組み込むことになる。

ここでは，次の事柄について説明する。

・回線のOPEN／CLOSE

・受信開始準備

・メッセージの受信

・メッセージの受信終了

・送信メッセージ待行列のサーチ

・送信開始準備

・メッセージの送信

・メッセージの送信終了

次の例を用いてメッセージの受信制御を説明するとよい。

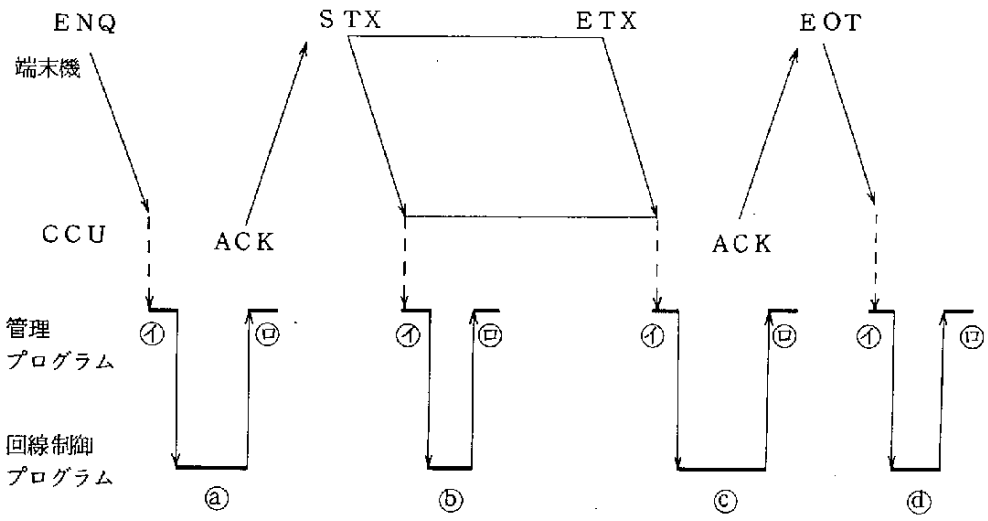


図5－1　メッセージの受信制御

(e) メッセージ制御プログラム

回線制御プログラムとメッセージ処理プログラムとの間に位置しシステムの要求にあった入力メッセージの制御を行なうプログラムである。

メッセージ制御プログラムは次のような機能を持っていることを説明する。

- インプット・メッセージのコード変換
- インプット・メッセージのフォーマット・チェック
- 通番のセット
- インプット・メッセージの識別
- メッセージの処理プログラムの選定
- メッセージの処理待行列への登録
- メッセージのロギング要求
- オペレータとの連絡

このプログラムも標準的に，システムに共通する部分と，システムごとに運用が異なるため準備しなければならない点では，回線制御プログラムと考えは同じである。これらの処理機能は設計段階で一部その機能を回線制御プログラムへ持たせたり，処理プログラムへその機能を入れることも考えられるので，システムごとに作成するのが一般的であろう。

(5) バッファ制御

オンライン・リアルタイム・システムでは，メモリの有効使用をはかるためにバッファ管理機能が必要とされる。使用バッファはプールにしておき必要なときそこから取り出し，使用後はそこに返却する方法が一般に用いられている。ここでは，**バッファ制御**について次の事柄を説明する。

(a) バッファの種類とその用途

- **ライン・バッファ**
- **システム・バッファ**

(b) バッファ制御情報

- 使用可能バッファ個数
- 取得バッファ・アドレス
- 返却バッファ・アドレス

(c) バッファ取得／返却用マクロ命令

- GETBUF
- PUTBUF

(6) 待ち行列制御

ここでは，オンライン・リアルタイム・システムにおいてメッセージ待ち行列の必要性を認識させるとともに**待ち行列制御**について次の事柄を説明する。

(a) メッセージ待ち行列の種類

- **処理待ち行列**（CIQ：Communication Input Queue）
  メッセージ処理プログラムの単位に作られる
- **送信待ち行列**（COQ：Communication Output Queue）
  送信端末ごとに作られ，数レベルの優先順位を持たせられるので送信メッセージの追越が可能となる。

(b) 待ち行列のサービス方法

- FIFO（First In First Out）サービス
- 優先順（Priority）サービス

(c) 待ち行列ファイルと媒体

- コア
- 磁気ディスク
- 磁気ドラム

(7) メッセージ処理プログラムとのインターフェイス

ここでは，メッセージ処理プログラムが端末装置との入出力をロジカル・レベルで行なうことができるマクロ命令を紹介することにより，通信プログラムとメッセージ処理プログラムとのインターフェイスについて以下の例を用いて説明し理解させる。

- ACCEPTマクロ命令
- CIQより1メッセージを読み出し，処理プログラムの作業エリアへ転送する。
- DISPLAY・マクロ命令
  送信メッセージを該当する端末のCOQへ登録し，その後コントロールを回線制御プログラムへ渡すことを説明する。

(8) 記憶保護制御

マルチプログラミング技法を使用する際，なくてはならない機能である。ここでは，ハードウェアの機能と関連づけて説明する。

(9) プログラム・セグメントのローディング

大規模のプログラムを考慮した場合にセグメント構造を作成し，オーバーレイの手法を取り入れる必要があることを説明する。

① 補助記憶装置上のプログラム・ファイルからプログラムを格納するプログラム・ロード機能について触れる。

② また，ロードされるプログラムのアドレスとメモリ格納アドレスとの対応についても触れる。

(10) ロギング・プログラム

ロギング・プログラムについて，その機能，特徴を簡単に説明する。

5.4 処理プログラム

オンライン・リアルタイム・システムにおけるユーザ独自の業務を処理するプログラムで使用頻度の高いものはメモリに常駐させ，低いものは必要の都度プログラム・ファイルから読み込むように作成される。またこの処理プログラムはメッセージ入出力命令を除いては特にオンラインを意識する必要はなく，一般に業務内容に群しいユーザが作成するのが普通であることを説明する。

一方，オンライン業務処理の運用上必要なプログラムとして，次のプログラムの必要性役割について簡単に紹介する。

- **システム・スタート・プログラム**
- **システム・リスタート・プログラム**
- **システム・クローズ・プログラム**
- **マンマシン・コミュニケーション・プログラム**（運用メッセージの処理）

5.5 支援プログラム

オンラインシステムにおける主役は，前述した制御プログラムと処理プログラムである。しかし，これらがうまく実行できるように援助的に作られる支援プログラムの必要性とその役割について以下の各プログラムについて説明する。

① **システム診断プログラム**

② デバグ用プログラム

- シミュレータ
- メモリ／ファイル・ダンプ・プログラム
- トレーサ

③ テストデータ・ジェネレータ

④ システム・メンテナンス・プログラム

⑤ **リカバリ・プログラム**

指導上の留意点

(1) オンライン用ソフトウェアの各コンポーネントはメーカの提供する汎用オペレーティング・システムの体系に包含される。ここでは，オペレーティング・システムの通信管理機能をデータ通信システムの必要性から把えて説明する。

(2) オンライン・システムでは，システムごとにその機能仕様や定量的な見積りを決める興味ある問題がある。

例えば，トラフィック機能，障害処理機能，各種テーブル，バッファの見積などについてオンライン・システムの設計上の問題についても触れる。

(3) 各ソフトウェア・コンポーネントの関連づけおよびデータの流れについて，明確に把握できる知識を養う。

(4) オンライン特有のマクロ命令についても整理して，モデルを設定して各機能に関連づけて説明するとよい。

第6章　オンライン・システムのファイル

用　語　この章では次の用語を教える。

キューイング・ファイル，ロギング・ファイル，チェック ポイント・ファイル，ランダマイジング，シノニム，バケット方式

目　標

オンライン・リアルタイム・システムでは，各種のファイルが使用される。また，オンライン・システムでのファイル設計は，その対象システムの良否を左右する重要なポイントになると一般にいわれている。ここでは，基本事項としてのファイルの種類，ファイル編成などの特徴を説明することによりオンライン・システムのファイルが具備すべき条件について理解させることを目標とする。

内　容

6.1　ファイルに対する基本的要求事項

オンライン・システムのファイルは特殊の性格からバッチ処理システムのファイルとは異なり，一般的に次のような基本的な要求事項が条件として考えられることを紹介する。

- 大容量性
- 高速性
- 高信頼性
- 経済性

これらの条件は互いに相反する事項であり，全て平等にこれらの要求を満足するファイル設計は困難である。そこで，システム要件からくるファイル設計のポリシーをどの条件に設計者がウェイトづけるかである。

6.2　ファイルの種類

ここでは，オンライン・システムで取扱う各ファイルの特徴およびそれらが使用される位置づけについて説明する。

(1)　プログラム・ファイル

管理プログラム，処理プログラム支援プログラム等，オンライン・システムで使用するぼう大なプログラム群が記憶され，これらはオペレーティング・システムのもとで管理される。

(2)　キューイング・ファイル

キューイング・ファイルは管理プログラムとメッセージ処理プログラムの間に介在し，メッセー

ジの処理待ち行列，送信待ち行列ファイルである。このファイルの存在により端末からインプットされるメッセージのランダム性とそれを処理するプログラムの同期がとれることを認識させる。

またこれらファイルの編成については使用するオペレーティング・システムに規定されていることに留意する。

(3) マスタ・ファイル

対象システムごとに業務処理に必要なファイルである。このファイルはユーザの責任において設計される。その際，オンライン・システムの性格から出来る限り目的のレコードが高速に索引できるファイル編成を考慮しなければならないことを認識させる。

(4) ロギング・ファイル

ロギング・ファイルは，メッセージの受信記録，送信記録をファイルしたもので，復旧処理（リカバリ）とかオンライン終了後のバッチ処理へのインプット情報として使用される点に触れる。

(5) チェックポイント・ファイル

復旧処理を短縮するために，システムの正常状態をあるタイミングで外部に吐き出しておき，復旧時に最新のチェックポイント情報をリストアし，以降をロギング・データから復旧することにより時間が短縮される。このような**チェックポイント・ファイル**について説明する。

6.3 ファイルと媒体の対応

各ファイルを具体化する際，どのような媒体を使用して実現するかが問題となる。そこで，ファイルを構成する装置として一般的には，ランダム・アクセス用の磁気ディスク，ディスク・パック，磁気ドラムがあり，シーケンシャル・アクセス用として磁気テープが考えられる。ここでは，各媒体の特徴を理解させ，それぞれの媒体に適した使い方が必要であることを認識させる。また，媒体と各ファイルを一般的観点から対応づけると次のようになろう。

- プログラム・ファイル
- キューイング・ファイル

} ……磁気ドラム（ディスク・パック）

- マスタファイル ……磁気ディスク（ディスク・パック）*

- ロギング・ファイル
- チェックポイント・ファイル

} ……磁気テープ

6.4 ファイル編成とアクセス法

オンライン・システムのファイルの編成方法には，その代表的なものとして以下のものがある。ここでは各ファイル編成構造について説明する。またお互いのファイル編成の長所，欠点についても明確にし，最終的にはファイル設計におけるファイル編成の選定基準の判断が出来る認識づけを行なう。

- 順ファイル編成
- 順索引ファイル編成

*ディスク・パックは中容量のファイルとして使用される。

表6－1　媒体の特徴

媒体	ランダム処理	容量	アクセス・タイム	価格コスト/ビット
磁気ドラム	可能	小	早い	高い
磁気ディスク	可能	非常に大	遅い	やや高い
ディスク・パック	可能	大(パック取替可能)	やや遅い	やや安い
磁気テープ	不可能	大	―	安い

・直接ファイル編成

このファイル編成で問題となるのは，レコードのキーをある種の計算法を用いて装置の物理的アドレスの範囲内にうまく変換する必要がある。この過程を**ランダマイジング**とよぶ。以下のような代表的なキー変換方法の特徴を紹介する。

・除算法

・切り捨て法

・基数変換法

・折りたたみ法

・抽出法

これらの技法を用いても，キーの特性がいろいろあり必ずしも一意的なアドレスを割当てることは不可能で，**シノニム**の発生に対して何んらかの対処をしなければならない。その考えを取り入れたのが**バケット方式**であることを説明する。

指導上の留意点

(1) この章では，ファイルの必要性および使われ方を認識させる。

(2) 特にマスタファイルに関しては，対象システムごとにユーザが最適な構造を持つファイル設計をしなければならない。その際の留意事項についても触れ，設計上の基礎知識を養うこと。

(3) ロギング・ファイル，チェックポイント・ファイルについては，リカバリ処理と関連づけて指導する。

第7章　信　頼　性

用　語　この章では，次の用語を教える。

通番，バッチカード，パスワード，排他的制御（エクスクルーシブ・コントロール），稼動率，フォールバック・システム，バックアップ・システム

目　標

信頼性の考え方は，近年いろいろなシステム作りに応用されている。特にデータ通信システムでは，システム機能の一時停止による影響は極めて大きく，関係部門のみならず企業全体ひいては社会的に影響を引き起すため，この考え方は極めて重要である。

データ通信システムでは，ハードウェア，ソフトウェアおよび人間を合んだ複雑なシステムであるので，システムの信頼性はそれらを全て含めて総合的に検討しなければならない。

そこで本章では，システムに必要な信頼性を確保するために，各種の障害に対してどのような障害の検出方法や対策を立てたらよいか，また，障害復旧後のリカバリの考え方や方法について理解させることを目標とする。

内　容

7.1　障害の種類

システムの信頼性をおびやかす原因としての障害には次のものがあることを説明する。

(1) ハードウェアの障害

① 端末装置のエラー

② 通信回線のエラー

- 回　線　断
- 雑　　音
- 機器障害

これらの障害によりメッセージの誤り，メッセージ紛失などがある。

③ 中央処理装置側の障害

- 入出力装置エラー
- メモリ・エラー

(2) ソフトウェアの障害

(3) 運用上の障害

① オペレータの操作ミス

② 原始データのエラー

(4) 過負荷による障害

① ライン・バッファなし

② キューイング・ファイルのオーバーフロー

7.2 情報に対する信頼性

データ通信システムにおいては，エラーの発生する場所，可能性は前述のごとく極めて多い。ここではインプット・データのチェック方法，メッセージ紛失防止対策について説明する。

(1) 原始データ・チェック

ここでは，端末側から入力されるデータに関する記入ミス，転記ミス，パンチミスなど，コンピュータの処理段階でチェックする方法について次の事柄を紹介する。

① 復記チェック

② チェック・ディジット・チェック

③ バリディティ・チェック

④ ハッシュ・トータル・チェック

⑤ バランス・チェック

⑥ フォーマット・チェック

⑦ リミット・チェック

(2) メッセージの紛失防止

① 通番の付加

・入 通 番

・処理通番

・出 通 番

② メッセージのロギング

・受信確認

・処理確認

・送信確認

(3) ファイルの保護

① ファイルの機密保護

・バッチカードまたはパスワードの使用

・ファイルの参照可能な端末装置の限定

・ファイルの参照可能なプログラムの限定

② ファイルの故障に対する対策

・オンライン中はリードだけにし，オンライン終了後にロギング・テープよりバッチ処理にて更新

・オンライン終了後ファイル・コピーをしておく

・ファイルの更新に対して更新前のビフォア・イメージあるいは更新後のアフタ・イメージをロギ

ングしておく。

- オンライン中に一定時間ごとあるいは一定件数処理後にファイルの内容その他必要情報を磁気テープにダンプしておく。

③ **排他的制御（エクスクルーシブ・コントロール）**

- ファイル単位
- デバイス単位
- シリンダ単位
- トラック単位
- ブロック単位

7.3 システムの信頼性

これまでは主として，システムを構成する機器が正常に動作していることを前提としたものである。しかし，どのようなハードウェアを使用しても機械は壊れるものであり，問題はそのシステムが故障によりどの程度損害が生ずるか，故障の頻度および修正にどのくらいの時間がかかるかである。ここでは，システムの信頼度を定義する用語および信頼性を考慮したシステム構成を説明する。

(1) 信　頼　度

(a) 信頼度の表示方法

システムの信頼性を表わす指標として次の用語の意味づけを行なう。

- MTBF (Mean Time Between Failure)
- MTTR (Mean Time To Repair)
- 稼働率＝MTBF／(MTBF＋MTTR)

(b) 信頼度の計算

システムの信頼度を計算する近似式を紹介し使い方を理解させる。

① 予備機を持たないシステムの場合

$$\mathrm{MTBF} \fallingdotseq \frac{1}{\frac{1}{U_1}+\frac{1}{U_2}+\cdots\cdots\cdots\cdots+\frac{1}{U_n}}$$

$$\mathrm{MTTR} \fallingdotseq \frac{\frac{D_1}{U_1}+\frac{D_2}{U_2}+\cdots\cdots\cdots\cdots+\frac{D_n}{U_n}}{\frac{1}{U_1}+\frac{1}{U_2}+\cdots\cdots\cdots\cdots+\frac{1}{U_n}}$$

- 各構成装置のMTBF，MTTRをそれぞれ U_i , D_i とする。
- $D_i \ll U_i$

$$A = a_1 \times a_2 \times \cdots\cdots \times a_n$$

A ： システムの稼動率

a_i ： 各装置の稼動率

② 予備機を持つシステムの場合

同一の装置がn 個あり，少なくともr 個稼動していれば故障でないシステム

$$MTBF \fallingdotseq \frac{(n-r)!\,(r-1)!\,U^{n-r+1}}{n!\qquad D^{n-r}}$$

$$MTTR \fallingdotseq \frac{D}{n-r-1}$$

但し　U：装置のMTBF

D：装置のMTTR

$D \ll U$

稼動率がa_1，a_2 の2台の装置を並列したときの全体の稼動率 Aは

$$A = 1 - (1 - a_1) \times (1 - a_2)$$

(2) 信頼性を考慮したシステム構成

ここでは障害に備えたシステム構成の特徴について説明する。

① **フォールバック・システム**

必要最小限の機器のみでシステムを構成し，障害に備えて予備を設けることはしない。

② **バックアップ・システム**

そのシステムの最も重要な箇所，あるいは信頼性の不足する部分を重点的に予備機を設けて補強することに触れる。

③ デュプレックス・システム

別名スタンドバイ・システムという。

④ デュアル・システム

デュプレックス・システムと対比して説明するとよい。

7.4 障 害 処 理

障害発生に対し障害原因を分析し適切な処理を行なうことにより，情報の保護およびシステムのサービスの停止を回避してその続行をはかる一連の処理を障害処理という。ここでは図7-1をもとに障害発生から復旧までの障害処理の概要を紹介する。

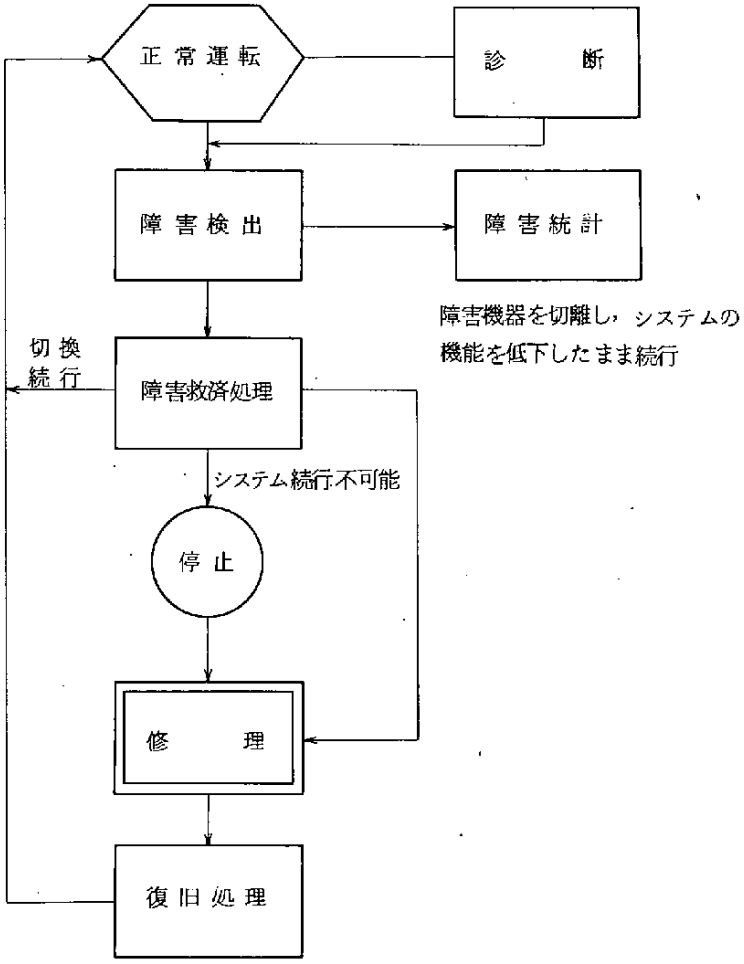

正常運転
診　断
障害検出
障害統計
障害機器を切離し，システムの
機能を低下したまま続行
切換
続行
障害救済処理
システム続行不可能
停止
修　理
復旧処理

図7－1　障害処理の概要

(1) 障害の検出

障害の種類については7.1で述べたので，ここではこれらの障害を診断プログラム等で早急に検出しなければ，誤処理やデータの紛失，ファイルの損傷などを招き，取り返しのつかないことになることを認識させた上で以下の障害検出方法について紹介する。

- ハードウェアによる検出
- ソフトウェアによる検出
- オペレータによる検出

(2) 障害の救済処理

システムの障害の検出された後の対策として，次のことを考慮すべきであろう。

- 何回かやり直してみる
- 回復出来なくても停止しない
- 停止せざるを得ない状態でも，ただ停止するのではなく，復旧処理を短時間にできるような必要なデータを吐き出したのちに停止する。

(3) 復旧処理

障害によるシステム・ダウンが生じると，再度システムを継続していくためには，システムを中断時点の元の状態に復旧（リカバリ）する必要がある。ここでは以下のような復旧方法の考え方をロギングファイルと関連づけて説明する。

① メッセージの復旧

次の各ロギング内容を突き合せることにより，システム内に滞留しているメッセージCIQ，COQへ復旧することを説明する。

表7－1　メッセージの復旧

受信確認	処理確認	送信確認	復旧範囲
○	×	×	CIQへ復旧
○	○	×	COQへ復旧
○	○	○	復旧対象が除く

② メモリ上の各種テーブルの復旧

トランザクションによって刻々変化するメモリ上の各種テーブル，カウンタの復旧方法について，次の例を用いて説明する。

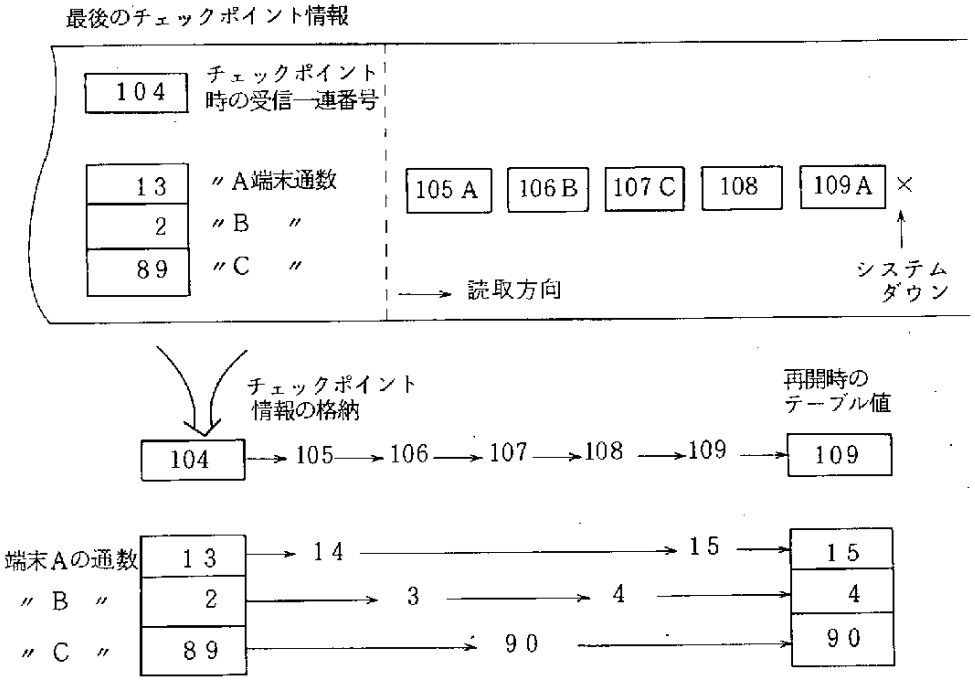


図7－2　メモリ上のテーブルの復旧

③　ファイルの復旧

オンライン・システムでは，ほとんどのシステムはファイルを保有しているが，CPUの障害が起るとファイルの内容は保証されない。そこて，ファイルを復旧するためにはファイル更新時のイメージ（ビフォア・イメージ，アフタ・イメージ）と前日のファイル・コピーされたものと突き合せてファイルを復旧するのが一般的であろう。次の例で説明する。

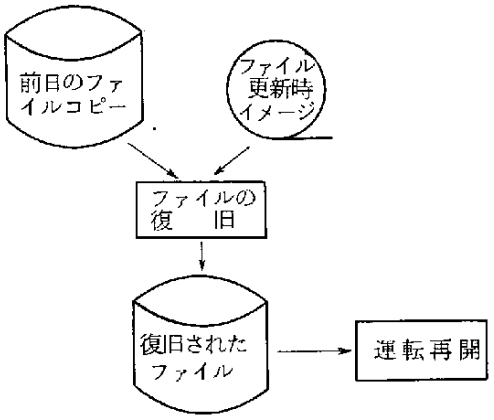


図7－3　ファイルの復旧

④ システム・リスタート

復旧処理とシステムの再開を行なうシステム・リスタートは密接な関係がある。ここではシステム・リスタートの機能の紹介を行なう。

- CCUの接続
- 送受信ラインのOPEN
- システム復旧メッセージの送信
- 各種ファイルのOPEN
- 復旧すべきコア上の各種テーブル情報のセット
- 復旧メッセージのCIQ，COQへの登録

指導上の留意点

(1) オンライン・システムにおける信頼性の必要性について身近な事例を引用して認識させた上で話を進めていくとよい。

(2) どんなに高価なシステムでもハードウェアの障害を無にすることはできない。システム作りの際はあらゆる可能性の障害を想定した対策を検討すると同時にシステム・ダウン中の手作業についての運用上の配慮も実際上はかなり重要であることにも触れる。

(3) ファイルの復旧アルゴリズムについては興味あるところであろう。適当なモデルを設定して考えさせてディスカッションするのもよい。

(4) 復旧処理は出来る限り短時間にしかも出来る限り人手の介入しない方法がベターである。これらの対策についても触れる必要があろう。

参考文献

(1) J．マーチン著，「リアルタイム・プログラミング」北原安定訳，日本経営出版会

(2) 緒方研二編，「オンライン・システム入門」企画センター

(3) 大野豊編，「オンライン・リアルタイム・システムの設計」産業図書

(4) 電気通信協会編，「データ伝送の基礎知識」オーム社

(5) 電子計算機通信学院 「データ通信の基礎知識」

(6) 上野著，「データ通信システム入門」日刊工業新聞社

(7) 日本電気情報処理教育部編，「データ通信システム入門」日本能率協会

(8) 植田義明著，「データ通信のためのデータ通信利用マニアル」企画センター

(9) 新田謙治郎，高山 由著，「タイムシェアリング・システム」 富士学院

(10) コンピュータ・マネジメント社編 「リアルタイム・システム設計」

ハードウェア (II)

目　次

第1章　処理方式 …… 364

1.1　多重プログラミング …… 364

1.2　アドレッシング …… 370

1.3　多重プロセッサ・システム …… 376

1.4　ハードウェアとソフトウェアの機能分担 …… 377

第2章　性能評価 …… 379

2.1　単体アドレス …… 379

2.2　ハードウェア・システムの性能 …… 388

第3章　信頼性 …… 388

3.1　信頼性 …… 388

3.2　可用性 …… 390

第4章　互換性 …… 392

第5章　実　習 …… 395

科目「ハードウェア（II）」

教育の目標

ハードウェアそのものに関する知識を習得させるというよりは，ハードウェアを利用して具体的なコンピュータ・システムのシステム分析・設計，ソフトウェア作成等を実施する際必要となるハードウェア諸方式を深く広く修得させ，コンピュータ・システムに対する理解力，応用力，判断力を養わせる。

時間配分

章	履習時間（時間）		
	講義	演習	実習
1. 処理方式	12		
2. 性能評価	20		
3. 信頼性	5		
4. 互換性	3		
5. 実習			20
合計	40		20

第1章　処理方式

用　語　この章では次の用語を教える。

割込制御処理，特権命令，レジスタ方式記憶保護，ページ方式記憶保護，セグメント方式記憶保護，リング方式記憶保護，ダイナミック・リロケーション，仮想記憶，アドレス空間，ストーレッジ空間，ページング，オーバーヘッド，セグメンテーション，2次元アドレス，メモリシェア，ロードシェア

目　標

コンピュータの種々の処理方式のうち，特にシステム設計，プログラミング等において理解しておく必要のある項目である多重プログラミング，アドレッシング，多重プロセッサ，ハードウェアとソフトウェアの機能分担につき説明を行ない，ソフトウェア及びオペレーティング・システムの説明と共にコンピュータを総合的に理解させる。

内　容

1.1　多重プログラミング

多重プログラミングについては既に基礎編「ハードウェアⅠ」第1章システム構成で概略を説明したが，ここでは更にその目的，必要なハードウェア機能について深く理解させる。

(1) 多重プログラミングの目的

多重プログラミングを実施する目的を次のように説明し理解させる。

(a) 周辺装置の処理速度は一般的に中央処理装置の処理速度に比較して格段に遅い。

例えば毎分1,000枚処理するカード読取装置を例にとるとカード1枚当り 60ms の処理時間要するが，中央処理装置はこのカード上の情報を演算するのに普通数ミリ秒しか要さない。従って 60ms のうち50ms 以上は中央処理装置は他の処理を実行する余力がある。

(b) 普通1個のプログラムでも複数個の周辺装置を同時に動作させ中央処理装置を有効に利用する方法がとられている。

例えばカード読取装置で読取ったカード情報により，磁気テープ装置のマスタファイルを更新する一方，更新情報をラインプリンタに印字させる等はよく行なわれている。

しかし1個のプログラムでは，ある処理が完了しなければ次の処理へ移れない場合がよくある。このようなときに複数個のプログラムを切換えながら処理すると効率よく中央処理装置を利用出来る。

(c) このように周辺装置の処理速度と中央処理装置の処理速度の差を利用して複数個のプログラムを併行処理することを多重プログラミングと称している。

(d) 従って多重プログラミングはプログラム処理が周辺装置の処理速度によって制限されているような

場合に効果的であって，例えば中央処理装置の内部演算処理のみを実行するようなプログラムにおいては多重プログラミングの効果は期待出来ないことをよく理解させるべきである。

(e) 多重プログラミングにより期待出来る効果は更に次のようなことがあることを理解させる。即ち，システムの資源（中央処理装置，主記憶，チャネル装置，周辺装置）をすべて管理プログラムにより集中管理し，全資源を最も有効に動作させることにより装置の遊びを少なくし，また複数プログラムによる装置の取りあいを分散させることが出来る。

例えば，あるプログラムが磁気ディスク装置とラインプリンタを対象として処理を実施している間，カード読取装置と磁気テープ装置を使用してデータ変換を実施するプログラムを実行させたり，また2個のプログラムで同一ラインプリンタへの出力要求がある場合は，一方の処理結果を一時的に磁気ディスク装置へ出力し処理を進め，他方のプログラムでラインプリンタが不必要になった時点で磁気ディスク装置から処理結果をラインプリンタへ出力するなど，プログラムを一本づつ逐次処理する場合よりシステム資源を有効に利用することが可能となる。

(2) 多重プログラミングに必要なハードウェア機能

多重プログラミングという場合には一般に全く関連のない2個以上のプログラムを併行処理することをいうが，このため多重プログラミングを実施する場合には，ハードウェアとして特殊な機能を備えている必要があり，またそれを管理制御する管理プログラムが必要となることを説明し，ここでは特にそのためのハードウェア機能の説明を行ない理解させる。尚，多重プログラミングは管理プログラムと密接な関連があり，ハードウェア機能の説明のみでは理解しにくいので，ある程度管理プログラムの機能についても関連させ説明した方がよい。但し管理プログラムの詳細説明は別カリキュラムで実施することとする。

多重プログラミングで必要となるハードウェア機能は次のものがあることを説明し，各々動作概念を明確に理解させる。

(a) 割込機能

割込機能については，その概念を明確に理解することが重要であるので，先ず割込の概念につき相当時間をかけ理解させる。

① 割込の概念

初期の段階では主に周辺装置に関連した割込が考えられ，これは必ずしも多重プログラミングと関連したものでもなかった。即ち，1個のプログラムで例えばカード読取装置より情報を読取りその結果磁気テープ上のマスタファイルを更新し，一方更新情報をラインプリンタに印字するといった処理において，プログラムで各周辺装置の動作状態をチェックすることは非常に非能率であり，逆に周辺装置の方から動作終了時点に中央処理装置に対して特別な信号（割込信号）を発生させ，この信号により中央処理装置は実行中のプログラムを一時中断し，割込を発生した周辺装置の処理に移る方式が考えられた。

後にこの割込は単に周辺装置のみならず，種々の分野に拡張されてきており，特定の条件（例えば周辺装置の動作終了など）をハードウェアで検出し，割込信号を発生させることにより中央処理装置がその時実行中のプログラムを中断し別のプログラム処理に移ることを割込（プログラム割込）

と呼んでいる。

② 割込の種類

最近のコンピュータでは割込を発生する原因としては数十種類以上持つものが多く，またコンピュータによって特徴がある。従って，割込の種類を次のように機能的に大別し，各々例を二，三あげて説明する。

- 周辺装置などの外部要求によるもの

  例） 周辺装置の動作終了，操作卓の割込ボタン等

- プログラムによるもの

  例） スーパバイザ・コール命令，不正命令，オーバーフロー等

- ハードウェア障害によるもの。

  例） 主記憶のパリティエラー，電源異常等

③ 割込制御処理

割込が発生し，それが中央処理装置に認識され処理される過程，即ち**割込制御処理**を次の手順でハードウェア，ソフトウェア機能に関連させながら説明する。

- 割込の発生
- 割込の認識及び受付け
- 管理プログラ・モードへの切換え
- レジスタ，インディケータの状態退避
- 割込原因の判別
- 処理プログラムの選別，準備
- 処理プログラムの実行

上記説明におけるハードウェア機能，ソフトウェア機能はコンピュータにより分担の度合が異なることを理解させる。

また上記説明の中で，次のことを説明する。

- 割込モードの概念，必要性
- レジスタ，インディケータの状態退避の必要性
- 割込受付の優先順位と処理の優先順位

(b) 記憶保護機能

記憶保護機能は多重プログラミングの概念と共に生まれたものであり，非常に密接な関連がある。ここでは記憶保護機能の概念と種々の方式につき次のように説明し理解させる。

① 記憶保護の概念

多重プログラミングにおいては，多数のプログラムを併行処理するが，このプログラムの中には管理プログラムや互いに無関係な処理プログラムが含まれている。更に，その中にはデバグ中の未完成なプログラムも含むことがある。多重プログラム実行中に，例えばこのデバグ中のプログラムがプログラム・ミスにより，管理プログラムや他の処理プログラムを破壊したならば，その処理結果は保証されなくなる。このような危険性があったとしたらそのシステムは全く信頼性を失う。

従って多重プログラムにおいては，処理されるプログラムの処理状態を常に厳重に監視しなければならない。このため機能として記憶保護機能がある。

この概念を具体的に理解させるために比較的理解の容易な主記憶の書込保護の例をあげて説明する。（図1－1参照）

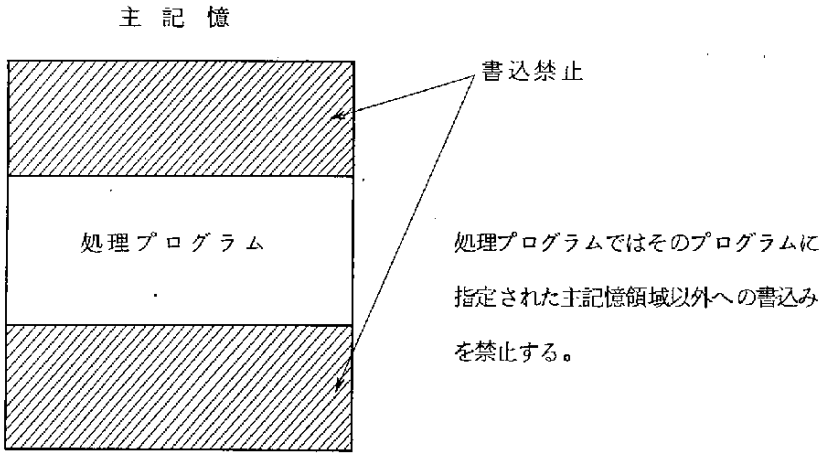


図1－1　主記憶の書込保護

② 記憶保護の方式

記憶保護の方式としては大きく分けてモード制御とアクセス制御があることを説明し，これが組み合わされて記憶保護が実施されることを理解させる。尚記憶保護という言葉より主記憶の書込保護のみにとらわれがちであるが，最近ではそれだけではなくもっと広義に使われていることを教える。

・モード制御

割込モード（特権モード，マスタモード等ともいわれる）と非割込モード（非特権モード，スレーブモード等ともいわれる）によりモード制御を行なっている例が多い。

割込モードにおいては管理プログラムが動作し，ハードウェアの全機能を使用出来るが，非割込モードは，一般の処理プログラムが動作し，このモードにおいては一部命令（特権命令等といっている……具体的な命令をあげて特権の意味を説明する）の使用が制限され，また主記憶の使用可能領域も制約される。この制約はアクセス制御により実施される。

モードにおける制約はハードウェアにより監視されるが，モードの切換えはハードウェアで実施するものもあれば，ソフトウェア（管理プログラム）の設定によるものもある。

③ アクセス制御

アクセス制御としては，主記憶への書込制御，読出し制御及び命令実行制御があることを説明する。

・書込制御

指定された主記憶領域外への書込禁止

・読出制御

指定された主記憶領域外からの読出禁止

・命令実行制御

指定された主記憶領域外の命令の実行禁止

④ 主記憶の領域指定

主記憶の領域指定には**レジスタ方式（記憶保護）**，**ページ方式（記憶保護）**及び**セグメント方式（記憶保護）**がある。

・レジスタ方式

本方式について下図の例により説明する。尚本方式は後記1.2アドレッシングの(2)アドレス変換の ① レジスタ方式と密接な関連を持つことを説明する。

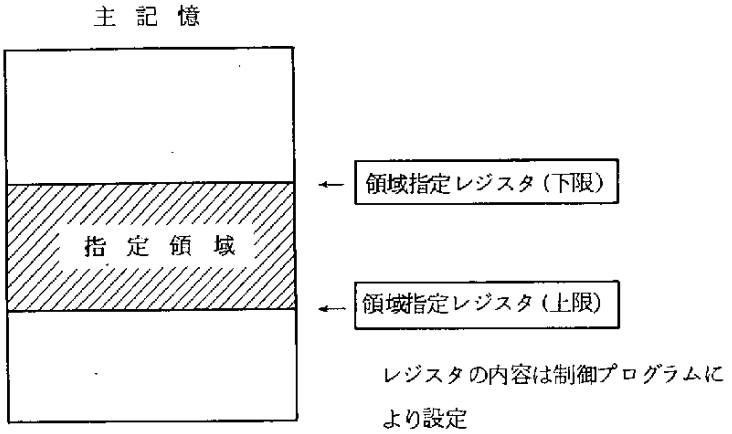


図1－2　レジスタ方式

・ページ方式

本方式について下図の例により説明する。尚本方式は後記1.2アドレッシングの(3)仮想記憶の(b)ページングと密接な関連を持つことを説明する。

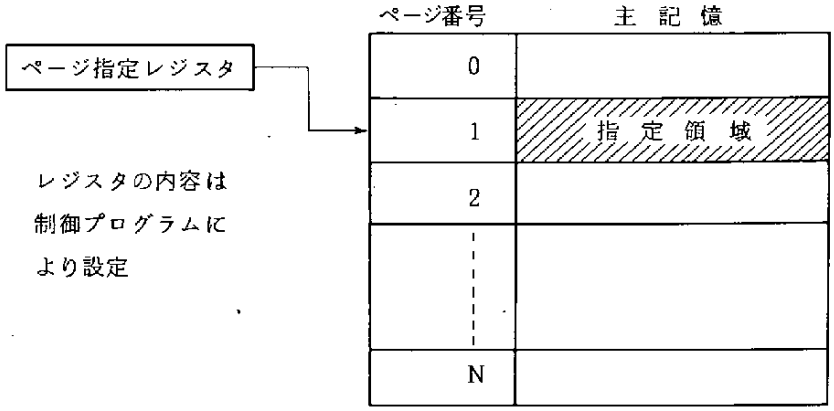


図１－３　ページ方式

・セグメント方式

セグメント単位にキーを持ち，各モードでのキー番号と比較する。キーはセグメントの種類，書込禁止，読出禁止，命令実行禁止などを含む。

尚本方式は後記 1.2 アドレッシングの(3)仮想記憶の(c)セグメンテーションと密接な関連を持つことを説明する。

⑤　リング式記憶保護

記憶保護はモード制御とアクセス制御の組み合わせたものとして実施されるが，更にこれらを各種レベルに細分化した**リング式記憶保護**についても説明する。

例えば書込制御を可能・不可能の２種だけでなく，次図の如き概念でレベル分けし，記憶保護を精密化している。これは読出制御，命令実行制御などにも適用される。

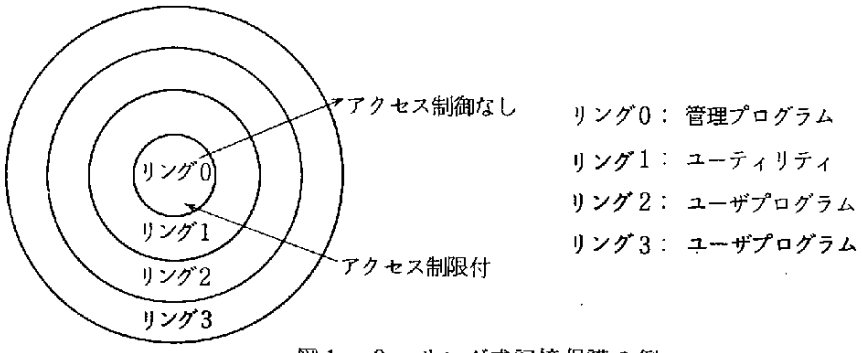


図１－３　リング式記憶保護の例

(c) プログラムの再配置機能

多重プログラミングを実行する場合にプログラムの再配置（リロケーション）の必要性につき説明し，種々の方式について理解させる。

① プログラムの再配置機能の必要性について次の点をよく理解させる。

- 主記憶には複数個のプログラムがロードされ，併行処理される。
- 処理の終了したプログラムのあった主記憶は次の処理するプログラムに利用されるが，その領域はそのプログラムをロードする時点迄予測出来ないことが多い。
- 実行中のプログラムでも，より処理の緊急度の高いプログラムを処理するために一時中断する必要が生ずることがあり，このとき中断されたプログラムは，一時補助記憶装置等に退避され，後に再び主記憶にロードされ処理が続行される。このときロードされる主記憶領域は中断前と異なる場合がある。

このように多重プログラミングにおいては，プログラムは主記憶のどの領域にロードされても実行可能である必要がある。この機能をリロケーション（relocation）と呼んでいる。

リロケーションには更に静的リロケーションと動的リロケーションがあることも簡単に説明し，詳細は次節で説明を行なう。

- 静的リロケーション……プログラムを主記憶にロードするときにリロケーション可能
- 動的リロケーション（**ダイナミック・リロケーション**）……プログラム実行中でもリロケーション可能

② プログラム配置の方式には，現在レジスタ方式とマッピング方式の2種があることを説明し，詳細は次の1.2アドレッシングの所で説明する。

1.2 アドレッシング

一般に中規模以下のコンピュータではプログラムが対象とするアドレスは主記憶装置に付されたアドレスを取り扱うものが多い。

即ち主記憶装置に付された0番地より主記憶容量迄の連続アドレスがそのままプログラムの対象とするアドレスと一致するのが一般的である。

しかし最近の中規模以上のコンピュータではプログラムで取り扱うアドレスは必ずしも主記憶装置のアドレスと一致しない場合が増えている。

この理由としては一つには多重プログラミング等における主記憶装置の有効利用と仮想記憶の概念によるものがあげられる。

これらに関して以下のように説明を行なう。

(1) 論理アドレスと物理アドレス

プログラムで取り扱われるアドレスを論理アドレスと呼び，主記憶装置にハードウェア的に付されたアドレスを物理アドレスと呼んでいるが，これらに関して以下のように説明し，理解させる。

(a) 論理アドレス

例えばプログラムにおける 加算命令ADD ／A，B （ A＋B →B ）のAアドレス及びBアドレスで示されるアドレスは論理アドレスであり，一般にはこれらは主記憶装置に付された物理アドレスと一致している。しかしコンピュータによっては論理アドレスと物理アドレスは固定された関係はなくその対応が変化するものもある。この対応はアドレス変換により実施される。

(b) 物理アドレス

ハードウェアで固定されソフトウェアでは変更できないアドレスを物理アドレスと呼んでいる。

物理アドレスは演算処理装置より見た場合のアドレスであって，実際に主記憶装置が一度に読出し書込みを行なえる単位とは独立していることも理解させる。

例えば物理アドレスはバイト単位であるが，主記憶装置の読出し書込みは4バイト単位で行なうなど，処理速度と経済性により種々方式がある。

(2) アドレス変換

論理アドレスと物理アドレスが独立しているコンピュータでの両アドレスの対応を行なうアドレス変換機構について説明を行なう。

現在主として用いられている次の二つの方法につき概略説明し，理解させる。

(a) レジスタ方式

図1－4のように，論理アドレスの開始アドレスを特定のレジスタ（リロケーション・レジスタなど）で物理アドレスの特定アドレスに設定する方式。この方式は，主としてプログラムの再配置の必要性により生まれた手法であり，比較的簡単なハードウェアで実施出来るので，よく使用されている方式である。

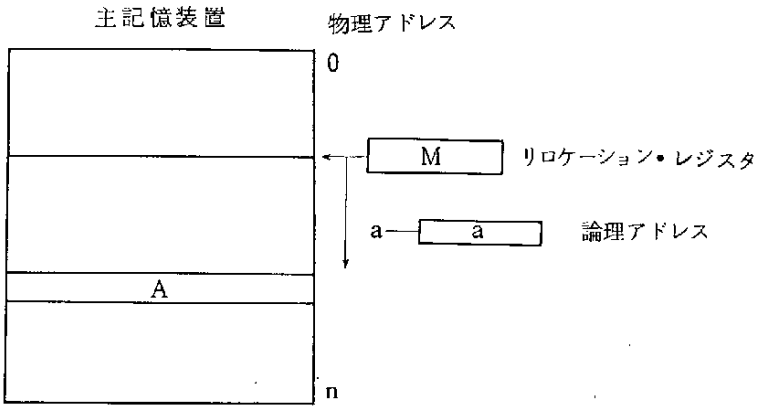


図1－4　レジスタ方式

(b) マッピング方式

論理アドレスと物理アドレスの変換テーブルを持ち，このテーブルにより対応づけする方式。変換テーブルは主記憶装置内に持つが，その内の一部は高速連想記憶装置により高速変換を実施するのが普通である。

