

情報処理技術者教育に関する調査報告書

— カリキュラム '80—1 —

(第 1 種 編)

昭和 56 年 3 月

財団法人 日本情報処理開発協会
情報処理研修センター



ま え が き

当研修センターでは、「特定機械情報産業振興臨時措置法」の成立と、同法に基づく「ソフトウェア業の高度化計画」の告示により、情報処理技術者、中でも情報処理技術者試験の特種、第一種合格者レベルの技術者の養成が急がれている現状から、昨年度より「情報処理技術者教育に関する調査」を実施し、情報処理技術者試験を対象とする技術者養成のためのカリキュラムを作成する事業を進めて参りました。

本カリキュラムは過去に実施された情報処理技術者試験問題の分析を基礎とし、編集も同試験の種別に合わせて特種編、第一種編、第二種編の3分冊となっておりますが、情報処理技術者を育成していく上での標準的カリキュラムともなり得るよう配慮しております。

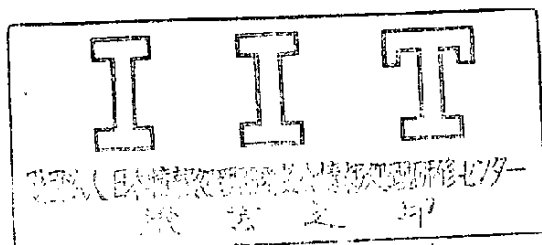
本事業の実施にご尽力いただいた関係各位に厚くお礼を申し上げますとともに、本カリキュラムが各方面で利用され、情報処理技術者育成の一助となるよう念願しております。

昭和 56 年 3 月

(財) 日本情報処理開発協会

情報処理研修センター

所長 手 島 篤 二

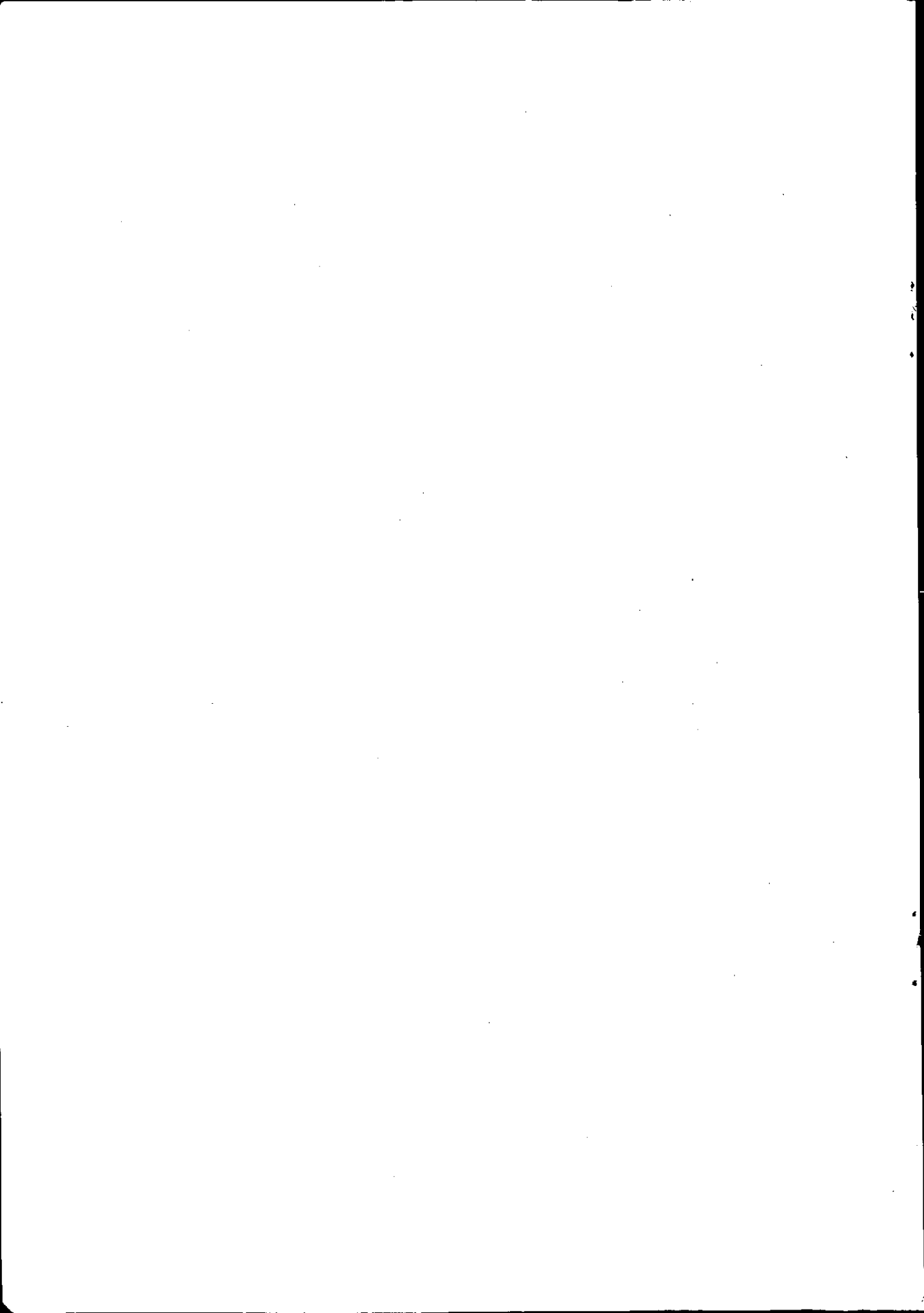


寄贈

カリキュラム小委員会

(敬称略, 五十音順)

委員長	魚 木 五 夫	産 業 能 率 大 学
委員	岡 部 正 文	日 本 鋼 管 (株)
"	大 日 方 真	日 本 コ ン プ ュ ー タ (株)
"	菊 池 光 昭	(株) リ ー ド
"	南 條 優	日 産 プ リ ン ス 自 動 車 販 売 (株)
"	西 村 真 一 郎	(財) 日 本 情 報 処 理 開 発 協 会 情 報 処 理 研 修 セ ン タ ー
"	畠 山 信 順	中 国 工 業 (株)
"	服 部 幸 英	東 洋 情 報 シ ス テ ム (株)
"	廣 松 恒 彦	(財) 日 本 情 報 処 理 開 発 協 会 情 報 処 理 研 修 セ ン タ ー
"	水 原 邦 保	(株) 日 立 製 作 所
"	元 植 郁 夫	広 島 修 道 大 学



目 次

ま え が き

総 論

第 1 章	システム概説 (S 1)	1
第 2 章	情報システムの分析 (S 7)	13
第 3 章	社会とコンピュータ技術 (S 9)	25
第 4 章	情報処理技術概説 (C 1)	37
第 5 章	コンピュータシステム (C 3)	57
第 6 章	ファイルシステム (C 5)	75
第 7 章	プログラミング (C 2)	91
第 8 章	ソフトウェアの設計と開発 (C 14)	105
第 9 章	データベース (C 11)	123
第 10 章	オペレーティングシステムとアーキテクチャ (C 6)	145
第 11 章	データコミュニケーション (C 10)	165
参 考		185

I
1
55-E002-1-C

総 論

1. ま え が き

1.1 技術者の不足と育成

近年、情報処理技術は急速に発達し、わが国でも計算機がめざましく普及すると共に、その適用範囲もますます広がりがつつある。コンピュータを中心とした情報処理システムは、ますます複雑かつ大規模化する傾向を示し、関連する諸技術は高度化の傾向を保ち続けている。これに伴って、情報処理技術者の需要もまた、量的にも質的にも増大を続けており、こうした傾向は今後ともかなり長期にわたって続くものと考えられる。

この情報処理技術者に対する長期的な需要予測を示したものとしては、昭和51年に発表された、産業構造審議会情報産業部会の報告〔通商産業省1976〕がある。この中で、昭和49年度末から昭和60年度末への需要の伸びは、表1-1のように示されており、この約10年間で技術者需要は約4倍と予測され、特に上級技術者としてのシステム・エンジニアの需要増加は顕著で、6倍と予測されている。

表1-1 情報処理技術者の需要予測

	49年度末	60年度末	倍 率
情報処理技術者需要	154 ^{千人}	578 ^{千人}	3.8
内			
システム・エンジニア	20	120	6.0
プログラマ	43	275	3.6
小 計 (構成比)	63 (40.9%)	275 (47.6%)	4.4

一方、昭和44年度以降実施されてきた、「情報処理技術者試験」(ただし、初年度のみ「情報処理技術者認定試験」)は、すでに10年以上を経過している。その区分別累積合格者数は、表1-2に示す通りである。

表1-2 情報処理技術者合格者数

試 験 区 分	合 格 者 数
特 種 (46~55年)	2,841 人
第1種 (44~55年)	9,909
第2種 (44~55年)	35,240
合 計	47,990

(通商産業省調べ)

従って、表1-1および表1-2の数字を比較すると、技術者の需要に対し、技術者試験の合格者数が、著しく少ないことが分かる。その理由としては、この試験がいわゆる資格試験ではないために、まだ多くの未受験者が、概当する技術者の中に居るということも挙げられるだろう。

しかし、ここ数年間における、情報処理技術者試験での合格率を見ると、特種および第1種については約1割、第2種については約2割となっていて、この数字は他の多くの国家試験における合格率の数字と比べても、決して良いとは言えない。

こうしたデータは、わが国において、今後共に情報処理技術者が不足し続けるであろうことや、情報処理技術者の質の向上が強く望まれるということを示していると言えよう。従って、仮に技術者試験が技術者の質の評価につながる唯一の尺度ではないにしても、少なくともこうした試験に合格できる実力を持つ、情報処理技術者の育成は国家的急務であると考えられる。

1.2 育成指針の流れ

情報処理技術者の育成を進めるためには、具体的に種々の方策が考えられるが、何よりもまず重要なのは、育成カリキュラムであろう。カリキュラムは、育成システムの中の計画活動の成果であり、効果的な育成が行なわれるための、重要な必要条件である。

また、カリキュラムの開発は、個々の組織体や事業所等で行なうには、負担が大き過ぎるきらいがある。できれば、ある程度標準的な状況設定の中で、汎用性の高いカリキュラムを開発し、個々の組織体などに対しては、それを適宜修正したり、部分的変更を加えるなどによって、それぞれに適したものを得るのが効率的である。

こうした考えの下に、情報処理技術者育成のための標準カリキュラムの開発は、欧米諸国でも1960年代に始まっている。わが国でも、この種の標準カリキュラムとして、最初の本格的なものが昭和43年(1968年)度以降に開発された。これが、通商産業省の編さんによる「上級情報処理技術研修ガイドブック」〔開発協会1970〕である。

これは、情報処理技術者のうち、いわゆる「上級」者の育成を対象としたものであったが、その後、初心者インプットと想定したカリキュラムとして、昭和45年度に「初級情報処理技術者育成指針」〔開発協会1971〕が開発された。さらに、昭和48年度には、上記初級カリキュラムによる育成のアウトプットを、インプット側に想定した、上位のカリキュラムとして、「中級情報処理技術者育成指針」〔開発協会1974〕が開発されている。

こうして、形式的には「初級—中級—上級」という、一連の情報処理技術者育成のための標準カリキュラムが完成したわけであるが、この開発計画は全体を統合して行なわれたものではなく、最初に開発された「上級」と、その後開発された「初級—中級」との間には、余り明確な関連は与えられていなかった。それと共に、こうした標準カリキュラム開発期間中における、情報処理技術の進歩発展は目ざましく、コンピュータを中心とした情報処理システムの一般社会に対する普及も、年々その度合いを高めてきた。

以上の様な背景のもとで、むしろ「初級→中級」の流れに適合し、なおかつ新しい時代の社会的要請や技術的動向に合致するものとして、昭和54年度には新しい「上級情報処理技術者育成指針」〔開発協会1979〕が完成した。このようにして、この時点までに、かなり統合的な、初級から中級を経て上級に至る、情報処理技術者育成のための標準カリキュラムが出そろったことになる。

1.3 新カリキュラムの必要性

前節で述べたように、昭和54年までに開発された「育成指針」は、初級—中級—上級と形式的にはそろったものの、現在までに既にいろいろな問題点を示してきている。

まず、標準カリキュラムとしての問題点を挙げると、第1に全体の開発が統合な計画の下に行なわれたものでない点がある。例えば、下位のカリキュラムのアウトプットが、すぐ上位にあるカリキュラムのインプットに想定されているが、全体的な内容調整や分量的なバランスなどについては、問題がないとは言えない。第2に、開発が時間的には直列的に行なわれたため、個々のカリキュラム開発時点での、技術的・社会的環境の違いが現われてしまっていることである。また、第3には、内容の記述がカリキュラムとしては詳細に過ぎ、これに従った教育の実施面では、柔軟性に欠けることである。これはむしろ、最初に開発された、「上級情報処理技術研修ガイドブック」で作られたスタイルが、直接的にも間接的にも影響していると言える。

さらに、外部的な条件としては、「情報処理技術者試験」との関係が挙げられる。元来、この標準カリキュラムと、技術者試験とは、独立に進められてきたプロジェクトである。しかし、技術者育成のシステムから考えると、カリキュラムは計画サブシステムの要素であり、試験は評価サブシステムの要素であって、各サブシステムの中で重要な役割を持つものである。従って、もし技術者育成をシステムとして考えようとするなら、両者を分離するわけにはゆかない。

しかも、技術者試験はまた、他の多くの国家試験とは異なり、評価されようとする知識や技術に関し、それを体得させるための教育カリキュラムを明示しないまま、10数年間続けられて来た。このことは、想定される受験者にとって、合格のための具体的な勉強方法などが、全く不明のままであるという事態をもたらしている。技術者試験における合格率が、他の国家試験などに比べて低いのも、一つにはこのためであるとさえ言われている。

このような事情と、「育成指針」開発後の期間における、情報処理技術の進歩や、社会環境の変化などを勘案すると、構想を新たにしたカリキュラムの開発が、必要となるのである。そして、この新カリキュラムでは、少なくともここで指摘したような問題点が、一つでも多く解決されるべきであろう。

1.4 新カリキュラムの特徴

本書で報告する、情報処理技術者育成のための新カリキュラム「カリキュラム'80」は、前節に述べたような必要性に基づき、指摘された問題点に対する一つの解決案として開発されたものである。従って、このカリキュラムでは、次のような点を特徴としている。

① 統合的な計画：

各水準に対する個別の育成カリキュラムを、単独に計画するのではなく、統合的な構想の下で立案され、計画されたものであること。

② 並行的な開発：

個別のカリキュラムを、逐次的に開発することは、時間的なズレを生じ、それが内容上のズレに発展する危険性もある。このカリキュラムでは、計画立案に参加したメンバーにより、十分な意志統一や調整を図った上で、各水準に対するカリキュラムを、ほぼ並行的に開発した。

③ 柔軟性の高い内容：

従来の「育成指針」に見られる、展開し過ぎた記述や、断定的な表現などを避け、実施上の問題を考慮した、幅のある柔軟性に富んだ内容表現にしたこと。

④ 最新の状況への配慮：

「上級育成指針」〔開発協会 1979〕以後における、ACMなどでの標準カリキュラムについての研究・開発状況も参考にしながら、同時に最新の情報処理技術に関する動向や、国内外の社会的環境などの変化についても考慮した。

⑤ 技術者試験との整合：

過去数年間における、情報処理技術者試験の内容を詳細に分析し、試験で想定していると思われる特種・第1種・第2種の各情報処理技術者の水準と、本カリキュラムで想定する技術者の水準とを、できる限り明確に関連づけた。それと同時に、今後予想される試験内容の改善などに対しても、ある程度示唆を与え得るような、進歩的内容にするよう心掛けた。

2. 構想のフレーム・ワーク

2.1 システム、マネジメント・システム、情報システム

情報処理技術者の育成に取組む上で、「システム」としての捕らえ方、すなわち「システムズ・アプローチ」を仮定する。組織体を捕捉する際に、行政システムとか企業システムとか社会システムなどというふうに見るのは、日常的である。システムの見地から、それを取巻く環境を見て状況を判断し、企画する姿勢である。

システムズ・アプローチを前提にすれば、発想・取組方・成果などは、必ずしも一様ではなく、多種多様な形態もありうる。多様さに関しては、分類とか比較は相当な程度までできるが、その統一は容易ではない。本カリキュラムでは、多種多様な捕らえ方のうち、「マネジメント・システム」としての捕らえ方を重視する。本カリキュラムの主旨は、情報処理技術者育成に関する、構想および実施についての、諸方策の提唱にあるからである。

マネジメント・システムの構造・構成要素などについて、次のような要約ができる。

マネジメント・システムの構成要素

- ① 「問題」であるような状況を認識する。
(Identify Problems.)
 - ② 「意思決定」する。進むべき方向や実施の方策について決断する。
(Make Decisions.)
 - ③ 上記の決定や決断に関する、実施や執行のための「管理システムや制御系」を保有する。
(Control the Decisions Made.)
 - ④ 上記のすべて(①, ②, ③)を実行するのに必要な「情報システム」を持っている。
(Provide the Information.)
(〔Ackoff 1974〕による)
- ①, ②, ③, ④は、マネジメント・システムの主要な構成要素であり、従って相互作用の(interactive)である。

本カリキュラムで、システム、マネジメント・システム、情報システム、…と述べるのは、主にこのような文脈の中においてである。また、情報処理技術などについて考察するときも、前述のような意味合いでの、「情報システム」との関連での取扱いを重視してゆきたい。

2.2 情報システムの構成要素とそのライフ・サイクル

本カリキュラムでは、情報処理技術や、情報処理技術者の役割を、「組織システム マネジメント・システム 情報システム」のフレーム・ワークの中で、捕らえようとする。この

際、情報システムは例えば次のような要素から構成されていると考えられる。

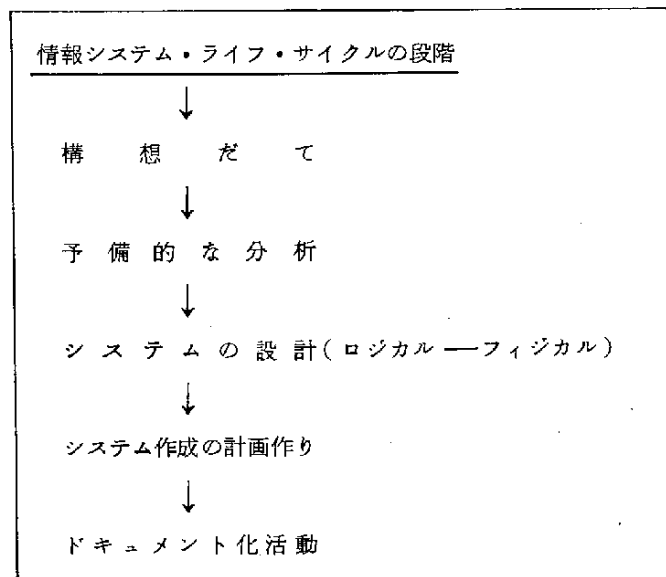
- ・ 情報処理業務担当者
- ・ コンピュータ・システムで代表される各種の情報処理ファシリティ
- ・ その他の諸資源

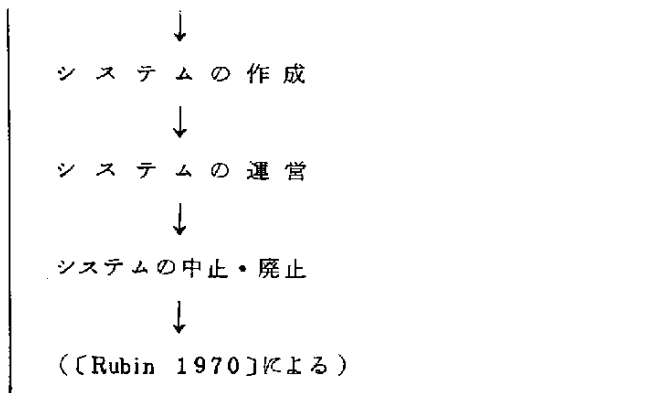
情報処理業務担当者は、コンピュータ・システムなどのファシリティを利用しつつ、マネジメント・システムの各種構成要素に情報を供給する。コンピュータ・システムとの間で、直接的なインターフェイスを持つか持たないかを問わなければ、現代の事務労働者の大多数は、情報処理業務の担当者としての側面を持ち合わせる。また、コンピュータ・システムのインターフェイスを含めて、いわゆる「事務」と情報処理業務との間の差異は、次第に希薄化する。しかし、これは直ちに同質化、均質化にはつながらず、（別の判断での）別種の分化が既に進行中である。「情報処理技術者」の役割や位置づけ等に関する、このような変化のプロセス内における確認・チェックが必要であろう。

〈情報システムのライフ・サイクル〉

情報システムの構成要素にしても、それら相互の関係についても、時の流れなどに応じ、それなりの変化がある。一つの情報システムは、それぞれ様々な局面を迎え、それらの局面において各種の資源や技術が要請される。このような変容・変化を組織的に解釈するのに、「システム・ライフ・サイクル」などの表現法も便利である。このようなことを手がかりにして、段階・局面・状況などの特色を捕らえることができる。

次にその一例を示す。





システム・ライフ・サイクルを更に要約し、次のような4段階に分けることもできる。

- ・システム分析 〔analysis〕
- ・システム設計 〔design〕
- ・システムの作成（実施） 〔implementation〕
- ・システムの運営（運用） 〔operation〕

これらのうち、〈分析—設計—作成〉の各段階をあわせて、「システムの開発」と呼ぶ慣習もある。「情報システムの開発」の質は、作られたシステムの効率や有効性に重大な影響を与えうる。こうしたことのため、それなりの固有の技術が必要である。また、今後開発されるシステムの多くは、実際には「再開発および再設計」の形をとるであろう。再開発や再設計は、今後において注目すべき特色の一つである。

2.3 情報処理技術者の「育成」

「育成」ということは、組織体の存続・維持・発展などと近い関係にある。そのため、その構想・形態は複数の側面を持ち合わせる。例えば、社会システムとしての「育成」、特定の組織体内での養成、組織人および新規参加者にとっての「体系」などである。これらの事情は、情報処理技術に限ったことではない。

にもかかわらず、多くの関係者にとって、「情報処理技術者」というとらえ方は、いささか唐突な感じを与えさせる。少なくとも、社会的側面についてはそうである。この一因は、急速な技術の進歩と短期間における技術者の増加にある。情報処理技術は、ここ20～30年の間に、段階的というよりは、むしろ飛躍的に進展してきた。そのために、情報処理技術に関連する「人的」諸資源に関する諸対応策は遅れがちであり、育成システムについてもまたそうである。

また、別の言い方では、「情報処理技術」の「異質性」もある。これは、技術体系自体の独特な個性と、時代や環境等との関連によるものである。情報処理技術者と、それ以外の「技術者」とを対比させたとき、汎用性と固有性、一般性と専門性、抽象性と具象性、開放(open)性と閉鎖(closed)性などの特徴を観察できる。これらをまとめると、情報処理「技術者」と、その他の多くの「技術者」に関する(社会的な)イメージは違っている。

〈育成の公共性と社会性〉

「育成」は、まず公共に関する、社会における、行政上の、構想・政策・施策の「系」である。社会・産業・市民などについての、振興・保護・助成等の一環である。情報処理技術者の育成が、有用かつ有効であるためには、育成以外の諸施策との間においても、また育成体系の内部構造についても、システム的な意味でのバランスや調和を、はかり続けて行くのが肝要である。

このような過程(プロセス)を持ち続けることによって、社会的公具としてのあり方により、育成システムはその効率性・有用性・有効性を備え、保持できる。「情報処理技術者」というとらえ方は、育成等の諸制度・諸活動と、いわば「抱き合せ」の概念関係にある。

〈育成における各組織体の役割と限界〉

個々の企業体・公共事業体等にとっては、情報処理技術は、マネジメントに関する戦略の一部として捕捉すべき重要な課題の一つである。情報に関する技術的側面は、(個々の)組織体の存続・発展等に、致命的な様態に関連する。同時に、これに関連しての人的諸問題に関する配慮も、極めて重要である。しかし、一つの組織体のことを、技術体系や情報(処理)システムと見立て、その様な角度からだけの対策に終始しようとするのは矩慮である。情報処理技術に関して、個々の企業体や公共事業体は、(大局的には)状況のすべてを制御・管理しきれない。情報処理技術のうち、ある部分は組織体の中で取組みうる。しかし、外部の環境条件として、対処すべきものも少なくない。「情報処理技術者の育成」についても、そういうことが言える。

3. 想定される環境

3.1 環境考察の概要

情報処理技術者を取巻く、現在から近い将来にわたっての、環境の動向を考え、これらが技術者育成の具体的な場面にどのような影響を与えるかについて考察を行っている。

まず、一般的な環境の動向として、

- (1) 組織体の巨大化と複合化
- (2) 「開発」に関する諸負担の増大
- (3) 情報システムの日常化と定着
- (4) 諸変化に起因する社会的な当惑と混迷

などを列挙した。

次に、このような状況への取組み方については、平面的な見方として、

- ・ オーガニゼーショナルな発想
- ・ テクノロジカルな側面
- ・ マネジメント的側面からの見方

の形で示すことにした。これまでに、しばしば行なわれてきた、〈技術系と事務系〉、〈ゼネラリストとスペシャリスト〉等といった見方もあるが、ここでは「社会システム、技術システム、管理システム」等の関連から、そうした見方を用いないことにした。

現実的な問題として、情報処理における「人的」諸問題がある。ここでは、

- (1) 「制度」内での社会的・人事的構造

終身雇用、年功序列、運命共同体、組織人的な生活意識、など。

- (2) 労務管理的な職務分担の構造

分業体制、編成、管理、など。

- (3) 技術等を基準とする職能価値体系

スペシャリスト、プロフェッショナリズム、など。

といった諸側面について触れた。

また、このような状況下において、情報処理の分野では、

- ・ 法典化された知識が少なく
- ・ 行動における規範が未確立
- ・ 当事者における対応能力の不足

などが見受けられる。

3.2 環境の時代的特徴

情報システムにとって、マネジメント・システムや、システムとしての組織体（行政システム、企業システム、社会システム、など）は、一種の「環境」を形成し、外部的な諸条件ともみなすことができる。

現代における、こうした「環境」の代表的な特徴を挙げると、

- ① 組織体の巨大化と複合化
- ② 「開発」に関する諸負担の増大
- ③ 情報システムの日常化と定着
- ④ 諸変化に起因する社会的な当惑と混迷

といったことがある。以下、これらについて説明を試みることにする。

(1) 組織体の巨大化と複合化

組織体は、マネジメントや情報処理などの諸技術に支えられながら巨大化し、同時にその過程は、技術の進歩・発展の過程でもあった。システムの構成要素数は増加し、また、その数が思いのほか多いことにも気づかれてきた。巨大化する組織体は、同時にまた複合化をも高め、構成要素相互の関係の複雑さについても、また構成要素と同じ事情が観察されてきた。

従来は相互に無関係と見なされてきた事柄の間に、密接なかかわり合いがあると主張されるようになった。広域化、大規模化、多様化、国際化、…といった視点は、その例である。きめ細かい対応や広い視野からの把握などは、しばしば巨大化・複合化のプロセスを加速する。

(2) 「開発」に関する諸負担の増大

情報システムの開発・作成・運用は、このように複雑な様相を持つ組織体での、複合化した問題状況群を客体としつつ、営まれるのが通常である。現在では、いわゆる「規模」からすれば中以下と目される事業体などにおいてすら、その諸活動を集約的に把握したり、多岐にわたることがらについての判断において、関係各者間で「合意」に達するのは、なかなか容易ではない。

(3) 情報システムの日常化と定着

単一の、独立した情報システムは、組織体の中では余り重要な位置を与えられにくい。既に述べたような「環境」の中にあって、情報システム自体もまた巨大化し、複合化している。従って、情報システムに関する状況判断や方策などにおいても、多大な資源などを必要とすることが分ってきた。

情報システムの開発や再開発について、相当な経費負担を要する例があり、このことにつ

いて慎重な考慮や工夫が望まれることも、認識されつつある。

(4) 情報システムの日常化と定着

多くの組織体において、例えばコンピュータなどの用具を含む情報システムは、何らかの形で定着したとか、または定着しつつあるものと認められるようになった。少なくとも、こうした情報システムが、組織体にとって新奇だとする人の数は多くない。

情報処理の技術は、組織体の効率や、(時としては)その有効性について、無視できないほどの影響力を持つようになった。

(5) 諸変化に起因する社会的当惑と混迷

このような、情報システムの日常化や定着の傾向と共に、システムの巨大化・複合化などといった「変化」は、それに関係する人々に適応・対応を求め、時にはこれを強制することともなった。

「適応」という過程の中で、さまざまな出来事を通じて見られた人々の当惑や混迷などは、その一例である。生活意識や生活様式などにおける、模索や再設計なども行なわれている。

3.3 組織・技術・マネジメントからの取り組み方

〈「組織システム」-「マネジメント・システム」-「情報システム」〉という思考上の枠組を想定し、その中で「情報システム」の役割・形態などを検討する。なお、ここで言う「情報システム」は、「情報/データ処理部門」のことではない。

情報システムの機能・役割などは、外部システムとしての組織システム、特に、マネジメント・システムとの接触・関連・調和・整合などの関係において、定まるものとしておく。このような領域に関心を寄せることを、「オーガニゼーショナルな(organizational)発想」と呼ぶことにする。

一方において、情報システムは、組織システムやマネジメント・システムを外部環境とするような、一つの技術システムと見ることもできる。「技術」を手がかりに、情報システムの構造・形態・動態などに目を向けるのを、「テクノロジカル(technological)な側面」と呼ぶことにする。テクノロジカルな角度から捕らえた情報システムは、「実現・実際・具体化」などのかかわり合いによって、「組織行動」においてそれなりの形態・動態をなす。これらの事情に関心を寄せるのは、情報システムの「マネジメント的な(managerial)側面」における見方である。

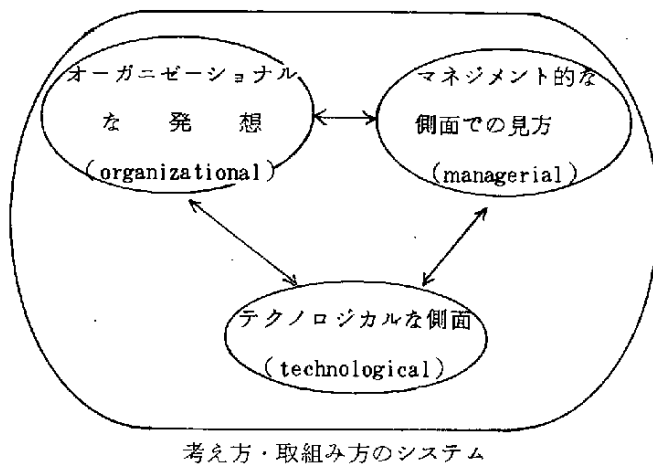


図3-1 情報システムについての考え方・取組み方

これらは、図3-1に示すように、「情報システムについての考え方・取組み方」という、一つのシステムの構成要素として位置づけている。従って、オーガニゼーショナルなもの、テクノロジカルな事柄、マネジメンツ的な観点は、それぞれ同一レベル、並置的な関係に置き、緊密な相互依存の関係において捕らえるのが便利であると思われる。

3.4 情報処理における人的諸問題

情報システムの構成要素として、「人間」に注目してみよう。組織体の中において、情報システムの機能や役割は、要素としての人間で構成される、さまざまな集団・組織活動を通じて実現されるものと考えられる。このとき、集団や組織体は、次のような構造的側面をあわせ持つと想定される。

① 「制度」内での人事的・社会的構造

終身雇用、年功序列、運命共同体、組織人的な生活意識、など。

② 労務管理的な職務分担の構造

分業体制、編成、統括・調整、など。

③ 技術等を基準とする職能価値体系

スペシャリスト、プロフェッショナリズム、など。

これら三者は、しばしば密接な相互依存の関係にある。そして、これらは互いに切り離すべきでなく、同一の事柄の異なる側面であるとしたい状況もある。しかし、そうばかりとは限ら

ない。三者のうち、いずれか一つについて考察するとき、他の二者を外部環境としておくのは、思考上の便宜である。それを承知して、情報システムにおける職能体系の構造化について述べる。

まず、情報システムの内部についての、構造的把握や構造化がおくれている。このことは内部の統制と、外部との責任ある接触とを妨げている。技術職能体系において、他のシステムと比較するとき、何が情報システムにおいて固有なのか、何が共通で一般的なのかが、識別や合意されないまま今日に至っている。これと同時に、情報システムは大規模化・複雑化している。

また、情報処理技術の分野では、法典化された知識(codified knowledge)が少なく、行動形態についての規模(norm)も定まっていない。あわせて、知識や技術の陳腐化・形骸化の過程が進行中である。このために、システム標準の作成や教育構想の立案などが、困難になっている。

第三に、「制度」内での人事的・社会的構造」や、「労務管理的な職務分担の構造」などを含む枠組内での、情報システムに対する取組みの経験が浅い。そのため、情報システムの記述形式や伝達技術が未発達である。これは、さまざまな「評価」の障害ともなっている。こうしたことや、職務分担などにおける細分化の風潮などによって、職務と職能との間の対応関係は不明確になってきた。他方において、組織の要員における高年齢化、再訓練・再配置の必要性、組織的な志気(モラル)と倫理感の低迷、教育・訓練における経費負担の増大などの対処にも迫られている。

総じて、このような状況下では、個々の組織体にとっては、「情報処理技術者の育成」は管理できる事柄とは言えず、公共的な諸サービスを利用しつつ、判断や適応の比重を高めて行かねばならないような事情となっている。

4. 情報処理技術者

この章では、各企業体などにおける実状、育成のシステム、教育制度などとの関連の下で、情報処理技術者の位置づけを行った。

まず、職種や職務などの層別の性格を、これらの分化・発展の系統の中で再確認した。「情報処理技術者」の概念については、若干の手直しは必要であっても、これをさらに細分化すべき段階には至っていないと考える。

情報処理技術者をグループ分けするときの識別基準として、

- ・ 技術的な知(識)的能力
- ・ 社会人としての一般常識と素養
- ・ 実務的行動力

などを挙げ、基準体系の性格について述べた。

これらを前提とし、情報処理技術者の位置を、情報処理技術者育成のシステム、情報処理技術者試験制度、教育制度などとの関連の下に示した。

4.1 情報処理技術者の層別

情報処理技術者の活動分野は多岐にわたるので、育成の方策を立てる際には、何等かの層別が必要である。従来、職種や職務などを手がかりにして、各種の分類や比較が行なわれ、それなりに論じられてきたが、いずれも決めてはなっていないと思う。

とりあえず層別化の便宜のため、コンピュータ・システムなどの利用に携わる「データ処理部門」などでの、社会的・組織上の「分化」の系統について再確認する。この大略の傾向は、図4-1に示すようである。職種または職務の呼び名のうち、社会的な定着度の高いものについては、特に線で囲んで示した。

図4-1では、ラインに関することを中心とし、補助的な職務や、マネジメントに関するものを省いた。このような分化は、データ処理技術の特質や要求だけから生じたものでなく、むしろ労務管理や人事管理などからの発想が濃厚なのではないだろうか。

コンピュータに関すること

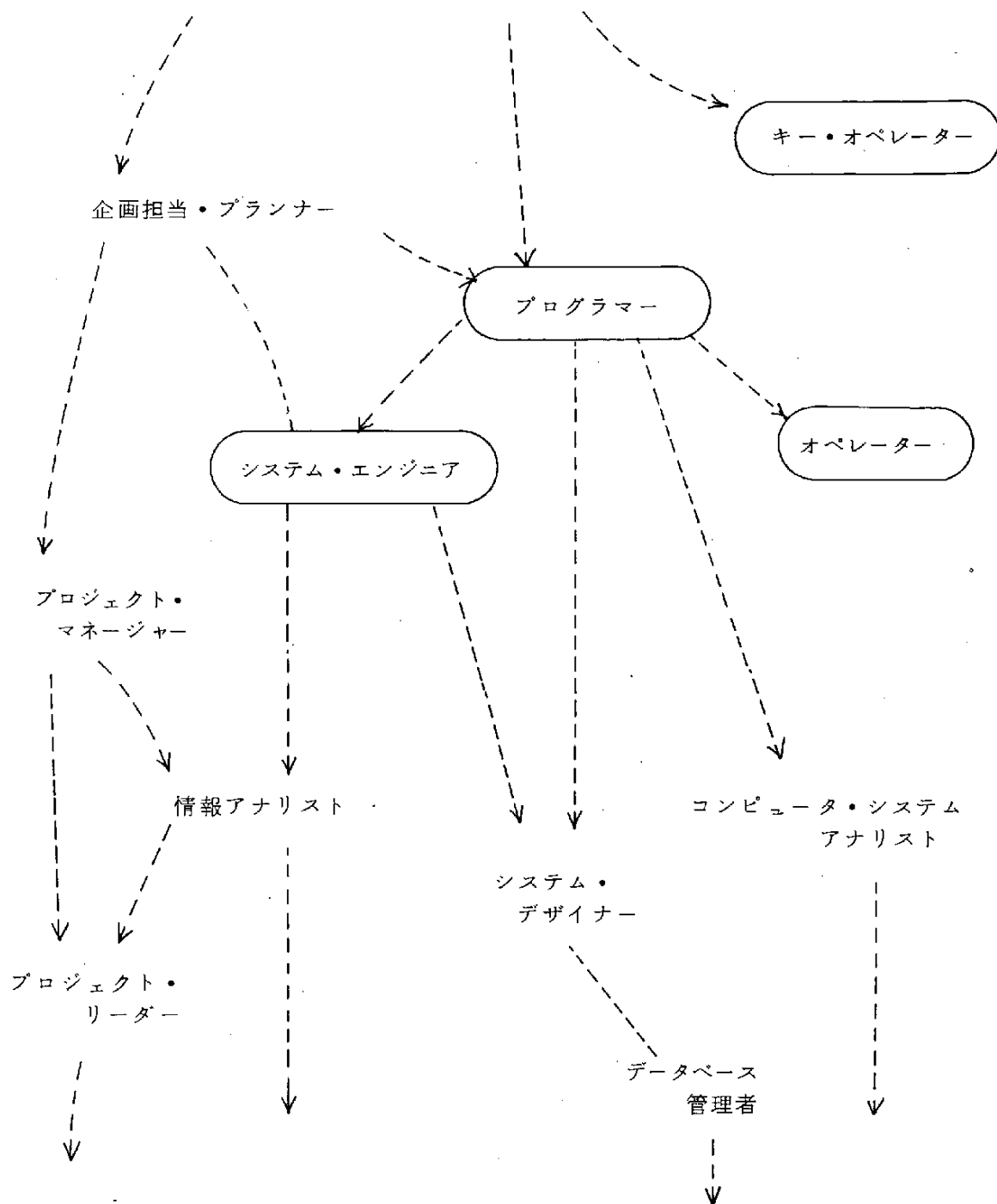


図 4 - 1 データ処理部門等における分化の系統

このことは、「システム・エンジニア」および「プログラマー」について、特に顕著である。どちらも、かなりあいまいな表現であり、あいまいであるが故に、社会的通念として受入れられてきた。図に例示したように、システム・エンジニアやプログラマーは、更に分化する傾向を持っているが、現状は必ずしもそうはなっていない。

一方、労務管理や人事管理などに関する要請から、職務の細分化は、中規模以下と目される組織体においても、日常的である。しかし、だからといって、直ちに情報処理技術者も分化すべきである、ということにはならない。「分化」に対応できる姿勢は必要ではあるが。

4.2 識別の基準

このカリキュラムでは、情報処理技術者をグループ分けするときの識別基準として、

- ・ 技術的な知(識)的能力
- ・ 社会人としての一般常識と素養
- ・ 実務的行動力

を基底に置きたい。

より一層具体的な補足を加えてまとめると、図4-2に示すようになる。

① 技術的な知的能力

- ・ 基礎的能力
- ・ オーガニゼーションな能力と、テクノロジカルな能力
- ・ マネジメント的側面に関する能力

② 社会人としての一般常識と素養

- ・ 学力・学歴などに関する事柄

③ 実務的行動力

- ・ プログラム作成に関する実務能力
- ・ システム設計に関する実務能力
- ・ その他文書作成などに関する実務能力

図4-2 識別の基準

基準体系の内容と、育成システム内での取扱い方について、従来の「育成指針」類とは、次の点で違いがある。

〔技術的な知(識)的能力〕においては、既に開発された指針類や、諸分野における研修の実績と体験、またこれらについての調査や研究などによる、情報の蓄積がなされている。これらに支援され、内部構造をより明確に提示できる段階に達しているものとする。また、技術力のことを、〔実務的行動力〕または〔「実務」状況下における行動力〕と見たときの、研修に関する計画・実施・評価を、より重視したい。

〔社会人としての一般常識と素養〕については、高等学校から大学へ、さらに大学院へと進む、一般の教育課程と対応させながら取り組んできた。しかし、「実務的行動力」については、従来の一般的諸制度ではまかないきれない。また、高齢化・再訓練(リハビリテーション)のような社会構造の変化による諸要請もあり、これらとの関係についても考えておく必要がある。

次に、育成システム内での基準体系の役割のことであるが、このカリキュラムでは、「下限」についての識別のことを、より強調したい。つまり、特定の人が基準を満たしているということは、今後、実務や研修に取り組む際の必要条件を備えていることを意味する。言い換えれば、基準を満たしているから、「望ましい」、「期待される」技術者であるとは言えない。逆に、基準を満たしていない人が、そのまま実務に取り組んだり、研修に参加したりすると、著しい不自由や効率の低下が予想される。現状では、情報処理技術者の水準に関する価値や評価の基準は多様化しており、そのために社会的な評価の用具としての有効性を減じている。

4.3 情報処理技術者の位置

情報処理技術者の能力等に関する判断基準として、「技術的な知的能力——社会人としての一般常識と素養——実務的行動力」から成る体系を想定した。その上で、本カリキュラムの対象となる情報処理技術者について、次のようなグループ分けをした。

- ① 情報処理技術者としての基本的な能力
- ② 上記に関する研修等に参加する上での予備的な基礎能力

そして、ここに挙げた①の能力を更に、

- ・ オーガニゼーションな知的能力
- ・ テクノロジカルな知的能力

の二つに区分した。この区分は、主として力点の置き方や、選択のことであり、この段階であたかも専門化するかのように取扱うのは、大局的に見て好ましくない。また、オーガニゼーションなや、テクノロジカルな基本的能力に達するための研修への参加資格・条件として、情報処理技術の基礎に関する研修が必要である。「基礎」の研修は、情報処理技術や、コンピュータ等についての知識を前提とはしない。以上の事柄を図4-3に示す。

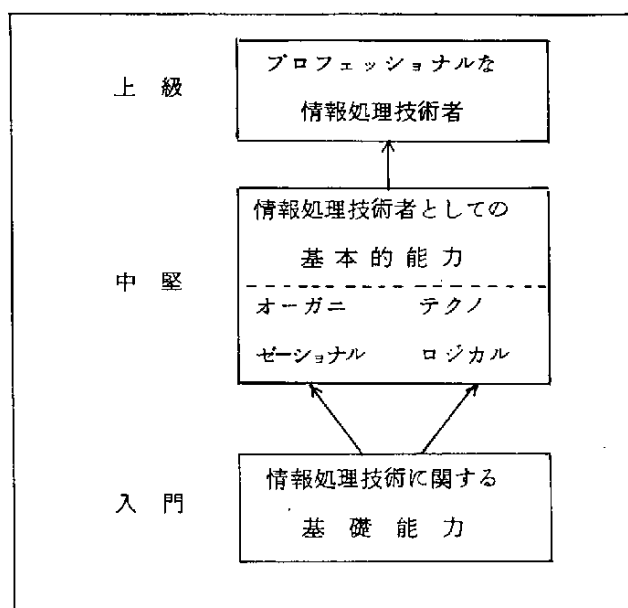


図 4-3 情報処理技術者のグループ分け

次に、情報処理技術者の社会評価面では、「情報処理技術者試験」制度がある。この制度は既に定着してきており、それなりの社会的イメージを生んでいる。この制度における種別と、これまでに述べた情報処理技術者のあり方との間における、近似的に関連や対応の関係を示すと、図 4-4 のようになる。

情報処理技術者	「試験」制度
(オーガニゼーション)	特 種
(テクノロジカル)	第 1 種
(基 礎)	第 2 種

図 4-4 情報処理技術者と試験種別との関係

違いの傾向について述べると、次のようになる。試験制度における特種・第1種の技術者は、社会的な文脈で「上級」であり、また「特種」は「第1種」よりも上位にあるとするようなイメージがある。本カリキュラムでは、オーガニゼーションな技術者育成と、テクノロジカルなそれとは、ほぼ並置的であり、相互補完的な関係に配置される。また、現行制度による

特種および第1種の情報処理技術者は、中堅的な情報処理実務への出発点または入口（entry point）で、必要とされる技術的な知的能力を備えているものと見たい。総じての技術能力については、〔知的能力 — 社会的能力 — 実務的行動力〕での、因果・相関に関する研究も進められるべきである。図4-5によって、本項のとりまとめとする。

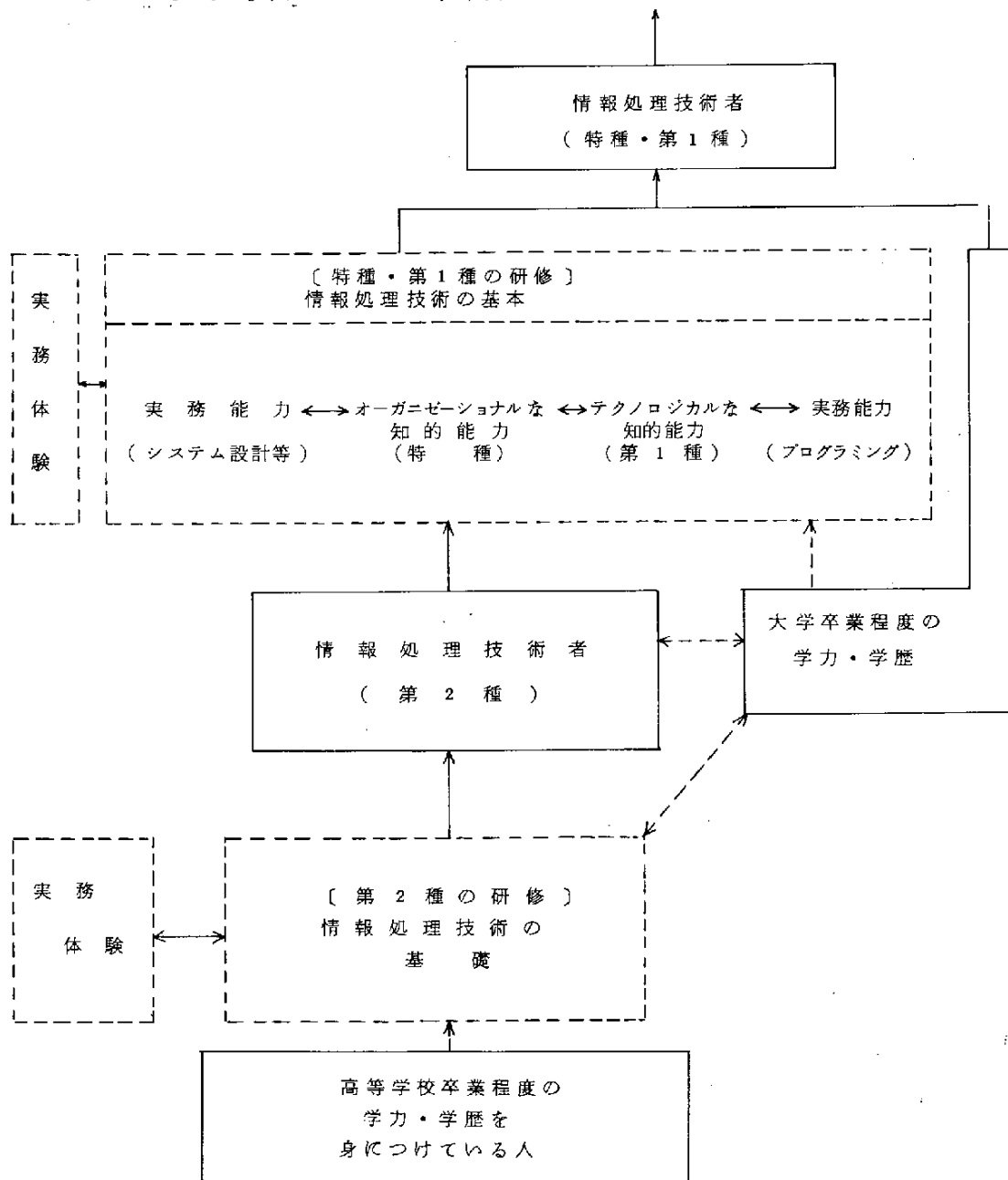


図4-5 情報処理技術者の位置

5. 育成のシステム

5.1 育成システムの構造

情報処理技術者の育成を、一つの社会的システムと見立てたときの構成要素は、例えば図5-1のようになるだろう。

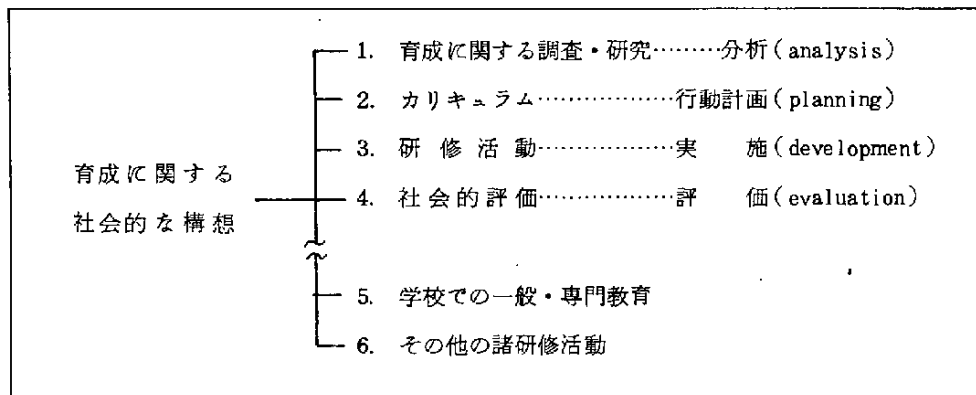


図5-1 社会的な育成システムの構成要素

このような見立てについて補足すると、「育成」は、種々の調査・研究機関、研修組織、企業体、行政体などにおける複合的な活動である。各組織体の諸機能を連係させることによって、有効性と効率性が、さらに発揮されうる。そのため、「育成」に関する、一種の「社会的マネジメント・システム」の側面に注目しておく必要がある。

〔1.育成に関する調査・研究〕、〔2.カリキュラム〕、〔4.社会的評価〕は、基本的に公共的な活動である。一方では、社会性や標準性などに関する配慮が必要であり、他方ではこのための相当な経費が必要ともなるので、ある程度の社会的負担を覚悟しなければならない。

〔3.研修活動〕においては、さまざまな苦慮が見受けられる。従来、実施に関する事柄は、各組織体独自の社内教育や事業所内教育など、あるいは公的・私的な教育・研修機関で扱われてきた。しかし、技術体系における大規模化、分化とか、社会・経済的環境の変化等をあわせての状況は、著しく変わって来ている。

まず、社内教育や事業所内教育の方式では、経費、技術指導力、マネジメント能力等のいずれにおいても、状況への対応が困難であることに気付かれてきた。また、企業体等における組織の構造は、諸種の要因により、細分化とか硬直化の傾向を示している。

職場におけるこのような実態は、情報処理技術者の育成のように、ある程度総体的な発想と両立した形では、にわかには受け入れがたい事情にある。こと情報処理技術に関しては、いわ

める職場内（on the job）の訓練は、すでに有名無実となっている。

しかし、社会全体としての需要の大きさと、技術者の一時的な使い捨て現象の防止とを考え合わせると、「5.学校での一般・専門教育」や「6.その他の諸研修活動」とも連係した方策の再開発が必要と思われる。

5.2 育成指針とカリキュラム

ここでは、「研修の組織行動計画（planning）システム」と、カリキュラムならびに育成指針の関連について述べる。研修の計画プロセスで扱うべき要素の代表例を、図5-2に示す。

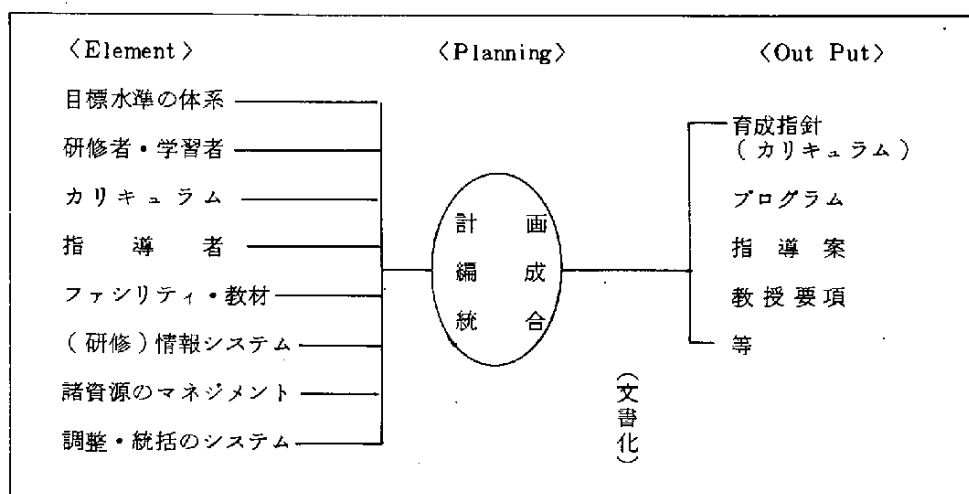


図5-2 行動計画の構成要素

ここでは、カリキュラムを、研修の内容・構成などの狭い意味で用いている。カリキュラムを、目標—内容—方法等にわたる広い意味で解釈すれば、図5-2における「<Planning>」に相当する部分のことを、「カリキュラムの編成」とも呼びかえることができる。「計画・編成・統合」は、一つの過程（プロセス）であり、諸要素（element）は相互関連（interaction）の形で取扱われる。その結果は、文書化された計画案類としてアウトプットされる。「育成指針」もその一つの形態であり、広義でのカリキュラムがその中核をなす。

本報告では、情報処理技術者育成のための「カリキュラム'80」を提唱する。このカリキュラム'80は、全体として、情報処理技術者としての「基本的な諸能力」の養成を目標にしている。これは、図5-3のように、三つの具体的なカリキュラムから構成されている。

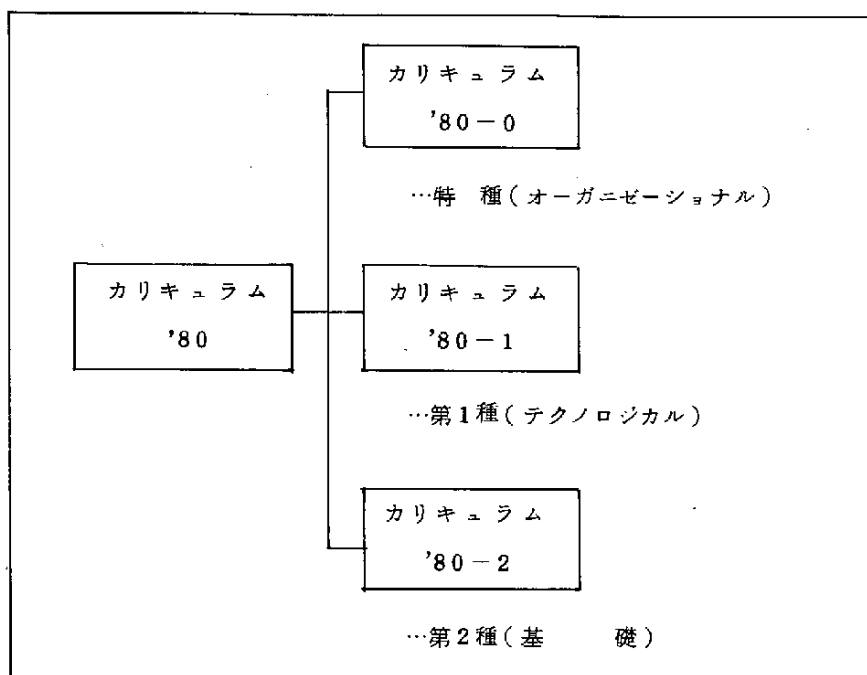


図5-3 カリキュラム '80の構成

「カリキュラム '80-0」では、情報処理技術者の役割のうち、オーガニゼーショナルな側面に重点を置いて、技術者を育成する。情報システムの開発などは、その代表的なテーマである。「情報処理技術者試験」で言えば、「特種」の技術者に近いイメージである。

「カリキュラム '80-1」では、テクノロジカルな側面に重点を置く。ソフトウェアの開発は、中心的な課題の例である。「情報処理技術者試験」では、「第1種」の技術者のイメージが、これに近い。

「カリキュラム '80-2」では、「情報処理技術者試験」で言う、「第2種」の技術者に期待されているような、基礎的な事柄を扱っている。

特種「情報処理技術者」と、第1種「情報処理技術者」との違いは、集中度や関心度、さらに選択における性向などから生ずるものである。両者の間には共通的なものが少なくないので、この段階では、いわゆる「専門化」のこととして受取るのは早計であろう。第2種「情報処理技術者」は、特種および第1種の段階における研修に対して、「予備門」的な性格を持つ。言い替えば、「基礎」的な研修に重点を置いている。

5.3 育成の実施と評価

これらについては、カリキュラムそのものの実体が、ようやくでき上ったばかりであるために、具体的な提言は今後の機会にゆずりたい。そのため、現況において着目すべきと思われることがらについて、若干述べておく。

情報処理技術者に限らず、一般的に育成・教育等の過程（プロセス）は複合的であり、各人にとってあいまいな要素を含んでいる。育成は総じての、あいまいな状況下での、構想・立案・施策である。そのため、教育制度に依存しつつ、また、それとは一線を画したくなるような事情の下で、育成活動が営まれる。

情報処理技術者の評価等については、「技術力——知的能力——実務力」等に関する因果関係や、相関に関する科学的な解明が不十分である。アセスメントの手段・技法の開発と合わせて、既に社会で定着・認知されている、資格とか検定の諸制度についても、それらの経過・実状・意義等にも関心を寄せ、より公正なアセスメント・システムの開発にも貢献できることが望まれる。

6. カリキュラム '80

6.1 カリキュラム '80の性格

わが国で、これまでに作成された育成指針類や、米国における情報処理教育に関するカリキュラム等の開発の系統の中で、「カリキュラム '80」の位置づけをすると、図6-1のようになる。

「カリキュラム '80」は、さし当っての局面では、「初級育成指針」〔開発協会 1971〕と「中級育成指針」〔開発協会 1974〕の改訂である。両「育成指針」が刊行されてから、すでに7年以上が経過しており、技術の進展と諸環境の変容も著しい。改訂は、まずこれらへの対応である。

さらに、米国計算機学会（ACM）によって開発された、大学の学部や大学院教育のための諸カリキュラムは、コンピュータ科学や技術における教育について大きな影響力を持ち、これらについての配慮も必要であった。

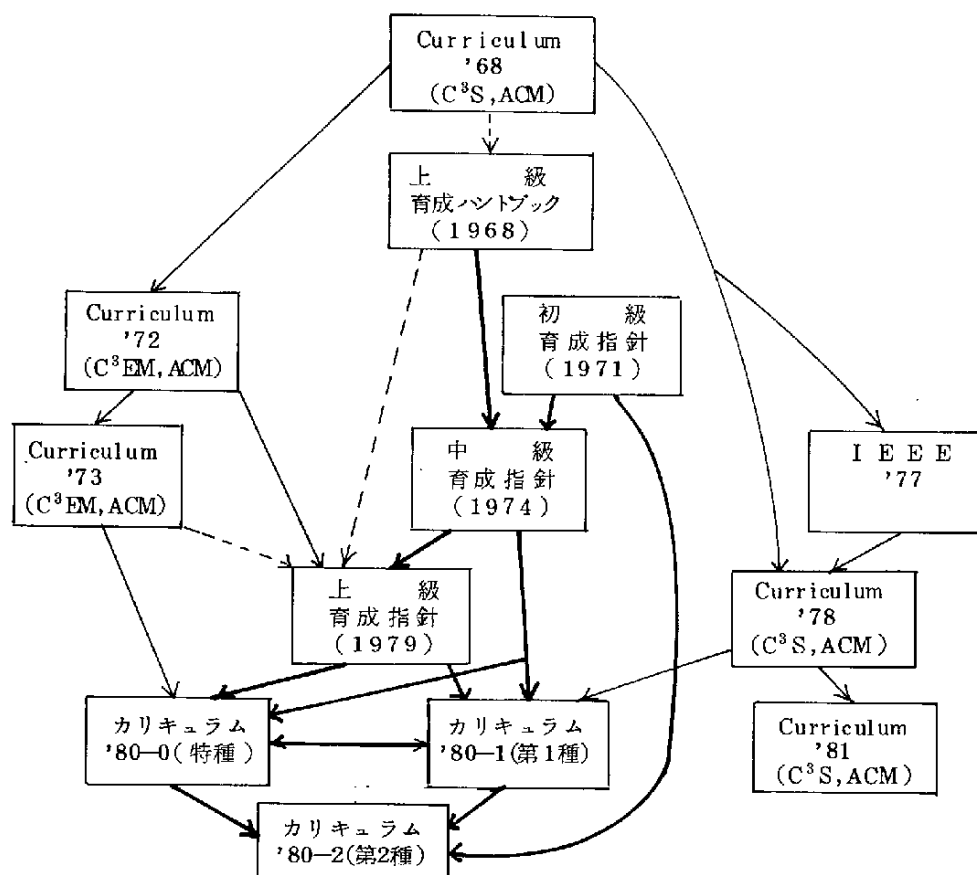


図6-1 諸カリキュラムの系統におけるカリキュラム '80の位置

6.2 特種・第1種のための基本コース

カリキュラム '80-0 および、カリキュラム '80-1 のプランニングに当って、図 6-2 に示すようなコースを設定した。ここで、コース (course) というのは、概念・計画・実施等に関する一つの「作業単位」である。コース概念の設計に当っては、既に作成された諸「育成指針」との間における調和を図りつつ、同時に米国計算機学会 (ACM) によるカリキュラム類、特に次の二つを参考基準とした。