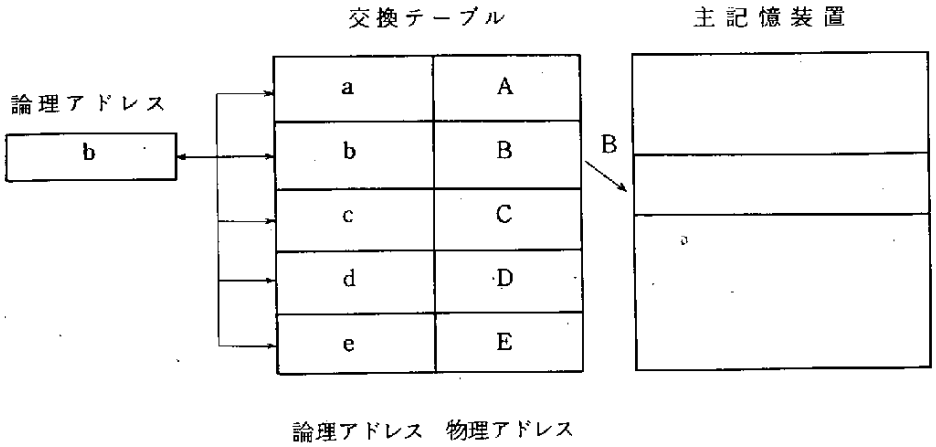


図1－5　マッピング方式

この方式はプログラムの再配置及び主記憶の効率的利用のために考えられたものである。レジスタ方式より融通性が高いが，ハードウェアが複雑てなり，またアドレス変換に時間がかかることもある。

(3) 仮想記憶

コンピュータの実際の主記憶容量（物理アドレス）以上の論理アドレスをプログラムで取り扱える方式を**仮想記憶**と呼んでいる。

ここでは仮想記憶の原理，方式，特徴につき説明を行ない理解させる。

(a) アドレス空間とストーレッジ空間

プログラムが対象とする論理アドレスの大きさを**アドレス空間**と呼び，実際の主記憶装置の持つ物理アドレスの大きさを**ストーレッジ空間**と呼んでいる。

仮想記憶の場合，アドレス空間はストーレッジ空間よりはるかに大きく，このため主記憶装置のバッキング・ストア（Backing Store）として補助記憶装置が使用される。

仮想記憶の説明に当って一般的に限定された主記憶容量以上のメモリを必要とするプログラムを処理する方法として次の方法を説明する。

- LCS（Large Core Storage）の使用
- プログラム・セグメントのオーバーレイ
- ロールイン／ロールアウト

(b) ページング

ページングとは小さな主記憶容量で大きなアドレス空間を実現する仮想記憶方式の一手段であり，その原形は1960年頃に英国のマンチェスタ大学で開発されたATLASというコンピュータに採用されている。

ここではページングの原理，概念，利害得失につき次のように説明を行なう。

① ページングの原理

図1－6，1－7の例をあげて説明を行なう。

- 論理アドレスと物理アドレスの変換をマッピング方式で実施
- 使用するページの主記憶装置と補助記憶装置間の出入制御のための制御プログラムが必要
- プレページングとデマンド・ページングの説明

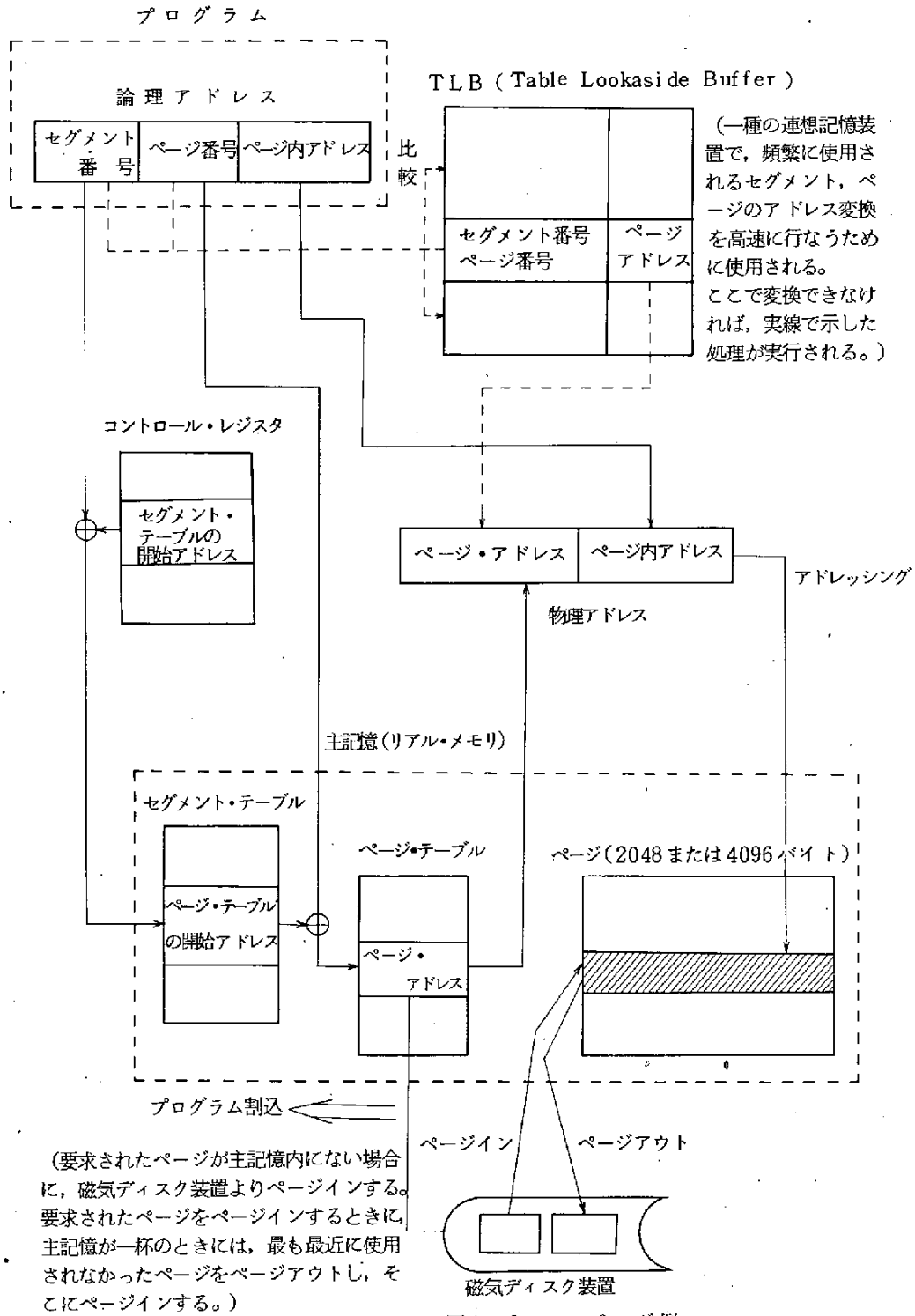

プログラム
論理アドレス
セグメント番号
ページ番号
ページ内アドレス
比較
TLB（Table Lookaside Buffer）
セグメント番号
ページ番号
ページアドレス
（一種の連想記憶装置で，頻繁に使用されるセグメント，ページのアドレス変換を高速に行なうために使用される。ここで変換できなければ，実線で示した処理が実行される。）
コントロール・レジスタ
セグメント・テーブルの開始アドレス
ページ・アドレス
ページ内アドレス
アドレッシング
物理アドレス
主記憶（リアル・メモリ）
セグメント・テーブル
ページ・テーブルの開始アドレス
ページ・テーブル
ページ・アドレス
ページ（2048または4096バイト）
プログラム割込
ページイン
ページアウト
（要求されたページが主記憶内にない場合に，磁気ディスク装置よりページインする。要求されたページをページインするときに，主記憶が一杯のときには，最も最近に使用されなかったページをページアウトし，そこにページインする。）
磁気ディスク装置

図1－6 ページング例

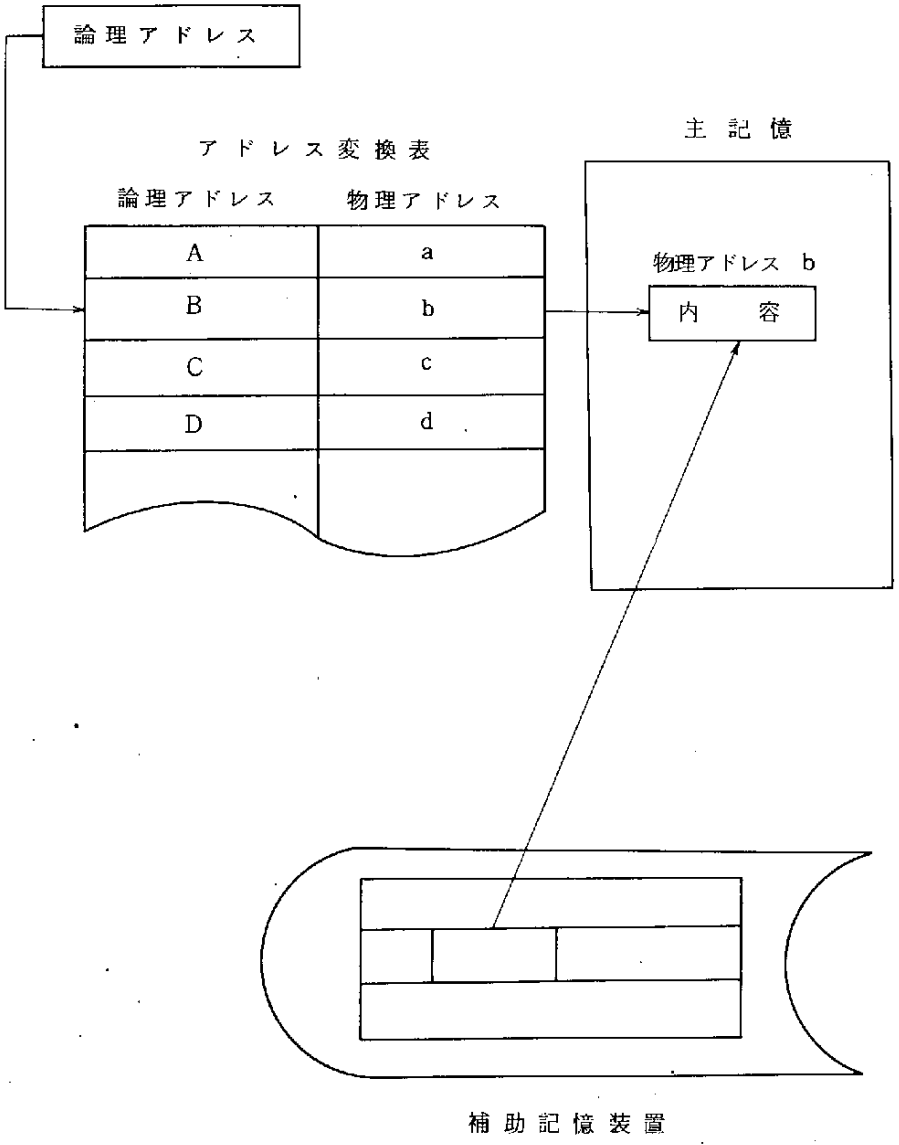


図1－7　アドレス変換

② ページングの利害得失

・利　点

仮想記憶により小さな主記憶容量で大きなプログラムを処理可能

主記憶の有効利用が可能

・欠　点

オーバーヘッドが増大する。

管理プログラム及びハードウェアが複雑となる。

(c) セグメンテーション

セグメンテーションとは，1個のプログラムを幾つかの単位（セグメント）に分割し，セグメント毎に補助記憶装置より主記憶装置にロールイン／ロールアウト処理するものであり，ページングが物理的単位であったのに対してセグメンテーションは機能的にまとまったセグメント単位で実施される。

セグメンテーションは一般のプログラムではソフトウェアでよく行なわれる手法であるが，ここでは特にセグメント単位でアドレスする機能とセグメント単位で記憶保護を行なうハードウェアを持つものをいうことを理解させる。

① セグメンテーションの原理

次図の例をあげて説明する。

論理アドレス

セグメント番号	セグメント内アドレス

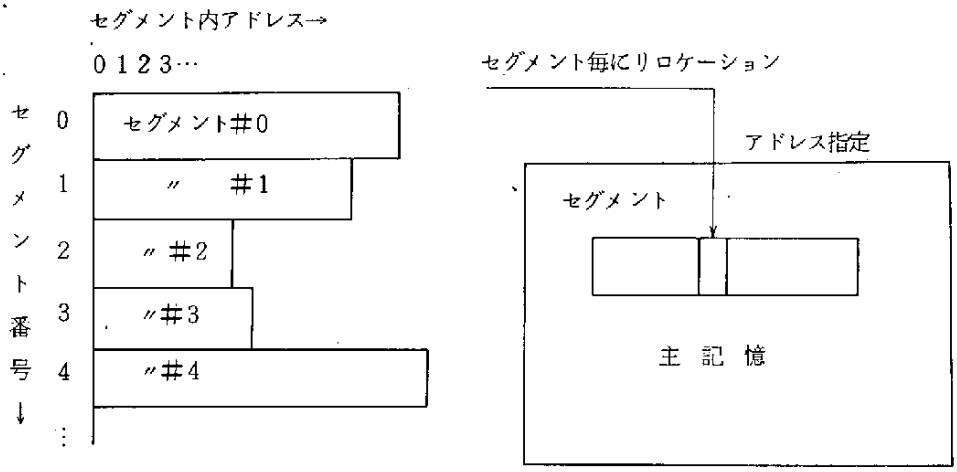


図1－8 セグメンテーションによるアドレス指定

- **2次元アドレス**
- セグメント内アドレスの大きさはストーレッジ空間を越えられない。
- セグメンテーションとページングは概念の異なるものであり，これらを組み合わせて利用するとより効果的である。

② セグメンテーションの利害得失

- 利　点

セグメントの独立性によりプログラミングの融通性，生産性の向上，プログラムやデータの共有が簡単，セグメント同志の動合が可能

セグメント単位でのきめ細かい記憶保護が可能

・欠　点

ロールイン／ロールアウトがセグメント単位で実施されるので，そのセグメント以上の容量の空きメモリがないとロールインが出来ない。

セグメント分割およびロールイン／ロールアウトのアルゴリズム構成によっては頻繁にロールイン／ロールアウト実行が必要となり処理効率が低下することがある。

1.3 多重プロセッサ・システム

複数個の中央処理装置を直接接続し，一つのシステムとして動作させるシステムを多重プロセッサ・システムと呼んでいる。

ここでは多重プロセッサ・システムの目的，方式，ハードウェア機能等につき説明し，特にオンライン・リアルタイム・システム等で多く利用されるこのシステム方式について知識を修得させる。

(1) 多重プロセッサ・システムの目的

次のような目的があり，用途に応じて使い分けられる。

(a) 信頼性の向上

複数台の中央処理装置を用いることにより，その一部が障害を起こしても他の装置により処理を続行出来る。

(b) 処理速度の向上

複数台の中央処理装置で処理を分担することによって高速化が図れる。

この方式はまた中央処理装置を各々分担処理業務によって特殊化することによりシステムの価格／性能比を改善することが出来る。例えばこの典型的な例としてミニコンピュータ等を使用した通信処理装置などをあげて説明する。

(2) 方　式

多重プロセッサ・システムの方式を大きく分けると，中央処理装置の処理が一般的なものか特殊化されているか及び主記憶を共有するか否かの二つの方式があり，これらが組み合され，また中央処理装置の処理分担の度合いなどにより様々なシステムが構成される。

この代表的な例を次のようにあげて説明を行なう。

(a) デュアル・システム（Dual System）とデュプレックス・システム（Duplex System）

同一処理を実行する2台の中央処理装置で構成され，各々の処理効果を照合し，一致を確認しながら処理を進めるシステムがデュアル・システムで，主に信頼性の向上が目的であり，主記憶は各々独立して持つ。

デュプレックス・システムは，通常は2台の中央処理装置は独立した処理を実行しているが，一方が障害を起こし，その処理が重要なものの時には他方の実行中の処理を中断し，障害を生じた側の処理をき受け続行するシステム。

デュアル・システムは信頼性は高いが，冗長度が多くデュプレックス・システムは信頼性は多少低くなるが，処理能力の方に重点を置いたシステムといえよう。

(b) 主記憶を共有する方式

複数台の中央処理装置が主記憶及び周辺装置を共有するシステム。

先ず一つのオペレーティング・システムのもとに中央処理装置，周辺装置の処理分担がダイナミックに割り当てられ，**メモリシェア**，**ロードシェア**がシステム全体を通してダイナミックに実施される方式について説明を行ない，次に各中央処理装置毎にオペレーティング・システムを持ち各々その制御のもとに動作するシステム，各中央処理装置の処理分担が例えば演算処理用と周辺装置処理用などと定められているシステムなどを説明する。

(c) 特殊処理用中央処理装置を使用する方式

例えば通信回路及び端末装置の回線制御と誤り処理などのみ行なう中央処理装置，周辺装置の制御処理のみを行なうもの，アレイ•プロセッサなど特殊演算を高速に行なうものなど処理の特殊化された中央処理装置を使用した多重プロセッサがあることも説明する。

(3) ハードウェア機能

多重プロセッサ方式のためには，中央処理装置，周辺装置は通常の機能に加えて，そのための種々の機能を備えている必要がある。これらの機能は各コンピュータにより種々考えられ，またソフトウェアとの関連も非常に密接となるので，ここでは次のような一般的な多重プロセッサ方式のためのハードウェア機能の説明にとどめておく。

① モジュール化

中央処理装置，主記憶装置，周辺装置はシステムで共通に使用されるためにモジュール構造をとり，任意に接続・切り離し出来ることが必要

② 情報交信

中央処理装置間で互いの状態，処理状態，処理分担などの交信を行なえる機能が必要

1.4 ハードウェアとソフトウェアの機能分担

プログラム内蔵式コンピュータでは，本質的には1個の命令「Subtract and Store」さえ備えておれば，論理的に全ての動作を行なうプログラムを実行出来る。

しかし一般のコンピュータが数十から数百もの命令を備えているのは，これでは到底演算速度や主記憶容量の点で実用に供さないためであり，この点からコンピュータの動作はハードウェアとソフトウェアの適切な機能分担が必要であり，またこれは時代変化を受けることを説明する。

理解を容易にするために，先ずハードウェア命令とソフトウェア・サブルーチンの関係について，編集命令，符号変換命令，索表命令などの命令が考えられた経緯を説明し，これもハードウェアとソフトウェアの機能分担の一例であることを説明する。

一般に機能分担をハードウェアかソフトウェアかに定める場合には次の点が考慮され決められる。

① ハードウェアで実施する場合

- 処理速度が速い。
- 主記憶容量の低下がはかれる。

② ソフトウェアで実施する場合

・機能の向上，汎用化が計れる。

・機能追加・変更の融通性が高い。

しかし最近では回路部品の低価格化，小型化が進み，またマイクロプログラム技法の進歩もあって，コンピュータのより高速化，高機能化のためにハードウェアの機能分担の比率が高まって来ていることを説明する。またマイクロプログラム技法によりハードウェアのソフトウェア化も計られ，いわゆるファームウェアの概念があることも説明する。

最近の機能分担の例を次のようにあげ，概略を理解させる。

① 特殊演算処理装置

ベクトル処理，マトリクス処理などをハードウェアで実行

② 周辺制御装置

各種周辺装置の動作制御をマイクロプログラムで実施することによりきめ細かい制御が実現。

また，今後の発展方向としてはハードウェア部品の進歩とマイクロプログラム技術の発展により次のようなことが考えられることを説明する。

① 中央処理装置

・演算制御機能のモジュール化，特殊化

・マイクロプログラム方式の拡大…経済性，柔軟性

・割込制御処理のファームウェア化

・ハードウェア機能の拡張による信頼性，稼動性，機能向上（RAS…Reliability, Availability, Serviceability）

② 周辺装置

・チャネル装置の機能拡張

チャネル・スケジューリングの自動化，チャネルの高速化，多数チャネルの制御，エラー処理の自動化などRASの拡張

・周辺制御装置の機能拡張

例えば磁気ディスク制御装置のシーク，サーチ，キューなどの自動処理，エラー処理の自動化などRASの拡張

指導上の留意点

本章は言わば基礎編「ハードウェア(I)」の応用編に当るものであり，従って「ハードウェア(I)」の内容理解が前提となっている。

このため「ハードウェア(I)」の理解を高めるために復習をさせたり，補足説明を適時行ないながら説明するとよい。

また，出来るだけ深く幅広く理解させるために代表的コンピュータを多くあげその方式，特徴等につき具体的に説明を行なうことが望ましい。

第2章　性　能　評　価

用　語　この章では次の用語を教える。

カタログ性能，実効記憶容量，実効データ転送速度

目　標

コンピュータの処理能力について実効性能の算出法を中央処理装置及び周辺装置の単体毎に説明し，またそれらが組み合わされたシステムとしての性能について説明するとともに，カタログ性能と実際の実効性能に相異があることを理解させる。

コンピュータのハードウェア性能を評価する場合，どういう観点で考慮すればよいかを説明し，評価能力を養わせる。

コンピュータの性能はハードウェア性能と共にソフトウェア及びアプリケーションの性能にも大きく左右され，これらの総合評価により判定されるものであるが，ここではハードウェアについて評価力を養い他の科目と併せて総合評価力を養わせる。

内　容

2.1 単　体　性　能

ハードウェア単体性能について，代表的な次の装置をとりおげてその性能評価法を説明し理解させる。

なお，これらの装置に関するいわゆる**カタログ性能**については，ＪＩＳ－Ｃ－6272電子計算組織構成機器の性能表示として制定されている項目について内容を説明し，次に各装置について具体的な機器を二，三例にとり，カタログ性能を表示し，実際にシステムとして使用されるときの実効性能を説明対比させてカタログ性能の受取り方及び実際の性能について理解させる。

特に下記の装置については詳細に説明を行なう。

- 中央処理装置
- 入出力機器制御装置
- 磁気ドラム装置
- 磁気ディスク装置
- 磁気テープ装置
- カード読取装置
- カードせん孔装置
- 紙テープ読取装置
- 紙テープせん孔装置

• ラインプリンタ

なお，これらの装置に関して互換性，標準化，拡張性，設置性，運用性，信頼性，保守性等については最初に一般的に説明し，装置特有のものに関しては各々の所で説明を行なうようにする。

(1) 中央処理装置

中央処理装置については特に，処理速度としてとらえられる性能評価だけでなく種々有する機能，例えば割込機能や記憶保護機能など機能評価も非常に重要な要素であり，これらを併せて検討する必要がある。これらはコンピュータの利用目的により一方を重要視して考えねばならない場合もあるが，一般的には両者のバランスを考慮し評価することが多い。

性能評価及び機能評価に関しては何れも一義的に行なうことは非常に難かしく，利用目的を明確にした上で行なう必要がある。しかしそれにしても容易なことではなく，ある程度経験的判断によらざるを得ないことをよく理解させる必要がある。

先ず中央処理装置のカタログ性能に関してＪＩＳ－Ｃ－6272に従い各項目の意味などを説明し理解させる。

次に各項目に関して実際のコンピュータを評価する場合の注意する点，考え方などにつき次の点を注意すべきことを説明する。このとき実際のコンピュータを二，三例にとり説明を行なうようにする。

① データ形式について

事務処理用途の場合には以下の点に注意する。

• 可変長データ（英字，数字，記号，カナ文字）の取扱い及び１字のビット数

• 10進数の取扱い

科学演算処理用途の場合には以下の点に注意する。

• 固定長データの取扱い及びそのデータ長

• 浮動小数点データの取扱い

② 命令形式について

事務処理用途の場合には以下の点に注意する。

• ２アドレス（メモリ対メモリ演算）または３アドレス方式の可否。２アドレス方式の重要性を１アドレスで実行させたときのプログラム•ステップ数，演算時間と対比させて説明するとよい。

• アドレス修飾方式

インデックス修飾，間接アドレスなどの種類，レジスタの数修飾のレベル，速度

• 命令の種類

編集命令，コード変換命令，索表命令などの有無。これらの命令の必要性を認識させ，サブルーチンで実施させたときのプログラム•ステップ数，演算時間を対比させて説明するとよい。

• レジスタの種類，数，速度

• 演算時間

個々の命令の種々の条件（データ長など）における演算時間と実際にプログラムで使用する際の命令の組み合わせによる時間。このとき10ステップ程度のプログラム例をあげ，それを実行す

るのに必要な時間を算出させて見るとよい。

同一命令でも種々の条件により演算時間が大幅に異なるときが多いことを理解させ，従ってコンピュータの平均命令実行時間の算出法は極めて困難であり，キブソンミックス値なども一つの目安として見ることが重要であることを説明する。

科学演算処理用途の場合には以下の点に注意する。

- 1アドレス（レジスタ対メモリ及びレジスタ対レジスタ演算）方式の可否
- 命令の種類

2進固定小数点，演算命令，浮動小数点演算命令，論理演算命令（AND，OR，NOT 等），ビット処理命令などの有無。

- アドレス修飾，間接アドレスなどの種類，レジスタの数，修飾のレベル，速度
- 演算時間

個々の命令の種々の条件（データ長など）における演算時間と実際にプログラムで使用する際の命令の組み合わせによる時間。

③ チャネルについて

特に事務処理用途では多くの周辺装置を動作させる処理が多く，チャネルの機能，性能は処理速度に大きな影響を与える。

- チャネルの種類

セレクタ，マルチプレクサ，ブロック•マルチプレクサなど，用途に適しているか否か

- チャネルの数と接続周辺装置の数及び種類

必要とする周辺装置に見合うチャネル及びサブチャネルの数及び種類が確保されるか否か

- データ転送速度と同時動作

必要とする周辺装置に見合うデータ転送速度を有するか否か。またチャネル（周辺装置）の同時動作はシステムの要求を満たすか否か。

チャネルの動作により，中央処理装置の演算速度に与える影響の大小

④ 主記憶装置について

- 記憶容量

必要とする容量があるか否か，またプログラムの変更等による記憶容量の増減が容易に経済的に可能か否か

- サイクル•タイム及びアクセス•タイム

サイクル•タイム及びアクセス•タイムは充分速いか，この速度は中央処理装置の処理速度に絶対的な影響を与えるとはいい切れないが，重要な要素の一つであることは疑いない。

1回の読出し書込みで取扱われるデータの単位の大きさは適当なものか否か。

⑤ その他の機能

割込機能，記憶保護機能，多重プログラミング機能，誤り制御方式などその他特徴的機能

これらはコンピュータにより種々特徴があるので具体的なコンピュータを出来るだけ多くあげ，それらが持つ機能の特色を説明し幅広い理解力を養わせるようにする。

(2) 入出力機器制御装置

磁気ドラム装置，磁気ディスク装置，磁気テープ装置などの種々の周辺装置の動作制御を行なう入出力制御装置にはコンピュータによって様々なものがあり，機能，性能も多様化している。

これらは，周辺装置の種類に対応して独立筐体を持った，入出力制御装置が用意されているものと，周辺装置の筐体に実装される場合がある。

ここでは先ずＪＩＳ－Ｃ－6272について一般的な説明を行ない，次に具体的な例として2，3種類（磁気ディスク制御装置，磁気テープ制御装置，カード読取制御装置など）をとりあげて接続装置台数，種類などを説明し，入出力制御装置の持つべき機能，性能を理解させる。

(3) 磁気ドラム装置

磁気ドラム装置は比較的カタログ性能値と実効値が一致しており，ＪＩＳ－Ｃ－6272の補足説明及び実際の装置を二，三の例をあげ簡単に説明するだけで理解させられる。

特に記憶容量については下記の補足説明を行なう。

① 固定長記録方式（セクタ方式）の場合

１セクタが例えば128バイト等の如く固定長であるので，記憶するデータによっては，例えば100バイトのデータなどでは１セクタで28バイトの記憶容量が無駄となる。従って**実効記憶容量**はカタログ表示値より減少する。

② 可変長記録方式の場合

１セクタの記憶容量をプログラムあるいはハードウェアで定められた範囲内で設定出来るので１セクタの最適な記憶容量を設定することにより実効記憶容量を上げることが出来る。

しかしカタログ表示上は，ハードウェア的に装置として最大記憶容量となる１セクタの記憶容量を設定した値を表示することが多いので，やはり実効記憶容量は落ちることを理解させる。

(4) 磁気ディスク装置

磁気ディスク装置は磁気ドラム装置と比較して相違する点は，磁気ヘッドが移動し従ってその移動時間（シーク•タイム）を考慮する必要があること及び磁気ディスクの交換が可能なものがあることといえよう。（磁気ディスク装置でもトラック毎に磁気ヘッドを持ち，移動の必要性がないのもあるが，これは使用上は磁気ドラム装置と見なせる）従って磁気ディスク装置については磁気ドラム装置との相違を明確に説明し，シーク・タイムについては具体的な装置を二，三例にあげ，移動トラック数と時間のグラフを表示し，説明を行なう。その後，同一シリンダ内のデータや異なるシリンダ上のデータを処理する時間を幾つか例題として計算させる。

一般的には装置単位でこのシーク動作は独立して実行出来ることも説明する。

(5) 磁気テープ装置

磁気テープ装置については規格化，標準化が進んでおり，性能評価はある程度容易であるといえよう。

先ずＪＩＳ－Ｃ－6272につき説明を行ない，次に特に問題となるデータ転送速度について次のように補足説明を行なう。

カタログ性能値の場合のデータ転送速度は，ブロック間隔を無視した値であることをよく理解させ

る必要がある。

このため，例えば次のような図及び式をあげて実際の磁気テープ装置を二，三例にとり**実行データ転送速度**を説明するとよい。

① 連続してデータ転送を行なうとき

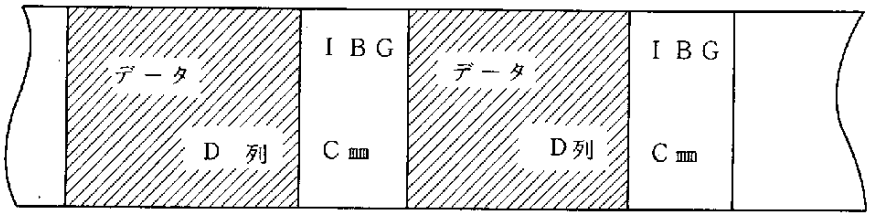
記録密度 ： A （列／mm）

テープ速度 ： B （列／秒）

IBG(Inter Block Gap) ：C （ mm ）

データ長 ：D （ 列 ）

磁 気 テ ー プ



実効データ転送速度 ： X（ 列 ／ 秒 ）

カタログ•データ転送速度 ： Y（ 〃 ）

$$X = \frac{D}{C + \frac{D}{A}} \times B \quad (\text{列}/\text{秒})$$

$$Y = \frac{D}{\frac{D}{A}} \times B = A \times B \quad (\text{列}/\text{秒})$$

図 2 － 1 連続してデータ転送を行なう場合

具体的な装置でこの数式を用いて計算し，カタログ値と実効データ転送速度がどの位異なるかを理解させる。

② 間けつ的にデータ転送を行なうとき

オンライン•リアルタイム処理における ロギング•ファイル として磁気テープ装置を使用する場合には，一般的に磁気テープ装置のデータ転送は間けつ的となり，このようなときには磁気テープ装置に対する書込みあるいは読出し命令が出されてからどの程度の時間内にそのブロックのデータ転送

が終了するかが重要な要素となることがある。

このため磁気テープ装置の起動時間と停止時間が問われることがある。

ここではそれらの時間は近似的に次のように算出されることを説明する。

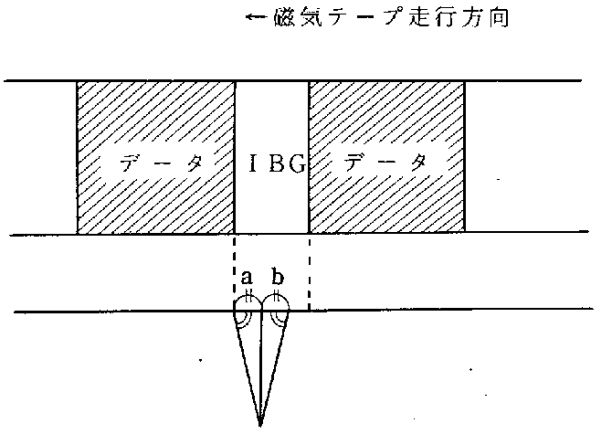


図2－2　間けつ的にデータ転送を行なう場合

今磁気テープ装置の起動時間≒停止時間とし，直線的な速度低下，上昇で起動，停止が行なわれるとすると（実際には多少異なるが近似的に等しいと考えて実用上支障ない），停止命令が出てから停止する迄磁気テープが走行する距離aと起動命令が出てから磁気テープが規格走行速度に達する迄に走行する距離bは等しい。

また，停止時にa だけ走行する時間と起動時にb だけ走行する時間は，規格走行速度でa ＋b だけ走行する時間と等しい。

これは停止動作途中で起動命令が出たときにもいえる。

従って起動命令を受けてから次のデータ転送開始迄に要する時間Tは，常に次の式で求められる。

$$T = \text{IBG通過時間} + t$$

$$\text{IBG 通過時間} = \frac{C}{B}$$

t ： 前のデータ転送終了時から次の起動命令が出される迄の遅れ時間。

連続データ転送時には　t ＝ 0 である。

なお，磁気テープ装置でも時々問題となるテープ1巻の記憶容量について，ＩＢＧも考慮した実効記憶容量の算出法についても説明しておく。

(6) カード読取装置

カード読取装置は比較的カタログ性能と実効性能が一致している。

従ってＪＩＳ－Ｃ－6272の説明を行ない，次に下記事項の補足説明で充分と思われる。

• ホッパの数

複数個持ってホッパの選択をプログラムで実行したり，あるいは，1個のホッパが空になると

他方へ自動的に切り換わるものなどがある。

• スタッカの数と機能

複数個のスタッカを持ち，カードの処理の結果，何れのスタッカへ入れるかがプログラムで選択出来たり，あるいは1個のスタッカでもエラーカードをオフセット•スタックするなどの機能を持つものがある。

• マーク読取機能

せん孔情報のみならず，マークを読取る機能を持つ装置もある。

(7) カードせん孔装置

カードせん孔装置も比較的カタログ性能値と実効性能値が一致している。

従ってＪＩＳ－Ｃ－6272の説明を行ない，次に下記事項の補足説明を行なえばよい。

• せん孔速度

長辺送りの場合にはせん孔欄数により速度が変化する装置がある。これの具体例をあげ説明を行なう。

• マーク機能及び印字機能

カード上にせん孔するのみならず，せん孔場所にマークをつける機能を持つものや，せん孔と同時に欄上部に印字を行なう装置があることを説明し，そのマークや印字されたカードの見本を見せて説明を行なう。

(8) 紙テープ読取装置

紙テープ読取装置も比較的カタログ性能値と実効性能値が一致している。

従ってＪＩＳ－Ｃ－6272の説明を行ない，次に下記事項の補足説明を行なう。

• 読取速度

最近の紙テープ読取装置ではほとんどが1字ずつの読取りが可能であるが，1字ずつの読取りを実施した場合，紙テープ読取装置に対する次の読取命令が一定時間内に出されない場合には規格の読取速度が出ないことがある。

例えば1000字／秒の読取速度の装置では規格の読取速度では1フレームの移動時間は1ミリ秒であるが，前のデータ転送終了後，例えば0.3ミリ秒の間に次の読取命令を出さないとテープ停止動作に入り，読取速度は500字／秒程度まで落ちることがある。

• 取り扱える紙テープの仕様

1台の装置で6単位と8単位などを操作員が切り換えて使用出来るものもある。

(9) 紙テープせん孔装置

紙テープせん孔装置も比較的カタログ性能値と実効性能値が一致している。

従ってＪＩＳ－Ｃ－6272の説明を行ない，取り扱える紙テープの仕様は紙テープ読取装置の場合と同様であることを補足説明する。

(10) ラインプリンタ

ラインプリンタの場合，印字する帳票の形式によってカタログ性能値と実効性能値が大幅に異なることを先ず認識させる必要がある。

ＪＩＳ－Ｃ－６２７２について内容説明を行なった後,特に印字速度について下記のように説明を行ない，実際の帳票で印字を行なう場合の印字速度の計算方法を修得させる。

一般にラインプリンタは活字ベルト式，活字チェーン式，活字トレイン式，活字ドラム式などがあるが，前三者の印字速度は，帳票での改行条件に左右される。従ってこれについては，実際に幾つか帳票形式を設定して改行速度も考慮した印字速度を算出して見せ，算出法を修得させるようにする。

一方活字ドラム式の場合はやや複雑であり，印字速度は一般に印字文字種類と改行速度によって影響される。

このことを次図のように図解し理解させる。

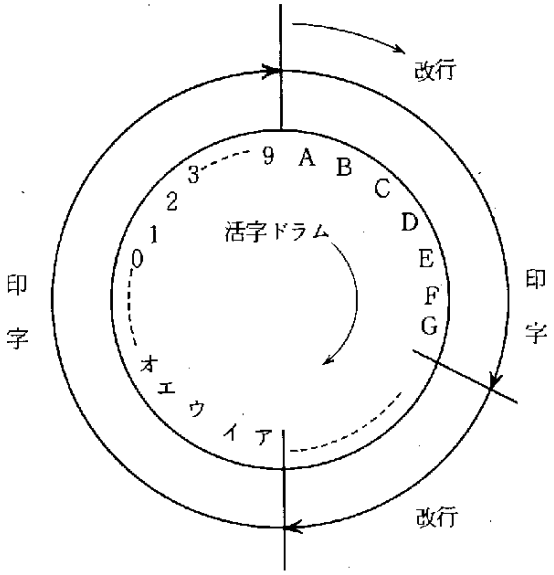


図２－３　活字ドラム式ラインプリンタ

即ち，印字動作は１行中で実際に印字される文字が印字し終ると直ちに改行動作に入り，改行終了時点で直ちに次の行の印字動作に入る。従って印字速度は１行中の印字文字種類と改行速度により常にに変動する。

活字ドラム方式の動作概念を理解させた上で次に具体的な装置を設定し，また帳票形式も設定し，印字速度の算出を行なって見せ，また行なわせる。

2.2　ハードウェア・システムの性能

コンピュータの単体性能評価でさえなかなか困難な点が多く，従ってそれらが組み合わさったシステムの評価となると極めて難かしくなる。

その困難な要素の一つとしてシステムの性能をいう場合にソフトウェアが入らずには意味がないことがあげられる。

またソフトウェアにはオペレーティング・システムなどの他に業務プログラムの内容により千差万別となり，実際に処理を行なうときに，ハードウェア性能のみでシステム性能を評価することは不可能である。

従って実際のシステムの性能はシステム分析を行ない，時にはシミュレーションやベンチマーク・テストを実施し測定することが多いが，ここではハードウェア・システムとして一般的な性能評価について説明を行ない，システム性能評価の基礎を理解させる。

ハードウェア・システムの性能評価をどういった観点より行なうべきか次の点につき説明する。

① 処理能力
- 中央処理装置の演算速度
- チャネルの種類，数，速度
- 周辺装置の種類，数，速度

② 拡張性・融通性
- 主記憶容量
- チャネルの種類，数，速度
- 周辺装置の種類，数，速度

③ 互換性
- シリーズマシン
- 他機種
- 標準規格

④ 運用性
- オペレーション
- 維持費

⑤ 信頼性・保守性
- MTBF
- MTTR
- RAS（Reliability，Availability，Serviceability）

⑥ 設置性
- 寸法，重量
- 電源（周波数，電圧，消費電力等）
- 環境条件（温度，湿度等）

指導上の留意点

ハードウェアの性能評価は先ず単体性能の評価法を確実に習得することが大事である。このために単体性能に関しては出来るだけ多く練習問題や実例を出し演習により習得させることが望ましい。

また性能評価という場合，単なる処理速度にのみとらわれやすいが，その他の機能，性能にも注意を払い，総合的評価が出来るよう指導することが必要である。

第3章　信　頼　性

用　語　この章では次の用語を教える。

MTBF，MCBF，MTTR，誤り再試行，自動診断，FLT

目　標

ハードウェアを構成する部品，回路，装置等につき故障の発生する原因，頻度等につき理解させ，故障が生じないための種々の予防方策や生じた場合の処置ができる能力を身につけさせる。

内　容

3.1 信 頼 性

今日のようにコンピュータが様々な分野に応用され，種々のシステムの中枢として利用されている現在，コンピュータの誤動作や機能停止はその影響力が極めて大きい。

このことを例えば銀行，座席予約，プロセス制御，航空管制などのオンライン・システムにおけるコンピュータの誤動作，機能停止が如何に社会的に多大の損失を与え，また場合によっては人命に影響を及ぼすことすらあることを説明する。

コンピュータの故障の度合や，誤処理の度合など，コンピュータの信頼度を信頼性（Reliability）と称している。

コンピュータの誤動作，機能停止の生ずる原因としては種々あるが，ここでは特にハードウェアにのみ着目し，それらの原因を説明し，それらの障害発生の避けられぬこと，そのためどのような対策がハードウェアでとられ，信頼性の向上を計っているか説明し，理解させる。

(1) 故障発生の原因

コンピュータハードウェアの故障発生の原因としては主に次のことがあげられる。

① 部品の劣化

例えばトランジスタ，集積回路などの経年変化による劣化，ハンダ付，コネクタ等の劣化，機械部の摩耗等

② 製造の欠陥

例えば，部品の不良品の混入，誤配線，接触不良などの製造ミス等

③ 設計の欠陥

例えば部品の規格値外の使用（オーバーロード，タイミングマージンなど），論理設計ミス，製造設計ミス等

④ 外部条件によるもの

例えば電源断，空調異常，短絡，ノイズ等

その他に，システムとして見ればソフトウェアの誤りや操作誤りに起因するものあるが，これらに関しては本章では説明する必要はない。

ハードウェア障害はこれらの原因により発生し，それらが多発するとそのハードウェアの信頼性は非常に低いものとなる。この信頼性を評価する方法としてMTBF（Mean Time Between Failures）やMCBF(Mean Character Between Failures)があることを説明し理解させる。

コンピュータの信頼性をあげることは非常に重要であり，その最も基本的なものは前述の部品や装置の信頼性を高めることとなるが，このため集積回路等の部品信頼性の向上，製造自動化による誤製造の防止，品質管理強化による不良部品混入防止，コンピュータを使用した自動設計システム（CAD……Computer Aided Design）による誤設計防止，電源変動，温湿度変化に耐えうる部品の使用，設計などが用られ最近ではそれらの信頼性はかなり高められている。

しかしコンピュータで使用する部品はぼう大な数にのぼり，それらの信頼性向上のみでは限界がある。例えば磁心記憶の例をとり次のように説明し理解させる。

例）262 Kバイトの磁心記憶では約200万個の磁心が使用されている。

磁心1個が100年間故障しない信頼度を持っていたとしても2百万個集まれば30分に1個故障する確率となる。

従って上記部品，装置の信頼性向上を計ると共に，次の如き考慮が計られていることを説明する。

(2) 障害の検出と訂正

一般のコンピュータでは部品や装置の障害を検出する方策が考慮されており，出来る限り障害による誤った処理を見逃さないよう考慮されている。これはコンピュータの種類により種々の方式や検出程度の詳しさについて様々なレベルはあるが，一般に多く採用されている次の様な方法について説明を行なう。

① パリティ・チェック

奇数パリティ，偶数パリティがある。比較的簡単で確立された方式であり，現在最も広く採用されている。

パリティ・チェックの使用例について説明する。

例）主記憶，磁気テープ，通信回線，加算回路，データ転送路など。

② ハミング・コード・チェック

主記憶のECC（Error Check and Correction）などが代表的である。

ハミング・コード・チェックの特長と原理について説明を行なう。

③ 二重照合チェック

同一処理を二重の回路を用いて処理し，結果を比較する方式で演算部や紙テープ読取装置の読取部などで多く採用されている。

④ そ の 他

その他目的に応じて種々の方式があるが，代表的なものとして次のものを紹介し，動作概略を説明する。

- 不正命令コード・チェック
- 命令読出実行のタイムアウト・チェック
- 不正データ・チェック
- 磁気テープのリード・アフタライト・チェック（Read After Write Check）及びブロック・チェック（サイクリック・リダンダンシ・チェック…… Cyclic Redundancy Check ……CRC）
- エコー・チェック（Echo Check）……機械部の動作確認
- 電源異常チェック（電圧異常，周波数異常等）
- 温湿度異常チェック

以上のように種々の障害検出を実行し，障害が発生した場合でも，そのためにコンピュータが誤った処理結果を出さないよう考慮されているがそれでも完全といい切ることは出来ない事が現状であることも認識させ，このため非常に重要なシステムにおいては更に複数台のコンピュータを使用して同一処理を実行させ，結果を照合したり，ソフトウェアでのチェックやシステム的な配慮を払う場合もあることを説明する。

簡単なコンピュータでは障害検出だけを行ない，障害発生時にはそのコンピュータは動作を停止する方法をとっているが，最近では障害が発生すると自動的にエラーを訂正したり予備回路に切り換えたりして自動修復を計るものもある。これについては3.2の可用性の所で説明を行なう。

なお，これら障害の検出や自動訂正はコンピュータの信頼性向上のために考えられているものであるが，これらを実施するためにはこのためのハードウェア（部品，回路等）が付加されることとなり，この結果コンピュータの部品点数が増加することとなり，これは部品の信頼性の観点よりのみ見た場合にはコンピュータの障害発生を増加させることとなる。従って両者のバランスを常に考慮し，その度合を決定する必要がある。例えば，小さなコンピュータでは余分な検出・訂正回路は付けないで部品点数を出来るだけ減少させることによってコンピュータの信頼性を向上させる方法もよくとられていることを説明しておく。

3.2 可 用 性

コンピュータでは信頼性とともに可用性（Availability）も重要視される。

可用性の意味，重要性，可用性を向上させる方策について次の如く説明を行なう。

(1) 可用性の意味と重要性

コンピュータの信頼性が高い事は重要であるが，現状のハードウェアでは障害が発生することは避けられない。障害が発生した場合にそれを如何に局部的に押え，システムへの影響を少なくするか，またその障害を如何に速く復旧させられるか（MTTR …… Mean Time To Repair）が非常に重要になってくる。ここで可用性即ち如何にしてシステムを最良の稼動状況に保つかが重要であるかを認識させる必要がある。

・ハードウェアは必ず障害を起こす。

・障害を起こしたときシステム・ダウンとなることは極力避けなければならない。

・障害は出来るだけ早く発見し，修復させねばならない。

(2) 方　策

可用性を高めるための種々の方策について一般的によく採用されている次のものにつき説明を行なう。尚,これらの方策は必ずしも全てのコンピュータで採用されているとは限らず，ハードウェアの冗長度に伴なう信頼性や経済性などを考慮し，比較的大きなコンピュータで多く採用されていることを理解させる。

① **誤り再試行**

誤りを検出すると,その処理を再度実行してみる方式で,一時的誤り(intermittent error)の場合に有効である。

② 誤り自動訂正

誤りを自動的に訂正する方式で，主記憶装置のECCや磁気テープ装置のCRC（cyclic redundancy check）等が代表的である。

③ モジュール構成

障害個所をシステムより切離し，予備のものを接続使用出来るようシステムの構成要素（装置，機能モジュール等）をモジュール化した方式である。

④ **自動診断**と記録

障害個所をある単位（パッケージ，集積回路単位等）で自動的に検出したり，障害発生前に不良個所を出したり，障害個所と発生頻度を自動的に記録し，保守，予防保守に役立てる。

・マイクロ診断

・FLT（Fault Locating Test）

・リモート・オンライン診断

指導上の留意点

世の中一般でいわれる正確無比なコンピュータとはいわゆるコンピュータ・システムのことであり，ハードウェアの単体あるいは部品のみを考えて見るとそういわれるほど信頼性の高いものではないことをよく理解させ，コンピュータ・システムとして如何に高い信頼性を保つかにどのような技術，手法が用いられているかをよく理解させる。また実際にコンピュータ・システムを設計するときに如何なる点に注意を払うべきかを習得させるようにする。

第4章　互　換　性

用　語　この章では次の用語を教える。

データの互換性，プログラムの互換性，上位方向互換性

目　標

コンピュータの普及に伴い，コンピュータ間の互換性が重要視されてきているが，これはまた同一ユーザで機種の増設，置き換えを実施するときには非常に重要な問題となることを理解させる。そして，この互換性を保つための種々の条件につき理解させる。

内　容

基礎編「ハードウェア(I)」の1.3標準化で説明したように，互換性は最近特に重要視されて来ていることを説明し，このための国際的なＩＳＯと国内的なＪＩＳの動向と活動の概略を説明し，現状を認識させる。

次に国内におけるコンピュータの互換性・標準化の状況をデータの互換性とプログラムの互換性に分け説明を行なう。

(1) データの互換性

最近の国産コンピュータ及びＩＢＭ社の主力コンピュータを紹介し，これらが全てシリーズ・マシン化されていることを説明し，従ってそれらはシリーズ内でほとんど小型から大型迄**データの互換性**を保っていることを説明する。

異なるコンピュータ・システム間でのデータの互換性については，代表的なデータ記録媒体である磁気テープ，カード，紙テープ，磁気ディスク・パックについて説明を行なう。

(a) 磁気テープ

磁気テープに関しては種々ＪＩＳが制定され，媒体そのものについてはほとんどのコンピュータがこれを採用し互換性を保っているが，符号体系と記録方式についてはコンピュータ独自のもの，あるいはＩＢＭ方式（EBCDIC……Extended Binary Coded Decimal Interchange Code）を採用しているものが多い。

但しこの場合でも媒体そのものの互換性をハードウェアで持ち，符号変換や記録方式（ラベル，テープマークなど）をソフトウェア（一部ハードウェアのものもある）で変換しデータを取り扱うものが多い。このために変換用プログラムを標準的に用意しているコンピュータが多い。

代表的コンピュータの磁気テープ装置を例にあげ説明を行なう。

(b) カ　ー　ド

カードには現在代表的なものとしては80欄カード，96欄カード，90欄カードがある。中でも80欄

カードはほとんどのコンピュータで標準的に採用されていて，カード規格もＪＩＳで制定されているが，カードコードとしてはEBCDIC系とBCD系が多く，ＪＩＳコードは余り採用されていない。この理由をEBCDIC，BCD，ＪＩＳコードの関連及びPCSとの関係を説明し理解させる。

但しほとんどのコンピュータでは主とするのは1種類のカードコードであるが，他のカードコードも読取り，ソフトウェアでコード変換するなどしてデータの互換性を保っていることが多いことを説明する。

96欄カードはＩＢＭシステム／3で初めて採用されたものであり，まだ一般的ではない。

また90欄カードは前にUNIVACが使用していたが最近ではほとんど使用されなくなっている。

(c) 紙テープ

紙テープは電信，さん孔タイプライタ等種々の機械で使用されているデータ記録媒体で，様々な規格のものがあったが，最近ではＪＩＳ等が制定され，6単位と8単位のものが多くなっている。しかし使用されている符号体系は数多く残っており，しかもコンピュータ以外で使われている場合も多くあり，従って一般のコンピュータでは，符号体系の如何を問わずハードウェアで扱い，符号変換をソフトウェアで実施し互換性を保つものが多いことを説明する。

また6単位と8単位の両方を手動操作により1台の紙テープ読取装置で取り扱えるものもある。

(d) 磁気ディスク•パックに関してはその媒体の特質により，情報交換用というより，コンピュータの置き換えを実施する際に前のコンピュータで保有していた磁気ディスク•パック上のデータの取り扱いの時に問題となることを理解させる。

現在ＪＩＳでは規格が制定されておらず，従って標準化は余り進んでおらず，異なるコンピュータ間のデータの互換性については現在余り考慮されていない。

(2) プログラムの互換性

プログラムの互換性については既に「ハードウェア(I)」の1.3標準化で述べたように下記方法で互換性が保たれているが，これらの利害得失について説明し理解させる。

① ソース•プログラムでの互換性

② オブジェクト•プログラムでの互換性

③ エミュレータによる互換性

④ シミュレータによる互換性

⑤ コンバータによる互換性

一般的にシリーズ•マシンでは②の方法による互換性が配慮されており，小型コンピュータで作成されたオブジェクト•プログラムをそのまま大型コンピュータで処理出来る場合が多い。

しかし大型コンピュータでは小型コンピュータにない種々の機能，例えば浮動小数点演算機能，多重プログラミング機能などが備わり，このため大型コンピュータで作成されたオブジェクトプログラムは小型コンピュータでは処理できないのが普通である。このように，より大型コンピュータへの互換を保つことを**上位方向互換性**と呼んでいる。

また小型コンピュータと大型コンピュータの使用形態は当然異なっており，小型コンピュータ用に作成されたプログラムをそのまま大型コンピュータで処理させても大型コンピュータ本来の処理能力

は発揮出来ない。

このため単にハードウェアで互換性を保つ上記①の方法のみでは不充分となり，最近では上記①，③あるいは⑤の方法が採用されることが多い。またこれらの方法が目的によって組み合わせて使用されることも多い。

①の方法は小型コンピュータは小型なりに，大型コンピュータは大型なりの性能を発揮でき，しかも互換性を保つときに良く採用されている方法である。この方法はユーザビジブルなソフトウェア仕様のみを小型〜大型コンピュータで統一しておけばハードウェア仕様は異なったものであっても達成出来るが，ハードウェア機能の相違を全てソフトウェアで補うこととなり，ソフトウェア効率を落しシステム性能に影響を与えることがある。従って出来るだけハードウェア仕様も統一化し，その上でソフトウェア仕様も統一することが望ましく，このためアーキテクチャの一貫性が重要視されることとなる。

③の方法は，アーキテクチャの異なるコンピュータでオブジェクト・プログラムの互換性を保たせる場合に良く採用されている。エミュレータによる場合のプログラム処理速度は本来の性能より数倍劣るのが普通であって従ってこの方法は，コンピュータの置き換えなどのとき従来のコンピュータで保有していたプログラムを性能は多少落ちても新しいコンピュータでそのまま使用したいときなどに良く利用される。

ＩＢＭシステム／３６０のＩＢＭ１４０１ エミュレータなどを例にとり，エミュレータの性能や仕様について説明を行なう。

⑤の方法は全く互換性のないプログラムをコンピュータで変換用プログラム（コンバータ）を利用して自動的にプログラムをそのコンピュータ用に変換するものである。この場合にハードウェア，ソフトウェア共に関連のないコンピュータ間の自動変換には限界があり，ある程度人手でプログラムの変換を行なう必要性が出てくるのが普通である。

④の方法は互換性のないプログラムを一旦主記憶に全て読み込み，そのプログラムの動作をソフトウェアでシミュレートしプログラム処理を実行する方式で従って処理速度は格段に遅くなる。このため最近では余り採用されていない。

指導上の留意点

コンピュータを導入する場合あるいは利用する場合，単にそのコンピュータの処理能力のみならず互換性の立場でも良く検討する必要があることを理解させる。

即ち，現在においてはコンピュータの利用は単にそのシステム内に止まることは少なく，他システムからのデータの処理や他システムへの出力などが要求されることが多い。

また作成されたプログラムやデータファイルはユーザの大きな財産の一つであり，これの将来利用の保証が如何に大事なことかを理解させるようにする。

第5章　実　習

目　標

前章までに履修した事項に関して実際にコンピュータを使用した実習を通してコンピュータに対する具体的知識を修得し評価能力を養わせる。

内　容

第1章から第4章の順で以下のごとく説明し，実習させる。

(1) 処理方式

先ず実習用コンピュータに関する説明書を充分勉強させた上で，そのコンピュータがどのような処理方式を採用しているかを理解させる。特に第1章の教えるべき用語に関する事項についてはそのコンピュータの具体的方式を理解させるために，ハードウェアレジスタのダンプを行なったり，プログラムを実行させての実習を通して説明を行なう。

(2) 性能評価

第2章でとりあげている各装置に関して実効性能を机上計算で算出させ，次に実際にコンピュータを使用して実測データを求めこれらをカタログ性能と比較させる。実習は各種ケースについて幾つか実施し，実効性能を正しく机上計算で求められるよう指導する。

(3) 信頼性

実習用コンピュータで信頼性に対して如何なる方策がとられているかを説明し，可能ならば疑似的に障害を発生させ，その影響や障害個所の検出法，復旧法などを実習させる。また，実習用コンピュータの稼動実績を集計しておいて，障害がどのような所で発生するかやそのときの状況等を説明したり，定期保守時などに保守の状況を見学させたりするとよい。

(4) 互換性

互換性については，実習用コンピュータが複数機種用意されていなければ実習は困難であるが，出来るだけ他機種も使用出来る機会を設けて，プログラムの互換性やデータの互換性に関してどのような考慮が払われているか，またどのような点に注意する必要があるかを実習により体得させることが望ましい。

指導上の留意点

前章まで説明したような機能性能を全て備えたコンピュータは現状では非常に数少なく，また，コンピュータ・システムも大規模なものとなる。実習で使用出来るコンピュータでは機種も限定され，機能性能も制約されたものとなる。従って実習はその範囲でのものとならざるを得ないが，実習するコンピュータの方式にのみとらわれるような指導は避け，常に一般的見地よりそのコンピュータを評価するよう指導する必要がある。

参考文献

(1) J. Fotheringham著 「Dynamic Storage Allocation in ATLAS Computer, Including Automation Use of Backing Store」 Communication ACM 昭和36年

(2) 「Multics System」 MIT助教授 J. H. Saltzer 講演 日本電子工業振興協会 昭和46年

(3) 工業技術院 「大型プロジェクトによる開発。概要（超高性能電子計算機）」 昭和45年

(4) H. Barsamian, A. DeCegama 著 「Evaluation of hardware-firmware-software trade-off with mathematical modeling」 Proc. of SJCC 昭和46年

(5) S.S ハッソン著 元岡 達訳 「マイクロプログラミング」 日本経営出版会 昭和48年

(6) J. マーチン著 北原安定訳 「リアルタイム」 日本経営出版会 昭和43年

(7) 日本規格協会 「JIS ハンドブック（情報処理）」 昭和46年

(8) 情報処理学会 「システムの性能評価特集号」 情報処理 1972 Vol. 13 No. 11 昭和47年

(9) 日経エレクトロニクス 「仮想記憶と仮想計算機の歴史と現状」 昭和48年3月

(10) 情報処理学会 「RAS技術の現状」 情報処理 1971 Vol. 12 No. 8 昭和46年8月

(11) 情報処理学会 「小特集 RAS」 情報処理 1972 Vol. 13 No. 9 昭和47年9月

(12) W. L van der Poel 著 「The Essential Types of Operations in an Automatic Computer」 Nachrichtentechnishe Fachberichte Vol. 4 1956

オペレーティング・システム

目　　次

第1章　オペレーティング・システムの基本 …… 398

1.1　ソフトウェアとハードウェア …… 398

1.2　オペレーティング・システムの定義とねらい …… 398

1.3　オペレーティング・システムの構成 …… 400

1.4　オペレーティング・システムの処理形態 …… 400

第2章　プログラム管理 …… 401

2.1　プログラム処理過程 …… 401

2.2　プログラムの分割 …… 401

2.3　プログラム構造 …… 401

2.4　プログラム再使用性 …… 402

2.5　デバグ機能 …… 402

2.6　プログラム・ライブラリ管理 …… 402

第3章　ジョブ管理 …… 404

3.1　ジョブ・ステップ …… 404

3.2　ジョブ制御言語 …… 404

3.3　ジョブ・スケジューリング …… 406

3.4　資源管理 …… 408

3.5　オペレータ・コミュニケーション …… 409

3.6　アカウンティング機能 …… 410

3.7　プログラム・ロード …… 411

第4章　タスク管理 …… 412

4.1　割込制御 …… 412

4.2　タスク制御 …… 413

4.3　演算装置割付 …… 414

4.4　メイン・メモリ管理 …… 414

4.5　時刻管理機能 …… 415

第5章　データ管理 …… 416
5.1　2次記憶媒体 …… 416
5.2　データ管理の概念 …… 417
5.3　ファイル制御 …… 417
5.4　アクセス手法 …… 418
5.5　入出力制御 …… 419
5.6　データ管理のユーティリティ …… 421
第6章　通信管理 …… 423
6.1　オンライン・リアルタイム・システムの目的と基本 …… 423
6.2　ハードウェア・システム構成と特性 …… 424
6.3　端末機と中央処理装置の通信 …… 425
6.4　リアルタイム・システムのプログラム構成 …… 429
6.5　障害対策 …… 431
第7章　障害管理 …… 433
7.1　高利用度の必要性 …… 433
7.2　リスタート能力のサービス …… 433
7.3　ファイル・ソフト機能 …… 434
7.4　診断ルーチン …… 434
7.5　障害情報のロギング …… 434
第8章　タイム シェアリング・システム …… 435
8.1　タイムシェアリング・システム概念 …… 435
8.2　タイムシェアリング・システムの基本機能 …… 436

科目「オペレーティング・システム」

教育の目標

コンピュータのハードウェアの発展とともに，ハードウェアの機能を有効的に活用するためにオペレーティング・システムは重要な役割を持つようになった。

現在では，オペレーティング・システムなしのコンピュータ・システムを考えることは出来ない。

ここでは，このオペレーティング・システムのうち特に管理プログラムを中心とした機能について理解させ，オペレーティング・システムを評価する能力を養うことを目標とする。

時間配分

章	履習時間（時間）		
	講　義	演　習	実　習
1.　オペレーティング・システムの基本	5		
2.　プログラム管理	6		
3.　ジョブ管理	15		
4.　タスク管理	7		
5.　データ管理	15		
6.　通信管理	12		
7.　障害管理	5		
8.　タイムシェアリング・システム	5		
合　　計	70		

第1章　オペレーティング・システムの基本

用　語　この章では次の用語を教える。

アイドル・タイム,

目　標

ここでは，コンピュータ・システムのオペレーティング・システムを理解させるための準備として「オペレーティング・システムの定義」，「オペレーティング・システムはどのような必要性から発展してきたか」，「現在のオペレーティング・システムのねらいは何か」，「オペレーティング・システムの処理形態」，「オペレーティング・システムの構成」を示しながら，オペレーティング・システムとは何か，またオペレーティング・システムの位置づけ，役割を認識させる。

内　容

1.1　ソフトウェアとハードウェア

コンピュータ・システムのアルゴリズムは，一部がハードウェアの中に，また一部がソフトウェアの中に展開されている。また最近では，ハードウェアとソフトウェアの間にファームウェアなるものが介在するシステムもある。ここでは，ソフトウェア，ファームウェア，ハードウェアの概念，特徴，位置付けを行なう。

1.2　オペレーティング・システムの定義とねらい

(1) オペレーティング・システムの定義

オペレーティング・システムの定義は，ＪＩＳ用語にも登録されていない。しかし，ここでは，何らかの定義を行なうことに努める。

オペレーティング・システムは，ターン・アラウンド・タイムの短縮，スループットの増大等によるコンピュータの有効利用，更にプログラムの作成の手法および操作の簡潔化と標準化を目的としたものである。

(2) オペレーティング・システムのねらい

オペレーティング・システムは，オペレーション上で生じる遊び時間を短縮し，コンピュータ・システムの利用効率の向上をねらいの一つとして発展して来た。これをアイドル・タイムの短縮，ノン・プロダクティブ・タイムの短縮，ノン・プロダクティブ・タイムの有効利用の立場から説明する。

① アイドル・タイムの短縮

コンピュータの操作に関して人手による介入（そのときに生じたコンピュータの遊び時間を**アイ**

ドル・タイムという）を最少にすること。そしてきまりきった手順は，人間よりもはるかに「正確に」，「より速く」やってくれるコンピュータにこの操作をまかせること。

② ノン・プロダクティブ・タイムの有効利用

中央処理装置の動作と入出力装置の動作には，処理速度に大きな差がある。そこで入出力装置が入出力動作を行なっている間（これをノン・プロダクティブ・タイムという）中央処理装置はほとんど遊んでしまう。この中央処理装置の有効利用について説明する。

- チャネル装置の時分割制御機能による入出力装置と中央処理装置の動作の重ね合せ（これは同時動作，並行動作という。）
- いくつかのプログラムを同時に記憶装置に入れ，お互いのプログラムでこのノン・プロダクティブ・タイムの利用を図るマルチプログラミング

現在のバッチ処理のためのオペレーティング・システムは，このようなコンピュータの遊休時間を少なくして，全体のスループットをあげる必要性から生まれたものであり，モニタとかスーパーバイザと呼ばれる管理プログラムが上記の問題解決のための主役を果している。

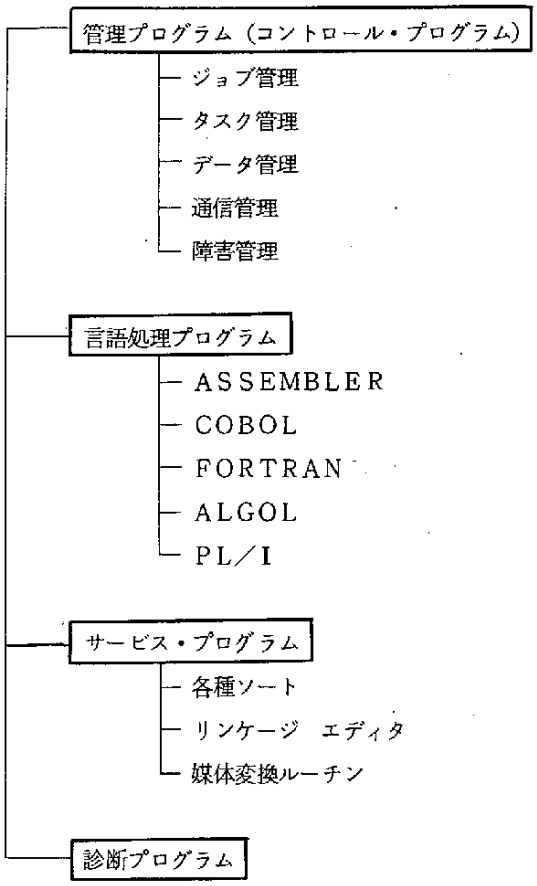


図1－1　オペレーティング・システム構成例

(3) 現在のオペレーティング・システムのねらい

ここでは，次の項目を通して，現在のオペレーティング・システムのねらいの紹介とともに，オペレーティング・システムの必要性を認識させる。

① スループットの向上

② 応答時間の短縮

③ 汎用性の向上

④ 使いやすさ

⑤ 適応性の向上

⑥ 拡張性の向上

⑦ アベイラビリティの向上

1.3 オペレーティング・システムの構成

ここでは，オペレーティング・システムの構成から，各モジュールの概要紹介を行なうことにより，今後説明を行なう個々の管理機能の位置づけを行なう。(図 1 －1 参照)

1.4 オペレーティング・システムの処理形態

オペレーティング・システムの処理形態ついて，名前をあげ，定義し，その処理方式について説明する。(図 1 －2 参照)

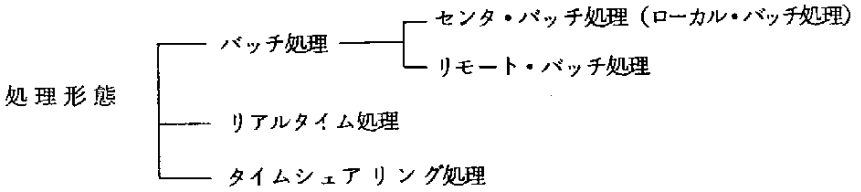


図 1－2　オペレーティング・システムの処理形態例

指導上の留意点

(1) この章の一部は，基礎編「ソフトウェア」で述べられていることの復習とする。

(2) オペレーティング・システムの位置づけは，ハードウェアとソフトウェアの接点であることを認識させる。

すなわちハードウェアとソフトウェアが一体となって，システムとして機能していることを理解させる。

(3) 各項目については，詳細な説明はさけて，大局的に紹介する。

第2章　プログラム管理

用　語　この章では次の用語を教える。

トレーサ，スナップ・ショット・ダンプ，ダンプ，ライブラリ

目　標

プログラマが，プログラムの設計から作成にあたり，コンピュータ・システムのもつ機能（資源）を十分活用するために必要な基本的知識ならびにオペレーティング・システムが，プログラマに対してプログラムの生産性の向上のために用意している機能について説明する。

内　容

2.1　プログラム処理過程

コンピュータである仕事をするとき，その仕事の手続きをプログラムに記述する。ここでは，オペレーティング・システムでの処理プログラムの位置づけ，プログラム言語，プログラム処理の流れについて説明する。

2.2　プログラムの分割

複雑で大きなプログラムを，扱いやすい小さなサブプログラムに分割して設計することは，プログラム作成上の効率を考える場合の有効な手段である。

このプログラム分割の考え方，プログラム分割の効果について次の観点から説明する。

① プログラム作成の共同作業

② プログラムの言語特性の利用

③ デバッグの容易さ

④ プログラム・メンテナンスの容易さ

⑤ サブプログラムの汎用利用

また，サブプログラムのプログラム構造について説明する。

2.3　プログラム構造

プログラムの分割，あるいは統合の概念は，サブプログラムの間で階層構造をもたらす。これは，主記憶の多重利用を意味する。

ここでは，それをプログラム構造としてとらえ，単純構造，計画オーバレイ構造，ダイナミック構造

に分類し，その概念，構造，特徴を説明し，その効果を理解させる。またアセンブラ，コボルなどの言語の中での機能仕様の有無について紹介する。

2.4 プログラムの再使用性

ここでは，ノン・リユーザブル，シリアリ・リユーザブル，リエントラント，リカーシブの各プログラムの構造について説明する。また特にシリアリ・リユーザブルとリエントラントのプログラムの特徴の比較を行ない，プログラム設計のポイントを認識させる。また，アセンブラ，コボルなどの言語の中での機能仕様について紹介する。

2.5 デバグ機能

マルチ・プログラミング・システムでは，プログラム・デバグにあたって制御卓を通して監視するような方法はとれない。

従って，ユーザに対するサービスとして，オペレーティング・システムは，デバグ機能を提供し，コンピュータの使用効率の向上，ターン・アラウンド・タイムの短縮に多大な効果を与えている。

ここでは，各プログラム言語が用意しているデバグ機能は，紹介程度にとどめ（詳細な説明は，各言語コースで行なわれる。），主たる内容をオペレーティング・システムが用意している診断ルーチンの機能と使用法について認識させる。

診断ルーチンは，特に論理ミスの検出の補助として使用される機能で，次の項目について機能，使用法を説明する。

① トレーサ
② スナップ・ショット・ダンプ
③ ステータス・ダンプ
④ メモリ・ダンプ
⑤ ファイル・ダンプ
⑥ テスト・データ・ジェネレータ
⑦ 特定時点で実行開始位置を指定しなおして再実行
⑧ 特定領域にデータをセット
⑨ 誤まりを検出したときに，各レジスタ指定領域の内容のダンプ

2.6 プログラム・ライブラリ管理

プログラムには，ソース・プログラム，オブジェクト・プログラム，ロード・プログラムと種々なレベルがある。プログラム・ライブラリの管理は，おのおのの特性に応じたプログラムの管理，メンテナンス機能をユーティリティ・プログラムとして用意している。また**ライブラリ**には，私用ライブラリ，システム・ライブラリの属性があり，プログラムの性格から分類されて管理出来る考慮が払われてある。これら機能について説明する。

指導上の留意点

(1) プログラムの設計の仕方の詳細が具体的なものは，基礎編「プログラム設計(I)」，専門編「プログラム設計(II)」で行なわれる。ここでは，特にオペレーティング・システムでは，プログラムはリソースとして管理対象となる点から指導する。

(2) ソース・モジュールからロード・モジュールまでのプログラム処理過程の中でコンパイラ機能，編集（リンケージ）機能とその使用法を把握させる。特に，オブジェクト・モジュールは，言語翻訳プログラムに無関係に共通構造を持つことを理解させる。

(3) 再使用性に関しては，構造の理解につとめる。たとえば，リエントラントは，コード部，データ部が独立していて，利用者は，データ部を持ち，コード部を共通利用出来る形式である。またシリアリ・リユーザブルは，1トランザクションに対してサービスが完了したときに自分自身を初期設定出来る能力を持つ形式であり，このプログラム構造は，待ち行列制御が必要である，といった見方で説明する。

(4) 仮想記憶が採用されているハードウェアは，ここでは考慮しなくてもよい。

第3章　ジョブ管理

用　語　この章では次の用語を教える。

ジョブ・ステップ，ジョブ，スタックド・ジョブ・スケジューリング，プライオリティ・ジョブ・スケジューリング，資源（リソース），領域（パーティション），アカウンティング機能

目　標

ユーザによりジョブがどのように計画されるか，そしてジョブはコンピュータ・システムで，いかに遂行されるか，又，そのために必要とされるシステムとのコミュニケーション機能などのジョブまわりの機能について修得させる。すなわち，ユーザは，コンピュータ・システムにジョブを投入するとき，ジョブ制御言語により，ジョブおよびジョブ・ステップの性格，各ジョブ・ステップを実行するために必要なシステム資源の記述を行なう。この一部は，コミュニケーション手段を通して与えたり，変更したりできる。加えてユーザによって用意されたジョブ制御言語を読み込み，解析し，各ジョブの実行をジョブ・ステップに従ってスケジュールし，ジョブの処理結果を指定された出力装置に出力する一連のジョブの流れを制御する機能について把握させる。

内　容

3.1　ジョブ・ステップ

ユーザがコンピュータ・システムに与えるときの仕事の単位であるジョブ，その構成要素であるジョブ・ステップの概念を説明する。**ジョブ・ステップ**は使用される資源，処理の機能の違いにより分割されたプログラムに対応づけられ，また，**ジョブ**はユーザの仕事の最終ゴールのとらえ方によって，決定される論理的な単位である。そして，次に述べるジョブ制御言語によって明確に規定され，コンピュータ・システムに投入されるものである。

3.2　ジョブ制御言語

ここでは，ジョブを規定する目的のために用意されているジョブ制御言語について，その機能，使用法を説明する。

(1) ジョブ制御言語の構文

ジョブ制御言語は，種々のパラメータを持っている。このパラメータの記述のために次の二つの説明が重要である。

・位置パラメータ

・キー・パラメータ

(2) 文の種類

ジョブ制御言語を分類すると，次のようになる。

・ジョブ記述文

・ジョブ・ステップ記述文

・ファイル記述文

・データ投入文

・ジョブ・フロー文

・オペレータ通信文

ジョブ制御言語に持たされている属性とそれら属性値は，オペレーティング・システムのユーザ情報の一つとして非常に重要である。ここでは次の二点を目的として説明する。

・ジョブ制御言語でみられる属性と属性値を認識し，文を記述する能力を養なうこと。

・個々のジョブ制御言語の組合せによりジョブを記述する能力を養なうこと。

(3) 代表ジョブ制御言語とその属性の例

① JOB文

・利用者の名前

・優先度

・限定指定

打切実行時間

打切出力桁数

打切出力ページ数

打切出力カード枚数など

・ジョブ開始の時刻指定

・ジョブ名

・指定終了時刻など

② EXEC文

・プロシージャ名

・ジョブ・ステップ名

・メモリの大きさ

・プログラムに与える情報

・ダンプ情報

・デバグ・モード情報

③ FD文（ファイル装置定義文）

・パーマネント・ファイル指定

・ファイル名

・機密保護情報

・ファイルの割当情報

・ファイルの拡張割当情報

・ファイル情報

(4) ジョブ記述の例

```
$JOB      JOBA
$EXEC     STEP1
$FD       FD1A
$DATA
   ≀
$EOF
$EXEC     STEP2
$FD       FD2A
   ≀
$FD       FD2Y
$DATA
   ≀
$EOF
   ≀
$EOF
$EOJ
$JOB
```

ジョブステップ

データ

ジョブステップ

データ

ジョブ

(5) ジョブ制御マクロ

ジョブはジョブ制御言語で記述される。しかし，また，ジョブ制御言語は多種あり，また，各々のジョブ制御言語は多くのパラメータを持っている。これはオペレーティング・システムが種々なユーザの要求に応ずるべく能力を提供するためである。しかし，このことはユーザにとっては，一面，わずらわしい要素となる。その対策としてオペレーティング・システムがジョブ記述の便宜法として用意しているジョブ制御マクロ機能について，次の事柄を説明する。

① ジョブ制御マクロの作り方

② ジョブ制御マクロの登録（カタログ）

③ メンテナンス

④ ジョブ制御マクロの呼出

3.3 ジョブ・スケジューリング

(1) ジョブの入力／翻訳処理

一般にジョブは，その実行に先立って他のジョブの実行とは併行的にシステムに入力される。ここでは，この仕事を行なうシステム・プログラムの機能について説明する。

① ジョブ入力装置の種類

② 入力されたジョブの待ち行列の形式

③ ジョブ制御言語の文法エラー，解析

また，このシステム・プログラムの起動のされ方について説明する。

④ 動的に動作させるために，このプログラムのために資源を確保しておく。

⑤ 入力装置（カード・リーダ）の起動によりのこのシステム・プログラムを起動させる。この時，メモリ資源が用意出来ないときは，優先度の低いプログラムを一担ロールアウトして該当プログラムを優先して処理させる。

(2) ジョブの始動

ジョブの始動はすなわちジョブの選択法は，大きく二つに分類出来る。

・スタックド・ジョブ・スケジューリング（シーケンシャル・ジョブ・スケジューリング）

・プライオリティ・ジョブ・スケジューリング

これらのスケジューリング法，ジョブ始動に伴なうSYSIN SYSOUT ファイル等の準備機能について説明する。またプライオリティ・ジョブ・スケジューリングについては，そのスケジューリング・アルゴリズムについて説明する。すなわち，アルゴリズム要因として，

・優先度

・リソースの競合状態

・ジョブの大きさ

・ジョブの始動開始時刻

・ジョブの終了時刻

・ジョブが待たされている時間

などがあり，この要因がシステムでどのように取扱われているかについて，その概要を説明する。

また，リアルタイム・ジョブ，タイムシェアリング・ジョブのスケジューリングについて若干ふれることも必要である。

(3) ジョブ・ステップの始動

ここでは，ジョブ・ステップ始動時の仕事，すなわち

・EXECステートメント処理

・FDステートメント処理

等，ジョブ制御言語処理を通して，システムに依頼されるジョブ・ステップを遂行するための準備事項として下記のものを説明する。

・メモリの確保

・入出力装置の割当

・ファイル／スペースの確保

・ボリュームの認識／取付け指示

(4) ジョブ・ステップの実行

以上(3)までに紹介されてきた機能を通して，ユーザのジョブが実行される。(3)では最終的にオペ

レーティング・システムでの用語でいうとジョブ・ステップ・タスクが生成されたことになる。このタスクがいかなる制御のもとで実行されていくかは，第4章で説明される。

(5) ジョブ・ステップの終了

ここでは，ジョブ・ステップの終了に伴なって行なわれる，後始末処理について説明する。

- 異常終了の後始末処理
- メモリの解放
- 入出力装置，ファイル装置等の解放
- 会計情報の記録
- パーマネント・ファイルのカタログ・ファイルへの登録
- 次のジョブ・ステップのスケジューリング準備

(6) ジョブの終了

ここでは，ジョブの終了に伴なって行なわれる後始末処理について説明する。

- SYSOUTファイルのクローズ
- ジョブに割当てた資源の開放
- ジョブ・コントロール・ブロックの開放

(7) ジョブの出力

一般にジョブは，直接出力装置（プリンタ装置等）に出力しても，システムのパフォーマンスが低下しない場合を除いて，ユーザが定義するファイル（テープ・ファイル，ディスク・ファイル）とかシステムのファイル（SYSOUTファイル）に出力される。そして，ジョブが実行されたあとで最終的に出力装置に出力される。ジョブの出力機能について，次の事柄を説明する。

① ジョブ出力の目的
- 出力装置の有効利用
- 一つのプログラムでn種の報告書を作成する能力の提供
- いくつかのプログラムで同時に報告書を作成する能力の提供

② SYSOUTファイルの構造

③ ユーザ定義ファイル，SYSOUTファイルの選択

④ ジョブ出力の能力

3.4 資源管理

ジョブおよびジョブ・ステップをスケジュールするには該当するジョブおよびジョブ・ステップをサポートするハードウェアとソフトウェアを**資源（リソース）**という。ここでは，次の資源（リソース）について説明する。

- メモリ
- ボリューム／デバイス
- ファイル

しかし，各資源のとらえ方にはシステムによって異なることを注意する必要がある。

(1) メモリ

メモリ資源のとらえ方には次のものがある。

① **領域（パーティション）**として，イニシャリゼーション時に固定されるもので，この場合は資源管理の対象とはならない。

② ジョブ・ステップのロード・モジュールのために開始時にメモリの割付を行なう。

③ プログラムの実行時に動的にメモリ割当の要求，解放が出来る。

(2) ボリューム／デバイス

ボリューム／デバイスの割当／解除などボリューム／デバイススの管理用テーブルを返して行なうオペレーティング・システムの資源管理機能について説明する。

(3) ファイル

ファイル資源の割当はファイルによって，その機能が異なる。すなわち，ディスク装置で代表されるダイレクト・アクセス装置で維持されているファイルは，処理モードと共用モードの組合せによる割当制御機能が働く。これらファイル割当機能について説明する。また，デッドロックの問題を考慮する必要がある。

3.5 オペレータ・コミュニケーション

オペレータ・コミュニケーションには大別して2種類ある。

- オペレータ・コマンド
- オペレータ・メッセージ

ここでは，この二つの観点から説明する。

(1) オペレータ・コマンドの種類

① オペレーティング・システムの状態表示コマンド

- 現在，動作中のジョブの表示とジョブ・ステップの位置
- ジョブの開始を待っているジョブのリスト
- アウトプット・ライタによる処理を待っているジョブのリスト
- 入出力装置の状態，2次記憶装置のスペース状態
- オペレータ応答メッセージの未応答リスト

② オペレーティング・システムの状態を変化させるコマンド

- 入出力機器の状態の変化

③ ジョブに対する指示コマンド

- ジョブの中断
- ジョブの取消
- ジョブの中止
- ジョブの優先度の変更

④ メッセージの応答コマンド

- システム開始時の日付，時刻の入力

・ボリュームの取付，取はずし指示と準備完了指示

・入出力エラー等に対する指示

⑤ プロシージャの開始，終了指示コマンド

・リーダーの起動指示

・アウトプット・ライタの起動指示

⑥ ジョブ・ステップ・フローに関するコマンド

⑦ オペレーティング・システムの停止の指示コマンド

(2) オペレータ・メッセージ

コントロール・プログラム，ジョブ・プログラムとオペレータの間でかわされるシステム運用のための色々な情報がある。この機能で大切なことは，オペレータがこの情報を迅速な判断のもとで対処出来ることであり，このためのサービスとしてメッセージの中にいくつかの属性がある。

この属性の役割について説明する。

① タスク標識

② メッセージを出力したモジュール標識

③ 通　番

④ 対処法

・オペレータに情報通知

・オペレータに動作指示

・オペレータに応答要求指示

(3) オペレータ・コマンド形式

オペレータ・コマンドは，システム・パフォーマンスに対して大きな要因となる。従って以下のような考慮が一般に払われている。

・コンソール・タイプライタは低能力装置ゆえ，メッセージ・コマンドは情報機能を失わない程度に出来るだけ短かくする。

・高能力なディスプレ装置の採用

・マルチ・コンソール・システム機能

・コマンド入力に対する修正機能

・メッセージ／コマンドの識別機能

・不必要なメッセージの出力をさける

3.6 アカウンティング機能

アカウンティング機能は会計情報の記録機能である。会計情報として，次のようなものがある。

・演算装置の使用時間

・メイン・メモリの使用量

・チャネルの使用量

・入力カード数

・出力行数，または，ページ数

・入出力の回数

・二次記憶装置のスペースの使用量

・ジョブ・ステップの開始と終了の時刻

・ジョブ・ステップ識別用の情報

3.7 プログラム・ロード

2次記憶媒体上のプログラム・ファイルからメモリにプログラムを格納するプログラム・ロード機能について説明する。また，この際，プログラム・アドレスとメモリの格納アドレスとの対応づけのための方法についても説明する。

指導上の留意点

この章では，特にジョブ制御言語，ジョブ制御マクロ，オペレータ・コミュニケーションの説明に重点を置いて，ジョブの流れの中でのその役割について理解させることに努める。

第4章　タスク管理

用　語　この章では次の用語を教える。

タスク，タスク管理プログラム，イベント，スワッピング，ロール・イン／ロール・アウト

目　標

コンピュータ・システムの中でタスクは中央処理装置，主記憶装置，データの集合，プログラムなどを使用して生産活動を行なう。この資源の使用を調整するコントロール・プログラムは，タスク管理と総称され，その機能を果たすルーチンがスーパーバイザである。

とくに，複数個のプログラムを平行的に取扱う多重プログラミングの場合は，それぞれのタスクを識別し，優先度を知り，それにもとづいてシステムの資源を割当てなければならない。

ここでは，タスクの概念とタスクを遂行させる論理的条件となる事象（イベント）および，スーパーバイザの役割と機能を説明し，スーパーバイザと処理プログラムとのコントロールの流れ，システムの資源を競合するタスクへ割当てるスケジューリング・アルゴリズムなどを理解させ，さらにコンピュータ・システムの効率的な使用に対するオペレーティング・システムの働らきを認識させる。

内　容

4.1　割込制御

ハードウェアの割込機能をマルチ・プログラミング・システムの基本事項として説明する。

① 次の事柄について応用編「ハードウェアII」の復習を行なう
- コンピュータ・システムの割込導入の目的
- ハードウェアの割込時の処理
- 割込の種類

② 割込のソフトウェア利用技術について，その代表的な割込をとらえて説明する。
- コンソール割込
- 入出力割込
- タイマー割込
- プログラム割込
- マシン・チェック割込
- 割込命令による割込

③ 最後に多重割込制御と割込マスク機能について説明する。

4.2 タスク制御

(1) タスクの定義

オペレーティング・システムが規定した仕事の単位である**タスク**の概念を説明する。 しかしタスクを定義し理解させるには，抽象的になりがちである。

そこで，次の観点から説明を行なう。

① ユーザからみた仕事の単位ジョブ，ジョブ・ステップとタスクの関係

② プログラムおよびデータはタスクを生成するリソースである。

③ プログラム構成，特に共用プログラムにおけるプログラムとタスクの関係。

一般には，一つのジョブ・ステップは一つのタスクとなる。しかし一つのジョブ・ステップが複数個のタスクとして実行管理されるシステムもある。

(2) TCBとタスク状態

コンピュータ・システムの中では，仕事は，タスクとして管理される。**タスク管理プログラム**は，このタスクを管理するためにTCB（Task control block）なる制御テーブルを用意している。ここでは，TCBの存在とTCB属性の代表的なものを説明する。

タスク状態とは，タスクを制御するためにタスク管理プログラムが設定している状態であり，その状態はTCBテーブルで管理される。代表的な状態に次のものがある。

① RUNNING状態

② READY状態

③ WAIT状態

ここでは，この各々の状態の明確な定義，各状態の推移について説明する。

(3) タスクの生成／消滅

タスク管理プログラムのタスク生成／消滅機能について説明する。

① タスクの生成

基本的には，制御プログラムはジョブ・ステップが最初に実行されるとき，一つのタスクを生成する。しかし，1ジョブ・ステップの中で二つ以上のタスクを生成し，一つのジョブ・ステップの中で並列操作を可能にするシステムがある。これはATTACHマクロ命令をユーザ・プログラムに提供することによってなされる。

ここでは，次の事柄について説明する。

- タスクの生成の時点
- ATTACHマクロによるタスクの生成
- 生成されたタスクの優先順位
- タスク間の連絡

② タスクの消滅

タスクは正常に消滅する場合と，異常に消滅する場合がある。異常な消滅は，ハードウェア・トラブル，プログラム・ロジックの誤りなどに起因する。ここでは，特にタスクの異常消滅に伴なう

制御プログラムの仕事について説明する。

・タスクの実行のために用意したシステム資源の開放

・タスクがシステムに要求していた仕事の放棄

・リスタート機能

また，タスク消滅のためのDETACHマクロ，ABENDマクロ命令の機能を説明する。

(4) タスクの同期

① イベント制御

非同期処理にて，自らの制御下にない事象を**イベント**といい，イベントを中心に非同期制御を行なうのがイベント制御方式である。ここでは，イベントの概念とそのマクロ機能（WAITマクロ，POSTマクロ，ATTACHマクロ）について説明する。

② ファシリティ制御

ファシリティをタスクでリザーブしたいときに使用される同期サービスでファシリティは待ち行列管理のもとで遂行される。ここでは，ファシリティの概念とそのマクロ機能（ENQマクロ，DEQマクロ）について説明する。また，デッドロック問題について考慮することも必要なことである。

4.3 演算装置割付

演算装置は，コンピュータ・システムの一つの資源であり，それも最も重要な資源である。ここでは，演算装置（CPU）をタスクに割付するアルゴリズムについて説明する。このとき，システムで考慮されることは，処理の多重度をいかに高め，スループットを向上させるかという問題と，ユーザのターン・アラウンド・タイム の要求にいかにこたえるかが問題である。この問題意識のもとで，次の事柄について説明する。

① ディスパッチング基本アルゴリズム

・ラウンド・ロビン方式

・リニア方式

② タスクの優先処理

・ジョブ制御言語機能

・マクロ機能

③ タイム・スライス制御

4.4 メイン・メモリ管理

4.3の演算装置と同様に，メイン・メモリもコンピュータ・システムの主要な資源である。ここでは，次の各事柄について説明する。

① メモリ保護

② メイン・メモリ割当の自由度

③ フラグメンテーション

④ ダイナミック・リロケーション

⑤ **スワッピング**

⑥ **ロール・イン／ロール・アウト**

⑦ トランジェント・ゾーン管理

⑧ メモリの確保／解放命令機能

ただし，ページング，セグメンテーション機能については，第8章で紹介するが，基本的な考え方は，対比の意味で紹介してもよい。

4.5 時刻管理機能

オペレーティング・システムにおける時間および時刻の管理機能について次の事柄を説明する。

- 日付および現時刻管理
- 時間管理
- CPUアカウンティング機能

(1) 現時刻の管理

システム・イニシャリゼーション時に指定された時刻に従って現時刻を管理する機能について説明する。

(2) 時間管理

次の2種の時間管理のサービスについて説明する。

- ジョブ，ジョブ・ステップの実行時間管理
- ジョブ・ステップ内の時間管理

① 実行時間管理

指定された時間（これは，ジョブ制御言語で行なわれる）が経過したら強制的にそのジョブの実行を停止させる機能サービスについて説明する。

② ジョブ・ステップ内の時間管理

プログラマがプログラム上で必要となる時間管理機能は，次のマクロ命令機能としてサービスされている。このマクロ命令の機能と使用法について説明する。

- TIME　マクロ命令
- STIME　マクロ命令
- TTIME　マクロ命令
- XTIME　マクロ命令

③ CPUアカウンティング管理

ジョブ（ジョブ・ステップ）が，そのジョブ（ジョブ・ステップ）の実行のために費やした時間を計測する機能である。CPUアカウンティング管理機能について説明する。

指導上の留意点

(1) コンピュータ・システムでの仕事の制御の仕方には，システムによって種々な考え方が導入されて

いる。大きく分類すると，

① モノ・プログラミング・システム（シングル・プログラミング・システム）

② マルチ・プログラミング・システム

があるが，この立場をはっきりと認識して紹介しないと研修生は，混乱をきたすだろう。

(2) また，マルチ・プログラミング・システムの中でもタスクを静的に取扱かうものと，動的に取扱かうものとがある。

(3) 仮想記憶まで含めたメモリ管理は，第8章で紹介される。

第5章　データ管理

用　語　この章では次の用語を教える。

フィジカル・レコード，ロジカル・レコード，エクステント，デバイス・インディペンデンス
アクセス・モード，処理モード，レコード形式，ファイル編成，順次ファイル編成，順次区分ファイル編成，索引順次ファイル編成，直接ファイル編成，索引ファイル編成，パスワード，　乱ファイル編成，リファレンス・モード，

目　標

われわれが，コンピュータ・システムで仕事を行なうとき，色々な問題がある。なかでも，特に困難な問題として2次記憶の情報の取扱いがある。ここでは，2次記憶上でのデータの記録形式，2次記憶上でのファイルの管理，また，2次記憶とプログラムとのデータの受け渡しなど，総合的にデータを管理しているオペレーティング・システムのデータ管理プログラムについて，その機能の習得を図る。

内　容

5.1　2次記憶媒体

ここでは，基本的事項として2次記憶媒体について知ることを目的とする。最初に2次記憶装置の特性を説明する。

① ボリューム／デバイスの種類，物理的構造
② 特　性
③ 能力（容量，アクセス・タイム）
④ 記録形式
- 各2次記憶のデータの記録
- 磁気テープのＩＲＧ
- ディスクのフィジカル・レコード形式（インデックス・マーク，ホーム・アドレス，レコード0など）

また，次にオペレーティング・システムの磁気テープ，磁気ディスクのボリューム規約について説明する。

① ボリューム構造
② ＶＴＯＣ形式

5.2 データ管理の概念

ここでは，次の基本的事項の説明を行なう。

① データ

② ファイル，ファイル名

③ ボリューム

④ ラベル（ボリューム・ラベル，ファイル・ラベル）

⑤ レコード（ロジカル・レコード，フィジカル・レコード）

⑥ ブロック

⑦ エクステント

5.3 ファイル制御

オペレーティング・システムは，データを2次記憶上で，ファイルを基本的単位として管理している。（なお，ディスク装置では，ファイルは，いくつかのエクステントで構成される。）ここでは，このファイルがいかに2次記憶媒体上で管理されているか，その基本事項について以下のものを習得させる。