① 「ACMカリキュラム '73」

— 学部水準における「情報システム」の教育計画に関する勧告 (Curriculum Recommendations for Undergraduate Programs in Information Systems)

文献 [ACM 1973]

② 「ACMカリキュラム '78」

— 学部水準における「コンピュータ・サイエンス」の教育計画に関する勧告 (Curriculum Recommendations for the Undergraduate Program in Computer Science)

文献 [ACM 1979]

図 6-2 は、「特種」および「第1種」のカリキュラムにおける、中核的・基本的なコースである。各コースを編成・配置して、「中核的なカリキュラム (core curriculum)」を作る。「中核的なカリキュラム」と、情報処理技術に関する、あるいは、情報処理技術以外の各分野に関するコース類とを組み合わせ、より包括的なカリキュラム体系を設計することもできる。

なお、図 6-2 の各コースを示すのに使った、S1, C1, ...等のコードは、前述の ACM によるカリキュラム中の各コースとの関連や、本カリキュラムの中における位置づけなどを考慮して、設定したものである。本カリキュラム開発中における統合・分割などの作業により、これらのコードには空きが生じた部分もあるが、混乱を避けるため再設定は行なわなかった。

中核的なカリキュラムに含まれない情報処理技術関連コースの扱いについては、次のような方法を考える。

① 基礎的なもの

コンピュータ・システムや、プログラミングの入門・基礎などは、カリキュラム '80-2 (第2種) で扱う。

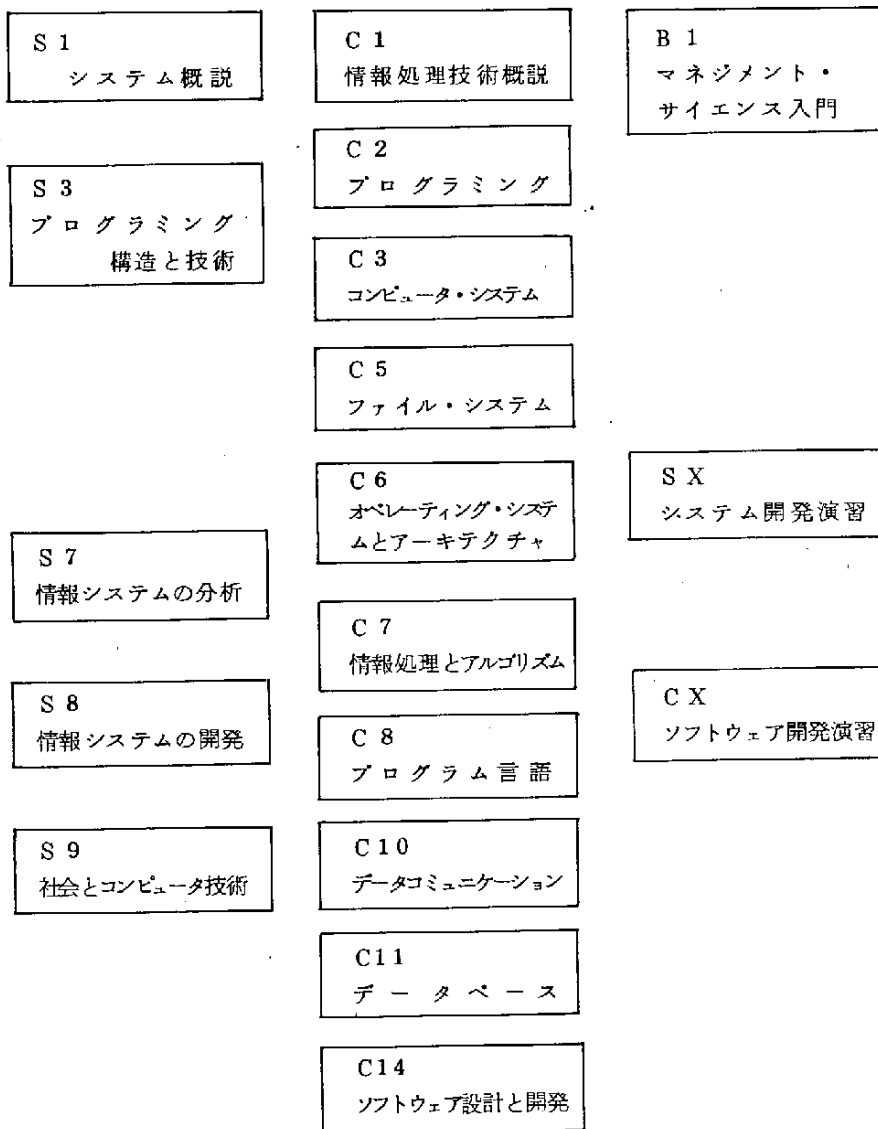


図 6-2 特種・第1種のための基本コース

② 選択的なものと併置的なもの

ビジネス、エンジニアリング、コンピュータ技術等について、各種のコースが考えられる。これらは、情報処理技術以外の各分野に関するコース（アカウンティング、マーケティング、エンジニアリング）と同じような方式で組み合わせる。この場合、研修者は二つ以上のコー

ス体系を履修する (multiple major) ことになる。

③ プロフェッショナルなもの

上級育成指針との関係で考える。

なお、カリキュラム '80 においては、先に述べたように、ACMによる情報システムおよび計算機科学の両カリキュラムについて、その融合を図っている。ACMの最近の報告[ACM 1981a]の中にも、同様の主張が見られるのは興味深い。

6.3 第2種のためのカリキュラム

カリキュラム '80-2 は、第2種情報処理技術者向けのものであり、そのインプットとしては、情報処理に関しての初心者を想定している。また、そのアウトプットは、カリキュラム '80-0 または '80-1 のインプットとしての必要条件を、(少なくとも情報処理に関する基礎的知識や技術に関しては) みたす者を想定している。

これを構成する中核的コースは、「カリキュラム '80-2」の各論に詳述してある通りであるが、これは主として計算機科学の基礎的分野を扱っている。第2種情報処理技術者の育成には、これに関連分野の基礎的知識や技術を習得させるためのコースを、適宜加えるのが望ましい。

7. カリキュラム'80-1 (第1種)

7.1 基本コースとその相互関係

カリキュラム'80-1の中核を構成する各コースについて、他のカリキュラム類との関連性・類似性を示したのが、表7-1である。

また、これら基本コース間の相互関係を示すと、図7-1のようになる。個々のボックスが一つのコースに相当し、矢印は履修の順序を示している。また、これらの基本コースを、主題・内容等によってグループ分けすると、表7-2のようになる。

表7-1 本カリキュラムと他との比較

コースの名称	他のカリキュラムとの関連・類似性		
	「カリキュラム'73」 〔ACM 1973〕	「カリキュラム'78」 〔ACM 1978〕	中級育成指針 〔開発協会1974〕
S1 (第1章) システム概説	UA8 Systems Concept and Implications		情報処理概論, コミュニケーション技法, システム分析・設計(I), 経営実務, システム概論, など
S7 (第2章) 情報システムの分析	UD8 Information Systems Analysis		システム分析・設計(I), 経営情報, など
S9 (第3章) 社会とコンピュータ技術		CS9 Computers and Society	情報処理概論, コミュニケーション技法, など
C1 (第4章) 情報処理技術概説	UC1 Information Structures	CS1 Computer Programming I	ハードウェア(I), ソフトウェア アセンブリ言語
C2 (第8章) プログラミング		CS2 Computer Programming II CS7 Data Structure and Algorithm Analysis CS8 Organization of Prog. Languages	ハードウェア(I), ソフトウェア アセンブリ言語, プログラム設計(II), プログラム言語 (FORTRAN), プログラム言語 (COBOL), など

コースの名称	他のカリキュラムとの関連・類似性		
	「カリキュラム'73」 〔ACM 1973〕	「カリキュラム'78」 〔ACM 1978〕	中級育成指針 〔開発協会 1974〕
C 3 (第5章) コンピュータ・システム	UC 2 Computer Systems	CS 3 Introduction to Computer Systems CS 4 Introduction to Computer Organization	ハードウェア(I), ハードウェア(II), プログラム設計(I), プログラム設計(II), 汎用プログラム・パッケージ
C 5 (第6章) ファイル・システム	UC 3 File and Communication Systems	CS 5 Introduction to File Processing	データ・ベース
C 6 (第10章) オペレーティング・システムとアーキテクチャ		CS 6 Operating Systems and Computer Architecture I CS 10 Operating Systems and Computer Architecture II	オペレーティング・システム, ハードウェア(II)
C 10 (第11章) データ・コミュニケーション		(Special Topics)	データ通信システム
C 11 (第9章) データベース		CS 11 Database Management Systems Design	データ・ベース
C 14 (第7章) ソフトウェアの設計と開発	UC 4 Software Design	CS 14 Software Design and Development	
CX ソフトウェア開発演習			汎用プログラム・パッケージ

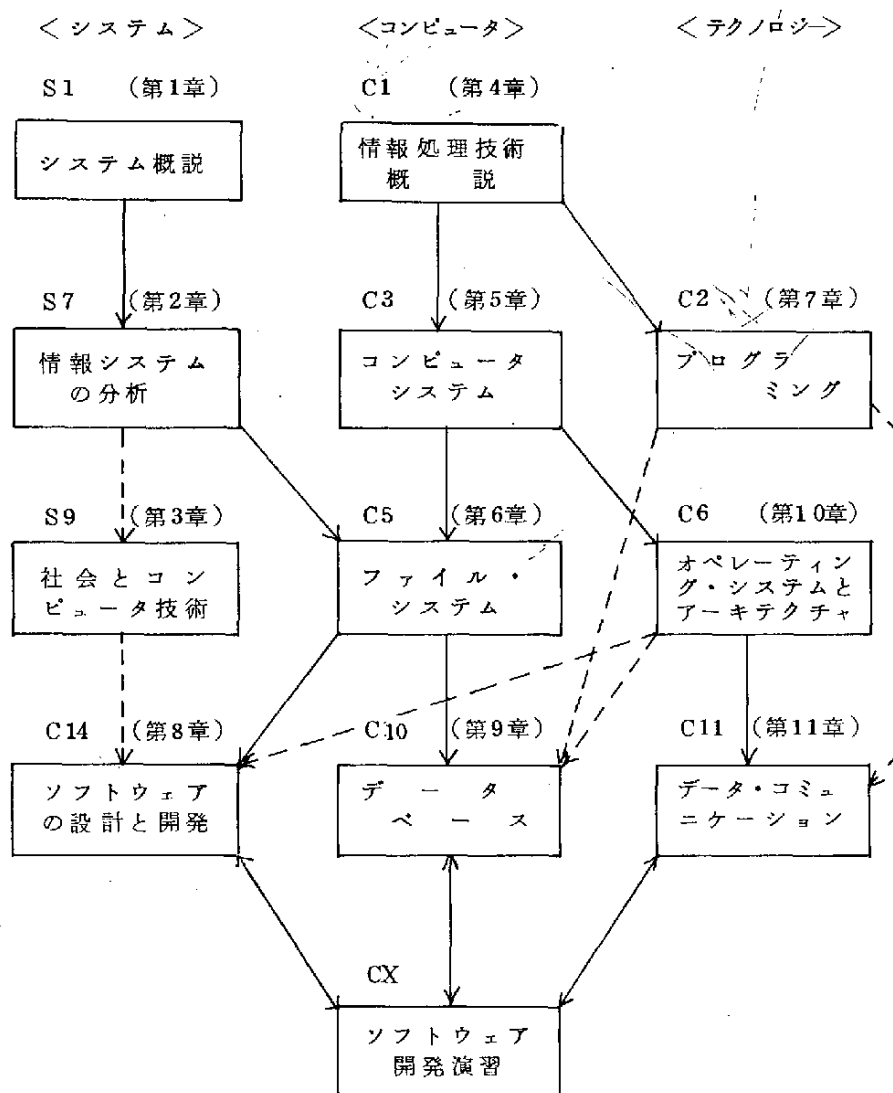


図 7-1 基本コースの相互関係

表 7-2 本カリキュラムを構成するコース

〔コンピュータ技術の基本に関連する系列〕

- C 1 情報処理技術概説
- C 3 コンピュータ・システム
- C 5 ファイル・システム
- C11 データベース
- C14 ソフトウェアの設計と開発
- CX ソフトウェア開発演習

〔システムの環境に関連する系列〕

- S 1 システム概説
- S 7 情報システムの分析
- S 9 社会とコンピュータ技術

〔テクノロジーの習得に関連する系列〕

- C 2 プログラミング
- C 6 オペレーティング・システムとアーキテクチャ
- C10 データ・コミュニケーション

〔予備的な群〕

(数学等に関して、色々なコースが考えられる)

〔コンピュータ技術の基本に関連する系列〕では、情報処理実務における事項を扱う。従って、

- C 1 情報処理技術概説
- C 3 コンピュータ・システム
- C 5 ファイル・システム

の三つのコースは、カリキュラム '80-0 (特種) と、ほぼ共通な位置に配した。講師や教材などが適切であれば、カリキュラム '80-0 (特種) との間で、流用や兼用も可能である。

〔システムの環境に関連する系列〕では、上記のコンピュータ技術関連の諸業務の、背景となる事柄を扱う。カリキュラム '80-0 (特種) との連系である。

〔テクノロジーの習得に関する基本〕では、既に分化し、更に分化を続けているコンピュータ・テクノロジーの、習得に必要な基本的な事柄を扱えるといふ。同時に、分化や専門化を性急に行わずに済ませることが好ましい。

〔予備的な群〕では、本カリキュラムによる研修に当って、必要とされる前提的な知識などを扱う。数学や、プログラミングの初歩等は、その代表例である。その具体的な内容や教育の進め方は状況による。また、この群はカリキュラム'80-2（第2種）における基礎的素養と、本カリキュラムとの接触面（インターフェイス）で、研修者に継続性を確保させる役割も持っている。また、

S 1 システム概説

C 1 情報処理技術概説

についても、継続性の確保を受け持たせたい。

カリキュラム全体についてのガイダンス等は、

C 1 情報処理技術概説

で実施するのがよい。

7.2 各論の構成

カリキュラム'80-1の中核を構成する各コースについては、本報告書の各論に詳述してある。各論では、個々のコースに対し1章ずつを割当て、解説している。各章に対するコースの割当ては、便宜的なもので、その前後関係は特に意味を持つものではない。コースと章との対応関係は、表7-1の中に示した通りである。

各論中の各章は、

- ① 「キーワード」
- ② 「目標」
- ③ 「内容」
- ④ 「指導上の留意点」
- ⑤ 「補足説明」又は「解説」
- ⑥ 「参考文献」

の諸項目から成っている。ただし、章によっては、⑤または⑥の無いものもある。

① 「キーワード」

章（コース）中で扱われる内容を示す、主要事項を端的に示すための用語リストである。

参考のため英語を併記した。

② 「目標」

そのコースにおける学習の目標を示したもの。

③ 「内容」

コースの内容についての一例を示したもので、諸環境の変化や技術の進歩などに適応させながら、実際の場面では適宜修正が望ましい。

④ 「指導上の留意点」

他のコースとの関連、受講者に対する配慮、教材の利用方法、演習や実習などとの関連、その他具体的に教育実施上必要と思われる点について指摘している。

⑤ 「補足説明」又は「解説」

③の「内容」では、記述がやや一般的であるので、これを補足する意味で、具体的な内容事例などを示したものの。

⑥ 「参考文献」

コース内容について、教育計画立案や教育の実施などに際し、参考になるとと思われる文献の紹介。これらは、受講者向けのテキストという意味ではないことに注意。

7.3 「ソフトウェア開発演習(CX)」

図7-1に示したように、

C10 「データ・コミュニケーション」

C11 「データベース」

C14 「ソフトウェアの設計と開発」

の3コースと特に強い関係を持つ形で、コースCX「ソフトウェア開発演習」が設定されている。このコースに関してのみ、各論中には概当の章が用意されていない。

このコースは、カリキュラム'80-1の中における、総合演習的な意味も持っており、特に強く関係のあるのは、C10, C11, C14 の3コースであるが、他の各コースで習得した知識や技術なども、適宜参照することが好ましい。演習の題材としては、受講者のレベル、一般的な環境条件、教育上の制約条件などを考慮して、効果的なケース(事例)を利用するとよい。

また、このコースの実施に関しては、C10, C11, C14 の3コースと、ほぼ同時期に並行的に行ない、C10, C11, C14 などにおける学習事項の理解に役立つよう、工夫することが望ましい。

参考文献

〔開発協会 1970〕

通商産業省編, 「上級情報処理技術者研修ガイドブック」, 日本情報処理開発センター, 昭和45年。

〔開発協会 1971〕

日本情報処理開発センター編, 「初級情報処理技術者育成指針」, 日本情報処理開発センター, 昭和46年。

〔開発協会 1974〕

日本情報処理開発センター編, 「中級情報処理技術者育成指針」, 日本情報処理開発センター, 昭和49年。

〔開発協会 1979〕

情報処理研修センター編, 「上級情報処理技術者育成指針」, 情報処理研修センター, 昭和54年。

〔通商産業省 1976〕

産業構造審議会 情報産業部会, 「昭和60年度における我が国の情報化及び情報産業の計量予測」, 通商産業省, 昭和51年。

〔ACM 1968〕

ACM Curriculum Committee on Computer Science(C³S), "Curriculum '68 — Recommendation for Academic Programs in Computer Science", Comm. ACM, Vol.11, pp.151~97, 1968.

〔ACM 1971〕

Teichroew, D. (Ed.), "Education Related to the Use of Computers in Organization (Position Paper)", Comm. ACM, Vol.14, pp.573~88, 1971.

〔ACM 1972〕

Ashenhurst, R.L. (Ed.), "Curriculum Recommendation for Graduate Professional Programs in Information Systems", Comm. ACM, Vol.15, pp.363~98, 1972.

〔ACM 1973〕

Couger, J.D. (Ed.) "Curriculum Recommendations for Undergraduate

Programs in Information Systems", Comm. ACM, Vol.16, pp.727~49, 1973.

[ACM 1979]

ACM Curriculum Committee on Computer Science (C³S), "Curriculum '78 — Recommendations for the Undergraduate Program in Computer Science", Comm. ACM, Vol.22, pp.147~66, 1979.

[ACM 1981]

ACM Curriculum Committee on Computer Science(C³S), "Recommendations for Master's Level Program in Computer Science", Comm. ACM, Vol.24, pp.115~23, 1981.

[Ackoff 1974]

Ackoff, R.L., "Redesigning the Future", Wiley, 1974.

[Rubin 1970]

Rubin, M., "Introduction to the System Life Cycle(Vol.1)", Auerbach, 1970.

論 各

第1章 システム概説 (S1)

キーワード

システム	system
実体システム	actual system
情報システム	information system
コンピュータシステム	computer system
システムの構造	structure of system
システムの目的	objective of system
環境条件	environment of system
階層構造	hierarchical structure
経営活動	management (business) activity
経営の機能	function of management
経営活動の過程	management process
経営活動の階層	management hierarchy
作業レベル	operational level
戦術レベル	tactical level
戦略レベル	strategic level
コミュニケーション	communication
情報要求	information needs
情報のタイプ	type of information
管理のサイクル	management cycle
意志決定	decision making
生産システム	production system
販売システム	sales (marketing) system
財務システム	financial system
社会的システム	social systems

目 標

情報処理技術者は、主として情報システム、コンピュータシステムを取扱う。しかし、これらの情報システム、コンピュータシステムは、種々の社会的活動（企業活動に代表されるような）を遂行する上でのニーズを満足させるために存在するものである。又ここでも

簡単に「システム」という言葉を使ったが、「システム」という言葉が統一された概念のもとで日常用語として使われているのか？という疑問もある。

本章では

(1) システムとは何か……「システム」の

定義

- (2) システムの種類……情報処理技術者が対象とするシステム
- (3) システムの一般的な構造
- (4) 経営活動（企業活動）とシステムの関係
- (5) システム — 情報システム — コンピュータシステムの役割関係

について学習することによって

- (1) 自分たちの業務の対象としてのシステム概念をしっかりと身につけさせる
- (2) システム的なものの見方を習得させることを目標とする。又同時に
- (3) 情報処理技術者の役割を自覚し、
- (4) 他の各章で学習する種々の情報処理技術の重要性を認識させるとともに
- (5) 社会活動における業務処理実体に関する知識の必要性をも認識させ

これらの知識の自発的な学習を促すことも目標とする。

内 容

1. システムの概念

1.1 システムとは

システムを理解し、システム概念を身につけるための基礎として、まずシステムの定義を明確にする。システムの定義はいろいろあるが平均的には「システムとは、ある目的を達成するために、種々の構成要素（人、物、設備、情報）を、人為的に、合理的・効率的に関連づけた仕組み」と定義される。

ここで重要なことは、システムには「目的」がある、ということである。

1.2 システムの種類

(1) 大別

システムを大別すると

- ① 自然システム……生体システム、自然界の諸現象

- ② 人為システム……社会の各種活動に対応したシステム

に分けられるが、情報処理技術者が対象にするのは人為的システムである。即ち、社会の諸活動を最も合理的に運営する仕組みを作りあげることが、情報処理技術者の役割であることを認識させる。

(2) 層別

- ① 諸活動システム（実体システム）
- ② 諸活動の運営をスムーズに行うためのシステム（情報システム）
- ③ 情報システムを効率的に運営するためのシステム（コンピュータシステム）

一口にシステムというが、夫々上記のような役割、位置づけをもっていることを理解させる。

1.3 システムの構造

(1) 基本構造

システムは基本的に

- ① インプット（を受取り）
- ② （それを）処理し
- ③ （その結果を）アウトプットする

という構造をもっていること。又

- ① アウトプットが目的物である
- ② 処理が目的を達成するための働きであり、この働きをするための仕組みを創造するのがシステム化である。（この働きを機能という）
- ③ インプットには2種類ある。1つは、アウトプットが先に決っており、そのアウトプットを作り出すのに必要となるインプットであ

り（例えば生産システム）、いま1つは、先にインプットが決っていて、それをアウトプットに変えることが目的となるようなインプットである（例えば教育システム）、ことを理解させる。

(2) システムの目的

システムは目的をもっていることを先に説明したが、

- ① あるシステムは、それを含む1つ上位のシステムから見ると手段であること。
- ② 故に、システムを設計する際には出来るだけ上位のシステムを把えて、その目的を明確にし、その目的を達成する手段としての役割を目的として設定しないと、部分最適システムになってしまうこと。（企業間のシステムは、全て企業目的を達成するための手段システム）

を十分説明し、「システムの目的」の意味と、システム目的の明確化が如何に重要かを理解させる。

(3) 環境条件

システムは、その周りの環境条件によって影響を受ける。即ち環境に適応するように作られねばならない（生産システムの場合、高度成長型と安定成長型では異なる）ことを例をあげて説明し理解させ、だからシステム設計に当っては、

- ① 何が環境条件なのか？をはっきりと把えること
 - ② それによって「システム目的」が決められること
- についても理解させる必要がある。

(4) 階層構造

システムの目的のところでもふれたが生産システムや、企業組織の階層構造を例にして、システムは階層構造をもったいくつかのサブシステムによって構成されることを説明する。又この構造を把握する場合には、出来るだけ上位の階層から下位へ分解していくことが重要であることも理解させる。

(5) その他の要素

- ① システムの結合度……関連する他システムへの影響の度合
 - ② 確定的システムか確率的システムか
 - ③ フィードバックコントロールの必要性
 - ④ 環境条件の変化に対する予測……変化に対する対応力
- 等についても言及し、システムを構成する要素について十分理解を深め、全体として「システム概念」を身につけさせることが重要である。

2. 経営活動とシステム

ここで経営活動というのは単に私企業に限らず、全ての組織体において行なわれる活動にあてはまるものである。これらの組織体における経営活動はすべてシステムとして把えるべきものであり、情報技術者の対象になるものである。ここではこのような組織体における経営活動をいろいろの角度から概観し、全般的な知識を与えることを目的に、以下のような内容について学習させる。

2.1 経営の目的

組織体は夫々に目的を持っており、その目的に向かって経営活動を行っていることを、種々の目的の異なる組織体の種類を例示して説明する。

2.2 経営活動の機能

経営活動は大別すると

(1) 経営管理活動……経営管理機能

(2) 作業の執行活動……執行機能

に分けられる。又経営管理機能は

① 計画機能

② 組織化機能

③ 指示機能

④ 統制機能

⑤ 調整機能

に分けられる。これらの各機能について

て具体的にどのような活動が行われる

かについて説明する。

2.3 経営活動の過程

経営活動は、

(1) Plan ……計画の設定（意志決定の過程）

(2) Do ……計画の実施（作業の執行過程）

(3) See ……結果にもとづく計画および実施過程の評価と統制（評価・統制の過程）

の繰返し、および階層によって遂行されていることとその内容について説明する。

2.4 経営活動の階層性

(1) レベル

経営活動は、機能的な分類、過程的な分類の他に領域的（階層的）にも

① 作業レベル

② 戦術レベル（管理レベル）

③ 戦略レベル（経営レベル）

という三つの階層に分類できると、夫々のレベルにおける活動を、

① 機能分類との関係

② 過程分類との関係

③ (2)で説明する経営階層との関係

と関連させながら説明する。

(2) 経営階層（組織の垂直的分化）

組織体においては、経営機能を適宜適切に機能させるため職務権限や管理活動の範囲にもとずいて、その組織に階層構造をもたせている。このような階層構造の存在、例えば

① 経営者層

② 管理者層

③ 監督者層

④ 作業者層

について説明するとともに

① 各層の役割

② 責任と権限

③ 分権と集権

④ 組織体の規模と階層構造…（事業部制）

などについても解説する。

(3) 経営活動の専門化（組織の水平的分化）

組織体での活動は、その活動の主たる対象物や対象領域の違いによって区分することができる。例えば Fayol, H.) によれば、次の6種の基本活動に分けられる。

・技術活動

・商業活動

・財務活動

・保全活動

・会計活動

・管理活動

又最近では、調査や企画などのスタッフ的な活動の専門化も進んでいるのでラインとスタッフの関係についても言及する。これらの各々の専門的な活動について

① 職務の内容

② 組織体の発展や外的環境の変化

と分化の過程

を解説するとともに、この区分は多くの組織体の活動を、共通した枠組みの中で取り扱うことを可能にするものであり、経営システムを考える上での分析上の視点となりうることを説明する。

3. 経営と情報システム

組織体の経営活動は、その組織体を取りまく外部環境とのコミュニケーションと、組織体内部の各組織間のコミュニケーション、即ち情報のやりとり、および処理に基づいてその機能・目的を達成すべく活動をおこなっている。いかえれば経営活動にとって、コミュニケーション（情報）は必要不可欠なものである。そしてこのコミュニケーションを有効に維持、促進させるためのシステムが情報システムである。この経営活動（経営システム）——コミュニケーション（情報）——情報システムの間を、以下の内容によって理解させる。

3.1 経営活動とコミュニケーション

(1) 外部環境とのコミュニケーション

組織体が社会存在である以上、外部との相互コミュニケーションが不可欠なことを、日常的な作業（注文を受けて商品をとどけるといった）から戦略的な判断にいたるまで、いろいろの例によって説明する。

(2) 内部組織間のコミュニケーション

組織体は通常、その目的を達成するために必要な活動を専門に分担する組織に分かれ、それぞれが機能を果している。これらの機能を相互に調整し、統合し、全体として共通の目的の達成につとめるために、コミュニケーションが不可欠であることを説明する。その際、2.2～2.4で

説明した

- ① 機能活動間の関係
- ② 経営活動の過程との関係
- ③ 階層間の問題

と合せ、それぞれにおけるコミュニケーションの役割を説明するとよい。

3.2 経営活動と情報要求

3.1では、コミュニケーションの役割、必要性を説明したが、ここではコミュニケーションのために要求される情報について、いろいろの角度から考えさせ、理解させる。

(1) 組織体における情報

組織体の内外で伝達される情報は、どのような形で組織体の中に存在するのかについて理解させる。

① 情報のタイプ

- ・組織体内情報（内生情報）と組織体外情報（外生情報）
- ・対外（報告用）情報と対内（経営用）情報
- ・公式情報と非公式情報
- ・定形的情報と非定形的情報
- ・文書に記述された情報と非記述情報
- ・計画情報と実績情報
- ・予測（未来）情報と事実（過去）情報
- ・計数情報と非計数情報

といった分類の各々について、例をあげて説明し、組織体における情報システムは、こうした種々のタイプの情報を取り扱わねばならないことを理解させる。

② 情報とデータの違い

- データは
- ・必要な人に
- ・必要な時に

・適切に
利用されてはじめて情報といえる
ことを説明する。

(2) 情報要求の分類

① 経営過程による情報要求の違い 経営過程を

- ・計画の設定
- ・計画の実施
- ・結果の評価・統制

の三つの過程に分け、さらに計画
の設定過程を

- ・情報過程（目的、評価基準、制
約条件の設定、すなわち問題の
明確化を行う）
- ・設計過程（実行可能な代替案を
探索したり、創造する）
- ・選択過程（複数の実行可能な代
替案を評価基準に照して評価し、
一つの実行案を決定する）

の三つに分け、その各々の過程で
の情報要求の違いを特徴づけて説
明し理解させる。

② 経営活動の階層構造と情報要求 の違い

経営活動の階層、すなわち

- ・作業レベル
- ・戦術レベル
- ・戦略レベル

の各レベルに含まれる活動は、そ
れぞれ違った特性を持っており、
それに応じて必要とする情報の性
質やタイプが異なる。各レベルの
活動例をあげて、その特性と必要
とする情報の性質やタイプを説明
し、理解を深める。

③ 機能の違いによる情報要求の違 い

経営活動を構成する機能と、そ

れぞれの機能が必要とする情報の
性質やタイプについて説明する。

④ 専門組織の違いによる情報要求 の違い

企業を例にとれば、専門組織は

- ・生産、加工、製造
- ・購買、調達、資材、外注、仕入、
在庫管理
- ・販売、サービス
- ・人事、労務
- ・企画、調査
- ・研究開発、技術開発
- ・庶務、総務
- ・広告、市場開発、販売促進
- ・経理、会計、原価計算
- ・財務、資金
- ・営繕、保全

等に分けられる。これらの組織に
より、必要とする情報が異なるこ
とを説明する。

⑤ 対象別にみた情報要求の違い 経営資源の夫々、すなわち

- ・人に関する情報
- ・物に関する情報
- ・資金に関する情報

によっても要求される情報の性質、
タイプが異なることを説明する。

⑥ 意志決定者の違いによる情報要 求の違い

レベルによる違いに似ているが、
これを意志決定者という見方で

- ・トップマネジメント
- ・ミドルマネジメント
- ・第一線担当者

といった面から、要求される情報
の違いを説明する。

⑦ 外部からの情報要求

組織体に関係を持つ外部集団も、

その組織体に対して情報提供の要求をもっている。企業を例にとれば

- ・株主
- ・自治体、政府
- ・投資家
- ・金融機関
- ・顧客
- ・地域住民

が外部集団として考えられる。これら外部集団の違いによって要求する情報の性質、タイプの異なることを説明する。

3.3 情報システムの役割

前節で、組織体の活動には種々の情報が要求されること、すなわち組織体の活動にとって情報は不可欠なものであることを説明した。この情報を夫々の活動目的に適合するように

- ① (情報の)収集
- ② (情報の)処理・加工
- ③ (情報の)提供
- ④ (情報の)貯蔵

するための最も効率的・合理的に構成された仕組が「情報システム」であることを説明し、組織体の活動における情報システムの役割、重要性を理解させる。

4. 経営レベルと情報システム

4.1 作業レベルの情報システム

作業レベルの活動は、その組織体の目的を達成するための具体的な日常の活動である。作業レベルの情報システムは、このような意味で上位のレベル(戦術レベル、戦略レベル)の情報システムとは基本的に異なった特性をもっていること、およびそこで作り出される情報が、上位レベルの情報システ

ムの基礎になることを、具体的な例を使って下記の内容によって理解させる。

- (1) 作業レベルの位置づけ
- (2) 作業レベルにおける情報ニーズ
- (3) 作業レベルの情報のタイミング

この際の実例としては、企業を例にし以下のような分類で例示するとよい。

(1) 財務情報システム

- ・原価計算システム
- ・経費予算管理システム
- ・財務諸表作成システム

等

(2) 生産情報システム

- ・生産計画システム
- ・工数計画システム
- ・日程進捗管理システム

等

(3) 販売情報システム

- ・オーダー受付処理システム
- ・代金回収システム
- ・物流システム

等

(4) 人事情報システム

- ・給与計算システム
- ・個人経歴管理システム

等

4.2 戦術レベルの情報システム

戦術レベルは、管理レベルともいえるレベルで、作業レベルからの情報と、トップからの指示、および環境条件の制約にもとづき、主としてミドルマネジメントが意志決定をするレベルであるから、戦術レベルの情報システムには、ミドルマネジメントの意志決定を支援する仕組が組み込まれていなければならない。このことを以下の内容によって理解させる。

- (1) 管理のサイクル
- (2) 戦術的意志決定
- (3) 意志決定の内容と必要な情報
- (4) フォローと評価

4.3 戦略レベルの情報システム

戦略レベルとは、経営の意志決定レベルであり、主体はトップマネジメントである。

トップマネジメントは、どのような意志決定をしなければならないか、経営の意志決定はどのような特徴を持つかを、以下の内容により説明し、戦略レベルの情報システムが、経営の意志決定を支援するために如何なる仕組みを持つべきかを理解させる。

- (1) 戦略レベルと戦術レベルの差（戦略レベルの特徴）

- (2) 意志決定の場

- ① 将来の外部環境変化への対応

この対応には

- ・ 攻撃的対応
- ・ 環境適合対応

の二つのタイプがあること、特に環境適合対応の例として、ポートフォリオ戦略について説明するとよい。

- ② 総合経営計画

5. 経営活動と情報システム

組織体の活動をシステムとしてとらえ、その活動を支援するのが情報システムであることを理解するには、企業システムを例にするのが最も分りやすい。

5.1 主要業務と相互関連

企業における主要業務を生産・販売・財務の三つの機能に分け、それぞれの機能とその相互関連について説明し、企業活動の仕組全般をマクロ的に理解させる。

- (1) 生産
 - ・ 生産の要素
 - ・ 生産要素の調達
 - ・ 生産方式
- (2) 販売
- (3) 財務
 - ・ 資金管理
 - ・ 企業会計
- (4) 生産と販売の関係
- (5) 販売と財務の関係
- (6) 財務と生産の関係

5.2 生産システム

生産を、製造業におけるそのみでなく、行政のサービスまでを含めた一般論として、生産活動の目的と具体的な活動について理解させるため、以下の内容について説明する。

- (1) 生産システムの設計

- ① 生産システムの範囲と定義
- ② 生産システムの目的
- ③ 製品開発と設計
- ④ 標準化
- ⑤ 生産設計と工程計画
- ⑥ レイアウト

- (2) 生産システムの計画と管理

- ① 生産計画と管理の機能
- ② 生産計画と予測の機能
- ③ マスタースケジュール
- ④ 日程計画
- ⑤ 在庫管理
- ⑥ 進度管理

- (3) 生産情報システム

これまでの説明をもとに生産情報システムの持つべき特徴を理解させる。

- ① 総合的性格とシステムの発展性
- ② 変化への対応の必要性
- ③ 処理手続の実用性

- ④ 物と情報の一体性
- ⑤ システムの新陳代謝

5.3 販売システム

企業における販売活動の位置づけと、外部環境との関係を説明し、販売システムの仕組みを理解させる。

(1) 企業における販売システムの位置づけ

- ① 外部環境と販売システム
 - ・商業環境との関係
 - ・技術環境との関係
 - ・政治社会環境との関係
 - ・文化環境との関係
- ② 企業内他システムとの関係
 - ・生産と販売
 - ・仕入と販売
 - ・技術と販売
 - ・経理と販売
- ③ 販売システムの内部機能
 - ・販売活動
 - ・市場分析
 - ・価格政策
 - ・製品計画
 - ・流通管理

(2) 業態別販売活動

- ① 中間材仕込生産における販売
- ② 生産財注文生産における販売
- ③ 消費材の販売
- ④ 輸出

(3) 販売情報システム

これまでに説明したことをもとに、販売情報システムが、次の機能を持つべきことを理解させる。

- ① 販売業務を支援する。
- ② 販売業務活動の状況や結果の情報を販売管理機能へ提供する。
- ③ 多種多量の周辺情報から必要な情報を正確かつタイムリーに抽出

する。

- ④ 外部から収集した情報を整理分析し社内の必要部署に提供する。

5.4 財務システム

企業システムの要素は、人・物・設備といわれるが、これらの要素の調達・運用・処分には必ず資金が動かされる。このような各企業活動と資金の関係や資金管理（財務管理）の仕組みを説明することにより、経営全般の総合的把握の仕方を理解させる。

(1) 企業活動と資金

(2) 資金の調達

- ① 資本金
- ② 社債
- ③ 借入金
- ④ 買掛債務・未払債務
- ⑤ 内部留保資金

(3) 資金の運用

- ① 資金運用の形態
- ② 資金運用の手続

(4) 資金の回収

- ① 代金の請求
- ② 債権の管理

(5) 資金繰管理

- ① 資金計画
- ② 資金収支管理
- ③ 資金管理の道具
 - ・資金繰表
 - ・資金運用表
 - ・資金移動表

(6) 財務会計と管理会計

(7) 経営管理分析

- ① 原価分析
- ② 生産性分析
- ③ 資金管理分析
- ④ 利益管理
 - ・付加価値分析

・資本回転率分析

・損益分岐点分析

6. 社会的なシステム

これまで企業システムを中心に説明してきたが、一般の社会生活に関連するシステムについても説明し、システムは身のまわりのあらゆる場面に存在することを理解させることによって、システム概念を完全に理解させ、あわせて情報技術者の役割の重要性を認識させる。社会的なシステムとしては、

- (1) 行政システム
- (2) 医療システム
- (3) 交通システム
- (4) 情報伝達システム
- (5) 配送システム
- (6) 教育システム
- (7) エネルギー供給システム
- (8) 商店システム

等について説明するとよい。

指導上の留意点

1. 上級情報処理技術者の大部分がプログラミング技術の習得から出発する現状では、ともすればコンピュータ中心の思考から抜けきれないことがある。システムという概念についてもなんとなくコンピュータと密着したものという誤解が生じやすく、コンピュータを導入することがシステム化と考えられていることが多い。ここでは「システム」という概念がもっと広い分野で使われていること、広く適用性のあるものだということを徹底して認識させる必要がある。
2. そのため、概念を説明するとともに出来るだけ具体例をあげて説明するよう心掛ける必要がある。
3. 経営情報システムを経営システムの中に

正しく位置づけることが出来る能力の育成を目指し、特に

① 経営（企業のみでなく、一般の組織体として考え）をシステムとして把握する方法

② 経営活動における情報の役割について、実戦的な知識・能力の付与を目標に指導する。

4. 企業システムを取り扱う場合には、対象業務の全体システムにおける位置づけを十分に把握・理解した上でシステム化にとりかかるべきであることを念頭に、企業行動の本質、企業の各機能の関係を習得させると共に、企業活動を知る上でより深く学習すべき項目、ポイントについても指導することが望ましい。
5. 組織体の活動という目をはなれ、個人個人が社会生活を営む上でかかわりのある現象すべてが「システム」であることを理解させることにより、「システム」の概念をより具体的な知識として身につけさせる。
6. この章の内容は情報システムの分析以降の各章での学習の基礎であることをよく認識させること。

参考文献

- (1) 前川良博他共著 「経営情報管理」(経営工学シリーズ10) 日本規格協会 1981
- (2) 日本情報処理開発協会 情報処理研修センター編 「上級情報処理技術者育成指針」(各論1 組織システムの分析) コンピュータエージ社 1979
- (3) 小野寛徳著 「経営事務論」(経営学全書27) 丸善 1970
- (4) 北川敏男著 「組織と情報」(NHK現代科学講座5) 日本放送協会 1966

- (5) 宮川公男著 「意志決定論」 (経営学全書 21) 丸善 1975
- (6) C・H・ケブナー, B・C・トリガー著 上野一郎訳 「管理者の判断力」 産業能率大学出版部 1972
- (7) 今井賢一, 岡本康雄, 宮川公男編 「企業行動と経営組織」 日本経済新聞社 1971
- (8) デァディン (Dearden, J.) 他 安藤嘉昭他訳 「新・経営情報システム」 建帛社 1976
- (9) サイモン (Simon, H. A.) 松田武彦訳 「経営行動」 ダイヤモンド社 1968
- (10) 石田武雄著 「経営システム論」 (経営学全書 26) 丸善 1970
- (11) ファイヨール (Fayol, H.) 都築栄訳 「産業並に一般管理」 風間書房 1958
- (12) 山城章著 「経営原論」 (経営学全書 1) 丸善 1970
- (13) 米花稔著 「経営環境論」 (経営学全書 6) 丸善 1970
- (14) 山城章他共著 「各種経営学」 (経営学全書 8) 丸善 1970
- (15) 高田馨, 細井卓編 「日本の経営の特質」 ダイヤモンド社 1974
- (16) 津田真澄著 「日本的経営の論理」 中央経済社 1977
- (17) 岡本康雄著 「現代の経営組織」 日本経済新聞社 1976
- (18) R. リッカート著 三隅二不二訳 「組織の行動科学——ヒューマン・オーガニゼーションの管理と価値」 ダイヤモンド社 1968
- (19) 郷原弘著 「ラインとスタッフ」 日本経済新聞社 1970
- (20) 中井重行編 「生産管理」 (現代経営学全集第13巻) ダイヤモンド社 1971
- (21) 通商産業省産業構造審議会 管理部会編 「工程管理」 日刊工業新聞社 1971
- (22) Samuel Eilon 「Elements of Production Planning and Control」 Macmillan London 1962
- (23) 吉谷竜一編 「生産システム設計ハンドブック」 日刊工業新聞社 1972
- (24) 清水晶著 「マーケティング経営論」 (経営学全書 30) 丸善 1970
- (25) P. Kotler 「Marketing Management」 Prentice-Hall 1972
- (26) 内藤和美著 「販売情報作戦 88 の要訣」 ビジネス社 1977
- (27) 吉田彰著 「意志決定のための管理会計」 同文館 1974
- (28) 河野豊弘著 「財務管理」 ダイヤモンド社 1968
- (29) 加藤久蔵著 「会社経理の知識」 日本経済新聞社 1967
- (30) 秋葉中編著 「経営管理分析の考え方・すすめ方」 同文館 1972

第2章 情報システムの分析 (S7)

キーワード

環 境	environment
コミュニケーション	communication
情報要求	information needs
情報システムの役割	role of information systems
人間行動	human behavior
情報システムの開発	development of information systems
開発の動機	motive of development
システム・ライフサイクル	system life cycle
目 的	objective
予備検討	preliminary investigation
システム分析	system analysis
代 替 案	alternatives
評価基準	criteria
分析手法	techniques for analysis
面 接	interview
事務分析	job analysis
予測の手法	techniques for forecasting
P P B S	planning programming budgeting system
評 価	evaluation
システムの評価尺度	measures of system performance
直接効果と間接効果	direct and indirect benefits
システムの費用	cost of system
システム仕様書	systems specification

目 標

S1において、システムの概念とともに、情報システムは組織体の活動に貢献するものであること、逆に云えば組織体の活動は情報なくしては成立たないことを学習した。この章でも今一度、組織体の活動における情報システムの役割を再確認するとともに、情報の活用主体である人間の行動（心理）について

も学習し、情報システム分析に当る際の心がまえや検討態度を理解させた上で、

- ・システム開発における情報システム分析の位置づけ
- ・システム分析の各種技法

について学習し、情報システムの開発に当って真に役立つ上級情報処理技術者の育成を目

標とする。

内 容

1. 組織体の活動と情報システム

1.1 組織体を取りまく環境

組織体はそれを取りまく

- ・経済的環境
- ・社会的環境
- ・政治的環境

の中で、その組織体の目的を達成するための諸活動を行っている。夫々の環境条件が組織体の各活動にどのような影響を及ぼしているかを具体的に整理、理解させる。又環境条件自体も他の環境条件の影響を受けていること、一つの環境条件の変化が他の環境条件の変化につながり、ひいては組織体の活動も複雑な影響を受けることを説明し、組織体の目的を効率よく達成するためには、環境条件の影響の正しい認識と同時に、その変化を早期に把握・認識し、適正に評価し対策を立てることの重要性についても理解させる必要がある。ここで、変化には

- ・替在的な要因の発生
- ・具体的な変化の兆候出現
- ・変化の発生（一般的認識）

の三段階があること、早期に発見する（予測する）ためには、十分な問題意識と幅広い知識が必要であることも強調しておく。

1.2 組織体の活動と情報システム

組織体が、社会的な存在として、また複数の人間からなる集団として、機能を分担しながら目的を達成すべく活動するためには、外部環境との間の相互コミュニケーションと、組織体にお

ける構成員間、ならびに部分組織間の相互コミュニケーションが不可欠であること、およびこのコミュニケーションの内容が情報であり、コミュニケーションを有効に維持・促進させるためのシステムが情報システムであることを、以下の内容によって理解させる。

(1) 組織体の活動とコミュニケーション

- ① 外部環境とのコミュニケーション
- ② 内部部分組織間のコミュニケーション

③ 構成要員間のコミュニケーションに分けて、具体的にどのようなコミュニケーションが行われるかを例をあげて説明する。

(2) 組織体の活動と情報要求

組織体における活動を

① 経営過程からみた区分

- ・意志決定の過程
- ・作業の執行過程
- ・評価・統制の過程

② 階層構造からみた区分

- ・作業レベル
- ・戦術レベル
- ・戦略レベル

③ 経営活動に含まれる機能からみた区分

- ・管理機能（計画、指示、統制、調整等）
- ・各専門職務の執行機能（会計・経理、製造、購買、販売、物流、人事・労務、総務等）

④ 組織階層（人）からみた区分

- ・経営者層
- ・中間管理者層
- ・下位管理者層
- ・担当者、作業者層

というように、いろいろの角度から区

分し、夫々の立場において、外部、構成員間、組織間に行われるコミュニケーションの際にどのような情報が要求されるかを説明する。学習者の身近な職務については自ら考えさせるとよい。

(3) 情報システムの役割

以上で、組織体の活動のいろいろの場面で要求される情報について学んだが、組織体の活動がいくつかの部分組織に分かれ、かつ組織階層をもち、その集合体として目的の達成をはかっていることから、そのコミュニケーション、すなわち情報の伝達も複雑な関連を持っている。このような情報について

① 情報の伝達経路を規定する

② 情報間の関係（一つの情報を生み出すために必要とされる情報）を明確にする

③ 情報を作り出す手続・手順を明確にする

ことにより、コミュニケーションを有効に維持・促進させ、もって複雑にいら組んでいる組織体の活動を一個の有機体として統制された首尾一貫したものに保つものが、情報システムであることを十分に説明し理解させる。

2. 人間行動と情報システム

情報システムを真に効率よく、活用しやすいものにするためには、情報の処理、伝達の手順・手続の検討に当って人間工学的な配慮が必要であることを。

(1) 人間が持っている諸特性、能力

- ① 情報検出能力
- ② 情報処理能力
- ③ 記憶能力

・ 記銘、保持、想起

- ・ 再生、再認
- ・ 忘却
- ・ 連想 等

④ 操作能力

⑤ 信頼性

⑥ 人間の持つ特徴

- ・ 学習能力
- ・ 発見的問題解決能力
- ・ 能力の個体間、個体間変動
- ・ 動機づけ

(2) 環境条件の影響

① 照明

② 色彩

③ 騒音

④ 温熱条件

(3) 集団における問題

① 地位と役割

② 組織構造と作業効率

について説明することによって理解させる。

なお、コンピュータを活用する場合には、この他に

(4) データの入出力、およびその機械操作（マン・マシン・インターフェイス）における人間工学的な問題

① 視覚

② 時間感覚 等

についても考慮が必要なことを説明する。

しかし、いずれの場合にも、ここでは基本的な説明にとどめ、より詳しい知識の習得には心理学の学習が必要であることを付言し、そのための参考文献の紹介をするとよい。

3. 情報システムの開発

3.1 情報システム開発の動機

組織体の活動には情報システムは不可欠であるから、組織体の存在するところには必ず情報システムが存在する。

だから、情報システムの開発といわれるものの殆んどは現状の改善である。

このような改善の動機には

- ① トップマネジメントからの指示
- ② 実務部門からの要求
- ③ 情報システム部門主導

の三つがあり、夫々のケースによって

- ① 要求の明確度
- ② 影響範囲の大きさ
- ③ アプローチの仕方が異なることを説明する。

3.2 システムのライフサイクル

システムは

(1) 開発過程

- ① 問題の提示とニーズの把握
- ② 予備検討
- ③ システム設計
- ④ コンピュータシステム設計
- ⑤ プログラミング

但し④、⑤はコンピュータシステムの場合のステップ

(2) 実施と運用過程

(3) 問題の発生 — (1)へもどる

の三つの過程のライフサイクルをもっていることと、夫々の過程での作業内容の概略を説明し、システム開発過程におけるシステム分析の位置づけ、および重要性について理解させる。

3.3 開発プロジェクト

情報システムの開発は、情報処理技術者や情報システム部門が中心的な機能を果たすることが多いが、ユーザ部門の役割、特にユーザ部門管理者の役割が重要である。情報システムの開発プロジェクト管理について

- (1) マイルストーンの設定（工程の細分）
- (2) 開発ステップにおける情報システ

ム部門とユーザ部門の役割

(3) 進捗状況の把握

(4) 各マイルストーンにおけるユーザ部門管理者の役割

を説明し、システム開発の効率的な進め方を理解させる。

4. システム分析

システム分析は、システム開発の過程の中の

- ① 問題の提示とニーズの把握
- ② 予備検討

のステップをさす。システム分析の過程は、次のステップであるシステム設計の方向を規定し、ひいては創出されるシステムの成果をも左右する重要なステップであることを強調し、以下の内容について十分理解させるとともに、いろいろな対象のシステムに対して最適なアプローチをするためには、出来るだけ多くの分析手法を身につけていることが大切なことを認識させ、分析手法の学習意欲を持つよう指導する必要がある。

4.1 システム分析のステップ

システム分析のステップは、

(1) 目的の明確化

- ① 問題の明確化
- ② 目的の選択

(2) 代替案の作成

(3) 代替案の評価・選択

4.2 システム分析の特徴

- (1) システムをその背景との関係で構造のあるものとしてとらえる。（構造）
- (2) 問題の本質を細分化することによって明らかにしようとする。（解析）
- (3) 問題の不確実性を認識して、解決法を唯一つだけあげようとせず、代替案を多くとりあげて検討する。（不確実性）

- (4) 解決の方向として、与えられた問題だけを考えるのではなく、その目標にまでさかのぼって考えてみて、より根本的な解決を見出そうとする。

(目的)

4.3 システム分析の要素

- (1) 目的
- (2) 代替案
- (3) モデル
- (4) 費用
- (5) 効果
- (6) 評価基準

この中で、費用・効果の推計や要因の変動による結果の変化を予測するのにモデルが重要な役割を果たすことを説明する。

4.4 システム分析の具体的作業内容

- (1) システムの範囲の位置づけ
システム対象を定義する。
- (2) システムの構成要素の決定
設備、人間、組織、時間、情報、等
- (3) 開発の立場を明らかにする
 - ・現状改善か、新システムの開発か。
 - ・全体システムか、部分システムか。
- (4) 業務分析、予備検討をおこなう
 - ① マスター・プランを作る
 - ・要望と制約の調整
 - ・範囲、目標の設定
 - ・システム実現のための環境を保証する
 - ② 予備検討
 - ・必要性、実現可能性、継続保守の保証
 - ・サブシステム
 - ・開発スケジュール
 - ③ 承認
 - ・現状分析

- ・モデル作成
- ・システム提案

5. システム分析の手法

(1) 情報収集の技術

① 面接の技術

情報収集活動の中で効果的な方法であり、情報処理技術者のマスターすべき要件になりつつあるので

- ・情報収集方法のなかの面接の位置づけ

・面接の方法、面接の実際手順について、コミュニケーション理論を参照しながら説明する。

② 情報整理の技術と一体になっている技術

- ・KJ法
- ・ゴードン法
- ・オズボーンの方法
- ・日本ユニパック社のNUPS法等について説明する。

(2) 情報整理の技術と発想の技術

現行システムに関する実態調査が進展した段階で、収集した情報の整理を行う。この整理の過程での工夫が、次の発想へと結びつく。ここでは情報整理と、整理した情報から提案を導き出す技術について説明する。

(1)―(2)の他に

- ・梅棹忠夫氏
- ・中山正和氏

等の提案が参考になる。

(3) 事務分析

問題の発見と現状分析に役立つ事務分析フローチャートを中心に説明する。フローチャートの書き方には、NOMA式、日能式、能大式等がある。

(4) IE的分析手法

生産工程での生産性向上と管理のレベ

ルアップに、情報処理技術が適用される
場合が増大するにつれ、情報処理技術者
が I E の基礎知識を必要とする場合が多
くなっている。ここでは、情報処理技術
者が最低限要求される基礎知識を中心に、
I E の基本的なアプローチの仕方につい
て理解させる。

学習項目としては

- ・特性要因図
- ・バレー図
- ・工程分析（作業分析、動作分析）
- ・価値分析（V.A., V.E.）
- ・時間分析
- ・機能分析（ワーク・デザイン等）

(5) 予測の手法

予測の手法は、システム分析のいろ
ろの場面で必要になるので、手法の技術
的側面のみでの学習ではなく、どうい
う場合に利用されるか、その場合の前提は
何か、その手法の限界は何か、といった
側面を理解させることが肝要である。次
にあげるような各種の予測手法を説明
する。

① 客観的予測

主要な手法として、グラフ化、指数
平滑法、季節調整法、回帰分析を用い
ての予測、微分方程式による決定論的
予測、確率モデルによるものなどにつ
いて、概要、特徴を理解させる。

② 主観的予測

アンケート調査、デルファイ法、関
連樹木図による予測などについて、説
明する。

③ システム的予測

種々の要因が複雑にからみあってい
る問題に対する予測で、コンピュータ
の利用が可能になってから特に発達し
た技法である。

- ・計量経済学モデル

- ・産業関連モデル
- ・システムダイナミクス
- ・クロス・インパクト・マトリックス
法
- ・シナリオ・ライティング
- ・各種シミュレーション

(6) システム開発技法

システム開発技法としてまとめられて
いる手法には、分析段階と設計段階が
一体になっているものが多い。次のもの
中に参考になるものがある。

- ・IBM社のIPT
- ・ミンガン大学のISDOS
- ・TRW社のSREM
- ・Sof Tech社のSADT
- ・BOEING社のSSDM

(7) 分析結果の整理および表現

分析結果の整理表現には次のような方
法が有効なヒントを提供する。

- ① ディンジョン・テーブル
- ② マトリックス・デザイン
- ③ HIPO
- ④ KJ法
- ⑤ ワークデザイン

(8) 総合的な分析技法

問題の把握から代替案の評価決定、将
来の予測まで、四つの段階の分析技法を
まとめたものとして「KT法」があり、
非常に参考になる内容を含んでいる。

6. システム代替案の検討

システム分析の基本要素のうち、システ
ムの目的と代替案の考え方について、さら
にくわしく学習する。

6.1 システムの目的と代替案

問題の本質は分析のプロセスで次第
にあきらかにされてゆく。また、目的
と手段とは相互作用関係にあり、一定
の目的に対して最善の手段を選択する

といった単純なプロセスでないのが普通である。表面的には代替案を選ぶようにみえる問題も、その実は目的の選択の方が本質的であることが多い。このような目的と手段の連鎖関係を説明することにより、目的の選択と代替案の設計・選択が同程度に重要であることを理解させる。

6.2 代替案の開発と選択

システムの目的が設定され、それが分析的に記述されると、代替案を列挙することになる。代替案の列挙は種々の異なる観点から考えなければならぬ。観点として

(1) 目標と代替案

目標のたて方により、システム計画案の方向は大きく異なるものになる。

(2) 与件条件と代替案

与件条件や資源制約条件の認識の違いにより、代替案の範囲や選択可能性が変化する。

(3) 手段・方式と代替案

特に、自動化、EDP化を考える場合に多くの代替案が考えられる。等について説明し、代替案の開発を効率的に進めるための知識を与える。

6.3 代替案の類型

オプトナーによれば、代替案は基本的に

- ① 現行の方法を続ける
- ② システムを構成する要素を1つずつ変換する
- ③ まったく新しいシステムを設計する

の三つに分類できる。代替案とは何かを、より具体的に把握するために、代替案に関する各種の類型を列挙し、そ

の内容をさらに細分して、代替案の比較検討をいろいろな観点から行い、システム分析における代替案の開発・選択の重要性を認識させることも必要である。

6.4 PPBS

代替案の選択問題は、システムの規模が大きくなると、その困難さが幾何級数的に増大する。PPBSは、このような問題の解決案として非常に整備されたシステムであり、概要を解説しておくといよい。

7. システムの経済性評価

システムの経済性評価は、

① 事前評価として

- ・代替案の選択
- ・システム設計終了時（実施直前）の予測

等、開発の各ステップにおける評価

② 事後評価

に分けられるが、ここでは、①を中心に学習する。

7.1 システムの評価基準

システム（代替案を含めて）を評価するには、まず評価基準が明確でなければならない。評価の基準となるものにはどのようなものがあるかを分類して説明し、理解させる。

(1) 経済的評価尺度

- ① 金額による評価
- ② 数量化できる項目による評価
- ③ 数量化できない項目による評価

(2) 性能評価尺度

- ① 時間的尺度
 - ・照会から応答までの時間の短縮によるサービスの向上
 - ・待ち時間の短縮によって得られる回転率の向上

・処理時間の短縮によって得られる処理効率、処理件数の向上等がある。

- ② 正確性に関する尺度
- ③ 信頼性に関する尺度
- ④ 柔軟性に関する尺度
- ⑤ 安全性、機密性に関する尺度
- ⑥ 余裕に対する尺度

(3) 業務目的別の評価尺度

当該業務の目的に合せ、当該業務を評価に適したレベルまでブレークダウンし、きめ細かい尺度を適用すべきである。

7.2 システムの効果

システム効果の把握の仕方を説明することにより、システムの効果の内容、把握の仕方（計量方法）について理解させる。なお数量化できない効果についても考察を行う。

(1) 目的との関連における効果

- ① 主要効果
- ② 副次効果

(2) 直接効果

- ① 金額的に計測できるもの
- ② 物量で計測し、金額換算ができるもの

(3) 間接効果

(4) 数量化又は金額換算が困難な効果

- ① 迅速な情報処理が可能になった。（迅速性）
- ② 正確な情報が得られるようになった。（正確性）
- ③ システムが安定している。（信頼性）
- ④ 天災、人災から安全である。機密の漏洩がない。（安全性、機密性）
- ⑤ 現在および将来にわたり、多少

の負荷の増大に対しては対処できる。（余裕）

- ⑥ 現在および将来にわたり、多少の要因の変動に対しては対処できる。（柔軟性）

これらの効果に対してもできる限り、計量化の努力が必要なことを、総合評価との関連で説明することが必要である。

7.3 システムの費用

情報システムの費用も、他の経営活動一般の費用管理と異なるところは少ないのであるから、原価計算の一般的な手法を説明し理解させる。

(1) コスト計算の方法

- ① 費目別原価計算
 - ② 部門別原価計算
 - ③ プロジェクト別原価計算
- について概要を説明し理解させる。

(2) 費目別原価計算

- ・直接費
- ・間接費
- ・変動費、固定費区分
- ・管理可能費、管理不可能費区分

(3) 部門別原価計算

- ・部門別原価計算の目的
- ・部門区分
- ・直接部門、間接部門区分

(4) プロジェクト別原価計算

- ・部門別計算を行わない場合
- ・部門別計算を行う場合
- ・間接費のプロジェクト別配賦

7.4 経済性評価

対象となるシステムの効果および費用の見積りが出来たあと、総合評価をどのようにするかについて説明する。

(1) 収益額による方法

(2) 資本回収法

(3) ROI法(Return on Investment)

(4) 回収期間法

等について、その概要と、どのようなケースにどの方法が適しているか、各方法の長所短所を説明し、理解させるとともに実際の通用が正しく出来るような知識を与える。

8. システム仕様書

システム分析のアウトプットであり、次のシステム設計以降のステップに対する指示書になるのがシステム仕様書である。又システム仕様書はその他にも、いろいろの機能をもっている。これらの機能とそのために記載すべき項目について理解させる。

8.1 システム仕様書の目的

システム仕様書は、各階層の関係者に対して異なる役割を持つことを説明する。

(1) システム部門の管理者に対して

プロジェクト開発の総合管理者として、管理下の資源の有効活用、業務遂行の管理のための資料として機能すること。

(2) ユーザ部門に対して

システムに期待した事が、効果的に正しく遂行されることの保証書であり、かつ、新システムへの移行時には、運用要件の記述書として機能すること。

(3) システム設計者に対して

システム・ニーズの達成のために、解かねばならない課題は何かを指示し、又その課題解決のための効果的方法に関する記述書として機能すること。又、入手可能な資源制約と環境、満足規準を明示すること。

(4) システム運用部門、ハードウェア

管理部門の関係者に対して(コンピュータシステムの場合)

現行のハードウェア運用条件を考慮し、新システム移行時の問題を事前に提示すること。例えば、最大負荷時への対応、スケジュール上の特記事項等を明らかにし、必要な対策をたてる上での条件を記述すること。

8.2 システム仕様書の記載事項

上記の目的のために、システム仕様書を構成する文書と、その文書に記述しなければならない内容について説明する。

(1) システム要約書

- ① システムの概要
- ② システムの目的
- ③ システムの解答
- ④ コスト・効果の評価
- ⑤ 開発スケジュール
- ⑥ システムの評価基準

(2) 新システムの機能要件書

- ① システム・フローチャート
- ② 情報の流れ図
- ③ インプットの記述
- ④ アウトプットの記述
- ⑤ ファイル
- ⑥ データ・コミュニケーション