(1) ファイルと記憶媒体

2次記憶媒体上のスペースの割当，スペースの解除，スペースの拡張割当，未使用スペースの開放の機能について説明する。

(2) パーマネント・ファイル，テンポラリィ・ファイルの取扱い。

ユーザ・アプリケーションで使用されるファイルには，パーマネント・ファイル，テンポラリィ・ファイルの分類があり，その選定はユーザにある。

テンポラリィ・ファイルとは，ジョブ・ステップが動作しているときに必要とされるが，ジョブ・ステップが終了したときには消滅出来るファイルである。

ユーザが，このファイル特性を考えてファイルの管理をオペレーティング・システムにゆだねることは，オペレーティング・システムのファイル管理の立場から重要である。

(3) ファイル・カタログ機能

昨今，コンピュータ化が進展するにつれて，増大するデータの管理は，大変な仕事となってきている。ここでは，オペレーティング・システムによってファイルが入っているボリューム・ネームを記憶することによって，ファイルの位置づけ，ファイルの分類，ファイルの更新管理を能率よく制御するファイル・カタログ機能について説明する。

(4) ファイル保護機能

(5) ファイル機密保持

(6) ファイルの再編成

データは，一般に流動する。すなわち情報の追加，削除，更新などの処理が加えられる毎にファイルは形が変化する。これは，2次記憶上の効率を低下させる原因ともなり，また，ひいては中央処理装置のアクセス効率の低下にもつながる。この点から，この機能の必要性について説明する。

(7) ファイルの再配置

ファイルも，また流動的である。ファイルはボリューム上で生成されたり削除される。そのたびにボリューム上のスペースが細分化され，使用効率の低下となる。また，ファイルは，その使用の頻度によって媒体が選択される場合がある。この機能とプログラムの保証（デバイス・インディペンデンス）について説明する。

(8) デバイス・インディペンデンス

ここでは，**デバイス・インディペンデンス**の概念，利益について説明する。

5.4 アクセス手法

前節までで，「データが記憶される媒体について」，「論理的に関連のあるデータは，ファイルとしてまとめられ，そのファイルがいかに制御されるかについて」説明してきた。そこで，ここでは，データの取扱いについて説明することにする。

データの取扱いは，オペレーティング・システムの対象ではなく，すべてがユーザ・プログラムにその責任がある。しかしオペレーティング・システムは，ユーザ・プログラムからのデータの読込要求，またデータの書込要求に対して便宜を提供している。これは，

- ユーザのアプリケーションに適応する組織化されたデータ構造
- そのデータ構造に基づいたアクセス・マクロ

機能の提供の責任である。

ここで，次の各事柄について説明する。

(1) アクセス・モード

アクセス・モードをファイル編成と対応して説明する。また，次のアクセスについて，その概要を理解させる。

- 順アクセス
- 乱アクセス
- 移動アクセス

(2) 処理モード

処理モードをファイル編成と対応して説明する。また「入力」，「出力」，「入出力」，「更新」，「追加」の各処理モードの定義を行なう。

(3) レコード形式

ここでは，**レコード形式**について，その代表的な五つの形式について説明する。

- 固定長非ブロック化
- 固定長ブロック化
- 可変長非ブロック化
- 可変長ブロック化
- 不定長

(4) ファイル編成とアクセス言語

① ファイル編成

ここでは，**ファイル編成**，名前の紹介およびファイル構造について説明する。また，お互いのファイル編成の長所，欠点を明確にする。

さらに，このことから最終的には，ファイル編成を選定する基準を研修生が認識できるようにする。い。

- **順ファイル編成**
- **順区分ファイル編成**
- **索引順ファイル編成**
- **直接ファイル編成**
- **索引ファイル編成**
- **乱ファイル編成**

② アクセス言語

ここでは，アセンブリ言語で提供するマクロ命令を説明する。特にファイル編成と処理モードを関連づけた説明をするとよい。さらに高水準言語（COBOL　FORTRANなど）の言語についても説明する。しかし，いずれも詳細は，他のプログラミング言語に関する科目で説明されるので，ここでは体系的にその概要を把握させればよい。

(5) ファイル定義とファイルの入出力手続き

ファイル定義は，ファイル名，ブロック・サイズ，レコード・サイズ，**パスワード**などファイルの属性を記述することである。ファイルは，プログラムでＦＤマクロで定義され，ファイルの入出力を行なうためにＦＣＢ（File control block）　が作られる。このＦＣＢは，さらにファイルのディレクトリ，ジョブ制御言語のＤＥＦＩＮＥ文で完成され，プログラムはこのＦＣＢを介してファイルと論理的に結びつく。ここでは，このＦＣＢまわりの機能，とくにＦＣＢの属性値の決定場所について説明することは重要である。また，ファイルの入出力の手続きの流れを通して，今までの要約を行なう。

5.5 入出力制御

今までは，ファイルとかデータとかを論理的に取扱つて来た。そこで，ここでは，実際にプログラムから要求されたデータの読取，書込などの要求が，いかにオペレーティング・システムの制御をうけて遂行されるかについて把握させる。

(1) 入出力機器制御

ここでは特に，次の２点について説明する。

① 入出力命令の待ち行列管理と実行

待ち行列管理の方法は，色々あるが，現在，入出力命令の待行列管理として一般的に行なわれている次の方法について説明する。

・FIFO (First in First Out)

・タスクの優先度に基づいたスケジューリング

・デバイスに優先度を設定したスケジューリング

・デバイス特性に基づいたスケジューリング

② エラー処理

データ管理のエラーには，フィジカルなエラーとロジカルなエラーがある。チャネル装置，デバイス装置など機械的な故障によるフィジカル・エラー，ロジックの誤りによるロジカル・エラーについての認識と，オペレーティング・システムの処理方法について説明する。

フィジカル・エラーに対して，オペレーティング・システムは，オペレータにその旨メッセージの通達を行ない，オペレータ・アクションを要求する。これに対して次の機能を説明する。

・該当プログラムの中断指示

・再試行指示

・エラーの無視

・エラー・レコードの無視

また，ロジカル・エラーに対して，データ管理は，その旨メッセージを伴なったユーザの出口を用意する。

ここでは，ロジカル・エラーが生じる原因とその対処法について説明する。

(2) 論理入出力制御

ここでは，次の各事項について説明する。

① ブロッキング／デブロッキング

2次記憶のデータは，ブロックが処理単位である。また，一方ユーザ・プログラムの処理単位は，ロジカル・レコードである。一般に，ブロックは，n 個のロジカル・レコードで構成される。

ここでは，ブロックからロジカル・レコードの分解，あるいは，ロジカル・レコードからブロックへの組立の制御を行なうブロッキング／デブロッキング機能について説明する。

② バッファ・コントロール

バッファの定義，バッファの割当などの取扱いについて説明する。また順処理のダブル・バッファ・コントロールの効果について説明する。

③ **リファレンス・モード**

「移動 (MOVE)」，「位置 (LOCATE)」，「交替 (SUBSTITUTE)」の各モードについて，その機能と特徴についてブロッキング／デブロッキング機能と関連づけて説明する。

④ 共用制御

ディスク上のファイルなど，ダイレクト・アクセス・ファイルは，そのデータをアドレスを通して管理出来る。

従って，必要ならば，1個のファイルをいくつかのプログラムに同時に割当て，お互いのプログラムで使用することが可能である。しかし逆に，データの保証についても考慮せねばならない。すなわち，共用して使用する機能と共に，ある一定期間，一つのプログラムが占有して使用できる機能も

必要である。このために用意されているマクロ命令の機能，「その使用法」，処理モードの関係からファイルの共用の範囲について説明する。またお互いのプログラムでお互いにいくつかのファイルを共用制御しながら処理する場合に生じるデッド・ロックの問題について，説明する。

⑤ その他

- ファイル・ラベル・コントロール
- エンド・オブ・ファイル・コントロール
- エンド・オブ・リール・コントロール

について説明する。

5.6 データ管理のユーティリティ

ここでは，データ管理プログラムの機能を補助するユーティリティ・プログラムについて，その種類と機能を説明する。

① ファイルの複写

同一ファイル編成からなるファイルの作成

② ファイルの退避／復元

大記憶ボリューム上のファイルの複写および複製

③ ファイルの照合

④ ファイル内容の印字

⑤ 文字列の変更

ファイルのある一部個所のバッチ処理

⑥ ボリュームの複写

⑦ ボリュームの退避／復元

⑧ ボリュームの比較

⑨ ボリュームの内容の印字

⑩ ボリュームの初期設定

- ホーム・アドレスおよび，トラック記述レコードを各トラックに書く。
- 不良トラックの検査と代替トラックの用意
- ボリューム・ラベルの書込
- イニシャル・プログラム・ローダの生成
- VTOCの作成

⑪ エクステントの再配置

⑫ ボリュームの使用状況の製表

- ボリューム・ラベル名
- ファイル名とエクステント情報
- 未使用エクステント情報
- 未使用スペース量

⑬ スペース割当／割当解除

⑭ ファイルからのレコードの抽出

⑮ ファイルの再編成

ファイルの中の論理的な削除レコードを物理的に削除，また，オーバフロー・レコードの削除

⑯ 索引ファイルの2次インデックスの作成

指導上の留意点

(1) この章では，ファイル編成の選定，媒体の選定についての観点がうえつけられる能力をも養なうこと。

(2) データ・ベース問題については，ここで紹介する必要はない。

第 6 章　通信管理

用　語　この章では次の用語を教える。

コンテンション方式，ポーリング方式，メッセージ制御プログラム，回線制御プログラム，メッセージ処理プログラム，ビフォア・イメージ，アフタ・イメージ，サイクリック・ダンプ

目　標

昨今のオペレーティング・システムは，汎用システムの名のもとにバッチ処理，リアルタイム処理，タイムシェアリング処理など種々の処理形態に適用出来る能力を持っている。ここで，紹介を加えようとしている通信管理機能は，リアルタイム・システムを具体化するために用意されたオペレーティング・システムの管理プログラムである。しかしながら，リアルタイム・システムは，バッチ処理にみられるシステムの汎用性はなく，ほとんどオーダ・メイド的に各システムごとに開発されているのが現状である。メーカもオンライン用のモニタとかオンライン用パッケージを開発するなどオンライン・システムに大きく力をそそいではいるが，オンライン・システムの各種のニーズに応えるべき所までは未だ達していない感がある。そこで，ここでは，オンライン・システムからみて，オペレーティング・システムの通信管理機能を位置づけ説明する。

内　容

6.1　オンライン・リアルタイム・システムの目的と基本

(1) オンライン・リアルタイム・システムの目的

コンピュータによるデータ処理の目的，バッチ処理システムの欠陥などと対比し，オンライン・リアルタイム・システムの持つ目的について説明する。

(2) オンライン・リアルタイム・システムの要求条件

オンライン・リアルタイム・システムの要求される条件を通して，オンライン・リアルタイム・システムの特徴を説明する。

① 応答時間

② ランダムな処理

③ 多数の入出力

④ 信頼性（故障の少ないシステム）

⑤ 人間・機械間のスムーズなインターフェイス

(3) リアルタイム・システムの種類

リアルタイム・システムで処理されるべき情報が発生したとき，（この情報をメッセージとかトラ

① データ収集システム

② データ分配システム

③ インクワイアリィ（問合せ）システム

④ メッセージ交換システム

6.2 ハードウェア・システムの構成と特性

ここでは，リアルタイム・システムのハードウェア構成要素の紹介とその特性を知ることにより装置の選定の基準についてその概要を説明する。

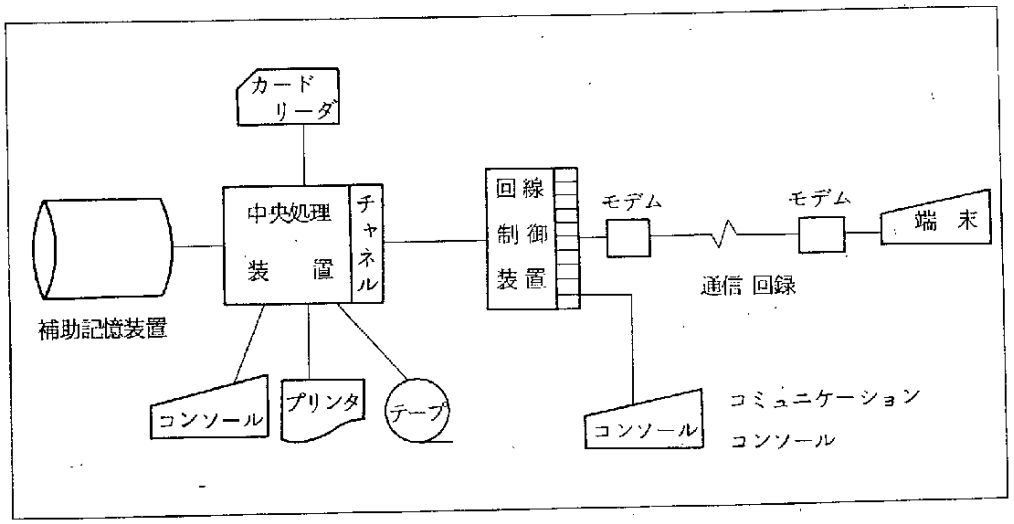


図6－1　リアルタイム・システムのハードウェア構成

(1) 中央処理装置

リアルタイム・システムの心臓部であり，通常のシステムでは，一つまたはそれ以上の中央処理装置を持つ。中央処理装置の使用形態としては次のようなものがあり，その能力，信頼性，拡張性，経済性などの観点から比較説明する。

① 片方が他方のバック・アップ・システムとして使われるもの（Stand by, Duplex）

② Dual システムとして使われるもの

③ 片方が回線制御専用に使われるもの（Master－Slave）

④ 双方で負荷を分担するもの

(2) 補助記憶装置

補助記憶装置としては，ドラム，ディスク，磁気テープ装置が主要なものである。ここでコンピュータ•システム設計上の問題となる点として，中央処理装置との接続，記憶容量，アクセス・タイム，データの移送速度，価格がある。この点に注目して整理を行なう。

(3) 通信回線

これは一般に，日本電信電話公社提供のデータ伝送サービスが利用される。この詳細な説明は，別の科目「データ通信システム」で行なわれるのでここでは簡単に説明するだけでよい。

(4) 端末装置

端末機器は，人間と機械とのインターフェイスであるから，最も効率よく操作が出来ることが望ましい。

(5) 回線制御装置

文字の組立，分解，制御コードによる割込の発生などの機能をもつ回線制御装置について説明する。

(6) その他

- ライン・コンセントレータ
- モデム
- 分岐装置
- システム・コンソール
- 切換装置
- インターバル・タイマー
- 時刻装置
- オンライン・アダプタ

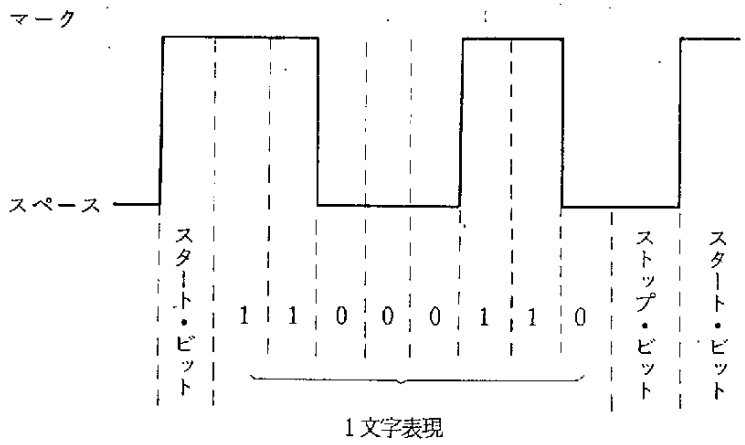
などオンライン・システムを構成する装置の役割について説明する。

6.3 端末機と中央処理装置の通信

ここで紹介されることは，別に詳細な説明が科目「データ通信システム」でなされる。従って，ここでは，通信機能を理解するのに最小限の知識を与えることを目的とする。

(1) データの表現と転送能力

独立同期，調歩同期，スタート・ビット，ストップ・ビット，制御コード，BPSなどの基本用語と機能について，またJIS，ANSCII，ISOなどのコード体系について概要を説明する。



データが伝送回線上を通るとき，文字は二つの電圧の状態に対応するコードで表現され，それぞれスペース，マークと呼ばれる。

図6－2　調歩同期，8単位(例)

(2) メッセージの形式

メッセージの形式について，説明する。しかし，これは定型化されたものではない。そこでたとえば，一例を紹介すると次のようになる。

例1）調歩式

SOH × × × × × … × STX × × × × … … … × ETX

ヘッダ　テキスト

例2）調歩式

STX × × × × … … … … … × ETX

例3）同期式

SYNC SYNC SYNC SYNC STX × × × … … … … … … … × ETX

図6－3　メッセージの形式例

(3) 接続確認

コンテンション方式，ポーリング方式について説明する。

① コンテンション方式

次図の例を通して，接続確認を説明する。

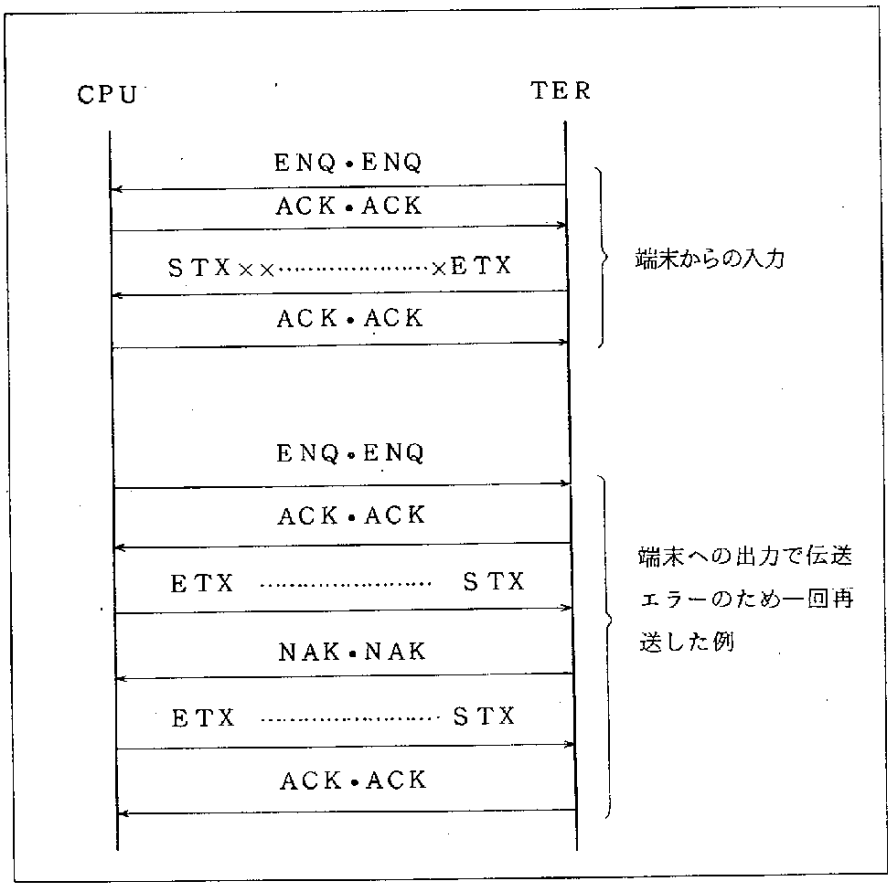


図6－4　コンテンション方式の例

② ポーリング方式

次図の例を通して，接続確認を説明する。

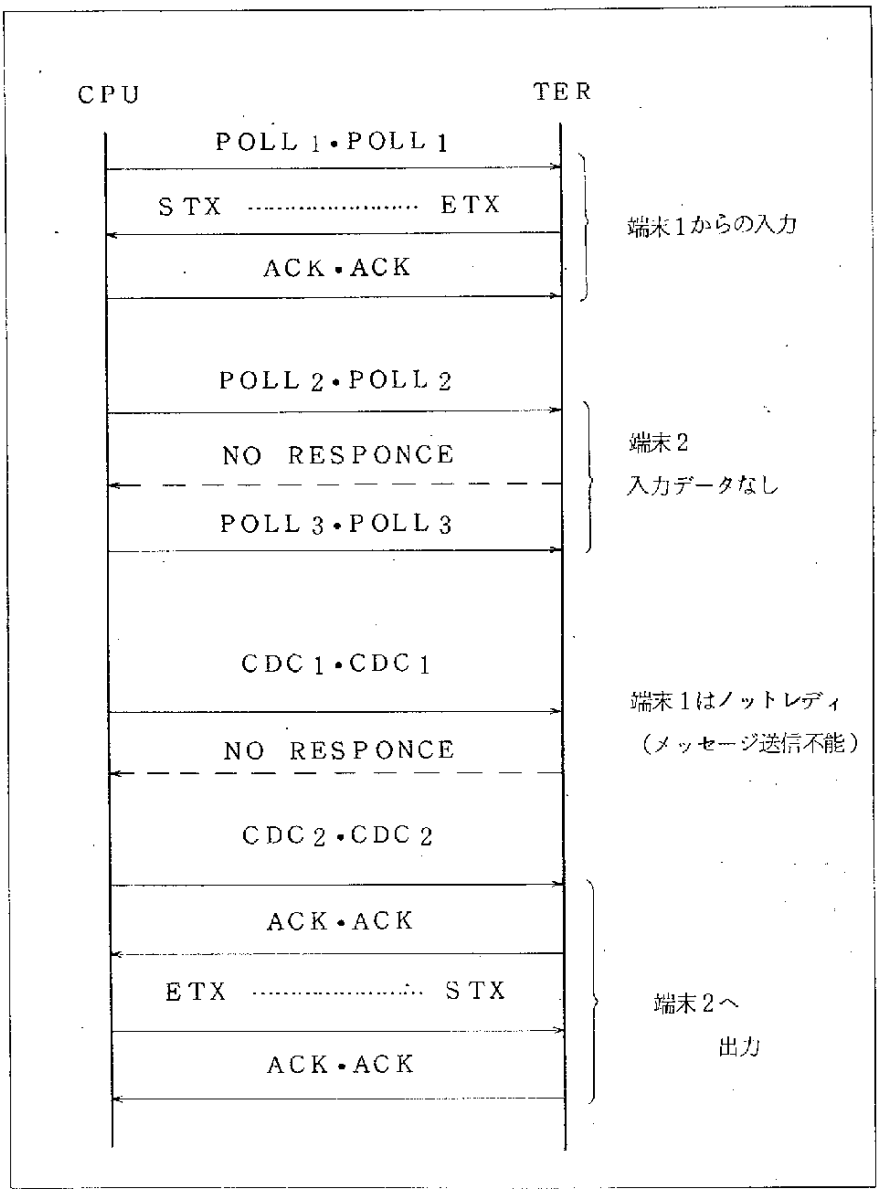


図6－5　ポーリング方式の例

③ 特　徴

コンテンション方式とポーリング方式の相違，特徴を対比説明する。

6.4 リアルタイム・システムのプログラム構成

リアルタイム・システムのプログラムは，その機能面から分類すると次のようになる。

- 管理プログラム
- 処理プログラム
- 支援プログラム

これらについて，その概要を説明する。

(1) 管理プログラム

① オペレーティング・システムの通信機能

- 割込処理と優先度の高いプログラムへの切換
- 中央側の入出力装置制御
- タイマを利用した時間制御
- マス・ストレージ等の割当／開放
- プログラムのロードとコア・エリア管理
- 各種エラーの検出と処理
- トランザクションの待ち行列制御
  メッセージの優先処理
  キューイング・ファイル
- バッファ制御
- 迂回送信機能
- マルチ・プログラミング機能

② **メッセージ制御プログラム**

- インプット・メッセージの識別
- メッセージの処理プログラムの選定
- メッセージのコード変換
- メッセージのフォーマットの作成
- 処理プログラムのロードとコア制御
- メッセージのロギングとリスタート準備
- 処理プログラムの監視
- オペレータとのインタフェース
- 統計資料の作成
- エラーの検出と処理
- リスタート処理

③ **回線制御プログラム**

これは，システムの要求にあった種々の端末機とのデータ交換のために準備されるプログラムで，端末機の種類ごとに必要とされる。

この回線プログラムは，次のような機能を持っている。

- 端末機との接続確認
- 伝送制御コードの判断および追加
- メッセージの形態チェック
- データ・コードの変換
- パリティ等のエラー・チェック
- オペレータへの通信
- 端末機，回線の状況管理

(2) 処理プログラム

実際にメッセージ処理を行なうプログラムで，使用頻度の高いものはメモリーに常駐させ，低いものは必要に応じてマス・ストレージから読み込むように作成される。ここでは，管理プログラムとのインタフェースの問題を中心にメッセージの受け渡しを行なう**メッセージ処理プログラム**について，マクロ機能を通して説明する。

(3) 支援プログラム

支援プログラムとして，代表的な次の各プログラムについて，その概要を必要性，役割から説明する。

① システム診断プログラム

② デバグ用ユーティリティ

③ システム・メインテナンス・プログラム

④ 障害回復プログラム

(4) プログラム構成図

個々のプログラムが，機能分類から紹介されてきたが，ここではオンライン・プログラムの構成図を通して，各プログラムの関連，メッセージの流れを中心に要約する。

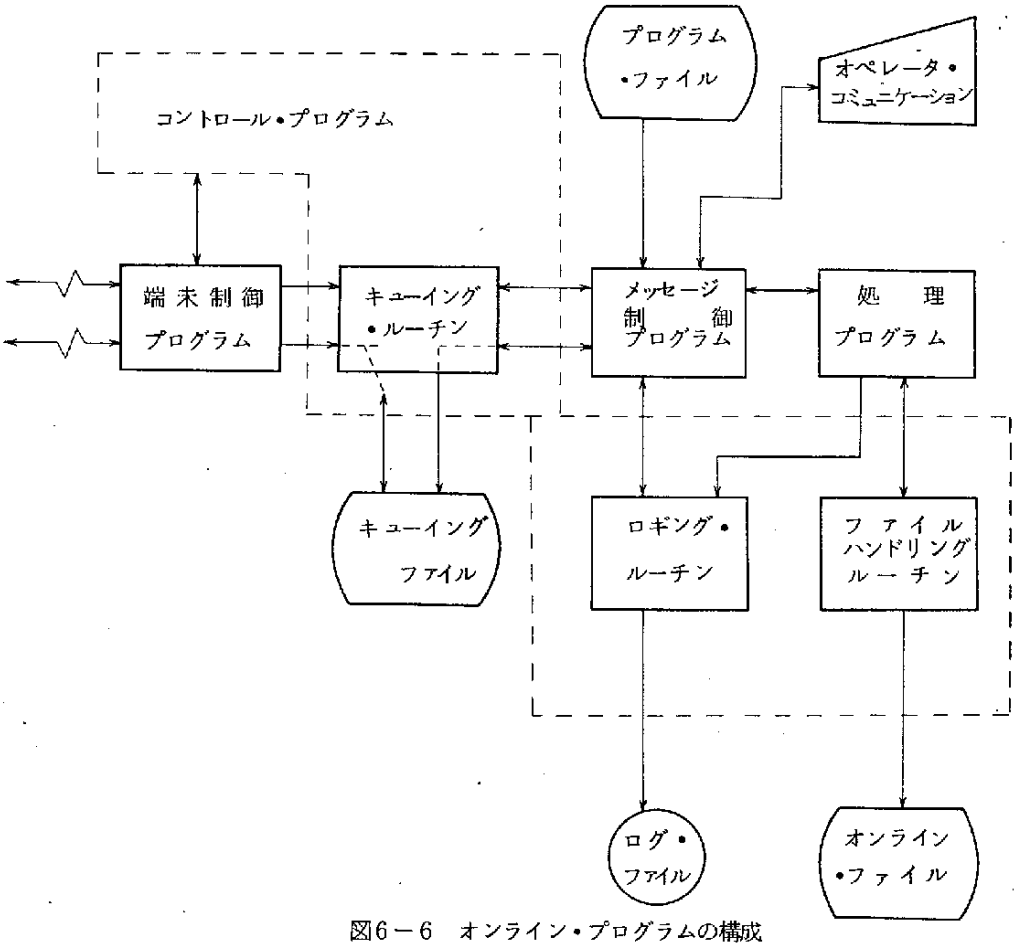


図6－6　オンライン•プログラムの構成

6.5 障害対策

リアルタイム•システムの障害は，極めてその影響が大きく，関係部所のみならず企業全体しいては社会的に影響を引き起す。また反面コスト面からの制約もある。この相矛盾した要素の中で各オンライン•システムが行なう障害対策について説明する。

(1) 障害の種類

① 構成機器の障害

端末機とか周辺機器などのハードウェア機器障害

② 中央処理装置の障害

③ ソフトウェアの障害

④ 運用上の障害

オペレーションとか周辺環境の変化で生じる障害

⑤ システム設計不備にともなう障害

トラフィック確定の不備とか機器能力見積のミスから発生する障害

(2) 障害対策

ここでは，オンラインで一般に行なわれている障害対策について，次の項目を紹介する。

① 中央処理装置の障害対策

中央処理装置は非常に重要な資源である。これはシステムの信頼性の評価のもとで2重化対策がとられる場合がある。

- デュプレックス・システム
- デュアル・システム
- マルチ・プロセッサ・システム

② システム・リスタート

障害によってシステム・ダウンが生じ，システムが停止したときすみやかに復旧が図れるように定期的に行なわれるシステムのチェック・ポイント・ダンプ機能について説明する。

③ ファイルの障害対策

オンライン・ファイルは一般に破壊処理が行なわれる。（ただしファイルの更新はオンライン時に行なわないシステムもある。）従ってこの時，このファイル破壊に対してなされる対策について説明する。

- **ビフォア・イメージ**
- **アフタ・イメージとサイクリック・ダンプ**
- 二重書き（Dual Write）
- ファイルのロック／リリーズ機能

④ メッセージの復旧対策

オンライン・システムの重要なことの一つにシステムがダウンしたとき，端末からの種々のメッセージが失なわれることを防止せねばならない。この対策として次のことを説明する。

- メッセージのログとサイクリック・ダンプ
- 二重書き（Dual Write）

特にこの場合，立ち上がりの時間に対しての対策も考慮されねばならない。この観点からの説明も行なう。

指導上の留意点

(1) オペレーティング・システムの通信管理機能に限定せず，オンライン・リアルタイム・システムとして機能をとらえながら説明する。

(2) 6.4を中心テーマとして説明する。

(3) また，ネットワーク設計，キューイング・ファイル設計，バッファ設計，トラフィック見積りなど興味あるオンライン・システム設計の問題がある。これらの問題についての概要紹介あるいは，動機づけを行なうとよい。

(4) 6.5は，第7章と関連させて説明してもよい。

第7章　障害管理

用語　この章では次の用語を教える。

リスタート，チェック・ポイント・リスタート

目　標

コンピュータ・システムのより高度な利用技術としてのリアルタイム・システムに限らず，従来からの利用形態であるバッチ処理システムにおいても，近年，システム利用度（アベイラビリティ）が非常に大きくとらえられつつもる。

ここでは，ハードウェア，ソフトウェアなど，システム運転中に生じる障害に対して，いかなる制御手段を考慮しているか，その機能を習得する。

内　容

7.1　高利用度の必要性

システムの高い利用度は，信頼性の向上，保守性の向上とともに，いかにシステムの運用を維持するかの問題がある。これらは，次の理由から特にシステムに要求されていることを説明する。

① ＥＤＰ市場からの要求

多重プログラミング，オンライン・リアルタイム・システム，大規模データ・ベースの占める割合は，増々増大しつつある。したがって，これらシステムにおいて高い稼動性が要求されてきている。

② 保守費用

近年，ハードウェアおよびソフトウェアの複雑化ならびに保守技術者の不足に伴ない，従来の保守対策の改善への要求がなされている。

7.2　リスタート能力のサービス

① ＣＰＵ再試行

ハードウェアによって自動的に行なわれるＣＰＵの再試行機能について説明する。

② 入出力操作の再試行

ハードウェア，ソフトウェアによって行なわれる再試行機能について説明する。

③ チェック・ポイントとリスタート

プログラム実行中，ハードウェア障害によってプログラムの遂行が不能になる場合がある。このとき，中央処理装置のロスの防止のためにある定められた時点（これをチェック・ポイントという）

にリスタートのための情報を記憶し，障害が発生したとき，そのチェック・ポイントからの再試行を可能とするチェック・ポイント・リスタート機能について説明する。

④ システム・リスタート

システムの緊急事態により，システムをシャット・ダウンせねばならないときの復旧手段について説明する。

7.3 フェイル・ソフト機能

システムが運転中生じるハードウェア障害によって，システム全体の停止するような回数を出来るだけ制限する機能であるフェイル・ソフト機能について説明する。

7.4 診断ルーチン

システムが障害を起こしたとき重要なことは，その障害をすみやかに検出し，その復旧を図ることである。このために用意されているパッケージ「診断ルーチン」について説明する。

また7.3とも関連することであるが，多重プログラミングの環境の中で正常なジョブの実行と併行してオンライン診断や故障の修復，修復後の受入れテストが出来るように回復手段を用意することの重要性についても説明する。

7.5 障害情報のロギング

後刻，保守システム（予防診断，予期診断）のために重要な情報である障害情報の記録機能について説明する。

指導上の留意点

この章は，詳細な機能説明はさけて，場合によっては用語説明にとどめるぐらいでよいだろう。

第8章　タイムシェアリング・システム

目　標

タイムシェアリング・システムは，従来のバッチ処理と並んでコンピュータの新しい共同利用方式として脚光を浴び，さらに来たるべき情報化社会における強力な道具として活躍するだろうことは，今や明白な事実になりつつある。日本では昭和43年頃からいくつかの大学や企業内でタイムシェアリング・システムが稼動し始め，さらに日本電信電話公社が昭和45年から始めたデータ通信サービスをきっかけとして，わが国のタイムシェアリング・システムもようやく実用化の段階に入ろうとしている。ここでは，このタイムシェアリング・システムについてその概念，基本的な技術を理解させる。

内　容

8.1　タイムシェアリング・システム概念

(1) タイムシェアリング・システムの生いたち

バッチ処理形態のオペレーティング・システムは，コンピュータの遊休時間を少なくし，スループットを向上させる必要性で生まれて来た。これは一般の事務計算のように，プログラムを一度作っておいてあとは定期的にデータを変えて処理すれば間に合うような場合にはより効果的であるが，大学や研究所さらに一般企業の技術者のようにコンピュータを道具として，必要なときに使いたいという人達にとっては何とも不便である。ではどうあったらいいかについて研修生と話し合いを持つとよいだろう。

(2) タイムシェアリングの定義

ここでは，タイムシェアリングの定義を次のように行なう。「複数の利用者が，端末装置を使って，一つのコンピュータを同時に，それぞれの利用者からみれば独占しているような感覚で利用できるシステム」

(3) タイムシェアリング・システムへの要求

次の事柄について紹介するとともに，その技術的な背景についてその概要を説明する。

① マン・マシン相互の通信が容易であること。

② 使いたいときに即座に使えること。

③ 安心して使えること。

④ 複数の利用者が共同で利用出来ること。

⑤ 利用者はプログラムを作り実行できること。

⑥ システムのライブラリが利用出来ること。

⑦ ファイルの作成や更新が行えること。

⑧ ハードウェアの構成を意識しないで使えること。

⑨ 適正なる料金体系がととのっていること。

8.2 タイムシェアリング・システムの基本機能

(1) 時分割

複数のユーザに対して，おのおのが自分で独占してコンピュータを使っているような感覚でコンピュータを利用させる場合，その基本となるＣＰＵタイムの時分割スケジューリングについて説明する。特にバッチ処理スケジューリングと対比して説明するとよりわかり易くなるだろう。

(2) スワッピング

多くの利用者のプログラムをすべて主記憶に常時存在させることは主記憶が高価であること，また主記憶装置の容量からして一般に無理である。このためにプログラムは一般に主記憶と補助記憶との間で受け渡しが行なわれる。このスワッピング機能について説明する。

(3) ページングとセグメンテーション

ページングやセグメンテーションの概念は，ＭＩＴのMULＴＩＣＳシステムで導入された。そして，この技法は，タイムシェアリング・システムの不可欠の要素とみなされているし，また，タイムシェアリング・システム以外の利用形態のシステムの中でもその特徴を利用してとり入れられつつある。このページング，セグメンテーション機能について説明する。

① ページング

ページングは，プログラムのもつ局地性に着目された考え方がハードウェア機能としてとり入れられている。ここでは，次の事柄について説明する。

- ページングの定義
- ページングの効果

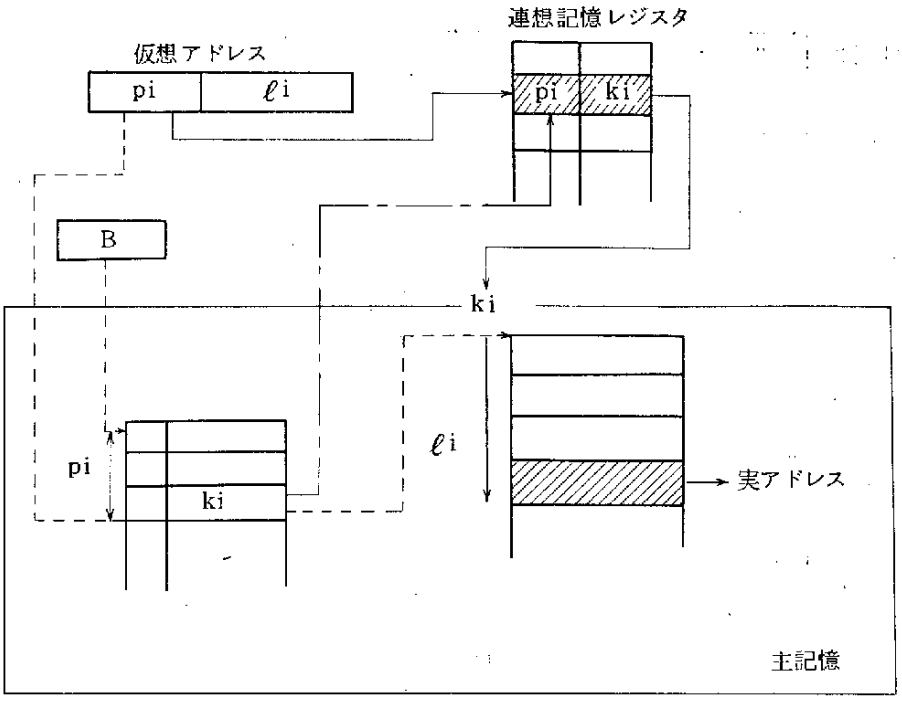
スワッピング・オーバヘッドの減少効果

プログラムはメモリの大きさの制限から解放

メモリ割当管理が容易

② 連想記憶

仮想アドレスは，アドレス変換のためにページ・テーブルを介さねばならないことは，実アドレスをアクセスするために２回のメモリ・アクセスが必要となる。これは処理能力の低下を伴なう。この問題解決のために用いられる連想記憶について説明する（図８－１参照）。



－－－－－ 第１回目のアクセス

―――― 第2回目以後のアクセス

図８－１　連想記憶レジスターによるアドレス変換

③　セグメンテーション

プログラムは，通常一つの主ルーチンといくつかのサブルーチンおよびいくつかのデータの部分から成り立っている。それらのルーチンやデータの部分はセグメントと呼ばれる。二つ以上のプログラムが同一セグメントを共用しようとしたときに，それぞれのプログラムからみれば異なった仮想アドレスを持っていても実際にはただ一つだけ主記憶上にあげれば，共用利用出来るような技法であるセグメンテーションについて説明する（図８－2,８－３参照）。

- セグメントの定義
- セグメンテーションの効果

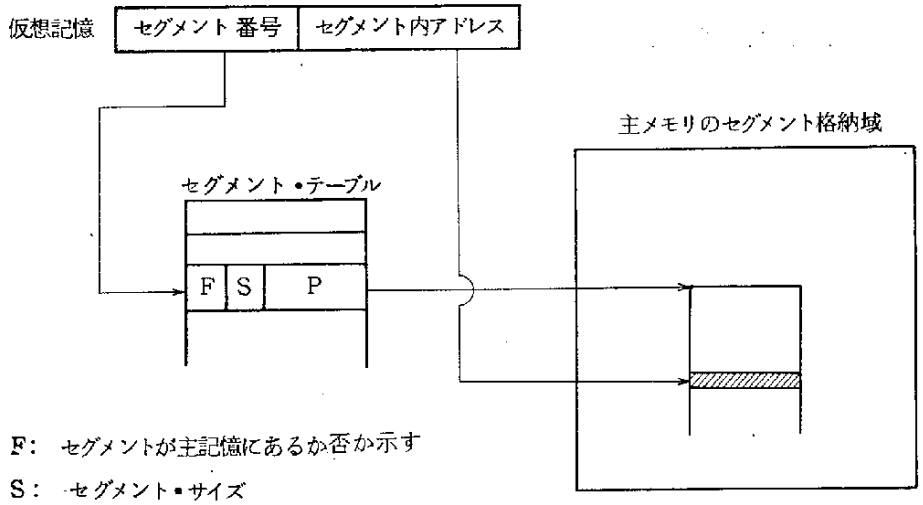
　スワッピング・オーバヘッドの減少効果

　モジュール化基本メカニズムの提供

　きめ細かな記憶保護機能が可能

　モジュール単位での共用能力

- 仮想アドレスと実アドレス



F: セグメントが主記憶にあるか否か示す

S: セグメント・サイズ

P: セグメント開始アドレス

図3－2　2次元アドレス

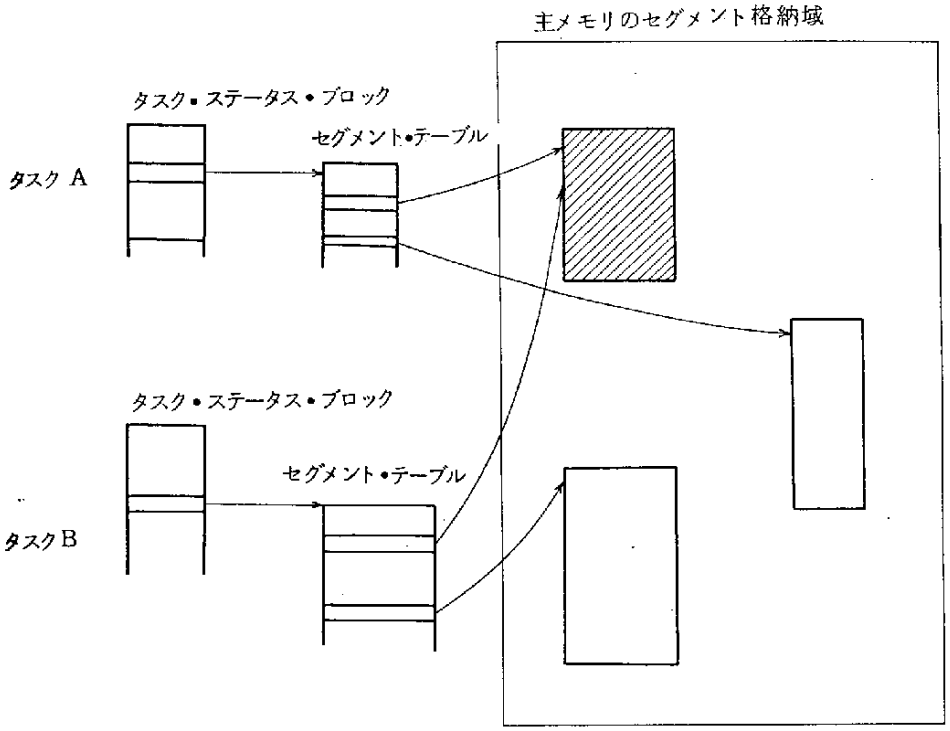


図8－3　タスクのメモリ空間

④ ページング機能を持つセグメンテーション方式

ページング方式，セグメンテーション方式のお互いの長所を併せ持つ方式である。ここでは，MULTICSのアドレス方式を説明する（図8－4参照）。

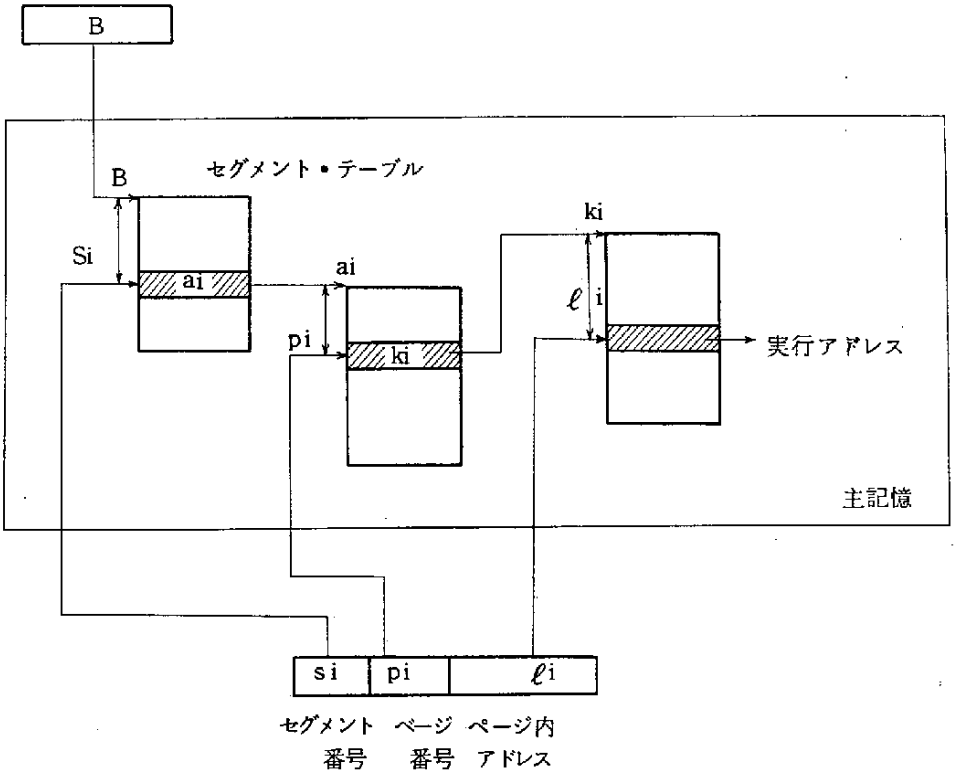


図 8 － 4 マルピクスによるアドレス方式

(4) 基本スケジューリング・アルゴリズム

ユーザのプログラムが実行されるためには，そのプログラムが必要とするリソースの割当が行なわれなければならない。一つの資源の中でCPUタイムの割当を行なうスケジューリング・アルゴリズムについて基本的な図8－5，図8－6のアルゴリズムを説明する。

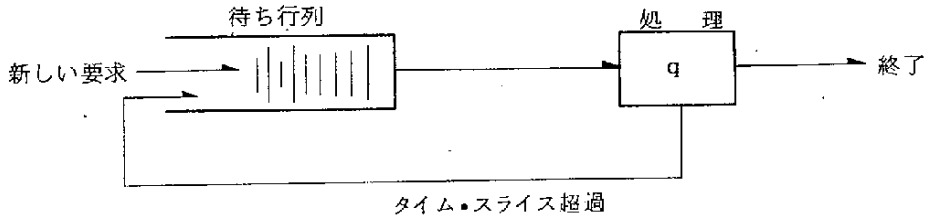


図８－５　ラウンド・ロビン方式

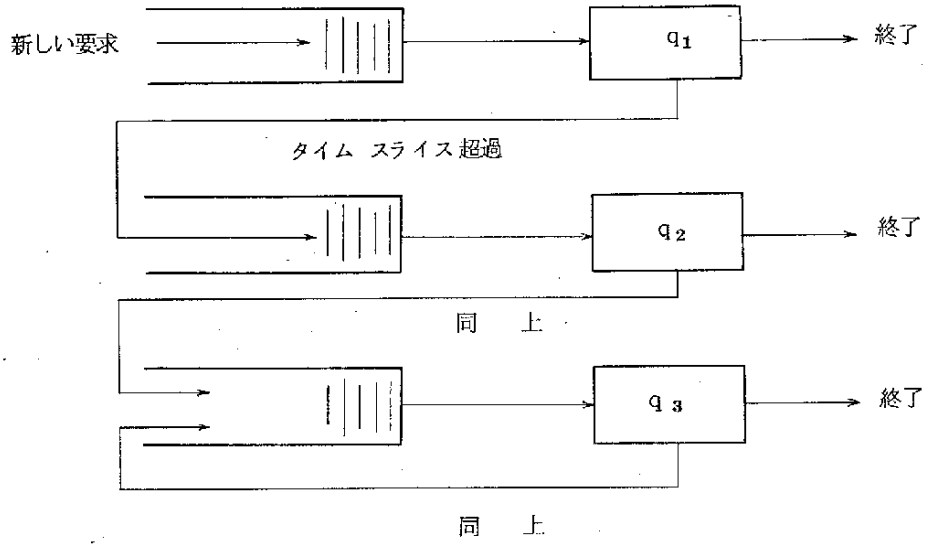


図８－６　多重レベルの優先方式

指導上の留意点

(1) タイムシェアリング・システムは，詳細な機能説明よりも，基本的な機能を中心にアウトラインの習得に心掛ける。

(2) 最近のハードウェアには，ページング，セグメンテーション機能が取り入れられつつある。これを機能特徴を中心に説明する。

参考文献

(1) 上条史彦著，「オペレーティング・システム入門」 日本生産性本部　昭和44年

(2) 萩原 宏監，前川 守著，「オペレーティング・システム」 日本評論社　昭和47年

(3) 藤井 純著，「オペレーティング・システム」 産業図書　昭和46年

(4) 新田謙治郎，高山 田著，「タイム シェアリング・システム」 富士学院　昭和46年

(5) コンピュータ・マネージメント社編 「リアルタイム・システムの設計」

言　語　プ　ロ　セ　サ

目　　次

第1章　言語の記述 …… 444
　1.1　シンタックスとセマンティックス …… 444
　1.2　文脈自由文法 …… 446
　1.3　BNF …… 446
　1.4　演習問題 …… 447
第2章　基礎技法 …… 449
　2.1　トリー（木） …… 449
　2.2　表操作 …… 450
　2.3　演習問題 …… 451
第3章　プロセサ …… 452
　3.1　トランスレータ …… 452
　3.2　インタプリタ …… 453
　3.3　ジェネレータ …… 453
　3.4　演習問題 …… 454
第4章　コンパイラの講成 …… 455
　4.1　総　論 …… 455
　4.2　スキャンナ …… 458
　4.3　パーサ …… 460
　4.4　記号表 …… 462
　4.5　セマンティックス解析 …… 463
　4.6　コード生成前処理 …… 463
　4.7　コード生成 …… 465
　4.8　デバグ・サポート …… 465
　4.9　その他 …… 466
第5章　設計演習 …… 467
　5.1　アセンブラ …… 467
　5.2　コンパイラ …… 467

科目「言語プロセサ」

教育の目標

コンピュータ用の各種プログラミング言語の基礎となっている文法構造と，その言語を処理する言語プロセサの構造について理解させる。この目標を達成するために次の手順で説明する。

まず数多く存在するプログラミング言語を統一した見地から眺められるように言語の記述についての一般理論について理解させる。

次に，言語プロセサでよく使用される諸技法についての予備知識を与える。

言語プロセサとしてはどんなものがあるのか，ここで学ぶことがどの分野に役立つのかということを理解させた後，言語プロセサの典型的な例としてコンパイラを採り，その内部構成，種々のオプションについて理解させる。

最後に，簡単な例題によって実際に言語プロセサの一部を設計させる。

本科目を履修することにより，直ちに言語プロセサを設計，作成する能力がつくという水準までには至らないが，ユーザとして言語プロセサを使いこなすための十分な知識と洞察力を修得させる。またより高度な技術修得のための方法論を身につけさせる。

時間配分

章	履習時間（時間）	
	講　　義	演　　習
1. 言語の記述	4	2
2. 基礎技法	4	1
3. プロセサ	4	2
4. コンパイラの構成	23	4
5. 設計演習	1	15
合　　計	36	24

第 1 章　言語の記述

用 語　この章では次の用語を教える。

シンタックス，セマンティックス，シンタックス・トリー，メタ言語，導出，メタ記号，シンタックスのあいまいさ，形式言語，チョムスキーの分類，文脈自由文法，BNF

目 標

世の中でプログラミング言語として認められているものだけで数百種類は下らないといわれ，これに個々の企業，組織体で使用している各種の言語を加えるとその数は数十倍，数百倍になるであろう。それに伴って言語プロセサの数も多く存在する。さらに種々の要求に応じての新規言語の開発も盛んである。

このように言語には多くの種類が存在するが，これらに共通して理論的な見方をすることが可能である。この見方を身につけることにより，言語を超越した言語に対しての洞察力を持ってもらうことが本章の目標である。

まず言語のシンタックス，セマンティックスを中心に形式言語理論からのものの見方ができるようにさせるとともに，現在のプログラミング言語で頻繁に利用される文脈自由文法について，他の文法との位置づけを理解させる。

次に，文脈自由文法の表現によく利用されるBNF（Backus －Normal Formal）について，その詳細を理解させる。

内 容

1.1 シンタックスとセマンティックス

一つの言語を定めるものに**シンタックス**と**セマンティックス**とがあることを説明する。

例としてごく普通の英文をとり，その文法的な分解を示すことにより，シンタックスというものが日常の言語にも見られることを説明する。

図1－1　を用いて**シンタックス・トリー**を説明する。

なお，以下の説明では現在よく研究されているシンタックスについて主に話を進めることとし，セマンティックスの記述についてはいろいろの試みがあるが，まだ実用の段階ではないことを説明する。

言語のシンタックスを記述する言語として**メタ言語**が必要な理由を説明する。

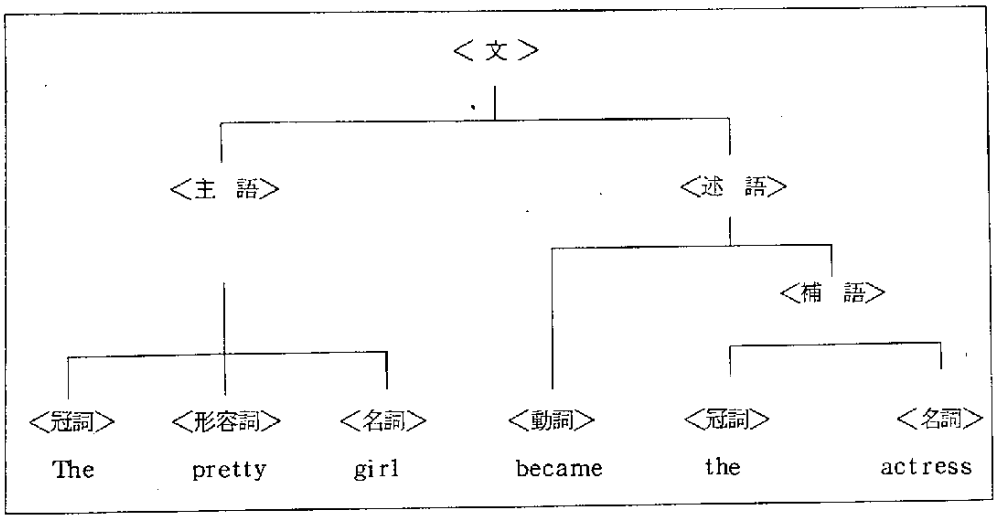


図１－１　シンタックス・トリー

図１－１　の文を**導出** (deriuation)する文法（シンタックスの記述）として図１－２のような図を利用してメタ言語の説明をする。

＜　文　＞ :: ＝ ＜主語＞ ＜述語＞
＜主　語＞ :: ＝ ＜冠詞＞ ＜形容詞＞ ＜名詞＞
＜述　語＞ :: ＝ ＜動詞＞ ＜補語＞
＜冠　詞＞ :: ＝ the
＜形容詞＞ :: ＝ pretty
＜名　詞＞ :: ＝ girl
＜動　詞＞ :: ＝ became
＜補　語＞ :: ＝ ＜冠詞＞ ＜名詞＞
＜名　詞＞ :: ＝ actress

図１－２　文法の一例

以下の事柄について説明する。

① メタ言語の一部としてのメタ変数，**メタ記号**の役割

② 図１－２に示したような文の生成規則に従っての具体的な文の導出とシンタックス・トリーとの関係（参考文献(1)の 2.1 参照）

③ 生成規則とそれによって定まる文法の定義（参考文献(1)の 2.3 参照）

④ 非終端記号と終端記号の別（参考文献(1)の 2.3 参照）

⑤ 導出の正確な定義（参考文献(1)の 2.3 参照）

⑥ 文の正確な定義（参考文献(1)の 2.3 参照）

⑦ ある文法の下での言語の定義（参考文献(1)の 2.3 参照）

⑧ **シンタックスのあいまいさ**（参考文献(1)の2.4参照）

⑨ あいまいさのある文法とない文法の例（参考文献(1)の2.4参照）

1.2 文脈自由文法

前節で述べた**形式言語**の方法を用いて**チョムスキーの分類**について説明する。（図1－3参照）

type 0 の文法（句構造文法）の生成規則の型

$u ::= v$　ここで $u \in V^+$, $u \in V^*$

type 1 の文法（文脈規定文法）の生成規則の型

$xUy ::= xuy$　ここで $U \in V-T$, $x \in V^*$, $y \in V^*$, $u \in V^+$

type 2 の文法（文脈自由文法）の生成規則の型

$U ::= u$　ここで $U \in V-T$, $u \in V^*$

type 3 の文法（正規文法）の生成規則の型

$U ::= N$　ここで $N \in T$, $U \in V-T$, $W \in V-T$

$U ::= WN$

ここで

V ＝文法で使用する記号の集合

T ＝終端記号の集合

$V^+ = \bigcup_{n=1}^{\infty} V^n$ ただし $V^n = \{ x_1 \ x_2 \cdots x_n \mid x_i \in V \}$

$V^* = V^+ \cup \{ \varepsilon \}$　ただし ε は空記号列

図1.3　チョムスキーの分類（参考文献(1)の2.10参照）

文脈自由文法が，type 1の文法とどこが異るかを説明し，type 1はtype 0より，type 2はtype 1より，type 2は type 3よりそれぞれ制限がきつくなっていることを説明する。

1.3 BNF（Backus －Normal Form）

メタ言語の一種であるBNFについて説明する。（参考文献(1)2.3参照）

また，BNFを若干拡張することによって使用法が便利になる次の記号法について説明する。（図1－4参照）

① 中かっこ｛　｝によって，指定最大回数あるいは不定回数の記号列の繰返しを表現する方法

② 角かっこ〔　〕によって，省略可能な記号列を表現する方法

{　}の用法

BNFで

$$<\text{list}> ::= <x> \mid <\text{list}> <x>$$

によって定義される <list> は

$$<\text{list}> ::= <x> \{<x>\}$$

と等価である。（不定回の繰返しの指定）

BNFで

$$\begin{cases} <\text{list}> ::= \varepsilon \\ <\text{list}> ::= <x> \\ <\text{list}> ::= <x> <x> \\ \quad\vdots \\ <\text{list}> ::= \underbrace{<x> <x> \cdots\cdots <x>}_{<x>\ \text{のn回の繰返し}} \end{cases}$$

によって定義される<list> は

$$<\text{list}> ::\ \{<x>\}\ n$$

と等価である。

〔　〕の用法

〔<x>〕は {<x>} 1　（<x>の　0回あるいは1回の繰返し）と等価である。

図 1－4　{　}と〔　〕の用法

1.4 演習問題

(a) 図1－1にならって，たとえば I am a boy というような英語の文を生成するようなシンタックスを考えさせる。

(b) FORTRANで使う名前，数値定数などを生成するシンタックスをBNFあるいはその拡張で表現させる。

(c) 一定の文法を与えておいて，それに対し，何個かの文を勝手に与え，対応するシンタックス・トリーを完成させる。

(d) ジョブ制御言語，アセンブラ言語等の文法をBNFで記述させる。

(e) 図1－5で示す文法に存在する3種の不合理さ①同じ記号を同じ記号で置き換えるムダ，②どこでも使われない記号の存在③終端記号からのみ成る記号列を導出できない非終端記号の存在を説明する。

Z :: = <a><b>e

<a> :: = <a> ①の不合理

<b> :: = <c> | <d> ③の不合理

<d> :: = f <d> ③の不合理

<g> :: = f | f <g> ②の不合理

但し Zはこの文法の初期記号

< > のついている記号は非終端記号

< > のつかない記号は終端記号

図１－５　適当でない文法の例

指導上の留意点

(1) 第１章では，形式言語理論からの物の見方を教える。出現する記号等については，本科目の終りまで同じものを利用するようにして，研修生の混乱をすくなくすること。

(2) チョムスキーの分類の所に出現するV^+，V＊等に関する若干の演算については，説明と演習とを十分に補うこと。

第 2 章　基 礎 技 法

用 語 この章では次の用語を教える。

トリー(木)，バイナリ・トリー，バランスト・トリー，表(テーブル)，リニア・サーチ，バイナリ・サーチ，ハッシング，LIFO，プッシュ・ダウン，ポップ・アップ，FIFO，テーブル・ルック・アップ

目 標

言語プロセサ作成に際しては，種々のプログラミング技法が必要とされる。この章ではそれ等の中でプロセサの各所で使用される比較的一般性のあるテクニックを身につけさせる。

なお，これらのテクニックは，単に言語プロセサだけでなく一般のデータ処理にも広汎に応用できる性質のものである。

内 容

2.1 トリー(木)

トリー(木) は，種々の概念の表現に便利である。言語プロセサに関連するものでも

① シンタックス・トリー（第1章参照）

② 代数式の表現

③ COBOLあるいはPL／Iの構造の表現

④ 記号テーブルの作成（FORTRANプロセサでの例，参考文献1)のP. 224参照）

⑤ コード生成のために用いるケース

などがあることを説明する。

トリーについての次のような基本的な定義あるいは性質について説明する。

① トリー，ルート（root），ノード（node），サブ・トリー（sub tree）

② ノードの位（degree）

③ 葉，枝

④ ノードのレベル（level）

⑤ **バイナリ・トリー**

⑥ **バランスト・トリー**（balanced tree）

⑦ トリーの探索という観点からバランスト・トリーの持つよい性質

2.2 表　操　作

言語プロセサの大部分の操作は，**表（テーブル）**の操作に関連しているといって過言でなかろう。

たとえば，変数，名札，手続き等を記憶する記号表などがすぐ想い浮ぶ。この節ではこのような表の索引方法を中心にして以下の技法を説明する。

(1)　リニア・サーチ

リニア・サーチの方法は最も簡単であるが，以下の技法と比較して有利な場合に用いるべきである。

(2)　バイナリ・サーチ

バイナリ・サーチの場合には，表の要素をあるキーの値に関して上昇順あるいは下降順に並べておく必要がある。表の要素の挿入があるときには不向きである。

(3)　ハッシング

ハッシングは別名，計算記入表（computed entry table）scatter table, randomized table, keytransformation table 等といわれ，色々の変種がある。

以下の諸技法について更に説明する。

① オープン・ハッシュ（Open Hash）（参考文献(2)のP. 20参照）

② オーバフロー・ハッシュ（Overflow Hash）（参考文献(2)のP. 24参照）

③ オーバフロー域でチェイニングする方法（Overflow with chaining）（参考文献(2)のP. 24参照）

④ オーバフロー域なしでチェイニングする方法（参考文献(2)のP. 25参照）

⑤ ハッシングを扱うときには避けようのない同義語の発生，要素の削除等を制御する方法

⑥ またキーから表中のアドレスへの変換方法としてtruncation, folding, division, radix conversion, 2乗採中法等

なお，この変換は表が一次記憶装置にあるときには，サーチに要する時間のかなりの部分を占めることも説明する。

更に表自身がオーバフローしたときの問題についてもリハッシュ（re—hash）などの技法について説明する。（参考文献(2)のP. 21参照）

(4)　その他の方法

次の事柄について説明する。

① スタック

これはLIFO（Fast—in First—ont）のモードで用いる。表に要素を入れるのが**プッシュ・ダウン**で取出すのが，**ポップ・アップ**である。

リカーシブな制御，数式翻訳，ブロック構造の言語は記号表での利用法について説明する。

② キュー

FIFO（First—in First—out）で要素を出し入れするのに用いる。

③ **テーブル・ルック・アップ**

キーとデータ，アドレスとを辞書式に対にする方法

2.3 演習問題

(a) 簡単なハッシングの変換法を与えて，同義語の発生，要素の探索方法，削除方法を考えさせる。

(b) ハッシングで，たとえば要素が削除されることがないとかあるいは新しく追加されることがない等の条件があるとき，どのような方法がそれに適しているかを考えさせる。

(c) リニア・サーチ，バイナリ・サーチ，ハッシングにおいて，要素の探索の頻度，追加の頻度，削除の頻度がどのように効率に影響するか考えさせる。

(d) テーブル・ルック・アップ，ハッシングの手法を使っているとき，キーによって識別されるデータが1次記憶装置にある場合と2次記憶装置にある場合とで影響を受けるファクタを列挙させる。

指導上の留意点

(1) 言語プロセサで使用するプログラミング技法はここにあげたものだけでは十分ではない。他の科目で教える技法もフルに使用するということを強調する。

本来ならば，この科目に含めておきたい内容で他の科目との競合で割愛したものがある。たとえば次の処理についての一応の知識は望ましい。

- 各種リスト処理
- 動的主記憶アロケーション
- サブルーチン
- コールーチン（co－routine）[4]

(2) 上と同様の意味で，実際に大規模言語プロセサを開発する上で他に気を付けねばならない各種の技法，例えばマクロの使用とかダミー・セクションの用い方，ライブラリの利用法，デバグ方法等についても記述が抜けている。

第3章　プロセサ

用語　この章では次の用語を教える。

マクロ・プロセサ，プリコンパイラ

目標

この章では，まず，各種言語プロセサの簡単な説明を行なう。

特殊なプロセサを除いて，その内部構成，構築方法は，第4章で説明するコンパイラで採られている方法が利用できることを教える。

本章の目標は，主にこれらのプロセサのユーザ側から見た外部仕様を理解させることであり，種々のアプリケーションにどのようなタイプの言語プロセサを用意することが適当かという判断を身につけさせる。

内容

3.1　トランスレータ

トランスレータの定義は色々とあるであろうが，ここでは一つの言語あるいはデータをもう一方の言語あるいはデータに変換するプロセサ（与えられた入力情報を処理して何等かの出力情報を生み出すプログラム）と考える。

(1) アセンブラとマクロ・プロセサ

アセンブラにも色々の種類のものがあるが，ここでは対象コンピュータの機械命令に大体1対1で対応して記号で書かれる命令を，オブジェクト・プログラムに変換するものとして説明する。

アセンブラの簡単な動作原理を説明し，中で使用する記号表，パスの回数などについて簡単に触れる。

アセンブラについては別科目「アセンブリ言語」として存在するので詳しくはそちらを参照のこと。

また参考文献(3)にはよくまとまったアセンブラの内部構成の話がある。

アセンブラの機能に入るかもしれないが，**マクロ・プロセサ**も高級な処理をさせるのに適した言語処理能力である。その外部仕様，普通の使い方については，やはり別科目の「アセンブリ言語」中の指導に負うとして，ここでは次のことを説明する。

① 大規模ないしは標準化された作業においてのマクロ利用の利点。

② デバッギング・エイドとしてのマクロ利用

③ 本来のアセンブラ言語仕様の拡張機能としてのマクロの役割

④ nested macro call と nested macro definition の使い方

⑤ プログラムのデータ独立性を高める機能としてのマクロ

(2) コンパイラ

FORTRAN，ALGOL，COBOLなどの高水準言語を論理的にそれと等価なオブジェクト・プログラムに変換するプログラム。

本科目の中心課題であり，引続き第4章でその内部構成に至るまで説明する。

(3) プリコンパイラ

コンパイラがそのオブジェクト・プログラムを機械語ないしはそれに近い水準の言語に変換するのに対し，**プリコンパイラ**はオブジェクト・プログラムが，別のコンパイラの入力となる言語である点が特徴である。

IDS．CICSなどのデータ・ベース・システムに見られるプリコンパイラについて説明する。(本育成指針の他の科目「データベース」参照) PL／Iの compile time facilities も一種のプリコンパイラの例である。

3.2 インタプリタ

インタプリタの定義は広義のものと狭義のものとがある。広義のものは，

① ソース・プログラムを内部的な表現に変換し

② でき上った内部表現のプログラムを逐一解釈し，即実行する

ものであり，狭義のものは上記②だけの機能を持つものをいう。

上記①の性質はコンパイラの機能と考えて，以下では狭義のインタプリタに話を限り以下のような説明をする。

① インタプリタの実行速度は，論理的に等価な処理をする機械語実行速度に比較して遅い。

② 実行時のデバッギング・エイドはインタプリタ方式では楽に組込める。

③ プロセサの作成が比較的容易である。

④ 簡単な数式を扱う言語をインタプリタ方式で作成する時の内部構成

3.3 ジェネレータ

たとえばRPG (Report Program Generator) のように，プログラムの骨組と，そのプログラムを作成するための条件をパラメータとして与えることによって目的のプログラムを作り上げることのできるプログラムである。

ジェネレータの説明としてはRPGタイプのものを説明するのが適当である。

特定目的のために開発されたプログラムないしはユーティリティも，一通りの使い方だけではなく，パラメータの与え方によって，若干ではあるが数種類の使用法があるのが普通である。

このパラメータの与え方を，コンパイラ等と同じように一定の文法あるいは規則に従う言語の形にして与えるのがジェネレータである。

たとえば共通ルーチンの多いプログラム群をひとまとめにして，パッケージとし，適当に定めた入力言語あるいはパラメータによってそれらプログラムに指定されたデータ処理を行なわせる例などを説明する。

コンパイラの構成（第4章で詳述）にならって，これらの入力パラメータの解析方法，指定されたオブジェクト・プログラムの出力方法等について簡単に説明する。

3.4 演習問題

インタプリタ，コンパイラ，ジェネレータ等の利害得失を議論させる。

指導上の留意点

(1) ここで指導上の都合上，言語プロセサをトランスレータ，インタプリタ等と数種に分類ないしは定義づけたが，これらの定義は互いに重複する部分があるかもしれないし，各々の定義が定説と若干かけ離れているかもしれない。問題は，しかし，定義の良し悪しではなく，ここで説明しようとした対象を理解させ，たくみに使用する能力をつけさせることである。

(2) アセンブラ，マクロ機能等については別科目「アセンブリ言語」における育成指針を最大限に活用すること。

第4章　コンパイラの構成

用　語　この章では次の用語を教える。

記号表（シンボリック・テーブル），定数表，スキャンナ，レコグナイザ，トークン，アトム，シラブル，コード生成，メモリ割付，有限オートマトン，遷移図，パーサ，トップ・ダウン法，ボトム・アップ法，オペレータ順位，シンタックス・ディレクテド・コンパイラ，ポリッシュ記法，ドープ・ベクトル，Illifeベクトル，リンカ，TWS，コンパイラ，拡張可能言語

目　標

コンパイラと称される高水準言語のトランスレータに一般に通用する構築法を理解させる。

コンパイラは，ソース・プログラムの読込みに始まり，その構文解析，セマンティックス解析，コード生成，最適化等の一連の類似した処理手続きを有している。

これらの設計方針はもとより一通りではないが，標準的なパターンを主軸として説明することにより，研修生に正統的な方法論を身につけさせる。

さらに最近研究されているTWS（Translator Writing System）の動向などについても触れ，研修生のより深い研究に役立つようにする。

内　容

4.1　総　論

コンパイラの構成は，機能的にいくつかの部分に分かれている。それらの間の関係を明らかにした上で，おのおのを切り離して説明すると，説明が容易になるので，4.1では4.2以下に説明する個々の機能部分の全体での位置について説明する。

ここで説明される事柄の多くは，4.2以下の各論で再び詳細に説明される。

図4－1に典型的なコンパイラの構成を示す。教師は，この図を用いながら，各構成部分について，次の事柄を説明する。

(1) 情報テーブル

コンパイラにとって加工する情報はすべて一度はここに蓄えられる。

FORTRANなどの言語を例にして次の情報テーブルの必要性並びにその性質について簡単に説明する。

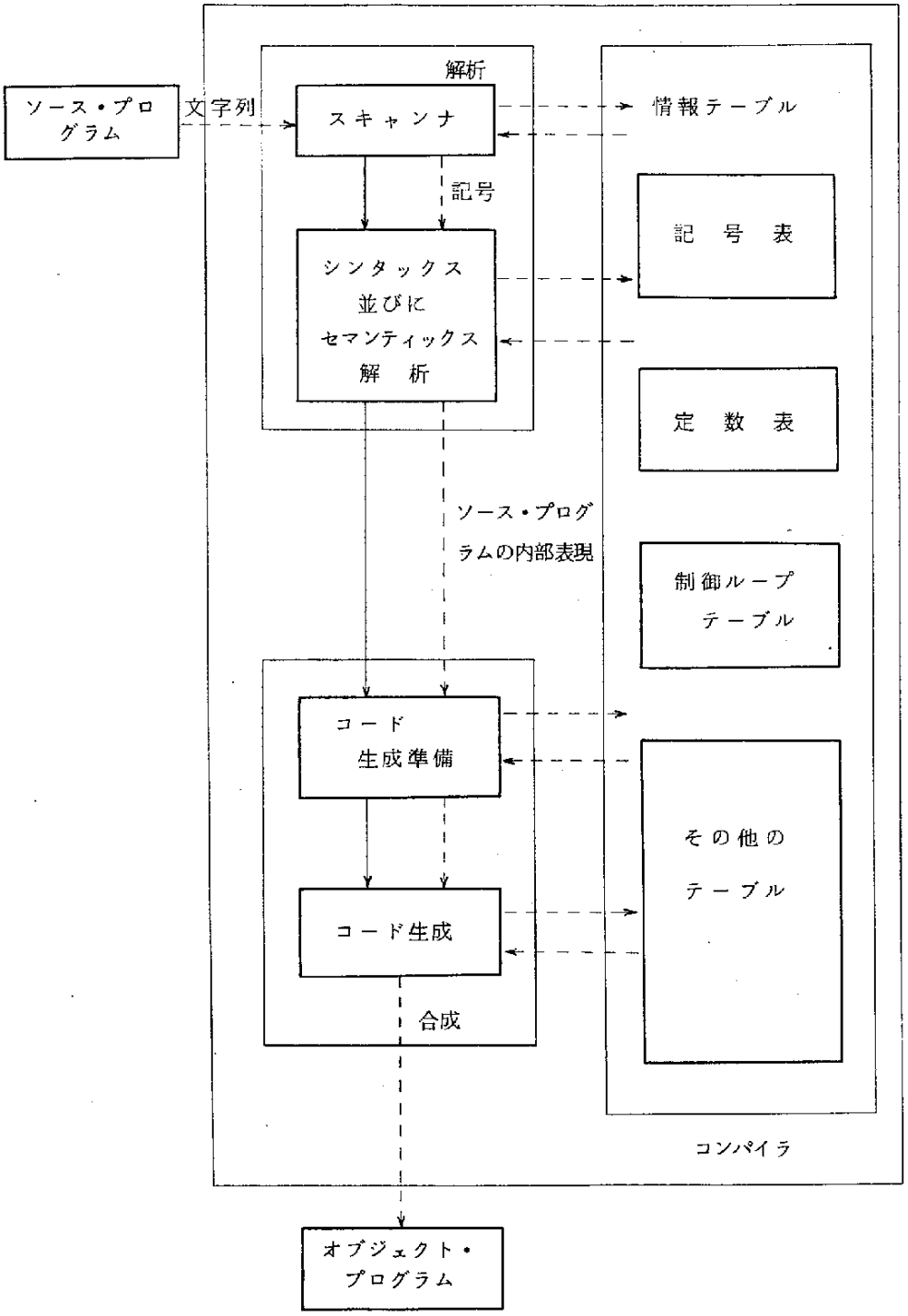


図4－1　コンパイラの構成例

① **記号表**

FORTRANでいえば変数名，配列名，文番号表に当たるもの。

② **定数表**

③ 制御ループ表

FORTRANでいえばDOループに関する情報を蓄える表

④ その他の表

FORTRANでいえば，COMMON，EQUIVALENCEに関する表など。

(2) スキャンナ (Scanner)

スキャンナはレコグナイザ（Recognizer），あるいは Lexical Analyzer などとも呼ばれる。

ソース・プログラムの文字列を読込んで来て，スキャンを行ない，次の処理ルーチンに渡すため，文字列を分解して記号，あるいは**トークン**（Token），**アトム**（Atom），**シラブル**（Syllable）等と呼ばれる小部分のつらなりにする。

(3) シンタックス並びにセマンティックス解析

コンパイラの中心部分であるシンタックス解析は，ソース・プログラムの構文を解析する。セマンティックス解析は，この結果を用いて意味的にそれが誤っていないか否かをチェックし，必要情報をテーブルに残し，**コード生成**に用いるためのソース・プログラムの内部表現を作成する。

この二つの機能は，設計によって一つのルーチン内に混在する場合もあるが，論理的には分割することが可能である。

以下では，これを分割してシンタックス解析を4.3パーサの項で，セマンティックス解析を4.5でそれぞれ説明する。

(4) ソース・プログラムの内部表現

解析の完了したソース・プログラムの情報は，適当な内部表現に変換されて次の処理ルーチンに渡される。この形はハードウェアあるいはコンパイラの設計思想によっても色々と変種がありうるが，ポリッシュ表現などの形を取ることを説明する。

(5) コード生成準備

ソース・プログラムの内部表現を直ちにオブジェクト・コードに変換する以前に次の処理が必要となる。

① **メモリ割付**

② プログラムの最適化

(6) コード生成

オブジェクト・プログラムの作成である。

＜演習問題＞

次に示すコンパイラの例を調べさせ，構成にどれだけの特徴があるかを比較させる。プロセサの思想，概要について知るだけで十分であろう。

① ALCOR 参考文献（6）参照のこと

② 参考文献（7）の ALGOL コンパイラ

③ WATFOR 参考文献（8），（1）

④ FORTRAN (H) コンパイラ
参考文献（1）で紹介している。

⑤ 参考文献 (10)，(11)，(12)，(13)，(14)，(15)，(16) のコンパイラについて適当なものを選んでその概要を調べさせる。

4.2 スキャンナ

スキャンナは，4.1 で述べたようにソース・プログラムを文法に従って記号単位に分解するのがその任務である。

スキャンナの識別する記号は，多くのプログラミング言語においては図 4－2 にあげたようなものであり，これらは図4－3 で示されるような生成規則により記述されることを説明し，従って，チョムスキーの Type 3 （正規文法）として考えられることを説明する。

有限オートマトンを定義し，この考えを用いるとこの文法に従う記号の識別がエレガントにできることを説明する。

状態遷移図と有限オートマトンとの関係を説明し，スキャンナへの応用を述べる。（図 4－4 参照）

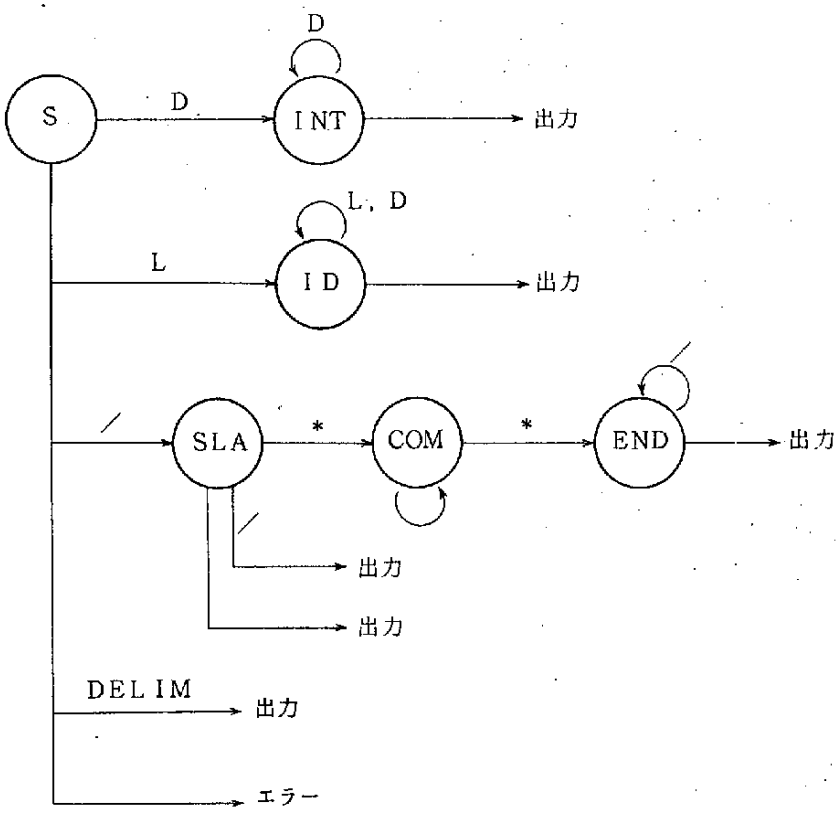
スキャンナについては，実際の言語に使用に際しては，空白の取扱い，FORTRANなどでは文の種別の判定などなお繁雑な問題があることを理解させる。

識 別 子
予約語（識別子の部分集合を成す）
整 数
1 文字からなる分離符（例 ＋，－，＊，／，（，）など）
2 文字からなる分離符（例 //，／＊，＊＊，：＝など）

図 4－2 高水準言語の記号例

＜識別子＞ ：：＝文字 | ＜識別子＞文字 | ＜識別子＞数字
＜整 数＞ ：：＝数字 | ＜整数＞数字
＜分離符＞ ：：＝＋ | － | （ | ） | ／ | ＊ | …
＜分離符＞ ：：＝＜SLASH＞／ | ＜SLASH＞＊ | ＜AST＞＊ | ＜COLON＞＝
＜SLASH＞ ：：＝／
＜AST＞ ：：＝＊
＜COLON＞ ：：＝：

図 4－3 記号を導出する生成規則



但し　S　：初期状態
　　D　：数字
　　INT：整数
　　L　：文字
　　ID　：識別子
　　SLA：SLASH
　　COM：コメント　／＊…＊／の間にはさまれたもの
　　DELIM：／以外の分離符

図4－4　記号を識別する状態遷移図

4.3 パーサ

与えられた文からシンタックス・トリーを作成するのが**パーサ**の役目である。

この方法には大きく分類して**トップ・ダウン法**と**ボトム・アップ法**とがある。

トップ・ダウン法は，シンタックス・トリーをルート側から構成していく方法であり，ボトム・アップ法は逆に葉の側から構成していく方法である。

これを図4－5の文法の場合について図4－6，図4－7を使ってわかり易く説明する。

たとえ文法が与えられていても与えられた記号列（この例では文字列）を識別してシンタックス・トリーを構成する方法は直ちには見つからない事情を説明し，そのために，識別の方法に色々の変種がある理由を納得させる。

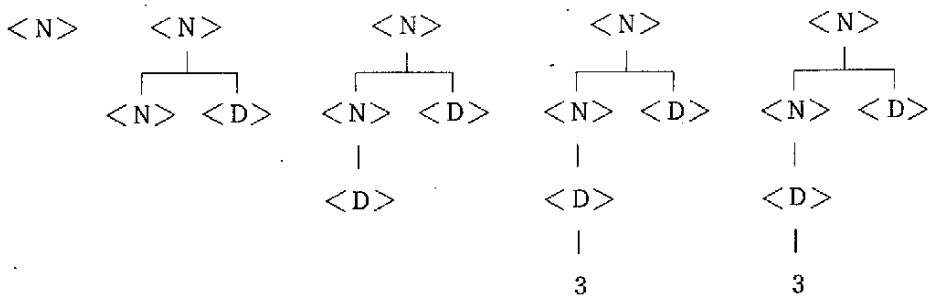
すでにスキャンナで記号の段階まで文字列がまとめられているので，パーサが識別する対象は，FORTRAN，ALGOLなどの数式を含む言語の場合には

2 + 2 * 5

のような数式が中心になる。この場合には幸いにして演算のオペレータが優先順位を持つためにこの性質を利用して，シンタックス・トリーを構成する方法―― **オペレータ順位**文法に応用できる方法（図4－8）について説明する。

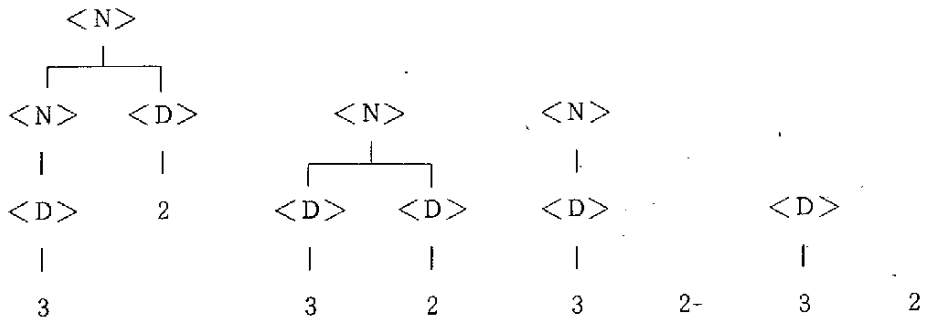
<N> ::= <D> | <N><D>
<D> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

図4－5　数字列を生成する文法



<N> ⇒ <N><D> ⇒ <D><D> ⇒ 3<D> ⇒ 32

図4－6　トップ・ダウン法　32を識別する例



<N> ⟹ <N><D> ⟹ <D><D> ⟹ <D> 2 ⟹ 32

図４－７　ボトム・アップ法　32を識別する例
（処理は右から左向きに進行する。）

この辺の議論は，シンタックスその他を定めると普遍的に定まるところなので，シンタックスを外部からデータとして与えて，コンパイルする言語に自由度を持たせようとする**シンタックス・ディレクテド・コンパイラ**の考え方を説明する。

〔準　備〕　オペレータ用のスタック　OPTOR
　　オペランド用のスタック　OPAND
を設ける。
〔アルゴリズム〕
以下でＳ１はOPTOR上の一番上にある（最後にプッシュ・ダウンした）オペレータを示し，Ｓ２は，スキャンナから次々と渡される記号を示す。
(1) Ｓ２がオペランドならば，Ｓ２をOPANDにスタックし，(4)に行く。
(2) Ｓ２がオペレータのケースである。Ｓ２の順位の方が，OPTORの一番上にあるオペレータの順位よりも低くない（たとえばOPTORのトップＳ１が＋でＳ２が＊のようなとき）ときには，単純にＳ２をOPTORにスタックし(4)に行く。
(3) 逆のケースには，S1により定まるセマンティックな解析ルーチンを呼出す。（4.5(2)参照）
このルーチンは，必要なセマンチックな処理を行なった後，S1と必要とされるオペランドとを夫々のスタックから除去し，S1の結果を示す情報をOPANDにセットする。
(4) 次の記号の処理に進む

図４－８　オペレータ順位文法でのパーサ

(注)　オペレータの優先順位の定め方には，別にややこしい問題があるが，詳しくは例えば参考文献（1）を参照のこと。

＜演習問題＞

図４－８で示したアルゴリズムによって

・簡単な数式

・かっこが用いられている数式

がどのように処理されるかを実際に指導する。

4.4 記 号 表

識別子についての情報はすべてこのテーブルに登録される。

変数及び手続きについての登録情報について説明する。（図４－９参照）

情報の１個１個についてその必要性，利用法等について説明すること。

この記号表には名札も含まれている。コンパイルには，この他にＦＯＲＴＲＡＮでいえばＤＯループ，ＡＬＧＯＬでいえば ｆｏｒ ループの制御のためのテーブルが必要と考えられるかもしれないが，これについては最適化の説明の個所で一緒に取扱う。

登録される情報：変数のケース。

(1) 型（整数，実数，文字列，複素数，名札その他の別）

(2) 精度，スケーリング，長さ

(3) 形体（単純変数，配列，構造，その他）

(4) 実行時アドレス

(5) 配列のときその次元数，次元のサイズが固定のときその値

(6) 構造あるいはその一部ならば，他の項目との関連

(7) 形式パラメータか否かの情報，もしそうならば対応するパラメータの型

(8) （ＦＯＲＴＲＡＮのとき）ＣＯＭＭＯＮあるいは ＥＱＵＩＶＡＬＥＮＣＥとの関連

(9) 宣言がすでに処理されたか否か

(10) この変数に値を与える表現のあるなし手続きのケース

(11) プログラムの内部手続きか否か

(12) 関数か否か，その型

(13) 宣言はすでに処理済か

(14) リカーシブか否か

(15) 形式パラメータは何か，その関係がリンクをたどって分るようになっていること。
（実パラメータとの比較に用いる）

図４－９　変数及び手続きの記号表

4.5 セマンティックス解析

セマンティックスの解析にも色々の流儀が存在するが，ここでは**ポリッシュ記法**による方法を中心に説明する。

(1) ソース・プログラムの内部表現

ソース・プログラムの内部表現としてポリッシュ記法を説明する。

① 算術式のポリッシュ記法による表現の方法

② ポリッシュ記法で表現された式の評価方法

③ 算術式以外にもポリッシュ記法を適用する方法

たとえばブランチ命令，条件ブランチ命令にも使用できる。

(2) ポリッシュ記法への変換方法

これは先に図4－8(3)で若干説明した，「セマンティックな解析ルーチン」のなす作業である。簡単な例を用いてどのようにポリッシュ記法によるソース・プログラムの内部表現が作られるかを説明する。この辺のロジックはパーサと連動して動くことを理解させる。

(3) 条件飛び越し，無条件飛び越し命令，繰返し命令の処理

対応する適当なポリッシュ記法に変換する方法を説明する。

(4) 演習問題

(a) 数式をポリッシュ記法に変換する方法について再度実例で指導する。

(b) 入れ子になるＩＦ文をポリッシュ記法に変換する方法を考えさせる。

(c) DOループあるいは for ループなどの繰返し命令をポリッシュ記法にすることを考えさせる。

4.6 コード生成前処理

総論で述べたように次の２項目について説明をする。

(1) メモリ割付

次の事柄について説明する。

① 配列の扱い方

- 一般的な方法
- **ドープ・ベクトル**
- **Illifeベクトル**

② 文字列の扱い方

③ 構造の扱い方

④ 実パラメータと仮パラメータとの結合法

- Call by Reference
- Call by Value
- Call by Result
- Dummy Arguments

・Call by Name

・配列名を実引数に指定したとき

・手続名を実引数に指定したとき

・ブロック構造の扱い方

・初期値を持つ変数の扱い方

・レジスタの有効利用*

・一時的作業用変数の扱い方*

・（FORTRANの場合）COMMONとEQUIVALENCE宣言された変数または配列の扱い方

メモリ割付は，大変微妙な問題をかかえ難解なので，単なる問題の指摘だけでも十分である。ここで列挙されないもっと複雑な問題のあることだけは断っておくこと。

(2) 最適化

この主題も大変に微妙且つ高度な問題であるので，深遠な問題があることを一応断つた上で，以下のごく基本的な事項についてのみ説明する。（図4－10参照）

① レジスタの有効利用*

② 共通項の削除

③ 定数項の計算の削除

④ ループの制御変数の取扱い

⑤ ループ内の添字計算の取扱い

⑥ ループ外への移動

⑦ 仮引数の取扱い

――ループに関する最適化のために用いられる――

ループ・テーブルの1要素の情報：

・ループ変数，ループの制御変数

・外側のループの個数

・ループ用のCall by Value で用いられた単純仮パラメータ・リストの第1要素のインデックス

・同上最終要素のインデックス

・Call by Value で用いられている単純整数仮パラメータ・リストへのポインタ

図4－10 ループ・テーブル

* この最適化は，4.7のコード生成と同時に行なわれることが多い

(3) 演習問題

(a) 本節で述べられた項目の内適当な数項目についてより詳細に研究させる。

(b) 共通項の削除で用いられている木構造について調べさせる。

4.7 コード生成

コンパイラが最終的に出力するオブジェクト・プログラムの作成について以下の事項を順に説明する。

(1) オブジェクト・プログラム

次の3種の利害得失について説明する。

① 絶対アドレス・コード

② アセンブリ言語

③ リロケータブル・コード

(2) ポリッシュ記法からのオブジェクト・プログラムの変換方法

(3) 最適化の工夫

種々のオブジェクト・プログラムの最適化の工夫について述べる。

① レジスタの有効活用

② ユナリ・マイナス記号の扱い方

③ リテラル命令の活用

④ 負数ロード命令の活用

なおこれらの工夫は対象コンピュータのハードウェアと密着したものであることを理解させる。

(4) アドレッシングの問題

ALGOLなどデータ構造が複雑な場合には，データの参照に苦労が伴うことを説明する。方法までの説明は不要。

(5) コード生成モジュール縮小の工夫

インタプリタ方式によるコード生成法など，このモジュールのステップ数減少の工夫を説明する。

(6) リロケータブル・コード

次のような情報から成ることを説明する。

① 外部参照記号

② テクスト部分

③ リロケーション・ディクショナリ

④ END カード

またこれと関連して**リンカ**，ローダの役目についても簡単に説明し，その有効性を教える。

4.8 デバグ・サポート

ユーザがコンパイラを利用してデバグを効率よく行なえるために必要となる種々のサポート機能には次のようなものがあることを説明する，かつその内部ロジックを説明する。

(a) 添字付変数がその範囲を越えたときのチェック

(b) 0による除算など算術的な異常状態のチェック

(c) 値がセットされていない変数を参照したときの検出

(d) エラーの生じた場所を入力ソース・プログラムの位置で知らせる機能

(e) 記号を使ってのスナップ・ショット出力，あるいはエラーを起した場所での参照される可能性のあるデータの出力

4.9 その他

コンパイラに関する最近の発展，その他の話題について簡単に解説する。

(1) TWS（Translator Writing System）（参考文献（1）の 20.1参照）

TWS，コンパイラ・コンパイラ等についての狙いと発展の歴史を解説する。

(2) コンパイラの記述言語

アセンブラだけでなく，他の高水準言語が使われる傾向について解説する。

IBMの FORTRAN Ⅳ(H) はそのよい例である。（参考文献（1）の1.3.（1）の 20.1参照）

(3) コンパイラの内部テーブル（参考文献（1）の21章参照）

オーバフローの考慮や，2次記憶に吐き出すときの原則について述べる。

(4) リエントラント・コード（参考文献（1）21章参照）

マルチプログラミング，タイムシェアリングのため，リエントラント・コード の出力が望ましいことを説明する。

(5) ブート・ストラッピング（参考文献（1）の 21章参照）

コンパイラ自身を記述するのに便利なような機能の大きなコンパイラをはじめて開発するのに有利な方法であることを説明する。

(6) 拡張可能言語（参考文献（1）の 20.1参照）

拡張可能言語は，ALGOL 68 などで採用された高級な概念であることを説明する。

(7) 演習問題

コンパイラでの設計思想の重要性について問う。

指導上の留意点

(1) コンパイラの構成は，開発される対象のコンピュータ，設計思想，最適化の程度，使用目的の特殊性などを反映して著しく変わることに留意する。

本章で設けた数節が，設計の如何によっては全く存在しないようなコンパイラも別段めずらしいことではない。

(2) コンパイラ程度の複雑な長大プログラムには，一貫した設計思想が必要となることを理解させたい。

第5章　設計演習

目　標

演習を通じて簡単な言語プロセサの設計を行なうことによって今までの復習をさせる。

まずアセンブラの設計演習によって，数式以外の言語プロセサの設計能力を身につけさせ，次にコンパイラの設計演習によって数式処理に関連する処理を設計させる。

内　容

5.1　アセンブラ

アセンブラの最小限の機能，即ち

① インストラクションが記号であること

② アドレス部に記号が使用できること

③ 数値，文字定数が定義できること

を満す言語仕様を与え，設計の指導を行なう。

5.2　コンパイラ

数式表現が許せる最小限の機能のプログラミング言語の仕様を与え，その設計を指導する。

シンタックスは，BNFで記述されたものが望ましい。（図5－1参照）

```
<program>    ::  = <assignment> | <assignment>; <program>
<assignment> ::  = <variable> = <arith expr>
<arith expr> ::  = <term> | <arith expr> + <term>
<term>       ::  = <factor> | <term> * <factor>
<factor>     ::  = <variable> | <integer> | (<arith expr>)
<variable>   ::  = <letter> | <variable><letter> | <variable><digit>
<integer>    ::  = <digit> | <integer><digit>
<letter>     ::  = A|B|C|D|E|F|G
<digit>      ::  = 0|1|2|3|4|5|6|7|8|9
```