(3) コンバージョン(新システムへの切換)のための機能要件書

- ① コンバージョンの概要
- ② コンバージョン実施計画表
- ③ コンバージョンの内容
- ④ コンバージョンのための教育訓練内容と計画

(4) 経済性分析書

- ① 所要コスト
- ② 期待できる効果
- ③ コスト効果分析

- ④ 検討を行った代替案とそのコスト効果分析
- (5) 開発計画表
 - ① 開発スケジュール
 - ② 開発体制

指導上の留意点

1. コンピュータシステムは、情報システムの自動化部分である、という前提に立って、ここでの説明には意識的にコンピュータに関する記述はさけてきた。しかし、今日の組織体の経営においては、規模の拡大、組織の複雑化、環境条件の変化の激しさ、等々により経営管理の高度化は避けられず、それにともなって、組織体内外の情報要求もより多様になってきている。情報提供の仕方に対して、迅速性や正確性、明示性が要求されてきている。こうした要求に応じて、多量の情報を処理するには、情報処理用機器、とりわけコンピュータを中心とした情報処理用機器の助けなしには事実上、情報システムは成立し得なくなっている。又一方で、情報処理用機器の発達が目ざましいものがあり、特にOAが提唱されてからは、情報処理機器の分野が非常に広がってきている。指導に当って、以上の事を説明するとともに各節での例示をする際にこの点の配慮をする必要がある。
2. システム分析にあたって、常に実体システム→情報システム→EDPシステムの関連を念頭において作業を進めるべきことを強調する。
3. システム分析の要素や各手順の学習には、S1で学習したシステム概念と関連づけて理解させるよう心掛けることが肝要である。
4. システム分析の手法は、アメリカを中心

に展開されたものが多く、必ずしも日本の経営風土になじまない面もある。アメリカ型のトップダウンの意志決定と日本型のボトムアップの意志決定の違いを認識させる必要がある。

5. システム分析者は、

- ① 分析しようとする対象に関する十分な知識
 - ② その対象をシステム化するに要する手段・手法に関する知識
- があってはじめて良いシステムの分析、設計が可能になる。システム分析手法の解説に当っては、分析対象との関係を出来るだけ具体的に示すよう心掛ける。

6. システム仕様書は、システム分析のアウトプットである。ということはシステム分析者の成果を問われるものであることを認識させねばならない。

参 考

1. 帰納法と演繹法

システム分析、システム設計を進めるための方法として、帰納法と演繹法とがある。帰納法は現状の調査分析から出発する設計方法であり、演繹法は現状にとらわれることなく、対象システムの本質的な機能や役割を追求して、合目的な理想システムをえがいてその実現を求める設計方法である。この二つの態度を比較すると、問題発見、問題形成の段階では、帰納法(分析)に重点があり、問題解決の段階では演繹法(設計)に重点があることが多い。

帰納法的設計は、リサーチ・アプローチと呼ばれる経験的方法でもあり、対象が経験的に安定化し、ルーチン化している場合に有効である。典型的な帰納的方法を定式化した代表的なものとして、ケブナー、ト

リゴーの問題分析法 (KT法の一部) がある。

演繹的設計は、デザイン・アプローチと呼ばれる理想システムを想定する方法で、新しいシステム開発技法の中に標準化されて採用されつつある。ナドラーのワークデザインがその代表的な例である。

2. システム分析の特徴〔参考文献(1)参照〕

- ① 従来の方法では、問題形成において、環境や方針や目標の位置づけがあいまいなのに対し、システム分析では問題の前提が明らかで、問題の構造化がしやすい。
- ② システム分析では、原因を、入力、制約条件、プロセスの次元に仕分けして、操作不能な要因のからまりとしてとらえられる。
- ③ 従来の方法では、対策が単なる原因の裏返しで非操作的であるのに対し、システム分析では、対策が具体的、操作的でありかつ目標と制約条件によって評価基準が明確なので、代替案の評価によって合理的な意志決定ができる。
- ④ システム分析では、制約条件が一定の場合だけでなく、変わった場合の対策も立てられるので、戦略的思考が可能である。

参考文献

- (1) 前川良博他共著「経営情報管理」(経営工学シリーズ10) 日本規格協会 1981
- (2) 日本情報処理開発協会 情報処理研修センター編「上級情報処理技術者育成指針」(各論1 組織システムの分析, 各論2 システム開発運用の背景, 各論4 情報システムの開発) コンピュータエージ社 1979
- (3) 石田武雄著「経営システム論」(経営学全書26) 丸善 1970
- (4) 杉山貞夫著「人間・機械比較論」(人間工学) 日本人間工学会 vol. 5 №1 1969. Feb. pp. 10-18
- (5) 近藤他訳「人間工学データブック」コロナ社 1972 (原著 C. T. Morgan, J. S. Cook III, A. Chapanis, M. W. Lund 「Human Engineering Guide to Equipment Design」 McGraw Hill 1963)
- (6) 梅岡・大山編「学習心理学」 誠信書房 1966
- (7) 青井他著「集団, 組織, リーダーシップ」 培風館 1970
- (8) 松井他訳(ハーシィ・ブランチャード著)「管理者のための行動学入門」 日本生産性本部 1971
- (9) 日本経営情報開発協会「システム・ライフサイクルのマネジメント」 1976
- (10) 佐藤允一著「問題の構造学 — 問題発見と解決の技法」 ダイアモンド社 1977
- (11) 吉谷竜一著「システム設計の実際」(日経文庫) 日本経済新聞社 1974
- (12) 齊藤美津子著「きき方の理論」 サイマル出版社 1972
- (13) 飽戸弘著「コミュニケーション」 筑摩書房 1972
- (14) 川喜田二郎著「発想法」, 「続発想法」 中央公論社, 1967, 1970
- (15) オズボーン著(上野一郎訳)「独創力を伸ばせ」 ダイアモンド社 1958
- (16) 梅棹忠夫著「知的生産の技術」 岩波書店 1969
- (17) 中山正和著「発想の論理」 中央公論社 1970
- (18) 植木繁訳「システム分析入門」 日本能率協会 1969 (原著 Optner, S. "System Analysis for Business Management" Prentice-Hall 1968)

- 19 宮川公男編「PPBSの原理と分析」
有斐閣 1969
- 20 前川良博著「情報処理部門運営の管理ポイント」(管理ポイントシリーズ) 税務
経理協会 1979
- 21 前川良博他共著「情報システム設計論」
産学社 1977
- 22 前川良博編著「EDPシステム設計ハン
ドブック」 日刊工業新聞社 1974
- 23 中村信夫著「こうすれば問題は解決でき
る — 新しい問題解決法の論理と手順 —」
開発社 1977

第3章 社会とコンピュータ技術 (S9)

キーワード

知覚, 認知, 学習	perception, cognition, and learning
人間の側面の考察	human factors engineering
集団行動	interpersonal and group behavior
動機, 参画, 伝達	motivation, participation, and communication
リーダーシップ	leadership
組織構造	organizational structure
組織変革	organizational change
プロジェクトチーム	project team
ハードウェア・モジュール	hardware modules
実行・ソフトウェア	execution software
オペレーション・ソフトウェア	operation software
ファイル操作・ソフトウェア	file handling software
データ, プログラムの取扱い	data and program handling
ファイル・システム	file systems
通信システム	communication systems
データ管理	data management
通信装置	communication facilities
制御とプロトコル	control and protocol
回線, 端末の見積	estimating line and terminal requirements
データベース概念	database concept

目 標

・コンピュータを利用した情報システムの開発と維持に必要な外的諸条件の知識を整理する。

この単元の受講者は、すでにS3の学習で、プログラミングの構造と技術の知識を修得しているのので、ここでは情報システムの内容に関する項目は省略する。コンピュータを使う側の立場の人間・組織・社会の性格を分析し、コンピュータの利用形式との適合を考察する。

ここでの学習内容は大きく分類すると2つ

のグループに分けられる。前段はコンピュータを利用する情報処理システムの外側にあって、これを使う立場の人間に関する事項であり、後段は、使われる立場のコンピュータの機能に関する事項である。人間に関する事項は、メンタルな特性、フィジカルな特性、個人としての人間、集団の中の人間、組織の中の人間、社会の中の人間等々のさまざまな視点の分析をおこなう。人間と情報システムのかかわりあい方についても、情報システムを

開発する立場、情報システムに対する要望を提示する立場、情報システムの運用によって影響を受ける立場などさまざまである。これらの全体像を環境条件として一括してここで学習する。道具としてみたコンピュータについても、ハードウェアとソフトウェア、データとプログラム、システム・プログラムとアプリケーション・プログラムなどのように交錯した視点があるが、ここではそれらによって作りだされる全体像の把握を学習する。特に将来の発展方向を推測することに目標をおく。

S3でのコンピュータ技術の学習は、内側からみた情報システムであるが、ここで考察するコンピュータ技術は、外側からみた情報処理の形態である。学習の重点を、データ管理の技術と通信システムのコンピュータとの連結におく。

「情報システムと人間」については、人間とコンピュータの情報処理能力の差異をあきらかにし、人間とコンピュータの協力の最もよい形態を探究する。心理学、生理学、人間工学などの研究成果を広く採りいれる努力が必要である。人間工学は、システム運用の改善に対しての細かい注意事項にヒントを提供する。

「個人と集団」では、社会的行動をする個人の行動様式について学習し、情報システムの開発と運用の場面における集団としての人間の性格を把握する。特に動機づけの問題を中心テーマとする。前単元が「個人としての人間」とコンピュータとのかかわりあいをつかうのに対して、この単元では「集団としての人間」とコンピュータとのかかわりあいをつかう。

「組織」と「組織変革」では、組織論の基礎を学習する。コンピュータを利用する情報処理システムの導入は、何らかの形で、組織

の変革をもたらす。このようなイノベーションの進展過程の一般形を明らかにするとともに、情報システムの導入に関する具体的諸問題を検討する。

以上の三単元は情報システムをとりまく環境条件の学習である。次にコンピュータそれ自身について学習する。コンピュータは「ハードウェア・モジュール」と「ソフトウェア・モジュール」とに項目がわけてあるが、これは便宜的な区分にすぎない。ここで修得すべき事項は、部分にとらわれず、コンピュータの道具としてのイメージを大局的に把握することである。むしろ新しい動向を正確に認識することが大切である。ハードウェア、ソフトウェアの両面について、モジュール化がどのように進展しているかという点などを明確にすればよい。

「ファイル・システム」と「通信システム」は、これからのコンピュータ技術の中核部分である。これらの技術の本筋を習得するには、或程度過去の技術の発展プロセスをたどってみることも必要である。いたずらに新しい技術の細目を暗記するだけでは技術の本質はつかめない。ファイルについては、アクセス方式の変遷、データ管理については、データベース概念の発展の由来、通信システムについては、通信方式の発達やプロトコルなどに重点をおいて学習するのがよい。さらにDB/DCシステムの具体例について、その仕様を調べてみるのが望ましい。

内 容

1. 情報システムと人間

1.1 心 理 学

情報システムの開発や維持に関連して学習すべき事項には、システムそれ自身の論理性の追求もあるが、それ以

上に大切なこととして、情報システムにかかわりあう人間の側の問題も十分考慮しなければならない。人間の側の問題の基礎事項を学習するには、心理学の研究で得られた成果が活用できる。心理学の全体系を習得することは無理な要求であるので、情報処理に関係の深い事項をひろいあげた形で編成して学習することが望ましい。次のような項目をとりあげる。

・知覚・認知・学習・記憶・思考に関すること

大脳生理学とも関連する分野の問題で感覚・知覚・認知のメカニズムや、学習、記憶、思考のプロセスなどを調べることは、情報システムの設計や、情報処理の実施について参考になる点が多い。

・動機づけ・欲求・感情に関すること

人間の行動をひきだし、これを制御する心の働きを研究することは、システムの開発時点での開発に対する動機づけの面でも、システムの運用時点での維持管理の遂行の面でも重要な条件である。

・個性・人格・知能に関すること

対人、対社会の人間行動を支配する心の働きを研究することは、情報システムをロングレンジで考える時に欠かせない視点である。

・心理学の応用に関すること

教育心理学、産業心理学、臨床心理学、心理的測定技術などの応用分野の成果は、そのまま直接的に利用できるものも多い。

1.2 人間工学

情報システムをマン・マシン・システムとしてとらえる考え方は、情報シ

ステムを設計する際に、有効なことが多い。心理学が自然科学的の見方であるのに対比すると、人間工学は工学的の見方に立脚している。次のような項目を学習する。

・人間—機械系の考え方

表示、制御、環境、インタフェースの概念

人間と機械との機能配分

信号検出能力

視覚特性

計器の設計

音響表示

時間感覚

・人間の作業特性

作業環境

心理環境

人間の動作特性

人体計測データ

・知能・学習・記憶

知能テスト

プログラム学習

C A I、テーチング・マシン

記憶、忘却

・コミュニケーション

説得的コミュニケーション

コミュニケーションの過程

言語—シンタックス、セマンティッ

クス、プラグマティックス

人間の情報伝達の諸相

・心理測定

測定法

心理テスト

2. 個人と集団

2.1 行動原理

心理学、人間工学で学んだ基礎的知識をもとにして、人間の対人行動や集

団行動の理解を深める。組織問題を考える際の内的要因である。社会心理学、組織心理学の分野でもある。次のような項目をとりあげる。

- ・ 欲求、動機
- ・ 行動、行動様式
- ・ 社会的相互作用
- ・ 社会的規範
慣習、伝統、因習、法
- ・ 人間関係
- ・ 対人関係
勢力基盤、認知と期待
社会集団、心理集団
ソシオメトリ
- ・ 態度
オビニオン・リーダー、説得
- ・ 集団
所属意識、地位、役割
集団規範
リーダーシップ、分派
グループダイナミックス

2.2 組織管理

前節で学んだ集団と個人の関係の諸問題の知識を基にして、組織管理に関する原理的な事項を整理する。管理論の基本的項目である。次のような点を学習する。

- ・ 管理に必要な動機づけの知識
基本欲求、マズローの欲求の5段階説など
- ・ 動機づけについての学説
マクレガーのX理論、Y理論
アジリスの成熟、未成熟
参画意識、川喜田二郎のKJ法など
- ・ リーダーシップ
グループダイナミックス
マネージャリアル・グリッドなど
- ・ 個人行動、集団行動に関する研究

文化人類学の分野での研究の成果の応用など

- ・ 組織構造の作業効率におよぼす影響
- ・ 情報の集団行動におよぼす影響

3. 組織

3.1 組織分析

情報システムは、経営の方策や組織の改編に関与する面が多い。そのため、情報処理技術者は、組織問題に関する知識を具備していなければならない。組織論の基本的な項目と、実務に関係の深い部分とを学習する。

- ・ 組織の目的、有効性、効率
- ・ 組織モデルとシステム分析
- ・ 現代の組織
- ・ 組織の成長過程
分化、統合、コンフリクト
- ・ 組織の設計
役割、権限、縦割と横割、インフォーマルな組織、文化的コンテキスト
- ・ 組織におけるコンフリクト
了解と合意

3.2 組織変革

イノベーションの導入過程での組織体における諸問題を検討する。特に情報システムの導入に関する事項を、実際問題に役立つような形で具体的に学習する。項目としてあげると次のような諸問題がある。

- ・ イノベーションのタイプと特性
- ・ 人間的側面
変革推進者と受人側
- ・ 計画的変革のプロセス
- ・ プロジェクトチーム
- ・ 実施のための準備
- ・ 導入から定着までの過程
変化への抵抗、コンフリクト解消の

方法

4. ハードウェアとソフトウェア

4.1 ハードウェア・モジュール

情報システムを構成するコンポーネントとしてのコンピュータ・システムの概要を把握する。ここでの学習はハードウェア・モジュールを細分化してその一つ一つの機能や特徴をこまかく検討するよりも、大局的視野から個々のモジュールの全体に対する役割を識別する程度でよい。学習項目の選択には大綱をつかむにはどうしたらよいかという工夫が必要になる。その編成例を次にしめすが、場合によってはこれにこだわらなくてもよい。項目の編成案を示すと、

- ・プロセッサの構成(アーキテクチャ)
- ・マイクロプログラミング
- ・記憶装置の特性と種類
- ・入出力装置
 - 入出力チャネル, 入出力制御
- ・通信制御装置
 - 通信制御, 端末装置
 - ネットワーク・システム
- ・構成方式、制御方式
 - マルチ・プロセッサ
 - 仮想記憶システム
 - インタフェース
 - 割込

4.2 ソフトウェア・モジュール

ソフトウェア・モジュールに関する事項はハードウェアと密接に関連していて、分離して学習することが困難なものも多い。それらについてはすでに前節で学んでいるはずであるので、ソフトウェアとして独立に学習できるモジュールについて調べる。オペレーテ

ィングシステムの機能に関連する実行用ソフトウェア・モジュールである。但しファイル管理と通信に関するものは、学習項目の量が多いので別の単元でとりあげることとする。

- ・マルチプログラミング, マルチプロセッシング
- ・インタプリタのモジュール(リスト処理など)
- ・記憶管理モジュール(実記憶と仮想記憶)
- ・チャネル制御モジュール
- ・リエントラントプログラム
- ・マルチアクセス, マルチプレクス
- ・対話モード
- ・ローダ, モニタ, 診断モジュール
- ・スケジューラ, アカウンティング・モジュール
- ・復元, プロテクション

さらに次のようなプログラム管理、データ管理のためのソフトウェア・モジュールについても概観しておく。

- ・媒体変換, フォーマット変換
- ・ファイル編成
- ・データセット管理の仕様
- ・トランスレータのモジュール
- ・コンパイラのモジュール
- ・ジェネレータのモジュール
- ・エディタ
- ・ディバギングのモジュール
- ・連系編集の仕様
- ・ジョブ制御の仕様
- ・識別と機密保持

5. ファイル・システム

5.1 記憶装置とアクセス

コンピュータを利用する情報システムでは、ファイル管理の機能はその中

枢的な位置を占めている。ファイルの記憶媒体としてのディスク装置の発達につれて、ファイル管理技術は特に精密化が進みつつある。データベース概念の発展もこれに立脚したものである。ここではファイル管理システムの基本になる項目を学習する。

- ・記憶装置の分類

- テープ, ディスク, マス・ストレージ, 階層記憶方式, 記憶保護機構

- ・ファイルの構成

- データ項目, 集団項目, レコード, ファイルなど

- ・ファイルの操作

- ファイルの生成, 索引, ディレクトリ, 逆見出しファイル

- ・記憶装置の構造とアクセス法式

- 順アクセス
索引順アクセス
直接アクセス
乱アクセス

- ・記憶域の割振り, シノニム, バケット (bucket)

- ・記憶容量, アクセス時間, アクセスコスト

- ・V SAM (Virtual Storage Access Method)

5.2 データ管理

データ管理の基本事項を学習する。

ファイル・システムの問題と管理プログラムの問題にわかれる。管理プログラムは従来はオペレーティング・システムの中の一部分の機能であると考えられていたが、汎用データベース管理システムの発達につれて、分離して論ぜられるようになった。データベース管理システムについても注目する。

- ・ファイル・アクティビティ

- ・ファイルのレイアウト

- ・データ・ディクショナリ, ディレクトリ

- ・問合せ言語

- ・DBMSの実例

- ・データベース概念, データ管理者, ユーテリティ

6. 通信システム

6.1 通信システムの構成

通信システムの構成を、ハードウェアのモジュールと処理系のモジュールの両面から基本事項を学習する。端末装置のインテリジェント化にともなって、処理系の区分も変りつつある。基本を明確にするために、通信理論の基礎も学習する。

- ・通信理論

- チャネル, 雑音, 誤り検出, 誤り訂正

- ・伝送モード

- 同期 (シンクロナス), 調歩同期 (アシンクロナス)

- 変調, モデム, インタフェース

- ・通信方式

- 単向 (シンプレックス)

- 半二重 (ハーフ・デュプレックス)

- 全二重 (フル・デュプレックス)

- ・通信装置

- 変復調, 中継, 網制御, 分岐, 端末

6.2 通信システムの形態

通信回線の接続形態, ネットワークの形態, 通信システムの処理形態の概要を学ぶ。

- ・接続形態

- ポイント・ツー・ポイント

- マルチ・ドロップ

- 変換接続

- 集信装置による接続

電電公社の規格、種別

・ ネットワークの形態

集中型

環状型

網状型

分散型

・ 処理形態

リアルタイム処理

リモート・バッチ処理

T S S 処理

・ 伝送制御とプロトコル

データ・リンク

制御手順, コンテンション, ポーリング, 制御符号

切換, ストア・アンド・フォワード誤り制御

・ 回線の見積

端末の選定, 端末の処理能力

回線網の設計, 回線使用時間, 使用率
通信システムの評価, 待ち行列理論
シミュレーション

・ DB/DCシステム

端末管理機能

言語インタフェース

分散型データベース

指導上の留意点

1. 情報システムと人間の問題は、広範囲で且つ多岐にわたっている。心理学、大脳生理学、人間工学などの研究成果の多くが、実務上の問題解決にヒントをあたえてくれることが多い。しかし一方でこれらの学問分野は、体系や定説が確立していないため、諸説があって採択に迷うことも多い。まず必要なことは多くを読み、自分で考えてみることである。アメリカにおける assignment の方式を採用して、入手できる参考

書をかなり大量に投入して、読むことの訓練をおこなうのも学習指導の一方法である。心理学については、知覚と記憶に関する項目に重点をおくのがよい。人間工学に関しては、CRT表示装置についての人間工学的考察などに焦点をあてるのも一方策である。機械の側からみた時、現在の情報処理システムの弱点は入力装置、入力方式である。この意味では入力問題に注目するのもよい。パターン認識、人工知能などの最新の話題に発展する線も考えられる。

2. 個人と集団の単元は、社会心理学の分野の問題である。前単元が自然科学的アプローチであるのにくらべ、ここでは人文科学的アプローチを試みる。人間関係に関する著作も多数の発表がある。前単元と同じように多読の訓練をさせることが望ましい。その場合、ただ漫然と読むのではなく、問題意識をもって深く読むことの必要を強調する。動機づけに関する学説、リーダシップなどをテーマに選ぶとよい。この分野の本は従来はアメリカでの議論の紹介程度のものが多かったが、最近では企業活動の中から発生した日本独自の話題も活発になっている。それらの生きたテーマをとりあげることも重要である。文化人類学の分野の著作などに新鮮な話題が多い。

3. 組織研究はアメリカでの経営学の古い伝統があるが、企業経営の実状の違いから、日本では抽象的な概念論になりがちである。それらの話題は教養程度にとどめて深入りしないほうがよい。むしろ身近な組織の実例について、ケーススタディとして考えてみることを実習させたほうがよい。組織そのものを研究するよりも、現存する組織の中の情報の流れ（コンピュータ処理による

情報だけに限定せずに)を、流れ図に表現することによって、組織の構造や役割をつかむことができる。参考文献の多くは翻訳であるが、これらを入念に読むことによって、彼我の経営風土の違いを読みとることもなり、組織を考えることに間接的に役立つ。また翻訳だけでなく、原書で読むことも必要である。特にコンピュータに関連する話題は、雑誌などの論説に注目すべきものが多いことを指摘するとよい。

4. ハードウェアとソフトウェアに関する学習は教科書風なまとめ方で学習する必要はない。むしろコンピュータ技術がこれからどう発展してゆくかという展望の眼を養うことに重点をおいて学習指導をおこなう。ただしこれはジャーナリスト的なトピックを追うことを意味するのではない。動向をつかむにはむしろ基本的な本質さえ把握していれば容易であるという逆説的な見方もできる。この章の学習の指導には、教える側のアイディアやセンスが大切である。コンピュータのモジュール化の進展の経過と、何故それが必要であるのかを分析するのも一案である。あるいは学習者に自分で考えさせることにして、そのための材料としてハードウェアとソフトウェアの多数のモジュールの実例を提示するだけにとどめるのも一案である。例示としては仮想記憶システム、リエントラントプログラム、エディタなどが適当であろう。
5. ファイル・システムについてはS8に対する予備知識という意味も含めて説明するのが望ましい。技術的内容についてはC5でとりあつかっているので、それとの重複をさけるためには諸機能を概観する方向で取り組んだほうがよい。即ちC5での取扱

いよりも広く浅くみるということである。ファイル・システムを構成するハードウェアについては効率や経済性の比較などをおこなない、その技術動向などにもふれる。ファイルの編成方法の機能比較、選択基準を検討することは、データベース管理システムの学習の基礎になる。DB/DCのシステムの検討では、この分野の技術動向や標準化の動きなどに注目するように指導する。実務的な学習も必要であるが、資料の入手が限定されるので、DBMSの一部の機能についてニーズの側から深く分析してみるといった学習方法をとってもよい。

6. 通信システムについては技術発展が活潑な分野であるので、むしろ将来動向の見通しに重点をおくこととする。細目の技術内容については流動的なところも多いので教材として最新の資料を入手することに努力をはらう必要がある。基礎事項としては、用語の正確な理解が重要である。現実の通信システムの設計には、日本電信電話公社や国際電信電話株式会社のサービス制度、関連法規なども知っておく必要がある。これらの概要を示す。ディジタル交換網、パケット交換網のサービス、端末のインテリジェント化、ネットワーク・システムなどの理解に必要な基礎知識の習得に重点をおいて学習指導をすすめるとよい。全体の姿とこれからの発展動向を把握するには、歴史的な発達過程をふりかえってみるのも現時点では適切なアプローチであろう。教材としてはこの章の全体をカバーするものが少ないので、テキストを編集すること自体が教育内容にもなりうる。

参考文献

- (1) 宇津木, 他「心理学のあゆみ」有斐閣新書, 1977
- (2) 野口, 他「心理学入門」有斐閣新書, 1978
- (3) 南 博(監訳)「図説現代の心理学(全6巻)」講談社, 1976
- (4) 南 博「初歩心理学」光文社(カッパブックス), 1976
- (5) 宮城音弥「精神分析入門」岩波新書, 1969
- (6) 時実利彦「脳の話」岩波新書, 1962
- (7) 時実利彦「人間であること」岩波新書, 1970
- (8) 品川嘉也「脳とコンピュータ」中公新書, 1972
- (9) 田多井吉之介(訳)(フェスター著)「考える・学ぶ・記憶する」講談社(ブルーバックス), 1976
- (10) 麦島文夫「情報の心理学」講談社現代新書, 1971
- (11) 角田忠信「日本人の脳」大修館書店, 1978
- (12) 渡辺慧「認識とボタン」岩波新書, 1978
- (13) 水原(編)「社会心理学」(講座心理学13)東大出版会, 1976
- (14) ジェームズR. スレイグル(南雲・野崎訳)「人工知能」産業図書, 1972
- (15) M. アプター(村井忠一監訳)「人間の行動とコンピュータ」日本能率協会, 1972
- (16) E. A. フェイゲンバウム他(阿部・横山監訳)「コンピュータと思考」好学社, 1969
- (17) De Greene, K.B. (eds.)「System Psychology」McGraw-Hill, 1970
- (18) 林・大川・井口編「人間・機械システムの設計」人間と技術者, 1971
- (19) 渡辺・中井(訳)(マイスター他著)「システム設計とヒューマンファクターズ」丸善, 1971
- (20) 村井・小牧(訳)(ジャペニ著)「企業の行動科学2, 人間と機械」ダイヤモンド社, 1968
- (21) 近藤, 他(訳)(モルガン, 他著)「人間工学ハンドブック」コロナ社, 1972
- (22) 和田・大山・今井(編)「感覚, 知覚心理学ハンドブック」誠信書房, 1969
- (23) 小林孝夫「電子ディスプレイのABC」日本放送出版協会, 1971
- (24) 井戸剛「人間-機械系の話」日本放送出版協会, 1972
- (25) 梅岡・大山(編)「学習心理学」誠信書房, 1966
- (26) 田中靖政「コミュニケーションの科学」日本評論社, 1969
- (27) 小林(訳)「一般言語学講義」岩波書店
- (28) Martin, J.「Design of Man-Computer Dialogue」Prentice-Hall, 1974
- (29) 東洋(編)「講座心理学8, 思考と言語」東大出版会, 1970
- (30) 井上, 福沢, 平栗「一般意味論」河野心理研究所, 1974
- (31) 片桐ユズル「意味論入門」思潮社, 1970
- (32) 西宮満(訳)(ハヤカワ著)「言語と思考」南雲堂, 1972
- (33) 川本茂雄「ことばとところ」岩波新書, 1976
- (34) 川本茂雄「ことばの色彩」岩波新書, 1978
- (35) 入谷敏男「ことばと人間関係」講談社現代新書, 1971
- (36) 入谷敏男「ことばの心理学」中央新書, 1975
- (37) 入谷敏男「ことばの生態」NHKブックス, 1968
- (38) 鈴木孝夫「ことばと文化」岩波新書, 1973
- (39) 鮎戸弘「コミュニケーション」筑摩総合大学30, 1972
- (40) 斎藤美津子「話しことばの科学」「書き方の理論」サイマル出版

- 41) 齊藤美津子(訳)(コンドン著)「ことばの世界」サイマル出版
- 42) 桑名一央(訳)(パート著)「絵と図形による伝達の技術」産業能率短大出版部, 1973
- 43) 小松左京, 高階秀爾「絵の言葉」講談社学術文庫, 1976
- 44) 青井, 他「集団, 組織, リーダシップ」培風館, 1970
- 45) 水原(編)「社会心理学」東大出版会, 1971
- 46) 南, 他(訳)(ベレルソン, スタイナー著)「行動科学事典」誠信書房, 1966
- 47) 三隅, 他(訳)(カートライト, ザンダー著)「グループ・ダイナミックス」誠信書房, 1959
- 48) 豊原, 他(編)「人間関係の心理学」ダイヤモンド社, 1972
- 49) 豊原, 他(編)「組織行動の心理学」ダイヤモンド社, 1972
- 50) 山田「組織科学の話」日経文庫, 1973
- 51) 吉田「心理統計学」丸善, 1976
- 52) 森本三男(訳)(ブライス著)「組織効率」産業能率短大出版部, 1970
- 53) 上田淳生(訳)(クルーランド著)「システム・マネジメント」ダイヤモンド社, 1960
- 54) 小林肇(訳)(ザイラ著)「組織と人間行動」丸善, 1969
- 55) 伊吹山, 中村(訳)(アジリス著)「組織とパーソナリティ」日本能率協会, 1976
- 56) 三隅二不二(訳)(リッカート著)「経営の行動科学」ダイヤモンド社, 1969
- 57) 三隅二不二(訳)(リッカート著)「組織の行動科学」ダイヤモンド社, 1967
- 58) 高橋達男(訳)(マクレガー著)「企業の人間的側面」産業能率短大出版部, 1966
- 59) 松田, 井上(訳)(サイヤート, マーチ著)「企業の行動理論」ダイヤモンド社, 1967
- 60) 土屋守章(訳)(マーチ著)「オーガニゼーションズ」ダイヤモンド社, 1972
- 61) 松田, 高柳, 二村(訳)(サイモン著)「経営行動」ダイヤモンド社, 1965
- 62) 三菱経済研究所(訳)(チャンドラ著)「経営戦略と組織」実業之日本社, 1967
- 63) 吉田博(訳)(ローレンス著)「組織の条件適応理論」産業能率短大, 1977
- 64) 木村尚三郎「組織の時代」潮新書, 1971
- 65) 北川敏男「組織と情報」NHK現代科学講座5, 1966
- 66) 川喜田二郎「組織と人間」NHK現代科学講座6, 1966
- 67) 荒川幾男「管理社会」講談社現代新書, 1976
- 68) 関計夫「人間関係の幾何学」講談社現代新書, 1970
- 69) 加藤秀俊「人間関係」中公新書, 1966
- 70) 加藤秀俊「人間開発」中公新書, 1969
- 71) 加藤秀俊「情報行動」中公新書, 1972
- 72) 川喜田二郎「パーティ学」講談社現代教養文庫, 1964
- 73) 川喜田二郎「チームワーク」カッパブックス
- 74) 川喜田二郎「可能性の探検」講談社現代新書, 1967
- 75) 川喜田二郎「海外協力の哲学」中公新書, 1974
- 76) 幸田一男(訳)(ベニス著)「組織の変革」産業能率短大出版部, 1968
- 77) 高橋, 鈴木(訳)(ベッカート著)「組織づくりと戦略モデル」産業能率短大出版部, 1972
- 78) 上野一郎(訳)(ブレイク, 他著)「動態的組織づくり」産業能率短大出版部, 1969
- 79) 伊吹山太郎(訳)(リビット, 他著)「変革のダイナミックス」ダイヤモンド社, 1970
- 80) 清水勤(訳)(オコネル著)「組織をエンジニアリングする」産業能率短大出版部, 1970

- 81 藤竹暁(訳)(ロジャース著)「技術革新の普及過程」培風館, 1966
- 82 日本能率協会(訳)(ルシュ著)「行動科学—その概念とマネジメントへの適用」日本能率協会, 1970
- 83 松井夫(訳)(シャイン著)「組織心理学」岩波書店, 1966
- 84 穴吹, 他(訳)(シュロッサ著)「マネジメント・システム(上, 下)」産業能率短大出版部, 1971
- 85 横山保(監訳)「システムの理論とマネジメント」日本生産性本部, 1971
- 86 豊原恒男「経営心理学」経営全書37, 丸善, 1971
- 87 斉藤雅彦「情報と経営」新管理者選書6, 日本経営出版会, 1976
- 88 稲葉, 倉井(訳)「意思決定の科学」産業能率短大出版部, 1979
- 89 都築, 他(訳)(ウッドワード著)「技術と組織行動」日本能率協会, 1972
- 90 斉藤, 亀谷(訳)(クリバリー著)「重役室の中の呪術」講談社, 1972
- 91 中根千枝「タテ社会の人間関係」講談社現代新書, 1967
- 92 会田雄次「日本人の意識構造」講談社, 1973
- 93 Berelson, B., and Steiner, G.A. 「Human Behavior」Harcourt Brace and Jovonovich, 1964
- 94 Fogel, L. J. 「Human Information Processing」Prentice-Hall, 1967
- 95 Katz, D., and Kahn, R. L. 「The Social Psychology of Organizations」John Wiley & Sons, 1966
- 96 Khahr, D., and Leavitt, H. J. 「Tasks, organization structure and computer programs」In Myers, 1967
- 97 Miller, G.A. 「The Psychology of Communication」Basic Books, 1967
- 98 Morison, E. E. 「Men, Machines, and Modern Times」MIT Press, 1966
- 99 Myers, C. A. (Ed.) 「The Impact of Computers on Management」MIT Press, 1967
- 100 Orden, A. 「Man-machine computer systems」In Shultz and Whisler, 1960
- 101 Sackman, H. 「Computer, System Science and Evolving Society」John Wiley & Sons, 1967
- 102 Toffler, A. 「Future Shock」Random House, 1970
- 103 Withington, F. C. 「The Real Computer」Addison-Wesley, 1969
- 104 Woodward, 「Industrial Organization」Oxford U. Press, 1965
- 105 江崎玲於奈(監修)(ダニエル・ベル他著)「コンピュータ・社会・経済」
江崎玲於奈(監修)(ジェール・モーゼス他著)「コンピュータ・個人・生活」
江崎玲於奈(監修)(マービン・ミンスキー他著)「コンピュータ・科学・技術」以上, コンピュータ・エージ社, 1980
- 106 石井善昭「計算機の基本方式」共立出版, 1973
- 107 日経エレクトロニクス編「コンピュータ」日経マグローヒル社, 1978-9
- 108 藤野喜一, 他「計算機システム基礎論」共立出版, 1973
- 109 朴木実「データベース・導入と設計」企画センタ, 1975
- 110 藤井純, 鈴木伸夫「オペレーティング・システム」産業図書, 1970
- 111 情報処理「ジョブ制御特集」Vol. 15, 1974
- 112 斉藤信男「OSの基礎理論」情報処理, Vol. 15, 1974
- 113 北野安定(訳)(マルチン著)「リアルタイム・システムの設計, 管理, 運用」日本経営

- 出版会, 1975
- (114) 大野豊「オンライン・システムのソフトウェア」産業図書, 1977
- (115) 斉藤信男(訳)「オペレーティング・システムの基礎」日本コンピュータ協会, 1976
- (116) 鳥居, 杉藤, 直野(訳)(ブラウン著)「マイクロ・プロセッサとソフトウェアの移植性」近代科学社, 1977
- (117) 喜安善市(編)「入出力装置」共立出版, 1967
- (118) 森宗正, 他「ハードウェア入門」共立出版, 1971
- (119) 北川節(訳)(ナシュルスキー著)「電子計算機の基礎」培風館, 1968
- (120) 浦昭二「データ構造」共立出版, 1974
- (121) 上條史彦「データベース・システム」産業図書
- (122) 江村潤朗(訳)(マーチン著)「データベース」日本コンピュータ協会, 1977
- (123) 西村恕彦(訳)(ケーガン著)「データベース入門」近代科学社, 1975
- (124) 西村恕彦, 他(訳)「データベース・システム」bit臨時増刊, 共立出版, 1973
- (125) データベース言語研究会(訳)「データベース用データ記述言語」情報処理学会, 1977
- (126) 西村恕彦(監訳)「データベースの選び方」bit臨時増刊, 共立出版, 1977
- (127) Date, G.J. 「An Introduction to Data Base Systems」Addison-Wesley, 1975
- (128) Lefkovitz「Data Management for On-line Systems」Prentice-Hall, 1976
- (129) 「Guide to Data Base Management」Aulrbach, 1975
- (130) 山本岩(監)「データ通信」産報, 1971
- (131) 電気通信協会(編)「データ通信の初歩」オーム社 1971
- (132) 横井満「データ通信」日刊工業新聞社 1971
- (133) 「データ通信便覧」データ通信協会
- (134) Karp(ed.) 「Basics of Data Communication」McGraw-Hill, 1976
- (135) 広田憲一郎「データ伝送システム」産業図書, 1971
- (136) 斉藤正治「TSSの計画と運用」TBS出版会, 1977
- (137) 猪瀬博(訳)(アブラムソン著)「コンピュータ・ネットワーク」オーム社, 1975
- (138) 猪方研二(編)「オンライン・システム入門」企画センター, 1968
- (139) 北原安定(訳)(マーチン著)「リアルタイム」日本経営出版会
- (140) Buckley 「Communications and Data Management」John Wiley & Sons, 1976
- (141) Lyon 「The Data Base Administrator」John-Wiley & Sons, 1976
- (142) Meadow, C.T. 「The Analysis of Information Systems」John Wiley & Sons, 1967
- (143) Minker, J., and Sable, J. 「File organization and data management」In Cuadra, 1967
- (144) Salton, G.J. 「Automatic Information Organization and Retrieval」McGraw-Hill, 1968
- (145) Senko, M.E. 「File Organization and Management Information Systems」In Cuadra, 1969
- (146) Parkhill, D. 「The Challenge of the Computer Utility」Addison-Wesley, 1966

第4章 情報処理技術概説 (C1)

キーワード

バッチ処理システム	batch processing system
オンライン・システム	online system
システム構成	system configuration
一括処理	batch processing
多重プログラミング	multi programming
多重プロセッシング	multi processing
スループット (処理能力比)	throughput
応答時間	response time
互換性	compatibility
トップダウン・プログラミング	top-down programming
段階的な細分化	refinement
アルゴリズム	algorithm
ストリング	string
リスト構造	list structure
木構造	tree structure
配列	array
表	table
情報構造	information structure

目 標

1. コンピュータのハードウェアの知識を整理し、オンライン・システムを含むシステム構成要素と各要素の機能を理解させる。
2. 一括処理、多重プログラミング、多重プロセッシングの処理方式の概要を理解させ、処理方式を中心にハードウェアとソフトウェアの機能分担の仕組を理解させる。
3. システムを設計したり評価するときに重要な信頼性、性能評価、互換性、標準化の概念と方法を理解させる。
4. アセンブリ言語、COBOL、FORTRAN、PL/I、ALGOL、BASIC、PASCALなどのプログラミング言語の特性を理解させる。
5. プログラミングの日程計画と手順を理解させる。
6. 構造化されたプログラミングを例にして、プログラムを「わかりやすく」書くことの重要性を理解させる。
7. アルゴリズム設計の知識を整理し、アルゴリズム設計と共に機能設計が重要であることを理解させる。
8. 情報構造に関連して使用される用語を整理し、ストリング、リスト、木、配列、表

などの情報構造のモデルを理解させる。情報の分類と探索の仕組と代表的な手法を説明する。

これは、第2種情報処理技術者及びそれに相当する技術力を有する人が、第1種（シニアプログラマ）又は特種（システムズ・エンジニア）の技術レベルをめざそうとするときの必修のコースである。第2種相当の知識を整理、復習し、次の「コンピュータ・システム」への導入に必要な概括的な知識を習得させる。

内 容

1. コンピュータのハードウェア

(1) システム構成

バッチ処理システムとオンライン・システムの典型的なシステム構成を取りあげて、その構成要素を簡単に説明する。

システムを構成する装置として、センタ装置、オフライン装置、端末装置と通信回線、付帯設備と付帯装置を取りあげて、機能と構成を説明する。センタ装置では、中央処理装置、記憶装置、入出力装置を分けて説明する。オフライン装置では、データ作成装置とデータ変換装置を説明する。

(2) 処理方式

一括処理、多重プログラミング、多重プロセッシングの処理方式を簡単に説明する。ハードウェアの機能を中心に説明し、ソフトウェアのサポートが必要なることを理解させる。ハードウェアとソフトウェアの機能分担は、次第に変化してきていることを説明する。

(3) 信頼性、性能評価

コンピュータ・システムの信頼性を高めるための代表的な技術を簡単に説明す

る。信頼度、MTBF、MTTR、アベイラビリティなどの信頼性の尺度、情報の高信頼化の方法、障害の検出方法を説明する。

システム性能を評価する尺度として、スループット、応答時間、システム資源の使用率などの概念を説明する。この評価の尺度には、スループットが大きくなれば応答時間は長くなり、資源使用率は高くなるという関係があることを理解させる。

(4) 互換性と標準化

互換性の概念を説明し、ハードウェア、プログラム、データ/ファイルについて、それぞれの互換性の特性を説明する。互換性と標準化を高めることの重要性を理解させる。標準化についての国際動向、JIS規格を説明する。職場における標準化の取組み方を理解させる。

2. コンピュータのプログラミング

(1) プログラミング言語とプログラミング

アセンブリ言語、COBOL、FORTRAN、PL/Iの特性を、文法、性能、適用分野、覚え易さ、読み易さ、書き易さなどの面から比較検討しながら説明する。その他の言語としてALGOL、BASIC、PASCALなどについて簡単に説明する。

プログラミングの手順とプログラム作成の日程計画を説明する。プログラム設計、コーディングとデバッグ、個別テスト、全体テストの作業内容について、従来とられて来た方法、問題点、改善の努力などを理解させる。

(2) 構造化されたプログラミング

よいプログラムのもっとも重要な要素は、「わかりやすさ」であることを理解させ、プログラムをわかりやすく作るた

めの技法を説明する。トップダウン・プログラミング、機能中心の思考、処理手順とデータ構造の段階的な細分化について説明する。プログラムは、連続、選択、繰返しという3つの制御の型だけで組み立てることができることを理解させる。

(3) アルゴリズム設計

アルゴリズムの意味を説明し、事務計算の分野でもっとも基本的な次の4つのアルゴリズムの標準的な形を十分に理解させる。①明細印刷②合計印刷③ファイルの順次更新④ファイルのランダム更新。機能設計とアルゴリズム設計の関係を説明する。

3. 情報構造

(1) 情報の基本概念

コンピュータの内部と外部の情報表現の違いを説明する。ビット、バイト、フィールド、項目、レコード、ファイル、データベースの概念を復習する。数字（整数、実数）と文字、定数と変数の扱い方を説明する。コンピュータの算術演算、データの識別について復習する。

(2) 情報構造のモデル化

ストリング、リスト構造、木構造、配列、表の概念、適用例、扱い方を説明する。

(3) 情報の分類と探索

大量データ処理における順次処理の利点を説明し、分類の重要性を理解させる。分類に関連して、ファイル、レコード、キー、照合順序、分類と併合、昇順と降順、系列、パス、内部分類と外部分類などの用語を説明する。内部分類法と外部分類法の代表的な方法を2、3取りあげて説明する。

探索技法について、逐次探索法、2分探索法、2進木探索法を説明する。

分類と探索は補いあう関係にあるので、どちらに比重を置くべきかについて、検討の仕方を説明する。

(4) 情報構造の利用例

データベースで用いられる基本的な情報構造を説明する。

指導上の留意点

1. このコースでは、第2種情報処理技術者又はそれに相当する技術力を持った人の知識を整理し、そのあとのコースを学習するための導入を行う。
2. コンピュータにハードウェアとプログラミングと情報構造という3つの面からアプローチし、バランスのとれた導入を行うことが望ましい。
3. いずれも新しい知識を導入する部分については、概要を紹介する程度にとどめ、あまり詳しく内容に立ち入らないようにする。詳しい内容は、このあとのコースで学ぶことになるので、おおまかなことがわかればよいのである。
4. システム構成については、できるだけ具体的な例を示して説明する。
5. 多重プログラミングと多重プロセッシングについては、その概念を理解させると同時に、どんなハードウェア環境のもとで、どんな仕組で実行されるか、どんな機能が必要かをおおまかに理解させる。
6. 信頼性については、計算方法を説明し、簡単な計算ができるように指導する。また、信頼性が高く、障害に対する備えを持ったシステム構成について、例をあげて説明する。
7. 標準化では、情報処理のJIS規格について正しい理解を持たせることが重要である。JISハンドブック情報処理編（日本

規格協会)を示す必要がある。

8. コンピュータのプログラミングについては、文法的な説明に深く立ち入る必要はない。言語の特徴を適用分野と対比させながら説明する。
9. 従来のプログラミングでは、「正確さ」が最も重要であるとされてきた。しかし、最近では、プログラムの検査や保守を容易にするために、プログラムを「わかり易く、読み易く」作ることを重視するようになった。このことは重要なので、具体的なコーディングの例を示して、「わかり易さ」の意味を理解させ、その重要性を納得させる必要がある。
10. アルゴリズム設計は、プログラミングの作業の中で依然として大きな比重を占めているが、重要なのは機能設計とアルゴリズム設計の関係である。機能設計の作業が一層重要になってきていることを理解させる。
11. 情報構造について、このあといろいろなコースで詳しく説明するので、ここでは、一般的な用語の意味と基本概念を理解させることができればよい。
12. このコースは、特種又は第1種情報処理技術者をめざす人が最初に学ぶコースであり、第2種の情報処理技術、又はそれに相当する技術力を持っていることが唯一の前提条件である。
13. このコースを学んだ後は、「コンピュータ・システム」又は「プログラミング」のコースに進むことを想定している。

参 考

1. 用語の説明

バッチ処理

あらかじめ蓄えられた、データの処理又はジョブの実施で、その処理の進行中は、利

用者が、その処理に影響を及ぼすことができないような手段によるもの。

オンライン

計算機の直接の制御下で機能単位の操作が行なわれることについての用語

システム(データ処理の)

指定された一連のデータ処理機能を遂行するために組織された、機器、方法、手順、及びおそらくは人も含めた集まり

一括処理

バッチ処理と同じ

多重プログラミング

一つの処理装置によって、二つ以上の計算機プログラムの交互配置による実行の機能を備えた操作様式

多重プロセッシング

多重プロセッサの二つ以上の処理装置による並列処理の機能を備えた操作様式

スループット

処理能力比ともいう。与えられた時間内に計算機システムによって遂行される仕事の量の測定

応答時間

計算機システムに対する問合せ又は要求の終りから応答の始まりまでの経過時間

互換性

媒体、装置、プログラム、データなどが、機器や組織の環境を変えても、支障なく、正常な機能や性能を発揮できる性質

トップダウン・プログラミング

プログラム全体の機能を最初に定義し、それを段階的に細分化しながら次第に詳細な機能を定義するという方式でプログラム作成を行なう方法

段階的な細分化

最初に全体をおおまかに定義し、細分化を繰り返しながら次第に詳細に明確に定義していくこと。その様子は木構造で表現でき

る。木構造の下位機能を合わせると上位機能と等価になる。

アルゴリズム

算法ともいう。有限回で問題を解くための、明確に定義された有限個の規則の集まり

ストリング

列ともいう。文字や物理的要素の1次元的並び

リスト

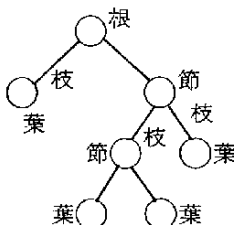
系列、並びともいう。項目の順序づけられた集合

木(構造)

データやプログラムを次の図のように展開

したものを木構造という。

これを木にみたてて、線のことを枝、枝の集まっているところを節、枝の入ってない節を根、枝の出ない節を葉という。



配列

一つ以上の次元に従って要素を並べたもの。

表

データの配列で、その各項目が一つ以上の引き数によって、あいまいさなく識別できるもの。

情報

データを表現するために用いた約束に基づいて、人がデータに割り当てた意味

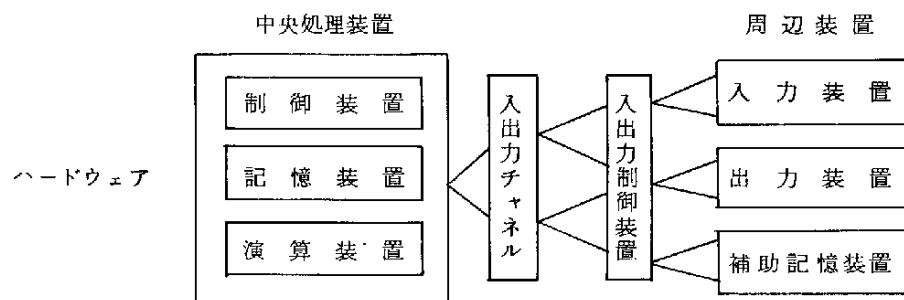
2. コンピュータ・システムの構成

コンピュータ・システムの構成要素を、ハードウェアとソフトウェアにわけて考えることができる。

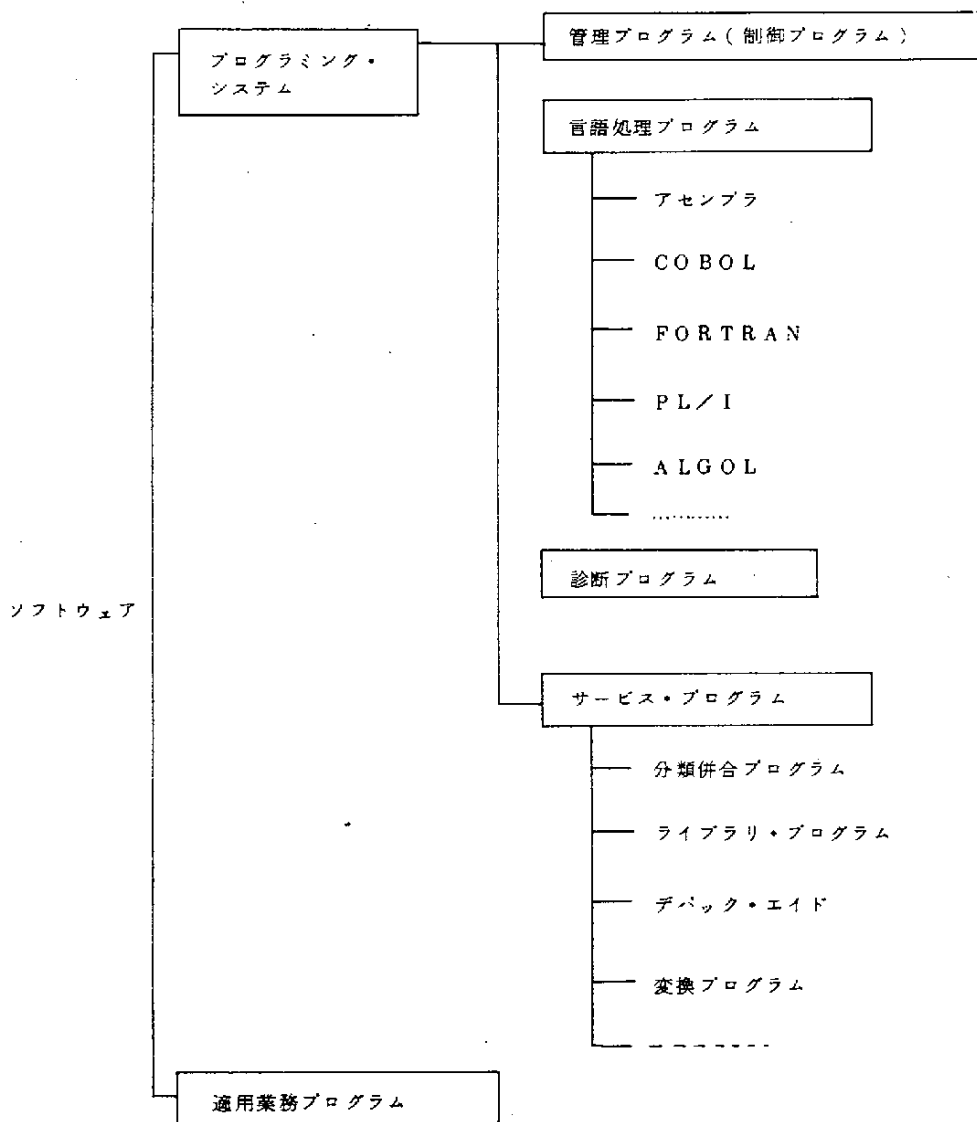
コンピュータ・システム { ハードウェア
ソフトウェア

(1) バッチ処理システム

ハードウェアの一般的な構成は、次のようになっている。



ソフトウェアの一般的な構成は、次のようになっている。



(a) 管理プログラム

管理プログラムには、ジョブ管理、データ管理、タスク管理などの機能がある。

プログラミング・システムは、プログラミング言語と言語処理プログラム

から成る。

プログラミング言語の中心になるのは、アセンブリ言語とコンパイラ言語であるが、その他にもさまざまな言語がある。アセンブリ言語に対する言語処理プログラムをアセンブラといい、

コンパイラ言語に対する言語処理プログラムをコンパイラという。

COBOL, FORTRAN, PL/I, ALGOLなどはコンパイラ言語である。

プログラミング言語を手続き向き言語と問題向き言語に分けることもできる。

(b) 診断プログラム

これは主として保守用に用いられるハードウェア診断プログラムである。記憶装置テスト・プログラム、周辺装置テスト・プログラムなどがある。

(c) サービス・プログラム

これはユーティリティ・プログラムとも呼ばれ、分類併合プログラムの他、磁気テープや磁気ディスクのハンドリング・プログラム、プログラム・テスト用のデバッグ・プログラム、プログ

ラム・ライブラリの支援プログラム、各種の変換プログラムなどがある。

(d) 適用業務プログラム

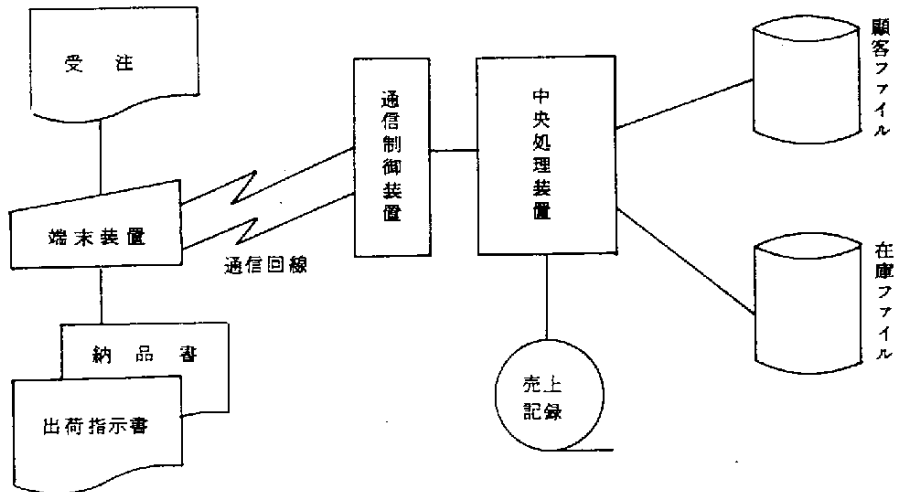
適用業務プログラムの代表的なものは、ユーザ・プログラムである。その他に、

- 各種の科学技術計算
- 線型計画
- シミュレーション
- 予測
- プロジェクト管理
- 在庫管理
- 情報検索

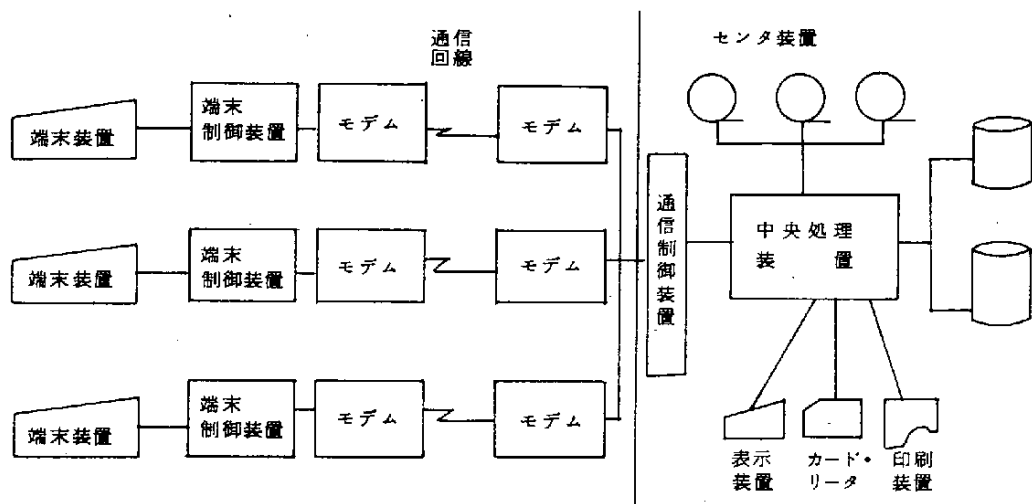
などの分野について、いろいろな汎用の適用業務プログラムが用意されている。

(2) オンライン・システム

(a) データ・エントリ・システムの例



(b) データ通信システムのハードウェア例

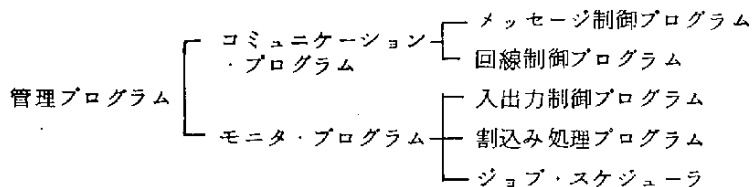


(c) データ通信システムのソフトウェア

次のような3つのプログラムに分けて考えることができる。

- 管理プログラム
- 処理プログラム
- 支援プログラム

管理プログラムの例



処理プログラム

ユーザが準備するオンライン・システムの各種の適用業務プログラムである。

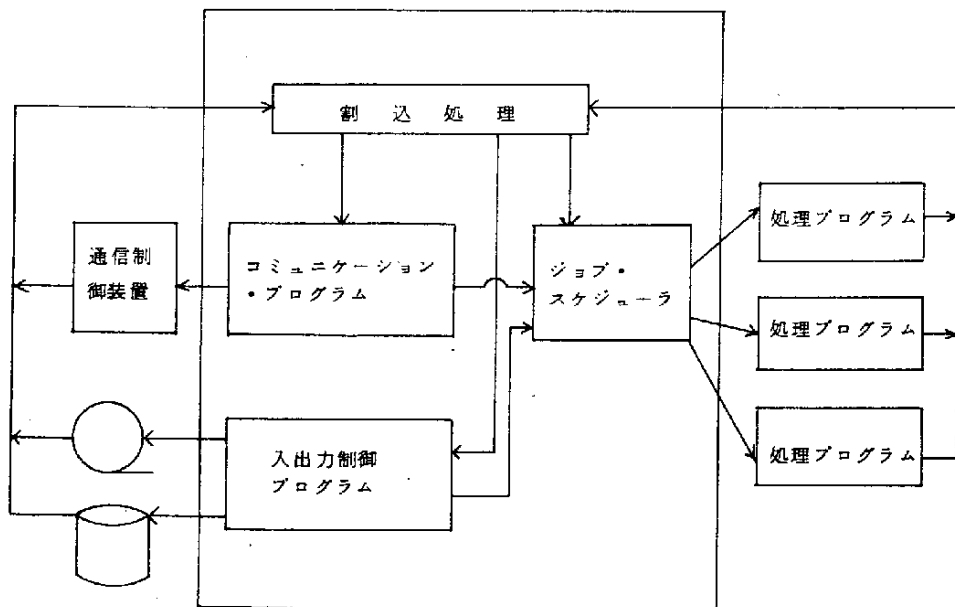
支援プログラム

システム診断プログラム、テスト・データ・ジェネレータ、デバッグ用プログラム、障害回復プログラムなどがある。

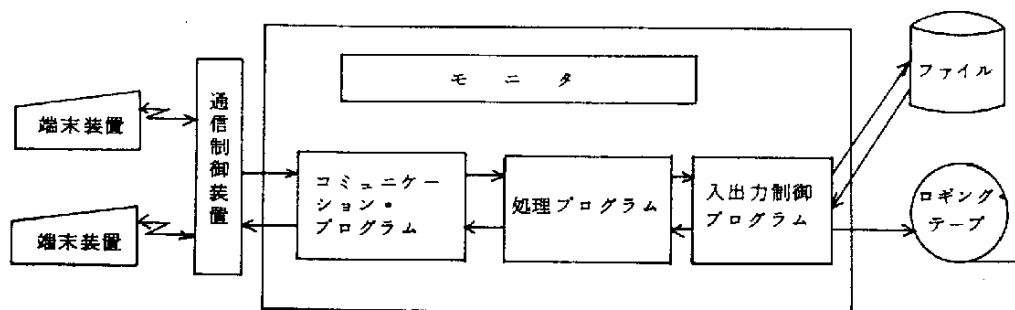
それぞれのプログラムの関係は次のようになる。

制御の流れ

管理プログラム



データの流れ



3. 多重プログラミング

(1) 多重プログラミングの目的

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並行処理することを多重プログラミングという。

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り、例えば、中央処理装置の内部演算処理だけを実行するようなプログラムでは、多重プログラミングの効果は期待できない。