図5－1　簡単な言語のシンタックス例

指導上の留意点

(1) 設計は研修生の能力，時間を考慮して適当に粗くして，5.1，5.2の両方を行なう。

参考文献

(1) David Gries, Compiler Construction for Digital Computers, John Wiley & Sons, Inc. 1971

(2) R. A. Hopgood, Compiling Techniques, American Elsevier Publishing Company, Inc. 1969
同上訳，ホップグッド（首藤 訳),「コンパイラの技法」，サイエンス社，1973

(3) C. William Gear, Computer Organization and programming, McGraw-Hill Book Company, 1969
同上訳，ギア（藤井 訳） 「コンピュータの構造とプログラミング」，産業図書，1972

(4) Donald E. Knuth, The Art of Computer Programming, vol. 1, Fundamental Algorithms, Addison-Wesley, 1968

(5) 情報処理学会，情報処理ハンドブック，オーム社，1972

(6) Gries, D., Paul, M., and Wiehle, H. R., Some techniques used in the ALCOR ILLINOIS 7090, CACM 8 (Aug. 1965), 496−500

(7) H. Kanner, P. Kosinski, and C. L. Robinson, The Structure of Yet Another Algol Compiler, CACM vol. 8 July 1965, 427−438

(8) Cress, P. H. et al., Description of/ 360 WATFOR, Dept. of Applied Analysis and Computer Science, Univ. of Waterloo. Report No. CSTR−1000

(9) Saul Rosen, Programming Systems and Languages, McGraw-Hill Book Co., 1967

(10) 竹中 靖， ALGOL Compiler の作製について，情報処理，第2巻1号

(11) 池野信一， M1B計算機のAUTO CODE 1 とそのコンパイラー情報処理，第3巻2号

(12) 田村康男 他，自動プログラム "EASE" について，情報処理，第3巻3号

(13) 辻ケ堂 信 他，FACOM 222 FAST (CDT 402型)，情報処理，第4巻5号

(14) 井上謙蔵 他， ISSP ALGOL コンパイラ，情報処理，第5巻1号

(15) 田中 明 他，プロセス用計算機のコンパイラについて，情報処理，第5巻5号

(16) 三浦大亮， ALGOL − 翻訳の例，情報処理，第5巻6号

アセンブリ言語

目　次

第1章　プログラミングのためのハードウェア …… 471

1.1 システムの概要 …… 471

1.2 データの形式 …… 472

1.3 レジスタ …… 472

1.4 機械語命令の形式 …… 473

1.5 アドレス方式およびアドレス修飾 …… 473

1.6 割 込 み …… 474

1.7 チャネルと入出力動作 …… 474

第2章　アセンブリ言語の構成 …… 475

2.1 概 要 …… 475

2.2 コーディング用紙 …… 475

2.3 記号（シンボル）…… 476

2.4 項（ターム）…… 476

2.5 記 号 式 …… 476

第3章　実行命令およびアセンブラ命令 …… 478

3.1 命令形式 …… 478

3.2 機械語命令 …… 478

3.3 アセンブラ命令 …… 479

3.4 アセンブラ制御命令 …… 481

第4章　プログラムの分割と連絡 …… 482
4.1 概　要 …… 482
4.2 コントロール・セクション …… 483
4.3 外部記号 …… 483
第5章　マクロ言語 …… 484
5.1 概　要 …… 484
5.2 ポジショナル・マクロ …… 485
5.3 条件つきマクロ …… 486
5.4 キー・ワード・マクロ …… 487
5.5 混合モード・マクロ …… 487
第6章　実習のためのオペレーティング・システム …… 489
6.1 ファイル …… 489
6.2 システム・マクロ …… 489
6.3 ジョブ制御文 …… 489
6.4 連けい編集プログラム …… 490
6.5 アセンブラの制御文 …… 490
6.6 ユーティリティ・プログラム …… 490
第7章　実　習 …… 491
7.1 データの形式と編集 …… 491
7.2 アドレス修飾およびくり返し（ループ） …… 491
7.3 サブルーチン …… 492
7.4 マクロ …… 492
7.5 事務計算 …… 492
7.6 総合実習 …… 492

科目「アセンブリ言語」

教育の目標

アセンブリ言語を理解し，アセンブリ言語による高効率（実行速度，主記憶占有域などの点で）のプログラムを作成する能力およびマクロ機能，プログラムの分割などに関する知識を実務面で活用し融通性，汎用性，拡張性，保守性などに富んだプログラムを作成する能力を養う。

実習では，プログラム設計，デバッグ，文書化などを含めたアセンブリ言語によるプログラミング作業の綜合実習を行ない，教育終了後直に実務を遂行できる基礎能力を修得させる。

このため綜合演習は専門編「プログラム設計(II)」の終了後行なうことが望ましい。

留意事項

(1) アセンブリ言語は，汎用言語と異なりハードウェア，オペレーティング・システム（OS）に依存する部分が多く，特定のコンピュータのハードウェア，OSの知識なしにアセンブリ言語を理解することは難しい。

このためこの科目では特定のシステムのハードウェア，OSについて，アセンブリ言語に関連の深いものは触れている，すなわち次の項目中プログラムの作成，実行に直接関係のあるものはこの科目で取扱う。

(a) ハードウェア（レジスタ，割込みなど）

(b) ジョブ制御言語

(c) システム・マクロ

(d) データ管理

(e) 連けい編集プログラム

(2) この科目ではハードウェア，OSに関連する部分の記述は簡略化されているが，「ハードウェア(I)」「ハードウェア(II)」，「オペレーティング・システム」の各科目を適宜参考にすると良い。

(3) 研修生が使用するテキストは，実際にはコンピュータ・メーカ提供のマニュアルを利用せざるを得ないであろう。

このため説明順序，説明の仕方（例えばアセンブリ言語の各命令を機械語命令と一つ一つ対応させて説明しているもの，アセンブリ言語の文法と機械語の説明を独立に行なっているものなど）に大きな違いがあり，採用するテキストに従って順序を変更してよい。

また用語に関してもコンピュータに依存するものが多く，採用するテキストの用語に従った方が便利である。*

---

* 第3章～第5章の用語はHITAC EDOS アセンブラ（参考文献(2)）のものを使用している。

(4) コンピュータによってはここに記述してあるものゝすべてを実現していないものもあり，実現していない部分は省略してよい。

ただし，この科目の目標の一つにアセンブリ言語の一般的な知識の習得があるので次の項目が実現されていないものがあれば，他の適当なアセンブリ言語について，その部分のみ追加説明をすることが望ましい。

(a) マクロ機能

(b) プログラムの分割と連結に関する機能

また逆にここに記述されていない機能が実現されている場合には適宜追加して説明する。

(5) 入出力命令（入出力マクロを除く），チャネル・コマンド，入出力に関連した割込み等直接ハードウェアの入出力機能，動作に関する部分は，研修生の今後の業務を考えて，概要説明ないしは省略してよい。

すなわち制御プログラムなどを作成する者以外は上記の項目については，「ハードウェア(I)」，「ハードウェア(II)」の 各科目の研修で十分であり，この科目では概要説明あるいは省略してよい。

(6) 実習は第7章でまとめて行なうようになっているが，各命令の文法，機能などについては適宜演習問題を与えて説明した時点で理解させるようにする。

時間配分

章	履習時間（時間）	
	講　義	実　習
1. プログラミングのためのハードウェア	5	
2. アセンブリ言語の構成	3	
3. 実行命令およびアセンブラ命令	7	
4. プログラムの分割と連結	5	
5. マクロ言語	10	
6. 実習のためのオペレーティング・システム	5	
7. 実　習		145
合　計	35	145

第1章　プログラミングのためのハードウェア

目　標

この章ではプログラミングに必要なシステム構成，ハードウェアの仕様を理解させる。

すなわち効率のよいプログラムを作成するためには，システム構成および構成要素の諸元（主記憶容量，チャネル，入出力装置，補助記憶装置の速度，台数など）を知っている必要がある。

またアセンブリ言語はハードウェアに密接に依存しているため，データ形式，各種レジスタの機能およびその使用法，割込み，機械語の形式，アドレス方式，番地の修飾などアセンブラ言語の理解，プログラミングに必要な知識を習得させる。

内　容

1.1　システムの概要

アセンブリ言語の対象となるシステムにつきシステムの特徴，構成を説明する。

(1)　システムの特徴

①　互換性

該当機種のファミリの中での位置づけ，上位，下位機種間および異なったオペレーティング・システム間での互換性について，その概要を説明する。

②　可用性

システムの稼動性を高め，また種々の使用目的に対して能率のよい運用をするため，システムとしてどのような配慮がなされているか，次のような項目について，その特徴を説明する。

- 信頼性の向上
- 障害時の対策
- 保守性の改善
- 操作性の向上

③　オペレーティング・システム

該当オペレーテイング・システムについて，その処理形態（バッチ，リモート・バッチ処理……），および機能（ジョブ管理，タスク管理，データ管理，ハードウェア 管理……）の概要を説明する。

(2)　システム構成

システムを構成する装置につき，数量，容量，速度など，構成図を示してその概要を説明する。

①　主記憶装置

②　中央処理装置

- 固定記憶（ROM）
- 各種レジスタ

・記憶保護機構

・経時時計機構

③ チャネル制御装置

④ チャネル装置

・セレクタ・チャネル装置

・マルチプレクサ・チャネル装置

⑤ 入出力制御装置

⑥ 入出力装置，補助記憶装置

(3) コンピュータ室の見学

システムの構成機器，操作法などを見学させる。

1.2 データの形式

（特定の）コンピュータで取扱われるデータの形式の選択法などをプログラムの効率（記憶装置の占有量，演算速度など）に関連づけて説明する。

すなわちコンピュータ内でのデータ，　表現形式には固定小数点，浮動小数点，パック形式，10進数，文字（EBCDICなど）形式があり，また　データ形式によっては格納する境界（バウンダリ）の制限があることから,目的に応じて，最適なデータの表現形式で主記憶に格納すべきことを理解させる。

データ形式の選択に当って考慮すべきものに次のような事項がある。

① 数値の格納密度

② そのまゝの形式で四則演算が可能か

③ 可変長データに対する処理が可能か

④ 表現できる数値の精度，大きさ

⑤ 演算時の位取り

⑥ 演算速度

⑦ 編集（パンクチエイション）が容易か

⑧ 16進（8進）表示でダンプした時の内容の判読のし易さ

⑨ 境界（バウンダリ）条件

1.3 レジスタ

アセンブリ言語の理解，およびアセンブリ言語によるプログラムの作成に必要な各種レジスタ，特に下記に示すレジスタの諸元，機能，使用法について説明する。

① アキュムレータ

② インデックス・レジスタ

③ 浮動小数点レジスタ

④ 割込みマスク・レジスタ

⑤ コンディションコード・レジスタ

(演算結果の状態を示すレジスタ)

⑥ シーケンス・カウンタ

(次に実行する命令の番地を格納)

⑦ 割込みフラッグ・レジスタ (割込み原因を格納)

⑧ 入出力チャネル・レジスタ

- チャネル・アドレス・レジスタ
- チャネル・コマンド・レジスタ
- 装置状態レジスタ

(注1) 上記のレジスタは，いくつかが統合され，一つのレジスタでいくつかの機能を持つようなものもある。

(注2) 主記憶，その他が使用され，レジスタと呼ばないものもある (割込み原因など)

(注3) 入出力チャネル・レジスタは入出力命令を直接アセンブリ言語で書く場合のみ必要であって，他の場合は機能の概要説明だけでよい。

(注4) 上記以外のレジスタでプログラミングに直接関係のないものは機能の概要説明ないし省略してよい。

1.4 機械語命令の形式

一般に機械語命令は命令の種類，機能などによって形式がことなるが次の事項について説明する。

① 命令形式

② 形式による一般的機能および特徴

③ プログラミング上の利点および留意点

1.5 アドレス方式およびアドレス修飾

(1) アドレス方式

通常のプログラムが対象とするアドレスは論理アドレスであるが，物理アドレスと論理アドレスが一致しないコンピュータではその相違，アドレス変換機構について説明する。

(2) アドレス修飾

アドレス修飾にはインデックス・レジスタによるものと間接アドレスによるものとがある。

① 間接アドレスによる修飾

② インデックス修飾

③ 間接アドレスおよびインデックス

修飾が同時に指定された場合などについてその実効アドレスの算出法およびその使用法について

説明する。

1.6 割 込 み

中央処理装置には割込み制御を容易にするために，いくつかの処理状態（モード）がある。通常のプログラムは普通処理状態で実行されるがこれは割込みにより割込みを処理する状態に切り換えられる。これらの処理の種類およびその概念を説明する。

① 割込みの種類

割込みの種類（条件）について説明する。割込みの種類でプログラミングに重要なものは

- スーパバイザ・コール
- 特権命令
- 不当命令コード
- アドレス誤り
- データ誤り
- けたあふれ
- 指数部オーバ／アンダ・フロー
- 除算エラー
- 経時時計

などである。

② 割込み処理

割込み条件が発生してから割込みが受け付けられるまでの過程を流れ図を使って説明する。

③ 割込みの優先順位

④ 記憶保護

⑤ 割込みの制御

⑥ 割込み原因の判定

1.7 チャネルと入出力動作

入出力命令実行の際使用するレジスタ，チャネル・アドレス語，チャネル・コマンド語について説明する。

指導上の留意点

(1) この章はハードウェアに対する一般的な知識のあることを前提としているが，指導に際しては，「ハードウェア(I)」，「ハードウェア(II)」の各科目も必要に応じて利用するとよい。

(2) テキストに使用するマニュアルはハードウェアの仕様，機能を中心に書かれている嫌いがあるが，プログラム中でその機能が十分活用できるよう，代表的かつ具体的な使用例で説明することが重要である。

(3) 研修後の業務を考慮して，入出力に関連する事項，割込みに関連する事項は適宜省略してよい。

第2章　アセンブリ言語の構成

用　語　この章では次の用語を教える。

ロケーション・カウンタ，記号，絶対記号，相対記号，項，自己規定項，リテラル，記号式

目　標

アセンブラの概要を把握させ，ソース・ステートメントの構成要素（名札，ニモニック・オペレーション・コード，記号式など）の定義，意味，使用法を理解させる。

内　容

2.1 概　要

下記の項目を中心にして，アセンブラの機能および特徴を説明する。

① ニモニック・オペレーション・コードの使用

② **ロケーション・カウンタ**および記号によるアドレス表示

③ 10進，2進など文字形式（EBCDICなど）による多種類のデータ表示法

④ プログラムの分割と結合のための手段の提供

⑤ マクロ機能（定義および参照）

⑥ プログラム・リストの作成

- ソース・リスト
- 文法誤りの検出，表示
- クロス・リファレンス

2.2 コーディング用紙

コーディング用紙にはカード用，紙テープ用などがあり，それぞれ形式がことなっている。またあるシステムではinput format control というアセンブラ制御命令によりフォーマットを変更することができる。一般には次の項目について説明する。

① 制御欄

② ステートメント・フィールド

- 名札部（ロケーション・フィールド）
- 操作部（オペレーション・フィールド）
- オペランド部
- コメント部

③ 継続行指定欄

④ 行番号フィールド

⑤ コーディングに使用される文字

2.3 記号（シンボル）

記号はアセンブリ言語を理解する上で最も重要なものであり，記号に対する値の割付け方およびその値の持つ意味を下記の項目に従って理解させる。

① 値の割付け

・ロケーション・カウンタによるもの

・ＥＱＵ命令（3.3アセンブラ命令参照）によるもの

・連けい編集プログラム（4.3外部記号参照）によるもの

② 値の意味

・絶対記号

・相対記号

をプログラムのリロケーションに関連づけて説明する。

2.4 項（ターム）

アセンブリ言語では，オペランド部に記述された番地やデータを機械語の対応するフィールドに埋め込むことによって機械語命令を生成するが，その番地やデータを指定する便宜のために次の**項**が用意されていること，および各項について機能，用法，絶対項と相対項の区別などについて説明する。

① 自己規定項

・10進自己規定項

・16進（8進，2進）**自己規定項**

・キャラクタ自己規定項

② リテラル

実装されている主記憶装置の全領域のアドレッシングができない方式のコンピュータでは，**リテラル**で指定された定数の格納領域（リテラル・プール）についても考慮が必要であるなど，コンピュータの特殊性に応じて留意事項を適宜説明する。

③ ロケーション・カウンタの参照

ロケーション・カウンタの参照を利用することによって，名札の数の削減には有効であるが，機械語命令が可変長の場合，プログラムを変更する場合など誤りを犯し易いので，プログラミング上の留意事項も併せて説明する。

2.5 記号式

記号式を下記に従って文法，意味，用法を説明する。

① 式の種類

・単一記号式

・複合記号式

② 式の値

・絶対値 …… 絶対記号式

・相対値 …… 相対記号式

特に複合記号式の値（絶対値を持つか，相対値を持つか）はアセンブラによって異なる（例えば，相対値＋相対値は一般に無意味であって式として誤りであるが，FACOM 230−25/35ではバイトの相対アドレス（相対値）を示し，正しい式である）ので，コンピュータの特殊性を考慮し，式の意味および用法は具体例を中心にして説明する。

指導上の留意点

リテラルの考え方はアセンブリ言語で重要であり，実現されていない場合でも，その概念を説明する。

第3章　実行命令およびアセンブラ命令

用　語　この章では次の用語を教える。

アドレス定数，重複因子，リテラル・プール

目　標

この章では，実行命令とアセンブラ命令を修得させることを目標とする。まず，実行命令については，ソース・ステートメントとそれが翻訳されて生成される機械語命令との対応関係を理解させる。

また，各機械語命令の機能，速度，用法を理解し，アルゴリズムが流れ図などで与えられたとき，適宜この機械語命令を組合わせて，効率（実行速度，主記憶占有量など）のよいルーチンを組み立てる能力を養う。次に，アセンブラ命令については，プログラム中で使用する各種定数の定義，領域の確保，プログラム・リストの制御などの命令につき機能，使用法を理解させる。

内　容

3.1　命令形式

一般に機械語命令には，いくつかの形式があり，その形式によってアセンブラ言語での表現形式も異なっている。各形式につき，ソース・ステートメントの各フィールドと機械語命令の各フィールドとの対応関係を説明する。

3.2　機械語命令

機械語命令は機種によって異なるが，下記の命令について，命令形式，機能，速度，使用法，異常ケース（誤データ，オペランドの指定誤りなど）について説明する。

① 固定小数点演算命令

② 10進演算命令

③ 浮動小数点演算命令

④ 論理演算命令

⑤ 分岐命令

⑥ 状態制御命令

（スーパーバイザ・コール，割込み制御など）

⑦ 特権命令

異常ケースとしては次のようなものがあるが割込み動作，命令実行終了後のレジスタ，オペランドの内容などについて説明する。

① アドレス・エラー

・アドレス付けエラー

実装されている記憶装置の容量を超えてアドレスを指定

・境界（バウンダリ）エラー

・レジスタ指定エラー

命令によってあらかじめ定められたレジスタを指定していない。

・記憶保護違反

② データ・エラー

10進数の符号，数字に不当なビット・パターンがある。

③ 桁あふれ

・固定小数点演算の桁あふれ

・10進演算の桁あふれ

・浮動小数点演算の指数部オーバ／アンダー・フロー

・シフト命令による桁あふれ

なお，各命令の説明に当っては，その命令の代表的な使用例，また慣用的な使用法などを同時に説明することが望ましい。

例えば，

① メモリとメモリ間での転送ができるコンピュータで，オペランドを重複して指定することによる領域のクリア

② ブランチ・アンド・リンク命令に関連して，サブルーチン中でのパラメータの参照法

③ シフトアンドカウント命令を使用した浮動小数点データの正規化

④ インデックス・レジスタのテスト命令に関連して，くり返し（ループ）数の制御

⑤ 固定小数点演算の桁あふれに関連して単精度演算命令を使用した倍精度の加減算

⑥ ストア・マルチプル（複数のレジスタの内容を同時に主記憶に格納する）命令に関連して，サブルーチンの入口，出口におけるレジスタの内容の保存，回復

⑦ シフト命令を利用した2のべき乗数による乗除算

3.3 アセンブラ命令

アセンブラ命令は定数の定義，領域の確保，番地割当ての制御，ソース・リストの出力に対するオプションの指定などの機能を持っており，翻訳の結果実行命令は生成されない。アセンブラ命令の中にはプログラムの読み易さ，書き易さ，プログラムの拡張性，保守性の向上を目的とした命令が多く，説明は文法，機能と併わせて具体的用法，用法上の利点などをあげ，直に実務に応用できるよう指導する。

(1) Equate（EQU）命令

EQU命令はプログラムに拡張性，融通性を持たせる意味で非常に重要である。

① 値の割当て

② 値の意味（絶対値，相対値）

③ 用法上の利点

・オペランドの短縮

・拡張性，融通性に富んだプログラム

・プログラムの読み易さ

・主記憶の多目的（重複）利用などを中心に説明する。

(2) 定数定義命令

まず，ここで定義できる定数にはデータ定数（固定小数点，浮動小数点，パック形式10進数…）と**アドレス定数**の2種類があり，それぞれプログラム上での用法の違いがあることを説明し，定数の値，定数の形式，境界（バウンダリ），**重複因子**，定数の長さ，小数点の位置（スケーリング）等に関する文法，機能，用法を説明する。

(3) プログラムの開始，終了

プログラムの開始，終了を指定するものに，一般にSTART，END命令（アセンブラによって異なる）が用意されている。これらの命令はプログラムの開始，終了の指定の他，プログラム名，格納開始番地，実行開始番地の指定などオペレーティング・システムに関連のあるものがあり，連けい編集，プログラム・ロード，プログラムの実行に関連づけて具体的に説明することが望ましい。

(4) ロケーション・カウンタ・セット命令

（Set Location Counter Origin：ORG命令）

① 格納開始番地の指定

② 主記憶の重複使用

など用法を中心に説明する。

(5) 境界調整命令

境界を調整するための命令につき文法，使用法を説明する。

この命令により調整されて生じた空き領域には，ノー・オペレーション命令が埋められ，定数定義命令や領域定義命令，ORG命令で調整されたものと異なることを，パラメータとサブルーチンへの分岐命令との例などを通して説明するとよい。

(6) リテラル・プールの指定命令

リテラル・プールの指定命令（Literal Origin：LTORG命令）はソース・プログラム中で使用されたリテラルを格納する領域の開始位置を指定するが，主記憶上の任意の番地がアドレッシングできないコンピュータで大きなプログラムを書く場合に重要な意味をもつ。

また同一プログラム中に，いくつかのコントロール・セクション（4.2コントロール・セクション参照）がある場合の用法については，プログラムの格納についての知識も必要であり，これらを部分的に理解させた上で説明することが望ましい。

(7) ベース・レジスタの指定

ベース・レジスタのないもの，ベース・レジスタはあっても機能が異なるものなどがあり，システムの特殊性を考慮し文法，機能，具体的用法を説明する（この命令が実行命令であるようなシステムもある）。

ベース・レジスタがユーザに開放（レジスタ番号の指定，内容の変更など）されているシステムではレジスタの指定，解放に関する命令が用意されている。この種のシステムではベース・レジスタの使用法により，実行速度に大きな影響を与えるので，その有効な使用法についても説明する。

3.4 アセンブラ制御命令

アセンブラ制御命令は，アセンブラに対して制御情報を与えるものであり，ソース・リスト，マクロの引用，連けい編集プログラムに関する制御情報を指定するものなどがある。

以下の機能を持った命令について文法，使用法を説明する。

① ＴＩＴＬＥの指定

ソース・リストのヘッディング，出力カードの行番号フィールドの１部に出力する文字列を指定する。

② 改ページの指定（ＥＪＥＣＴ命令）

③ スキップの指定（ＳＫＩＰ，ＳＰＡＣＥ命令）

ソース・リストに空白の行をおく

④ ソース・リスト，目的プログラムの出力制御命令

⑤ ソース・プログラムの形式の変更（input format control）

⑥ ソース・プログラムのシーケンス・チェック

⑦ 連けい編集プログラムへの制御情報の出力

⑧ マクロ参照の最適化

マクロ参照を効率的に行なうため，ワーク・ファイルに格納するマクロ名，その優先順位などを指定する命令であり，まずマクロの展開をアセンブラはどのような手順で行なっているかを理解させた後説明する。

指導上の留意点

(1) 各命令の機能が有効に応用できるよう，命令の機能と使用例を関連づけて説明することが重要である。文法の説明に当っては各命令を逐一とり上げて行なうことは必要でなく，共通な形式のものは，まとめて行なう程度で十分である。

ただし文法上誤り易いもの，ある条件のもとではその使用が好ましくないもの，制限事項などはその理由を含めて十分理解させることが必要である。

(2) アセンブラ命令の一部には第４章，第５章で触れるコントロール・セクション，マクロ命令の機能の概要を理解していることが必要なものがあるが，その都度必要な部分のみ前記につき説明する。

(3) ソース・リストの出力に関する制御命令では，望ましいリスト形式（リストの書式規定が作られていれば，その規定）を説明する。

第4章　プログラムの分割と連結

用　語　この章では次の用語を教える。

コントロール・セクション，コモン・セクション，ダミー・セクション，外部記号

目　標

大きなプログラムを，いくつかに分割して作成することは，

- 並行作業による工期の短縮
- 設計のし易さ
- アセンブリのためのコンピュータ時間の節減
- 主記憶必要容量の軽減（オーバーレイ）
- 保守の容易さ

などの点から重要である。

ここでは，アセンブラ，連けい編集プログラムが取扱うデータの最小単位であるコントロール・セクション，分割されて作られたプログラムの連結（制御の受け渡し，共通領域の定義，参照）に関する知識を修得させる。

内　容

4.1 概　要

アセンブリの単位でありかつプログラムの連けい編集の単位である**コントロール・セクション**の概要を把握させる。システムによっては他に**ダミー・セクション**，**コモン・セクション**が用意されているものがある。またコントロール・セクションと殆んど同じ意味でセグメントという用語を用いているものもあり，適宜そのシステムに準じた説明を行なう。用語（定義を含む）もシステムで用いているものを使用した方が便利である。

説明に当っては，コントロール・セクションの概念がプログラム作成上なぜ必要かをまず理解させることが重要である。

またここではアセンブリの結果出力される目的プログラムの形式についても，更に理解を深めるために概要を説明する。

特にコントロール・セクション，外部記号および外部記号の参照に関しての情報が目的プログラム中にどのような形で格納されており，連けい編集プログラムでどのような処理が施されるか図を用いて説明する。

4.2 コントロール・セクション

コントロール・セクション，ダミー・セクション，コモン・セクションなどの定義命令に関する文法，機能を説明した後，実例をあげてさらに理解を深めさせる。

これらのセクションはシステムにより概念が異なっているので，以下を参考にして，そのシステムに合った具体例で説明する。

(1) コントロール・セクションとOSとの関連

・ベース・レジスタとの関連

・プログラムの格納や連けい編集との関連

(2) サブルーチン間での領域の相互参照との関連

特に汎用言語（FORTRAN，COBOLなど）で書かれたプログラムとの相互参照（例えばFORTRANの名前付共通ブロックとコントロール・セクション，無名共通ブロックとコモン・セクションとの関係など）

(3) 記述順序と格納順序

例えば目的プログラムをアセンブリ言語とするコンパイラでは，1ステートメントから生成される命令に続けて，そのステートメントで生成される定数，作業領域の定義命令を出力できる。

(4) テーブル操作の簡略化

例えばダミー・セクションの使用により同一フォーマットを持った複数個のテーブル（コントロール・ブロックなど）に対して同一処理を行なうルーチンなど

4.3 外 部 記 号

独立にアセンブルされた他のプログラムから参照される記号，および他のプログラム中で定義されている記号（**外部記号**という）を参照するために次の命令が用意されている。下記につき機能，使用法を説明する。

① 外部から参照される記号

一般にENTRY，GLOBAL などの命令で定義する。

② 外部で定義されている記号の参照

EXTRN命令（identify external symbol）などで宣言される。

なお，ベース・レジスタなどで実行時に命令の修飾を行なうシステムでは，外部記号で定義されたデータの参照，制御，受け渡し（特に入口点（エントリ・ポイント）が複数の場合）が複雑になる場合が多いので実例でプログラム間での問題を含めて説明する。

指導上の留意点

連けい編集，プログラムのロードに関する事項（目的プログラムのフォーマット，ロード・モジュールのフォーマットなど）はメーカのマニュアルだけで不十分なことが多いので担当SEに情報の提供を依頼するのがよい。

第5章　マクロ言語

用　語　この章では次の用語を教える。

ポジショナル・マクロ，記号パラメータ，ヘッダ・ステートメント，プロトタイプ・ステートメント，モデル・ステートメント，トレイラ・ステートメント，キー ワード・マクロ，システム可変記号，セット可変記号

目　標

オープン・サブルーチン，テーブルの作成など，プログラム中で適宜マクロ機能を利用し，効果的なプログラミングを行なう能力を養う。

内　容

5.1　概　要

ここでは5.2以降の各論がよく理解できるよう，まずマクロの全体についてその機能の概要を説明し，更にマクロの種類およびマクロ中で用いられる可変記号の役割などを理解させる。

(1)　マクロの概要

マクロ命令は最初，まとまった機能をもつインストラクション・セットを定義しておき，これをソース・プログラム中にオープン・サブルーチンとして挿入する目的で考案されたが，現在ではマクロ定義のため，高度の記述能力を持つマクロ言語が開発され，その利用範囲も拡大された。

下記の特徴を参考にして，マクロ言語の概要を説明する。

(a)　オープン・サブルーチンによるコーディング作業の簡略化，コーディング誤りの減少

(b)　ファンクショナル・コーディングの標準化

(c)　マクロ定義の変更により，機能の拡張，変更などができる（拡張性，ゆう通性）

(d)　条件つきマクロ命令の使用により，用途に合った冗長部分のないプログラムが生成できる（ファイル・コントロール・プログラムなどで）

(e)　キー ワード・パラメータ，標準値の採用による，実パラメータ指定の簡略化（ルーチン，各種コントロール・ブロックなどで）

(2)　マクロの定義

(a)　マクロは次のような一連のステートメントによって定義されるがその機能，用法の概略を説明する。

①　マクロ・ヘッダ・ステートメント

②　マクロ・プロトタイプ・ステートメント

③ マクロ・モデル・ステートメント

④ マクロ・トレイラ・ステートメント

(b) マクロ定義は次の2通りの方法でプログラムに組込ませることができることを説明する。

① ソース・デック中で定義する

② マクロ・ライブラリ中にあらかじめ定義しておく

(3) マクロのタイプ

ポジショナル・マクロ，キーワード・マクロの機能，用法の違いについて説明する。

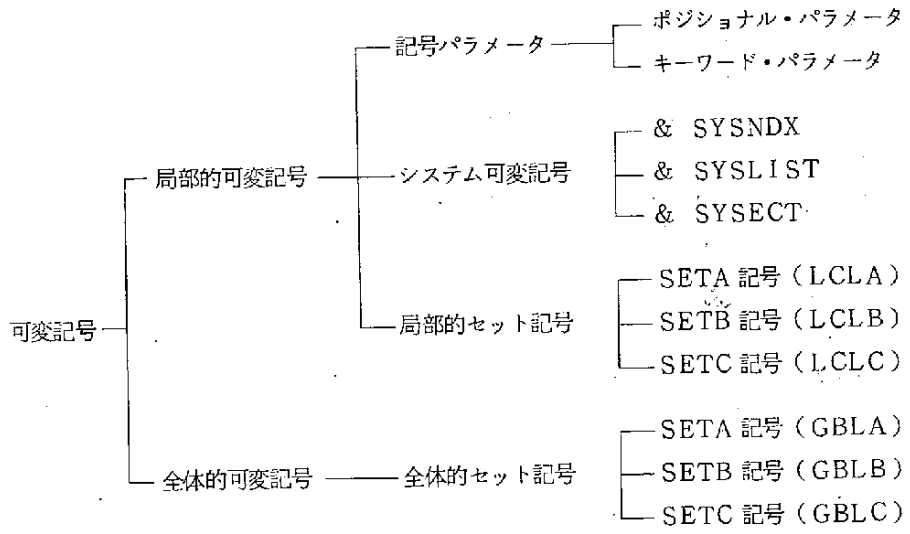
(4) マクロ・コール・ステートメント

このステートメントによりマクロ定義がソース・プログラムに展開される。

マクロ・コール・ステートメント中の実パラメータと定義中の仮パラメータの結合，命令のジェネレーションについてその概要を説明する。

(5) 可変記号

可変記号は次表のように分類されるがその機能や用法の概要について説明する（システムによって異なるのでそのシステムに合ったもので説明する，次表はHITAC EDOS アセンブラの例である）。



(6) ジェネレーションの変更

コンディショナル命令と可変記号が持つ値によって，生成される命令の数，出力順序が変更できること，およびそれらの使用例をあげ可変記号と条件つきマクロ・ジェネレーションのしくみを理解させる。

5.2 ポジショナル・マクロ

ここでは**ポジショナル・マクロ**の定義および参照の詳細説明を行なう（条件付きは5.3で行なうのでシステム可変記号，セット可変記号の説明は行なわない）。

下記の順序で文法，機能を説明する

① **記号パラメータ**

② マクロ定義

- **ヘッダ・ステートメント**
- **プロトタイプ・ステートメント**
- **モデル・ステートメント**
- **トレイラ・ステートメント**

③ マクロの参照

マクロコール・ステートメント無効パラメータ（実パラメータの省略），マクロのネストなどについても実習に応用できるよう具体例で説明する。

5.3 条件つきマクロ

条件つきマクロは実パラメータの値，セット可変記号の値などにより生成されるモデル・ステートメントの順序，数などを変更することができ用途に適した冗長度のないプログラムを生成できるので応用範囲が広くプログラム技法上非常に重要である。

詳細は説明をする前に次の事項について解説する。

① 条件にはどんなものがあるか

② プログラム中で条件はどのように設定されるか

③ 条件の判定はどんな命令でなされるか

④ プログラム生成の変更はどのようなしくみで行なわれるか（一般にインタプリティブに行なわれるので，実行時間などにも触れる）。

次に下記の項目について文法，機能，用法を説明する。

(1) セット可変記号

セット可変記号について，以下を説明する。

① セット可変記号の種類

② 割当てられるデータの形式

③ 記号の有効範囲（全体的か局所的か）

④ 記号の命名法

(2) システム可変記号

システム可変記号について以下を説明する。

① マクロ・コール・インデックス（& SYSNDX）

② カレント・コントロール・セクション（& SYSECT）

③ マクロ・オペランド・フィールド（& SYSLST）

なお，マクロのネスト時（マクロ定義中で他のマクロを参照する）のシステム可変記号の値についても具体例で説明する。

(3) セット命令

① 命令とセット可変記号の種類との対応

② 値の定義および割当

③ 初期値

④ サブストリング

他の可変記号の値のサブストリングの割当ては名札の生成などでよく使用されるので具体例で説明する。

(4) コンディショナル命令

次の命令について説明する。

① 条件により指定されたモデル・ステートメントまでスキップ（AIF）

② 無条件スキップ（AGO）

③ 条件により指定されたモデル・ステートメントまで戻る（AIFB）

④ 無条件に戻る（AGOB）

(5) その他の命令

① ノー・オペレーション（ANOP）

② マクロ・ジェネレーションの終了指示（MEXIT）

③ メッセージ・リクエスト命令（MNOTE）

④ デバグのための命令

・マクロ・トレース（MTRAC）命令

・ノー・トレース（NTRAC）命令

5.4 キーワード・マクロ

まず，**キーワード・マクロ**の特徴である

・標準値の採用

・実パラメータの省略

・任意な実パラメータ記述順序

についてその概要を説明し，次の事項につき文法，機能用法を説明する。ただし，キーワード・マクロとポジショナル・マクロの相異点はマクロ・コール・ステートメントのオペランドの形式，プロトタイプ・ステートメントのオペランドの形式が異なることおよびマクロ・オペランド・フィールド（&SYSLST）の値の参照ができないこと，だけなので用法を除いては次の2項のみの説明でよい。

・キーワード・マクロ・プロトタイプ・ステートメント

・キーワード・マクロ・コール・ステートメント

5.5 混合モード・マクロ

同一のマクロ定義の中でキーワード・マクロとポジショナル・マクロとの機能を同時に利用することもできる。

次の項について文法，機能，用法を説明する。

① 混合モード・マクロ定義

② 混合モード・マクロ・コール命令

指導上の留意点

(1) 説明には具体例を示すことが重要である。この具体例として，システムが提供しているマクロを採用するのも一つの方法である。

(2) 用語の定義をより明確化するために，用語，命令語は一部HITAC EDOSアセンブラのものを利用した。

ここで定義が不十分なものについては参考文献(2) を参照されたい。

第6章　実習のためのオペレーティング・システム

目　標

アセンブリ言語でプログラムを作成する場合，入出力マクロ命令等システムが提供している各種の機能の利用なしに，アセンブラ言語だけでプログラムを作成することは難しい。

この章ではプログラムの作成，アセンブル，連けい編集，実行といった一連の作業が，その基本的なものについてできるよう

- 順編成ファイルの定義
- 順編成ファイルに関するマクロ命令
- プログラムの実行に必要なシステム・マクロ命令
- 連けい編集プログラム
- ジョブ制御言語

の基本的なものについて，その記述を理解させる。

内　容

6.1　ファイル

順編成ファイルについて，下記の項目の機能，記述法を説明する。

① ファイルの定義
　（ファイル・コントロール・ブロック）

② 入出力マクロ命令
　システムにより種々の水準のマクロ命令が用意されているが，最も記述が簡単な最高水準のものが望ましい。

6.2　システム・マクロ

次のような機能を持つマクロについて機能，用法を説明する。

① コンソール・タイプライタへの出力およびコンソールからの入力

② ジョブ・ステップの終了の指定

③ 割込み処理ルーチンのアドレス指定
　割込み原因別に，ユーザの割込み処理ルーチンの先頭アドレスを示し，割込みが発生したとき，その処理ルーチン制御を渡すよう，コントロール・プログラムに指令する。

④ 割込みが生じた点に戻るためのマクロ

6.3　ジョブ制御文

次の文について記述法および機能を説明する。

① ジョブ記述文

② ジョブ・ステップ文

③ ファイル記述文

6.4 連けい編集プログラム

アセンブルからプログラムの実行までのステップを図示して説明し，次の項目について解説する。

① 機　能

下記に示すような連けい編集プログラムの機能を説明する

・セグメント構造の作成

・プログラム・ライブラリから必要なプログラムを選択し，自動的にモジュールに組込む（自動組込み）

・作成されたモジュールに名称を与える。

・未定義の外部記号を他のセクションから探し連結（結合）させる。

・処理内容を印刷する。

② 制御文

制御文の記述法およびその機能について説明する。制御文は次に示すような簡単な内容のものでよい。

・分割して作成されたプログラムから単一，セグメント構造のモジュールの作成

・プログラム・ライブラリからの自動組み込み

6.5 アセンブラの制御文

アセンブリを行なうために必要なジョブ制御文について説明する。

6.6 ユーティリティ・プログラム

プログラミングに必須なユーティリティ・プログラムの機能，使用法を説明する。必須なものとして次のようなものがある。

① ボリューム／ファイルのプリント

② ボリューム／ファイルのラベルのプリント

③ VTOCのプリント

④ プログラムのアップデイト

⑤ マクロ・ライブラリへの登録

⑥ 自動組込み用のプログラム・ライブラリへの登録

指導上の留意点

プログラムの実行に関する基本的なものについて理解が得られればよい。詳細については，適宜オペレーティング・システムの科目を参照すること。

第 7 章　実　習

目 標

実習を通して，アセンブリ言語による一連のプログラミング作業（プログラムの作成，デバグ，総合テスト，文書化，工程管理など）を遂行する基本的技術を習得させる。すなわち特定のコンピュータを対象にして

① アセンブリ言語，機械語命令を理解し，与えられた問題に対して，効果的なインストラクションを選択し，効率のよい（実行時間，主記憶占有容量など）ルーチンを組立てる能力

② プログラムの分割，マクロ（定義，参照）の使用に関する知識

③ 基本的なシステム・マクロ，データ管理機能の使用に関する知識

④ ジョブ制御言語，連けい編集プログラム，デバッギング・トゥールの使用に関する基本的知識

⑤ 与えられた問題に対して，問題の定義（外部仕様），プログラム設計（内部仕様），コーディング，デバッギング，総合テストの各段階をふんだ，一連のプログラミング作業を遂行する基礎能力を実習を通して修得させる。

内 容

以上に示すような問題についてプログラムの作成，実行を行なう。

7.1 データの形式と編集

固定小数点，浮動小数点，10 進パック形式……の各データ形式の特徴や相互の関係を理解させるために，あるデータ形式のデータを他のデータ形式に変換させる。

また数値データにコンマ，小数点，通貨記号，符号等を挿入する。

7.2 アドレス修飾およびくり返し（ループ）

表（テーブル，配列）に関連した問題が適当である。

① 探索（テーブル・ルックアップ）

逐次探索（リニア・サーチ），2 分探索（バイナリ・サーチ）など

② 内部ソート（インターナル・ソート）

主記憶上のデータを昇順，降順に並べ換える。

③ ハッシュ・アドレス方式による探索。

④ リスト構造を持ったテーブルの探索

なお，テーブルの作成に当っては今迄に学習した事項をできるだけ多く活用するよう心掛ける。

例えば

① 初期値を各種のデータ定義命令で与える。またデータ定義命令のオペランドにロケーション・カウンタの参照（＊）を書き，規則点に変化する定数テーブルを作成させる。

② EQU命令を用い，テーブルの要素数が変更されても，関連するEQU命令のオペランドのみ変更すれば変更に応じられるようにする。

③ EQU命令，ORG命令を使って作業領域，テーブルの重複使用をさせる。

④ 初期値をデータ定義命令で与えた場合と初期化ルーチンで初期値を設定する場合との相違について考えさせる。

7.3 サブルーチン

(1) 主ルーチンとの情報の授受

外部記号，COMMON領域，パラメータなどを通して主ルーチンと情報の授受を行なう。なお，今迄実習で作成したプログラムをサブルーチンに変換してもよい。

サブルーチンのリンケージに関する規則は一般にOSで規定されているが，規定のない場合，教師は実務上想定される規定を作って，それに従わせること。

(2) 再帰的（リカーシブ）プログラム

実務上での必要性はあまりないが，スタックなどによる作業領や返り番地（リターン・アドレス）の管理技法など有用である。N！の計算程度のものを出題する。

7.4 マ ク ロ

固定小数点データの倍精度演算命令，四捨五入などのマクロを作成する，また条件付マクロ命令ではビット・ストリングの転送（ビット・ストリングが語単位である時と，そうでない時ジェネレートされる命令群が異なる）などが考えられる。

7.5 事 務 計 算

① 金種別表や社員，株主，取引先などに関する各種統計表の作成

② マスタ，トランザクション・ファイルによる在庫や売掛金等の残高更新および小計，中計，……総計の計算

なお，以下の点に注意する。

- データ・エラー，演算結果の桁あふれなど異常状態で割込みを起こさせ，その処理ルーチンも作成させる。
- ファイルの終了時（EOF，EOD）などのOSに関連した処理ルーチンも作成させる。

7.6 綜 合 実 習

問題の規模としては卓上計算機程度の機能をもつ，簡単な計算向き言語のインタプリタなどを3～5人のチームで作成する。

実務に合わせた開発ステップ

① 分担，スケジュール決定

② 外部仕様書の作成

③ プログラム仕様書の作成

④ コーディング，デバッギング

⑤ 綜合テスト

をふませる。

指導上の留意点

(1) 実習は単一機能を持ったプログラムを作成，実行させる単体実習と，問題が与えられて，外部仕様の作成……綜合テスト，文書化までの一連のプロクラミング作業を行なってプログラムを作成させる綜合実習に分けて考えている。

単体実習（7.1データの形式と編集～7.5事務計算）はアセンブラやシステム・マクロ，ジョブ制御言語などの部分的な機能の理解，およびそれに関連したプログラムの作成，実行が目的であるが綜合実習（7.6綜合実習問題）は実務で必要とそれるプログラミング技法，作成工程の管理なども含んだ綜合的実習である。

したがって与える問題も実務上取扱われる問題に近いものが望ましい。

(2) 実習時間145時間中，単体実習に45時間，綜合実習に100時間を予定している。

単体実習は1題5～10時間を要するものを5～7題，前記実習指針に従って作成し出題する。

実習時間を有効に使用するため，各自が作成する上記5～7題のプログラミングが並行して行なえるよう，同時に全問出題することが望ましい。

全研修生に同一の問題を解かせてもよいが，研修生の理解度を広範囲に亙って知るため,10～20題出題し適宜選択させるのがよい。出題した各問題を少なくとも1人は選択するであろうし，難解の部分や教師が不適切な教え方をした箇所など質問や誤りとなって現れると思われる。そこで，解く問題の範囲が広ければ広い程質問や誤りも広範囲から現われ，前記の不適切さを実習で補うことができる。

(3) 単体実習でもプログラムの機能仕様書，流れ図は原則として作成させること，ただし簡単なものは流れ図だけでよい。

(4) デバッギング・トゥール（メモリ・ダンプ，トレーサなど）は使用するシステムに依存する部分が多いので，デバッグの仕方も含めて使用法を実習で十分指導することが望ましい。

特にトレーサなど時間や印刷用紙を多量に消費するものゝ使用に当っては，その乱用をさけるよう指導する。

(5) 総合実習は工程管理，文書化まで含めたプログラミング作業の綜合的な実習であるので，「ソフトウェアの生産管理」，「プログラム設計(II)」の科目終了後行なう。

(6) 総合実習は3～5人からなるグループを編成し，協同して一つのプログラムを作成するなど，実務におけるチーム編成と同一形式で実習を行なうことが望ましい。

また総合実習での入力データなどは実務でのデータ・ファイルを使用することも実務に近づけるという意味で有効であろう。

同様に用語，流れ図の表記法（特にボックス内の表記法），プログラム設計仕様書の書き方など，そこ（企業等）で定められている規定に従って行なわせることが望ましい。

(7) 総合実習で取扱う問題は教師が与えてもよいし，研修生が考えたものでもよい。問題の中には各種のプログラム技法が多く使用されるようなものが望ましいので，問題の規模が大きくなりがちであるが，1人当り150～200ステップ位のプログラムを作成させる。すなわち問題ではプログラム技法を優先にし，使い易さ（エラー・メッセージ，融通性に富んだ外部仕様）などはプログラムの大きさを小さくするため，犠牲にしても差支えない。

この場合，チーム中の上級情報処理技術者の分担作業は教師が代行する。

(8) 総合実習では，機能仕様書，流れ図が完成した時点で，その説明会を行なって長所や短所の指摘を行ない，他人の長所を取得し，自分の短所を矯正する機会を与えることが望ましい。

(9) 総合実習はプログラミング作業に必要な全項目の実習であり，文書化，プログラム作成の工程管理など省略してはならない。

(10) 総合実習終了後，この実習に関するレポートを提出させる。またシステムにアカウンティング・ルーチンが提供されていれば，モジュールや各個人ごとのコンピュータ使用時間などの統計を作成し反省の材料に供するのもよい。

参考文献

HITAC　8000シリーズ

(1) ハードウェア・マニュアル

H－8450形　処理装置，8450－2－001

(2) プログラム・マニュアル

EDOS－MSO

アセンブリ・システム，8000 － 3 － 003

(3) プログラム・マニュアル

EDOS－MSO

FCPとエグゼクティブ連絡マクロ，8000 － 3－005

(4) プログラム・マニュアル

ED・OS－MSO

コントロール・システム，8000 － 3 － 002

FACOM　230 － 45／55

(5) FACOM　230 － 45 S

ハードウェア総合解説編，EX－011 － 1 － 1

(6) FACOM 230 OSⅡ
FASP解説編Ⅰ, 46 EX0811−1
(7) FACOM 230 OSⅡ
FASP解説編Ⅱ, 46 EX0812−1
(8) FACOM 230 OSⅡ
解 説 46 EX0411−2
(9) FACOM 230 OSⅡ
ジョブ制御文法編 SP−045−1−3
(10) FACOM 230 OSⅡ
データ管理文法編 SP−043−1−3
(11) FACOM 230−45
LIED 文法編 SP−181−1−3
(12) FACOM 230 OSⅡ
システムユティリティ文法編 SP−113−1−3
(13) FACOM 230 − 45
LIBE 文法編 SP−131−1−4
(14) 佐久間紘一, 電子計算機基礎講座②: 計算機とプログラム, 共立出版 (株)
(15) 日本電信電話公社編, 図説電子計算機とプログラミング, 電気通信協会
(16) 日本電信電話公社編, プログラミング (入門編, 基礎編, OS編), 電気通信協会
(17) IVAN FLORES, COMPUTER SOFTWAR Programming System for Digital Computer, PRENTICE−HALL, INC.