多重プログラミングにより期待できる効果は、例えば次のようなものである。

システムの資源（中央処理装置、主記憶、入出力チャネル、周辺装置など）をすべて管理プログラムによって集中管理し、全資源を有効に動作させることによって装置の遊びを少なくし、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できる。

例えば、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、カードリーダー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、2個のプログラムで同一の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て処理を進め、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1本ず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できる。

(2) 多重プログラミングに必要なハードウ

ェア

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、またそれを制御する管理プログラムが必要である。

多重プログラミングで必要になるハードウェアには次のようなものがある。

(a) 割込み機能

初期の段階では、主に周辺装置に関連した割込みが考えられ、それは必ずしも多重プログラミングと関連したものではなかった。

例えば1個のプログラムでカードリーダーより情報を読み取りその結果磁気テープ上のマスタファイルを更新し、更新情報をラインプリンタで印刷するという処理で、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、非常に非効率で、逆に周辺装置の方から動作終了時点で中央処理装置に対して特別な割込み信号を発生させ、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、割込みを発生した周辺装置の処理に移るという方式が考えられた。

この割込みは、単に周辺装置だけでなく、種々の分野に拡張されてきており、特定の条件をハードウェアで検出し、割込み信号を発生させることによって、中央処理装置がその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一般に割込みと呼ぶようになった。

割込みの種類

割込みにはいくつかの種類があり、コンピュータによって特徴がある。代表的なものには、次のようなものがある。

- 周辺装置など外部要求によるもの
周辺装置の動作終了
操作卓の割込みボタンなど

- プログラムによるもの
スーパーバイザ・コール (SVC)
命令
不正命令
オーバフローなど

- 時刻装置によるもの
- ハードウェア障害によるもの
主記憶のパリティ・エラー
電源異常など

割込み制御

割込みが発生し、それが中央処理装置に認識され処理される過程、即ち割込み制御は次のような手順で行われる。

- 割込み発生
- 割込みの認識
- 管理プログラム・モードへの切換え
- レジスタや標識の状態退避
- 割込み原因の判別
- 処理プログラムの選別、準備
- 処理プログラムの実行

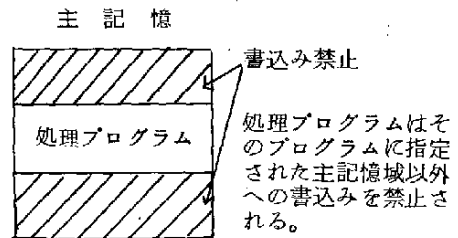
(b) 記憶保護機能

記憶保護機能は、多重プログラミングの概念とともに生まれたもので、多重プログラミングと密接な関連がある。

多重プログラミングでは、多数のプログラムを並行処理するが、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を処理プログラムも含まれている。

また、中にはデバッグ中の未完成なプログラムも含まれる。多重プログラム実行中に、例えばこのデバッ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としたら、その処理結果は

どうなるかわからない。このような危険性があるとしたらシステムの信頼性は失われてしまう。従って、多重プログラミングのもとでは、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のである。このための機能として記憶保護機能がある。



記憶保護の方式

記憶保護の方式には、モード制御とアクセス制御がある。これが組み合わされて記憶保護が実施される。

◦ モード制御

割込みモードと非割込みモードによりモード制御を行う。

割込みモードでは、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、非割込みモードでは、一般の処理プログラムが動作し、このモードでは、一部の命令(特権命令という)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。

◦ アクセス制御

アクセス制御としては、主記憶への書き込み制御、読出し制御、命令実行制御がある。

書き込み制御では、指定された主記憶域以外への書き込みが禁止される。

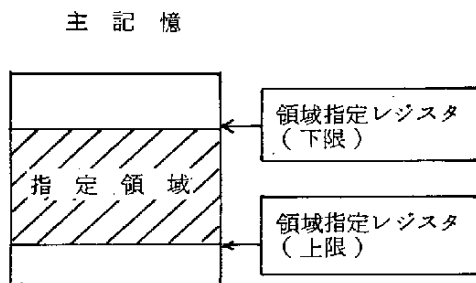
読出し制御では、指定された主記憶域外からの読出しが禁止される。

命令実行制御では、指定された主記憶域外での命令の実行が禁止される。

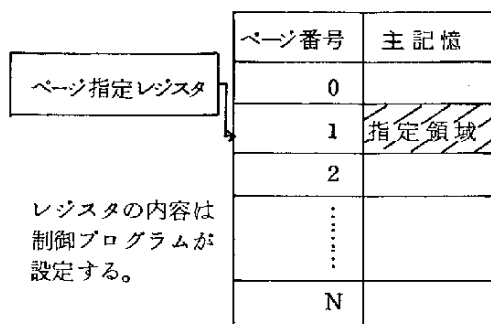
主記憶の領域指定

主記憶の領域指定には、レジスタ方式、ページ方式、セグメント方式がある。

レジスタ方式は、次の図のとおりである。



ページ方式は、次の図のとおりである。



セグメント方式は、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、書込み禁止、読出し禁止、命令実行禁止などを含む。

(c) プログラム再配置機能

多重プログラミングを実行する場合には、さらにプログラムの再配置機能が必要である。

プログラム再配置の必要性は、次の

ような点にある。

- 主記憶には複数個のプログラムがロードされ、並行処理される。
- 処理を終了したプログラムのあった主記憶域は、次の処理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まで予測できない。
- 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あることがあり、このとき中断されたプログラムは、一時補助記憶装置などに退避され、後に再び主記憶にロードされ、処理が継続される。このときロードされる主記憶域は、中断の前と異なる場合がある。

以上のような状況を考えて、多重プログラミングのもとでは、プログラムは主記憶のどの領域にロードされても実行できるような仕組みになっなければならない。この機能を再配置（リロケーション）機能と呼んでいる。

リロケーションには、静的リロケーションと動的リロケーションがある。

また、プログラム配置の方式には、レジスタ方式とマッピング方式がある。

4. 多重プロセッシング

複数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・システムと呼び、多重プロセッサ・システムによる処理を多重プロセッシングと呼んでいる。

多重プロセッシングには、次のような目的がある。

- 信頼性の向上

複数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、処理を続行できる。

○処理速度の向上

複数の中央処理装置で処理を分担することにより、高速化がはかれる。また、中央処理装置をそれぞれ分担業務によって特殊化すれば、システムの価格性能比を改善できる。

(1) 処理方式

多重プロセッシングの方式は、中央処理装置の処理が一時的なものか特殊化されているか、主記憶を共有するか否かのそれぞれ二つの方式があり、これが組み合わせられて、いろいろな方式が考えられる。

そのうちよく知られた代表的なものは、次のとおりである。

(a) デュアル・システム

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・システムである。主に信頼性の向上が目的であり、主記憶は各々独立に持つ。

(b) デュプレックス・システム

通常は2台の中央処理装置が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引き受け続行するのがデュプレックス・システムである。

デュプレックス・システムは、信頼性はデュアル・システムより低いが、処理能力の方に重点を置いたシステムである。

(c) 主記憶共有方式

複数の中央処理装置が主記憶と周辺装置を共有するシステムである。

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、周辺装置の処理分野が動的に割り当てられるもの、各中央処理装置ごと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もの、各中央処理装置の処理分野が演算処理用、周辺装置処理用などと定められているものなどがあ

(2) ハードウェア機能

多重プロセッシングを行なうためには、中央処理装置、周辺装置は通常の機能の他に、種々の特別な機能を備えている必要がある。その代表的なものには、次のようなものがある。

(a) 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、任意に接続したり、切り離したりできることが必要である。

(b) 情報交信

中央処理装置間で互いの状態、処理状態、処理分担などについて交信を行える機能が必要である。

5. システムの信頼性

コンピュータ・システムの信頼性を高めるために用いられる技術としては、次のようなものが考えられる。

(a) 素子の高信頼化

○高信頼性素子の開発

故障の物理的究明と改善

大規模集積技術による信頼性向上

○環境条件の改善

温度、湿度、振動、衝撃などの物理的条件の改善

許容限度以下の条件のもとで使う

(b) システムとしての高信頼化

◦ システムとしての信頼性向上

冗長な構成

修理期間の短縮

システム回復技術の開発

ソフトウェアの信頼性向上

◦ 情報の信頼性向上

情報への冗長性の付加

情報の保護

信頼性を表わす尺度には、次のようなものがある。

◦ 信頼度 (reliability)

◦ MTBF (mean time between failure)

◦ MTTR (mean time to repair)

◦ アベイラビリティ (availability)

MTBF, MTTR, アベイラビリティ

の関係は、次のようになる。

$$A_i = \frac{D_i}{U_i + D_i} \quad U_i = \text{MTBF} \\ D_i = \text{MTTR}$$

(c) 信頼性を考慮したシステム構成

信頼性を考慮したシステムとしては、

次のようなものがある。

フォールバック・システム

バックアップ・システム

デュプレックス・システム

デュアル・システム

多重プロセッサ・システム

(d) 情報の信頼性

システムの信頼性を高めるには、ハードウェアの信頼性だけでなく、ソフトウェア的な問題もある。主なものは次のとおりである。

◦ 入力データのチェック

検査数字チェック

範囲チェック

妥当性チェック

集計チェック

書式チェック

通番検査

ハッシュ合計チェック

◦ メッセージの紛失防止

通番付加

メッセージのロギング

◦ ファイルの障害対策

ファイルの二重化

仮ファイルの利用

更新を仮ファイルに対して行い、

処理単位が終了してから実ファイル

の更新を行う。

チェックポイント方式

インクリメンタル・ダンプ方式

トータル・ダンプ方式

◦ 情報の保護

(e) 障害の検出と訂正

◦ データの誤り検出と訂正

パリティ・チェック

ハミング・コード

巡回符号 (CRC)

バースト誤りの検出ができる

二重照合チェック

◦ 演算、制御の誤り検出

多重化法

nCmコード法

パリティ・チェック

タイムアウト・チェック

◦ ソフトウェア・エラーの検出

不正命令コード

記憶保護侵害

オーバフロー

アンダフロー

(f) 障害の救済方法

障害が検出された場合、ハードウェアで自動的に訂正が行われることもあるが、ハードウェアによる自動訂正ができない

ときは、ソフトウェアによって障害処理を行う。次のようなものがある。

- 再試行
- 障害の区分け
- 障害部分の切離し、再構成
- 再構成ができなければ停止する

システムを再構成して再開する場合は、次のような手順がとられる。

- メッセージの回復

ロギングにもとづいてメッセージの状態を再現する。再入力を要求することもできる。

- 主記憶上の各種テーブルの回復

最新のチェックポイント・データにもとづいて回復する。

- ファイルの回復

トータル・ダンプとインクリメンタル・ダンプによって回復するが時間がかかる。

- バッチ・システムの回復

2つの方法がある。

ジョブ・ステップを再開する。

チェックポイントをとった時点から再開する。

6. 情報構造

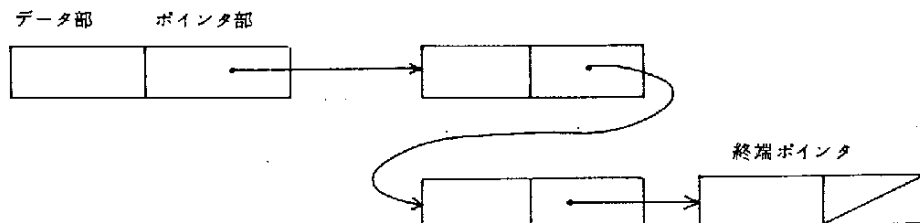
情報構造には、論理構造と物理構造がある。論理構造は、データが本来持っている構造で、当面の問題解決に必要があると認められたものを外部構造ともいう。

物理構造は、コンピュータ内でデータの構造を表現するために作られた構造で内部構造ともいう。

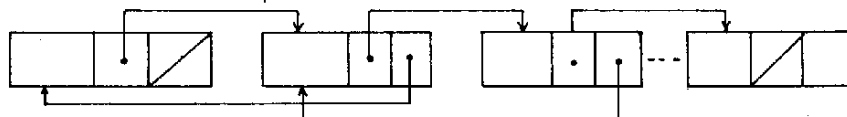
論理構造と物理構造は、1対1に対応するわけではない。例えば論理構造として木構造が考えられる場合、それを表現するための物理構造には、配列を用いているかも知れない。

(1) 線型リスト

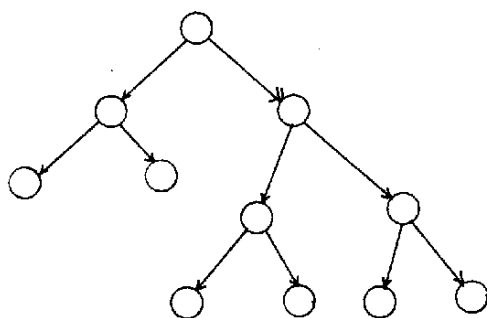
一方向のリスト



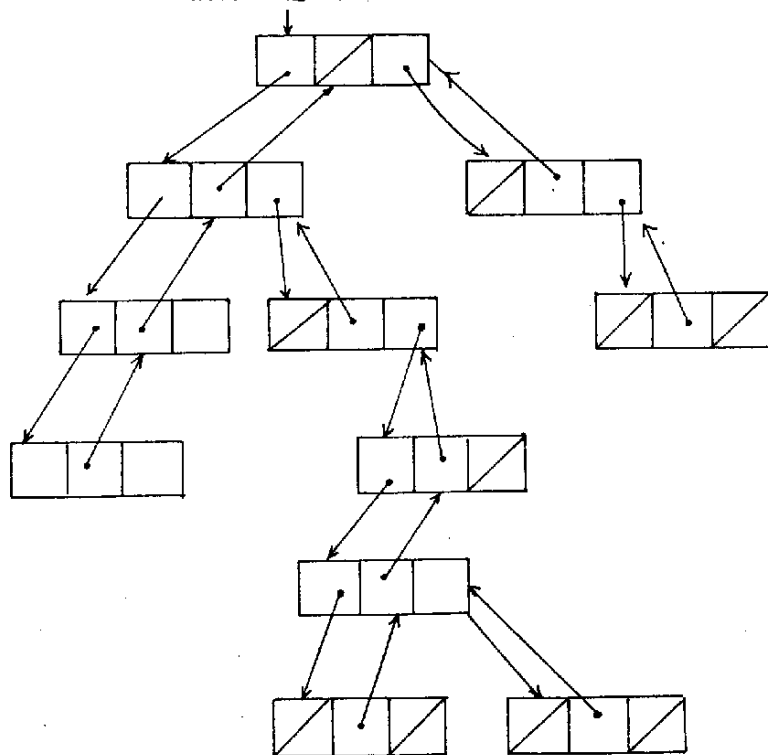
両方向リスト

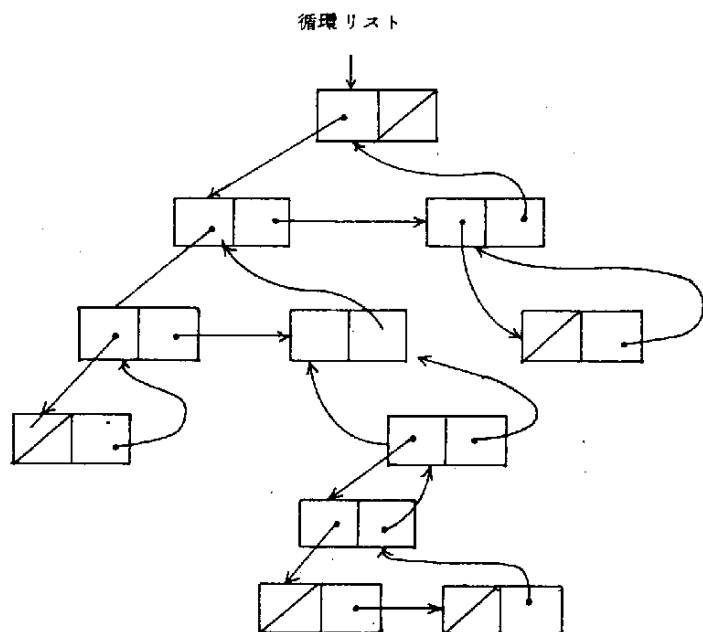


木構造



両方向の2進の木





7. J I S規格

正式には日本工業規格といい、工業標準化法に基づいて制定される国家規格である。J I Sは、適正な内容を維持するために、5年ごとに見直しが行われることになっている。

J I S規格の中の情報処理に関連する部分は、日本規格協会からJ I Sハンドブック情報処理として、毎年出版されている。

1981年4月現在の、情報処理について制定されているJ I S規格は次のとおりである。

〔一般〕

- (1) 情報処理用語
- (2) 情報処理用流れ図記号
- (3) 2値論理素子記号
- (4) 電子計算組織構成機器の性能表示

〔キャラクタセットとコーディング〕

- (1) 情報交換用符号

- (2) 情報交換用および数値制御機械用符号の紙テープ上での表現

- (3) 情報交換用符号の磁気カセットテープ上での表現

- (4) 情報交換用符号の紙カード上での表現

- (5) 情報交換用符号の磁気力セットテープ上での表現

- (6) 情報交換用漢字符号系のための制御文字符号

- (7) 情報交換用漢字符号系

- (8) 情報交換用機能キャラクタの図形表現

- (9) 情報交換用符号の拡張法

- (10) 光学式文字認識のための情報交換用符号

- (11) 情報交換用磁気テープのラベルとファイル構成

〔文字認識〕

- (1) 光学式文字認識のための字形(英数字)
- (2) 磁気インキ文字読取用字体及び印字仕

様 (E13B)

- (3) 光学式文字認識のための字形 (片仮名)
- (4) 光学式文字認識のための印字仕様
- (5) 光学式文字認識のための手書き文字 (片仮名)
- (6) 光学式文字認識のための手書き文字 (数字)

〔入出力媒体〕

- (1) 印刷電信機のけん盤配列および符号
 - (2) 情報処理系けん盤配列
 - (3) 情報交換用磁気テープ
 - (4) NRZ-1方式による情報交換用磁気テープの情報記録様式
 - (5) 情報交換用磁気テープリール
 - (6) 情報交換用紙テープ
 - (7) 情報交換用紙カード
 - (8) 情報交換用紙テープの孔の位置と寸法
 - (9) 情報交換用紙カードの孔の位置と寸法
 - (10) 6枚形磁気ディスクパックの機械的互換性
 - (11) 6枚形磁気ディスクパックの磁気的特性
 - (12) 情報交換用磁気カセットテープ
 - (13) 情報交換用磁気カセットテープの情報記録様式
 - (14) 位相変調方式による情報交換用磁気テープの情報記録様式
 - (15) 情報処理用連続伝票
 - (16) 11枚形磁気ディスクパックの機械的互換性
 - (17) 11枚形磁気ディスクパックの磁気的特性
 - (18) 1枚形磁気ディスクカートリッジ (上面着脱形) の機械的互換性
 - (19) フレキシブルディスクカートリッジ
- 〔プログラム用言語〕
- (1) 電子計算機プログラム用言語FORTRAN (水準7000)

- (2) 電子計算機プログラム用言語FORTRAN (水準5000)
- (3) 電子計算機プログラム用言語FORTRAN (水準3000)
- (4) 電子計算機プログラム用言語COBOL
- (5) 電子計算機プログラム用言語ALGOL (水準7000)
- (6) 電子計算機プログラム用言語ALGOL (水準6000)
- (7) 電子計算機プログラム用言語ALGOL (水準5000)
- (8) 電子計算機プログラム用言語ALGOL (水準4000)
- (9) 電子計算機プログラム用言語ALGOL (水準3000)
- (10) 電子計算機プログラム用言語ALGOLの入出力 (水準70)
- (11) 電子計算機プログラム用言語ALGOLの入出力 (水準60)
- (12) 電子計算機プログラム用言語ALGOLの入出力 (水準50)
- (13) 電子計算機プログラム用言語ALGOLの入出力 (水準40)
- (14) 電子計算機プログラム用言語ALGOLの入出力 (水準30)

〔データ通信〕

- (1) 伝送回線上のキャラクタ構成と水平パリティの用法
- (2) モデムと通信制御装置およびデータ端末装置とのインタフェース
- (3) 基本形データ伝送制御手順
- (4) ハイレベルデータリンク制御手順のフレーム
- (5) ハイレベルデータリンク制御手順の手順要素
- (6) ハイレベルデータリンク制御手順の手順クラス

〔データコード〕

- (1) 都道府県コード
- (2) 市区町村コード
- (3) 日付の表示(コード)
- (4) 時刻の表示(コード)
- (5) 性別コード
- (6) 産業分類コード
- (7) 職業分類コード
- (8) 商品分類コード
- (9) 勘定科目コード
- (10) 情報交換用単位記号
- (11) 学歴区分コード
- (12) 大学、高等専門学校コード
- (13) 続柄コード
- (14) 地域メッシュコード
- (15) 地目コード

現在JIS規格として制定されていないが、今後の標準化項目として標準化が計画されているものは次のとおりである。

〔情報交換用媒体等ハードウェア及び情報記録形式関係〕

- (1) 磁気ディスクバックのトラックフォーマット
- (2) 磁気ディスクバックの磁気特性及び機械的互換性
- (3) OCR用ひらがな字形及び同印刷仕様
- (4) OCR用手書き文字(英字、記号、仮名、漢字)
- (5) I/Oインタフェース(物理仕様)
- (6) コーディング用紙
- (7) データシート
- (8) 磁気カード
- (9) CRTディスプレイ
- (10) 光学式マークカード
- (11) マイクロフィルム
- (12) 記録カード類
- (13) 日本語情報処理用漢字けん盤配列、字形

- (14) ファクシミリ
- (15) マイクロコンピュータ

〔情報の形式に関するもの〕

- (1) 磁気ディスクバック上での情報交換用符号の表現
- (2) 16進数の表現方法
- (3) グラフィックディスプレイ用符号

〔データ伝送関係〕

- (1) DTE/DCEインタフェース
- (2) データ通信ネットアーキテクチャのためのプロトコル

〔言語及びソフトウェア関係〕

- (1) プログラミング言語(BASIC, PASCAL, PL/I, APL, マイクロコンピュータ専用言語)
- (2) カセットテープのラベルとファイル構成
- (3) 磁気ディスクのラベルとファイル構成

〔情報の表現に関するもの〕

- (1) 「人」の属性コード
- (2) 国名コード
- (3) 事業体コード
- (4) オリジナルコード
- (5) ローカルタイムの表現
- (6) 科学技術用語コード
- (7) 貿易用語コード
- (8) 取引伝票の設計基準
- (9) POS符号
- (10) 商品別取引コード設計基準

〔環境・安全関係〕

- (1) 環境条件
- (2) 電源・安全条件

参 考 文 献

- (1) 石井善昭「計算機の基本方式」共立出版,
1973
- (2) 「ソフトウェアの標準化」日本経済新聞
社, 1977
- (3) J. G. Burch. Jr and F. R. Strater
Jr 「Information Systems」Hamilton,
1974
- (4) R. J. Codon「Data Processing Sys-
tems」Reston, 1975
- (5) 浦昭二「データ構造」共立出版, 1974
- (6) J. Martin「Computer Data Base
Organization」Prentice-Hall, 1977
- (7) Horowitz Sahni「Fundamental of
Data Structures」Computer Science,
1976
- (8) Kaimann「Structured Information
Files」Melville, 1973
- (9) L. K. Albrecht「Organnization and
Management of Information Processing
Systems」The Macmillan Co., 1973
- (10) J. K. Hughes and J. I. Michtom「A
Structured Approach to Programming」
Prentice-Hall, 1977
- (11) 岩尾達男「なぜコンピュータを使うか」
日本能率協会
- (12) 岩尾達男「続なぜコンピュータを使うか」
日本能率協会
- (13) 「JIS ハンドブック情報処理 1981」
日本規格協会
- (14) 「電気通信小六法」一二三書房, 1978
- (15) 「情報処理ハンドブック」オーム社
- (16) 「共立総合コンピュータ辞典」共立出版
- (17) 「電子計算機ハンドブック」オーム社
- (18) 「情報処理技術者に関する調査報告書-
カリキュラム'80-1第2種編」勸日本情
報処理開発協会情報処理研修センター,
1980

第5章 コンピュータシステム (C3)

キーワード

コンピュータ・システム	computer system
オペレーティング・システム	operating system
インタフェース	interface
マイクロプログラム	micro-program
記憶素子	storage element
記憶保護	storage protection
システム資源	system resource
仮想記憶	virtual storage
絶対アドレス	absolute address
相対アドレス	relative address
間接アドレス	indirect address
マクロ命令	macro instruction
コルーチン	coroutine
再帰ルーチン	recursive routine
モジュール	module
プログラム連係	program linkage
再配置	relocation
真理値表	truth table
1の補数	one's complement
2の補数	two's complement

目 標

1. コンピュータ・システムをハードウェアとソフトウェアの両面から概観し、コンピュータ・システムの全体像を理解させる。
2. 中央処理装置のデータ処理機能とシステム制御機能について、その構成要素と特性、仕組を理解させる。
3. 記憶装置の構成要素と機能、入出力とデータ通信に関連する装置の機能と特性を理解させる。
4. オペレーティング・システム概念を理解させ、オペレーティング・システムの重要な機能であるシステム資源の管理がどのように行われるかを理解させる。
5. アセンブリ言語の文法、命令の種類、命令形式、データの定義、アドレス指定の方法、などの特徴的な点を機械語や高水準言語と比較して説明できるようにする。
6. マクロ命令の概念とマクロ命令に関連した用語を理解させ、入出力制御のマクロ命令を使用できるようにする。

7. アセンブリ言語によるプログラムの区分化と連係の方法，モジュール化の概念を理解させる。
8. アセンブラの構成及びアセンブラと対比させながら解釈ルーチンの仕組みを理解させる。
9. コンピュータの仕組みとアーキテクチャを，基本論理設計，符号化と数の演算の面から取りあげ，コンピュータの原理を正しく理解させる。
10. コンピュータの基本ソフトウェアについて，プログラムの実行制御とコンピュータの操作制御の機能を理解させ，関連する重要な概念を習得させる。

これは，特種情報処理技術者と第1種情報処理技術者をめざす人の必修のコースであり，次の「システム分析」，「ファイル・システム」，「オペレーティング・システムとアーキテクチャ」などを学ぶために必要なコースである。

内 容

1. コンピュータ・システムの構成要素

コンピュータ・システムを構成する要素をハードウェアとソフトウェアに大別して説明する。ハードウェアとしては，プロセッサ，記憶装置，入出力装置，通信制御装置，端末装置を概観する。ソフトウェアとしては，オペレーティング・システム，適用業務プログラム，システム開発，運用，保守を助けるサポート・ツールを取りあげる。

2. ハードウェアの基礎

(1) 中央処理装置

プロセッサの基本機能として，データ処理機能とシステム制御機能を説明する。

プロセッサの特性を，データ形式，命令セット，アドレス方式，レジスタ数，制御方式，記憶装置や入出力装置とのインタフェースなどの面から説明する。演算論理機構（ALU），レジスタ，制御回路の仕組みを説明する。マイクロプログラムの役割を説明する。

(2) 記憶装置

記憶装置の特性と種類，階層的分类，記憶素子，高信頼性と記憶保護について説明する。記憶容量，記憶方式，アクセス方式，アクセス速度について正確に理解させる。

(3) 入出力と通信制御装置

入出力アーキテクチャ，入出力チャネル，入出力制御装置，入出力装置などについてよく理解させる。ネットワーク・システムについて説明し，通信制御装置，端末装置の機能と特性を理解させる。

3. システム資源の管理

これは，オペレーティング・システムのもっとも重要な機能の1つであり，システム資源を効率的に使用できるかどうか，システムの性能に大きく影響することを理解させる。管理の対象になる代表的な資源として，主記憶，外部記憶，仮想記憶，入出力装置，バッファ，ソフトウェア資源を取りあげて管理の概要を説明する。

4. アセンブリ言語

(1) 機械語

代表的な計算機について，機械語の種類と形式を説明する。プログラム制御，文字の表現，命令の取出しサイクルと実行サイクル，タイミングについて説明する。入出力の制御の仕方も理解させる。

(2) アセンブリ言語

アセンブリ言語の特性を機械語と対比させて説明する。代表的なアセンブリ言語の文法を説明し、命令形式、データの定義、リテラルの使用、ロケーション・カウンタなど、アセンブリ言語の特徴的な点を理解させる。

アセンブラの機能と特性をコンパイラと対比して説明する。

(3) アドレス指定の技法

絶対アドレスと相対アドレス、間接アドレス方式の概念と仕組を説明する。指標の使い方を理解させる。

(4) マクロ命令

マクロ命令の概念、マクロ命令の使い方を理解させる。マクロ命令の定義、マクロ呼出しとパラメタ、マクロ展開、入れ子、条件アセンブリなどを説明する。

(5) ファイル入出力

アセンブリ言語における入出力制御の仕方を説明する。入出力誤りの検出と、誤り回復の基本的な考え方を説明する。入出力管理システム、オペレーティング・システムのもとでの入出力指示ステートメントによる指示と、プログラムによる指示を説明する。バッファリングとブロック化、デブロック化の制御の仕方を説明する。

(6) プログラムの区分化と連係

主ルーチン、サブルーチン、コルーチン、再帰ルーチン、再入可能ルーチンについて説明し、モジュール化の概念を理解させる。プログラム連係の基本的な仕組みを説明する。

(7) アセンブラの構成

1パス・アセンブラと2パス・アセンブラ、再配置、再配置ローダを説明する。

(8) 解釈ルーチン

インタプリタ、シミュレータ、トレー

スなどの解釈ルーチンの仕組みと特徴を理解させる。

5. コンピュータの構造

(1) 基本論理設計

コンピュータのアーキテクチャを理解させる。主として次の内容を説明する。

(a) 2進数値によるデータと制御情報の表現

(b) 基本的な処理機構、ゲートと情報の記憶、フリップフロップ

(c) 真理値表、ブール関数、タイミング図による表現

(d) ゲートによる組合せネットワークの分解と合成

(e) 並列レジスタと直列レジスタ

(f) 簡単な同期制御機構の分解と合成

(g) データとアドレス・バス

(h) アドレッシングとアクセスの方法

(i) 記憶域の区分化

(j) タイミング・パルス発生の方法

(2) 符号化

B C DコードとA S C I Iコード、パリティ検査などについて復習し、符号器と復号器、符号変換を説明する。

(3) 数の表現と演算

2進数の正の数の加減算、1の補数、2の補数、符号付の数、基数増し法、B C Dと3増し表現による基本的な演算の得失などを説明する。

6. 基本ソフトウェア

(1) 実行制御のソフトウェア

解釈モジュール(リストプロセッサなど)、実記憶、仮想記憶の記憶管理モジュール、プロセッサ/チャネル制御モジュールの機能を説明する。再入可能プログラムの重要性、多重プログラミングと

多重プロセッシングの制御, バッチ・モードと対話モードを説明する。

(2) 操作制御のソフトウェア

ローディング, 割込み監視, 診断モジュール, スケジューリング, 資源割振り, 性能監視, 状態回復と保護機能を説明する。

(3) データとプログラム操作のソフトウェア

媒体と形式変換, ファイル処理と管理, トランスレータ, コンパイラ, セネレータ, 編集とデバッグ, サブルーチン連係とジョブ制御, データ/プログラムの識別と機密保護などを説明する。

指導上の留意点

1. ハードウェアについては, アーキテクチャを理解させることが重要である。できればVTRなどの動きのある映像でハードウェアの構造や実際の動きを見せながら説明することが望ましい。
2. ソフトウェアについては, オペレーティング・システムの資源管理機能を理解させることが重要である。また, ハードウェアとオペレーティング・システムがどのようにからみあい, 役割を分担しあっているかを理解させる必要がある。
3. 機械語とアセンブリ言語については, 身近にあるコンピュータを対象にして説明してもよいが, アセンブリ言語でプログラミングさせることが目的ではない。ここでは, アセンブリ言語を通じて, コンピュータの機能をどう制御するか, その原理とアルゴリズムを正しく理解させることが重要である。
4. 情報処理技術者試験では, 仮想計算機COMP-Xのアセンブリ言語CAP-Xを

通してアセンブリ言語の知識を問うことになっているので, CAP-Xの知識を習得させることも重要である。

5. マクロ命令については, マクロ命令の概念と同時に, マクロ展開などマクロ命令の特徴的な事項をよく理解させる。
6. 基本論理設計, コンピュータ内部の数の表現と演算方式などは, 日頃なじみの少ない分野なので, 順序だててきめ細かく説明する必要がある。
7. 基本ソフトウェアについては, オペレーティング・システムの機能を中心に, 実行制御や操作制御のソフトウェアを概観する。詳しい説明は, このあとの「オペレーティング・システムとアーキテクチャ」で行うので, ここでは, おおまかな説明にとどめる。
8. このコースは, 「情報処理技術概説」を習得した人を対象にしている。このコースを学んだあとは, 「ファイル・システム」又は「オペレーティング・システムとアーキテクチャ」に進むことを想定している。

参 考

1. 用語の説明

オペレーティング・システム

計算機プログラムの実行を制御するソフトウェアで, スケジューリング, デバッグ, 入出力制御, 課金処理, コンパイル, 記憶域割当て, データ管理及び以上に関連した諸サービスを行うもの。

インタフェース

二つ以上の構成要素の境界, 又は境界において共用される部分。

二つ以上の装置を連結するハードウェアが構成要素である場合や二つ以上のプログラムで共有される記憶装置の一部又はレジス

タである場合などがある。

マイクロプログラム

特定の計算機の演算に対応する要素的な命令の列で、特別な記憶装置に格納され、通常、計算機の命令レジスタに計算機命令が導入されるとその列の実行が始まる。

記憶素子

基準のフレーム以外のデータを含まないデータ媒体、例えば、あらかじめ印刷されている書式、送り孔だけがパンチされたテープ、消去された磁気テープ

記憶保護

書込み、読取り又はその両方を阻止することによって、記憶装置へのアクセス又は一つ以上の記憶場所へのアクセスを制限すること。

システム資源

システムを運転するために必要な設備、例えば、主記憶装置、入出力装置、ファイル、

仮想記憶

計算機システムの利用者がアドレス可能な主記憶装置とみなしうる記憶空間で、仮想アドレスが実アドレスへ写像されるもの。仮想記憶の大きさは、計算機システムのアドレス指定方式及び使用可能な補助記憶装置の量によって制限され、主記憶場所の実際の大きさによっては制限されない。

絶対アドレス

計算機言語中のアドレスであって、中間的な参照を使用せずに記憶場所又は装置を識別できるもの

相対アドレス

基底アドレスからの差として表されるアドレス

間接アドレス

オペランドのアドレスとして取り扱われるデータ項目の記憶場所を指示するアドレスで、必ずしもその直接アドレスでなくても

よい。

マクロ命令

原始言語中における一つの命令で、同じ原始言語中の定義された一連の命令によって置き換えられるべきもの。

コルーチン

個々のプログラムのどれが主ルーチンでどれがサブルーチンというように呼出しに關しての主従関係が存在しなくて互いに相手のサブルーチンであるように呼び出すことのできる制御方法を備えたプログラム群のことである。

再帰ルーチン

サブルーチンの一つで、それ自身を直接呼び出したり、それ自身が呼び出した別のサブルーチンから呼び出されたりすることによって、それ自身をサブルーチンとして使うことができるもの。

再配置

計算機プログラム又はその一部を移動し、かつ移動後にその計算機プログラムが実行されるように所要のアドレス参照を調整すること。

真理値表

論理演算に対する演算表

1の補数

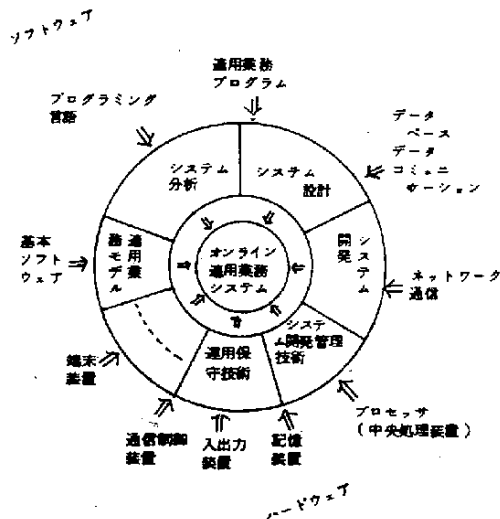
純2進記数法における減基数の補数

2の補数

純2進記数法における基数の補数

2. システムの構成要素とシステム構成技術

これは、オンライン・システムを設計開発する場合の例である。大きく見れば、ハードウェア的な要素とソフトウェア的な要素から構成されているが、それを分解してみると、図のようになる。



3. 中央処理装置の基本機能

中央処理装置の基本機能は、与えられた命令（プログラム）を実行することである。

実行する命令の特質によって、2種類に分けることができる。

(1) データ処理機能

記憶装置にあるデータの演算処理や判断を行う。

(2) システム制御機能

コンピュータ・システムを構成する資源を管理する。例えば、次のことを行う。

- (a) 入出力の起動
- (b) 割込み制御
- (c) 障害検出や障害修復

中央処理装置の特性を示す項目には次のようなものがある。

- (a) アーキテクチャ
- (b) 構成要素
- (c) 構成方式
- (d) 結合方式（インタフェース）

(1) アーキテクチャ

中央処理装置のアーキテクチャは、ユーザとの機能的接点を定める重要なものである。

アーキテクチャとは、一般に「プログラマから見たシステムの属性で、データの流れや制御、論理設計、物理的な実現方法とは別の概念的構造及び機能的動作」と定義されている。

アーキテクチャの基本要素には、次のようなものがある。

(a) データ形式

- 2進数、10進数
- 固定小数点、浮動小数点
- 配列、リストなどのデータ構造の表現、文字列、ビット列

(b) アドレス指定方式

- スタック方式
- 指標修飾
- 間接アドレス方式
- 相対アドレス方式

(c) 命令形式

- 命令語の構成
- 0アドレス方式
- 1アドレス方式
- 2アドレス方式
- 3アドレス方式

(d) システム制御機能

- 入出力の起動
- 割込処理
- 計時機構
- 初期プログラム・ロード
- 障害検出
- 記憶保護

(2) 中央処理装置の基本構成

演算論理装置（ALU）、レジスタとフリップ・フロップ、制御回路などがある。

(3) マイクロプログラム

最近の多くのプロセッサには、配線論理の代りに、マイクロプログラム制御方式が採用されている。マイクロプログラ

ムには、次のような特徴がある。

- (a) 不規則で複雑な制御論理の大部分が規則的なメモリ構造に置き換えられる。
 - (b) ハードウェアの細部をプログラムの制御でできる。
 - (c) 複雑な機能を持ったアーキテクチャを比較的 low 価格で実現できる。
 - (d) 制御論理の変更、修正が容易である。
- マイクロプログラムは、
- (a) エミュレーション
 - (b) 高水準言語処理のサポート
 - (c) 制御プログラムの高速化
 - (d) 診断と保守
- などに応用されている。

マイクロプログラムは、高速な制御記憶装置である固定記憶装置 (ROM) や書き込みも可能な制御記憶 (WCS) に記憶される。

4. 記憶装置

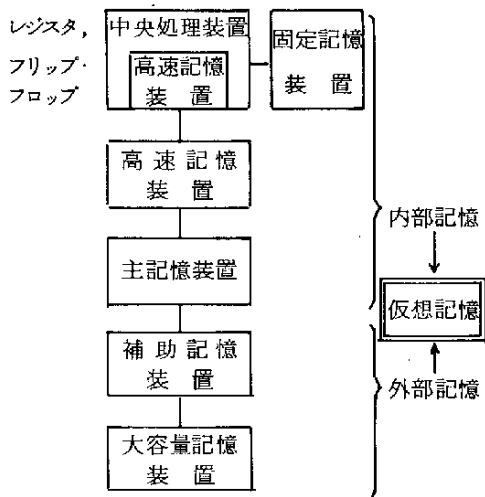
次のような点を計数的に把握することによって、記憶装置の特性をつかむことができる。

- (a) 参照時間 (アクセス時間)
- (b) 書き込み時間
- (c) サイクル時間
- (d) 読取りビット数
- (e) 書き込みビット数
- (f) 容量
- (g) ビット当りのコスト
- (h) データ転送速度
- (i) 揮発性/不揮発性
- (j) アクセス特性

記憶装置は、いくつかの種類に分類されるが、それは図のような階層を構成している。

- (1) 高速記憶装置は、中央処理装置が演算のために使用する。レジスタ、フリップ・フロップ、制御記憶、高速バッファ記

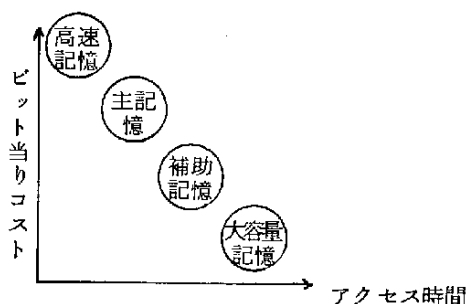
憶などがこれに含まれる。容量よりもアクセス時間の速さが重視される。



- (2) 主記憶装置は、中央処理装置が実行するプログラムとデータを格納する。中央処理装置の規模に応じた速度、容量、ビット幅を持つ。
- (3) 補助記憶装置は、主記憶装置を補うために使用する。中程度のアクセス時間で、少なくとも主記憶より10倍ないし100倍大きい容量を持ちコストは安い。
- (4) 大容量記憶装置は、ファイルを貯えるために用いる。容量が大きいことがもっとも重要で、次にアクセス時間やデータ転送速度が問題になる。
- (5) 固定記憶装置 (ROM—読取専用記憶装置) は、マイクロプログラムのような固定的に必要な情報を入れておくのに用いられる。通常の動作時には書き込みは行えず、書き込みは、製造時や使用時に特別な装置を用いて行う。
- (6) 仮想記憶は、ハードウェアの動的アドレス変換機構 (DAT) とオペレーティング・システムの仮想記憶アクセス機能により、内部記憶と外部記憶を一体のも

のとみなして構成された仮想空間にプログラムでアクセスできるようにした仮想的な記憶装置である。

ビット当りコストと記憶装置の性能との間には、次のような対応関係がある。



記憶素子では、半導体記憶素子と磁気記憶素子が重要である。

(1) 半導体記憶素子

半導体記憶素子の特徴は高速性にある。中でもバイポーラ型がもっとも高性能であり、高価である。MOS型は性能面ではバイポーラより劣るが、価格はバイポーラより安価である。CCD記憶は、ソフト・レジスタ型の構造で、性能より安価な点に特徴がある。一般に次のような組合せで使用される。

バイポーラ型	高速記憶装置
MOS型	主記憶装置
ROM	マイクロプログラム用記憶装置
CCD	高速補助記憶装置

(2) 磁気記憶素子

磁気記憶素子の特徴は、情報の不揮発性である。したがって磁気記憶は外部記憶として大量のデータを貯蔵する目的に用いられる。

ランダム・アクセス記憶

磁心(磁気コア)	主記憶
磁性膜	

磁性線(ワイヤ) } 高速補助記憶
ソフト・レジスタ型

磁気バブル } 高速補助記憶
回転型

磁気ディスク } 外部記憶装置
(可動ヘッド,
固定ヘッド,
フロッピー・ディスク)

その他
磁気テープ } 大容量記憶装置
磁気カード }
磁気テープ・カートリッジ
カセット・テープ }

記憶装置では、記憶されたデータを誤りなく再生できなければならない。そのために、次のような手法が用いられる。

誤り検出

奇偶検査(水平, 垂直)

比較検査

誤り訂正

誤り訂正符号による自動訂正

多数決論理に基づく修正

また記憶された内容を保護するために、次のような記憶保護方式がとられている。

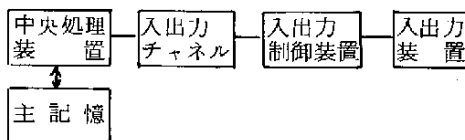
キーによる保護

アクセス権による保護

物理的手段による書き込み禁止(磁気テープの保護リングなど)

5. 入出力装置

入出力に関連する機器は、次のように構成される。



このような構成によって、入出力装置の動作を制御し、入出力装置と主記憶装置と

の間のデータの転送を実行する。

入出力装置のデータ転送速度は、主記憶のサイクル時間よりはるかに遅いので、次のようなことが必要になる。

- 入出力装置と主記憶の間のデータ転送（データの入出力）を中央処理装置の処理とは独立に行う。
- 複数の入出力動作を同時に実行できるようにする。

以上が、入出力アーキテクチャの概要である。

(1) 入出力チャンネル

入出力チャンネルは、中央処理装置によって起動され、入出力制御装置の制御を行う。

入出力チャンネルは、入出力指令を解読し、入出力制御装置や入出力装置に対する動作を指定し、データ転送を制御する。さらに、入出力装置の状態を把握し、障害の検出と処理を行う。

但し、小型コンピュータやミニコンピュータでは、入出力チャンネルの機能の一部又は、全部を中央処理装置が直接実行することもある。

代表的に入出力チャンネルに、選択チャンネルと多重チャンネルがある。多重チャンネルには、ブロック多重チャンネルとバイト多重チャンネルがある。

入出力命令の実行に伴なり入出力チャンネルの動作は次のように行われる。

(a) チャンネル・プログラム

一般に、入出力命令では入出力動作の起動だけを指定し、それ以降の実際の入出力動作は、入出力指令で指定する。

指令は、単独でも実行できるが、複数の指令を組み合わせでチャンネル・プログラムを作り、チャンネル・プログラ

ムで入出力動作を制御することもできる。

指令とチャンネル・プログラムについては、

入出力指令の形式

入出力指令の種類と機能

データ・チェーニング

指令チェーニング

などの知識が必要である。

入出力チャンネルは、チャンネル・プログラムを解読し、指定された入出力制御装置に指令を送って、データ転送を制御する。

(b) 入出力インタフェース

入出力チャンネルが入出力制御装置との間で、制御信号やデータの受け渡しのための標準的な手順が入出力インタフェースである。

入出力インタフェースについては、国内、国際的に標準化の努力が行われているので、標準化の動向に注目する必要がある。

(2) 入出力制御装置

入出力制御装置は、入出力チャンネルと情報を交換し、入出力装置を直接制御する。

入出力制御装置は、一般に次のような機能を持っている。

(a) 入出力チャンネルから送られてきた指令を解読し、指定された入出力装置を作動させる。

(b) 入出力装置のデータ転送を制御する。データ転送にともなり符号変換、直列並列変換、バッファ制御、誤り検出を行う。

(c) 入出力装置の状態を監視し、動作の完了や異常状態を検出し、状態情報を入出力チャンネルに送る。

(3) 入出力装置

入出力装置とは、広い意味では、中央処理装置又は主記憶装置へ入出力するすべての装置のことである。

したがって、磁気ディスク装置も入出力装置になる。

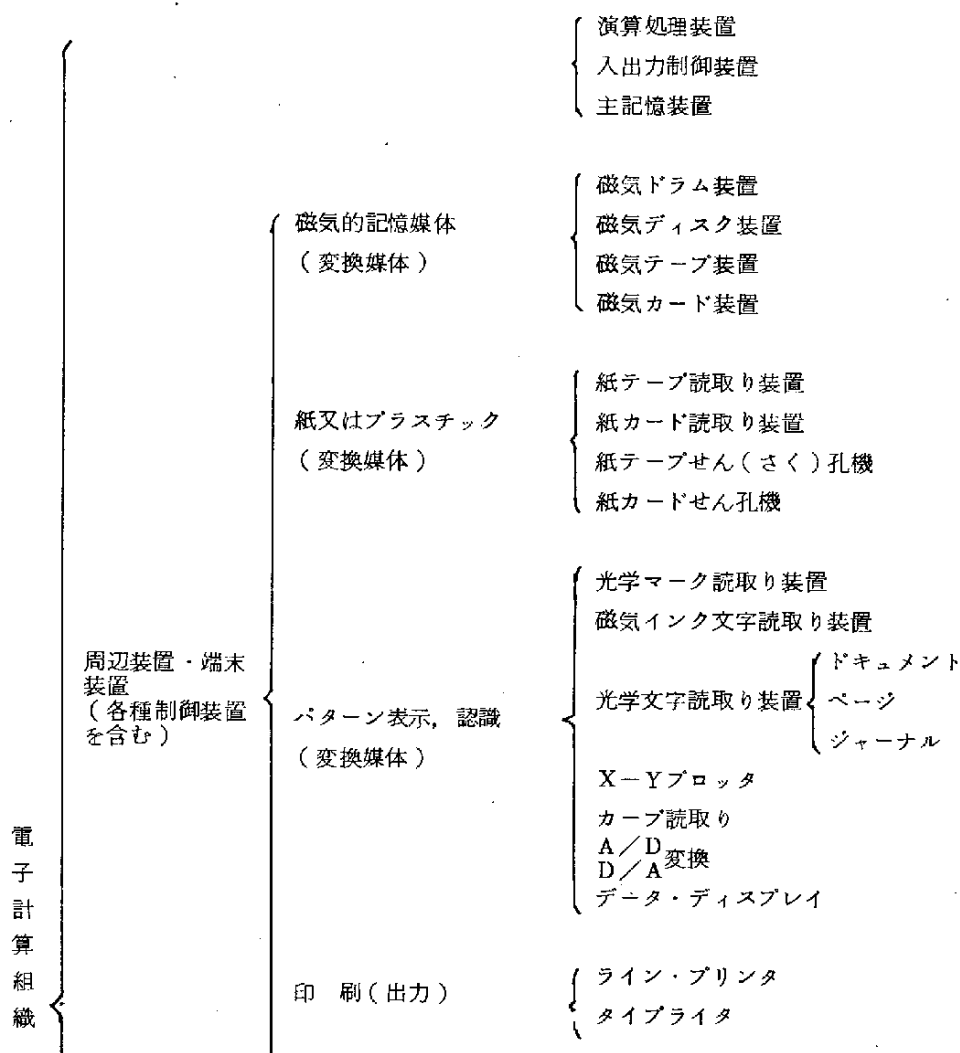
入出力装置ということばを狭い意味で使うときは、コンピュータ・システムと外部（人や装置）との入出力を行う装置

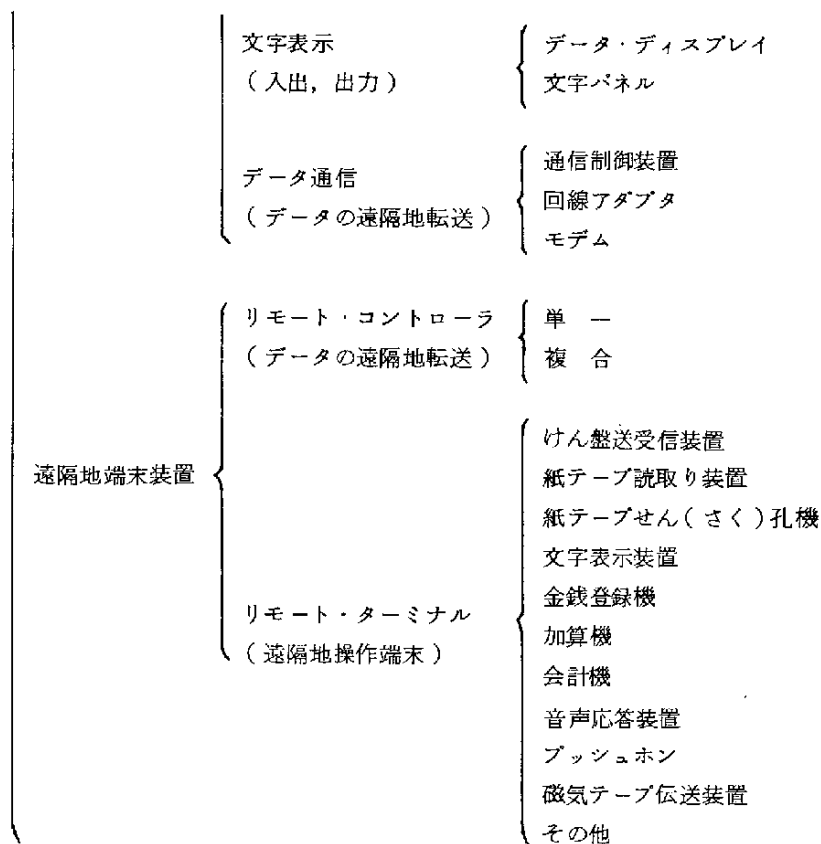
を指す。この場合は、システム・ファイルとして使用する磁気ディスク装置などは、補助記憶装置と呼び、入出力装置とは区別する。

JIS規格では、広い意味の入出力装置のことを周辺装置と呼んでいる。他に端末装置ということばもある。

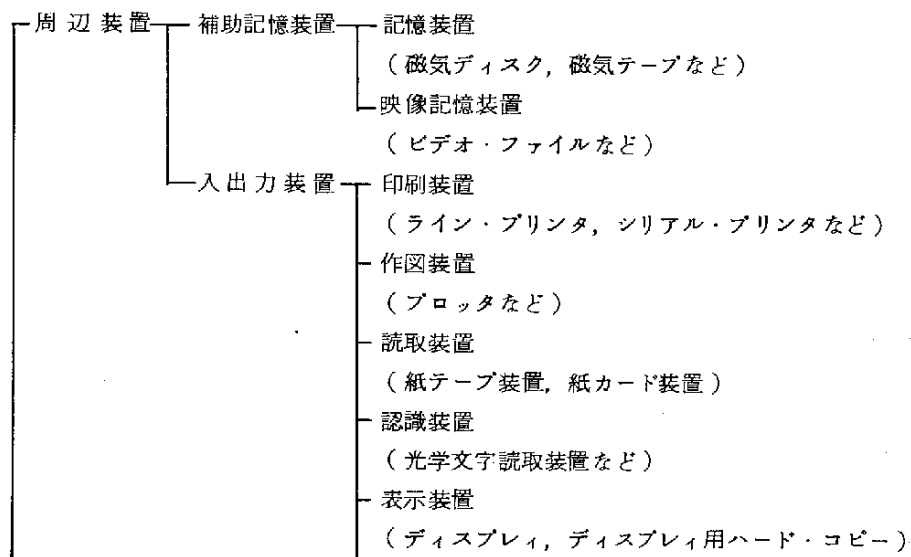
一般的な入出力装置には次のようなものがある。

電子計算組織構成図 (JIS C6272 解説より)

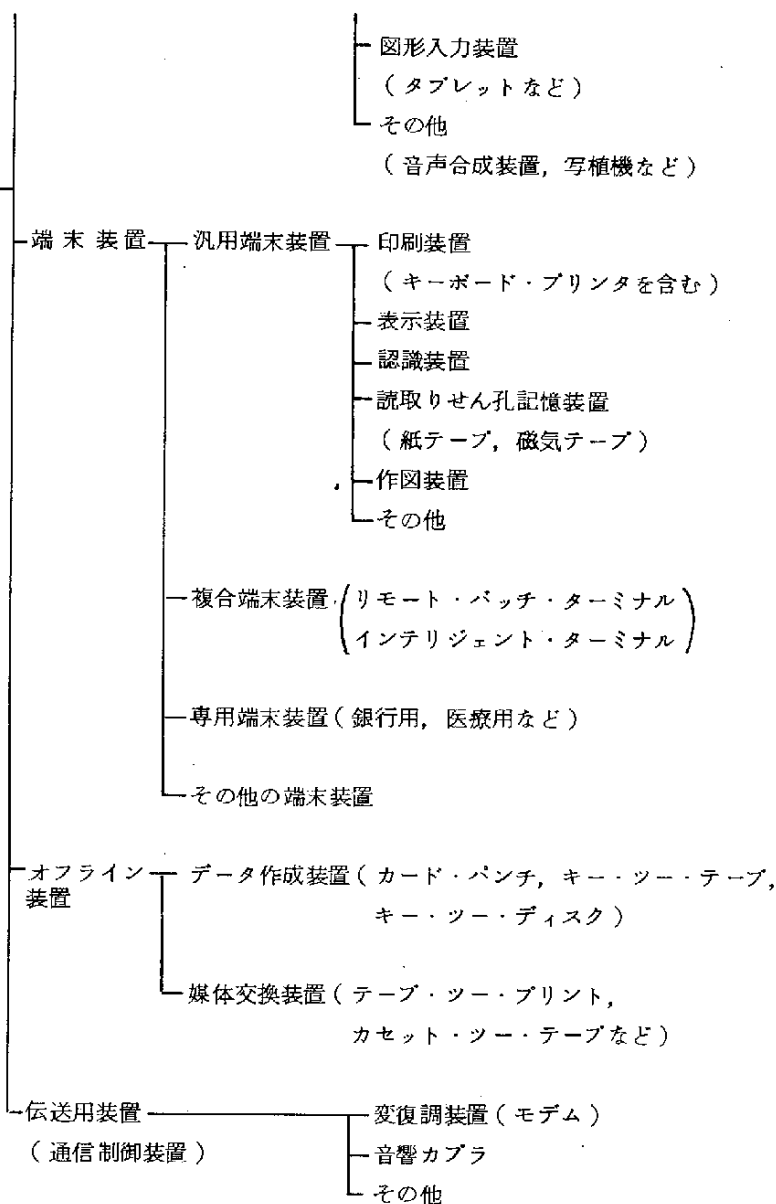




周辺端末装置分類 (日本電子工業振興協作成)



周辺
端
末
装
置



6. ネットワーク・システム

遠隔地の端末やコンピュータを通信回線を介してコンピュータと結合したネットワーク・システムには様々な形態があるが、代表的な形態は次のようなものである。

中央処理装置と端末の結合

一般的なオンライン・システム

タイムシェアリング・システム(TSS)

遠隔ジョブ入力(RJE)など

中央処理装置同志の結合

コンピュータ・ネットワーク

端末同志の直接結合

メッセージ通信

(1) 通信制御装置

通信制御装置は、コンピュータと通信回線を結合する機能を持つが、主として次のような機能を実行する。

(a) 回線制御

伝送制御
回線状態の監視
結合路の制御
誤り処理

(b) データ制御

コード変換
メッセージ形成

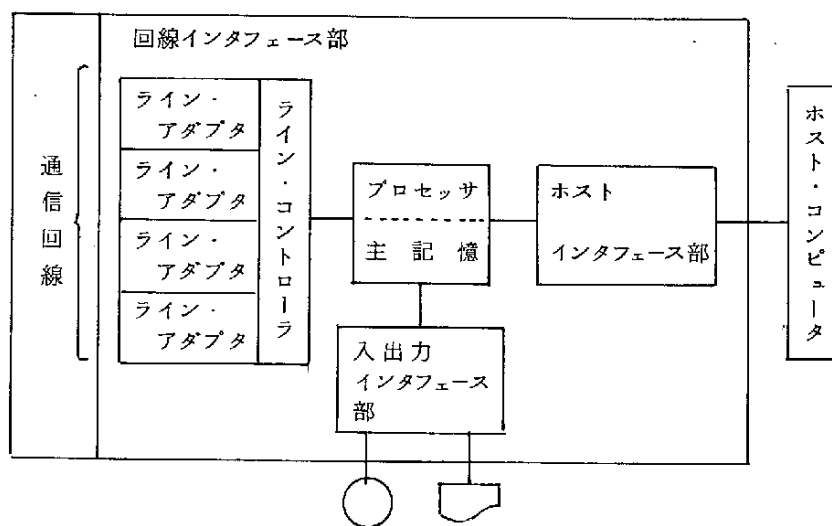
(c) メッセージ制御

メッセージのバッファリング機能

(d) ホスト・コンピュータとの通信制御

チャネルとのインタフェース制御

通信制御装置は、一般に次のように構成されている。



(2) 端末装置

端末装置には、タイプライタ型やディスプレイ型の単純な機能を持ったものと、ミニコンピュータやマイクロコンピュータの能力を内蔵した知能端末（インテリジェント・ターミナル）がある。

端末装置には、次のような基本機能と

拡張機能がある。拡張機能は、知能端末が有する機能である。その演算結果に従って、1か0がセットされる。その他の命令の実行によっては変更されない。

(6) 命令語は、次のような16ビットの構成を持つ。

ビット位置→	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
フィールド名→	OP				GR		XR		A D							

命令の各フィールドは、次のような意味を持つ。

(a) OPフィールド：命令コードを指定

する。

(b) GRフィールド：GRの番号又はJC命の判定条件を指定する。

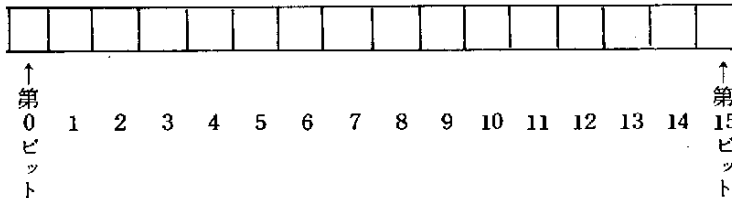
- (c) XRフィールド：アドレス修飾を行うGRの番号を指定する。第6ビットと第7ビットがともに0のときには、指標レジスタによるアドレス修飾は行わない。指標レジスタとして使用できるのは、GR1, GR2。

(a) 基本機能

回線制御
通信手順制御
誤り制御
入出力機器の制御
同時動作の制御
データの入出力

(b) 拡張機能

オペレータに対する指示
編集
誤り検査



- (3) 数値は、16ビットの2進数で表現する。負数は2の補数で表わす。
(4) 制御方式は逐次制御で、1アドレスの命令語を持つ。
(5) レジスタとしてBR(16ビット), GR(16ビット), SC(16ビット), CC(1ビット)を持つ。

BR(基底レジスタ)は、実効アドレス16ビットのうち、上位8ビットを定めるために用いられる。BRの下位8ビットは常に0である。

GR(汎用レジスタ)は、0番から3番まで4個あり、それぞれGR0, GR1, GR2, GR3である。この4個のレジスタは、算術演算と論理演

ファイル処理
コード変換
符号による特殊制御
演算処理

7. 仮想計算機COMP-Xとアセンブリ言語CAP-X

(I) 仮想計算機COMP-Xの仕様

- (1) COMP-Xは、1語16ビットの計算機で、0を含めて256の整数倍の番地から始まる連続した256語を1記憶ブロックとして、最小1記憶ブロックから最大256記憶ブロックを実装できる。N個の記憶ブロックを使用するとき、アクセスできるアドレスは、0番地から $256 \times N - 1$ 番地までである。
(2) 1語のビット構成は次のようになっている。

算に用いられる。このうちGR1, GR2, GR3は、指標レジスタとしても用いられる。

SC(制御カウンタ)には、ある命令の実行中、その命令が格納されているアドレスがセットされる。命令の実行を終わったときには次に実行する命令のアドレスがセットされる。すなわち、一般には命令の実行が終わるとSCに1が加算される。飛越し命令の場合は新しくセットしなおされる。

CC(条件コード・レジスタ)には、加算命令や演算命令の実行が終わったときGR3だけである。ただし、SFT命令ではシフトの方向を指示する。

(d) ADフィールド：アドレスの下位8ビットを指定する。

(7) 命令の実効アドレスは、16ビットで表される。そのうち上位8ビットは、BRの上位8ビットが常に用いられる。実効アドレスの下位8ビットは、ADフィールドの値にXRフィールドを指定したGRの第8ビットから第15ビットまでの値を加えて得られる値である。この値が256以上の値になったときは、256を法(モード)とする(モード256)0から255までの値がとられる。

つまり、ADフィールドの値と指標レジスタの下位8ビットとの和を256で割った剰余にBRの値を加えて、実効アドレスにする。

(8) COMP-Xには、入出力を行う命令はない。入力とは直接打鍵することによって任意の記憶域又は任意のレジスタに2進数をセットする。記憶域の内容又はレジスタの内容は、計算機を停止させた状態で直接読み取る。

(9) 命令コードには、次のような種類がある。

2進表示	16進表示	ニック表示	読み方	機能		
0000	0	HJ	Halt and jump	SCに実効アドレスを入れて停止する。その後、スタートボタンを押すとSCの示すアドレスから再び命令の実行が開始される。この命令のGRフィールドは無視される。		
0001	1	JNZ	Jump if GR is not zero	GRフィールドで指定するGRの内容が0でないとき、実効アドレスにジャンプする。		
0010	2	JC	Jump on condition	GRフィールド(第4, 第5ビット)の値によって次のような意味をもつ。		
				第4ビット	第5ビット	意 味
				0	0	何もせずに次の命令に移る。
				0	1	CCの内容が1のとき、実効アドレスにジャンプする。
				1	0	CCの内容が0のとき、実効アドレスにジャンプする。
				1	1	無条件に実効アドレスにジャンプする。
0011	3	JSR	Jump to subroutine	GRフィールドで指定したレジスタに現在のSCの内容+1をセットして、実効アドレスが指定する語の内容をSCとBRに入れる。BRの下位8ビットは0にリセットされる。この命令はBRの内容を変更する唯一の命令であり、任意のアドレスにジャンプすることができる。記憶ブロックの異なるサブルーチンへのジャンプとサブルーチンからの戻りはJSR命令で行う。		
0100	4	SFT	Shift	指定されたGRの内容を、符号ビット(第0ビット)		

				を除いて、右又は左へ、ADフィールドで指定されるビット数だけシフトする。XRフィールドが、0なら右へ、1なら左へシフトする。シフトの結果、空いたビット位置には、左シフトの場合は0が入り、右シフトの場合は、符号ビットと同じものが入る。
1000	8	L A I	Load address immediate	実効アドレスの下位8ビットが指定されたGRの下位8ビット(第8ビットから第15ビット)に入り、GRの上位8ビット(第0ビットから第7ビット)はすべて0になる。この命令は、GRの内容を0~255の数値に変更するために用いる。
1010	A	A D D	Add	指定されたGRの内容と、実効アドレスで指定する語の内容とを加算し、その結果を指定されたGRにセットする。演算の結果が負ならCCに1をセットし、0か正ならCCに0をセットする。
1011	B	S U B	Subtract	指定されたGRの内容から、実効アドレスで指定する語の内容を引いて、その結果を指定されたGRにセットする。演算の結果が負ならCCに1をセットし、0か正ならCCに0をセットする。
1100	C	L D	Load	実効アドレスで指定する語の内容を、指定されたGRにセットする。
1101	D	S T	Store	指定されたGRの内容を、実効アドレスで指定する語に入れる。GRの内容は変わらない。
1110	E	A N D	And	指定されたGRの内容と、実効アドレスで指定する語の内容とのビットごとの論理積を、指定されたGRに入れる。
1111	F	E O R	Exclusive or	指定されたGRの内容と、実効アドレスで指定する語の内容とのビットごとの排他的論理和を、指定されたGRに入れる。

(Ⅲ) アセンブリ言語CAP-Xの仕様

COMP-Xのためのアセンブリ言語をCAP-Xという。その文法は次のとおりである。

CAP-Xは、5種類の擬似命令START, END, RESV, CONST, ADCONと通常の命令とからなる。この命令は、ラベル欄、命令コード欄、オペランド欄の3つの欄をもつコーディング用紙に次のような形式で書く。

ラベル 命令コード オペランド
m START n

END n
m RESV n
m CONST h
m ADCON n
m 命令コード g, n, x

(1) ラベル欄

擬似命令END以外は、ラベルをつけることができる。ラベル欄のmは、ラベルを書くか又は空白(ラベルをつけない場合)にしておく。

ラベルは、3文字以内で次のいずれかの形式で書く。

x
xy
xyy

先頭の文字 x は、英字の大文字でなければならない。y は、英字の大文字又は数字である。

RESV 命令につけられるラベルは、RESV 命令でとられる領域の先頭の語のアドレスを示す。

START 命令につけられるラベルは、START 命令で始まるプログラムの先頭の語を指すと同時に、別のプログラムからの入口名として使用される。別のプログラムからは、ADCON 命令のオペランドにそのラベル名を書くことで参照できる。

(2) START n

プログラムの先頭に、必ずこれを書く。n は 10 進数で、プログラムの格納開始アドレスを指定する。

(3) END n

プログラムの最後には、必ずこれを書く。n は 10 進数か、又はラベル名で、プログラムの実行開始アドレスを指定する。n を省略することもできる。

(4) CONST h

h には 4 けたの 16 進数 (0~9, A~F) を書く。h に書かれた 16 進数が 1 語の定数として格納される。

(5) RESV n

n には 10 進数を書く。n 語の領域を確保することを指示する。プログラム格納時には、その領域の内容は変更されない。

(6) ADCON n

n にはラベル名又は 10 進数を書く。n がラベル名で、そのラベル名が同一プログラム内で定義されているときは、

CAP-X アセンブラによってアドレス定数が決まる。ラベル名が同一プログラム内で定義されていないときは、CAP-X アセンブラはアドレス定数の決定を別のシステム・ライブラリにまかせ、実行に先だって別のプログラムの START 命令のラベルと結合してもらう。

(7) 命令コード g, n, x

CAP-X の通常の形式である。命令コードにはニモニク表示の命令コードを書く。g には GR フィールドに入れる数字 0~3 を書く。n には SFT 命令の場合にはシフトするけた数 (10 進数 0~15) を書き、その他の命令の場合には、ラベル名又は絶対アドレス (10 進数 0~255) を書く。g と n の指定を省略することはできない。x は XR フィールドに入れる数字 0~3 を書く。x は省略できる。x を省略するときはその前のコンマも省略する。x を省略すると XR フィールドには 0 が入る。

参 考 文 献

- (1) 森宗正他「ハードウェア入門」共立出版,
1971
- (2) H. カッツアン・ジュニア; 大日方真訳
「コンピュータ・オーガニゼーションと I
BM システム / 370」TBS 出版会, 1972
- (3) B. W. Kernighan and P. J. Pleuger
「Software Tools」Addison-Wesley,
1976
- (4) 「ソフトウェアの標準化」日本経済新聞
社, 1977
- (5) 藤野喜一他「計算機システム基礎論」共
立出版, 1973
- (6) 三上, 亀田「コンピュータと情報システ
ムにおけるシミュレーション技術」電子通
信学会誌 Vol.60 p.761-769, 1977
- (7) 箱崎, 小野「ハードウェアモニタの技法
について」情報処理学会 S E 研資 74-1,
1974
- (8) 喜田村善一「コンピュータ・ネットワー
ク」1975
- (9) N. Abramson, F. K. Kuo, 「Computer
Communication Networks」Prentice-
Hall, 1972
- (10) Barber, 「Communications Networks」
online, 1975
- (11) Stuart E. Madnick, John J. Donovan,
池田克夫訳「オペレーティング・システム」
日本コンピュータ協会, 1976
- (12) 「IBM システム / 370 オペレーティン
グ・システム, アセンブラ言語」日本 IBM
- (13) D. W. Barron, 「Assemblers and
Loaders」MacDonald London and Ame-
rican Elsevier, 1969
- (14) 「JIS ハンドブック情報処理 1981」
日本規格協会

第6章 ファイルシステム (C5)

キーワード

ファイル	file
レコード	record
ファイル処理	file processing
順次アクセス記憶装置	sequential access storage device
直接アクセス記憶装置	DASD: direct access storage device
ファイル編成法	file organization
順編成	sequential organization
索引つき順編成	indexed sequential organization
直接編成	direct organization
区分編成	partitioned organization
データ構造	data structure
データベース	database
連結リスト, つなぎ並び	linked list
木構造	tree structure
B-木	B-tree
転置ファイル	inverted file

目 標

1. 伝統的なファイルとその処理システムの基礎を理解させ、2次記憶上でのデータの構造化と処理法を身につけさせる。
2. 2次記憶装置を利用する経験を与える。
3. ファイル処理の概念だけでなく、ファイルに対するデータ構造を設計し、これと適合した操作プログラムを作成する能力を養う。

ファイルの物理的構造、ファイル処理を支援するためのオペレーティング・システムの役割などについて説明し、以下の章に対する導入を行う。

(1) ファイルの基本概念

以下のような基本的な事項を説明あるいは復習し、ファイル処理についての基本的な概念を理解させる。

- ファイルの構成
- レコードの形式
- ブロック化
- ファイル処理

(2) 記憶媒体

ファイルに対して用いられる記憶媒体

内 容

1. ファイル処理の基本

ファイルの概念、ファイル処理の概念を理解させ、ファイルのための記憶媒体、フ

の種類、特性について整理し、知識を確認する。

- 順次アクセス記憶装置
- 直接アクセス記憶装置

(3) ファイルの構造

ファイルあるいは複数のファイル群からなるファイル・システムの構造表現に用いられるデータ構造やファイルの構成で慣用化されているファイル編成法について概略を示し、以降に対する見通しを与える。

- データ構造
- ファイル編成法

(4) オペレーティング・システムとファイル処理

ファイル処理に対するオペレーティング・システムの役割、ファイル処理のための命令について整理する。一般的なオペレーティング・システムでのファイル管理の様相に加えて、それぞれの教育環境で利用する計算機上でのオペレーティング・システムに基づき、具体的なマクロ命令やジョブ制御文について理解させる必要がある。

- アクセス法
- ファイル操作命令
- ファイルの割当て
- ファイルのカatalog
- ファイル処理に関するユティリティ・プログラムとライブラリ・ルーチン

2. ファイル編成法

ファイルを2次記憶媒体に構成し、そのアクセスを支援するために、オペレーティング・システムでは幾つかのデータ構造とそのアクセス手段を操作化しファイル編成法として提供している。これらのうち代表的かつ基本的な編成法について、その構造、特徴、処理方法を理解させる。

(1) 順編成ファイル

順編成ファイルの構造、特徴、処理法について理解させる。

- 順編成ファイルの構成
- 順編成ファイルの特徴
- 順編成ファイルの処理

ファイルの生成、更新、読み出し、分類(sorting)、併合(merging)、突き合せ(matching)

(2) 索引つき順編成ファイル

索引つき順編成ファイルの構造、特徴、処理法について理解させる。

- 索引つき順編成ファイルの構成
- 索引つき順編成ファイルの特徴
- 索引つき順編成ファイルの処理

ファイルの生成と索引の生成、ファイルの更新、読み出し、順アクセスと乱アクセスの組み合わせ、多段階索引による入出力の効率化、ファイルの再編成(reorganization)

(3) 直接編成ファイル

直接編成ファイルの構造、特徴、処理法について理解させる。

- 直接編成ファイルの構成
- 直接編成ファイルの特徴
- 直接編成ファイルの処理

ファイルの生成、更新、読み出し、代表的なハッシング法(hushing)、同義語(synonym)の解釈

(4) 区分編成ファイル

区分編成ファイルの構造、特徴、処理法について理解させる。

- 区分編成ファイルの構成
- 区分編成ファイルの特徴
- 区分編成ファイルの処理

ファイルの生成、更新、読み出し、再編成(reorganization)、圧縮(compaction)

(5) プログラム言語によるファイル処理支援

代表的なプログラム言語、COBOL、FORTRAN、PL/I、PASCALなどでのファイル編成法とファイル・アクセスに対する機能を理解させる。

3. ファイル・システムのためのデータ構造

2次記憶媒体を利用する場合の物理的構造に重点を置いた観点から計算機メーカにより経験的に集約されたアプローチである各種のファイル編成法に対して、データの論理的構造の表現を重視する時のアプローチとして、各種のデータ構造のファイル(ファイル・システム)の構成への利用を説明する。

(1) レコードの論理的構造

レコード間の論理的関係を構造として表現する考え方を説明する。このような見方として基本的には、線形構造、木構造、ネットワーク構造、階層構造などがあることを説明し、これらとその物理的な表現形態の関連を理解させる。このようなアプローチの進展した形態としてデータベース(database)があることを理解させる。

- レコード間関係
- データベース
- 物理的表現

(2) 連結リスト

最も基本的かつ柔軟なデータ構造としてポインタ(pointer)あるいはつなぎ(link)を用いて関連性を直接的に示す連結リストによるデータ構造化法を説明する。

- 連結リスト

連結リストの定義、循環連結リスト(circular-linked list)、操作アルゴリズム

- 多重つなぎ(multi-linked list)

(3) 木構造

レコードの論理的構造を表現するために最もよく用いられてきた木構造の利用について説明する。

- 木構造

木の定義、2分木(binary tree)

AVL-木(AVL-tree)、木の巡回アルゴリズム、均衡化(balancing)、2次記憶媒体での利用に対する配慮

- B-木

B-木(B-tree)、B^{*}-木(B^{*}-tree)、レコードの挿入、削除のアルゴリズム、木の動的再構成

(4) 転置ファイル

ファイル群に対してデータの物理構造に基いたアクセスを効率的に行うために古くから情報検索などで利用されてきた転置ファイルについてその構造、特徴を説明する。

- 転置ファイルの構造
- 転置ファイルの特徴

指導上の留意点

1. ファイル編成法は、計算機メーカを中心にした経験的な技法の集約である。この点で必ずしもデータ構造の表現法としては統一的に整理はされていない。この事を認識させ、2次記憶上で情報を表現する一つの方法として考えられるような視点を与えるべきである。
2. データ構造に関しては、プログラミングの章の内容と関連性を保てるように指導する必要がある。
3. 伝統的なファイルとその処理システムからデータベース・システムへの発展を理解

させる。同時に、データベース・システムが問題分野や利用環境によっては適当ではなく、従来のファイル処理が適している場合があることを認識させるべきである。

4. 教育上利用する計算機についてファイル処理のために提供されている機能を具体的に調べさせ、それにより具体的なプログラミング演習ができるように考慮すべきである。

参 考

1. ファイル処理の基本

1.1 キーワード

ファイル	file
レコード	record
キー	key
ブロック化	blocking
マスターファイル	master file
変動ファイル	transaction file
順次アクセス記憶装置	sequential access storage device
直接アクセス記憶装置	direct access storage device
データ構造	data structure
ファイル編成法	file organization method
アクセス法	access method
順アクセス	sequential access
乱アクセス	random access
基本アクセス法	basic access method
待機アクセス法	queued access method
緩衝法	buffering
ファイル・カタログ	catalogue of file

1.2 ファイルの基本概念

次のような事項により、ファイルとファイル処理の基本概念を理解させる。

(1) ファイルの構成

- ファイル
- レコード
- データ項目, データ・フィールド
- キー項目, キー

(2) レコードの形式

- 固定長レコード
- 不変長レコード
- 不定長レコード

について説明し、これらの特徴を理解させる。

(3) ブロック化

- 論理レコード(logical record)

- ブロック(block), 物理レコード(physical record)
- ブロック化(blocking), 非ブロッキング(deblocking)
- ブロック化係数(blocking factor)について説明し、ブロック化の意味を理解させる。

(4) ファイル処理

- マスター・ファイル(master file)
 - 変動ファイル(transaction file)
 - および
 - レコードの取出し
 - レコードの更新
 - レコードの追加
 - レコードの削除
- などの概念を説明し、伝統的なファイル

処理の形態を理解させる。

1.3 記憶媒体

ファイルに付して用いられる記憶媒体の種類、特性について整理し、知識を確認する。

(1) 順次アクセス記憶装置

一定の順序でデータの処理をするのに適した順次アクセス記憶装置の代表として磁気テープについて、以下のような点からその物理的特性および磁気テープ・ファイルについて理解させる。

- (a) 記録形式 (トラック)
- (b) データの記録方法
- (c) 記録密度
- (d) チェック・ビット
- (e) ブロック間隔 (IRG: inter-record gap)
- (f) 記憶容量
- (g) 転送速度
- (h) 磁気テープ・ファイル
 - ・ボリューム, ボリューム・ラベル
 - ・ファイル・ラベル
 - ・テープ・マーク
 - ・単一ボリューム・ファイル, 複数ボリューム・ファイル, 複数ファイル・ボリューム

(2) 直接アクセス記憶装置

データの順序に無関係にデータ処理が可能な直接アクセス記憶装置 (DASD)の代表として、磁気ディスク、磁気ドラムなどについて、以下のような点からその物理的特性、ファイルの形態について理解させる。

- (a) アクセス機構
- (b) トラック
- (c) シリンダ
- (d) レコード・アドレス
- (e) シーク動作とサーチ動作

- (f) 転送速度
- (g) 記憶容量
- (h) 磁気ディスク装置の種類
- (i) 磁気ディスク・ファイル
 - ・ボリューム目録 (VTOC)
 - ・エクステンツ
 - ・ボリューム・ラベル

1.4 ファイルの構成

第4節への導入として、ファイルの表現に用いられるデータ構造、ファイル編成法、アクセス法などについてその概略を説明し見通しを与える。

(1) データ構造

次のようなデータ構造とその表現について復習する。

- (a) 線形構造 (linear structure)
- (b) 木構造 (tree structure)
- (c) ネットワーク構造 (network structure)
- (d) 連結リスト (linked list) による表現

(2) ファイル編成法

ファイル処理に現われる代表的な構造とそのアクセス手段をオペレーティング・システムで支援するものとしてファイルの編成法がある。ここでは、標準化されたデータ構造としてのファイル編成法について概略を説明する。

- (a) 順編成 (sequential organization)
- (b) 索引順編成 (indexed sequential organization)
- (c) 直接編成 (direct organization)
- (d) 区分編成 (partitioned organization)

1.5 オペレーティング・システムとファイル処理

ファイル処理に関するオペレーティング・システムの役割、ファイル処理の為の命令について整理する。それぞれの教育環境で利用できるオペレーティング・システムにより具体的なマクロ命令やジョブ制御文に基いて説明する必要がある。

(1) アクセス法

次のような項目につき理解を整理する。

- (a) 順アクセス、乱アクセス
- (b) 基本アクセス法
- (c) 待機アクセス法
- (d) 緩衝域 (buffer)
- (e) 処理モード

(2) ファイル操作命令

次のような機能に対応する命令について説明する。データの動き、緩衝域や入出力域との関係について理解させる。

- (a) ファイルの open
- (b) ファイルの close
- (c) GET, PUT マクロ
- (d) READ, WRITE マクロ
- (e) 入出力操作の完了検査
- (f) 緩衝域の制御

(3) ファイルの割当て

ファイルの定義、ファイル領域の割当てのためのジョブ制御文、マクロ命令について整理する。

(4) ファイル・カタログ

オペレーティング・システムが利用するファイル管理の為の情報であるファイル・カタログについて、その構造と利用法、およびファイル呼出しの自動化、世代管理の自動化、機密保護、アクセス法のチェックなどの機能について説明する。

(5) ファイル処理に関するユティリティとライブラリ・ルーチン

具体的なオペレーティング・システムに基づいて、ファイル処理のために準備されているユティリティ・ルーチンやライブラリ・ルーチンについて調べる。

2. ファイル編成法

2.1 キーワード

順編成	sequential organization
キー	key
分類, 整列化	sorting
併合	merging
突き合せ	matching
索引つき順編成	indexed sequential organization
索引	index
あふれ領域	overflow area
再編成	reorganization
直接編成	direct organization
ハッシュ関数	hashing function
かち合い	collision
同義語	synonym
区分編成	partitioned organization
メンバ	member
ディレクトリ, 登録簿	directory

2.2 順編成ファイル

順編成ファイルの構造, 特徴, 処理法について理解させる。

(1) 順編成ファイルの構成

順編成ファイルではレコードは通常なんらかのキー(key) 値に関してある順序(昇順(ascending order), 降順(descending order))に並べられている。次の点について説明する。

- ・順編成ファイルの構造
- ・順編成ファイルを置く記憶装置

(2) 順編成ファイルの特徴

順編成ファイルの持つ次のような特徴を理解させる。

- ・アクセスが速い
- ・低価格ファイル媒体(磁気テープ)が利用できる
- ・記憶空間の利用に無駄がない
- ・順アクセスしかできないので, 任意のレコード呼出しはできず, レコードの

追加や削除ではファイル全体を書き直す必要がある。

- ・処理のためには一般に分類(整列化 sorting)が必要になる

(3) 順編成ファイルの処理

次の点に関して説明し, 順編成ファイルを扱う基本技法を理解させる。

- ファイル生成
- ファイルの更新
- ファイルの読み出し
- 分類
- 併合
- 突き合せ

2.3 索引つき順編成ファイル

索引つき順編成ファイルの構造, 特徴, 処理法について理解させる。

(1) 索引つき順編成ファイルの構成

基本的には順編成ファイルの構造であるが, それに何段階かの索引(index)を付加して, 順編成-順アクセスの長所

を残しつつ、レコードの直接も可能とすることを考えた古典的な索引づけの方法

である。次のような図により基本的な構造を理解させる。

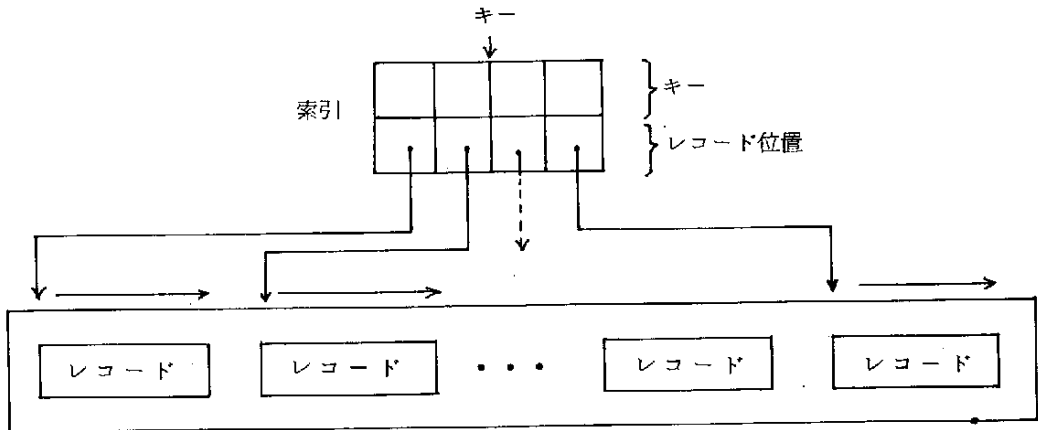


図 - 1 索引つき順編成

次の概念を説明する。

- (a) キー
- (b) 基本領域 (prime area)
- (c) 索引と索引領域

(2) 索引つき順編成ファイルの特徴

索引つき順編成ファイルの持つ次のような特徴を理解させる。

- レコードの順アクセスも乱アクセスも可能である
- 索引が主記憶にない場合には最低 2 回の入出力操作が必要である
- ファイルの途中でレコードの挿入や削除が可能
- 記憶領域の利用効率が低い
- あふれが多くなるに従ってアクセスが遅くなり、再編成 (reorganization) が必要になる

(3) 索引つき順編成ファイルの処理

次の点に関して説明し、索引つき順編成ファイルを扱う基本技法、問題点を理解させる。

- (a) ファイルの生成と索引の生成
- (b) ファイルの更新
- (c) ファイルの読み出し

- (d) 順アクセスと乱アクセスの組み合わせ

- (e) 多段階索引による入出力の効率化

- (f) ファイルの再編成 (ファイル生成のやり直し)

2.4 直接編成ファイル

直接編成ファイルの構造、特徴、処理法について理解させる。

(1) 直接編成ファイルの構成

直接編成は、レコード中のキーの値に一定の変換をほどこしてレコードの位置を計算する編成法であり、変換操作はハッシング (hushing) あるいはランダム化 (randamization)、変換に用いる計算法はハッシュ関数 (hushing function) と呼ばれる。次のような図により基本的な構造を理解させる。

次の概念を説明する

- (a) ハッシュ関数
- (b) かけ合い、同義語
- (c) バケツ (bucket)

(2) 直接編成ファイルの特徴

直接編成ファイルの持つ次のような特徴を理解させる。

- アクセスが速い（レコードの探索時間がファイルの大きさによらない）
- ファイルに内部構造を与えることはできない。
- 記憶領域の利用効率が低い
- 同義語の発生によりあふれ（overflow）が多くなるとアクセスは遅くなる
- ハッシュ関数を適切に選ぶことが難しい

(3) 直接編成ファイルの処理

次の点に関して説明し、直接編成ファイルを扱う基本技法、問題点を理解させる。

- ファイルの生成
- ファイルの更新
- ファイルの読み出し
- ハッシング法

次のように代表的なハッシング法（キー変換法）について説明する：

- 割り算法
- 掛け算法
- けた分析法
- 折りたたみ法

- 2乗法
- 基数変換法

(e) 同義語の解釈

同義語の発生は必然的であり、その解釈法を適當に選ぶことは直接編成で大きな問題である。次のような考え方を説明する：

- 開放アドレス方式（open addressing）
- あふれ領域（overflow area）を用いる方式

2.5 区分編成ファイル

区分編成ファイルの構造、特徴、処理法について理解させる。

(1) 区分編成ファイルの構成

区分編成ファイルは、順編成ファイルをメンバ（member）と呼ばれる部分に分割し、各メンバの開始位置を示すディレクトリ（登録簿 directory）を付加してメンバ毎の処理を容易にする方法である。

次の概念を説明する

- メンバ
- ディレクトリ

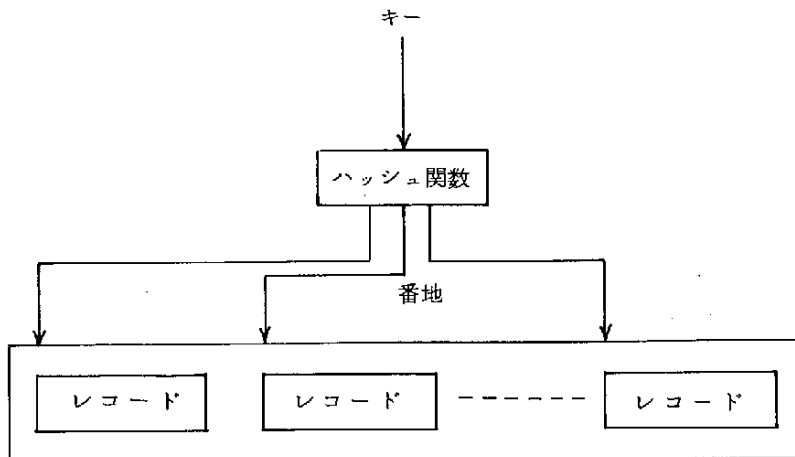


図 - 2 直接編成の機構

(c) ディレクトリとファイル本体の構造

(2) 区分編成ファイルの特徴

区分編成ファイルの持つ次のような特徴を理解させる。

- メンバ単位での処理が容易
- 再編成の必要性
- メンバ内での順アクセスとメンバ単位の乱アクセスが可能

区分編成ファイルは、その構造から、プログラム・ライブラリ・ファイル、カタログ・プロシージャ・ファイルなどオペレーティング・システムの用いるファイルで利用されることが多い。

(3) 区分編成ファイルの処理

次の点に関して説明し、区分編成ファイルを扱う基本技法、問題点を理解させる。

- (a) ファイルの生成
- (b) ファイルの更新
- (c) ファイルの読み出し
- (d) ファイル再編成 (reorganization), 圧縮 (compaction)

4. ファイル・システムのためのデータ構造

4.1 キーワード

レコード間関係	intra-record relationship
線形構造	linear structure
木構造	tree structure
ネットワーク構造	network structure
階層構造	hierarchical structure
データベース	data base
連結リスト, つなぎ並び	linked list
多重つなぎ	multi-linked list
2分木	binary tree
AVL-木	AVL-tree
木の均衡化	tree balancing
B-木	B-tree
B [*] -木	B [*] -tree
転置ファイル	inverted file
副索引	secondary index, alternate index

4.2 レコードの論理的構造

ファイル編成法は、物理的構造に重点を置いた大容量記憶装置を用いるための標準化された利用技術であった。ここでは、データの持つ相互の関連性を重視し、論理的構造を反映したファイル構成について説明する。

(1) レコード間関係

ファイル群に蓄積された種々のレコードから情報を適切に取り出すことを考える時、ファイルの物理的構造を離れ、応用プログラムや検索目的に適した論理的な構造を反映したファイル構成が望ましいことを理解させる。レコード間の論理的構造として次のような構造が考えられることを説明する。(図-3 参照)

(a) 線形構造

(b) 木構造

(c) ネットワーク(グラフ)構造

(d) 階層構造

(2) データベース

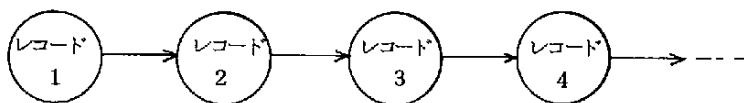
上のような論理的構造に基づく統合化されたファイル群としてデータベースが成立することを理解させる。

(3) 物理的表現

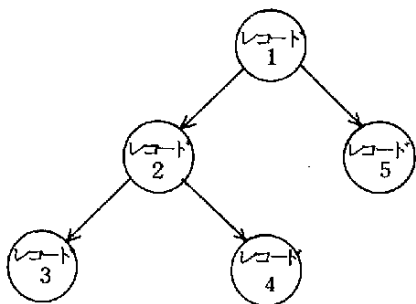
論理構造とそれを2次記憶媒体上で表現する物理的構造の区別を理解させる。木構造やネットワーク構造を連結リスト(linked list)で表現するなどは最も直観的に理解できる。また、木構造を基本的な論理構造とするデータベース・システムであるIMS(Information Management System)では、物理的表現として

- ・ 順編成
- ・ 索引つき順編成
- ・ 直接編成
- ・ 索引を持つ連結リスト表現

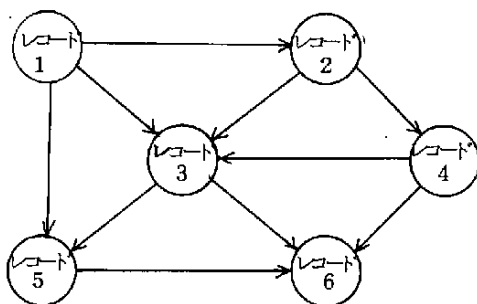
などが用いられることなどを例示するのもよいであろう。



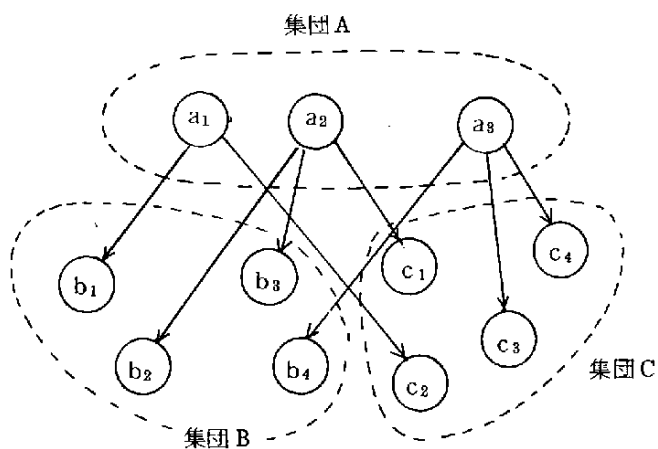
線形構造



木構造



ネットワーク構造



階層構造

図 - 3 各種の構造

4.3 連結リスト

レコードの関連性をポインタ(pointer)あるいはつなぎ(list)により陽に示す連結リスト(つなぎ並び)を用いたデータの編成法を理解させる。

(1) 連結リスト

レコードの物理的配置によりレコード間の関連性(順序)を示したのが順編成であるのに対して、レコード間の関係をポインタにより示すのが連結リストあるいは鎖(chain)である。次の概念により連結リストについて整理する。

(a) 連結リスト

(b) 循環連結リスト(circular-linked list)

(c) 連結リストに対する操作アルゴリズム:レコードの生成, 追加, 更新, 削除

(2) 多重つなぎ

レコード間に種々の論理的関連性を持たせたい場合の手段として、レコード中に複数個のポインタを持たせる多重つなぎ(multi-linked list)がある。多重つなぎによる複雑な構造の表現、多重つなぎの特徴について説明する。

4.4 木構造

ファイル構成に対する木構造の利用について理解させる。

(1) 木構造

レコードの論理的関係として最もよく利用されてきたのは木構造である。次のような概念により木構造による表現について理解させる。

(a) 木の定義

(b) 2分木(binary tree)

(c) AVL-木(AVL-tree)

(d) 木の巡回アルゴリズム

(e) 木の均衡化(balancing)の必要性

とそのアルゴリズム

(f) 部分木(subtree)をまとめて1つのブロック(ページ)中に配置するなど、2次記憶装置のアクセスを考慮した考え方

(2) B-木

木構造を基本とし、レコードの追加、削除にともなう動的に木の再編成を行うことによって、かなりのアクセス効率と記憶空間の利用効率を保證できるように工夫された方法としてB-木がある。IBM社のDSAMで利用され、また別の章で触れるように関係データベースでの索引構成によく利用され注目された方法である。次のような項につき説明する。

(a) B-木の定義

(b) B*-木の定義

(c) レコードの捜入、削除のアルゴリズムと木の動的再構成

4.5 転置ファイル

ファイル群に対して複数の索引を付して論理構造に応じたアクセスを効率化する方法として古くから利用されてきた方法に転置ファイルの方法がある。

ここでは転置ファイルの構造、特徴を理解させる。

(1) 転置ファイルの構造

次のような概念を説明し、転置ファイルの一般的構造を理解させる。

(a) 主キー(primary key)

(b) 副キー(secondary key, alternate key)

(c) 副索引(secondary index, alternate index)

転置ファイルは古くから文献検索などの情報検索の分野でよく利用されてきた方法である。転置ファイルの構成も、例えば、文献番号を主キーとし、他に著者

名、文献名、キーワードなどを副キーとする例などを例示するとよい。

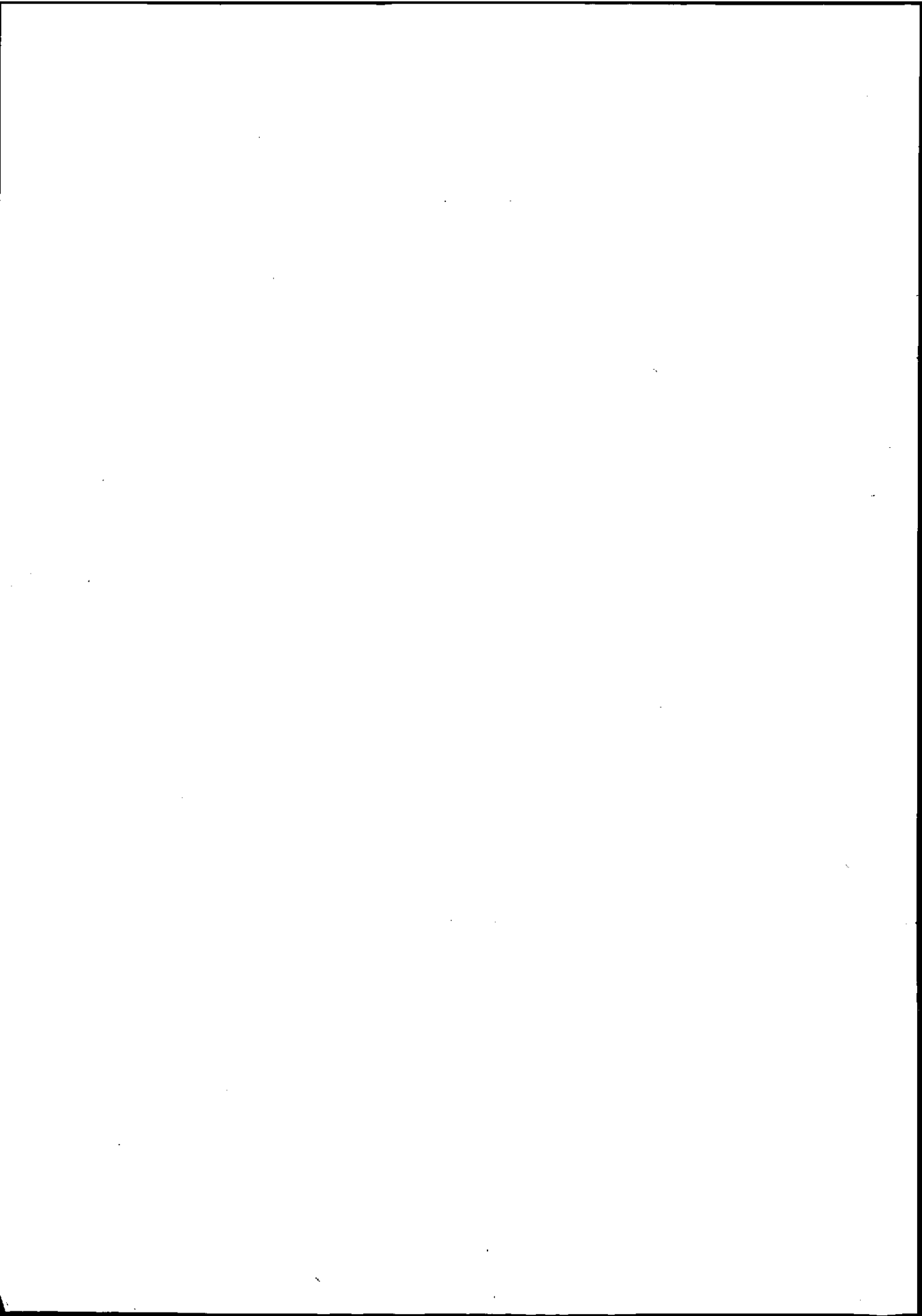
(2) 転置ファイルの特徴

次のような性質を説明し、転置ファイルによるファイル構成の特徴を理解させる。

- 複数のキーに基づいてデータをアクセス可能
- 検索が中心となる応用分野での効率が高い
- データの変動に対して常に索引を保守して行く必要がある
- ある種の論理的条件に対する検索では実際のファイル中のレコードのアクセスはせずに、索引の操作だけで処理可能である

参考文献

- (1) 浦昭二編「データ構造」共立出版, 1974
- (2) 情報処理学会編「情報処理ハンドブック」オーム社, 1980
- (3) Horowitz, E. and Sahni, S. : 「Fundamentals of Data Structures」Computer Science Press, 1976
- (4) S. E. Madnick and J. J. Donovan : 「Operating Systems」McGraw-Hill, 1974
- (5) J. J. Donovan : 「System Programming」McGraw-Hill, 1972
- (6) 井上謙蔵他「システム・プログラム」共立出版, 1972
- (7) D. R. ジャド, 戸川隼人訳「ファイル処理」サイエンス社, 1974



第7章 プログラミング (C2)

キーワード

制御構造	control structure
データ構造	data structure
プログラム言語	programming language
配 列	array
線型リスト	linear list
木	tree
整 列 化	sorting
探 索	searching
抽 象 化	abstraction
段階的詳細化	stepwise refinement
構造的プログラミング	structured programming
プログラムのスタイル	program style

目 標

1. プログラミングの基本的な知識と方法を修得させる。
2. 解り易く、正しいプログラムを作成することが第一の目的である。このために、プログラミングをどのような考え方で行うべきかを理解させ、その考え方を具体化する諸方法を身につけさせる。単に、種々の技法を知っているだけでは不十分であり、技法の背景にある考え方を理解させることが必要である。
3. プログラムの構造や性質、データの構造やその操作アルゴリズムの基本を身につけさせ、解くべき問題と置かれた環境にそくして適切なプログラム作成を行うことができるように、これらの基本概念や技法を道具として使いこなす態度を養う。

内 容

1. プログラムの構造

プログラミングで利用する基本的な構造を、制御構造 (control structure) と データ構造 (data structure) の観点から整理する。

(1) 制御構造

制御構造は、書かれたプログラムのアルゴリズムを表現する基本的な構造としての静的な (static) 構造と、個々のプログラムが実行される際の動的な (dynamic) 構造に分けられる。

静的な構造としては、

- 連接構造
- 選択構造
- 繰返し構造 (WHILE-DO 型と DO-UNTIL 型)

が基本的であることを理解させる。

動的な構造としては、

- プログラムのオーバレイ (overlay), 再配置可能プログラム (relocatable program)
- プログラム間のコミュニケーション

などについて理解させる。しかし、これ等の概念については「ソフトウェアの設計と開発」(第8章)で詳しく扱い、ここでは概略を示して見通しを与えられれば良い。

(2) データ構造

制御構造と補い合ってプログラムの論理を表現するために必要なデータ構造の考え方を理解させる。

データ構造とその処理方法は次節で詳しく扱い、ここでは展望を与えるにとどめる。

2. プログラム言語での表現

前節で整理したプログラムの構造が、実際のプログラム言語においてどのように表現されるかを復習しつつ、整理する。第4節や第8章で扱う解り易いプログラムの表現やプログラムのスタイルなどの考え方との関連に焦点を当て、FORTRAN, COBOL, PL/I, ALGOL, PASCAL, BASICなどの主な手続き的汎用言語での表現手段を理解させる。最近の動向として、FORTRAN や COBOL に対する構造的プログラミングやプログラム・スタイル論の影響、および設計時から構造的プログラムを意識した言語であるPASCALについても十分に触れるのが良い。

3. データ構造とアルゴリズム

プログラミングにおいて、プログラムの制御構造と同様に重要な要素となるのがデータ構造である。与えられた問題にそくし

た論理的なデータ構造の把握とそのプログラムあるいは計算機上の論理的表現という区別の下に、基本的なデータ構造やその操作アルゴリズムを利用する能力を養うことが必要である。

(1) 配列

最も基本的で利用度の高い構造である配列について修得させる。

- 配列の概念
- 配列の割り当て

(2) 線型リスト

一般に複数のデータがポインタな位置の順序関係で関連づけられた基本的構造として線型リストがある。線型リストのうち代表的な論理構造としてスタック (stack) と待ち行列 (queue) について説明し、一般の線型リストの表現法および操作法を理解させる。

- スタック
- 待ち行列
- 線型リストの表現
- 線型リストの操作

(3) 木構造

データ間の論理的関係を表現する方法としては最もよく用いられ、ファイル編成法やデータベースとも深い関連を持つ木構造の取り扱いの基本的事項を修得させる。

- 木の概要
- 木の基礎技術
- 木のたどり方
- 木の表現

(4) 整列化と探索

データ構造として表現された情報の集合に対して応用プログラムが必要とするアルゴリズムとして整列化と探索がある。ここでは、プログラミングの種々の状況における整列化や探索の必要性を理解さ

せ、それらのアルゴリズムや効率の問題を説明する。

- 整列化の必要性
- 整列化のアルゴリズム
- 探索のアルゴリズム

4. プログラムの作成

前節までで、プログラミングに利用できる基礎技術についての理解を得た。ここでは、これらの概念や技法を具体的プログラミングに利用して行く上での考え方や方法論を学ぶ。

ソフトウェア・システムの設計、モジュール化、プログラミングといった互に関連する場面において現在最も安定した考え方を提供できるのは構造的プログラミングにそったアプローチである。ここでは、このアプローチの周辺で利用できる方法論の幾つかを説明し、局面に応じてこれらの考え方をよいプログラムの作成の為に採用できる基盤を与える。

- 構造的プログラミングの概念
- 構造的プログラミングの方法
- プログラムのスタイル

指導上の留意点

1. 説明にあたっては、できるだけプログラムの具体例を多く示すようにする。ただし、あまり複雑な問題や大きなプログラムは避け、良いプログラミングの考え方の本質がかくされないように心がけるべきである。
2. プログラミングの基本的な態度としては、構造的プログラミングのアプローチに沿って段階的詳細化やプログラムのスタイルに大きな比重を置くのが良い。これらの方法論を場合に応じて選択し、使い分けられるような見識を養いたい。

補 足 説 明

1. データ構造とアルゴリズム

1.1 キーワード

配 列	array
表	table
多次元配列	multi-dimensional array
記憶変換関数	storage mapping function
割当て	allocation
疎な行列	sparse matrix
線型リスト	linear list
スタック	stack
待ち行列	queue
木	tree
部分木	subtree
有向木	oriented tree
2分木	binary tree
木の2分木表現	binary representation of tree
木のたどり方	traversal method
整列化	sorting
探 索	searching
内部整列化	internal sorting
挿入による整列化	insertion sort
交換による整列化	exchange sort
選択による整列化	selection sort
併合による整列化	merge sort
分配による整列化	distribution sort
逐次探索	sequential search
キー比較探索	searching by comparison of keys
2分探索	binary search
木探索	tree search
見出しつき探索	indexed search

1.2 配 列

最も基本的で利用度の高い構造として
配列（あるいは表）の概念を修得させる。

(1) 配列の概念

表や配列がどういうものであるのか
はあらためて説明する必要はないであ
ろう。ただし、1行あるいは1列だけ

からなる表や1次元配列が線型リストと
同じ構造であることは説明しておいた方
がよい。

さらに、このような表や配列と線型リ
ストとの性質の違いも次のように対照さ
せておく必要がある。

- 表や配列では、「一つ前の要素」も

すぐに得られるが、線型リストではそのためのポインタを特に持たせない限り得られない。

- 表や配列では、はじめから n 番目とか、ある要素から n 個前（あるいは後）とかも直ちにアクセスできるが、線型リストでは n 回ポインタをたどらなければならない。

- 線型リストでは、ポインタの分だけ余分の記憶域が必要となる。

- 表や配列では途中に要素を追加したり、途中の要素を削除したりすると、それ以降の要素をすべてずらさなければならないが、線型リストでは、ポインタのつなぎかえだけで実現できる。

(2) 配列の割当て

はじめに、記憶変換関数とか割当てとかの意味を説明する。

次に、いくつかの配列について記憶変換関数の形を具体的に示し、それを誘導して見せる。

(a) 1次元配列の要素 $a(i)$

$$\text{loc}(a(i)) = \text{loc}(a(1)) + (i-1)S$$

ここでは $\text{loc}(x)$ は x に対して割当てられるアドレス、 S は一つの要素に対して割当てられる語数である。

1次元配列では、添字の値の上限がわからなくても記憶変換関数を計算できることを注意しておく。

(b) 2次元配列の要素 $a(i, j)$

添字の上限をそれぞれ I, J とする。
2次元配列については、2通りの割当てが行なわれていること、それぞれ「行方向の割当て」、「列方向の割当て」と呼ばれていることをまず説明する。

- 行方向の割当て

$$\text{loc}(a(i, j)) = \text{loc}(a(1, 1)) + \{(i-1)J + j - 1\}S$$

- 列方向の割当て

$$\text{loc}(a(i, j)) = \text{loc}(a(1, 1)) + \{(i-1) + (j-1)I\}S$$

2次元配列の場合には、添字の上限 I または J がわからないと、記憶変換関数が計算できないことを説明する。

1.3 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り出すので、このような取り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、1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取り出しのしかた、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。

(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り出しを行なうので、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。

(3) 線型リストの表現

線型リストや一般のリスト構造を表現する場合に、図-1のように、項目(レコード)を箱で示し、その箱をデータ部とポインタ部に分け、ポインタは矢印で表現すること、および終端ポインタはで示す習慣が確立していることも説明する。さらに物理構造としては、ポインタとして0を使うことが多いということも簡単にふれておいた方がよいだろう。

次の項目をさすポインタつまり前進ポインタだけを持つ線型リストでは、「一つ前の項目」を知ることができないのでそのような項目を知りたい場合には、何らかの工夫が必要であることを説明する。具体的な工夫としては、

- 「一つ前の項目」をさす後進ポインタを各項目に持たせる。
- 先頭の項目をさすヘッド・ポインタを各項目に持たせる。

がただちに考えられる。これらの構造を説明する。

これらはいずれも各項目にポインタを二つずつ持たせる構造であるが、ポインタを一つだけにしておいて、しかも前の項目を得られるようにするため、終端ポインタでヘッドをさすようにした循環連結を説明する。

このとき、リストの先頭の項目を示すフラグを設けておいた方が便利であるこ

とを理解させる。

(4) 線型リストの操作

次の各種の線型リスト

- 単純連結の線型リスト
- 両方向リスト
- ヘッド・ポインタを持つ線型リスト
- 循環リスト

のそれぞれについて

- 新しい項目の追加
- 新しい項目の挿入
- 項目の削除
- 項目のアクセス

その項目があるかどうかわからない場合

その項目が存在することがあらかじめわかっている場合

の各作業を行なう場合に、どのポインタをどのようにたどればよいのか、また、どのポインタをどのように書き換えればよいのかを、理解させる。項目のアクセスについては、リスト内の各項目が順不同で連結されている場合、あるキーに従って分類されている場合、さらに、分類された上、目次(ディレクトリ)を備えている場合の3通りを説明する必要がある。

1.4 木構造

データ間の論理的関係を表現する構造としては最もよく用いられる木構造の利用技術の基本を修得させる。

(1) 木の概要

次のようなトピックスを題材にして、木構造の必要性を説明する。

(a) 鉄道網

一般の鉄道網で駅と駅のつながり具合を表現するには、分岐やループをあらわすことが必要である。

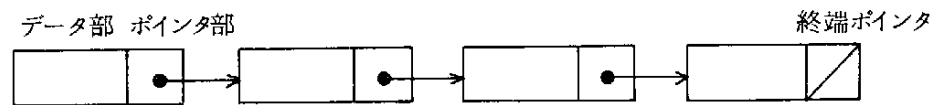


図-1 線型リストの図示

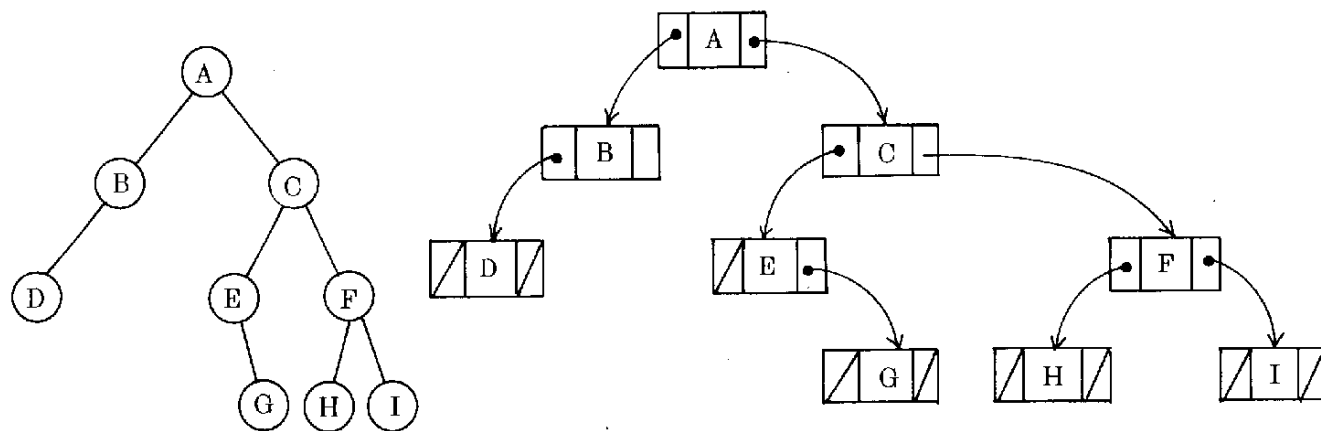


図-2 2進木の連結リスト表現

(b) 道路網

(c) 通信網

(d) ゲームの木

ゲームの局面の変化は木やネットワークとして表現できる。

(e) 流れ図

箱のつながり方が線型リストで表現できるような一本道の流れ図ではプログラムを作る意味がない。そこで分岐、ループが必要となる。

(f) 従業員の職制

(g) 系 図

(h) 本の目次

(2) 木の基礎技術

はじめに、前回示したトビックスについて、それを木やネットワークの形で書いたときに枝に方向がつくものと、そうする必要がないものがあることを示す。ここで、枝に方向がある木、すなわち有向木と、枝に方向のない木、すなわち木あるいは自由木とを導入する。多くの木構造が有向木として表現した方が便利であることを説明する。

有向木はたとえば次のように定義できる。

(a) はいってくる枝のない節がちょうど1つある。

(b) 節から出ていく枝の数についての制約はない。

(c) (a)の節を除いたすべての節には、はいる枝がちょうど1つある。

(d) ループがない。

次の話題は2分木である。2分木には、節にデータのついたデータつき2分木と節にデータのつかない通常の2分木とがある。2分木は一般の木との関連で重要である。一般の木がデータつき2分木で表現できることを、実例

を使い説明する。さらに左の枝で親、右の枝で子供たちをつないだ木を持つようにすると、同じ木を2分木としてもあらわせることを示す。

(3) 木のたどり方

次に3種類の木のたどり方を説明する。

すなわち

- 前置法
- 挿入法
- 後置法

(4) 木の表現

木 (tree) は非線型リストであるから、原則として、順次割当ではできない。一般に、木を連結リストとして表現し、そのまま連結割当てで記憶する。

木は2分木 (binary tree) に変換できるので、2分木が記憶の基本になる。2分木の記憶のための連結リスト表現例を図-2に示す。木登り (tree traverses) を高速化するためのつなぎ木 (threaded tree) 表現も図-2の記憶単位に1ビットのタグを加えるだけで記憶できる。

木を順次表現する方法がある。この方法には前置順次表現 (preorder sequential representation)、家系順次表現 (family-order sequential representation)、後置順次表現 (postorder sequential representation) がある。ポインタ、タグ、次数 (degree) を用いて木構造を表現する。この方法は2分木に限らず、一般の木を表現できる。この表現をそのまま反映させて木構造を記憶することができる。

また、木構造は結合行列 (connection matrix) で表現できる。このとき、結合行列はスパースになることが多いので、連結リスト表現し、連結割当てで記憶するとよい。

1.5 整列化と探索

整列化と探索の基本的アルゴリズムを理解させ、その特徴、効率についての知識を与える。

(1) 整列化の必要性

大量の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説明する。順次処理における、入力、処理および出力の各段階において、次の観点から整列化の必要性を確認させる。

- 入力段階…データの重複、欠落、順序のチェックなど
- 処理段階…ファイルの突き合わせを行なう際のレコードの順序
- 出力段階…要求された順序でのレコードの出力

整列化に関する基本的用語として次のものの意味を明確にする。

- ファイル、レコード、フィールド
- キー
- 照合順位
- 整列化と併合
- 昇順と降順
- 系列
- パス
- 内部分類と外部分類

(2) 整列化のアルゴリズム

整列化の対象となるデータ(レコード)の量が少なければ主記憶のなかで行なう内部分類だけですむし、データ量が多ければ、系列の作成のための内部分類と、系列の数を減らすための外部分類を組合せて用いる必要がある。

ここでは内部整列化のアルゴリズムについて説明する。代表的な内部整列化の方法は次のように分けられる。

- 挿入による整列化
- 交換による整列化

- 選択による整列化
- 併合による整列化
- 分配による整列化

それぞれの整列化の方法に対して2,3のアルゴリズムを取りあげ、例を示しながらパスの回数、比較・転送の回数、必要とする領域の大きさなどについて説明する。

(3) 探索のアルゴリズム

多数のレコードのなかから、目的とするキーの値あるいは特定のフィールドを持つレコードを探し出すことを探索という。ここでは種々の探索技法のアルゴリズムについて説明し、探索法の効率の評価法として照合回数およびレコードの変更、挿入および削除のしやすさなどについて理解させる。

(a) 逐次探索法 (sequential search)

対象となるレコードを先頭から順次調べてゆく方法である。レコードの順序がどうなっているも目的のレコードを挿することができる点が特色である。

(b) キー比較による探索法

最も代表的な方法が2分探索法(binary search)で、対象となるレコードをキーの順に分類して整列しておき、まずその中央のレコードを比較して目的とするレコードがファイルの前半にあるか後半にあるかを判断し、つぎにはそのどちらかのサブファイルに対して同様の手続を行なう方法を繰り返すものである。

そのほかの手法としてフィボナッチ数列を利用して2分探索と同様に探索を行なうフィボナッチ探索法(Fibonacci search)などがある。

(c) 木探索 (tree search)

2分探索法などのキー比較による探

索は、キーの順に整列された固定長の表に適している。レコードの追加、削除など表の構造を変える必要が頻発するような状況ではこの方法は表を再構成する費用の点で不適当である。したがって表自身を2分木(binary tree)の形で持つことにすれば表の変更の効率も、探索の効率も良くすることができる。この考えに基いているのが2分木探索法(binary tree search)である。この技法はコンパイラの記号表などに多く用いられている。

主記憶に木構造を持ち、この木を探索したり変更したりする場合には、効率を保証するために平衡木(balanced tree)とする方が良い。

多数のレコードが直接アクセス装置に置かれている場合には、2分木の構造ではアクセス回数が多くなりすぎる。このようなときに有効なものとして、多進木(multiway tree)とB-木(B-tree)がある。

(d) 見出し付き探索(indexed search)

キーを値として見ないで桁ごとに辞書式の見出しによって探索する方法である。

(e) ハッシュ法(hashing)

キーの値を引数として与えるとレコードの位置を返すハッシュ関数を用いて探索する方法である。

代表的なアルゴリズムについて、例えば照合関数に基いた効率の比較を行う。

2. プログラムの作成

2.1 キーワード

抽象化	abstraction
段階的詳細化	stepwise refinement
上昇型プログラミング	bottom-up programming
下降型プログラミング	top-down programming
構造的プログラミング	structured programming
プログラムのスタイル	program style

2.2 プログラミングの方法

プログラミングを行う上での基本的な考え方として、

- ・上昇型 (bottom-up) プログラミング
 - ・下降型 (top-down) プログラミング
- について理解させる。

上昇型プログラミングは、実際の計算機で直ちに実現可能なレベル、例えば利用するプログラム言語のレベルを基礎として、そこから手続きやモジュールを構成し、次にそれ等を用いて上のレベルの手続きやモジュールを構成する。この過程を繰り返すことにより、最も上のレベルとしての与えられた問題の表現に至る方法である。

下降型プログラミングは、逆に、与えられた問題のレベルを基礎とし、これをより具体的な副問題の組に分けて行く操作を繰り返すことにより、プログラム言語で表現可能なレベルに副問題を落して行きプログラムを構成する。

小さいプログラムであっても、プログラミングは高度に創造性を必要とする作業であり、いずれの考え方を採用にしても図式的、画一的に考えることはできないことを、演習を通して理解させる必要がある。

2.3 構造的プログラミングとプログラムのスタイル

(1) 構造的プログラミングの概念

構造的プログラミング (structured programming) の考え方と実現の形について理解させる。はじめに、構造的プログラミングが提案された背景を説明し、次に構造的プログラミングの形を、終りに構造的プログラミングの一応の定義を説明する。すなわち、次の各項を説明する。

- ・背景—プログラムの信頼性について
- ・背景—GOTO文無しについて
- ・GOTO文無しを実現するためのプログラムの基本構成要素
- ・抽象化 (abstraction) について
- ・構造的プログラミングの幾つかの定義

(2) 構造的プログラミングの方法

構造的プログラミングによるコード化の手法として、段階的詳細化 (stepwise refinement) 法を説明する。

構造的プログラミングが、E.W.ダイクストラの言う、プログラムの知力による扱いやすさ (intellectual manageability) を主体とするものであるならば、これを手法として実現したのが段階的詳細化 (stepwise refinement) 法である。

段階的詳細化は、プログラムの構造を

われわれ人間の概念の構成に合わせようとしたもので、抽象化(abstraction)と詳細化を繰り返して行く方法を基礎としたものである。

ここでは例題を挙げて、実際に段階的詳細化を行いながら解説する。

(3) プログラムのスタイル

プログラム構造の基本的な部分として、またはより具体的なテーマとして、プログラムのスタイルの良さについて理解させる。ここでは、①良いスタイルのための具体的な原則、②その原則に従がわないとどうなるか、③プログラムを改良する方法、などを、それぞれについて例を挙げながら説明する。原則や一般論だけを話すよりも、例題に基づいて進めた方が理解しやすく、また充実した内容となる。

良いスタイルのための基本的な原則を次に挙げる。

- コードの構造が明瞭で読みやすいこと。GOTO文の代りにIF-THEN-THEN-ELSE文やDO文、ブロック、サブルーチンを使い、またデータの表現にも注意深い配慮をすること。
- 不当な入力や、境界域にある入力も含めて、どのような入力に対しても正しく働くこと。
- 効率よりも明瞭さが優先すること。効率が悪ければ、アルゴリズムを変更する方がよい。
- プログラム自身が文書となるコードにすること。