プログラム設計(II)

目　　次

第1章　プログラム構造とモジュール設計 …… 498

1.1　モジュール構造の設計 …… 498

1.2　モジュールの設計 …… 499

1.3　オーバレイ構造の設計 …… 500

1.4　動的構造の設計 …… 501

1.5　再入可能プログラムの設計 …… 503

1.6　再配置可能プログラムの設計 …… 504

1.7　実　習 …… 504

第2章　データ構造 …… 505

2.1　表 操 作 …… 505

2.2　線型リスト …… 506

2.3　木 構 造 …… 507

2.4　その他の構造 …… 508

2.5　実　習 …… 508

第3章　分類と組合せの技法 …… 509

3.1　分類の技法 …… 509

3.2　組合せの技法 …… 511

3.3　実　習 …… 512

第4章　探索の技法 …… 513

4.1　逐次探索法と二分探索法 …… 513

4.2　リスト構造における探索 …… 514

4.3　実　習 …… 514

第5章　ファイル処理 …… 515
5.1　ファイル編成とファイル・アクセス法 …… 515
5.2　ファイル処理の一般的な技法 …… 516
5.3　順編成ファイルの処理 …… 517
5.4　索引順次編成ファイルの処理 …… 518
5.5　直接編成ファイルの処理 …… 519
5.6　区分編成ファイルの処理 …… 520
5.7　実　習 …… 520
第6章　オペレーティング・システムの利用 …… 521
6.1　プログラム管理 …… 521
6.2　タスク管理 …… 521
6.3　主記憶管理 …… 522
6.4　サービス・プログラム …… 522
6.5　回復管理 …… 524
6.6　操作員との対話 …… 524
6.7　実　習 …… 525
第7章　オンライン・システムのプログラミング …… 526
7.1　回線制御プログラム …… 526
7.2　メッセージ処理プログラム …… 527
7.3　サポート・プログラム …… 528
7.4　オンライン・プログラム・テスト …… 528
7.5　オンライン・プログラムの管理 …… 528

科目「プログラム設計(II)」

教育の目標

この科目では，研修生が専門的なプログラム設計の技法を身につけることをねらいとする。

「プログラム設計 (I)」ではプログラム設計の基本的知識と基本的方法を中心に教えたのに対し，本科目ではより専門的なプログラム設計技法，およびそれに関連するプログラミング技法について教える。

そのためにモジュールの設計，データ構造，オペレーティング・システムの使い方などが本科目の重点となる。

本科目では，講義はできるだけ具体的な内容としなければならないが，しかしそれによって研修生が体系的な技法の修得を得ることをさまたげてはならない。

本科目の指導にあたり留意すべき点は以下の通りである。

(a) 本科目では，アセンブリ言語，オペレーティング・システムの知識を前提としているので，研修生のレベルが不十分の場合には，あらかじめそれらを補うことが必要である。

(b) 説明にあたっては，できるだけコーディングの具体例を多く示すようにする。ただし，実例としてはあまり複雑なものや大きなプログラムは避けるべきである。

(c) マクロ命令やジョブ制御言語など具体的な機種と具体的なオペレーティング・システムを想定することがとうしても必要になる。これらは実習用機種に合わせるようにするのがよい。そのための参考資料として，関連メーカのオペレーティング・システムのマニュアルを利用するとよい。

時間配分

章	履修時間(時間)	
	講　義	実　習
1. プログラム構造とモジュール設計	4	8
2. データ構造	6	10
3. 分類と組合せの技法	6	6
4. 探索の技法	2	6
5. ファイル処理	6	8
6. オペレーティング・システムの利用	10	12
7. オンライン・システムのプログラミング	6	
合　計	40	50

第1章　プログラム構造とモジュール設計

目　標

プログラム構造とモジュール構造について学ばせ，研修生にプログラム設計の基本を身につけさせ，簡単なプログラム設計を行なえるようにすることを目標とする。

内　容

1.1　モジュール構造の設計

プログラムの書き方に関する一般的心がまえとしてプログラム開発の効率を高めるために，モジュール化によるプログラミングの重要性について復習し，研修生がこれを十分に理解していることを確認する。

① モジュール化によるプログラミングの利点は次のとおりである。

- 正確なプログラム設計を容易に行なうことができる。
- コーディングとデバッギングが，より能力の低いプログラマを使って短期間で完成できる。
- 総合テストをより短期間で行なえる。
- プログラム開発コストと工数を大巾に節減できる。
- プログラムのメンテナンスが容易になる。
- プログラムとプログラミングの標準化が行ないやすく，作業や技術の蓄積が容易になる。

② モジュール化によるプログラム設計は次のようにして行なう。

- プログラムの流れを制御する部分と処理を行なう部分を分けて別のモジュールにする。
- 入出力部分を別のモジュールにする。
- 開始処理，終了処理，エラー処理などの特別の処理部分を分けて別モジュールにする。
- 共通機能をまとめて一つのモジュールにする。
- 重要な定数の定義は1カ所で行なう。
- モジュール分けはシステム機能（外部機能）ではなく，プログラム機能（内部機能）によって行なう。
- 1モジュールの大きさはソース・ステートメントで200～400ステートメント程度がよい。

なお，モジュール化によるプログラミングを行なう際は，標準化と文書化が前提となることを確認する。

モジュール構造の設計手順として次のことを説明する。

① 与えられたシステム機能と諸条件を分析し，プログラム機能として再編成する。

② さきに述べたモジュール化の方法を考慮しながら，モジュール構造図を書く。これは各モジュールの関係（すなわちリンケージ）を明示しながら，全体の構成がどうなっているかを示す図で，各

モジュールは四角形の箱で示し，CALLを矢線で表わしたものである。

③ モジュール構造図をより単純化し，よいものに改良する。改良のためには次の六つが基準となる。

- モジュールの数を減らす。
- 構造を簡単にする。
- 処理効率を高める。
- 所要主記憶容量を少なくする。
- 各モジュールの中を単純にし，小さくする。
- 既成モジュールを使えるようにする。

④ 各モジュールの機能を明確にする。

⑤ でき上ったモジュール構造が当初の与えられた諸条件を満足することを確認する。

なお，モジュール化が不可能な場合でも，準モジュール化を行なうことによってかなりの効果をあげることができる。

さらにワーニエ方式やストラクチャード・プログラミングについての説明を加えてもよい。

1.2 モジュールの設計

プログラム・モジュールの設計に関して必要な次の項目について理解させる。

- そのモジュールとして必要な機能と諸条件の確認
- モジュール名
- 入口と入り方，およびこのモジュールに入るプログラムからもらうパラメータ
- 出口ともどり方，およびもどすパラメータ
- 他のモジュールへの呼び方ともどり方，およびパラメータ
- このモジュールで定義する外部名
- このモジュールの属性（再入可能か，再使用可能かなど）
- 受取ったパラメータのデータ・チェック
- モジュールの大きさ
- 記述言語

なお，内部名のつけ方，コメントの入れ方，レジスタの退避のしかたなど，名モジュールに共通のことは別に定めておく。

モジュールの設計とは直接関係ないが，ここで，CALL，RETURN，SAVE，RESTOREなどのマクロの使い方を具体的な機種において説明することが望ましい。また異言語によるモジュール間のリンケージで特別の注意が必要な機種の場合には，それについても説明する。

DSECT(ダミーコントロール・セクション）について説明しておくのが望ましい。なお研修生がもし各モジュールの最初で行なう汎用レジスタのSAVEについて十分理解していないときは，そのメカニズムについて理解させておかなければならない。

1.3 オーバレイ構造の設計

処理プログラムのための主記憶容量が不足した場合の解決法として，分割処理（または，プログラムの分割）とオーバレイ手法とがあることを復習する。

またオーバレイ手法には，プランド・オーバレイ構造（または単にオーバレイ構造）による方法と，ダイナミック・オーバレイ構造（または動的構造）による方法があることも復習する。

オーバレイ構造の設計法として，次の手順を学ばせる。

(1) オーバレイが可能なモジュール群の確認

オーバレイの対象となりうる各モジュールを明確にし，それぞれの大きさを調べる。

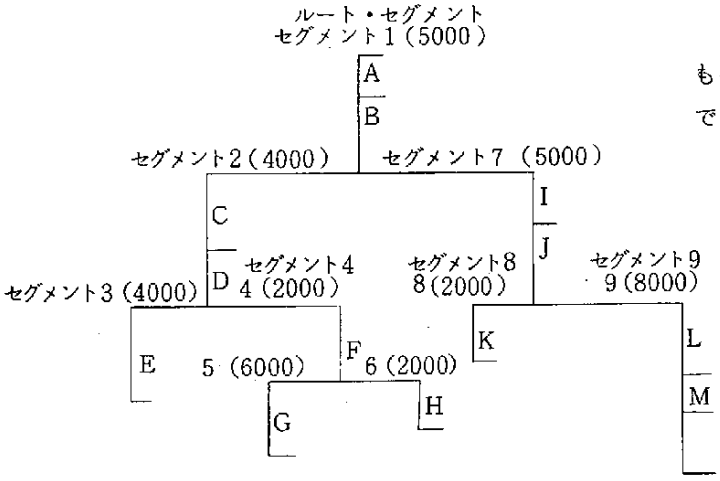
(2) 各モジュール間の制御の流れを調べる

各モジュール間で制御がどういう順番でどう流れるか，他モジュールへの参照がどう行なわれるかを調べる。結果は図で表わす。

(3) オーバレイ構造の決定

次のことに注意しながら，オーバレイ構造を決定する。

- 構造の上位レベルのモジュールへの参照は直接的に容易に行なわれるが，下位レベルに対する参照は間接的で時間がかかる。
- オーバレイされると，そのモジュールは主記憶になくなるから，何度も使用するモジュールは，その間はなるべくオーバレイさせないようにする。
- オーバレイを1回行なうと，かなりの時間を必要とする（これは具体的な例で実際の計算をさせるとよい）ので，オーバレイの回数は少ない方がよい。したがって，許された主記憶容量をできるだけ有効に活用する。



もっとも長いパスは（1, 7, 9）で18,000バイトである。

図1－1 オーバレイ構造の図の例

・以上の結論として，モジュール間の流れの制御を行なうモジュール（主プログラム）と，共通エリアはルート・セグメントに入れるべきである。でき上ったオーバレイ構造は図で表わす。最後にその構造で目的を十分に満足することを確認する。

(4) 処理時間の推定

決定したオーバレイ構造を用いて処理を行なう場合の処理時間推定を行ない，それが支障のないことを確認する。

なお，設計には直接関係ないが，オーバレイ構造を採ったモジュール間の呼出しのためのマクロ（CALL，SEGLD，SEGWTなど）および連係編集でのオーバレイ指定ステートメント（OVERLAY，INSERTなど）の説明を具体的な機種について行なうことが望ましい。

また仮想記憶とページングを行なう機種の場合についても説明することが望ましい。

1.4 動的構造の設計

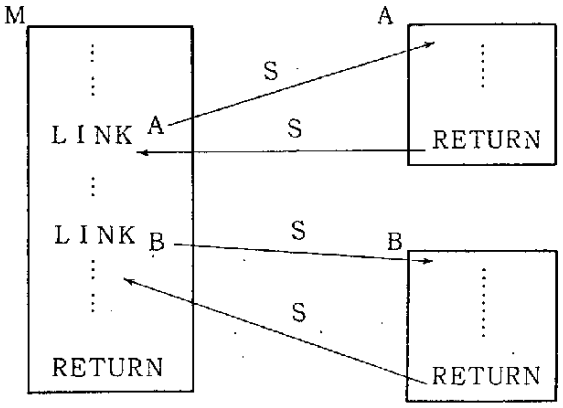
動的構造は，オーバレイされるモジュールの数が多く，実行順序があらかじめわからない時に用いられることを復習する。

動的構造の設計手順は，オーバレイ構造の場合とほとんど同じである。

動的構造を使用する際に必要な次のシステム・マクロについて，具体的な機種を想定して説明する。

(1) LINKマクロ

LINKマクロはCALLマクロとよく似ているが，ただ制御を渡す相手のモジュールが別のロードモジュールのものであるか，あるいはさらに主記憶になくてもよいという点が違っている。(図1－2)

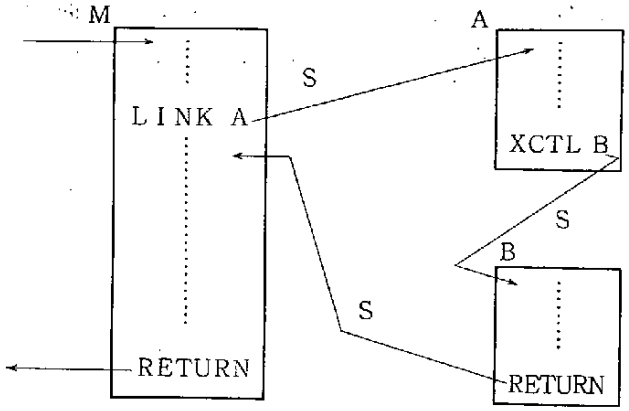


Sはスーパーバイザの介入を示す

図1－2　LINKマクロによる制御の渡し方

(2) XCTLマクロ

XCTL（（トランスファ・コントロール）マクロは，LINKマクロとよく似ているが次の点が違っている。(図1－3)



Sはスーパーバイザの介入を示す

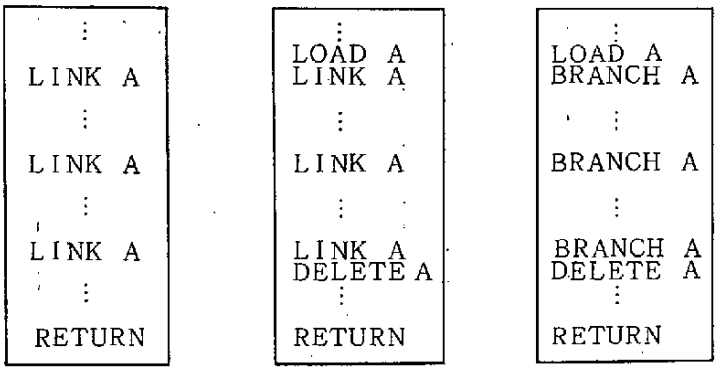
図1－3　XCTLマクロによる制御の渡し方

- 制御のレベルはXCTLを出したモジュールと同じになる。
- XCTLマクロを出したモジュールのエリアは解放される。

(3) LOADマクロ

LOADマクロはモジュールをフェッチする点でLINKとよく似ているが，次の点が違っている。

- LOADマクロでフェッチされたモジュールは，DELETEマクロが出されるか，あるいはそのタスクが終了するまで主記憶のなかにとどまっている。
- したがって，LOADマクロの実行の結果かえされる入口のアドレスを使ってBRANCH形式の命令を使うことができる。この場合はスーパーバイザの介入がなく処理時間が早まる。(図1－4)



(a)　(b)　(c)

(a) では，モジュールAを毎回フェッチしなければならない可能性がある。
(b) では毎回フェッチする必要はなくなる。
(c) ではスーパーバイザの介入なしにモジュールAに制御を渡せる。

図1－4　LOADマクロとDELETEマクロの使い方

(4) DELETEマクロ

DELETEマクロはLOADマクロでフェッチされたモジュールのエリアを解放する。

なお，以上のマクロに関連して，BRANCHタイプの分岐とATTACHマクロについて説明をしておくことが望ましい。

1.5 再入可能プログラムの設計

再入可能プログラムは，ふつうのプログラムに比べてプログラムを複雑にし，時間がかかるので，必要な場合以外は作らない方がよいことをまず理解させる。

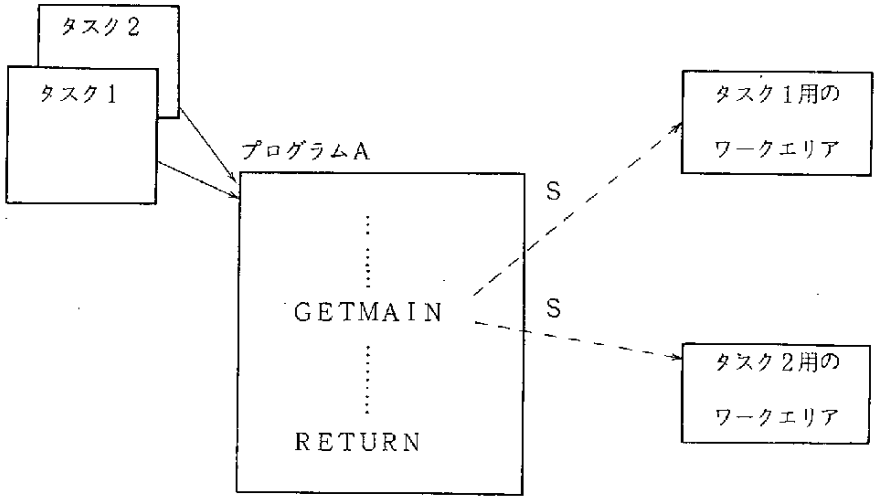
再入可能プログラムでもっとも簡単に作る方法は，ワークエリアをすべて自タスク内にあらかじめ設定しておいて，レジスタを介してそれを使うようにし，かつ再入可能プログラムから直接他のプログラムを呼ばないようにする方法である。

一般的には，再入可能プログラムを作るためには，GETMAINマクロを使わなければならない。GETMAIN，およびFREEMAINマクロについて具体的な機種を想定しながら説明する。

(1) GETMAINマクロ

プログラム中でGETMAINマクロを出すと，そのプログラムを使用しているタスク用のワークエリア（およびバッファ）を動的に確保してくれる（図1－5）。

このエリアはFREEMAINマクロが出されるか，あるいはタスクが終了するまで解放されずに残る。



Sはスーパーバイザの介入を示す

図1－5　GETMAINマクロ

(2) FREEMAINマクロ

FREEMAINマクロは，GETMAINマクロによって確保したエリアを開放する。

1.6 再配置可能プログラムの設計

プログラムを再配置可能に作ることは，きわめて重要なことである。

再配置可能プログラムは，ベースレジスタを使用し，かつ連係編集が可能な機種ではきわめて容易である。この場合にはアドレス定数をVタイプで作成するか，あるいは外部名として作成しておけばよい。もし研修生がベースレジスタの使い方，および外部名の連係編集のメカニズムを十分理解していないと判断される場合は，これらについて具体的な機種を想定して説明し，完全に理解させることが望ましい。

1.7 実　　習

具体的で単純な問題により，再入可能プログラムのコーディングをさせる。また，説明した各マクロ命令を使用してコーディングさせる。

指導上の留意点

実習機械における具体的なマクロ命令について説明する。

第2章　データ構造（data structure）

用　語　この章では次の用語を教える。

表，線型リスト，スタック，後入れ先出し法，プッシュダウン，ポップアップ，ポーランド記法，待ち行列，先入れ先出し法，木構造，網構造，鎖リスト，環構造，塵集め，データ構造

目　標

いろいろなデータ構造について学ばせ，研修生が簡単なデータ構造を扱えるようにすることを目標とする。

内　容

2.1　表　操　作

本節ではいろいろの表操作について学ばせる。

(1) 表の利用範囲

表（table）の代表的な利用方法として，次の三つについて説明する。

① データ表

関数値を表形式で表わしたものである。ふつうは変数を一定間隔にとり，そのときの関数値を表として配列し，とり出しやすくしたものである。プログラム上，このように配列した表を配列（array）といい，何番目の要素かを指定して参照するようになっている。この場合，表の変数の数をその配列の次元という。次元が2つ以上の配列の場合には，各次元ごとに何番目の要素であるかを指定して参照する。

② 演算結果または出力表

統計的データのクロス集計などを行なうとき，演算結果を表にしておくと便利である。

これはそのまま出力表としても利用できる。

③ 制御表

システムの状態やプログラムの状態を表として表わし，システムやプログラムの進行を制御するのに用いる。この場合には状態やプログラムのアドレスなどが表の中に入れられる。

(2) 表　操　作

表の定義のしかた，および表のなかの任意の項目のとり出し方について，高次言語（たとえばFORTRANとPL／I）およびアセンブリ言語の場合をそれぞれ説明する。

① FORTRANの場合

DIMENSION文，EQUIVALENCE文の使い方，添字を使った配列のなかの特定の要素

の参照のしかた，およびDOループを用いての表の要素の値を順番に加工する方法を例を用いて説明する。

② PL／I の場合

配列の宣言のしかた，変数の範囲，表演算のしかた，

3 × A

3 × A（＊，J）

などについて説明する。

③ アセンブリ言語の場合

表の定義のしかた，主記憶上でのN次元配列の1次元配列への展開のしかた（図2－1），およびインデックス・レジスタを用いて表の要素を順番に加工する方法などについて，例を用いて説明する。

A(1,1)	A(2,1)	A(3,1)	A(4,1)	A(2,1)	A(2,2)	A(2,3)	A(2,4)	A(3,1)	A(3,2)	A(3,3)	A(3,4)

図2－1　A（I，J）の1次元への展開の例

(3) 多段表操作

表の大きさが非常に大きいとき，あるいは変数の値に切れ目があり，かつ探索（第4章探索の技法参照）を行なわなければならないときには多段表操作を行なう。ここでは多段表の概念を教えるに止め，その探索法は第4章で教える。制御表についても同様である。

2.2 線型リスト

一般にいくつかの項目が相互にポインタや位置関係などで関連し合った構造をリスト（list）という。**線型リスト**（linear list）とは，要素の数が動的に変化する1次元のリストで，アクセスがその両端に限られているものである。線型リストのうち，スタックおよび待ち行列について学ばせる。

(1) スタック

スタック（stack）とは，線型リストのうち，一方の端だけからしか取出しを行なわないものである。スタックでは後から入れたものから先に取出すので，このような取出し法を**後入れ先出し法**（last in first out，LIFO）という。スタックに入れることを**プッシュダウン**（push down），スタックから取出すことを**ポップアップ**（pop up）という。

スタックの操作には，一次元配列の位置と，その終り位置を使う。実際のスタックへの追加ととり出しのしかた，およびそれらの際にチェックしなければならないことを具体的に説明する。

スタックを使うと便利な場合の例として，**ポーランド記法**（poland notation）およびコンパイラへの利用法を説明する。

(2) 待ち行列

項目を追加する端と，取出す端が固定的に定まっている線型リストを**待ち行列**（queue）という。待ち行列では，先きに入れたものから先きに取出しを行なうので，このような取出し方を**先入れ先**

出し法（first in first out, FIFO）という。

1次元配列を用いて待ち行列の操作を行なうには，項目を追加する位置と取出す位置を用いる。

実際の待ち行列への追加ととり出しのしかた，およびそれらの際にチェックしなければならないことを具体的に説明する。待ち行列を使うと便利な例として，入出力動作の制御への利用法を説明する。

2.3 木構造

二つのデータ項目の間の関係で，一方が他方に従属している場合，後者を親，前者を子という。

どの項目もたかだか一つしか親を持たないような項目間の関係を**木構造**（tree structure）という（図2－2）。

木構造においては，それを木にみたて，関係を示す線を枝（branch），枝の集まっているところを節（node），枝の入っていない節を根（root），枝の出ていない節を葉（leaf）という。

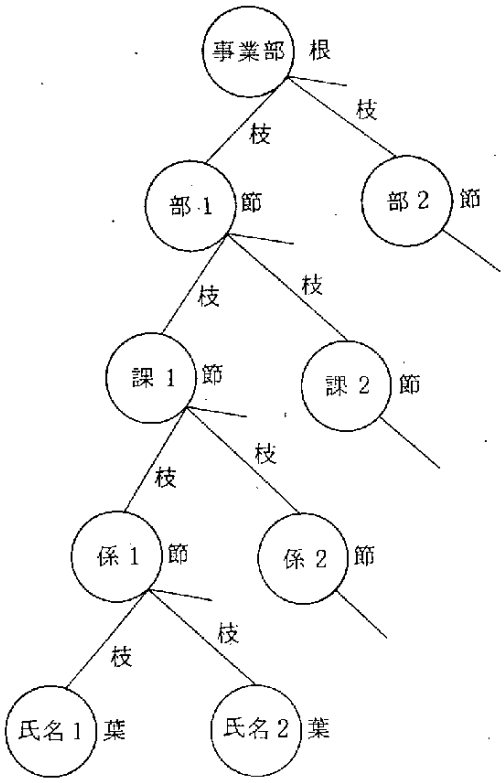


図2－2　木構造の例

記憶装置上で木構造を表わすには，根・節・葉ごとに表2－1(a)のような要素を考え，これを1次元の表かたち(同表(b))のように表わせばよい。なお同レベルの子のことを兄弟，兄弟のうち左から長兄，次兄というようにふつうの親子関係を表わすいい方をすることがある。

木構造をポインタで表わした場合すなわちリスト構造で表わした場合には，構造の変更はポインタを変更することによって行なわれる。

表2－1　記憶装置上で木構造を表わす方法の例

(a)　記憶装置上の木構造の要素の例

要　素	次弟へのポインタ	長子へのポインタ

(b)　記憶装置上で木構造を表わす表の例

アドレス			
1	乗用車事業部		2
2	エンジン製造部	7	3
3	組立第1課	8	4
4	第1係	9	5
5	田　中	6	
6	佐　藤	…	
7	車体製造部	…	…
8	組立第2課	…	…
9	第2係	…	…

木構造を表わすのに，下向きのポインタだけでなく，上向きのポインタを同時に使う場合もある。

木構造において，任意の要素を取出す方法を説明する。

固定的な木構造を扱かうことのできる高次言語（COBOL等）について説明する。

木構造を使うと便利な例について説明する。

2.4　その他の構造

その他の構造として，**網構造**（network structure），**鎖リスト**（chained list，linked list），**環構造**（ring structure）などについて，それらの概念，および利用すると便利な例に関して説明する。また，**塵集め**（garbage collection）の概念についても説明する。

また，これらいろいろな**データ構造**とデータベースとの関係についても説明する。

2.5　実　習

リスト構造を使ったプログラムを作成させる。

指導上の留意点

実習機械における具体的なマクロ命令に関する説明を行なうようにする。

第3章　分類と組合せの技法

用　語　この章では次の用語を教える。

キー比較法，選択法，交換法，そう入法，マージ法，配置法，ブロック分類法，数字分類法，アドレス計算法，バランス型分類法，アンバランス型分類法，カスケードソート，ポリフェーズソート　オッシレーティングソート

目　標

分類と組合せの技法について学ばせ，それらの特徴について学ばせるとともに，内部分類のプログラムが作れるようにすることを目標とする。

内　容

3.1　分類の技法

(1) 分類の重要性と基本的用語の確認

大量データ処理における順次処理の利点を，スペースと時間の節減という観点からまず説明する。

順次処理においては，入力，処理，および出力の各段階において，次の観点からの分類の必要性を確認する。

- 入力段階……ダブリチェック，順序チェックなど。
- 処理段階……突き合わせを行なう各ファイルの順序を合わせること。
- 出力段階……明細表における項目の印刷の順番を合わせること。

分類の技法に関係して，次の基本的用語の意味を確認する。

- 分類と組合せ（併合）
- 照合順位
- 昇順と降順
- シーケンス（系列）
- パス
- キー（コントロール・フィールド，またはコントロール・ワード）
- 内部分類と外部分類

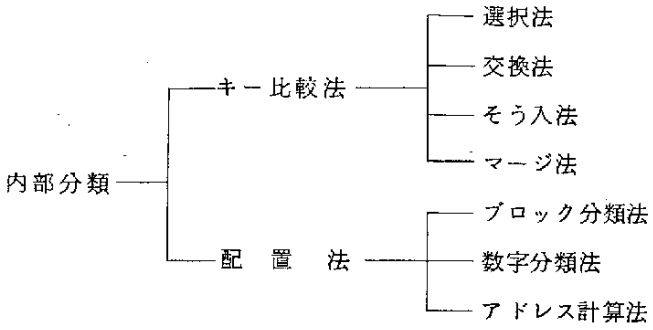
(2) いろいろな内部分類の技法

分類とは，たくさんのデータ（またはレコード）をあるキーの照合順位にしたがって並べかえることだが，データの量が少ない場合は主記憶のなかで行なわれる内部分類による。データの量が多いときは，内部分類すなわちシーケンス作成と，外部分類すなわちシーケンス数減らしとの併用によっ

て行なわれる。本節では内部分類の技法を教える。

代表的な内部分類の技法は，表3－2のように分けて考えることができる。

表3－2　代表的な内部分類の手法



以上のそれぞれの方法について，例を示しながらそのアルゴリズムを説明し，比較・転送の回数・ワークエリアの大きさなどについても注意させる。

(3) キー比較法

キー比較法（key compare technique）とは，二つのデータのキーを比較し，その比較結果に応じて分類を行なっていく方法である。

選択法（selection sort）とは，キーの値の最小（または最大）のものから順番に取出していく方法である。

交換法（exchange sort）とは，二つのデータのキーを比較し，順序が逆の場合に交換を行なうことをくりかえしていく方法である。

そう入法（insertion sort）とは，キーの値が順番として適切なところにそのデータをそう入していく方法である。

マージ法（merge sort）とは，主記憶上ですでに分類された二つのシーケンスから一つのシーケンスを作ることをくりかえしていく方法である。

(4) 配　置　法

配置法（allocation technique）とは，キーの値に応じてそのデータを所定のアルゴリズムで配置しながら分類を行なっていく方法である。

ブロック分類法（block sort）とは，キーの値に応じていくつかのブロックを定めておき，それらへの区分けを行なうことをくりかえしながら分類していく方法である。

数字分類法（radix sort）とは，キーを構成する桁につき，数字の値によって異る位置に配列することをくりかえしていく方法である。

アドレス計算法（address calculation sort）とは，キーの値に応じてそのデータを置く位置をあるアルゴリズムで求めて配置していく方法である。

以上のそれぞれの手法についても，例を示しながら方法を具体的に説明する。

いろいろの内部分類の手法で必要とするワークエリア，パス回数，比較・転送および交換の回数は表3－3のとおりである。

表3－3　いろいろの内部分類の効率の比較

		ワークエリアの大きさ	パスの回数	比較の回数	転送の回数	交換の回数
キー比較法	選択法	N	N	$N^2 - N$	N	－
	交換法	O	N（最大）	N（N－1）／2（平均）	－	N(N－1)／4
	そう入法	OまたはN	N－1	$(N^2 - N)/4$	－	$(N^2 - N)/4$
	マージ法	2N	$\log_2 N$	$N(\log_2 N)$	$N(\log_2 N)$	
配置法	ブロック分類法	NB／n	Dn	ND	ND	
	数字分類法	NB	D	ND	N（D＋1）	
	アドレス計算法	2.5 N	1	1.33 N	1.66 N	

N ： 項目の数

B ： 数の基数

n ： グループ（ポケット）あたりに割当てられる桁数

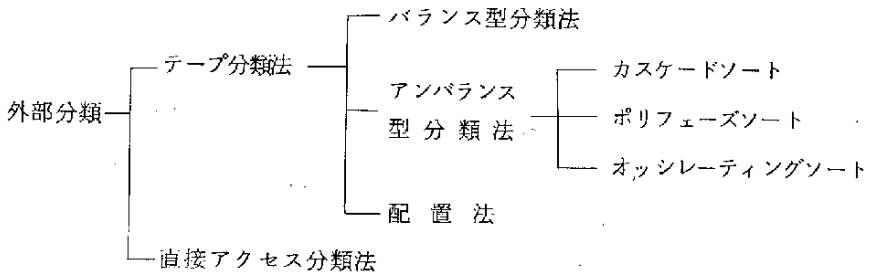
D ： キーの桁数

3.2 組合せの技法

(1) いろいろな外部分類の技法

外部分類の代表的な手法は表3－4に示すとおりである。

表3－4　代表的な外部分類の手法



以上のそれぞれの手法について，例を示しながらそのアルゴリズムを説明し，長所や効率についても理解させるようにする。

(2) **バランス型分類法** (balanced sort)

同じ数の入力テープ装置と出力テープ装置を用いるものである。

(3) **アンバランス型分類法** (unbalanced sort)

カスケードソート (cascade sort)とは，N－1本のテープを入力とし，1本のテープに出力する方法である。

ポリフェーズソート (poliphase sort) では，カスケードソートと同じように進むが，1つの入力テープが空になったとき，それを出力テープとして組合せをしていくものである。

オッシレーティングソート (oscillating sort) は，内部分類と外部分類を同時に行ないながら，N本のテープでつねにN－1ウェイの組合せを行なう方法である。

(4) 直接アクセス分類法

直接アクセス分類法としては，装置の特性を活かして，オーダの数を増やして組合せを行ない，効率を高める方法と，キーとポインタとで分類を行なう方法などがある。

3.3 実　習

内部分類のプログラムを作成させる。

指導上の留意点

実習機械におけるユーティリティ・プログラムによる分類組合せプログラムの使い方とタイミング・テーブルについても説明することが望ましい。

第4章　探索の技法

用　語　この章では次の用語を教える。

二分探索法，リスト構造

目　標

探索の技法について学ばせ，二分探索法のプログラムを作れるようにすることを目標とする。

内　容

4.1　逐次探索法と二分探索法

多数のデータ項目のなかから，目的とするキーの値を持つデータ項目を探して取出すことを探索という。ここでは逐次探索法と二分探索法のアルゴリズムと，これら探索法と表操作（2.1参照）との関連性を教える。また探索の効率の評価法としての照合日数（平均，最大）およびデータ構造の変更のしやすさなどについても学ばせる。

(1) 逐次探索法

逐次探索法（Sequential　Search）とは，探索対象となるデータ項目を順次調べていく方法である。

逐次探索法も，条件によっては次のようにきわめて効率の良いものとなる。逐次探索法の最大の利点は，探索対象項目をどういう順序で配列しておいてもよいという点である。次について，例を示しながら理解させる。

(a)　きわめて多数のデータ項目を探索したい場合

探索対象項目群と，探索したい項目群のそれぞれを同一キーで分類しておいてから探索を行なう。

(b)　探索対象の探索確率がいちじるしく異なる場合

探索対象項目群を探索確率の高いものから順に配列しておいて探索を行なう。

探索すべき項目が，探索対象群のなかに存在するかどうか不明の場合には，逐次探索法はもっとも効率の悪い探索法となる。

(2) 二分探索法

二分探索法（binary search）とは，探索対象群を探索キーによって分類して配置しておき，先ずその中央の項目と比較して，目的とする項目が前半にあるか後半にあるかを判断し，これをくりかえしていく方法である。

二分探索法の変形として，累積探索確率が$1/2$になる点で分けていく方法が考えられる。たとえばキーの値に対して等しい確率で探索が行なわれるとすれば，項目群の中央ではなくキーの値が半分の

もので区分していく方法である。

(3) ブロック探索法

ブロック探索法とは，キーの値で分類された項目群をいくつかのブロックに分けておく（等分の必要はない）。各ブロックの最終項目（または最初の項目）との比較を行ないながら，目的とする項目がどのブロックにあるかを求めていく方法である。

ブロック探索法は，多段表（2.1 (3)）形式の多段ブロック探索法に容易に拡張できる。多段ブロック探索法は，索引順編成ファイルへのランダムアクセスの方法として用いられている。

4.2 リスト構造における探索

リスト構造（list structure）における探索は，各節から平均3.6本の枝を出すようにするともっとも効率がよくなる。また，両方向向きのポインタをつけたり，環構造にすることにより，効率が高まる。以上のいろいろの探索法の効率を比較すると表4－1のようになる。

表4－1　いろいろの探索法の効率の比較

探索方法	平均照合回数	探索の効率	更新の効率	注
逐次探索法	$\frac{N+1}{2}$	×	○	ふつうの条件の場合
二分探索法	$\log_2 N$	○	×	
ブロック探索法	$\frac{N+s^2}{2s^2}$	△	△	
リスト構造	$1.24 \log_2 N$	○	○	各節から平均3.6本の枝が出ていてもっとも効率の良い場合

N：データ項目の総数　　s：ブロックの大きさ

4.3 実　習

具体的な条件（項目の最大数など）を与えて，二分探索法によって探索を行なうプログラムを作らせる。

指導上の留意点

説明はなるべく具体的に行なうようにする。

第5章　ファイル処理

用　語　この章では次の用語を教える。

あふれ域，マスタ索引，シリンダ索引，トラック索引，ランダマイズ技法

目　標

ファイル処理の技法について教え，それによって索引順編成ファイルの作成と入出力を行なえるようにする。

内　容

5.1　ファイル編成とファイル・アクセス法

ファイル処理の方法を学ぶにあたって，まずファイル，ファイル編成およびファイル・アクセスに関する次のような基本事項について復習する。ただし研修生のレベルによっては省略してもよい。

(1) ファイルの基本概念

ファイルの基本概念として，次のことの理解を確認する。

- ファイル，ブロック，レコード，キー
- 磁気ディスク装置，シリンダ，トラック，レコード番地
- シークタイム
- ボリューム
- ラベル，ボリュームラベル
- レコードの形式
- 磁気テープ，磁気ディスクの初期設定
- ハードウェアによるチェック

(2) ファイル編成

次の四つのファイル編成についての理解を確認する。

- 順編成
- 直接編成
- 索引順編成
- 区分編成

(3) ファイル・アクセス法と入出力制御

次についての理解を確認する。

- 順アクセス

- ランダムアクセス
- 基本アクセス方式，READマクロ，WRITEマクロ，CHECKマクロ，WAITマクロ
- 待機アクセス方式，バッファリング，GETマクロ，PUTマクロ
- ブロッキング，デブロッキング
- オープン処理，クローズ処理，OPENマクロ，CLOSEマクロ
- 装置からの独立性

5.2 ファイル処理の一般的な技法

次のファイル処理の一般的な技法について理解させる。なお，このあとすべて，本章では話を具体的にするため，特定機種の特定OSを想定して，その場合のマクロ命令やジョブ制御命令を示しながら説明するとよい。

(1) ファイル処理のためのマクロ命令

次のマクロ命令の書き方と機能について，編成方式，アクセス方式に共通の部分に関して，特定の機種・OSを想定して説明する。

- ファイル記述マクロ
- OPEN，CLOSEマクロ
- GET，PUTマクロ
- READ，WRITEマクロ
- WAIT，CHECKマクロ
- データ事象制御用ブロック
- バッファ制御用マクロ

(2) ファイル・スペースのアロケーション

ファイル・スペースのアロケーション（割当て）に関して次を説明する。

- ファイルのボリュームへの割当て
- 磁気テープへのアロケーションと磁気ディスクへのアロケーションの比較
- ブロックによる割当て
- トラックによる割当て
- シリンダによる割当て
- 絶対アドレスによる割当て
- スペースの分割割当て
- 追加スペースの割当て（一次割当てと二次割当て）
- アロケーションによる処理効率のちがい
- アロケーションの時期とスペース開放の時期
- ファイルの後始末（ディスポジション）

(3) 緩衝域の取得と制御について，次のことを説明する。

- 緩衝域プールの設定と開放

・緩衝域の制御

・単純緩衝法

・交換緩衝法

(4) ファイルのカタログ

次について説明する。

・ファイルのカタログの概念

・修飾名

・ディレクトリー，ボリューム目録

・カタログ名の検索のしかた

・カタログへの追加・変更・削除のしかた

・世代番号，世代別ファイル

(5) ファイルの保護

次について説明する。

・合言葉による保護

・エンキュー，デキューマクロによる保護

(6) ファイルの連結

次について説明する。

・ファイルの連結の概念

・ファイルの連結のしかた

(7) ライブラリの種類

ライブラリの種類として，次について説明する。

・システム・ライブラリ

・私用ライブラリ

・共用ライブラリ

・一時的ライブラリ

5.3 順編成ファイルの処理

(1) 順編成ファイル処理の基本型

順編成ファイル処理の基本的考え方として，次のことを理解させる。

・順編成ファイルが適しているケース

大量データ処理，単純記録，少量データの扱いなど

・順編成ファイルにおける分類と突合わせの重要性

・磁気テープファイルの特徴

磁気テープ・ラベル，逆読み，スタート時間，ストップ時間，巻もどし時間

(2) 順編成ファイルの作成と更新

次について説明する。

・順編成ファイルに対するファイル記述マクロ命令

・順編成ファイルに対する装置制御のためのマクロ命令

コントロール，バックスペース，プリンタあふれテスト，行送り，スタッカ選択など

・磁気ディスクを使った順編成ファイルの利点

レコードの位置ぎめ，更新などのマクロ，および手操作の簡略化など

・順編成ファイルの作成法と使い方

基本アクセス法，および待機アクセス法についてプログラムの実例を示して学ばせる

・順編成ファイルの必要スペースの計算法

5.4 索引順編成ファイルの処理

(1) 索引順編成ファイルに適しているケース

索引順編成ファイルの適している次のようなケースをプログラムの観点から説明する。

・メンテナンス頻度と活用度のきわめて低い場合の大量データ処理

・即時処理

(2) 索引順編成ファイルの構成と索引

次について説明する。

・基本域

・**あふれ域**（Overflow area）

・索引域

・**マスタ索引**（master index）；**シリンダ索引**（cylinder index），**トラック索引**(track index)

・レコードの追加

(3) 索引順編成ファイルの効率と再編成

次について説明する。

・取出しの所要時間と，索引，あふれ域との関係

・再編成の必要性とその方法

(4) 索引順編成ファイルの作成法

索引順編成ファイルの作成法について，ファイル記述マクロと，作成するためのマクロについて，実例を示しながら説明する。

(5) 索引順編成ファイルの使い方

次について実例を示しながら説明する。

・順取出し，更新，書出しのマクロとその使い方

・ランダムな取出し，更新，書出しのマクロとその使い方

・緩衝域および作業域の必要量

(6) 索引順編成ファイルの必要スペースの計算法

次について説明する。

・基本域の計算法

・索引域の計算法

・あふれ域の計算法

5.5 直接編成ファイルの処理

(1) 直接編成ファイルの適しているケース

直接編成ファイルの適している次のようなケースについて，プログラムの観点から説明する。

・索引順編成ファイルが適当でないようなランダム処理

・レコードの追加が多くない場合

(2) ランダマイズ技法

ランダマイズ技法（randomizing technique）について，次のものを説明する。

・除算法

・ディジット分析法

・重ね合わせ法

・基数変換法

(3) シノニム処理

次について説明する。

・シーケンシャル法

・チェーン法

・バケット法

(4) 直接編成ファイルの作成法

直接編成ファイルの作成法について，ファイル記述マクロと，ファイル作成のためのマクロに関し，実例を示しながら説明する。

(5) 直接編成ファイルの使い方

次について，実例を示しながら説明する。

・直接編成ファイルからの取出し

相対ブロックアドレス，相対トラックアドレス，相対トラックアドレスおよび実キー，実アドレス，探索など

・更新，追加

・書出し

・排他要求

(6) 直接編成ファイルの必要スペースの計算法

直接編成ファイルの必要スペースの計算法および，パッキング比について説明する。

5.6 区分編成ファイルの処理

(1) 区分編成ファイルの適しているケース

区分編成ファイルの適している次のようなケースについて説明する。

- プログラム・ライブラリ・ファイル
- カタログ手順ファイル
- テーブル・ファイル

(2) 区分編成ファイルの構成と登録簿

次について説明する。

- 登録等と本体との関係および，更新時のそれらの状況
- 登録簿の構造
- 本体の構造

(3) 区分編成ファイルの作成法

区分編成ファイルの作成について，ファイル記述マクロおよび作成のためのマクロに関し，実例を示しながら説明する。

(4) 区分編成ファイルの使い方

次について説明する。

- 登録記入項目リストの作成
- メンバーの位置づけ
- 登録簿項目の変更
- メンバーの取出し
- メンバーの更新
- メンバーの書きなおし
- 区分編成ファイルの再編成