参考文献

- (1) A.T. Berztiss, "Data Structures: Theory and Practice", Academic Press, New York, 1971
- (2) Dijkstra, E.W., "A Discipline of Programming", Prentice-Hall, 1976

- (3) D.L. Parnas, "A Technique for Software Module Specification with Examples", Communications of the vol. 15, pp. 330-336, 1972
- (4) ベルティス (Berztiss, A.T.), 古川康一他訳「データ構造 I, II」日本コンピュータ協会, 1974
- (5) 情報処理学会編「情報処理ハンドブック」オーム社, 1980
- (6) N. Wirth, 野下他訳「系統的プログラミング入門」近代科学社, 1978
- (7) 小林孝次郎「情報構造」サイエンス社, 1977
- (8) O.J. Dahl, E.D. Dijkstra, and C.A.R. Hoare, "Structured Programming", Academic Press, 1972
- (9) 野下他訳「構造化プログラミング」サイエンス社 1975
- (10) W.H. Burge, "Recursive Programming Techniques", Addison-Wesley pp. 46-50, 1975
- (11) Wirth, N., "On the Composition of Well-structured Programs", Computing Surveys, 1974
- (12) Wirth, N., "Program Development by Stepwise Refinement", CACM, 1971
- (13) Yourdon, E. and Constine, L.L., "Structured Design", Yourdon, 1975
- (14) Yeh, R.T. (Ed.), "Current Trends in Programming Methodology", Prentice-Hall, 1977
- (15) Myers, G.J., "Software Reliability: Principles and Practices", Wiley, 1976
- (16) Katzan, H., "Systems Design and Documentation: HIPO", Van Nostrand Reinhold, 1976
- (17) Weinberg, G.M.: "The Psychology of Computer Programming", Van Nostrand Reinhold, 1971
- (18) Kernighan, B.W. and Plauger, P.J., "Software Tools", Addison-Wesley, 1976
- (19) Jackson, M.A., "Principles of Program Design", Academic, 1975
- (20) N. Wirth, "Algorithms + Data Structures = Programs", Prentice-Hall, 1976
- (21) Myers, G.J., "Reliable Software through Composite Design", Mason/Charter, 1975
- (22) Kernighan, B.W. and Plauger, P.J.: "The Elements of Programming Style", McGraw-Hill, 1974
- (23) I. Nassi and B. Shneiderman, "Flowchart Techniques for Structured Programming",

- SIGPLAN Notices (ACM), vol. 8, pp. 12-26, 1973
- 24 E. Horowitz, S. Sahni, "Fundamentals of Data Structures", Computer Science Press, pp. 77-128, 140, 169, 182-197, 1976
 - 25 D.L. Parnas, "On the Criteria to be used in Decomposing System into Modules", CACM, vol. 15, pp. 1053-1058, 1972
 - 26 D.E. Knuth, "Structured Programming with Goto Statements", Computing Surveys, pp. 261-301, 1974
 - 27 D.L. Parnas, "Information Distribution Aspects of Design Methodology", Information Processing 71, North-Holland Publishing Co. pp. 339-344, 1972
 - 28 B. Liskov and S. Zilles, "Specification Techniques for Data Abstractions", Proceedings International Conference of Reliable Software, SIGPLAN Notices, vol. 10 pp. 72-87, 1975
 - 29 Brooks, F.P., "The Mythical Man-Month", Addison-Wesley, 1975
 - 30 B. Shneiderman, R. Mayer, D. McKey and P. Heller, "Experimental Investigations of the Utility of Detailed Flowcharts in Programming", Communications of the ACM, vol. 20, pp. 373-381, 1977
 - 31 A.V. Gelder, "Structured Programming in Cobol: An Approach for Application Programmers", CACM, vol. 20, pp. 2-12, 1977
 - 32 D.E. Knuth, "The Art of Computer Programming, Vol. 1, Fundamental Algorithms", Addison-Wesley, 1968
 - 33 D.E. Knuth, "The Art of Computer Programming Vol. 3," Sorting and Search, Addison-Wesley, 1973

第8章 ソフトウェアの設計と開発 (C14)

キーワード

オーバレイ構造	overley structure
連係編集プログラム	linkage editor
監視プログラム呼出し	supervisor call(SVC)
呼出し手順	calling sequence
インタプリタ	interpreter
再入可能	reentrant
再使用可能	reuseable
純 手 続	pure procedure
多重タスク	multi task
問題仕様書	problem definition specification
設計仕様書	program design specification
H I P O	hierarchy and input process output
構造化プログラミング	structured programming
構造化流れ図	structured flowchart
単体テスト	unit test
システム・テスト	system test
ダ ンプ	dump
スナップ・ショット	snap shot
トレース	trace
擬似コード	pseudocode

目 標

1. プログラムの実行時の構造として重要なオーバレイ構造とプログラム再配置の概念を理解させる。
2. プログラム連結に関する技法と連係編集プログラムの役割を理解させる。
3. プログラム間及びプログラム内のモジュール間のデータの受け渡しと制御の受け渡しの仕方を理解させ、制御の受け渡しに用いる命令を正しく使用できるようにする。
4. プログラム・コードの共用の仕方を、単
純な一括処理、多重プログラミング、多重タスクの場合について、比較して理解させる。
5. プログラムの文書化の重要性を認識させる。問題仕様書や設計仕様書だけでなく、原始プログラム・リストも立派な文書になり得ること、文書はわかりやすく書かれなければならないことを理解させる。
6. 伝統的な文書化の道具である流れ図と決定表を見直し、新しい道具であるH I P O

図と構造化流れ図の意義と使い方を理解させる。

7. 文書の自動化について理解させる。
8. 標準的なテスト手順を理解させ、さらに、ボトムアップ型とトップダウン型の違いをよく理解させる。
9. テスト・ケースの選択の仕方とテスト・データの作り方を理解させる。
10. デバッグの技法と代表的なデバッグ・エイドを理解させる。
11. プログラムのモジュール化の方法と、代表的なプログラム開発技法であるトップダウン方式を身につけさせる。擬似言語の用い方を理解させる。

内 容

1. プログラムの実行時の構造と連絡

(1) プログラムの実行時の構造

単純構造とオーバーレイ構造を詳しく説明する。プログラム再配置の概念を理解させ、動的再配置の仕組を説明する。ハードウェアの機能とソフトウェアの機能を関連づけて説明する。

(2) プログラムの実行時の連絡

単純なプログラム連結と動的なプログラム連結の方法を説明する。関係編集プログラムを紹介し、目的モジュールとロード・モジュールの違いを理解させる。監視プログラム呼出しと割込み動作を説明する。

コールチェーンの概念と仕組を説明する。

2. プログラムのコミュニケーション

(1) プログラム間のコミュニケーションと結合

プログラム間のデータの受け渡しについて、パラメタによる方法、共通領域を使用する方法、外部記号を用いる方法、

レジスタによる方法、ファイルを媒介とする方法を説明する。

プログラム間及びプログラム内のモジュール間の呼び出し手順（制御の受け渡しの方法）を説明する。分岐命令、PERFORM命令（COBOL）、CALL命令などの用い方を説明する。多重プログラミングにおける呼び出し手順を例をあげて説明する。

アセンブリ言語を用いてレジスタの扱い方を詳しく説明する。

関係編集プログラムの機能を詳しく説明し、関係編集プログラムに対する入力と制御文の用い方を理解させる。

(2) プログラム・コードの共用

最初に単一処理内のプログラム・コードの共用を説明する。単純な共有方式の仕組をよく理解させる。プログラムとデータの分離、グローバルなデータとローカルなデータの概念を説明する。

再帰の仕組を説明する。インタプリタとテーブル駆動型プログラムを説明する。

次に、複数処理間のプログラム・コードの共用を説明する。多重プログラミングを復習し、再入可能、再使用可能の概念を説明する。再入可能と関連して、純手続の使い方、コーディングを説明する。

(3) 並行処理の同期制御と排他制御

多重タスクの概念、多重タスク機能を用いる場合の資源の共用と割当て、事象の同期を説明する。単一タスクとの違いを十分に理解させる。

3. プログラムの文書化

(1) 文書化の重要性

文書化の意義と重要性を説明する。文書化は、プログラミングの手段であり、文書化の文法は、時代と共に変化していることを理解させる。

(2) 文書の標準化

問題仕様書、設計仕様書、原始プログラム・リスト、試験仕様書、利用者マニュアルを取りあげ、その意義、記述内容、形式や用字用語の統一について説明する。読みやすくわかりやすい文書を書く重要性を理解させる。

(3) 文書の管理

プログラム開発の初期から完成後までの全過程における文書の管理と保守の方法を具体例を示して説明する。

(4) 流れ図と決定表

流れ図と決定表の長所、短所を説明し、わかりやすく表現する方法を考える。

(5) HIPO図と構造化流れ図

HIPOの図式目次、概括図、詳細図の書き方を説明する。構造化プログラミング用に開発された構造化流れ図の書き方を説明する。トップダウン型のプログラム設計の仕方をあわせて説明する。

(6) 文書の自動化

文書の自動作成、文書からのプログラムの自動作成について、例をあげて説明する。

4. プログラム・テスト

(1) テストの概要

テストの目的と方法、テスト手順を理解させる。単体テスト、システム・テスト、受入れテスト、設置テストの目的を説明する。

(2) テストの方式

ボトムアップ（上昇型）テスト、トップダウン（下降型）テスト、両方向テストを説明する。ボトムアップ型とトップダウン型のテスト技法の違いをよく理解させる。

(3) テスト・ケースの選択

全数テスト、全経路テスト、入力値

の範囲をクラス分けする方法、機能分析方法を説明する。テスト・データの作り方を説明する。

(4) デバッグの技法

コーディング時の配慮、机上デバッグ、デバッグ・エイドの利用などについて説明する。

(5) デバッグ・エイド

ダンプとスナップ・ショット、トレースの機能と使い方、デバッグ・モード、オンライン、デバッグ機能、テスト、データの自動作成などを説明する。

5. プログラムの設計

(1) プログラムのモジュール化

モジュール化の概念を説明し、モジュール化の利点と重要性を理解させる。

(2) モジュール化の技法

トップダウン方式とボトムアップ方式による方法を説明し、トップダウン方式が優れていることを理解させる。トップダウン方式の特徴を具体的に説明する。モジュールの強度と結合度の概念を説明する。

(3) プログラム開発技法

トップダウン方式による方式、改良法による方式、擬似言語の活用の仕方を説明する。構造化プログラミングの技法を説明する。データの抽象化について説明する。

(4) プログラミング・スタイル

スタイルの良いプログラムとはなにかを理解させる。

指導上の留意点

1. ここでは、プログラム設計の標準的な手順や技法を習得させ、プログラム設計に使われる道具を使いこなせるように指導する

ことが重要である。

2. 最初にオペレーティング・システムが制御できる標準的なプログラムの構造をよく理解させる。連係編集プログラムの機能を正しく理解させることも重要である。
3. プログラム連結については、制御の受け渡しとインタフェースの規則をよく理解させる必要がある。
4. プログラムの文書化については、単に知識を身につけさせるだけでは意味がない。実際に良い文書化ができる能力を身につけさせることが大切である。良い文書とはどんなものか、プログラミングにとって文書化がなぜ重要かをよく理解させ、演習指導などを通して、文書化の能力を育てるようになる。
5. 流れ図と決定表について長所と短所を復習し、流れ図と対比しながら新しい文書化の技法である、HIPO図と構造化流れ図を紹介する。HIPO図と構造化流れ図は、構造化プログラミング用に開発された文書化の技法であるが、応用分野を広く考えることができるので、必ずしも構造化プログラミングに限定して使い方の説明をする必要はない。
6. プログラム設計については、プログラムのモジュール化とトップダウンの方法を使いこなせるように指導する必要がある。擬似言語については、擬似言語の定義と必要性を明確にし、利用者がその文法や用語を自由に定義して使用できる言語であることをわからせる。
7. このコースは、「ファイル・システム」を学んだ人を主たる対象者としているが、「オペレーティング・システムとアーキテクチャ」の次に学ぶこともできる。

参 考

1. 用語の説明

オーバーレイ

計算機プログラムにおいて、内部記憶中に永続的に保持されることのない区分。

連係編集プログラム

目的モジュール間の相互参照を解決し、可能ならば構成要素を再配置することによって、一つ以上の別々に翻訳された目的モジュール又はロードモジュールから、一つのロードモジュールを作成するために用いられる計算機プログラム

監視プログラム

通常はオペレーティングシステムの一部であって、他の計算機プログラムの実行を制御し、データ処理システムの作業の流れを統制する計算機プログラム。このプログラムを呼び出すための特別な命令を監視プログラム呼出し命令（SVC 命令）という。

呼出し手順

正式には呼出し系列という。ある場合にはデータをも含む命令の編成であって、呼出しを行うために必要なもの。

インタプリタ

解釈プログラムともいう。解釈するために使われる計算機プログラム。

再入可能プログラム〔ルーチン〕

計算機プログラム〔ルーチン〕の一つで、繰り返し実行開始ができ、かつ、その計算機プログラム〔ルーチン〕の先行する実行が完了する前に新たに実行開始できるもので、しかも外部プログラムパラメタやいかなる命令もその実行中に修飾されないという条件を満たしているもの。

再入可能プログラム〔ルーチン〕は、同時に二つ以上の計算機プログラムで使うこ

とができる。

再使用可能プログラム〔ルーチン〕

計算機プログラム〔ルーチン〕の一つで、一度格納されて繰り返し実行できるもので、しかも実行中に変更されたいかなる命令も初期状態に戻され、外部プログラムパラメータは変更されずに保持されるという条件を満たしているもの。

純手続

プログラムの実行の際にプログラムコード自身を変更することがないプログラムまたはルーチン

多重タスキング

二つ以上のタスクの同時遂行又は交互配置による実行の機能を備えた操作機式

問題仕様書

利用者の要求している問題を分析し、正確に仕様として表現したもの。この仕様書に示された内容が、これから開発するプログラムのすべてを決める基になる。

設計仕様書

プログラムの構造を決める仕様書で、問題仕様書に示された問題の解決法を決め、プログラム設計の基本方針を示す。

H I P O

プログラムの仕様をアルゴリズム中心ではなく、機能中心に記述するために開発された文書化の技法で、図式目次を入出力処理図から成る。

構造化流れ図

構造化プログラミングのための流れ図として考えられたもので、構造化プログラミングの三つの論理構造——連続構造、選択構造、繰返し構造と、その構造化を、モジュール全体の構造の中に位置づけて表現できるようにした図。

単体テスト

環境からは独立に、一つのモジュールを

単位として、そのモジュールに与えられた機能をテストする。

システム・テスト

システムに要求された機能が満たされているかどうかをテストする。システムを外部から見てテストするテストである。

ダンプ

特定の目的のために、例えば記憶装置の他への転用、障害もしくは誤りに対する防護手段として、又はデバッグとの関連において、通常は内部記憶から外部媒体上へ、記憶装置又はその一部の内容を書き出すことをダンプするといひ、ダンプされたデータのことをダンプという。

ダンプするユーティリティールーチンのことをダンプということもある。

スナップショット

速写ともいふ。選ばれた命令又は選ばれた条件に対してだけ、出力データを出り出す追跡プログラム。

一つ以上の指定された記憶域の内容の動的ダンプをスナップショットダンプという、トレース

正式には追跡プログラムという。命令が実行された順序及び通常はそれが実行された結果を提示することによって、別の計算機プログラムの検査を行う計算機プログラム。

擬似コード

実行に先立って翻訳を必要とするコード

2. プログラムの再配置と連結

(1) プログラムの再配置

コンパイルされたプログラムは、連係編集プログラムによって編集されてロード・モジュールになるが、実行時にロード・モジュールを主記憶にローディングする位置は、毎回変わるはずである。そこで、ローディングの際にプログラムの

格納場所を任意のアドレスから開始できるように工夫する必要がある。このようにプログラムの格納位置をローディング時に変更できる能力を再配置可能性という。

ハードウェアの機能によって、再配置可能プログラムの実現方式はいろいろあるが、代表的な方法は、次の2つである。

- (a) ベース・レジスタを用いる方式
- (b) プログラム・カウンタによりアドレスの修飾を行う方式

プログラムの再配置をプログラムのローディング時だけでなく、プログラムの実行時にも行うことができればさらに便利である。

プログラムを実行時に一時中断し、別の場所に格納しなおしても動作可能な機能を動的再配置という。

動的再配置には次のような利点がある。

- (a) ロールアウトされたプログラムを再びロールインするとき任意のアドレスに置くことができる。

- (b) 主記憶の割当てに無駄が生じたときすでにローディングされているプログラムを詰めて、連続した空き領域を作り出すことができる。

動的再配置を実現するには、ハードウェアとソフトウェアの役割が必要である。

- (a) ハードウェアの役割

ベース・レジスタやプログラム・カウンタの役割である。

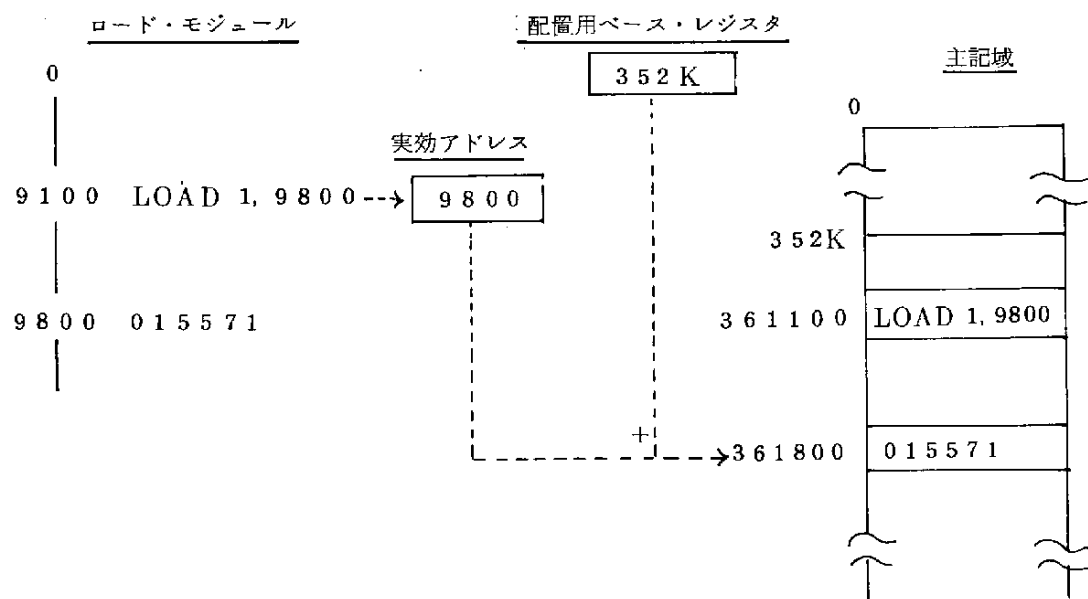
- (b) ソフトウェアの役割

動的再配置を可能にするにはオペレーティング・システムとの緊密な連携動作が必要である。

ベース・レジスタを用いる方法で、動的再配置のメカニズムを示す次のようになる。

- (2) プログラム実行時の連結

プログラムとプログラムを連結する最も一般的な方法は、連係編集プログラムによって、実行に先立ってアドレス空間に命令やデータを割り当てる方式である。



しかし、連結すべきプログラムが実行時まで決定できない場合や、プログラムの数が非常に多い場合は、この方式は不都合である。

そこで、動的な連結の技法が必要になる。例えば、次のような場合に、動的な連結を行うことになる。

- (a) ユーザ・プログラムが任意のプログラムを呼び出して実行したい場合

ユーザが必要とするプログラム（例えば誤りが発生したときの誤り回復プログラム）は、あらかじめ予想がつかないから、その必要性が生じたときに、ユーザ・プログラムは、動的連結機能によって、必要なプログラムを実行可能にし、それと連結することになる。

- (b) 必要なサブルーチンなどの数が多すぎる場合

多数のサブルーチンを使用するような大きなプログラムの場合、関係編集プログラムを用いたプログラムの連結処理は、指定方法が複雑であるだけでなく、処理自身にも多くの時間を必要とする。

動的な連結は、巨大なシステムの編集を容易にするのに役立つ。

プログラムの動的連結を可能にするために、オペレーティング・システムでは、システム制御のマクロ命令の一つとして、LOAD, LINK, XCTL, DELETEなどのマクロ命令を用意していることが多い。

- (a) LOADマクロ

必要なモジュールを外部記憶のライブラリから見つけ出してローディングし、呼び出したモジュールと動的に連結する。LOADしたモジュールに制御を渡したいときは、CALLマクロなど通常の制御受け渡しの命令を用いる。

- (b) LINKマクロ

LOADマクロとCALLマクロの機能をあわせ持つ。

- (c) XCTLマクロ

LINKマクロと似ているが、XCTLマクロによって呼び出されて動的に連結されたモジュールは、呼び出したモジュールと一体になり、主モジュール副モジュールの関係は生じない。

- (d) DELETEマクロ

LOADマクロで呼び出されて主記憶にロードされたモジュールの主記憶の領域を解放する。

- (3) 監視プログラム呼出し

監視プログラム呼出し(SVC)も動的連結の一つである。一般のプログラムの中で、SVC命令を実行すると、SVCで指定したコードによって定まる監視プログラム内の特定の対応処理ルーチンに制御が渡る。

次のような手順がとられる。

- (a) SVC命令が実行されるとSVC割込みが発生し、SVCハンドラに制御が渡る。

- (b) SVCハンドラは、SVCコードを解析し、レジスタなどの退避を行って対応するモジュールに制御を渡す。

- (c) 処理が終了したあとは、通常の割込み処理と同様にレジスタなどの回復を行ってSVCを実行したプログラムに制御を戻す。

- (d) 一般のプログラムと監視プログラムの間で制御の受け渡しが行われるときは、監視プログラム状態にあるかどうか監視プログラムによって容易に判別がつくようにシステムの状態の切り替えが行われる。

(4) コルーチン

コルーチンとは、個々のプログラムのどれが主ルーチンでどれがサブルーチンというように呼出しについての主従関係が存在しなくて互いに相手を自分のサブルーチンであるように呼び出すことのできる制御方法を備えたプログラム群である。

一般のサブルーチンと異なるのは、コルーチンへの入口点は、コルーチン呼出しの際に明示しない点である。各コルーチンごとに、次回に呼び出されるときはどこから処理を始めるべきかという活動記録を持ち、その情報に基づいて処理が実行される。

コルーチンを正しく働かせるには三つの内部操作が必要である。

(a) 初期設定

コルーチンごとに活動記録を作り、適切な情報を与える。

(b) 起 動

コルーチン名を指定し活動記録に従って制御を渡す。

(c) 後始末

最初に作られたコルーチンの活動記録、派生したプログラムとその活動記録を削除する。

サブルーチンの起動とコルーチンの起動には、活動記録の扱い方に差がある。サブルーチンの場合は、起動ごとに実行開始アドレスが指定されるから毎回活動記録の内容が新規に作られる。コルーチンでは、一定初期設定した後は呼び出されるコルーチンの活動記録の内容が呼出し例によって変更されることはない。

コルーチンの考え方は、FORTRANの入出力文の実行時処理プロセッサやシミュレーション言語などに有効に使用さ

れている。

3. プログラム間のコミュニケーション

サブルーチンの基本的な構造は次のようになっている。

(a) 呼び出されるプログラム

実行時に制御を受けるプログラムを呼び出されるプログラムという。オペレーティング・システムのもとでは、制御プログラムの制御を受けながら処理プログラムが実行されるが、処理プログラムは制御プログラムから制御を受けて処理を行うことになるので、制御プログラムに対しては呼び出されるプログラムである。

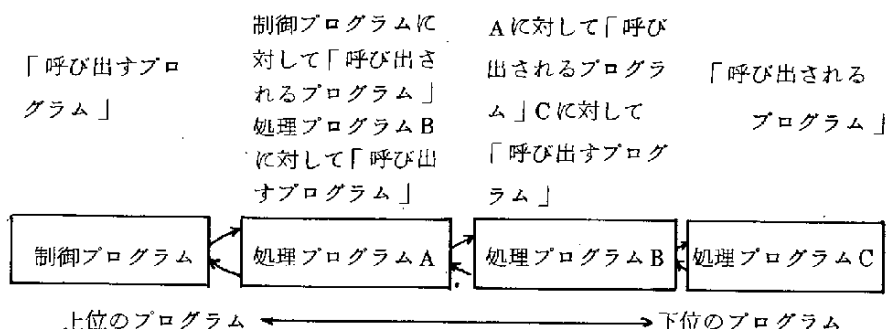
(b) 呼び出すプログラム

呼び出されるプログラムに制御を渡すプログラムを呼び出すプログラムという。制御プログラムは呼び出すプログラムである。

二つの処理プログラムA、Bがあり、処理プログラムAを実行しているとき、その処理プログラム中でさらに別の処理プログラムBに制御を渡して処理を行うことがあるとすれば、処理プログラムAが呼び出すプログラム、処理プログラムBが呼び出されるプログラムになる。

(c) 呼び出すプログラムと呼び出されるプログラム

呼び出すプログラムと呼び出されるプログラムの関係を示すと次のようになる。



(d) プログラムの連結

プログラムを呼び出し、呼び出されるプログラムに制御を移すと、それによって二つのプログラムが結びつくので、これをプログラムの連結という。プログラムの連結を行うには、プログラムの構造が一定の決まった形式になっていなければならないし、規則にかなったプログラムの呼出し手順を実行しなければならない。

(e) レジスタの働き

プログラムの呼出しを行うときは、レジスタを使うことが多い。例えば、次のようなAからEまでの五つのレジスタが、プログラムの呼出しのために使われる。このレジスタを連結レジスタと呼ぶことがある。

レジスタA 呼び出されるプログラムに渡すパラメタを入れる。

レジスタB ①レジスタAと組にして使い、呼び出されるプログラムに渡すパラメタを入れる。

②呼び出されるプログラムに渡すパラメタを記憶してある領域のアドレスを入れる。

レジスタC 呼び出すプログラムの持っている保管域のアドレスを入れる。

レジスタD 呼び出すプログラムの戻り点を入れる。

レジスタE 呼び出されるプログラムの入口点を入れる。呼び

出されるプログラムから呼び出すプログラムに戻るときは、戻りコードを入れる。

(f) レジスタの保管

プログラムを作成するときに、呼び出すプログラムと呼び出されるプログラムの関係を、いちいち考えなければならないとしたら大変わずらわしい。少なくとも互いに自由に、レジスタや領域を使用できるようになっていなければならない。

そこで呼び出されるプログラムの最初で、まずレジスタの内容を保管するという処理をしておく。呼び出されるプログラムで保管するレジスタに含まれている内容は、呼び出すプログラムで使用したものであるため、レジスタの内容を保管するための保管域は、呼び出すプログラムの中にとっておく。

レジスタの内容を保管することについての呼び出すプログラムと、呼び出されるプログラムとの役割の分担は、次のようになる。

呼び出すプログラム	呼び出されるプログラム
(a)保管域をとる	(a)自分より下位のプログラムがあって、自分の中に保管域がとってあるときには、保管域の先頭アドレスを上

	位のプログラム（呼び出すプログラム）の保管域に保管する。
(b)保管域の先頭のアドレスをレジスタCに入れる。	(b)レジスタの内容を呼び出すプログラムの保管域に保管する。

次の表は、保管域の1例である。保管域の形式は、その他にもいろいろ考えられる。

語の番号	説 明
1.	呼び出すプログラムの持っている保管域のアドレスを入れる。
2.	呼び出されるプログラムの持っている保管域のアドレスを入れる。
3.	レジスタDの内容（戻り点）を入れる。
4.	レジスタEの内容（入口点）を入れる。
5.	レジスタAの内容を入れる。
6.	レジスタBの内容を入れる。
7.	その他のレジスタの内容を入れる。
8.	〃

(g) プログラムの呼出し

プログラムを呼出してプログラムを連結するには、呼び出すプログラムで次のような処理が必要である。

- 呼び出されるプログラムの入口点（通常呼び出されるプログラムのプログラム名に一致している）をレジスタEに入れる。
- 呼び出されるプログラムが実行されて、呼び出すプログラムに戻るときの戻り点をレジスタDに入れる。
- 呼び出されるプログラムに渡すパラメ

タをレジスタA, Bに入れる。パラメタの数が多いたときは、パラメタを呼び出すプログラムの記憶域に入れておいて、その先頭アドレスをレジスタBに入れる。

- レジスタEを使って呼び出されるプログラムに制御を渡す。

(h) CALL命令

上記の処理を行わせるために、通常CALLというマクロ命令が用意されている。

CALLマクロを用いると以上のような処理を一つの命令で記述することができる。

(i) プログラムの戻り

呼び出されるプログラムでは、すでに述べたように、呼び出すプログラムから制御を受けたら、最初にレジスタの内容を保管する。

呼び出されるプログラムの実行が終り、呼び出すプログラムに制御を戻すときは、次のような処理を行う。

- 戻り点を保管管域から取り出してレジスタDに入れる。
- 戻りコードをレジスタEに入れる。
- 保管域に保管しておいたレジスタの内容を元に戻す。
- レジスタDを使って呼び出したプログラムに制御を渡す。

戻りコードは、呼び出されたプログラムの実行が正常に行われたかどうかを、呼び出すプログラムに伝えるために使われる。戻りコード自体は自由に決めることができる。これによって、上位のプログラムに実行が正常に行われたかどうかを知らせることができる。

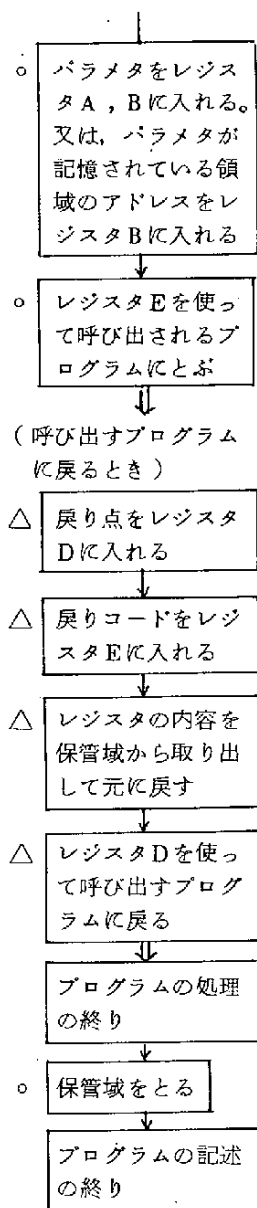
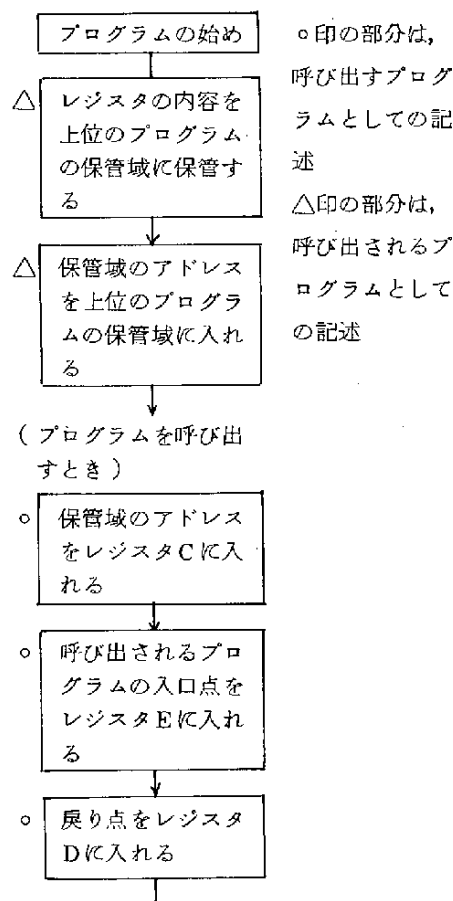
戻りコードをうまく利用すると誤りの程度がわかるので、上位のプログラムをそのまま実行できるかどうか判断し、誤

りがあったときの措置の仕方を決めることができる。

(j) プログラムの構造

呼び出すプログラムは、上位のプログラムに対しては呼び出されるプログラムになるし、呼び出されるプログラムは自分より下位のプログラムがあるときは、それに対しては呼び出すプログラムになるので、1単位のプログラムは、呼び出すプログラムと呼び出されるプログラムの両方の記述を含んでいなければならない。

したがって、一般的なプログラムの構造は次のようになる。



4. 構造化プログラミング

これからのプログラミングの方向をつかむ上で、構造化プログラミングを正しく理解しておくことは重要なことである。

構造化プログラミングの概念は、次のような背景のもとで生まれた。

- ハードウェアの大型化、高性能化に伴なりプログラムの規模の大型化と複雑化から、プログラム開発における人件費、開発過程での信頼性の問題が及ぼす費用

への影響が増加した。

- プログラムが増え、プログラムの寿命が伸びるにつれて、保守や改造の費用が増加した。
- 社会環境の中でシステムの占める役割が増加し、プログラムの誤りによるシステムの停止や誤動作が与える影響が深刻になった。

このようなことから、信頼性が高く、単純でわかりやすいプログラムの作成が望まれることになったのである。

プログラムの構造を単純にするためには、GOTO文の使い方を制限し、スパゲッティのように複雑に入り組んだプログラム論理を作らないことが重要である。

構造化プログラミングでは、次のような点が重要である。

(a) 下降型(トップダウン)の思考をする。

下降型の思考とは、1枚の紙を切りきざんでいく方式に似ている。

例えば、どんな紙でもよいから、1枚の紙を用意してその紙を三つに切る。それから最初の1枚を四つに切り、2番目のを二つに切り、残りを六つに切る。次に六つに切ったそれぞれを二つに切る。そして、切った紙を元の形に戻す。

この例には、重要な四つの要素が含まれている。

第1に元の紙を切り分ける過程で全体の大きさは、増えることも減ることもない。

これを機能に置きかえてみると、ある機能をいくつかの小さな機能に分けたとき、小さな機能を集めると元の機能になる。小さな機能を更にいくつかの小さな機能に分けた場合も同じ事が起こる。機能を合計しても、増減は起こらない。

第2に、部分に分ける方法は全く自由

である。あるものは大きく、あるものは不整形で、あるものは美しく、…どのような形にするかは、すべて設計者の創造的才能や美的センス次第である。

第3に、1度境界を決めたら、後からは変えられない。境界は神聖なもので、動かせない、1度切ってしまうと2度とつなぎあわせることはできない。また新しい境界線を決めるには、今までの境界線を考慮しなければならない。

第4に、機能の分解はどこまでも続けられる。特に制限はなく、1回で止めることも、100回分解することもできる。どこで止めるかは設計者の判断である。

(b) 機能中心に考える

機能中心とは、プログラミングにおいて論理や処理手順すなわちアルゴリズムをできるだけ考えないようにするということである。

まず全体的な機能を大まかに考え、次に、それを複数の機能に分解し、小さい単位の機能を明確にする。さらにそれを分解する。

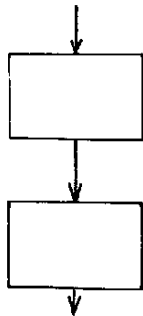
このような段階的な細分化を繰り返すことによってプログラムを作ることができると考える。

このような作業の形態をとっても、なお論理を組み立てるという要素が一部分残るが、従来のプログラミングとは非常に違ったものになるはずである。

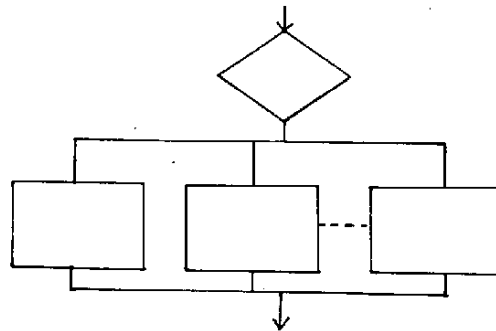
(c) プログラムは、三つの基本的な構成要素によって構成される。

三つの基本的な構成とは次のようなものである。

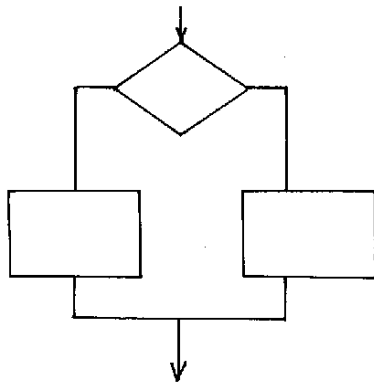
連続的な処理



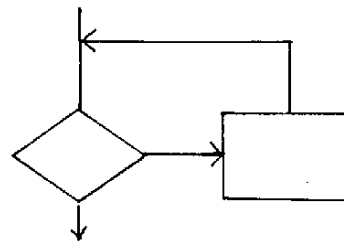
多数選択の処理



選択的な処理

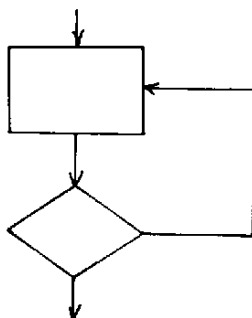


繰返しの処理



柔軟性を持たせるために、さらに次の二つの要素を加えることもできる。

繰返しの処理
(DOUNTIL型)



(d) プログラムのスタイルを良くする

プログラムをわかりやすくする上で、プログラムのスタイルを良くすることが重要である。

よいスタイルとは、

- 表現（特にデータ名や手続き名）が明瞭で読みやすいこと。
- プログラムの構造が単純明瞭で読みやすいこと。

よくモジュール化されており、1モジュールは小さくて単純に出来ている。

1モジュールの大きさは、20～50ステップ程度である。

- 字下げ、けたぞろえ、行がえ、空白行などを適切に行い、編集上の配慮がよくなされている。
- 適度に冗長度があり、注釈なども付

加されており、プログラム自身が文書となるようなコーディングがなされている。

5. 構造化プログラミングの道具だて

構造化プログラミングでは、従来の道具だて（例えばプログラム流れ図）があまり利用できない。さらに、新しい道具だてがいろいろ必要になる。

そこで、いろいろな道具だてが新しく開発され実用化されている。

次にその道具だてのいくつかを紹介する。

(1) HIPO

Hierarchy plus input-process-outputの略で、プログラム設計のための文書化の手法である。

HIPOは単純な文書化の手法としても使用できるし、下降型のプログラム開発の補助手段としても利用できる。

HIPO図は、次のような利点を持っている。

- システムの機能を階層的に表現できるので、機能の構造がわかりやすい。
- 入力、処理、出力の関連を明らかにできる。制御の流れとデータの流れを同時に示すことができる。
- プログラム全体の中に個々の機能を位置づけることができるので、機能の理解や機能の変更が容易である。

HIPOは、図式目次、概括HIPO図、詳細HIPO図から構成される。

(a) 図式目次

ここには、凡例、後続のHIPO図の相互関係、各HIPO図の内容の要約が示される。この図式目次を使用すれば、HIPO図全体を通読しなくても特定レベルの情報や特定のHIPO図を見つけ出せる。

図式目次の構成要素は、次の三つで

ある。

- 階層図
- 凡例
- 説明欄

主要部分は、階層図で、ここに概括HIPO図や詳細HIPO図の名前や識別番号も記入する。階層図のそれぞれの枠は、一つ一つがHIPO図に対応する。またシステムやプログラムの構成図としての機能も持つ。

凡例には、HIPO図に用いる記号とその用途を列記する。

説明欄は、階層図の枠に書く各機能を補足したいときに用いる。

(b) 概括HIPO図

概括HIPO図には、システムまたはプログラムの個々の段階で遂行する機能の一般的なアイデアを示す。

処理の部分には、どんな機能を遂行するかを、簡条書で記述する。

(c) 詳細HIPO図

詳細HIPO図には、個々の機能と関連するデータを詳細かつ簡潔に記述する。

機能の論理を示すのではなく、機能自体を詳細に記述する。

(2) 構造化流れ図

構造化流れ図は、構造化プログラミングのための流れ図として考えられたもので、構造化プログラミングの三つの構成要素——連続的な処理、選択的な処理、繰返し処理——とその構造化を、モジュール全体の構造の中に位置づけて表現できるようにした図である。

構造化流れ図を用いると、モジュールの論理設計を下降型で段階的に行うことができる。構造化流れ図は、次のような特徴を持っている。

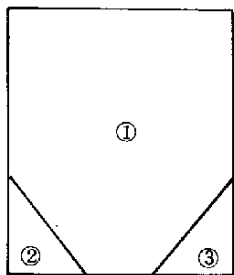
- 構造化プログラミングの設計思想に合わせて作成された図である。
- モジュール全体の構造が把握しやすい。
- 複数ページにわたる結合子はないので、一覧性に優れる。
- コーディングとの対応付けがはっきりしている。
- 図を修正したり、注釈文を挿入したりするのは困難である。

構造化流れ図では、次のような記号を用いる。

(a) 処 理

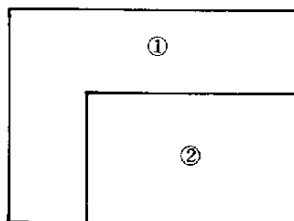


(b) 選択 (IF THEN ELSE)



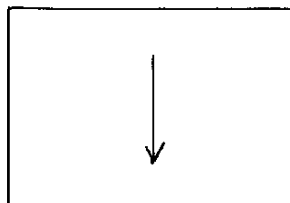
- ①に選択のテスト条件を書く
- ②にY(yes)を書く
- ③にN(no)を書く
- ②と③の下に長方形をつけて、それぞれの処理を書く

(c) 繰返し (DOWHILE型)



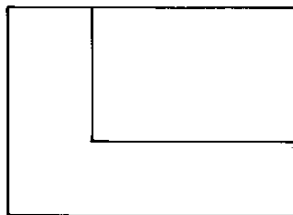
- ①に繰り返す条件を書く
- ②に繰り返す処理を書く

(d) 無 処 理

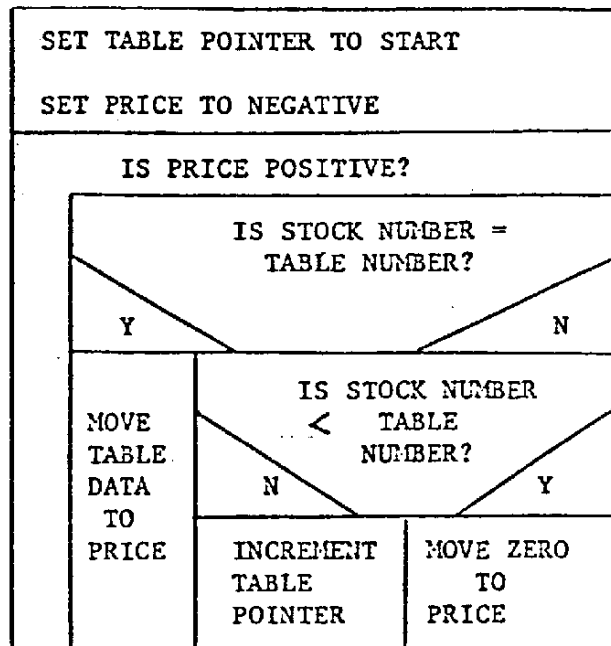


他に、次のような二つの応用形がある。

(e) 繰返し (DOUNTIL型)



- (f) 多数選択 (CASE)
構造化流れ図の例



(3) 擬似コード

擬似コードは、コンピュータやプログラミング言語の制約にしばられないで、機能を分かりやすく記述できるように人為的に用意された言語である。従って、コンピュータやプログラミング言語のことを考える必要はない。

自由に必要な動作を記述できる。但し、その記述は、最終的なプログラミング言語によるコーディングを連想できるもの

でなければならない。

使用するデータの名前には制約はない。どんなデータにどんなデータ名をつけてもよいが、できるだけ記述する機能が容易に読みとれるような名前を用いるのが良い。

次に擬似コードの記入例を示す。

```

SET STARTING VALUE OF TABLE POINTER
SET PRICE TO ZERO
SET FOUND FLAG TO FALSE
DOWHILE FOUND FLAG IS FALSE
    OR TR-STOCK-NO < TABLE-STOCK-NO (TABLE POINTER)
    IF TR-STOCK-NO = TABLE-STOCK-NO (TABLE POINTER)
        COPY PRICE FROM TABLE
        SET FOUND FLAG TO TRUE
    ELSE
        INCREMENT TABLE POINTER
    ENDIF
ENDDO

```

擬似コードは、英語でも日本語でもよい。また、動詞は英語、名詞は日本語という混在した使い方をしてもよい。動詞には機能を示す動詞と制御の型を示す動詞がある。また、機能を示す動詞には、業務寄りの大きな機能単位を示すものから、プログラムの1ステートメントに対応するような小さな機能単位を示すものまで、さまざまなレベルのものがある。

以上のような点を考慮しながら、使用する単語と文の表現についての文法を定めたものが擬似コードである。

擬似コードは、ユーザが必要に応じて自由に定めることができる。

また擬似コードは、単独で使用してもよいし、HIPOやその他の文書化の手法と組み合わせて使用してもよい。

参 考 文 献

- (1) マイヤー, 久保未沙他訳「高信頼性ソフトウェア」近代科学者, 1976
- (2) 久保未沙, 国友義久「効果的プログラム開発技法」bit Vol.8, Vol.9, 共立出版, 1979
- (3) 内川, 岡部, 竹内, 辻村編「現代の社会とコミュニケーション」東京大学出版会, 1973
- (4) 板坂元「考える技術, 書く技術」講談社, 1973
- (5) 山中秀男「技術文書のまとめ方」東洋経済新報社, 1971
- (6) 鮑戸弘「コミュニケーション」筑摩書房, 1972
- (7) N. Wirth「On the Composition of Well-Structured Programs」Computing Surveys, 1974
- (8) N. Wirth「Program Development by step wise Refinement」CACM, 1971
- (9) E. W. Dijkstra「Notes on Structured Programming」T. H. Report 70-wsk-03, 1970
- (10) F. T. Baker「Chief Programmer Team Management of Production Programming」IBM System Journal, Vol.11, No.1, 1972
- (11) T. Gilb and G. M.「Humanized Input; Techniques for Reliable Keyed Input」Weinberg, Winthrop, 1977
- (12) M. A. Jackson「Principles of Program Design」Academic, 1975
- (13) H. Katzan「Systems Design and Documentation: HIPO」Van Nostrand Reinhold, 1976
- (14) B. W. Kernighan「Software Tools」Addison Wesley, 1976
- (15) P. W. Metzger「Managing a Programming Project」Prentice-Hall, 1973
- (16) G. J. Myers「Reliable Software through Composite Design」Mason/Charter, 1975
- (17) 「HIPO-設計の補助手段と文書の手法」IBMマニュアル, 1975
- (18) G. J. Myers, 有沢誠訳「ソフトウェアの信頼性。ソフトウェア・エンジニアリング概説」近代科学社, 1977
- (19) B. W. Kernighan, P. J. Planger, 木村泉訳「プログラム書法」共立出版, 1976
- (20) 竹下享「ドキュメンテーションの標準化」日本経営出版, 1974
- (21) カーニバン, ブローガー, 木村康沢訳「ソフトウェア作法」共立出版
- (22) 「JISハンドブック情報処理1981」日本規格協会

第9章 データベース (C11)

キーワード

データベース	database
データベース・システム	database system
データベース管理システム	DBMS: database management system
データ独立性	data independence
スキーマ	schema
データベース管理者	database administrator
データ一貫性	data integrity
データ制御	data control
データモデル	data model
データ記述	data description
データ操作	data manipulation
データベースの編成	organization of database

目 標

1. データベースの考え方およびデータベース・システムについての全般的な理解を得ることを目標とする。
2. ここで対象とするのは、主として、応用プログラマあるいはプログラムによらないエンド・ユーザなどの一般的なデータベース利用者、あるいはデータベース・システムの構築や維持に従事する中堅的プログラマなどである。
3. 教育の重点は、データベースの考え方やその利用方法、データベース管理システムの利用方法といった面に置かれる。
4. データベース・システムを単にブラック・ボックスとして利用できる知識を与えるだけでなく、データ・モデルの考え方やデータベースの実現 (implementation) 技術の面で、理論的、技術的な知識の裏づけを有する技術者を育成することが目標であ

る。

内 容

1. データベースの概念

データベース、データベース・システムについての基本的な概念、考え方を提示し、データベース・システムの機能の概略を説明する。計算機による情報処理におけるデータベース・システムの役割について概観を与える。

(2) データベースの考え方

データベースという考え方およびデータベースを中心とするソフトウェア・システムとしてのデータベース・システムについて、出現の背景、目的、特徴、有効性について理解させる。

- データベース概念の出現

・データベース・アプローチ

(2) データベース・システム

(1)でのデータベース・アプローチを有効なものとするためにデータベース・システムが備えなければならない性質、機能を整理する。現実世界の情報、計算機上での情報の表現、組織体における情報の利用者、そしてこれらを媒介するシステムとしてのデータベース・システムのあり方を理解させる。特に、データ独立性の要請との関係からスキーマの概念が重要であることを説明する。

- ・データベース・システムに対する要請
- ・データ独立性とスキーマの概念
- ・利用者インタフェース

(3) データベース・システムの機能

データベース・システムが持つべき基本的な性質、機能について概観を与える。具体的なシステムについての詳細は3.で説明することとするが、データベース・システムの一般的構成、データ記述、データ操作の基本的機能について理解させる。また、データ制御の機能についても、データの正当性 (validity) の保証、同時実行制御、障害回復 (error recovery) に関する通常の機能を説明する。

- ・スキーマ
- ・データ記述機能
- ・データ操作機能
- ・データ制御機能
- ・データベース支援機能

2. データ・モデル

錯綜した現実世界の情報と計算機上での情報表現を媒介する抽象化 (abstraction) のステップとして、データ・モデルの考え方を理解させる。

(1) データ・モデル

データ・モデルの考え方の基本を理解

させ、その有用性を考えさせる。

(2) 階層モデル

階層モデル (hierarchical model)

の基本的概念を説明する。

- ・階層構造
- ・階層モデルによるデータベース・システム

(3) ネットワーク・モデル

代表的なデータ・モデルであるネットワーク・モデル (network model) の基本的概念を説明する。

- ・ネットワーク
- ・バックマン線図 (Bachman diagram)
- ・ネットワーク・モデルによるデータベース・システム

(4) 関係モデル

理論的に最も整理されており、実用上の影響が大きくなりつつある関係モデル (relational model) について概説する。

- ・関係
- ・関係代数系 (relational algebra) と 関係論理系 (relational calculus)
- ・関係モデルの意味
- ・関係モデルによるデータベース・システム
- ・データの正規化 (normalization of data)

3. データベース管理システム

データベース・システムの中核となるデータベース管理システムの概要を理解させる。それぞれの教育環境で利用できる実際のデータベース管理システムを例にとり、基本的な機能、使用法を説明あるいは調査させ、簡単な検索プログラム程度は書ける能力を養う。

(1) データベース管理システムの発展

データベース管理システムの現在に至

る発展の概略を説明する。

- データベース・システム以前の試み
- CODASYL方式のデータベース管理システム

- 商用データベース管理システムの現状

(2) データベース管理システムの基本的機能

- スキーマについて
- データベース管理システムの動作
- データ操作機能
- データ制御機能
- 支援機能

4. データベースの実現

データベースを計算機上に実現するための情報構造の表現方法について概説する。

- (1) ハッシュ関数の利用
- (2) つなぎ (link) による編成
- (3) 索引 (index) による方法
- (4) 実際のシステムでの実現法

論、実現方法、適用方法について批判的に考えられる基盤を与えるように心がける必要がある。

5. データベースの物理的編成方法については、プログラミングやファイル・システムでのデータ構造やファイル編成との関連性を保ち、これらの技術の応用として扱うべきである。

指導上の留意点

1. 適当なデータベース管理システムを例にとり、そのシステムを用いてデータベースを利用するプログラムを書く。あるいは対話的な問合わせを行うなどして、実際にデータベース・システムを利用する経験を与える必要がある。
2. 演習に利用するデータベース・システムとしては、最も一般的なシステムであるCODASYL型のシステムを用いるのがよいであろう。
3. データモデルの考え方を積極的に取り入れ、ある程度抽象的、形式的な態度でデータベースを扱う能力を養う必要がある。
4. データベースに対するソフトウェア利用形態、理論、実現技術はまだ完成された水準に達しているとはいえない。種々の方歩

参 考

1. データベースの概念

1.1 キーワード

データベース	database
データベース・システム	database system
ファイル・システム	file system
データ独立性	data independence
スキーマ	schema
概念スキーマ	conceptual schema
外部スキーマ	external schema
内部スキーマ	internal schema
データベース管理者	database administrator
問合せ機能	query facility
データ記述言語	DDL: data description language
データ操作言語	DMC: data manipulation language
親言語方式	host language system
独立言語方式	self-contained language system
データ辞書	DD/D: data directory/dictionary
アクセス制御	access control
一貫性制御	integrity control
同時実行制御	concurrency control
施錠	lock
障害回復	error recovery

1.2 データベースの考え方

(1) データベース概念の出現

1960年代後半から発展をとげたデータベース・システムがどのような必要性の下に出現してきたかを、データベースの概念とそれまでのファイル・システムに基づく情報処理との比較によって考察する。

(a) ファイル・システム

計算機の事務処理への応用以来、現在でも広く用いられている処理方式は、人手による事務処理のプロセスをほぼそのまま計算機に移しかえるアプロー

チである。その場合、帳簿など事務処理上のファイルに対応する計算機上のファイルの集まりであるファイル・システムとしてシステムが構成される。

ファイル・システムの特徴として次のような点を示唆しておくのがよい。

④ ファイルはそれを利用する応用プログラムに付属しており、プログラムの論理は個有のファイルと密接に結びついている。

⑤ ファイルの構成は計算機上で提供されている物理的表現手段に拘束されがちである。

- ③ 各利用者はそれぞれ専用のファイル
を生成し、管理して行く必要がある。
る。

(b) データ・ベースへの転換

システムの多角化やデータ量の増加により、ファイルによるシステム構成は管理・保全に膨大な作業量とコストが必要になる。次のような点を考察しておくのがよい。

- ④ データの種類。作業内容の拡大に伴う応用プログラムの複雑化、プログラム作成コストの増大。

- ⑤ 同じデータが種々のファイルに散在し、冗長 (redundant) になること。

- ⑥ 冗長なデータの間では整合性 (consistency) が失なわれがちなこと。

- ⑦ システム全体としての拡張性に乏しいこと。

以上の点から、プログラムとデータの分離を行ない、システム全体としてデータを管理し、共同利用するというデータベースのアプローチが生まれたことを理解させる。

(2) データベース・アプローチ

以上のような状況を改善するためのアプローチをして登場したデータベース・システムは次のような性格あるいは目的で特徴づけられよう：

- 多目的利用が可能である。
- プログラムとデータの分離が計られている。
- システム全体としての統一的なデータ管理。
- 種々の利用者がそれぞれの仕事に適した手段でデータを共同利用することができる。

- 定型的、定常的な作業だけでなくアドホックな情報要求に対しても対処しうる。

このような目的に合ったシステムが実現されれば次のような利点があることを説明する。

- データを種々の利用者が目的に応じて共同利用できる。
- データの冗長化を防ぐことができる。
- データ相互間の整合性を高めることができる。
- 応用プログラムの作成が容易になり、一般に、データのアクセスが容易になる。
- データ管理、データ処理に対する組織の標準 (standard) を確立しやすくなる。

1.3 データベース・システム

- (1) データベース・システムに対する要請
前節であげたような目的を達成あるいは有効性を発揮するためには、データベース・システムは少くとも次のような条件を満足させなければならないことを理解させる。

- (a) システムが保持するデータを体系的、組織的に記述し、管理できること。
- (b) データの形式や構造に対する変化から応用プログラムが独立していること。
- (c) 複数の利用者が大きなシステムを共同利用できるために、誤ったデータの投入、データの同時更新、障害回復、機密保護などに対する配慮が必要であること。
- (d) 個々の応用プログラムや利用者が自分の視点でデータをアクセスできること。

(2) データ独立性とスキーマの概念

- (1) で触れたような要請のうち技術的に

最も基本的で、データベース・システム
のアーキテクチャに重要な影響を与える
のはデータ独立性の問題である。

ここではデータ独立性と、これを達成
するための基本的なアイデアとしてスキ
ーマの概念について説明する。

(a) 3層スキーマ

現在最も広い支持を得ている方式は
ANSI/X3/SPARC 報告で提示され
た3層スキーマ構成であろう。ここ
でも次の3つのスキーマにしたがって説
明する。

• 概念スキーマ

データベース化する現実世界のデ
ータ全体の構造の記述

• 外部スキーマ

利用者から見たデータベースの構
造の記述であり、各利用者あるいは
利用者グループごとに存在する。

• 内部スキーマ

概念スキーマに適合する実データ
を計算機上で表現する為のデータベ
ースの物理的編成方式の記述。

現実世界の情報、計算機上での情報の
表現、組織体内の利用者ごとの見方と
いうデータベースを取りまく環境のあ
り方を考え、このような層別を行うこ
との自然なことを理解させるのがよい。

(b) データ独立性

データ独立性は、データベースを利
用する応用プログラムを、データベ
ースに用いる装置や媒体、データ表現の
物理的構造などから分離して、できる
だけこれらと独立に使用できるように
することである。

(3) 利用者インタフェース

(a) 利用者の種類

データベースの利用者に対して次の

ような区別が考えられることを説明する。

• 一般の利用者

• データベース管理者

一般の利用者は更に、データベ
ースとの交信方法により

• 応用プログラマ

• プログラムを用いない利用者 (non-
programmer, parametric user な
ど)

に分けられる。これらの利用者の特
徴とデータベース・システムが提供
すべき機能について検討させる。

(b) データベース管理者

データベース・システムの基本目的
の一つであるデータを集中的に管理し、
利用者が適切にデータベースを利用で
きるようにするためにはシステム全体
を統合的に管理する必要がある。

ここではデータベース管理者の存在
の重要性およびその主な任務について
認識させる。データベース管理者の任
務としては次のような観点から説明す
ればよいであろう。

① データベース・システムの導入計
画

② データベースの設計

主にスキーマによるデータベース
構造の設計であり、3層のスキーマ
に応じて、

組織体管理者 概念スキーマを設計

応用管理者 それぞれの外部スキ
ーマを設計

データベース管理者

内部スキーマを設計
のように細分化することもできる。

③ データベースの運用管理

• アクセス制御の方式

• データベースの使用状況監視

- データベースの一貫性の維持
- 障害回復の対策
- データベースの改善

④ 文書の作成・改訂

(c) 言語インターフェース

各種のスキーマの定義、データベースの操作は特定の記述言語を用いて書く。概念スキーマ、外部スキーマ、内部スキーマを書く言語はデータ記述言語 (DDL: data description language), また利用者によるデータ操作を書く言語はデータ操作言語 (DML: data manipulation language) と呼ばれる。

言語を介しての利用者とのインタフェースのあり方については、従来から

- 親言語方式
- 独立言語方式

の分類が行なわれており、現在でも知っておく必要はあろう。しかし、親言語方式を基本とするシステムでも広い利用者層に広げるために独立言語的な高水準の言語を用意するものも多いし、独立言語を親言語によるプログラム中で利用できるシステムなどがあり、両者の区別は意味を失いつつある。むしろ問合せ機能 (query facility) の水準の向上という点から眺めるべきであろう。

(d) データ辞書

データベース規模が大きく、複雑化するにつれ、データベースを管理するためのデータの管理が重要になる。この種のデータは本来のデータについてのデータという意味でメタ・データ (meta data) と呼ばれることがある。通常、

- データ・ディクショナリ

(data dictionary)

- データ・ディレクトリ
(data directory)

に分けられ、総称して DD/D (data directory/dictionary) と呼ばれている。DD/D の必要性、DD/D に必要な機能について概略を説明する。

1.4 データベース・システムの機能

1.3 まですでに触れたような要請を満たすため、データベース・システムがどのような構成をとり、又どのような機能が必要になるか、その概要を次のような観点から説明する。(図-1 参照)

(1) スキーマ

- 概念スキーマ
- 外部スキーマ
- 内部スキーマ

(2) データ記述機能

(3) データ操作機能

(4) データの制御機能

次のように分けて考える。

(a) データのアクセス制御

資格のない人がデータを勝手にアクセスすることを防ぐための機密保護機能。

(b) データの一貫性の制御

データベース上のデータが常に整合性 (consistency) を保っているようにするための機能で、次の3点を考える。

① 正当性 (validity)

データの格納、変更の時点で、その値がある正当性の基準を満たしているように検査できる。

② 同時実行制御

(concurrency control)

複数の利用者からのデータベースの同時アクセスにより、プログラム

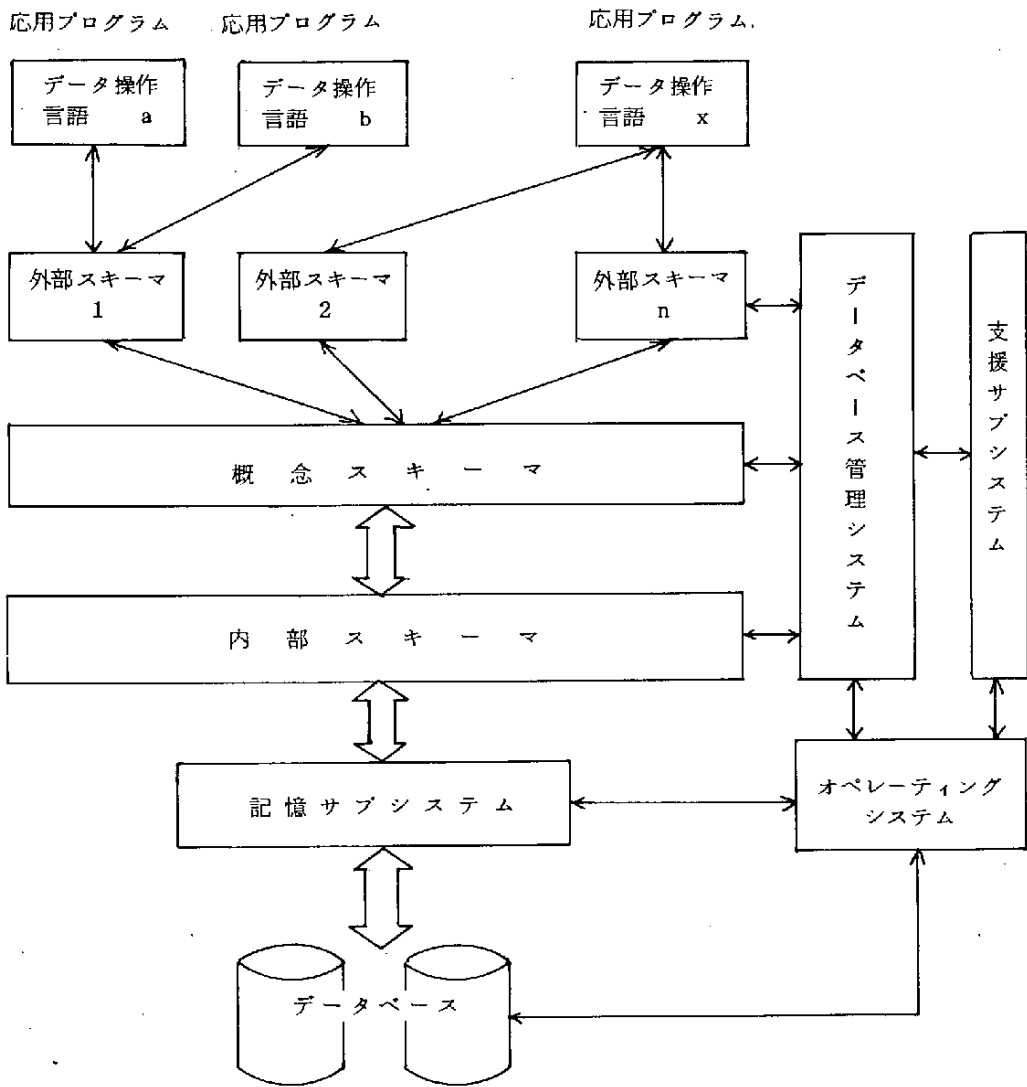


図 - 1 データベース・システムの概念図

間の干渉やデータの妥当性が失なわれたりすることを防ぐ。特に、同時更新 (concurrent update) などデータの整合性が失なわれる可能性を検討し、この問題を制御する為の施錠 (lock) 操作、デッド・ロック (deadlock) の可能性について説明する。

③ 障害回復 (error recovery)

次の概念およびこれらによる障害回復の概要を説明する。

- チェック・ポイント
(check point)
- 追跡控え (audit trail)
- 直前像 (before image)
- 直後像 (after image)
- 前進復帰
- 後退復帰 (roll back)

(5) データベース支援機能

- (a) データベースの初期生成
- (b) データベースの再構成
- (c) 障害処理
- (d) 性能評価
- (e) 統計的データの収容

2. データ・モデル

2.1 キーワード

データモデル	data model
階層モデル	hierarchical model
ネットワーク・モデル	network model
関係モデル	relational model
木	tree
順序木	ordered tree
ネットワーク	network
有向グラフ	directed graph
レコード型	record type
レコードの実現値, 具体値	occurrence, instance
親子集合	set
バックマン線図	bachman diagram
関係	relation
関係代数系	relational algebra
関係論理系	relational calculus
結合演算	join operation
アクセス経路	access path
データの正規化	normalization of data

2.2 データモデルの考え方

(1) データモデル

現実世界が持つさまざまな情報をデータベース上に集約するためには、これらの情報を錯綜したままの形で表現するわけにはいかない。現実世界の姿を抽象化し、論理的に情報の関連性を整理するための手段がデータモデルの考え方である。

ここでは、データモデルの考え方が、

- データの論理構造の整理
- 物理的表現に接近するステップ
- データベース・システムの理論的基礎
- データベース設計の足がかり

などの観点から有用であることを説明し、代表的なモデルを理解させる。

現在利用されている代表的なデータモデルとして次がある。

(a) 階層モデル

(b) ネットワーク・モデル

(c) 関係モデル

簡単な例題を用いてこれらのモデルによるデータの関係のとらえ方のアイデアを理解させる。

2.3 階層モデル

(1) 階層構造

データ処理でデータ間の関係を把握するモデルとして古くから利用されてきた方法に、木構造で表現される階層構造があることを説明する。

階層モデルでは

• レコード型 (record type)

• 親子関係

の2つが基本的である。

他のデータ・モデルと同様、レコード型とその実現値 (又は具体値 ; occurrence あるいは instance) の区別

をはっきり理解させる必要がある。親子関係は、2つのレコード型の実現値の間に成立する関係であるが、次の条件を満たしている必要がある。

- (a) どんな2つのレコード型の間にも高々1つの親子関係だけしかあってはならない。
- (b) 子となるレコード型の実現値は、親となるレコード型のただ一つの実現値とだけ関係する。

階層モデルは、これらの条件を満たす有向木 (oriented tree) として表現される。

(2) 階層モデルによるデータベース・システム

階層モデルに基づいたデータベース・システムとしては、IMS が代表である。

IMS では構造に従って順序関係が導入されており、この点も考えるなら部分木の間の順序も考えた順序木と考えた方がよい。

このようなモデルではどの節のレコードに達するにもただ1つの経路しか存在しない。データの検索は原則として順序木の根からの経路をたどることにより行なわれる。

2.4 ネットワーク・モデル

(1) ネットワーク

有向木あるいは順序木による階層モデルの表現力を向上させることを目的としている。次の図のような関係を表現するには木構造では不十分であることを説明する。

階層モデルで表現できない点は、

- 同じレコード型に至る枝が複数個ありうること、
- 1つのレコード型から (0個以上の) 複数個の他のレコード型を経由してそ

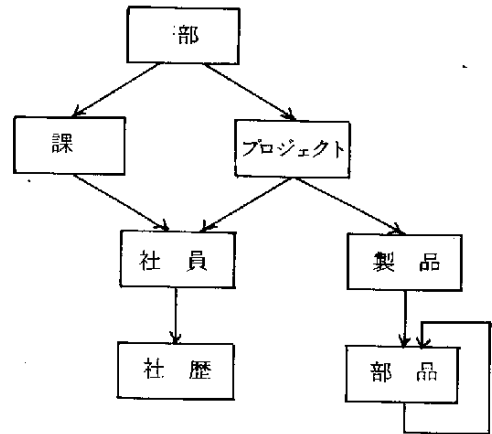


図-2 ネットワーク

のレコード型に戻る有向路がありうること、

である。この点を表現できるようにすることは一般のネットワーク (網, 有向グラフ) を考えることに他ならない。

ネットワーク・モデルでの基本概念は

• レコード型

• 親子集合型 (set type)

である。親子集合型は、始点となる1個のレコード型と終点となる複数個のレコード型からなる順序木である。始点のレコード型は親 (owner)、終点のレコード型は子 (member) と呼ばれる。ネットワーク・モデルでの構造は、

- 1つのレコード型は複数の親子集合型に関係することができる。

• 有向輪があってもよい。

という条件で親子集合型を組み合わせ構成される。

ネットワーク・モデルでは次の点に注意する必要がある。

- 階層モデルのように木ではないから根は存在しない。
- 1個のレコード型に到達する経路は一

般に複数個あり得る。

- 2つのレコード型の間に1以上の関係を親子集合型として考えることができるため、親子集合型には名前をつけて識別する。

(2) バックマン線図

親子集合による構造を表現する為によく用いられる手法としてバックマン線図あるいはデータ構造図 (data structure diagram) がある。

バックマン (Bachman, C. W.) は、ネットワーク・モデルの基本である親子集合 (set) の概念を考察し、初期のデータベース管理システム IDS でこれを用いた (1960 年代中頃)。

バックマン線図では、長方形でレコード型を示し、名前をつけた節で親子関係を示す。

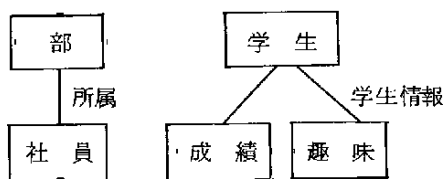


図-3 バックマン線図

バックマン線図はレコード型の間の構造を示すものであり、その実現値であるレコードに対する関係は表現していない。

このレベルで構造まで表現するためには次のような手法が用いられている。

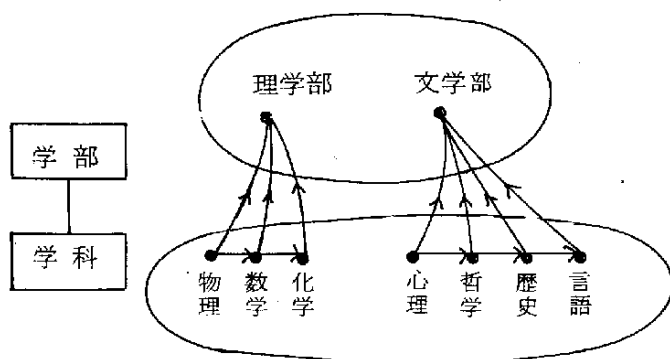


図-4 実現値の関係

(3) ネットワーク・モデルによるデータベース・システム

上記の IDS を始め、これに続く、影響を受けて提案された CODASYL データベース作業グループの仕様 (DBTG'71 提案) に基づく多くのデータベース管理システムがネットワーク・モデルに分類される。階層モデルの場合もそうであるが、ネットワーク・モデルの場合も、個々のデータベース・システムでの構造の表現方法は、実現値のレベルでの順序関係や物理的な編成方法から来る制約などのかなりこみいっているので注意が必要である。

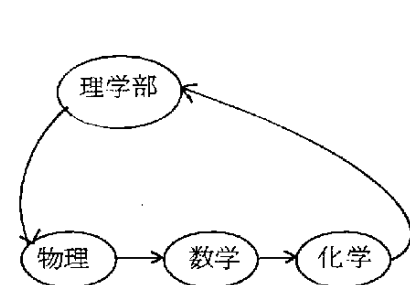
階層モデルと同様、こうしたモデルはデータの論理的な関連性としての構造と同時に、物理的なデータベース上のデータをアクセスする為の経路 (アクセス経路 access path) をも示していることに注意する必要がある。

2.5 関係モデル

(1) 関係

関係モデルでは、データ間の関連性を複雑なデータ構造で表現することは一切行わず、データベース上の構造としては数学上の意味での関係 (relation) だけを考える。そのかわり望むデータをアクセスするために、関係に対して適用される種々の基本演算を考え、この両者によりモデル化する。

(a) 関係モデルの基礎として次の数学上



の概念を復習する。

- 集合と集合演算

- 直積

- 定義域 (domain), と属性

(attribute), (n 項) 関係 (有限個の集合の直積の部分集合), 関係の要素としての組 (tuple) または n 個組 (n-tuple)

(b) 関係とファイル, 組とレコードの間に自然な対応関係が考えられることを示唆する。

属性名			フィールド名		
学部	学科	学生数	学部	学科	学生数
理学部	物理	30	理学部	物理	30
文学部	心理	50	文学部	心理	50

関係 ファイル

図-5 関係とファイル

(c) 関係モデルでは, 上のような関係の集まりをデータベースと考える。

(2) 関係代数系と関係論理系

関係モデルを提唱した codd. E. F. は上のような関係の集まりとしてのデータベースに対する操作を関係に対する演算と考え, 関係代数系を定式化した。

(a) 関係代数系での演算

- 通常の集合演算…合併 (union), 共通部分 (intersection), 差 (difference), 直積 (cartesian product)
- 独自の演算…射影 (projection), 結合 (join), 制限 (restriction), 除算 (division)

最も重要で, 直観的理解も得やすい結合演算などについて, 関係に演算をほどこす操作がデータベース上の必要な

データをアクセスすることに相当することを説明する。

(b) codd は更に, データベースに対する問合せを表現する形式系として関係論理系を定式化している。関係論理系は一階の述語論理系 (1st order predicate calculus) 相当の論理系である。

(3) 関係モデルの意味

関係モデルは,

(a) データベースと利用者とのインタフェースを単純化する。

(b) 次の意味でデータ独立性を向上させる: プログラムを,

- データの物理的配置や順序から独立させる。

- 索引 (index), 転置ファイル

(inverted file) などの検索効率の向上化のための手段から解放する。

- データのアクセス経路から独立させる。

(c) データベース・システムに対する理論的基礎づけを与える。

(d) データベースをアクセスするプログラムの記述レベルを手続き的 (procedural) なレベルから, 集合を基本とする非手続き的 (nonprocedural) なレベルに引き上げる。

などを目標として提案されている。

これ等の目標を追求する試みとして, 現実のシステムとしての効率には眼をつぶっている。しかし, データベースの世界にこのような強固な理論的基盤を与えつつあることは非常に重要であり, 又データの正規化の理論により現実のデータベースの構築に対しても目安を与えており, 今後は現場の技術者としても知らずに済まず訳には行かなくなろう。

(4) 関係モデルのデータベース・システム

上に述べたように、関係モデルは現実のシステムとして効率よく働くことは先ず度外視して研究された。現在のところまだ現実のシステムとして十分大規模かつ効率のよいシステムは実現されていない。しかし、system R、INGRESなど本格的なシステムの実動化が種々試みられており、注目に値する。

(5) データの正規化

関係データ・モデルの一つの成果としてデータの正規化の理論に触れる。

正規化の議論はデータベース理論においても重要であるが同時に、データベース上のデータ相互の関連性をどのような観点から考えるべきか、データベース上のデータをどのようにまとめておくべきかといった点で、データベースの設計上の問題にも有益であることを説明する。

正規化の理論への導入として、

- 第1正規形
 - 関数従属性 (functional dependency)
 - codd の第3正規形
 - Boyce-Codd による正規形
- について説明し、直観的意味づけを与える。

3. データベース管理システム

3.1 キーワード

RPG	RPG
情報代数	information algebra
IDS	IDS
CODASYL	CODASYL
DBTG 提案	DBTG
ADABAS	ADABAS
SYSTEM 2,000	SYSTEM 2,000
TOTAL	TOTAL
IMS	IMS
MARK IV	MARK IV
スキーマ	schema
サブスキーマ	subschema
記憶スキーマ	storage schema
データ記述言語	DDL: Data Description Language
データ記憶記述言語	DSDL: Data Storage Description Language

3.2 データベース管理システムの発展

以下のような流れに沿って現在のデータベース管理システムに至る発展の概要を説明する。

(1) データベース・システム以前の試み

データベース管理システムあるいはデータベース管理システムというある程度はっきりした概念に基づいたアプローチとなる以前から、ファイル処理の簡易化あるいは高水準化という方向で種々の動きがあった。

(a) RPGの系統

ファイル処理の手続きをパターン化した論理として内蔵し、プログラマの記述を非手続き的にしようとする試みは古くからあり、1960年代始めに成立したRPG (Report Program Generator) が典型である。RPG IIの形で現在でも強力な道具の一つとなっている他、データベース管理システム

のMARK IVにも影響を与えている。

(b) COBOLの拡張

度重なる改訂を経て、ファイル編成やアクセス方式の拡張、RPG的な報告書作成機能の導入のほか、後に述べるCODASYL方式のデータベース管理システムはCOBOLと整合性を保ちつつ開発されて行った。

(c) 情報代数

実動化された処理系は存在しないが、1960年代始めのCODASYLによる情報代数 (information algebra) のアイデアにも触れておくのがよい。数学的な観点からファイル処理を抽象化し、集合レベルの操作を考えることによりファイル処理を非手続き的に捉えようとした点、また計算機の物理的制約からの独立性を考えていた点など、今日のデータベース・システムに与えている影響は大きい。

(2) CODASYL方式のデータベース管理システム

ネットワーク・モデルに基づくデータベース管理システムであり、現在実用上最も普及していると考えられる。元来は、1960年代始めに登場したGE社によるIDS(Integrated Data store)の考え方を発展させたものである。

CODASYLの仕様案に従ったシステムが各計算機メーカ、ソフトウェア会社などで提供されている。

CODASYL関係の足どりとしては、

- 1969, 1971 データベース作業グループ(DBTG)による言語仕様案DBTGの1提案と呼ばれ、CODASYL方式の原型を与えた。
- 1973 スキーマ記述用言語DDLの仕様案
- 1976, 1978 COBOLデータベース機能仕様案
- 1977 FORTRANデータベース機能仕様案

といったところである。

(3) 商用データベース管理システムの現状
現在普及している商用システムとしては、

• ADABAS

Software AG, 1972年成立

• SYSTEM2000

MRIシステムズ社

1969-1972年成立

• TOTAL

Cincomシステムズ社,

1968年成立

• IMS IBM社, 1969年成立

• MARKN

Informatics社, 1968年成立

などが主なところである。他に、医療情

報処理システムとして注目されている

MUMPS(Massachusetts General Hospital)もデータベース管理システムと見ることもできよう。

3.3 データベース管理システムの基本的機能

ここでは、1.4で概略を説明したデータベース・システムが基本的に備えるべき諸機能が実際のデータベース管理システムでどのような形で提供され、実現されているかを理解させる。

本章では、教育環境で利用できる実際のデータベース管理システムを用いて簡単な応用プログラムが書けることを考えている。したがって、この節で扱うデータベース管理システムは教育上利用できるシステムである必要がある。以下では、最も一般的で機能的にも充実しているCODASYL型のデータベース管理システムを例として要点だけをあげる。

(1) スキーマについて

現在のCODASYL型のデータベース管理システムは、

- スキーマ(schema)
- サブスキーマ(subschema)
- 記憶スキーマ(storage schema)

による3層構成のスキーマを用いる。

スキーマはデータ記述言語DDL

(Data Description Language)により、

記憶スキーマはデータ記憶記述言語

DSDL(Data Storage Description

Language)により記述する。CODASYL

型では親言語方式を採用しており、

COBOL および FORTRAN を親言語

と考えており、利用者側の立場からは

サブスキーマとデータベース操作手順

をCOBOLあるいはFORTRANのデ

ータベース機能の仕様によって記述す

る。

DDLによるスキーマの記述と1応用プログラムから呼出されるサブスキーマを介しての対応関係を説明する。親子集合による構造の記述方法、親子属性、親子集合の順序 (set ordering)、

親子集合の選択 (set selection)などを中心に説明する。

(2) データベース管理システムの動作

次のような概念図によってシステムの動き、データの流れを説明する。

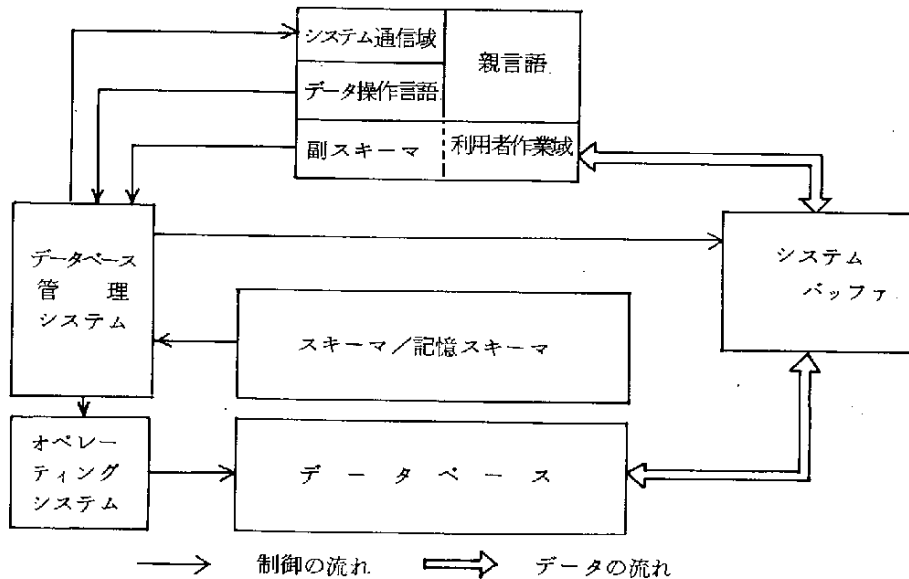


図 - 6 データベース・システムの動き

(3) データ操作機能

次のような概念、基本命令を説明する。

- (a) サブスキーマとの結合
- (b) 領域 (realm, area)
- (c) 現在指示子 (currency indicator) とその必要性
- (d) 1時1レコードのデータ操作論理
- (e) レコード・キー, データベース・キー

二

(f) 基本データ操作命令

- READ データベースのある部分の利用通知
- FINISH データベースのある部分の利用終了通知
- FIND データの検索条件を指示してデータの位置ぎめを行う。
- GET 位置のきまったデータの取り出し

STORE データの格納, 親子関係の設定

ERASE データベースからレコードを消去

CONNECT レコードを親子集合に加える

DISCONNECT レコードを親子集合からはずす

MODIFY データの変更

(4) データ制御機能

- (a) アクセス制御 (ACCESS-CONTROL)
- (b) 正当性の確認 (CHECK)
- (c) 同時実行制御 (領域, レコードの施錠)
- (d) 障害回復

これらの問題についてどのような機能が提

供されているかを説明し、あるいは調べさせる。

これらの機能の利用法、デッドロックの可能性への配慮など注意すべき点などを説明する。

(5) 支援機能

利用できるデータベース管理システムについて、次のような支援プログラムを調べさせる。

- データベースの初期生成
- データベースの再構成
- 障害処理
- 性能評価
- 統計的データの収集

4. データベースの実現

4.1 キーワード

データベースの物理的表現	physical representation of database
ハッシュ関数	hashing function
直接編成	direct organization
同義語	synonym
衝突	collision
つなぎ	link
ポインタ	pointer
循環つなぎリスト	circularly-linked list
ポインタ配列	pointer array
多重つなぎ編成	multi-linked organization
索引	index
索引つなぎ順編成	indexed sequential organization
あふれ	overflow
B-木	B-tree
B*-木	B*-tree
転置ファイル	inverted file

4.2 ハッシュ関数の利用

キーとなるデータの値にハッシュ関数を作用させてそのレコードの位置を決定する方式である直接編成について概説する。

(1) ハッシュ関数

理想的には、キーの値分布が偏っても一定の範囲に様に分布するようにできることがよい。ハッシュ関数を構成するための次の様なアルゴリズムを説明する。

- (a) 割り算法
- (b) 掛け算法
- (c) 微分析法
- (d) 基数変換法
- (e) 2乗法

(2) 同義語の発生とその処理

ハッシュ法の原理上からどのように関数を選ぶとも異なるキー値に対して同じ場所が与えられることは避けられない。

この同義語あるいは衝突 (collision)

の発生を説明し、同義語が生じた場合の解決法について説明する。基本的には次の2つの方法がある。

- (a) 開放アドレス方式 (open addressing)
- (b) 同義語の収容専用の別の領域を用いる。

(3) ハッシングの特徴

次のような性質があり、データベース編成法としてはつなぎによる編成といっしょに用いるなど補助的な役割で利用される。

- (a) レコードのアクセスに要する時間は記憶領域の使用率に依存する。
- (b) アクセス経路に沿ってデータを処理するといった用い方は不可能である。
- (c) 1個のレコードに対して複数のキーで検索することは一般には不可能である。

4.2 つなぎによる編成

データベースの編成法としては最も基本的

で利用度の高いのは、つなぎ(link)、あるいはポインタ(pointer)を用いる方式である。

(1) 基本的な概念

つなぎを用いた基本的な構造およびデータの投入、削除のアルゴリズムを復習する。よく利用される形態として

- 循環つなぎリスト

(circularly-linked list)

- ポインタ配列(pointer array)あるいはつなぎ索引(link index)

について説明する。

(2) 多重つなぎ編成

明らかにつなぎはレコードをアクセスする経路であり、データベース・システムのように1つのレコードを様々な観点からアクセスする必要がある場合には複数のつなぎを備えることになる。循環つなぎを利用した多重つなぎ編成によりこの機能を実現することができることを示す。

データベースで必要と考えられるネットワーク構造などが多重つなぎでどのように表現されるかを演習する。

(3) つなぎによる編成の特徴

次のような性質のあることを理解させる。

- (a) つなぎが物理的なアクセス経路と論理的な関係を同時に表現している。これには良い点も悪い点もある。
- (b) 複雑な構造の表現が容易である。
- (c) 構造操作のアルゴリズムは単純で容易に実現可能である。

4.4 索引による方法

(1) 索引つき順編成

古典的な索引技術である索引つき順編成についてはすでに学んでいる。基本的な索引構成方法、あふれ(overflow)の処理方法、特徴について復習する。

小規模なデータベース・システムでは索引つき順編成を基本としたものもいくつか存在する。しかし、レコードの追加が重なるにつれ最初に構成された索引がほとんど役に立たなくなり、レコードはあふれ領域を探しまわらないと見つけれなくなってしまう効率を回復するためには再編成を行わなければならない、これがデータベース編成としては大きな欠陥となっていることを理解させる。

(2) B-木

B-木についてもすでに学んだ。上のような欠陥を持つ索引つき順編成に対して、B-木の定義、操作アルゴリズム、B-木を改良したB*-木の手法、この方法の特性について復習する。

関係モデルに基づくデータベース・システムで盛んに利用され、例えば、イメージ(image)と呼ばれるsystem Rの索引がB-木の構成を用いている。

(3) 転置ファイル

情報検索(information retrieval)の分野でよく利用されてきた方法に転置ファイルによる索引がある。

転置ファイルを用いる大きな動機として、複数の異なるキーによってレコードをアクセスしたいというという要求があることを理解させる。

これまでに学んできたハッシュ関数、B-木、多重つなぎ、索引つき順編成などの手法を用いて転置ファイルが構成できることを示す。

実際のデータベース・システムについて、全ての属性に対して転置を行ったTDMSや指定された項目に対してだけ転置を導入したSYSTEM2000やADABASによって考察する。

4.5 実際のシステムでの実現法

- (1) 第3節で利用した現実のデータベース数量システムに基いて、スキーマの物理的表現に用いられている方法を解説あるいは調査させる。
- (2) データ・モデルで特徴づけられるデータ構造のレベルで、階層型、ネットワーク型、関係型の構造の表現としてどのような方法が考えられるかを検討する。

(13) 穂鷹良介「データベース要論」 共立出版、1978

(14) 植村俊亮「データベース・システムの基礎」 オーム社、1979

参考文献

- (1) 情報処理学会編「情報処理ハンドブック」 オーム社、1980
- (2) 西村怒彦監訳「データベースの選び方」 bit 臨時増刊 共立出版、1977
- (3) 西村怒彦他訳「データベース・システム」 bit 臨時増刊 共立出版、1973
- (4) J. Martin / 江村潤朗訳「データベース」 日本コンピュータ協会、1977
- (5) Nijssen, G. M. : 「Modelling in Data Base Management Systems」 North-Holland, 1976
- (6) データベース言語研究会訳「データベース用データ記述言語」 情報処理学会、1977
- (7) CODASYL Systems Committee : 「Feature Analysis of Generalized Data Base Management Systems」 ACM, 1971
- (8) Codd, E. F. : 「A Relational Model of Data for Large Shared Data Banks」 CACM, Vol. 13, No. 16, 1970
- (9) J. Martin, 「Computer Data Base Organization」 Prentice-Hall, 1977
- (10) CODASYL Systems Committee 「Selection and Acquisition of Data Base Management Systems」 1976
- (11) Date, C. G. : 「An Introduction to Data Base Systems」 Addison-Wesley, 1975
- (12) Lyon, J. K. : 「The Data Base Administrator」 Wiley, 1976

第10章 オペレーティングシステム(C6) とアーキテクチャ

キーワード

オペレーティング・システム	operating system
制御プログラム	control program
多重プログラミング	multi programming
ジョブ	job
ジョブ・ステップ	job step
プロセス	process
ジョブ管理	job management
データ管理	data management
プロセス管理	process management
資源管理	resource management
仮想記憶システム	virtual storage system

目 標

1. オペレーティング・システムを、特に制御プログラムを中心に理解させ、オペレーティング・システムを利用する能力を養う。
2. ジョブ制御言語などによりオペレーティング・システムを利用する経験を与える以外に、基礎的な部分に関してはオペレーティング・システムがどのような考え方および方法によって構成され、実現されているかを理解させる。
3. 教育の重点はあくまで応用プログラムのレベルでのプログラマである。オペレーティング・システムの設計、作成、保守などに対しては導入の役割は持たせられるが、このために十分な知識を与える事は目標としていない。

内 容

1. オペレーティング・システムの基礎
オペレーティング・システム、特に狭義のオペレーティング・システム(制御プログラム)の目的、機能、構成の概要を説明し、以下の章に対する概観を与える。
 - (1) オペレーティング・システムの歴史
オペレーティング・システムの発生、発展、現状と動向について概観を与える。
 - 初期のアプローチ
 - 多重プログラミングの支援
 - 現状と動向
 - (2) オペレーティング・システムの目的
システム資源 (system resource)の管理、スルプット (through-put)の向上、ターン・アラウンド・タイム (turn-around-time)の短縮、情報の共用と保護、RAS(信頼性・可用性

・保守性)といった観点からオペレーティングの狙いとするところを説明し、基本的な概念を理解させる。

(3) オペレーティング・システムの機能と構成

一般的なオペレーティング・システムの持つ機能と構成について概観を与える。

- ・ジョブ管理
- ・データ管理
- ・プロセス管理
- ・資源管理

それぞれの教育環境で利用できる計算機によってオペレーティング・システムの機能の概要を調べさせる。

2. ジョブ管理

ジョブ制御言語 (JCL) などによるジョブの定義方法とオペレーティング・システムによりジョブがどのように遂行されるかを理解させる。

(1) ジョブの記述

適当なオペレーティング・システムに基いてジョブ制御言語などによるジョブの記述方法を理解させる。

- ・ジョブ制御言語
- ・カタログ・プロシジャ

(2) ジョブの遂行

オペレーティング・システムによるジョブの遂行のプロセスを理解させる。

- ・ジョブの入力
- ・ジョブのスケジューリング
- ・資源の割当て
- ・ジョブ・ステップの実行
- ・ジョブ・ステップの終了
- ・ジョブの終了
- ・ジョブの出力

(3) ジョブ・スケジューリング

入力されたジョブを符号行列で管理し、実行すべきジョブを選択するスケジュー

リングについて説明する。

- ・ジョブ・スケジューリングの目的
- ・スケジューリング・アルゴリズム

3. データ管理

オペレーティング・システムによるデータ管理の概念、ファイルの管理とオペレーティング・システムの提供しているファイル構成法とアクセス法および入出力制御について理解させる。

(1) データ管理の基本概念

データ管理の基本として、2次記憶媒体の性質、ファイルの概念、オペレーティング・システムのデータ管理のあり方を理解させる。

- ・2次記憶媒体
- ・ファイル
- ・データ管理のシステム

(2) ファイル管理

ファイルの管理についての基本事項を説明する。

- ・ファイルの割当て
- ・ファイル・スペースの管理
- ・ファイルカタログ
- ・装置独立性
- ・ファイルの保護
- ・機密保護
- ・ファイルの再編成と再配置

(3) ファイルの編成とアクセス

ファイルによるデータ処理を支援するためオペレーティング・システムが提供するデータ構造 (data structure)とそのアクセス機能について説明する。具体的な利用法、プログラミング上の問題はファイル・システムの章で扱い、ここではオペレーティング・システムの持つ機能としての概要を理解させる。

- ・アクセス法
- ・処理モード

- ・レコード形式
- ・編成法
- ・プログラムとのインタフェース

(4) 入出力制御

ファイルによるデータ操作に対してオペレーティング・システムが行う具体的な作業とプロセスを理解させる。

4. プロセスの管理

オペレーティング・システムが行う作業の基本単位であるプロセス（あるいはタスク）の概念による並行的な動作の制御について理解させる。

(1) 割り込みとその制御

ハードウェアの持つ割り込み（interrupt）の機能と多重プログラミングを実現する機構を理解させる。

- ・割り込みの機構
- ・割り込みの種類
- ・割り込みの制御

(2) プロセスの制御

一般に並行的に遂行される計算（computation）を理解する上で重要な概念として、プロセスについての基本概念とオペレーティング・システムにおけるプロセスの制御の概要を理解させる。

- ・プロセスの概念
- ・プロセスとシステム資源
- ・プロセスの生成と消滅

(3) プロセッサの割当て

プロセスを計算機上で走らせるために必要な代表的な資源としてプロセッサ（processor）を考え、そのプロセスへの割り当ての考え方を理解させる。

- ・ディスパッチング（dispatching）の目的
- ・プロセスのスケジューリング・アルゴリズム

(4) プロセスの同期

並行的に動作し、全体として1つの仕事を遂行するシステムを考える場合、作業単位としてのプロセスとプロセス間の同期（synchronization）が重要な問題となる。これは、オペレーティング・システムだけにとどまらず、順次的（sequential）でない、並行的な（concurrent）プログラミングの観点からも重要な問題である。

- ・同期の必要性
- ・同期の問題
- ・同期の記述

5. システム資源管理

オペレーティング・システムの大きな機能の1つとして資源の管理がある。資源管理は、ジョブ、ジョブ・ステップ、プロセスなどオペレーティング・システムが対象とする種々のレベルにおいて行なわれる。ここでは、資源の管理の観点からオペレーティング・システムについて説明する。特に、最も重要な資源として主記憶管理（storage management）について説明し、仮想記憶システムの方法について触れる。

(1) システムの資源

ジョブ、ジョブ・ステップ、プロセスなどの実行に必要な各種の資源についての全般的な見方を理解させる。

- ・システム資源の分類
- ・システム資源とオペレーティング・システム

(2) 一般の資源割当て

システム資源の割当て（allocation）の考え方の概要を理解させる。

- ・静的な割当て
- ・動的な割当て

(3) 主記憶の管理

主記憶装置の割当ての概要を理解させ、

一般化しつつある仮想記憶システムでの
考え方を説明する。

- 記憶管理の概念
- 実記憶システムでの主記憶管理
- 仮想記憶システム

指導上の留意点

1. オペレーティング・システムは、多くの場合、計算機メーカーの作成によっており、オペレーティング・システムを眺める観点、重点の置き方、整理の仕方などさまざまである。一般化している代表的計算機メーカーの思想を理解させることも重要であるが、さらに柔軟な視野を持てるように指導したい。このためには、例えばMULTICSやUNIXなどを説明して視野を拡げる努力が必要である。
2. 他の章ではあまり取り上げる機会がない並行プロセス (concurrent process) についても触れるようにしたい。
3. オペレーティング・システムと計算機アーキテクチャの相互依存関係を折に触れて説明するのがよい。
4. プロセスの考え方は最近特に発展しつつある部分であり、このテーマに関してはプロセス同期問題など比較的新しい話題を取り入れて行きたい。

参 考

1. オペレーティングシステムの基礎

1.1 キーワード

オペレーティング・システム	operating system
制御プログラム	control program
入出力制御システム	IOCS: Input Output Control System
多重プログラミング	multiprogramming
スプーリング	spooling
アイドル・タイム	idle time
スループット	through put
ターン・アラウンド・タイム	turn-around time
RAS	Reliability-Availability-Serviceability
ジョブ	job
プロセス	process
タスク	task
ジョブ管理	job management
ジョブ制御言語	JCL: job control language
プロセス管理	process management
データ管理	data management
資源管理	resource management

1.2 オペレーティング・システムの歴史

オペレーティング・システムの発生の背景, その発展, 現状と動向についての概観を与える。

(1) 初期のアプローチ

入出力の為のプログラミングの面倒さを軽減し, 入出力機器を効率よく利用することを目的とした入出力制御システム (IOCS: Input-Output Control System) やジョブの連続処理 (stacked job processing) など初期のバッチ処理システムのためのオペレーティング・システムについて説明する。この時期の目的意識は,

- ・人間による操作を軽減し, アイドル・タイムを短縮する。

- ・CPUと入出力装置の処理速度の有効利用を目的とするスプーリング (Spooling: Simultaneous Peripheral Operation On Line)

などであることを説明する。

(2) 多重プログラミングの支援

本格的なオペレーティング・システムを特徴づける多重プログラミングの概念を理解させる。リアル・タイム・システムやタイム・シェアリング・システムなどの処理形態を実現する上でのプロセス管理の機能の重要性を説明し, プロセス管理を中核としてジョブ管理, データ管理などオペレーティング・システムの基本機能の捉え方が明確になってきたこと

を理解させる。

(3) 現状と動向

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を結合的に包括した高い信頼性を持つシステムとなっている。最近の方向としては、

- 仮想記憶 (virtual memory)
- 多重プロセッサ (multi-processor)
- 計算機ネットワーク (computer network)
- 分散処理 (distributed processing)

などの概略に触れる。

オペレーティング・システムは、従来かなり経験的に追求されてきた。最近は、並行プロセス (concurrent process) について、理論的研究が進み、プログラミング方法論の確立のきざしが見え、これがオペレーティング・システムの考え方、設計、実現方法に影響を与えていることも見逃せない。

1.3 オペレーティング・システムの目的

次のような視点からオペレーティング・システムの狙いとするところを説明し、基本的な概念を理解させる：

- (a) システム資源 (system resource) の管理
- (b) スループットの向上
- (c) ターン・アラウンド・タイムの短縮
- (d) 情報の共用と保護
- (e) RAS：信頼性 (reliability)，可用性 (availability)，保守性 (serviceability) あるいは RASIS：RAS に完全性 (integrity)，機密性 (security) を加えたもの

1.4 オペレーティング・システムの機能と構成

オペレーティング・システムの機能を体系的に分類し、ソフトウェア・システムとしての構成方法を考える上で一般的な見地が確定している訳ではない。また、従来、主にそれぞれの計算機メーカーが開発するシステムによってオペレーティング・システムの発展が支えられてきたことから、さまざまな観点があり、術語もかなり異なる。ここでは、比較的一般化している（しかし若干形式的な）見方により、次のように考える。

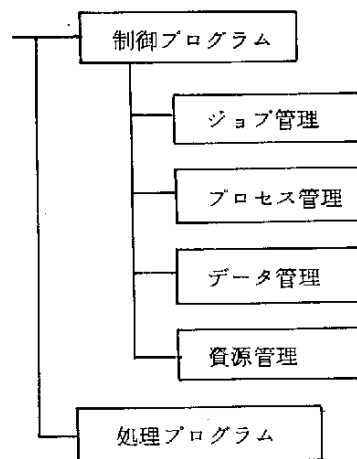


図-1 オペレーティング・システムの構成

オペレーティング・システムは粗く分けると、制御プログラムと処理プログラム (processing program) に分けられる。これは従来よく用いられてきた見方であるが、現在ではオペレーティング・システムといえは制御プログラムの方を指すのが普通である。第3節以降で学ぶのもこの意味でのオペレーティング・システムである。

• ジョブ管理

利用者のジョブを構成する ジョブ・ステ

ッブ (job step) を解析し、ジョブの スケジューリング (scheduling) を行い、必要な処理プログラムに処理を依頼する。

- プロセス管理

オペレーティング・システムが計算機上で行われる作業を管理する基本単位である プロセス の制御。プロセス (process) というかわりに タスク (task) という術語も同様の意味で用いられる。

- データ管理

入出力装置の制御を始め、ファイルの管理、各種のファイル編成法がアクセス法を提供する。

- 資源管理

CPU、記憶装置、入力装置などシステムの持つ各種の資源の各ジョブ、タスクへの割当てを行い、共用資源の効果的利用を管理する。

これらの機能の概要を、ジョブ、ジョブ・ステップ、プロセス (あるいはタスク)、ジョブ制御言語 などの基本的な概念と共に説明し、第3節以降に対する導入を行う。それぞれの教育環境で利用できる計算機によってオペレーティング・システムの概要を調べさせる。ミニ・コンピュータ程度の規模の計算機に対する汎用的なオペレーティング・システムを取り上げ、上のように分類した機能がどのような形で提供されているかを使用説明書などにより粗く調べるのがよい。

2. ジョブ管理

2.1 キーワード

ジョブ	job
ジョブ制御言語	JCL: Job Control Language
コマンド	Command
カタログ・プロシージャ	catalogued procedure
ジョブ制御マクロ	job control macro
ジョブ・スケューリング	job scheduling
到着順スケジューリング	FIFO(First-In First-Out scheduling) あるいは FCFS(First-Come First-Served scheduling)
優先順位スケジューリング	priority scheduling

2.2 ジョブの記述

オペレーティング・システムに対する作業の要求としてジョブを定義する手段を理解させる。適当なオペレーティング・システムのジョブ制御言語を例にとり、代表的なジョブ制御文(job control statement)を用いてジョブを記述する能力を身につけさせる。

(1) ジョブ制御言語

オペレーティング・システムにジョブを伝えるための言語は一般にジョブ制御言語と呼ばれている。利用できるオペレーティングシステムのジョブ制御言語により、次のような制御文について具体的に説明する：

- (a) ジョブの開始
- (b) ファイルの定義
- (c) プログラムの実行
- (d) データの記述
- (e) ジョブ・ステップの記述
- (f) ジョブの終了
- (g) 区切りや注釈のための文

これらの制御文で指示される属性、属性値の意味を理解させ、オペレーティング・システムによる処理にどのような意味があるかを説明する。

ジョブ制御文は、タイムシェアリング・システムでの端末からの指令のような場合にはコマンドと呼ばれることもある。又、プログラミング言語のように go to や条件分岐などの制御文を導入した言語もある。

(2) カタログ・プロシージャ

オペレーティング・システムの機能が大きくなり、利用者に対するサービスの種類が増えるにともない、ジョブ制御言語による記述は複雑化する。多くのオペレーティング・システムでは、カタログ・プロシージャあるいはジョブ制御マクロの機能があり、

- ・ジョブの記述を簡略化する
 - ・ジョブの記述を拡張する
 - ・定型的なジョブの記述を管理する
- などの目的で利用される。適当なオペレーティング・システムで、カタログ・プロシージャの定義方法、登録方法、呼出し方法を身につけさせる。

2.3 ジョブの遂行

オペレーティング・システムによるジョブの遂行のプロセスの概略を理解させる。

(1) ジョブの入力

ジョブは、システム入力装置や遠隔端

末から入力される。次について説明する。

- ジョブ入力装置の起動
- ジョブ入力のスプーリング
- ジョブ制御言語によるジョブ記述についての誤り検査
- ジョブ・スケジューリングのための待ち行列 (Job Queue)あるいはジョブ・スタック (Job stack)への登録

(2) ジョブのスケジューリング

2.4 で述べるようなスケジューリング方式により始動すべきジョブを選択する。

(3) 資源の割当て

選択されたジョブに対して静的 (static) な割当てを必要とする資源として、主記憶、専有する入出力機器、ファイルなどの割当てを行う。

ジョブに対して一括した割り当て方式やジョブ・ステップ単位でその都度必要な資源を割り当てる方式、ジョブ制御文でのパラメータとの関係などについて説明する。

(4) 各ジョブ・ステップの実行

実行されるジョブ・ステップの実行単位となるプロセスの生成とプロセス管理、動的 (dynamic) に要求される資源割当ての処理との関係を説明する。

(5) ジョブ・ステップの終了

ジョブ・ステップ終了の後処理として次について説明する：

- ジョブ・ステップ単位で割当てられている資源の解放
- 料金計算のための情報の収集
- ファイルの登録、抹消
- 次のジョブ・ステップの準備

(6) ジョブの終了

ジョブ終了の後処理として次について説明する：

- 割当てられている資源の解放

- 料金計算のための情報の収集
- ファイルや制御用領域の解放
- 出力用スタックを介してのジョブ出力の引渡し

(7) ジョブの出力

- ジョブ出力の形式、内容
- ジョブ出力のスプーリング

2.4 ジョブ・スケジューリング

待ち行列に入っているジョブから始動すべきジョブを選択する場合の考え方の概略を説明する。

スケジューリングを行う目的は、

- スループットの向上
- ターン・アラウンド・タイムの短縮

であることを理解させる。

スケジューリングのアルゴリズムは、現実的には、

- 到着順方式
- 優先順位方式

の2つである。優先順位方式では、各ジョブに対して

- ジョブ・クラス
- クラス内の優先順位

を指定させ、これを次のような要素と組み合わせて用いるのが普通である：

- 資源の競合状態
- ジョブの大きさ
- ジョブを始動すべき時刻
- ジョブを終了すべき時刻
- ジョブが待たされている時間

3. データ管理

3.1 キーワード

ファイル	file
ボリューム	volume
物理レコード	physical record
論理レコード	logical record
カタログ	catalogue
装置独立性	device independence
アクセス法	access method
順アクセス法	sequential access method
直接アクセス法	direct access method
基本アクセス法	basic access method
待機アクセス法	queued access method
編成法	organization method
順編成	sequential organization
直接編成	direct organization
索引順編成	indexed sequential organization
区分編成	partitioned organization
VSAM編成	VSAM(Virtual Storage Access Method) organization
ファイル制御ブロック	file control block

3.2 データ管理の基本概念

オペレーティング・システムでのデータ管理を理解する基礎として、2次記憶媒体の特性、ファイルの概念、データ管理システムの構造について説明する。

(1) 2次記憶媒体

次の点について種々の2次記憶媒体の性質を理解させる。

- (a) ボリューム／装置の種類、物理的構造
- (b) ボリュームの特性
- (c) 能力（容量、アクセス速度）
- (d) 記録形式

(2) ファイル

次の基本的事項について説明する。

- (a) ファイル
- (b) ボリューム
- (c) ラベル

(d) 論理レコード

(e) フィールド、フィールド値

(f) エクステンツ(extent)

(g) ブロック（あるいはページ）

(h) 物理レコード

(3) データ管理のシステム

オペレーティング・システムでのデータ管理は、ハードウェア・レベルでの入出力から各種のアクセス方式によるプログラム、インタフェースにいたる何等かの階層で構成されている。例えば次のような概念図(Donovan(1972))により、データ管理の機構の概略を理解させる。

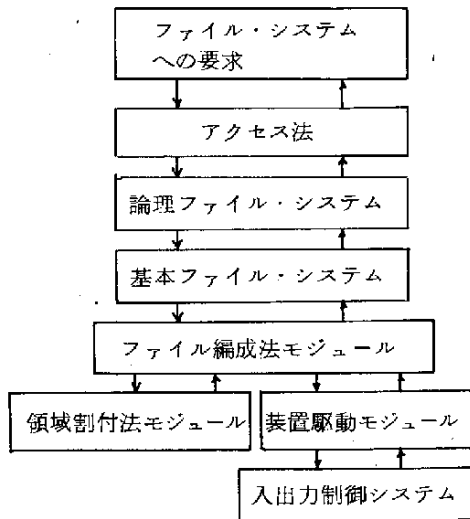


図-1 ファイル・システムのための階層

3.3 ファイル管理

2次記憶媒体上でデータを管理する基本単位であるファイルの管理について基本事項を理解させる。

(1) ファイルの割当て

次の点について説明する。

- (a) 内部ファイルと外部ファイルの対応
- (b) ファイルの割当て期間
- (c) ファイルの割当て時期
- (d) ファイル割当てのための指示

(2) ファイル・スペースの管理

2次記憶媒体上でスペースの割当て、解除、空きスペースの管理について説明する。

(3) ファイル・カタログ

オペレーティング・システムがファイルを管理する情報の集約であるファイル・カタログについて、その構造およびカタログを用いた。

- ・ファイルのアクセス
- ・世代管理
- ・機密保護

などについて説明する。

(4) 装置独立性

装置独立性の概念、目的、必要性を説し、これがファイル管理においてどのように実現されているかを説明する。

(5) ファイルの保護

(6) 機密保護

(7) ファイルの再編成と再配置

情報の追加、削除、更新などによりファイルの内部構造は徐々に変化する。この変化に伴う2次記憶スペースの利用効率の低下、アクセス効率の低下が生ずる。又、ファイルの生成、削除によってボリューム上のスペースが細分化される。これらに対処するためファイルの再編成および再配置が必要になることを説明し、このための手段を理解させる。

3.4 ファイルの編成とアクセス

ファイルによるデータ処理を支援するために多くのオペレーティング・システムは何種かのファイル編成法とアクセス法とを用意している。これは、

- ・利用者のさまざまなアプリケーションに適合した組織化されたデータ構造
- ・そのデータ構造に従ったデータ・アクセス機能

である。以下では次の事項について説明する。

(1) アクセス法

アクセス法について、次の2つの観点

(a) 順アクセス法と直接アクセス法

(b) 基本アクセス法(basic access method)と待機アクセス法(queued access method)

から説明する。

(2) 処理モード

(3) レコード形式

代表的な5つの形式について説明する。

- (a) 固定長非ブロック化
 - (b) 固定長ブロック化
 - (c) 可変長非ブロック化
 - (d) 可変長ブロック化
 - (e) 不定長
- (4) 編 成 法

現在多くのオペレーティング・システムで行なわれている次の編成法についてファイル構造と性質を説明する。

- (a) 順編成
- (b) 索引順編成
- (c) 直接編成
- (d) 区分編成
- (e) V SAM編成

こういった編成法はデータ処理のこれまでの経験から生れたものであり、2次媒体上でのデータ構造を考える上でただ一つの方法というわけではないことに注意すべきである。例えば、Harberman は次のような構成をあげている。

- 単純リスト構造
- 多重リスト構造
- 転置ファイル(invertedfile)構造
- 順ファイル構造

(5) プログラムとのインタフェース

プログラミングにおいて、以上のような編成法とアクセス法を用いる手段について概説する。

一般にマクロ命令として提供されているアセンブリ言語でのデータ・アクセス機能、FORTRANやCOBOLでのファイル処理機能について概略を把握させる。

3.5 入出力制御

プログラムから要求された入出力の指令がどのように遂行されるかの概要を理解させる。

次のような用語を説明し、ファイルがオープンされ、処理され、クローズされるまでの間に入出力動作がどのように遂行されるかを説明する。

- (a) ファイル制御ブロック (File Control Block)
- (b) ファイルのオープン (open)
- (c) ブロック化/非ブロック化
- (d) 緩衝域 (buffer)
- (e) チャネル・プログラム
- (f) 入出力割込み
- (g) ファイルのクローズ (close)

4. プロセスの管理

4.1 キーワード

割込み	interrupt
プログラム状況語	program status word
スーパーバイザ呼出し	supervisor call
優先順位	priority
プロセス	process
タスク	task
スーパーバイザ	supervisor
プロセス制御ブロック	PCB: process control block
資源の共有	resource sharing
スケジューリング・アルゴリズム	scheduling algorithm
プロセスの同期	process synchronization
同期問題	synchronization problem
同期基本命令	synchronization primitive
デッドロック(すくみ)	deadlock

4.2 割込みとその制御

ハードウェアが持つ割込みの機能を復習する。多重プログラミングの制御を行うための基本として、非同期的なできごとを組織的に制御する必要があること、そのために割込みがどのように働いているかを理解させる。

(1) 割込みの機構

次のような概念により、割込みの発生、プログラムの中断、割込み要因の解析、プログラムの再開のプロセスを理解させる。

- 割込み禁止状態
- ステート・ワード(state word)あるいはプログラム状況語(program status word)
- ステート・ワードの退避と復元

この機構により、種々の作業が並列に実行されているのと同様の効果が得られることを理解させる。

(2) 割込みの種類

- タイマ、入出力装置、他の計算機など

外部からの信号や要求によるもの

- スーパーバイザ呼出しによるもの
 - 計算機の誤動作、故障などによるもの
 - プログラムの誤りによるもの
- などの割込みについて、その意味と必要な処理の概要を説明する。

(3) 割込みの制御

- 優先順位
- 割込みマスク

について説明し、多重割込みの制御についての考え方を理解させる。

4.3 プロセスの制御

オペレーティング・システムにおける仕事の基本単位としてのプロセスについての基本概念およびプロセスの制御の概要を理解させる。

(1) プロセスの概念

オペレーティング・システムに限らず、一般に並行的に遂行される計算(computation)を理解する上で最も基本的な概念であるプロセス(あるいはタスク)の考え方を理解させる。ここでは、

- プロセス間でのシステム資源のアクセス
- プロセス間の通信の必要性
- 他のプロセスの生成、消滅
- プロセスの中断

といった面から、プロセスが必要とする性質を理解させる。

(2) プロセスとシステム資源

プロセスが計算機上で走行するために必要となる資源 (resource) を説明し、プロセスのおかれる環境を理解させる。

(a) 専有資源

- 専有機器：磁気テープ、端末、専有するディスクなど
- ソフトウェア：SRR (Serially Reusable Routine)
- プロセス制御ブロック (process control block)

(b) 共有可能な資源

- 本質的に共有可能なプログラムおよびファイル
- 時分割で共有可能なプロセッサ、入出力チャネルなど
- 空間分割で共有可能な主記憶装置、共用のディスクなど

これらのシステム資源をプロセスに割当て、各プロセスの遂行順を制御するのがスーパーバイザの役割であることを理解させる。

(3) プロセスの生成と消滅

(a) プロセスの状態推移

- RUN状態
- READY状態
- WAIT状態
- スワップ・アウト状態

(b) プロセス制御ブロック

プロセスを一意的に識別し、代表さ

せるためにはプロセス制御ブロック PCB (あるいはタスク管理ブロック、プロセス・ディスクリプタなど) と呼ばれるデータ構造を用いている。次のようなPCBの情報を示して、その役割を理解させる：

- プロセスの状態を定義する状態変数
- レジスタの退避領域
- プロセスが保持あるいは使用を宣言している資源に関する情報

(c) プロセスの生成／消滅

次の事項に基づき、一群のプロセスの協同によりジョブ・ステップが遂行される状況を説明する。

- ジョブ・ステップの開始にともなうプロセスの生成
- プロセスの正常終了
- プロセスの異常終了
- プロセスの生成／消滅のための要求 (ATTACH, DETACH マクロ命令など)

4.4 プロセッサの割当て

プロセスに対して割当てられる代表的な資源としてのプロセッサを取り上げ、その割当てについての考え方を理解させる。

実行待ち(WAIT)状態にあるプロセスが複数個ある時、これらのどのプロセスに次にプロセッサを割当てるかという問題は、プロセスあるいはタスクのディスパッチング (dispatching) と呼ばれ、システムの性能に大きな影響を及ぼす重要な問題である。狙いはスループットの向上と利用者の応答時間の要求を満足させることである。この観点から以下のような代表的なスケジューリング・アルゴリズムについて説明する。

(1) 優先度方式

- (a) プロセスの(静的な)優先度を用い

る方式

(b) 最短ジョブを優先する方式 (SJF: shortest-job-first)

(c) ダイナミック・ディスパッチング—入出力動作の多いプロセスを優先する

(2) ラウンド・ロビン方式 (RR: round-robin)

(3) フィード・バック方式 (FB: feedback)

4.5 プロセスの同期

(1) 同期の必要性

オペレーティング・システムのように多くのプロセスが互にほとんど独立に動作して全体として一つの仕事を遂行する場合、各プロセスは相互に直接的、間接的に連絡を取りつつ仕事を進めなければならない。並行的に動作するプロセスをどのように捉えるか、どのようにプログラミングするかを考える上で、プロセス間の同期 (synchronization) をどのように表現するかが重要である。次の2つの見方で同期の必要性を理解させる。

(a) プロセス間の協力

(b) プロセス間での競合の解決

(2) 同期の問題

次のような問題として定式化されている代表的な同期問題を説明する。オペレーティング・システムにおいてどのような場合にこれらの問題が現われるかを考えさせる。

(a) 相互排除問題 (mutual exclusion problem)

(b) 生産者-消費者問題 (producer-consumer problem)

(c) 読み書き問題 (readers-writers problem)

(d) 哲学者の食事の問題 (dining philosopher problem)

(3) 同期の記述

プロセス間の同期をプログラミングで表現する為の手段として、現在用いられている、あるいは提案されている代表的な同期基本命令について説明する。

(a) テスト・アンド・セット (test and set)

(b) POST/WAIT 命令

(c) セマフォ (semaphore)

(d) 条件つき危険部分 (conditional critical region)

(e) ENQ/DEQ 命令

その他、Hoare によるモニタ (monitor), Harbermann 等による順路式 (path expression) など、より高水準の記述方法についても概念を説明できるとよい。

また、同期のとり方によってはデッド・ロック (すくみ) が生じ得ること、デッド・ロックの検出や処理方法について触れる。

5. システム資源管理

5.1 キーワード

システム資源	system resource
共有資源	shared resource
資源の割り当て	resource allocation
デッド・ロック(すくみ)	deadlock
オーバーレイ	overlay
主記憶の割り当て	storage allocation
仮想記憶システム	virtual storage system
セグメント方式	segmentation
ページング方式	paging
ページ書換え	page replacement
局所性	locality

5.2 システム資源

ジョブ、ジョブ・ステップ、プロセスなどの実行には必要となる各種の資源についての全般的な見方を理解させる。

(1) システム資源の分類

資源の割り当ての観点から見るとき次のように分類される。

(a) 専有資源

- 専有機器：磁気テープ装置、端末、専有されるディスク装置など
- ソフトウェア資源：SRR(serially Reusable Routine)、プロセス制御ブロック、ジョブ制御ブロック(job control block)

(b) 共用可能な資源

- 本質的に共用可能な資源：プログラム、ファイルなど
- 時分割により共用可能な資源：プロセス、入出力チャネル、ディスク装置など
- 空間分割により共用可能な資源：主記憶、共用のディスクなどの領域

(2) システム資源とオペレーティング・システム

(1)のような各種のシステム資源に対す

る要求は一般に競合する。オペレーティング・システムは、これらの要求に対して矛盾のない、効率のよい資源の割り当てを行い、管理を行う必要がある。

次の点について説明する。

(a) システム資源割当ての時期

- 静的な割当て：ジョブ開始時
ジョブ・ステップ開始時
- 動的な割当て：プログラム実行中の必要時点

動的な割り当てほど使用効率はよくなりうるが、デッドロックなど複雑な制御が必要になることを理解させる。

(b) 資源の保護

共用が可能な資源に対しては又保護が必要である。次の2つの点から保護について説明する。

- アクセスの同時性が問題となる場合(排他制御)
- アクセスの権利が問題となる場合

(c) デッドロック

資源割当てで最も重要な問題の一つがデッドロックの可能性のあることである。デッドロックの定義、デッドロ

ックの生ずる条件、デッドロックの検出と回復あるいは防止法の概略を理解させる。

5.3 一般の資源割り当て

システム資源に対する割り当ての考え方の概略を理解させる。主記憶の割当てと管理については次節で多少詳しく説明する。

(1) 静的な割当て

ジョブ、ジョブ・ステップの開始時に行なわれる資源割当てについて説明する。

(a) 主記憶

① 実記憶方式のシステムの場合

- ・パーティション方式：ジョブ開始時に最大のジョブ・ステップ分を確保する。
- ・クォージョン方式：ジョブ・ステップ単位に必要なサイズを確保する

② 仮想記憶方式のシステムの場合

(b) ファイル

ファイルの割当ては一般にJCLで要求される。次の2つの場合を区別する必要がある：

① 既存ファイルの使用要求

論理的には排他制御だけを考えればよい。物理的には対象ボリュームがマウントされていない場合に装置割当てを考えなければならない。

② 新しいファイルの定義の要求

装置割当ての問題およびディスク領域の割当てが発生する。

(c) 装置

① ジョブのスケジュールの前／後に確保

- ・全ジョブ・ステップの要求数の和
- ・各ジョブ・ステップ中の最大要求数

② ジョブ・ステップ開始時に確保

(d) 割当ての順序

割当ての順序に関しては

- ・一括割当て
- ・多段割当て

がある。また、必要資源が一括しては割当てられない場合の対処法として

- ・一部分でも確保してしまう
- ・全部確保できるまで全然確保しないなどの考え方がある。

(2) 動的な割当て

(a) ファイル、装置、ディスク領域

多くのシステムではプログラム実行中にマクロ命令などで割当てを要求できる。JCLで予約だけしておき必要時点で割当てを受ける方式と予約を必要としない方式がある。

(b) 主記憶

次の3つの事項について説明する。

① 基本的に静的な割当てを行っている場合

マクロ命令などにより割当てを要求できる。

② 仮想記憶方式でのページ置換え

③ スワッピング

プロセスまたはジョブ・ステップを単位として主記憶と2次記憶の間で出し入れを行う。

5.4 主記憶の管理

最も重要なシステム資源である主記憶装置の割当ての概要を説明し、最近一般化している仮想記憶システムでの考え方を理解させる。

(1) 記憶管理の概念

多重プログラミングのシステムでは個のプロセスに与えられる主記憶は限られていること、またプログラムの大規模化などに対処するため、主記憶の効率的な管理がオペレーティング・システムの

機能として重要であることを理解させる。
人為的な手段であるオーバレイ構造を用
いる方法と次のような問題点

- プログラムの作成が複雑化し、生産性
が低い
- オーバレイされる最大の単位以上の記
憶域は必要になる
- プログラムの実行開始前にプログラム
の動的構造を決定しなくてはならない
- データ領域の動的な拡大、縮小に対処
できない

を説明し、記憶管理をオペレーティング
・システムが自動的に行う仮想記憶シス
テムに発展してきた流れを理解させる。

(2) 実記憶システムの主記憶管理

次のような管理方式について説明する。

(a) 単一連続割当て方式

多重プログラミングを考えない計算
機で用いられる。

(b) 固定区画方式

多重プログラミングのための最も単
純な方式で、主記憶領域を複数の固
定長の区画に分割し、各ジョブを大き
さに応じて適当な区画で実行させる。

(c) 可変区画方式

ジョブ開始時に、主記憶の区画の大
きさをジョブの大きさに合わせて決定
する。稼動にともない、利用可能領域
がこま切れに細分化されるフラグメン
テーション(fragmentation)の問
題が生ずる。フラグメンテーションを
防ぐためには、一定期間ごとに細分化
された領域を一つの領域にまとめるコ
ンパクション(compaction)の操作
が必要になる。

(3) 仮想記憶システム

(a) 仮想記憶の実現方法

代表的な方法としてセグメント方式
とページング方式について次のような
概念により説明する。

① セグメント方式

- プログラム・データの論理的な単
位としてのセグメント(segment)
- 論理アドレスから実アドレスへの
変換のためのセグメント・テーブ
ル
- セグメント変換例外割込み

② ページング方式

- ページ(page)
- 論理アドレスが実アドレスへの変
更のためのページ・テーブルと動
的アドレス変換機構(DAT:
dynamic address transl-
ator)
- ページ置換え