(5) 区分編成ファイルの必要スペースの計算法

区分編成ファイルの必要スペースの計算法を説明する。

5.7 実　習

具体的な条件を問題として与え，それに対する索引順編成ファイルおよび直接編成ファイルを作成させ，入出力を行なわせる。

指導上の留意点

実習機械における具体的なマクロ命令についての説明を行なう。

第6章　オペレーティング・システムの利用

目　標

プログラム管理，タスク管理，主記憶管理，サービス・プログラム，回復管理，操作員との対話などのオペレーティング・システムの利用について学ばせ，それらのためのマクロ命令を使えるようにする。

内　容

6.1　プログラム管理

プログラム設計上必要な次のことについて，具体的なマクロの使い方やコーディング例を示しながら説明する。

(1) プログラム初期設定

次について説明する。

- 基底レジスタの初期設定
- 汎用レジスタの保管

  再入不能プログラム，再入可能プログラムのそれについて
- ジョブ制御ステートメントからのパラメータの受取り方

(2) ロードモジュールの構造

次について説明する。

- 単純構造
- 計画オーバレイ構造
- ダイナミック構造

(3) 制御権の受渡しと

次について説明する。

- もどりのない制御権の渡し方
- もどりのある制御権の渡し方
- 制御プログラムへのもどし
- ダイナミック構造におけるロード・モジュールの探索と，それを主記憶に入れる方法

(4) ジョブの作り方

- ジョブ，ジョブステップの作り方とジョブ制御言語の使い方

6.2　タスク管理

(1) OSタスク管理の復習

最初にまずOSのタスク管理の機能の復習と，全体におけるタスク管理の関係についての確認を行なう。

(2) タスクの生成

次について具体的なマクロの使い方をコーディング例を含めて説明する（以上同様である）。

- タスクの生成のしかた
- タスクの優先順位

  限界優先順位と指名優先順位
- タスクの消滅のしかた

(3) タスクの管理

次について説明する。

- タスクの上下関係

  システム・タスクを含めて考える。
- タスクおよびサブタスク間の連絡
- タスク間の同期
- 逐次再使用可能な資源の使用法
- インタロックとその防止法
- タスクの情報を得る方法

6.3 主記憶管理

(1) 主記憶の割当て

次について説明する。

- 固定タスク・マルチプログラミングにおける主記憶の割当て
- 可変タスク・マルチプログラミングにおける主記憶の割当て

(2) 主記憶の管理

次について，具体的なマクロ命令と，コーディング例を示しながら説明する。

- 明示的な主記憶域の要求と開放
- 静的な方法と動的な方法
- 主記憶域の共有使用
- 暗示的な主記憶域の要求と開放

6.4 サービス・プログラム

(1) 連係編集プログラム

次について，コーディング例などを示しながら具体的に説明する。

- 連係編集プログラムへの入力の指定のしかた
- 主要入力ファイル，追加ファイル，自動呼出しライブラリ
- ライブラリの組込み指定のしかた
- 出力ロード・モジュール
- ライブラリ，メンバー，入口点

・診断出力

・モジュールの編集

外部名の変更，モジュールの置きかえ，モジュールの削除

・オーバレイ・プログラムの設計と指定のしかた

・オーバレイ・プログラムの必要スペースの計算法と効率

・連係編集プログラムの実行のためのジョブ制御言語の使い方

・連係編集プログラム用制御ステートメントの使い方

体系的に説明する。

(2) ロ ー ダ

次について説明する。

・ローダの機能と，連係編集プログラムとのちがい

・ローダの使い方

(3) 分類組合せプログラム

次について，コーディング例を示しながら具体的に説明する。

・制御ステートメントの体系と，分類組合せプログラムの機能の範囲

・分類制御ステートメントの使い方

・組合せ制御ステートメントの使い方

・レコード制御ステートメントの使い方

・その他の制御ステートメントの使い方

・入力ファイルの指定法

・中間ファイルの指定法

・出力ファイルの指定法

・分類組合せプログラムの実行開始の指定法

・分類組合せプログラムの修正法

・分類組合せプログラムの効果的な使い方

(4) 入出力などのユーティリティ・プログラム

次について説明する。

・ファイルの複写プログラム

・ファイルの比較プログラム

・ファイルの変換プログラム

・順編成ファイルを区分編成ファイルに変換するなど

・磁気テープの初期化プログラム

・磁気ディスクの初期化と代替トラック割当てのプログラム

(5) システム生成プログラム

次について説明する。

・システム生成の概要

・システム生成の手順の概要

・システム生成で使用するファイル

・機器構成の指定数

・ＯＳ制御プログラムの指定法

・コンパイラ，ユーティリティの指定

・その他の指定

・生成システムのテスト

・システムのバージョン

6.5 回復管理

(1) システムの行なう回復管理

次について説明する。

・ハードウェアで行なわれる回復管理

・ＯＳのシステムプログラムで行なわれる回復管理とその使い方

(2) ユーザの行なう回復管理

次についてマクロを含め，具体的なコーディング例を示しながら説明する。

・チェックポイントのとり方

・リスタートのしかた

6.6 操作員との対話

次について，具体的な実例を示しながら説明する。

(1) 初期ローディングのしかた

次について説明する。

・システム・パック

・初期ローディングの手順

・初期ローディングで必要なコンソール入力のしかた

・初期ローディングの際にシステムから与えられるメッセージ

(2) 実行中のシステムからのメッセージに対する処置

・ジョブの実行中にシステムから与えられるメッセージ

システムからのメッセージに対して与えなければならないオペレータの応答

(3) プログラムによる操作員への出力とその応答

次について説明する。

・プログラムから操作員へのメッセージの出力のしかた

・プログラムから操作員へのメッセージの出るタイミングと，メッセージの出方

・応答が必要なメッセージに対する操作員の応答のしかた

・操作員へのメッセージの出力に関するプログラミング上の注意

(4) 会計ルーチン

次について説明する

- 会計ルーチンの機能
- 会計ルーチンとジョブ制御言語，操作員への出力との関係
- 会計ルーチン作成上の注意とシステムへの組込み方

6.7 実　習

学習したマクロを使ったコーディングをさせる。

指導上の留意点

実習機械における具体的なマクロ命令について説明する。

第7章　オンライン・システムのプログラミング

用　語　この章では次の用語を教える。

回線制御プログラム，メッセージ処理プログラム，動的バッファリング，セグメント，ポーリング，セレクション，マルチプル・ウェイト，回線ＩＯＣＳ，メッセージ・キュー，オンライン・テスト・シミュレータ

目　標

オンライン・システムのプログラミングについて学ばせ，指導があれば実際のプログラミングができるようにすることを目標とする。

内　容

7.1　回線制御プログラム

回線制御プログラム（line control program）およびそれに関連して，次のことを学ばせる。

(1) オンライン・プログラムの特徴

オンライン・プログラムの特徴として，次のことを説明する。

- 待機処理
- 即時処理
- 同時処理
- 無停止処理

(2) オンライン・プログラムの構成

オンライン・プログラムの構成として，次のことを説明する。

- ＯＳの回線制御プログラムの機能の概要
- 回線制御プログラムの機能の概要
- **メッセージ処理プログラム**の機能の概要

(3) メッセージ・キュー・ファイル

次について説明する。

- メッセージ・キュー・ファイルの機能
- **メッセージ・キュー**（message queue）へのメッセージのキューのしかた
- メッセージ・キューからのメッセージの取出し

(4) 動的バッファリング

次について説明する。

- **静的バッファリング**(static buffering)**と動的バッファリング**(dynamic buffering)
- **セグメント**(segment)とセグメントの割当て

(5) ポーリング・セレクション

次について説明する。

- 回線制御のしかた
- 競合方式，**ポーリング**(polling)，**セレクション**(sellection)など
- ポーリング・セレクション方式の利点と欠点
- ポーリング・リスト
- ポーリング・セレクションのためのマクロとチャネル指令

(6) マルチプル・ウェイト

次について説明する。

- **マルチプル・ウェイト**(multiple wait)の機能と利点
- マルチプル・ウェイトの使い方

(7) 回線IOCS

次について説明する。

- **回線IOCS**(line IOCS)の種類と機能
- 回線ファイル定義マクロ
- 回線入出力マクロ
- その他の回線IOCSマクロ

(8) 回線制御プログラムの機能

次について説明する。

- 開始・終了
- 回線入出力
- メッセージ分析とコード変換
- メッセージ・キュー処理
- センタ端末処理
- エラー処理
- メッセージ処理プログラムとのインタフェース

(a) 回線制御プログラムの構成法と所要領域，処理時間の推定法

7.2 メッセージ処理プログラム

メッセージ処理プログラム(message processing program)に関連して次につき説明する。

- メッセージのとり出しともどしのためのマクロ

・メッセージ処理プログラムの作成上の注意
・メッセージ処理プログラムのための主記憶領域
・メッセージ・スイッチング型処理
・問合わせ型処理
・データ収集型処理
・メッセージ処理プログラムの構成法と所要領域，処理時間の推定法

7.3 サポート・プログラム

次について説明する。

・リスタート・プログラム
・ファイル・コピー，ファイル・ダンプ・プログラム
・統計集計プログラム
・オンライン・テスト・シミュレータ（online test simulator）

7.4 オンライン・プログラム・テスト

次について説明する。

・オンライン・プログラムのテストのむずかしさ
・カードとプリンタによるオンライン・テスト用ファイル
・テスト用ファイル
・端末シミュレータによるテスト

7.5 オンライン・プログラムの管理

次について説明する。

・オンライン・プログラムの初期化
・ロードモジュールの構造
・バッチ処理とオンライン処理のマルチプログラミング

参考文献

1. 情報処理学会編，「情報処理ハンドブック」，昭和47年，9,000円
2. コンピュータ・メカのオペレーティング・システムに関する各種ユーザ・マニュアル
3. 藤井純，鈴木伸夫，「オペレーティング・システム」，昭和45年，産業図書
4. 岩城三郎，「オンライン・システム設計入門」，昭和48年，日刊工業新聞社 1,500円

汎用プログラム・パッケージ

目　　次

第1章　汎用プログラム・パッケージ …… 530

1.1　汎用プログラム・パッケージの効用 …… 530

1.2　汎用プログラム・パッケージの分類 …… 531

1.3　汎用プログラム・パッケージの選択 …… 531

第2章　実　習 …… 534

科目「汎用プログラム・パッケージ」

教育の目標

この科目は，コンピュータのメーカ，ユーザにおいて開発されたプログラムが，できるだけ多くの人に利用してもらえるように，プログラム・パッケージとして整備されていることを理解させるとともに，提供されたプログラム・パッケージを，実際にコンピュータを利用して業務を行なうときにうまく使用できるような技能を修得させる。

時間配分

章	履習時間（時間）		
	講　義	演　習	実　習
1. 汎用プログラム・パッケージ	6		
2. 実　習			4
合　　計	6		4

第1章　汎用プログラム・パッケージ

目　標

コンピュータを使用する人の希望に応じて提供される，いわゆる汎用プログラム・パッケージの効用と機能を解説することによって，これまでに開発された各種プログラム・パッケージの特徴を理解させ，使用するコンピュータに適合したプログラム・パッケージを効果的に選択できる知識を習得させる。

内　容

1.1　汎用プログラム・パッケージの効用

一般に，コンピュータを使用して仕事をする場合には，メーカから提供された汎用プログラム・パッケージをできるだけ活用するほうが有利であることを述べ，ソフトウェア開発のための二重投資を避けるという，プログラム・パッケージ提供の主旨を理解させる。ただし汎用プログラム・パッケージを利用するにあたっては，コンピュータを使用して行なう仕事の内容とかコンピュータの機種や機能に適合したプログラム・パッケージをうまく選択することが重要である。たとえば「線形計画のためのプログラム・パッケージ」という名称のパッケージは数多く存在するが汎用プログラム・パッケージの名称が同じであっても，必ずしもすべてのコンピュータに利用できるとは限らないことを指摘しておく。

選択したプログラム・パッケージを不適当なコンピュータで利用した場合には，不正確な処理結果をアウトプットしたり，あるいは処理不能のメッセージを繰返えすなどして労力や時間を無駄に費やす恐れのあることを認識させる。

このように，汎用プログラム・パッケージの効用を十分理解させるためには，その長所・短所を明確に説明しておくほうがよい。

まず，汎用プログラム・パッケージは，利用者がコントロール・パラメータとデータをインプットさえすれば，コンピュータが所要の処理結果をアウトプットするように作られたプログラムであることを説明する。

たとえば，汎用プログラム・パッケージを利用して数組の観測データから回帰直線を求めるときには，利用者が，データの件数やデータ入力フォーマットを指定したコントロール・パラメータと観測データをインプットしてやると，コンピュータが所要の回帰直線をアウトプットすることなどを例にあげるとよい。コンピュータを使用して仕事する人達にとって，適当な汎用プログラム・パッケージがあれば，あらたにプログラムを作成するという手間が省けるので，その効用はきわめて大きい。一般に，数値計算用とか統計解析用のプログラム・パッケージは広く利用されている。このような利点を持っている反面，汎用プログラム・パッケージは，できるだけ数多くのコントロール・パラメータやデータを扱うことができるように作られているので，与えられた問題を解くときに必要以上の処理を実行したり，あるいは記憶容量を占有してしまうといった難点があることを述べる。線形計画のための汎用プログラム・パッケージを利用する

場合を例にとると，たとえ，与えられた線形計画の問題が10個の制約条件式しかない問題であっても，汎用プログラム・パッケージが100個の制約条件式の問題まで取扱えるようになっているものであれば，使用するコンピュータには，与えられた問題を解くために必要とする以上の記憶容量が要求されることになる。

いろいろな希望を持った多くの人達が利用できるように汎用性を持たせれば持たせるほどそのプログラム・パッケージは，利用者個人個人にとって余分な処理能力を備えている，いわゆる効率の悪いプログラムとなる点を注意しておく必要がある。

このように汎用プログラム・パッケージを利用するにあたっては，まず第一に，プログラム・パッケージの処理能力や制約条件を十分に把握することが重要であることを述べ，そのプログラム・パッケージが使用できるコンピュータ・システムの概要を明確に理解させておくことが望ましい。

1.2 汎用プログラム・パッケージの分類

汎用プログラム・パッケージの定義にはいろいろあって，簡単な数値計算や統計解析のためのプログラムから，特殊問題向け言語のプログラムまでを含めた，きわめて広範囲のプログラムを汎用プログラム・パッケージと称する場合もあるなど，今のところ明確な定義はないが，一般には，コントロール・パラメータとデータをインプットすると，所要の処理結果がアウトプットされてくるプログラムであると考えられていることを説明する。

利用者が提供された汎用プログラム・パッケージのなかから，必要とするプログラム・パッケージを簡単に選択できるように，汎用プログラム・パッケージは，目的別とか適用業種別に整理分類してあることを述べる。

プログラム・パッケージを大きく分けると，ライブラリ・プログラムとアプリケーション・プログラムの二つがあり，数値計算のためのプログラムなどはライブラリ・プログラムに含まれ，線形計画のためのプログラムやPERT計算のためのプログラムなどは，アプリケーション・プログラムに含まれることを述べる。さらに，アプリケーション・プログラムについては，適用業種別の分類があり，たとえば，製造業向けには工数計画プログラムや建設見積プログラム，流通業では，在庫引当てプログラムや在庫配分プログラムなどといった具合で各所で頻繁に利用するプログラム・パッケージをまとめていることを説明する。

とくに，化学工業などでよく利用されるプログラム・パッケージとしては，線形計画のためのプログラムと化学プラント建設のためのシミュレータ，建設業などでよく利用されるプログラムとしては，PERT計算のためのプログラムと土木測量のためのプログラム（COGO；Coordinate Geometry）などを例にあげるとよい。

汎用プログラム・パッケージを分類する場合に，同じ目的をもったプログラムであっても，各メーカごとに独自の名称をつけていることが多いので，利用者が混乱しないように注意しておく必要がある。

1.3 汎用プログラム・パッケージの選択

汎用プログラム・パッケージを利用して，与えられた問題を処理するにあたっては，まずはじめに問題

表1－1　代表的データ・マネージメント・システムの機能と特徴

システム	AEGIS	ASI－ST	CDMS	GIM	INFORMS	DS／2	CISS	IR／II	COGENTIII
開発会社	Programatics Inc.	Application Softwore, Inc.	S・D・C	T・R・W	General Analytics, Inc.	S・D・C	NEC	HITACHI	C・S・C
機械	IBM360／25以上	IBM360／30以上 Spectra 70／35以上	IBM360／50以上	IBM 7094, 1410, 360／30以上, GE 635	IBM 360／30以上 Spectra 70	IBM360／25以上	NEAC 2200／400以上	HITAC 8400／8500	IBM 360
オペレーティング・システム	DOS, TOS, OS	DOS, OS	TS－DMS	DOS OS	DOS, OS	DOS, OS	OS, MODIV	DOS	OS, MVT
必要メモリ	DOS, TOS 32K, OS 65K	DOS 32K, OS 65K,	256K	DOS 32K, OS 50K,	65K	DOS 32K	128KC	131KB	256K
I／O装置	リモート・ターミナルを含む標準I／O装置	リモート・ターミナルを含む標準I／O装置	カード, テープ, ディスク, ドラム, リモート・ターミナル, プリンタ, グラフイック・ディスプレイ	リモート・ターミナルを含む標準I／O装置	カード, テープ, ディスク, プリンタ	ディスク(2311), テープ(9／7TRIO, プリンタ(1403), カードリーダ(2540), ターミナル(2701)	MT(4～8台) プリンタ コンソール カードリーダ, ディスク	カードリーダ, プリンタ ディスク コンソール・タイプライタ	ディスク ドラム データ・セル
レコード構造	Fixed, Variable, repeating (2レベル)	Fixed, Variable repeating (99レベル)	Fixed, Variable repeating (16レベル)	Fixed, Variable	Fixed, Variable	Fixed	Fixed	Fixed	Fixed (COBOL) Variable
ファイル構成	シークエンシャル インデックス・シークエンシャル	シークエンシャル インデックス・シークエンシャル	インバーテッド	シークエンシャル, インデックス・シークエンシャル, インバーテッド・インデックス, ランダム	インデックス・シークエンシャル インバーテッド	インデックス・シークエンシャル	ダイレクト・アクセス	インデックス・シークエンシャル ダイレクト・アクセス	シークエンシャル インデックス・シークエンシャル ランダム
システムの機能	照会, 更新, 保護	ファイルの作成, 照会, 更新, 保護	ファイルの作成, 照会, 更新, 保護	ファイルの作成, 照会, 更新, 保護	ファイルの作成 保護	更新, 検索, 作表	ファイルの作成, 保持, 検索, 保護	定義, ファイルの作成, 保守, 検索, 管理	定義, ファイルの作成, 保守, 検索, 作表
言語形態	任意形式(シンタックス・コントロールしたもの)	任意形式 表形式	任意形式(シンタックス・コントロールした英語)	任意形式 (英語をベース)	自由形式	任意形式	COBOLストリング形式		
システム言語	BAL(シンタックス・ディレクトしたもの)	SPL(メタ・アセンブリ)	JOVIAL, BAL	低レベル・マクロ・アセンブリBAL	BAL, FORTRAN	MOL	CISS 言語 COBOL	IR／II独自の言語	BAL, COBOL
論理命令	AND, OR	ある			ある			ある	ある
算術命令	合計, HITS, 平均比率	ある			ある			ある	ある
処理方法	バッチ, オンライン	オンライン	オンライン, TSS	バッチ, オンライン	バッチ, オンライン	バッチ, TSS	バッチ, オンライン	バッチ	バッチ, オンライン
その他	マスク・テキスト・サーチ, テーブル・ルックアップ, BAL, FORTRAN, COBOLの機能的サブルーチン能力をもつ ソート・ルーチン	BALのコード・リンケージ, ソート・ルーチン, テーブル・ルックアップ, 関連ファイルをシークエンシャルにもランダムにも処理可能	リスタート機能 ソート・ルーチン	テーブル・ルックアップ 関連ファイルをシークエンシャルにもランダムにも処理可能			データ・ベースやオンライン入出力のサポート		

の内容を十分に吟味し，たとえば，線形計画の問題では，制約条件式の次元数などに相当するコントロール・パラメータを決定する必要があり，ついで，汎用プログラム・パッケージのマニュアルに記述されている処理内容の項と最小機器構成の項をよく検討して，問題の処理に合致したプログラム・パッケージと，その処理を実行できるコンピュータ・システムとを合わせて選択する必要がある。

もし，コンピュータ・システムの機器構成があらかじめ，決まっているならば，そのコンピュータ・システムで実行可能なプログラム・パッケージを選択することになる。所要の汎用プログラム・パッケージを効果的に選択できるようにするため，表1－1のような一般的なプログラム・パッケージの機能と性能を記載した一覧表を例示して，目的に合致した汎用プログラム・パッケージの概要を把握できる能力を身につけさせるとよい。

指導上の留意点

汎用プログラム・パッケージは，同じ問題の処理に対しても各メーカごとに異なった名称をつけているので，研修生が混乱しないようにとくに注意する。もし，研修生が使用できコンピュータ・システムや業務の内容がすでに決っているならば，その範囲内で指導するとよい。

第 2 章　実　習

目 標

汎用プログラム・パッケージを利用して実習を行ない，その経験をとおして，汎用プログラム・パッケージの利用に関する基本的な知識と能力を身につけさせる。

内 容

問題の把握からプログラム・パッケージのインプット・データの作成にいたる過程を解説し，適当なコンピュータ・システムで実習を行なうとともに，処理結果の検討方法を説明する。ここでは，線形計画の問題をとり上げＨＩＴＡＣ8400，ＬＰ8000を使用して解を求めることとした。

(1)　問題の設定

個々の栄養素に関して，最低必要な摂取量があり，また，それぞれの食物のなかに含まれている栄養素の量がわかっているとき，栄養摂取量の制約条件を満足し，しかも費用を最低にするには食物をどのように混合したらよいかを計算せよ。ただし

制約条件

① 各食物1kgおよび1単位あたりに入っている栄養素とコストは表2－1のとおりである。

② 各栄養素の最低摂取量は表2－2のとおりである。

(2)　機器構成の説明

この問題を解くためＨＩＴＡＣ8400，ＬＰ8000を利用する。まず，ＬＰ8000を使うために必要な最小機器構成を説明する。

① 中央処理装置：　1台（65Kバイト）

② コンソール・タイプライタ：　1台

③ カード読取機：　1台

④ ラインプリンタ：　1台

⑤ 磁気テープ装置：　6台

(3)　インプット・データの作成

インプット・データの形式について説明しなければならない。ＬＰ8000では，データ・カードとアジェンダム・カード（図2－1参照）と呼ばれるコントロール・カードがあることを述べ，それぞれのカードを作成してみる。

(4)　コントロール・パラメータの作成

インプット・データを用いて線形計画の計算を実行するために，アジェンダム・カードを作成する。そして，アジェンダム・カードで指定した計算手順について簡単に説明する。

このプログラムを動かすために必要な入出力装置の割当てを表2－3のように行ない，すべてのカードを準備する。

表２－１　食物重さ１kgあたりのコストおよび栄養素の量

	食物 / 栄養素	1 肉類	2 魚類	3 豆腐	4 ひきわり大豆	5 無機物	6 食塩	7 ほうれん草	8 ビタミン剤	9 あさくさのり
0	コスト（円）	580	700	263	308	113	100	226	3572	600
1	重さ（kg）	1	1	1	1	1	1	1	1	1
2	ほうれん草（kg）							1	(単位)	
3	ビタミン剤（単位）								1	
4	あさくさのり（kg）									1
5	魚類（kg）		1							
6	蛋白質（単位）	0.55	0.63	0.45	0.5			0.17	0.25	0.25
7	ビタミンB_2（単位）	0.26	0.2	0.13	0.12			0.7	41.6	2
8	ナイアシン（単位）	0.23	0.25	0.09	0.045			0.14	29.4	0.4
9	パントテン酸（単位）	0.02	0.04	0.055	0.06			0.14	9	0.4
10	燐（単位）	0.4	0.3	0.065	0.06	0.26		0.02	0.1	0.05
11	カルシウム（単位）	0.8	0.5	0.025	0.02	0.3		0.15	0.05	0.05
12	食塩（単位）	0.1				0.9	10			
13	食塩（単位）	0.1				0.9	10			

表２－２　各栄養素の最低摂取量

混合した食物の重さ	＝ 100
ほうれん草	≧ 1
ビタミン剤	≧ 1.1
あさくさのり	≧ 5
魚類	≧ 5
蛋白質	≧ 43
ビタミンB_2	≧ 70
ナイアシン	≧ 45
パントテン酸	≧ 16
燐	≧ 14
カルシウム	≧ 35
食塩	≧ 19 ≦ 24

図２－１　アジェンダム・カードの例

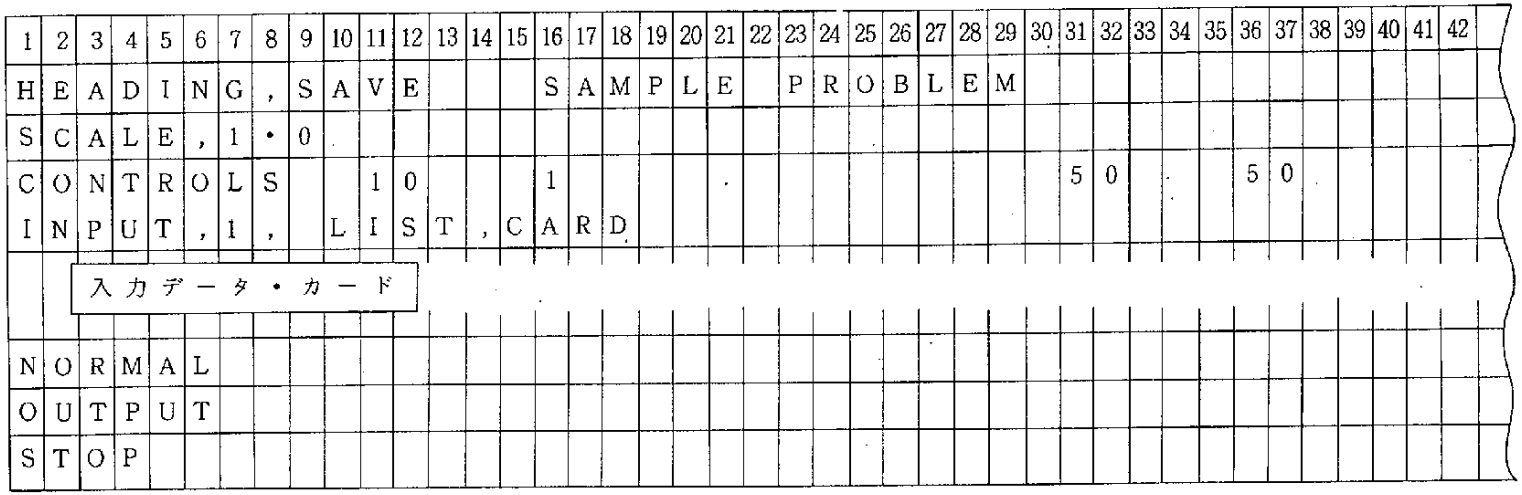


表２－３　入出力装置の割当て

装置名	装置タイプ	意味
READER	カード読取り機	アジェンダム・カードおよび入力データ・カードの読込み用
PRNTER	ラインプリンタ	オンラインおよびオフラインのアウトプット用
ATAPE	磁気テープ	バイナリィ・入力データ用
ETAPE	磁気テープ	Eta Records の入出用
MTAPE	磁気テープ	中間結果の出力用およびリスタート情報の入出力用
UTAPE	磁気テープ	スクラッチ用
VTAPE	磁気テープ	スクラッチ用

表2－4　LP 8000のロード方法

オペレータのアクション	結　　果
1) LP 8000のシステムをマウントする。	
2) TOSまたはTDOSまたはDOSエグゼクティブ・コントロール・システムをロードする。このときDATEおよび必要ならばTIMEの応答要求に答え，エグゼクティブ・コントロール・システムからSYSTEM READYのメッセージが出るまで行なう。 (注1)　既にエグゼクティブ・コントロール・システムがロードされている場合はこの必要はない。 (注2)　エグゼクティブ・コントロール・システムのイニシャル・ロードの方法はTOS／TDOSまたはDOSオペレーターズ・ガイドに詳しくでているので参照のこと。	エグゼクティ　・コントロール・システムがロードされ，日付けがセットされる。
3) LP 8000のランに必要なテープ（ATAPE, ETAPE, UTAPE, VTAPE, MTAPEさらに必要ならITAPE等）をマウントする。	
4) カード読取機に必要なラン・タイム・パラメータ・カードおよびアジェンダム・カードあるいは入力データ・カードをセットする。	
5) コンソール・タイプライタ上のCOINボタンをおす。	
6) E　LOD　LP 8000〔,mn〕〔,,pa〕〔,,mmmmmm〕をタイプインする。 (注1)〔　〕内は省略できることを示す。 (注2) mnは1）でマウントしたLP 8000プログラムの入っている装置のニモニック・コードを指定する。ただし，LP 8000プログラムがSYSRES中にある場合は省略してもよい。 (注3) paはラン・タイム・パラメータを読込む装置のニモニック・コードである。ラン・タイム・パラメータを与えないときはこの指定は行なわない。 (注4) mmmmmmはメモリの所要量を指定する。LP 8000ではこの指定を省略した場合には最大の値が与えられるようになっている。 (注5) この詳細はTOS／TDOSまたはDOSオペレーターズ・ガイドを参照のこと。	LP 8000プログラムがロードされると次のメッセージがタイプアウトされ最初のアジェンダム・カードが読まれる。

(5) オペレーションの方法

表2－4に示す方法でコンピュータにＬＰ8000をロードし，すでに作成したインプット・データおよびアジェンダム・カードを使って，ＬＰの解をもとめる。

(6) 結果の解析

コンピュータからアウトプットされてきたものの各項目の意味をアウトプットの例（図2－2参照）を用いて説明する。

```
LP/8000   01/21/74        SAMPLE  PROBLEM  OF   LP/8000                                                    PAGE    9

TOTAL    NO.    ETA    ROW        CURRENT        VECTOR  VECTOR   NEG.     RHS    C/V  NO.OF     CURRENT D/J
ITERS    ETAS   BLCKS  IDNT       VALUE          CHOSEN  REMVED   D/J+S    NO.    NO.  INFES     THETA/PHI

  16      27      0     1       43481.70123      *****   *****      0       1    ****     0    ****************


J(H)              BETA(H)              ROW(H)              PI(H)

00000            43481.70123           COST                1.00000
MEAT5               34.91152           RIN'                  .
FISH5                5.00000           NAIAS                 .
TOFU                42.07310           VITAB                 .
MUKIB               10.19935           WEIGT               3.42588
SALTA                 .63294           SALT1              10.34259-
HORE                 1.00000           HOREN              33.20366-
VITAK                1.18308           VITAM                 .
ASAK                 5.00000           ASAKU             301.87640-
- VITAM               .98308           PANT              380.68330-
- VITAB              5.46275           TAPAK             525.69299-
+ NAIAS              4.98887           FISH              178.48269-
- RIN                7.23950           KARSM             357.05857-
+ SALT2              5.00000           SAL-2                 .


#SOLUTION WAS CHECKED.

#MAXIMUM ERROR =          1.95179  *10**-10
```

図2－2　解析結果の例

指導上の留意点

実習では，できるだけ研修生の身近な問題をとりあげることが望ましい。ここでは，ＬＰの問題をとりあげて指導の手順を示してあるが，これにこだわる必要はなく，たとえば，研修生が建設業において日程計画などを担当している場合などには，ＰＥＲＴの問題をとりあげたほうが効果的である。

また，実習では，汎用プログラム・パッケージのマニュアルをどのように活用したらよいかを修得させることに重点をおき，受講後に，ほかの汎用プログラム・パッケージを気軽に利用できるように指導するとよい。

参考文献

(1) プログラム登録簿　　通産省

(2) プログラム登録集（昭和44年12月）
第1編アプリケーション・プログラム
第2編ライブラリ・プログラム
第3編システム・プログラム
(財) 日本情報処理開発センター

(3) プログラム・マニュアル　　各メーカ

索　　引

索　　引

【ア　行】

Illife ベクトル……463
アイドル・タイム……398
アウェルバッハ法……211
アカウンティング機能……410
アクセス・モード……418
アクティビィ・アナリス……137
アトム……457
アドレス空間……372
アドレス計算法……510
アドレス定数……480
アナログ信号……105
アナグロ・ディジタル変換(A/D変換)……106
アフタ・イメージ……432
後入れ先出し法……506
あふれ域……518
網構造……508
網制御装置……336
網設計……206
誤り再試行……391
誤り制御……326
アロー・ダイヤグラム……153
安全在庫……164
安定性……102
アンバランス型分類法……512
UNLOCK……297
意思決定システム……200
一意打切り可変長キーワード・トリー……295
1次おくれ……101
一部実施法……46
一般ファイル構造……295
一致チェック……294
インターバル・タイム……341
インディシャル応答……101
インバーテド・ファイル……312
インパルス応答（重み関数）……101
インライン／インハウス・システム……319
エクステント……417
X 一分布……33
(S,S)方式……164
SOP……178
SCERT……213
SDI……311
NC テープ……132
APT システム……133
FIFO……450
FLO……391
F 分布……33
MCBF……389
MTR……215
MTTR……390
MTBF……389
エラー・リカバリ（障害復旧）……290
エレメント誤り率……326
LIFO……450
LDU 分解法……13
エンティティ……80
オイラー法……6
オッシレーティング・ソート……512
オートメーション……98
オーバーヘッド……374
オペレーター順位……460
オンライン・アダプタ……342
オンライン・テスト・シミュレータ……528

【カ　行】

回帰分析……41
階級……11
χ 分布……33
回線 IOCS……527
回線制御プログラム……526
回線制御プログラム……429
階層構造型処理……187
カスケード・ソート……512
カタログ性能……379
緩急相補性……139
可観測性……113
可制御性……113
可能解……138

可変増分法…… 70
可変長完全キーワード・トリー……297
仮想記憶……372
拡張可能言語……466
稼動率……357
過渡応答……101
過渡状態…… 75
回路の簡略化……131
環構造……508
管理情報システム……254
管理図…… 36
ガーベジ・コレクション……300
ガウス・ジョルダンの消去法…… 12
ガウス・ザイデルの反復法…… 12
ア型分布…… 33
外部記号……483
キーワード・マクロ……485・487
キーワード……311
キューイング・ファイル……351
キー比較法……510
ギブソンミックス法……210
KEEP……297
KWIC……311
KWOC……311
木構造……295・507
機会損失費用……144
機能分析……202
機能分割システム……342
機密保護……290
基数変換法……294
基準入力…… 99
基底解……137
記号……476
記号式……476
記号パラメータ……486
記号表（シンボリック・テーブル）……457
企業内情報……250
企業外情報……250
局内端末……341
境界値…… 66
切換装置……342
擬似逆行列…… 13
擬似乱数…… 73
逆リスト構造……295
業務分割システム……342
クリティカル・パス……157
クラウト法…… 13
クレームの分析……175
クロック割込み……121
グレゴリー・ニュートンの補間式…… 26
区間コア・システム……120
鎖リスト……508
組合せ則……131
群計数検査……327
群管理システム……133
経営情報システム…… 253
経営のピラミッド…… 249
経営科学…… 252
計画情報システム…… 254
計算法（ランダマイジング）……294
計量モデル分析……168
形式言語……444
検策……310
検出……104
検出スイッチ……130
原始データ……104
減価償却……145
現価比較表……145
限時要素……131
誤字率……326
公衆通信回線……335
後進差分…… 22
構造的外乱……110
コア・システム……119
コア・マッピング……120
項……476
交換則……131
交換法……510
交路法…… 45
合理性チェック……294
コスト分析……175
固定コア・システム……120
固定増分率…… 70
固定長打切りキーワード・トリー……295
コード生成……457
コモン・セクション……482
固有値…… 6
固有ベクトル…… 14
混合形…… 65
コンテンション方式……427
コントロール・セクション……328
コンパイラ・コンパイラ……466
コンピュータ・シミュレーション…… 58

【サ　行】

再帰関係式……150

サイクリック・レダンダンシー・チェック……327
サイクリック・ダンプ……432
最小2乗解……13
最小2乗法……41
最早イベント時刻……155
最遅イベント時刻……155
最適化制御……111
最適計画……52
最適政策……148
先入れ先出し法……506
最適性の原理……150
索引……311
索引順次ファイル編成……419
索引ファイル編成……419
作業情報システム……255
作業の時間的余裕……157
産業連関分析……169
雑音……105
サブ・スキーマ……309
差分……26
サーボ機構……99
サム・チェック……294
3点見積り……158
サンプリング……71
CPM……159
時間稼動率……215
時間分析……195
シーケンス制御……99
シグナルフロー図……100
時系列分析……167
資源（リソース）……408
自己規定項……476
事実検索……311
4捨5入誤差……3
辞書索引……294
事象……79
指数分布……33
指数平滑法……168
システム・オーバーロード……120
システム可変記号……486
システム・クローズ・プログラム……349
システム・コンソール……341
システム診断プログラム……349
システム・スタート・プログラム……349
システム分析……178
システム編集……300
システム・モデル……108
システム・リスタート・プログラム……349
システム割込み……120
実効記憶容量……382
実効データ転送速度……383
自動索引……311
自動診断……391
シノニム……353
時分割多重伝送……105
GPSS……213
事務分析……182
シャドウ・プライス……139
状態変数……148
重回帰分析……51
乗算合同法……73
集線方式……334
集中型処理……187
状態空間……102
状態変数……102
周波数伝達関数……101
重複因子……480
主成分分析……55
主題分析……127
順次区分ファイル編成……419
順次ファイル編成……419
上位方向互換性……393
除算法……294
ジョブ……404
ジョブ・ステップ……404
初期条件……75
初期値……6
処理待ち行列……348
処理モード……418
シラブル……457
自立型言語……302
シリンダ索引……518
シンタックス……444
シンタックス・ディレクテド・コンパイラ……461
シンタックス・トリー……444
シンタックスのあいまいさ……446
シンプレクス法……78, 137
シンプソンの法則……27
信号速度……332
人工変数……138
推移行列……102
数字分類法……510
枢軸操作……12
推定モデル……109
スキーマ……302
スキャンナ……457

スタックド・ジョブ・スケジューリング……407
スタフ……250
スタック……506
数値解析……2
数値制御……132
数値制御機械……132
数値法……3
ステイト・チェンジ・アプローチ……68
ストップ・ワード……311
ストレージ……81
ストレージ空間……372
スナップ・ショット・ダンプ……402
スワッピングスワ……415
正規方程式……167
制御可能変数……68
制御不能変数……68
制御偏差……99
正則行列……10
静的最適化……112
静的バッファリング……527
セグメント……527
セグメンテーション……375
セグメント方式記憶保護……368
セット……81，303
正当性チェック……293
絶対記号……476
セット可変記号……486
セマンティックス……444
セレクション……527
遷移図……458
線形微分方程式……5
線型リスト……506
選択法……510
前進差分……26
全二重通信方式……324
相関係数……55
送信待ち行列……348
相対記号……476
装置独立性……289
双対定理……139
そう入法……510
属性……79

【タ　行】

代替案……178
帯行列……13
台形法則……27
ダイナミック・プログラミング……147
ダイナミック・リロケーション……370
多項式……17
多重プログラミング制御……344
多重リスト構造……295
タスク……413
タスク管理プログラム……413
多段決定過程……148
ダニレフスキー法……14
多変量解析……51
多変数制御……102
ダミー・セクション……482
単一システム……342
単一方程式モデル……342
探索法……78
ダンプ……402
チェック・ディジット……294
チェック・ポイント・ファイル……352
チェック・ポイント／リスタート……297
チェック・ポイント・リスタート……429
チェンジ・ジャーナル……293
弛緩法……12
超越関数……17
頂点保持制御……112
直交配列表……46
直結回線方式……333
直接アドレス……294
直接最適化……112
直接制御……111
直接ディジタル制御（DDC）……100
直接ファイル編成……419
直列伝送方式……325
塵集め……508
チョムスキーの分類……446
中心差分……26
追従制御……99
追値制御……99
通信制御……345
通番……355
ディジタル信号……105
ディジタル入力装置……107
TWS……466
デバイス・インディペンデンス……418
大分布……33
ディレクトリ・デコーディング……295
定型的情報……252
定期発注法……164
定数表……457
定常応答……101

定常業務処理……200
定常状態……75
定値制御……99
適応制御……111
データ構造……508
データ交換システム……320
データ収集システム……320
データ分配システム……320
データ量分割システム……342
データの互換性……392
データの非冗長性……284
データ・ベース……284
データ・ベース管理者……299
データ・ベースの更新……288
データ・ベースの生成……287
デッド・ロック……297
テーブル・ルック・アップ……450
デュアル・システム……215
デュプレックス・システム……215
伝送……104
伝送制御符号……327
伝送制御方式……327
伝送速度……332
伝達関数……100
電流信号……105
問合せシステム……320
同期……70
動的最適化……112
動的バッファリング……527
特権命令……367
トークン……457
導出……445
特性方程式……102
特定通信回線……333
凸集合……136
トップ・ダウン法……460
ドップ・ベクトル……463
ド・モルガンの定理……131
トラック索引……518
トランザクション……79
トリー（木）……449
トレイラ・ステートメント……486
トレーサ……402

【ナ　行】

2次おくれ……101
2次元アドレス……375
二乗採中法……294
二分探索法……513
入出力制御……345
入力データ作成システム……188
ニュートン法……20
ニュートン・ラフソン法……3
ネットワーク問題……141

【ハ　行】

排他的制御……356
排他制御……297
配置法……510
パフォーマンス測定基準……68
パーマネント・コア・エリア……119
バランスト・トリー……295
パラメータ外乱……111
バランス型分類法……512
バランスト・トリー……449
半二重通信方式……324
判断業務処理……200
反復法……2
PID調節計……100
BNF……446
非線形微分方程式……5
非定型的情報……252
ヒッチコック・ベアストウ法……23
表（テーブル）……450・505
費用対効果分析……178
ビフォア・イメージ……432
比率制御……99
比例ゲイン……101
バイナリ・トリー……295
バイナリ・トリー……295・449
バイナリ・サーチ……450
ハイブリット・シミュレーション……65
バインディング……289
パケット方式……353
パーサ……460
はさみうち法……18
パスワード……289，419，355
バックアップ・システム……357
ハッシング……450
ハッチカード……355
発注サイクル……162
発注点法……164
バッファ制御……347
PERT……152
配分則……131
パーティライン方式……106

ファイル配置……300
ファイル・バック・アップ……300
ファイル編成……419
ファンクショナル・アプローチ……68
フィジカル・レコード……417
フォールバック・システム……357
負荷配分システム……215
不動点……23
フィードバック……99
フィードバック制御……99
複合計画……339
プッシュダウン……450，506
浮動コア・システム……120
プライオリティ・ジョブ・スケジューリング…407
不連続制御システム……110
FREE……297
プリコンパイラ……453
フロー・アプローチ……68
プログラム制御……99
プログラムの互換性……393
プロセス外乱……110
プロセス制御……100
プロセス割込み……120
ブロック……79
ブロック誤り率……326
ブロック線図……100
ブロック・ダイアグラム……79
ブロック分類法……510
プロトタイプ・ステートメント……486
分割法……45
分岐回線方式……334
文献検索……394
文脈自由文法……446
並列伝送方式……325
ベキ乗法……14
ページ方式記憶保護……368
ページング……372
ヘッダ・ステートメント……486
変換……104
変調速度……332
変調方式……333
変復調装置……333
補間多項式……26
保管費……56
ポジショナル・マクロ……485
ホスト言語……301
ポップ・アップ……450，506
ボード線図……101
ボトム・アップ法……460
ホーナー法……21
ポリッシュ記法……463
ポーランド記法……506
ポリフェーズ・ソート……512
ポーリング……527
ポーリング方式……427
ポーリング／セレクティング方式……329

【マ　行】

埋没費用……144
マクロ・プロセサ……452
マージ法……510
マスタ索引……518
待ち行列……506
待ち行列制御……354
マルチプル・ウェイト……527
マルチプレクサ……106
マンション・コミュニケーション……349
ミルン法……9
無干渉制御……102
虫くい行列……11
メタ記号……445
メタ言語……444
メッセージ・キュー……526
メッセージ処理プログラム……526
メッセージ制御プログラム……429
メモリ割付……457
メモリ・シェア……377
面接調査……172
目的関数……68
目標値……99
モデル・ステートメント……486
モード制御……102

【ヤ　行】

ヤコビ行列……13
ヤコビ法……14
有限オートマトン……458
有限桁数計算……3
優先順位（prioity）……118
UDC……311
輸送問題……139
要因配置……42
予測子・修正子……9
予測モデル……109
呼出応答方式……327

【ラ　行】

ライン……250
ライン・アダプタ……340
ラプラス変換……100
乱塊法……328
ランダマイジング……353
ランダマイズ技法……519
乱ファイル編成……419
リカバリ・プログラム……349
離散形……65
離散分布……317
リスト構造……294，514
リスタート……434
リテラル……476
リテラル・プール……480
リード・タイム……164
リニア・サーチ……450
リファレンス・モード……420
領域（パーティション）……409
リンカ……465
リング構造……295
リング方式記憶保護……373
ルンゲ・クッタ法……8
レイト……84
レギュレータ……99
レコグナイザ……457
レコード形式……418
レジスタ方式記憶保護……368
レベル……84
連続形……65
連続制御システム……100
連続分布……317
ロギング・ファイル……352
ロケーション・カウンタ……475
ロジカル・レコード……417
LOCK……297
ロードシェア……377
ロール・イン／ロール・アウト……415
論理回路素子……130
論理式……312
論理代数……131
論理モデル……109

【ワ　行】

ワーキング・エリア……119
割り当て問題……140
割込制御処理……366

昭和49年3月発行

発行所　財団法人　日本情報処理開発センター
東京都港区芝公園3丁目5番8号
機械振興会館内
TEL　(434) 8211 (代表)

印刷所　株式会社　ジュコー
東京都品川区北品川4－8－15・Mビル
TEL　(449) 2264

48-E 002