(b) 仮想記憶システムでの主記憶管理

① プログラムの動的局所性

プログラムの実行中に参照される
アドレスの集合に見られる局所性
(locality)について説明し、局
所性を利用する1つのモデルとして
ワーキング・セット(working
set)の考え方を説明する。

② 置換え

主記憶領域が要求された時、未使
用領域が不十分であれば主記憶上の
プログラムを2次記憶に追出し主記
憶に空きを作らなければならない。
この時置換えるべき領域をどう選択
するかがシステムの性能に重要な影
響を及ぼす。次のような置換えアル
ゴリズム(replacement algo-
rithm)について説明する。

- F I F O方式 (first-in first-out)
- L R U方式 (least recently used)
- ランダム方式

参考文献

- (1) P. Brinch Hansen : 「The Nucleus of a Multiprogramming System」 CACM 13. 4. 238—241, 1970
- (2) A. C. Shaw : 「The Logical Design of Operating Systems」 Prentice-Hall, 1974
- (3) D. C. Tsichritzis, P. A. Bernstein 著, 齊藤信男訳「オペレーティング・システムの基礎」日本コンピュータ協会, 1976
- (4) 齊藤信男「OSの基礎理論」情報処理, Vol. 15. 1974
- (5) A. N. Habermann : 「Prevention of System Deadlock」 CACM 12, 7, 373—377, 1969
- (6) C. A. R. Hoare : 「Monitors : An Operating System Structuring Concept」 CACM 17, 10, 549—557, 1974
- (7) P. J. Curtois, F. Heymans and D. L. Parnas : 「Concurrent Control with Readers and Writers」 CACM 14, 10, 667—668, 1971
- (8) Tsichritzis and Bernstein : 「Operating Systems」
- (9) ハーバマン (Habermann, A. N.) 土居範久訳「オペレーティングシステムの基礎」培風館 1978
- (10) 情報処理学会編「情報処理ハンドブック」オーム社, 1980
- (11) P. Brinch Hansen : 「A Comparison of Two Synchronizing Concepts」 Acta Informatica 1, 3, 190—199, 1972
- (12) P. B. Hansen : 「Operating System Principles」 Prentice-Hall, 1973

第11章 データコミュニケーション (C10)

キーワード

データ通信	data communication
通信制御装置	communication control unit
通信回線	communication line
端末装置	terminal device
変復調装置 (モデム)	modem
変調速度 (ボー)	modulation rate (baud)
データ伝送路	data transmission line
同期式/非同期式	synchronous/asynchronous
変調	modulation
半二重通信方式	half duplex
全二重通信方式	full duplex
分岐接続	multi drop
リアルタイム処理	real time processing
リモートバッチ処理	remote batch processing
会話型処理	conversational processing
メッセージ交換	message switching
タイムシェアリング	time sharing
伝送制御手順	transmission control procedure
コンテンション	contention
ポーリング/アドレッシング	polling/addressing

目 標

1. データ通信にかかわる基本的な概念を理解させる。
2. データの通信速度、代表的な通信方式と伝送制御方式についての知識を身につけさせる。
3. コミュニケーション・システムの構成と機能を理解させる。さらにデータ通信路についての知識を身につけさせる。
4. データの送受信と通信方式について、一般的に広く使用されている概念を正しく理解させる。電電公社と国際電電のデータ通信回線サービスの概要を理解させる。
5. 通信回線の接続形態と一般的なネットワークの形態を理解させる。
6. リアルタイム処理、リモート・バッチ処理、会話型処理などの代表的なコミュニケーション・システムの処理形態について、機能と特性を理解させる。
7. データ制御手順と伝送制御符号について理解させる。

8. 伝送路における誤り検出と誤り回復方式について理解させる。
9. コミュニケーション・システムの評価の方式と評価の手段として使われるシミュレーションについて理解させる。
10. コミュニケーション・システムのソフトウェアをオペレーティング・システムに含まれるデータ通信機能と関連づけて理解させる。

このコースを学ぶときは、オペレーティング・システムの知識が必要である。

内 容

1. データ通信の基礎

(1) データ通信システムの概要

データ通信の定義を明確にし、データ通信システムを構成する機器として、コンピュータ、通信制御装置、変復調装置（モデム）、通信回線、端末装置の概要を説明する。

(2) 情報の符号化

JISコードを中心にして、データ通信で用いられるコード体系を説明する。

(3) 通信速度と通信方式

変調速度（ボー）、データ信号速度（BPS）、データ伝送速度（字分）について説明する。

単向通信、半二重通信、全二重通信の3つの通信方式を簡単に説明する。直通方式と分岐方式、伝送制御方式について、例をあげて簡単に説明する。

2. コミュニケーション・システムの基本構成

(1) コミュニケーション・システム

次の5つの要素について、各々の構成と機能、相互の関係を詳しく説明する。
コンピュータ・システム

通信制御装置

通信回線

データ通信用装置

端末装置

データ伝送路の種類と構成、2線式回線と4線式回線、直流伝送と交流伝送を説明する。

(2) データの送受信と通信方式

同期方式について、調歩同期方式と独立同期方式を説明する。変調方式について、振幅変調、周波数変調、位相変調を説明する。モデム・インタフェースについて説明する。

通信方式について、単向通信方式、半二重通信方式、全二重通信方式の原理と適用分野を説明する。

電電公社と国際電電のデータ通信回線サービスの概要を説明する。

3. コミュニケーション・システムと処理形態

(1) 通信回線の接続形態

直通接続（ポイント・ツー・ポイント）方式、分岐接続（マルチ・ドロップ）方式、交換接続方式とその複合形態について、特徴、利点、問題点を説明する。

データ通信の集中型、環状型、網状型ネットワークを説明する。

(2) コミュニケーション・システムの処理形態

リアルタイム処理、リモート・バッチ処理、会話型処理の特性を理解させる。
リアルタイム処理では、問合せ／応答型処理、メッセージ交換型処理、データ収集型処理、データ分配型処理を説明する。
タイムシェアリング・システムの機能と特性を説明する。

4. 伝送制御

(1) 伝送制御手順

データ・リンクの定義を明確にし、伝送制御手順として、回線接続、データ・リンクの確立、情報転送、終結、回線切断の5つの段階を説明する。

(2) 基本データ伝送制御手順

コンテンション方式とポーリング／アドレッシング方式を説明する。

10種類の伝送制御符号の意味と使い方を説明する。

5つの標準的なメッセージ形式を説明する。

5. 誤り制御

伝送路で生じた誤りを検出して訂正する方法を説明する。誤り検出方式としては、2度送り方式、エコー・チェック方式、パリティ・チェック方式、群計数チェック方式、サイクリック・リダンダンシィ・チェック方式を説明する。誤り訂正方式としては、再送方式と自己訂正方式を説明する。

6. コミュニケーション・システムの分析

(1) コミュニケーション・システムの評価
回線使用時間と回線使用率の定義と求め方を説明する。

コミュニケーション・システムの隘路（ボトルネック）を説明し、隘路における待ち状態を分析することの重要性を理解させる。

待ち行列理論による評価の方法を説明する。端末装置における待ち行列、通信回線の待ち行列、コミュニケーション・システムを構成する各装置における待ち行列と処理時間の算出を、具体的な例を用いて説明する。

(2) シミュレーション

評価の手段として、シミュレーションという有効な手段があることを説明する。

シミュレーション・プログラムの種類、汎用シミュレーション言語、アウトプットの利用の仕方などを説明する。

7. データ・コミュニケーション・システムのソフトウェア

オペレーティング・システムに含まれるデータ通信機能とその活用の仕方を説明する。

アセンブリ言語のマクロ命令を用いた、簡単な回線制御プログラムの例を示す。

プログラム・コードの共有と並行処理の必要性とその仕組みを例をあげて説明する。

指導上の留意点

1. 銀行のオンライン・システム、航空会社の座席予約システム、一般的なメッセージ交換システムなど、よく知られている代表的な例を取りあげて、データ通信システムの構成と各構成要素の役割を具体的に説明する。
2. 用語については、少なくともJISの情報処理用語に定義されているものを、体系づけて一通りその概念を説明する必要がある。
3. データ通信については、一般になじみの薄い用語が多いので、できるだけ現物を映した映像や、分析図などを用いてわかりやすく説明する必要がある。
4. データ・コミュニケーションについて要求される知識は、ハードウェアの知識が中心になる。システム設計やプログラミングの詳しい内容については、このコースが目標とする水準を超えるので、あまり深く立ち入る必要はない。
5. このコースは、「オペレーティング・システムとアーキテクチャ」を学んだ人を対象にしているが、「プログラミング」の次

にこのコースを学ぶこともできる。

参 考

1. 用語の説明

データ通信

リンク規約に従って、一つ以上のデータリンクを経由して、データ送信装置及びデータ受信装置間で、データを転送すること。

通信制御装置

データ回線を介して伝送されるデータの授受に関する制御を行う装置。

通信回線

データ回線という。

データ回線とは、二つのデータ端末装置間のデータ伝送の手段。

端末

利用者がデータ処理システムと交信するための入出力装置。

変復調装置

モデムともいう。信号を変調及び復調する装置。例えばデジタルデータをアナログ伝送設備を介して伝送できるようにする。

変調速度

変調された信号の次に現われる有意瞬間との間で最も短い公称時間間隔の尺度の逆数値。この測定値が秒で表わされれば、変調速度はボーで与えられる。

データ伝送路

データ伝送回線ともいう。離れた場所へ信号を転送するための媒体。

変調

直流信号を交流信号に変えること。

半二重伝送

一度には片方向であるが、どちらの方向へも伝送可能なデータ回線上でのデータ

伝送。方向の選択はデータ端末装置により制御される。

全二重伝送

同時に両方向へ伝送可能なデータ回線上でのデータ伝送

分岐接続

三つ以上のデータステーション間で確立される接続。この接続には交換装置を含むことがある。

リアルタイム

実時間ともいう。計算機外部の別の処理と関係を持ちながら、かつ外部の処理によって定められる時間要件に従って、計算機の行うデータ処理に関する用語。

リモートバッチ処理

遠隔一括処理ともいう。入出力装置がデータリンクを介して計算機にアクセスする一括処理。

会話型

会話様式、対話様式ともいう。利用者とシステムとの間で交互に取り交わされる投入及び応答が、二者間の対話と同様の手段でなされるような計算機システムの操作様式。

メッセージ交換

データ網の中で、完全なメッセージを受信、記憶及び送り出すことにより、メッセージの経路指定を行う処理。

タイムシェアリング

時分割ともいう。一つの処理装置において、二つ以上の処理を時間で交互配置させるようにする計算機システムの操作技法。

コンテンション

二つ以上のデータステーションが、同時に共用の通信路を介して転送しようとしたり又は、片方向選択通信で、二つのデータステーションが同時に転送しようと

した時に発生する状態。

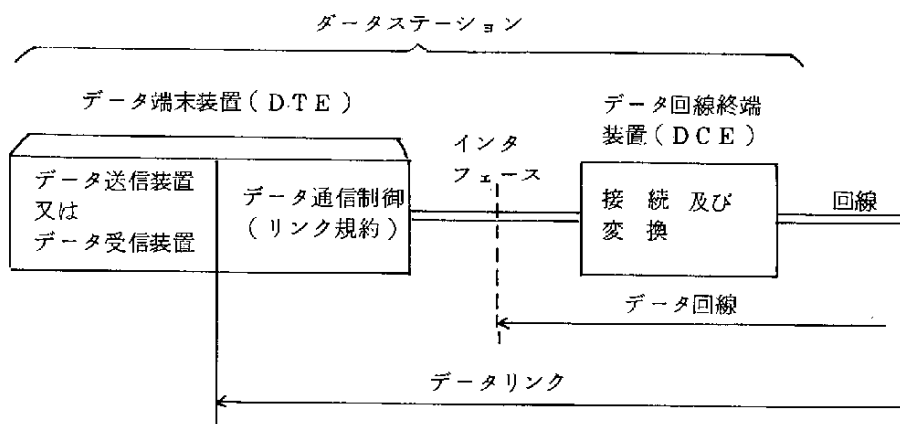
ポーリング

データステーションが一つずつ送信を促
がされる処理。

アドレッシング

正式にはセレクトイングという。一つ以
上のデータステーションに受信を促す処
理。

基本用語の相関関係

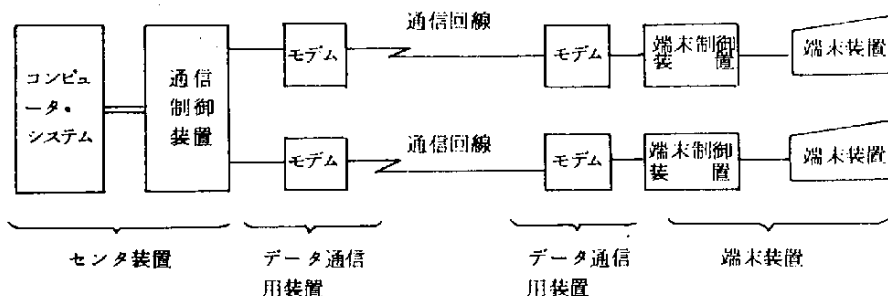


○印だけが使用できる。

文 脈		データ通信	データ伝送	
		データリンク	データ回線	通 信 路
伝送方式	全二重		○	
	半二重		○	
	単 向		○	○
使用方式	両 方 向	○		
	方向選択	○		
	片 方 向	○		

2. コミュニケーション・システムのハードウェア構成
- コミュニケーション・システムは、次のようなハードウェアで構成される。
- コンピュータ・システム

通信制御装置
通信回線
データ通信用装置
端末装置



(1) コンピュータ・システム

コミュニケーション・システムで使用するコンピュータ・システムのハードウェアには次のようなものがある。

中央処理装置

システム制御装置

磁気テープ装置などの周辺装置

また、オンライン・システムの適用業務によっては、システムの信頼性を高めるために次のようなシステムを採用する必要もある。

デュプレックス・システム

デュアル・システム

多重プロセッサ・システム

(2) 通信制御装置

通信制御装置は、次のような役割を行う。

- コンピュータの内部処理速度とデータ送信速度の差を吸収する。

- 並列から直列、直列から並列への変換を行う。

- 通信回線上のデータの多重処理

- 通信回線、変復調装置の制御

通信制御装置には、単純なものから、記憶装置や処理機構をもち、中央処理装置の機能を一部代行できるものもある。

一般に、通信制御装置には、次のような機能が必要である。

文字の組立てと分解

バッファリング

メッセージ処理

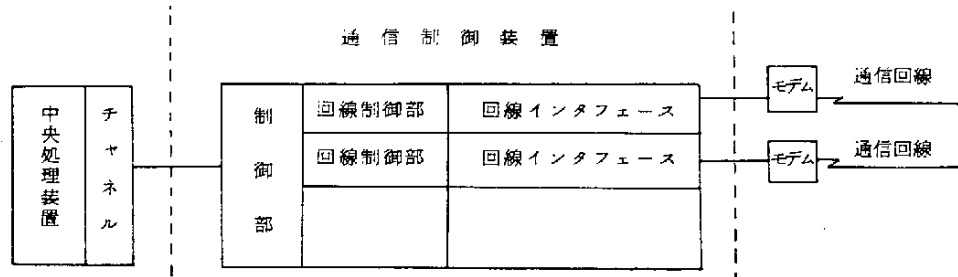
誤り検出、誤り制御

コード変換

伝送制御

回線制御

通信制御装置は、次のような構成になっている。



(3) データ通信用装置

必ず設置しなければならないのは、モデム（変復調装置）である。モデムは、通信制御装置あるいは端末制御装置から受け取った直流信号を交流信号に変えて（変調）、通信回線に伝え、また通信回線から受け取った交流信号を直流信号に変えて（復調）、端末制御装置や通信制御装置に伝える。

他に、中継装置や交換回線を使用する場合のネットワーク制御装置、1本の通信回線に複数の端末装置を接続する場合の分岐装置などもある。

(4) 端末装置

端末装置には入出力機能を持つだけのものと、演算機能や記憶機能を備えたインテリジェント端末などもある。

3. データの送受信

(1) データ伝送路

データ伝送路は、例外的に利用者が自

ら通信回線を設置してそれを使用する場合もあるが通常は、日本電信電話公社の専用回線又は交換回線を使用する。

通信回線には、2線式のものと4線式のものとがある。

データ伝送路上で情報を伝送する場合、一般には交流信号に変換して送るが、直流で送る直流電送もある。

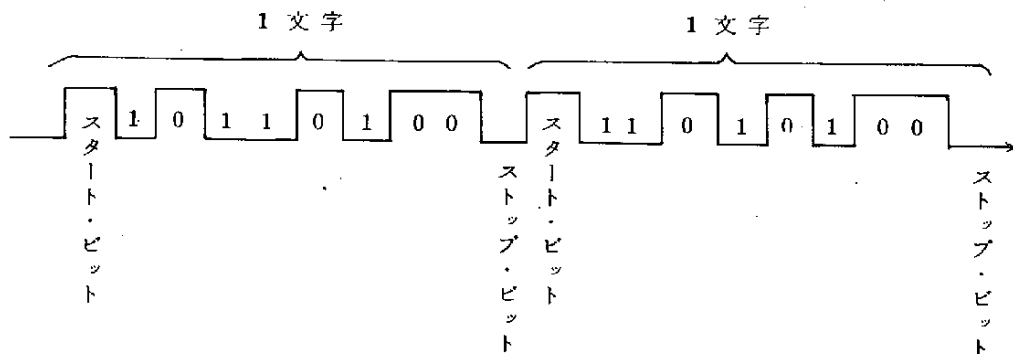
また、データ伝送路は、日本電信電話公社の回線を使用する建物外の部分と、構内回線の部分がある。構内回線の部分は私設回線である。

(2) 同期方式

調歩同期式（調歩式）と独立同期式（同期式）がある。

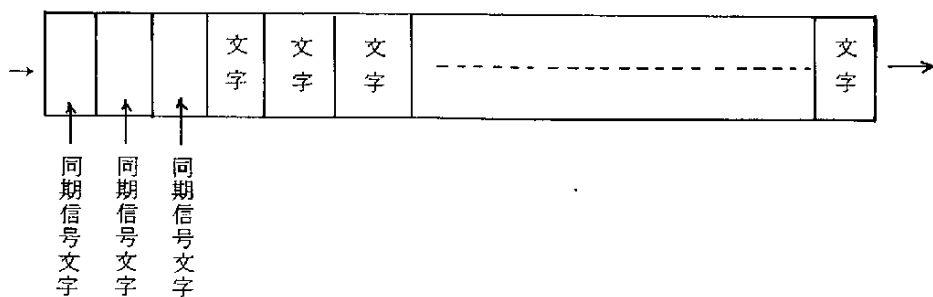
(a) 調歩同期式

この方式では、通信回線上を伝送する1文字（通常8ビット）の前と後にスタート・ビットとストップ・ビットをつけ、送受信間で1文字ずつ同期をとりながら伝送する。



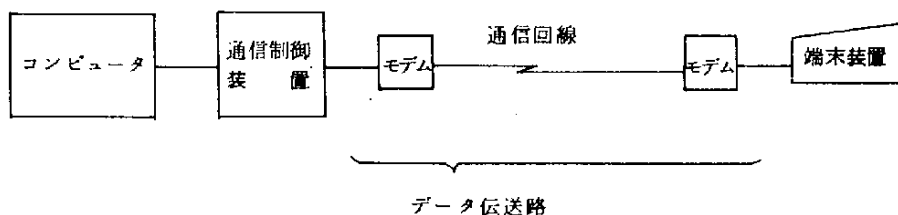
(b) 独立同期式

この方式では、1文字ごとに同期符号を付加せず、データ・ビットは直列に連続して伝送する。同期をとるためには、同期信号文字を送信する。



(3) 変調方式

直流信号を交流信号に変換する変調方式には、三つの方式がある。



(a) 振幅変調

技術的には簡単であるが、外界の影響を受けやすいので、あまり利用されない。入力信号の1と0を交流に変えるとき、電気信号の1に対応する振幅と0に対応する振幅を変えて電送する。

(b) 周波数変調

これは、電気信号の周波数をかえて、入力信号の1と0を電送する。

(c) 位相変調

これは、入力信号の1と0に対応し

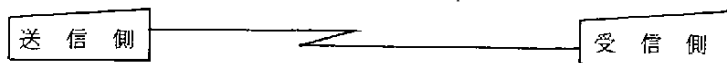
て、搬送波を変えて伝送する。

位相変調は、外界の影響に強く、周波数の帯域も少なくすむ。

(4) 通信方式

(a) 単向通信方式

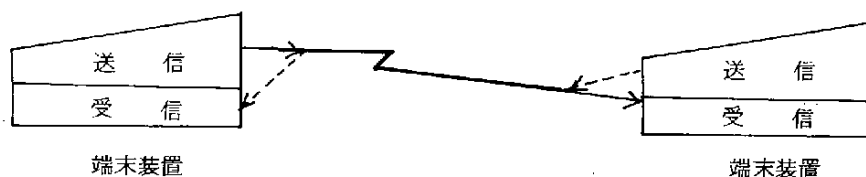
これは、送信側と受信側の2点間を次のような一対の通信回線で接続し、送信側と受信側を固定した形で行なう。



(b) 半二重通信方式

これは、次のように2方向の通信を

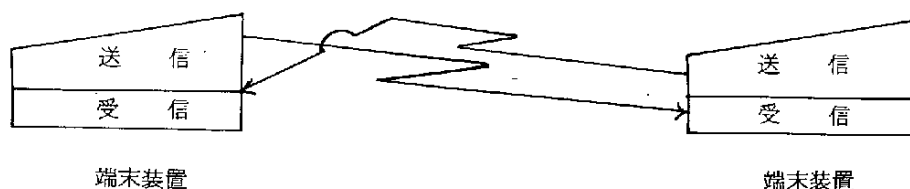
交互に行うデータ伝送方式である。



(c) 全二重通信方式

同時に行うデータ伝送方式である。

これは、次のように2方向の通信を



4. 日本電信電話公社のデータ通信回線サービス

は、特定通信回線と公衆通信回線がある。

特定通信回線には次のような種類がある。

(1) 回線の種類

電電公社が提供するデータ通信回線に

規 格	種 別	内 容
A 規格	A-1	50 BPS 符号伝送 (直流伝送)
B 規格	B-1	100 BPS 符号伝送 (直流伝送)
C 規格	C-2	200 BPS 符号伝送 (モデム直営)
D 規格	D-1	300 ~ 3400 Hz 帯域使用 (モデム自営)
	D-5	1200 BPS 符号伝送 (モデム直営)
	D-7	2400 BPS 符号伝送 (モデム直営)
	D-9	4800 BPS 符号伝送 (モデム直営)
I 規格	I-1	48 KHz 帯域使用 (モデム自営)
	I-3	48 Kbps 符号伝送 (モデム直営)
J 規格	J-1	240 KHz 帯域使用 (モデム自営)

公衆通信回線には次のようなものがある。

種 別	内 容
電 話 型	加入電話交換網を使うもの
電 信 型	加入電信交換網を使うもの

他に、東京、大阪で48KBPS交換回線サービスが提供されている。

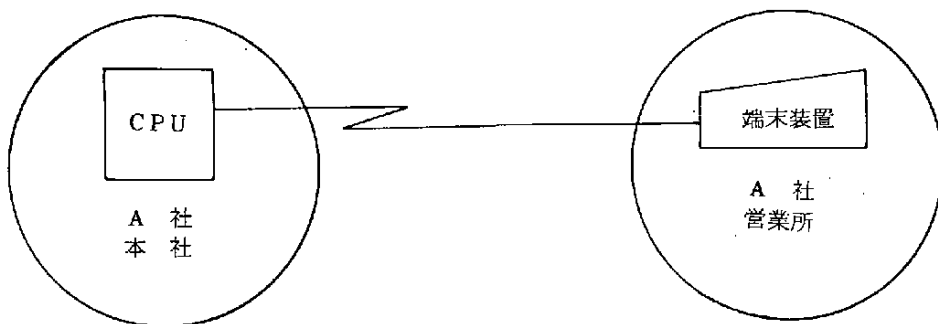
(2) 利用形態

利用形態には、本人使用、他人使用、

共同使用がある。

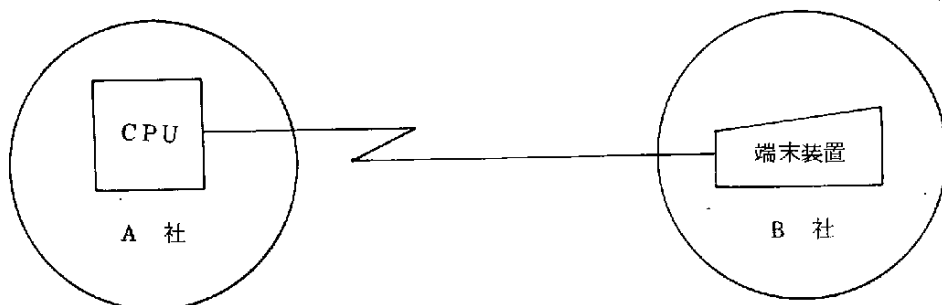
(a) 本人使用

センター側も、端末側も同一のユーザが使用する形態である。



(b) 他人使用

これは、電電公社や国際電電の回線を使用して、第三者のデータ処理サービスを行う形態である。



但し、他人使用には利用上の形態に制限がある。

- 国際特定通信回線では、CPUで処理された結果が入力された端末装置に戻されるシステムしか認められていない。

- A社がB社とC社の販売在庫管理のデータ処理を請け負って行うというような場合は、B社とC社の間に業務上相当な関係が必要である。この場合、A社のCPUがメッセージ交換をしていなければならない。

- この場合A社のCPUが複数台あれば、どのCPUで処理してもよいが、途中のCPUでも、何らかのデータ処理が必要である。単なるメッセージ交換にCPUが使われることは認められない。

- また、処理された結果を入力した端末装置が収容されているCPUと異なるCPUに接続されている端末装置に出力することはできない。

(c) 共同使用

共同使用は、複数の法人等が共同し

共同使用には、基準認可と個別認可があ

基準認可	利用形態上のもの	CPUで処理した結果が、必ず入力した端末装置に戻される一つのCPUと一つの端末装置に終始する場合。
	業務上の関係によるもの	郵政省令で定める業務上の関係があると認められる場合。
個別認可		上記基準認可に該当しないもので、郵政大臣の個別認可を受ける場合。

共同使用においても、メッセージ交換はいっさい認められない。

(3) 認定、適合審査

電電公社の回線に自営の端末装置を接

続する場合は、電電公社の技術基準に基づいた認定、審査を受けなければならない。認定、審査の種類、内容は次のとおりである。

適合審査	機器審査	個別審査	電子計算機，端末機，モデム，ファクシミリ など網制御，通話機能を持たないもの。
		型式審査	
	接続機能審査		
認定	個別認定		網制御装置，電話などの網制御や通話機能 に関するもの。
	型式認定		

また、利用者が回線を申請し、それが電電公社から提供されるとき（回線開通）、開通検査と機器検査が行われる。

(4) 関連法規

データ通信に関連する法規には、

公衆電気通信法

有線電氣通信法

があり、これに基づいてサービスが提供される。

(5) 電電公社の新データ網サービス

電電公社は、新たに次のような回線交換サービスとパケット交換サービスを開始した。

(a) 回線交換サービス

回線交換サービスは、現在の公衆通信回線がデジタル化されたもので、

これには200BPS以下、300BPS以下、1200BPS以下、2400BPS、4800BPS、9600BPS、48KBPSの各クラスがある。

(b) パケット交換サービス

パケット交換サービスは、データをパケットと呼ばれるブロックに区切って送受信する蓄積交換の一種である。パケット交換の方式としては、バーチャル・コートとパーマネント・バーチャル・サーキットの2種類がある。

新データ網のメリットは、回線品質が良く、接続時間が短く、合理的な料金体系である。

また、閉域接続、ダイレクト・コール

料金のセンター一括払い、ID信号サービスが可能である。

5. 国際電信電話公社のサービス

(1) 回線の種類

国際電電が提供するデータ通信回線には、国際特定通信回線と国際公衆通信回線がある。

(a) 国際特定通信回線

ク　ス　ス		速　　度
Voice Grade		使用する自営モデムによる。 国際電電も2400BPSモデムを提供する。
200 Band Speed		200 BPS
100 Band Speed		100 BPS
75 Band Speed		75 BPS
50 Band Speed	Full Speed	50 BPS
	Half Speed	20 BPS
	Quarter Speed	12.5 BPS

(b) 国際公衆通信回線

国際テレックスによる電信型のサービスである。

査があり、電電公社の場合と同じ法規が適用される。

(2) 利用形態

国際電電の回線の場合も、本人使用、他人使用、共同使用の利用形態がある。

他人使用の場合を除き、内容は電電公社の回線と同じである。

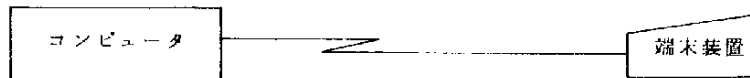
国際電電の場合も、自営機器の認定審

6. 通信回線の接続形態

(1) 直接接続（ポイント・ツー・ポイント）方式

コンピュータと端末装置が通信回線を介して次のように1対1で接続される。

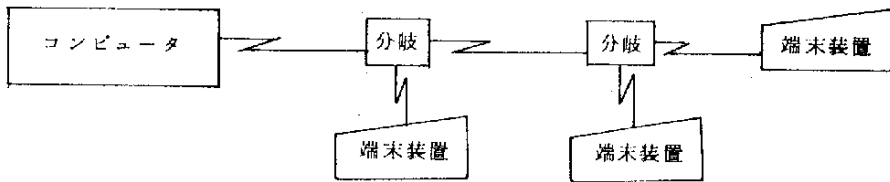
データ量が多い場合に有効である。



(2) 分岐接続（マルチドロップ）方式

次のように、一つの通信回線を複数の端末装置が共同で利用する。

データ量が少ないときは、この方式を用いれば、回線を経済的に有効に利用できる。

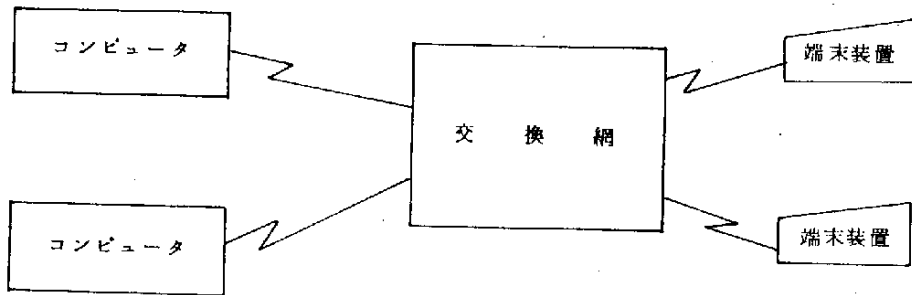


(3) 交換接続方式

次のように交換網を利用して、必要なときだけ、コンピュータから端末又は端末からコンピュータを呼び出して相互の接続を確立する。

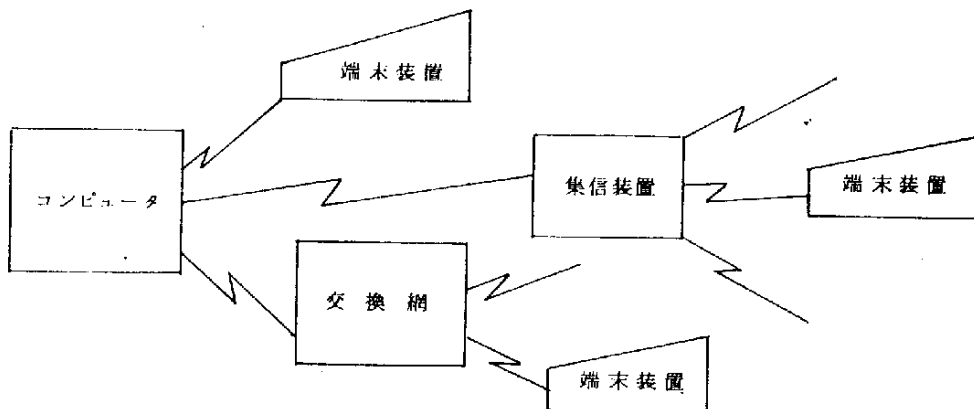
通信回線費用を低減できるし、任意の相手との接続に有効である。

交換網としては、電電公社の加入電話交換網と加入電信交換がある。



(4) 複合形態

以上の他に、コンピュータと端末装置の間に、集信装置を設置して、通信回線の集約を行う方法がある。



(5) ネットワークの形態

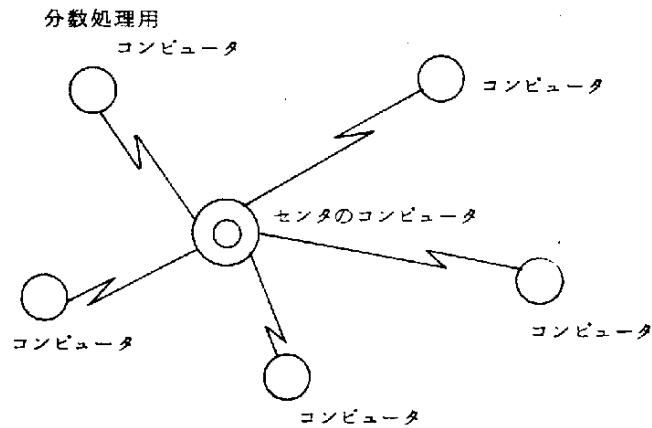
現在では、従来のセンタ集中型のシステムに代わって、分散形システムが増加しつつある。

分散形システムの場合、小型コンピュータが各地に点在し、センタのコンピュ

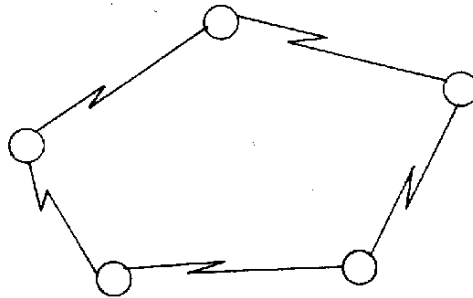
ータの機能を一部代行することになり、コンピュータのネットワークが形成される。

コンピュータ・ネットワークには、次のような形態がある。

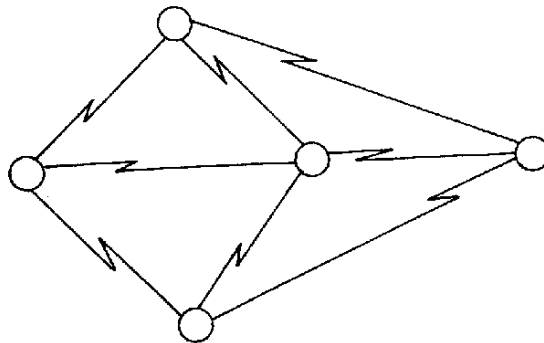
(a) 集中型



(b) 環状型



(c) 網状型



7. コミュニケーション・システムの処理形態

(1) リアルタイム処理

銀行のオンライン・システムやみどりの窓口に見られる座席予約システムなど、即時性、広域性の必要な分野で使用される処理形態である。

リアルタイム処理には、次のような四つの代表的な形態がある。

(a) 問合せ応答型処理

端末装置からデータを中央のコンピュータに入力し、それを即時に処理して同一の端末に出力する。

(b) メッセージ交換型処理

端末装置から入力された情報を交換

局の役目を果たすコンピュータを介して、他の端末装置へ出力する。

(c) データ収集型処理

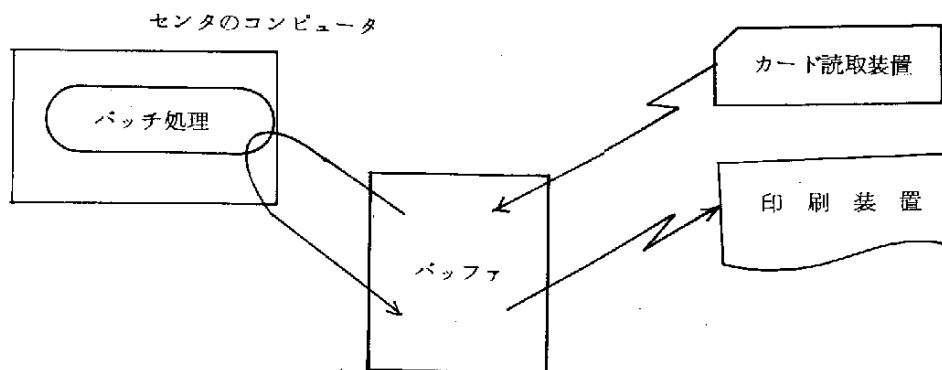
端末装置からのデータをセンタのコンピュータに集める処理で、収集されたデータは、一般にオフラインで一括処理される。

(d) データ分配型処理

センタのコンピュータで処理した結果を端末装置へ出力する。

(2) リモート・バッチ処理

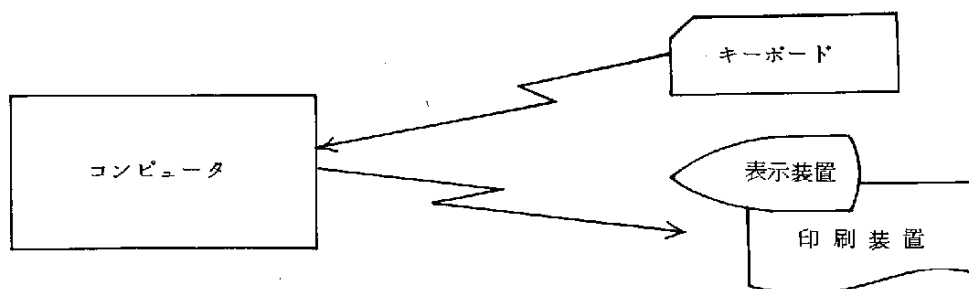
リモート・バッチ処理とは、遠隔地からバッチ処理を行う処理形態である。そのコンピュータ室で行うバッチ処理と同質のサービスを遠隔地のユーザに提供する。



(3) 会話型処理

会話型処理とは、人が端末装置を使用

して中央のコンピュータと直接、対話しながら作業を進めていく形態である。



会話型処理では、端末装置からプログラムを入力することができる。

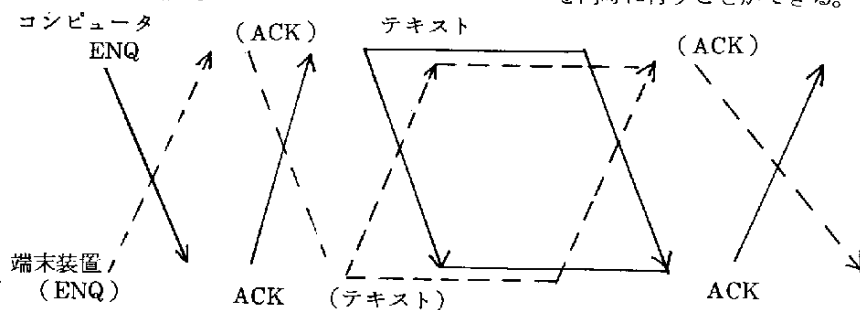
リアルタイム処理では、データだけを入力するので、この点がリアルタイム処理と異なる点である。

(4) タイムシェアリング・システム(TSS)

会話型処理機能を持つシステムは、通常的にはタイムシェアリング・システムである。

タイムシェアリング・システムには、次のような機能が必要である。

- 待ち行列の制御
- 自動スケジューリング機能
- 応答時間の保証
- 利用形態による優先度処理
- 機密保護機能



(2) ボーリング方式

ボーリング方式は、主としてマルチドロップ方式の回線接続に用いられる伝送制御手順で、コンピュータから端末装置へボーリングをかけて、データの送信を

- 共用ファイルのロック機能
- アカウンティング機能
- ロールイン/ロールアウト機能

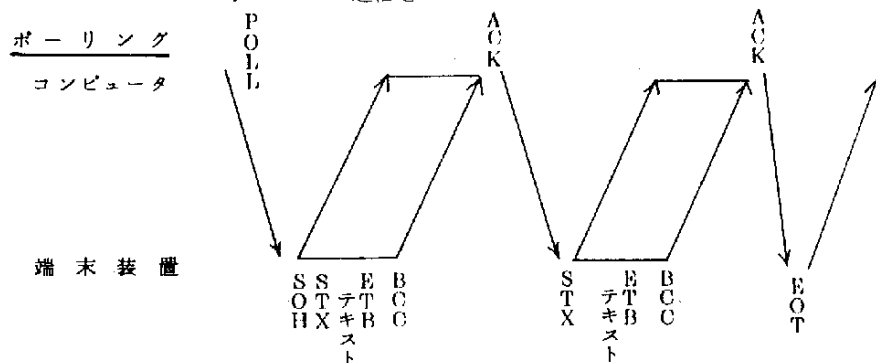
8. 基本データ伝送制御手段

(1) コンテンション方式

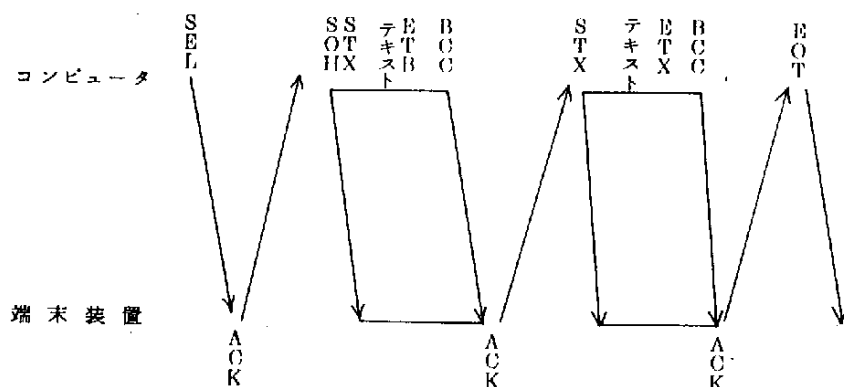
これは、ポイント・ツー・ポイント方式の接続について用いられる伝送制御手順である。この方式では、コンピュータまたは端末装置で送信データが発生した場合、随時に、送信要求を出すことができる。半二重通信回線の場合は、送信権を得ようとして、コンピュータと端末装置の競合(コンテンション)が発生する。

全二重通信回線の場合、両方向の送信を同時に行うことができる。

勧誘する。コンピュータから端末装置へデータを送り出すときは、端末装置の受信状態を問いあわせる(アドレッシング又はセレクトイング)。



セレクトィング
(アドレッシング)



(3) 伝送制御符号

伝送制御を行うには、制御用コードが必要である。

一般に次のような10種類のコードが用意されている。

符 号	名 称
SOH	Start of Heading (ヘディング開始)
STX	Start of Text (テキスト開始)
ETX	End of Text (テキスト終結)
EOT	End of Transmission (伝送終了)
ENQ	Enquiry (問合せ)
ACK	Acknowledgement (肯定応答)
NAK	Negative Acknowledgement (否定応答)
ETB	End of Transmission Block (伝送ブロック終結)
SYN	Synchronous Idle (同期信号)
DLE	Data Link Escape (伝送制御拡張)

(4) メッセージ形式

メッセージの形式には、次のような 5 種類がある。

S		E	B
T	テ キ ス ト	T	C
X		X	C

B C C は、ブロック検査文字

S		E	B
T	テ キ ス ト	T	C
X		B	C

S		S		E	B
O	ヘ デ ィ ン グ	T	テ キ ス ト	T	C
H		X		X	C

S		S		E	B
O	ヘ デ ィ ン グ	T	テ キ ス ト	T	C
H		X		B	C

S		E	B
O	ヘ デ ィ ン グ	T	C
H		B	C

参 考 文 献

- (1) J. Mantine 「Telecommunications and the Computer」 Prentice-Hall, 1976
- (2) J. マーチン, 北原安定訳「リアルタイム」日本経営出版会, 1974
- (3) 松平誠「データ通信入門」日本経営出版会, 1970
- (4) 斉藤正治「TSSの計画と運営」TBS出版会, 1977
- (5) 岩城三郎「オンラインプログラム設計入門」日刊工業新聞社, 1973
- (6) 西谷直二他「リアルタイム・システムズ・マネジメント」企画センター, 1972
- (7) 「Systems Analysis and Design of Real-time Management Information Systems」Thierauf, Prentice-Hall, 1975
- (8) D. E. W. Bucher 「Maintenance of the Computer Sciences Teleprocessing System」I CRS, 1975
- (9) マーチン, 北原安定訳「リアルタイム・システムの設計管理運用」日本経営出版会, 1975
- (10) 「分散システムに関する調査」電子協, 1977
- (11) 大野豊「オンライン・システムのソフトウェア」産業図書, 1977
- (12) 植田義明「データ通信のための通信回線利用マニュアル」企画センタ, 1973
- (13) 「データ通信便覧」データ通信協会, 毎年刊行
- (14) 「Handbook of Data Communications」NCC Publications, 1975
- (15) 「Basics of Data Communications」Karp, McGraw-Hall, 1976
- (16) 斉藤正治「TSSの計画と運営」TBS出版会, 1977
- (17) J. Martin 「Design of Real-time Systems」Prentice-Hall, 1976
- (18) 「データ通信のための公衆通信回線利用制度解説」東京出版センター
- (19) 広田憲一郎, 川端久喜, 砂川博「データ伝送システム」産業図書
- (20) JIS ハンドブック情報処理 1981」日本規格協会

参 考

「ACMカリキュラム'78」(要約)

("Curriculum '78", Recommendations for the Undergraduate Program in Computer Science, Comm. ACM, Vol. 22, pp. 147 ~ 66)

1. はじめに

米国計算機学会(ACM)の計算機科学カリキュラム委員会(C³S)では、継続的にカリキュラム開発作業を続けて来た。「カリキュラム'68」の発表以来、この分野での多くの顕著な活動が行なわれ、また多方面からの意見や協力も加わった。これらの結果が、本報告に集約されている。

本報告書は、最初に計算機科学の中核カリキュラムを、次に選択コースを示している。このあと、学部用プログラムの全容とサービス・コースの内容、さらに教育用設備やスタッフなどについても触れている。

2. 中核カリキュラム

2.1 はじめに

中核カリキュラムは、計算機科学の学部レベル・プログラムに共通な領域をまとめたもので、次の二つの水準から成っている。

① 基礎水準:

基本的なトピックスと五つのコース例を含む。

② 中位水準:

三つのコース例を含む。

これら8コースは、次のようなものである。

CS 1 計算機プログラミングI

CS 2 計算機プログラミングII

CS 3 計算機システム入門

CS 4 計算機構造入門

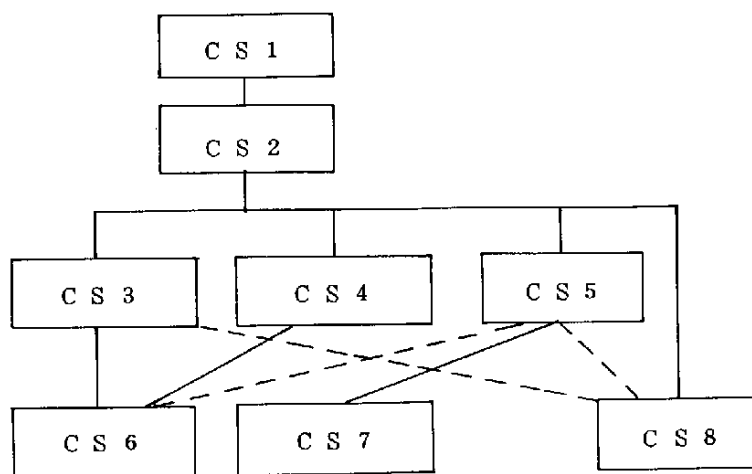
CS 5 ファイル処理入門

CS 6 オペレーティング・システムと計算機アーキテクチャI

CS 7 データ構造とアルゴリズム解析

CS 8 プログラム言語の構成

これらのコース間の事前履修条件に関する構造を示すと、第 1 図のようになる。



第 1 図 計算機科学の中核カリキュラム

2.2 目 標

中核カリキュラムによる教育を通じて、計算機科学専攻の学生は、少なくとも次のような目標を達成する基礎を身につけるべきである。

1. 適度な時間内に、正しく動作し、良く文書化され、読解可能なプログラムが書けること。
2. これらのプログラムが、適度な効率を持ち、適切に構成されているかどうか判断できること。
3. 一般に、どのようなタイプの問題が計算機処理に適しているかを知っていること。また、その様な問題の解決に必要な、種々の手段を知っていること。
4. 個人又はチームの一員として行なり、作業の意味を評価できること。
5. 基本的な計算機アーキテクチャを理解していること。
6. 一つ以上の適用領域での徹底した訓練や、計算機科学での更に進んだ教育を受ける能力を持っていること。

2.3 基本的題材

次のようなトピックスが基本的と考えられ、計算機科学の全ての学部用プログラムに、共通

に含まれるべきである。

プログラミングに関するトピックス

- P 1. アルゴリズム
- P 2. プログラム言語
- P 3. プログラミング・スタイル
- P 4. デバッグと検証
- P 5. アプリケーション

ソフトウェアの構成

- S 1. コンピュータ
- S 2. データの表現
- S 3. 記号コーディングとアセンブリ・システム
- S 4. アドレッシング
- S 5. マクロ
- S 6. プログラムの区分化と関係
- S 7. リンクとローダ
- S 8. システムズ・プログラムとユーティリティ・プログラム

ハードウェア構成

- H 1. 計算機システムの構成
- H 2. 論理設計
- H 3. データの表現と転送
- H 4. デジタル演算
- H 5. デジタル記憶とアクセス
- H 6. 制御と入出力
- H 7. 信頼性

データ構造とファイル処理

- D 1. データ構造
- D 2. 分類と探索

- D 3. 木
- D 4. ファイル用語
- D 5. 順次アクセス
- D 6. 直接アクセス
- D 7. ファイル入出力

2.4 実施上の考慮

基本水準の題材を教育する際には、プログラミング・プロジェクトも並行的に割当てる必要がある。また、これらの題材の教育では、そのすべてにおいて、良いプログラミング・スタイルを強調すべきである。このための訓練によって、効果的なアルゴリズムの開発が促進され、正確で分かりやすいプログラムの書ける学生が育てられる。従って、プログラミング演習では、効果的アルゴリズム、構造化プログラミングの諸技法、良い文書化などに重点が置かれるべきである。構造的プログラミングやプログラミング・スタイルについては、基本水準では特に専門のコースを置かず、こうした哲学をコース全体に浸透させるべきである。

また、基本水準の教育中、意味のある実際の適用例を引用し、検討すべきである。この場合、これらの社会的、哲学的、倫理的考察にも触れる必要があり、これらもまたコース全体に浸透させるべき問題である。

この水準での多くのコースでは実験を導入し、計算機による実際的な経験を与えるべきである。これは、指導者の緊密な監督下で、良いプログラミング技法を身につけさせるよう考えねばならない。

2.5 基本水準コースの例

以下に示す一群のコースは、基本水準での中核を実現する一例で、各コース名の後に単位数の例が示してある。例えば、(2-2-3)というのは、週2時間の講義と2時間の演習で、計3単位が与えられることを示す。

CS 1 計算機プログラミング I (2-2-3)

目標：(a) 問題解決方法とアルゴリズム設計の入門

(b) 広く利用されている高水準言語の教育

(c) 良いプログラミング・スタイルへの手法による、プログラムの設計、作成、デバッグ、文書化方法の教育

トピックス

- A. 計算機の構成 (5 %)
- B. プログラム言語とプログラミング (45 %)
- C. アルゴリズム設計 (45 %)
- D. 試験 (5 %)

C S 2 計算機プログラミングⅡ (2-2-3)

事前履修条件 : C S 1

- 目標 ; (a) プログラムの設計, スタイルと表現, デバッグとテストについての訓練を, 特にやや大きなプログラムについて継続。
- (b) アルゴリズム分析の入門
- (c) 文字列処理, 再帰的処理, 内部分類と探索, 簡単なデータ構造についての基本的性質の導入

トピックス

- A. プログラミング概観 (15 %)
- B. 構造化プログラミングの概念 (40 %)
- C. デバッグとテスト (10 %)
- D. 文字列処理 (5 %)
- E. 内部探索と分類 (10 %)
- F. データ構造 (10 %)
- G. 再帰的処理 (5 %)
- H. 試験 (5 %)

C S 3 計算機システム入門 (2-2-3)

事前履修条件 : C S 2

- 目標 ; (a) 計算機システムの基本概念の把握
- (b) 計算機アーキテクチャの導入
- (c) アセンブラ言語の教育

トピックス

- A. 計算機の構造と機械語 (15 %)
- B. アセンブラ言語 (30 %)

- C. アドレッシング技法 (5 %)
- D. マクロ (10 %)
- E. ファイル入出力 (5 %)
- F. プログラムの区分化と連係 (20 %)
- G. アセンブラの構成 (5 %)
- H. 解釈ルーチン (5 %)
- I. 試験 (5 %)

CS4 計算機構造入門 (3-0-3) 又は (2-2-3)

事前履修条件 : CS2

目標 : (a) 主要ハードウェア要素の構成と構造化についての導入

(b) デジタル計算機システムでの情報の転送と制御の機構についての理解

(c) 論理設計の基礎習得

トピックス

- A. 基本論理設計 (25 %)
- B. 符号化 (5 %)
- C. 数表現と算術演算 (10 %)
- D. 計算機アーキテクチャ (35 %)
- E. 事例 (20 %)
- F. 試験 (5 %)

CS5 ファイル処理入門 (3-0-3)

事前履修条件 : CS2

目標 : (a) 大容量記憶装置での、データ構造化の概念と手法の導入

(b) 大容量記憶装置利用の経験

(c) データ構造とファイル処理技法を応用するための基礎習得

トピックス

- A. ファイル処理の環境 (5 %)
- B. 順次アクセス (30 %)
- C. データ構造 (20 %)
- D. 直接アクセス (35 %)

E. ファイル入出力 (5 %)

F. 試験 (5 %)

2.6 中位水準コースの例

CS 6 オペレーティング・システムと計算機アーキテクチャ I (2-2-3)

事前履修条件 : CS 3 及び CS 4

(CS 5 も前提として望ましい)

- 目標 : (a) レジスタ間転送およびプログラミングのレベルにおけるシステム記述での, 計算機システムの構成とアーキテクチャの理解
- (b) オペレーティング・システムの諸原理の主要概念の導入
- (c) オペレーティング・システムと計算機システムのアーキテクチャとの相互関係の学習

トピックス

- A. 概観 (10 %)
- B. 動的手続きの活性化 (15 %)
- C. システムの構造 (10 %)
- D. 評価 (15 %)
- E. 記憶域管理 (20 %)
- F. プロセスの管理 (20 %)
- G. 修復手続き (5 %)
- H. 試験 (5 %)

CS 7 データ構造とアルゴリズム解析 (3-0-3)

事前履修条件 : CS 5

- 目標 : (a) 種々のデータ構造に働く非数値的アルゴリズムに対する, 解析と設計の手法の適用
- (b) データベース管理システム的环境下における, データ操作方法の選定に関する, アルゴリズム解析と設計基準の適用

トピックス

- A. 復習 (10 %)
- B. グラフ (15 %)
- C. アルゴリズム設計と解析 (30 %)

- D. 記憶域管理 (15%)
- E. システム設計 (25%)
- F. 試験 (5%)

CS 8 プログラム言語の構成 (3-0-3)

事前履修条件: CS 2 (CS 3 及び CS 5 も前提として特に望ましい)

目標: (a) プログラム言語の構成, 特にプログラムの実行時の振舞い, についての理解

(b) プログラム言語の詳述と解析についての形式的研究の導入

(c) 基本水準で導入された, 問題解決法やプログラミング技術の発展

トピックス

- A. 言語の定義構造 (15%)
- B. データ型と構造 (10%)
- C. 制御構造とデータの流れ (15%)
- D. 実行時の問題 (25%)
- E. 解釈型の言語 (20%)
- F. 字句解析とパーシング (10%)
- G. 試験 (5%)

3. 計算機科学選択コース

3.1 はじめに

計算機科学選択コースの中には, 基本水準および高位水準において, それぞれ適切と考えられるものがある。基本水準の選択コースは, 学生の教育プログラムを強化するものであり, 高位水準の選択コースは, 主要な要請を満たすためと, 計算機科学の特定領域を更に詳しく研究させるために選ばれるべきである。

3.2 基本水準

基本水準では, APL, Cobol, LISP, PL/I 等の特定のプログラム言語に基づくコースを提供できる機構を具え, 計算機科学専攻を始め他の専攻学生が, これらのコースを選択できるようになっていることが非常に望ましい。

3.3 高位水準

高位水準の選択コースを10コース提示する。計算機科学科は、可能な限りこれらのコースを定常的に開設しておくべきである。

CS 9 計算機と社会(3-0-3)

事前履修条件:基本中核コース

目標:(a) 社会的な価値と評価の概念の提示

(b) 計算機の社会に及ぼす影響を記述するモデルの導入

(c) 社会的影響についての考察や決定を行なう、専門的活動に対する基盤の提供

(d) 計算機の社会的影響についての諸問題に、適用可能な手段や技法の提示

トピックスの例

A. 計算方法と技術の歴史

B. 現代社会における計算機の位置

C. 計算機と個人

CS 10 オペレーティング・システムと計算機アーキテクチャⅡ(2-2-3)

事前履修条件:CS 6

トピックス

A. 復習(10%)

B. 並行処理(15%)

C. 名前の管理(15%)

D. 資源割付け(25%)

E. 保護(15%)

F. 高度のアーキテクチャとオペレーティング・システムの実現(15%)

G. 試験(5%)

CS 11 データベース管理システムの設計(3-0-3)

事前履修条件:CS 6及びCS 7

トピックス

A. データベース概念の導入(5%)

B. データ・モデル(15%)

C. データの正規化(5%)

- D. データ記述言語 (10 %)
- E. 問合わせ機能 (15 %)
- F. ファイル編成 (25 %)
- G. 索引編成 (5 %)
- H. ファイルの安全性 (10 %)
- I. データの統合性と信頼性 (5 %)
- J. 試験 (5 %)

CS12 人工知能 (3-0-3)

事前履修条件 : CS 7

トピックス

- A. 表現 (40 %)
- B. 探索戦略 (15 %)
- C. 制御 (20 %)
- D. 交信と知覚 (10 %)
- E. 応用 (10 %)
- F. 試験 (5 %)

CS13 アルゴリズム (3-0-3)

事前履修条件 : CS 7 及び CS 8

トピックス

- A. 組合わせ数学 (10-25 %)
- B. 数値計算 (10-25 %)
- C. システムズ・プログラミング (10-25 %)
- D. 人工知能 (10-25 %)
- E. 領域独立手法 (15 %)
- F. 試験 (5 %)

CS14 ソフトウェアの設計と開発 (3-0-3) 又は (2-2-3)

事前履修条件 : CS 7 及び CS 8

トピックス

- A. 設計技法 (50 %)
- B. 組織と管理 (15 %)
- C. チーム・プロジェクト (30 %)
- D. 試験 (5 %)

CS15 プログラム言語理論 (3-0-3)

事前履修条件: CS 8

トピックス

- A. 復習 (15 %)
- B. スキャナ (20 %)
- C. パーザ (40 %)
- D. 翻訳 (20 %)
- E. 試験 (5 %)

CS16 オートマトン, 計算可能性, 及び形式言語 (3-0-3)

事前履修条件: CS 8 及び MA 4 (第 4.1 節参照)

トピックス

- A. 有限状態の概念 (30 %)
- B. 形式文法 (35 %)
- C. 計算可能性とチューリング機械 (30 %)
- D. 試験 (5 %)

CS17 数値数学: 解析 (3-0-3)

事前履修条件: CS 1 及び MA 5

トピックス

- A. 浮動小数点演算 (15 %)
- B. 数学的サブルーチン・パッケージの利用法 (5 %)
- C. 補間法 (15 %)
- D. 近似 (10 %)
- E. 数値積分と微分 (15 %)

- F. 非線形方程式の解法 (15%)
- G. 常微分方程式の解法 (20%)
- H. 試験 (5%)

CS18 数値数学：線形代数 (3-0-3)

事前履修条件：CS1及びMA5

トピックス

- A. 浮動小数点演算 (15%)
- B. 数学的サブルーチン・パッケージの利用法 (5%)
- C. 線形方程式系に対する直接的解法 (20%)
- D. 誤差解析とノルム (15%)
- E. 反復法 (15%)
- F. 固有値と固有ベクトルの計算 (15%)
- G. 関連トピックス (10%)
- H. 試験 (5%)

3.4 特殊なトピックス

教育施設的环境が許すなら、次のような特殊なトピックスについてのコースも、提供されるべきである。

- A. マイクロコンピュータ実験
- B. ミニコンピュータ実験
- C. 性能評価
- D. テレコミュニケーション / ネットワーク / 分散システム
- E. システム・シミュレーション

(以下略)

4. 学部用プログラム

4.1 はじめに

計算機科学の18コースは次の通りである。

- CS1 計算機プログラミングⅠ
- CS2 計算機プログラミングⅡ

- CS 3 計算機システム入門
- CS 4 計算機構造入門
- CS 5 ファイル処理入門
- CS 6 オペレーティング・システムと計算機アーキテクチャ I
- CS 7 データ構造とアルゴリズム解析
- CS 8 プログラム言語の構成
- CS 9 計算機と社会
- CS 10 オペレーティング・システムと計算機アーキテクチャ II
- CS 11 データベース管理システムの設計
- CS 12 人工知能
- CS 13 アルゴリズム
- CS 14 ソフトウェアの設計と開発
- CS 15 プログラム言語理論
- CS 16 オートマトン, 計算可能性, 及び形式言語
- CS 17 計算数学: 解析
- CS 18 計算数学: 線形代数

これらのコース群の構造を図 2 に示す。なお, この図には米国数学会 (MAA) の学部数学プログラム委員会 (CUPM) による, 次の数学コース群との関連も示している。

- MA 1 初等微積分学
- MA 2 数学解析 I
- MA 2A 確率
- MA 3 線形代数
- MA 4 離散構造
- MA 5 数学解析 II
- MA 6 確率と統計

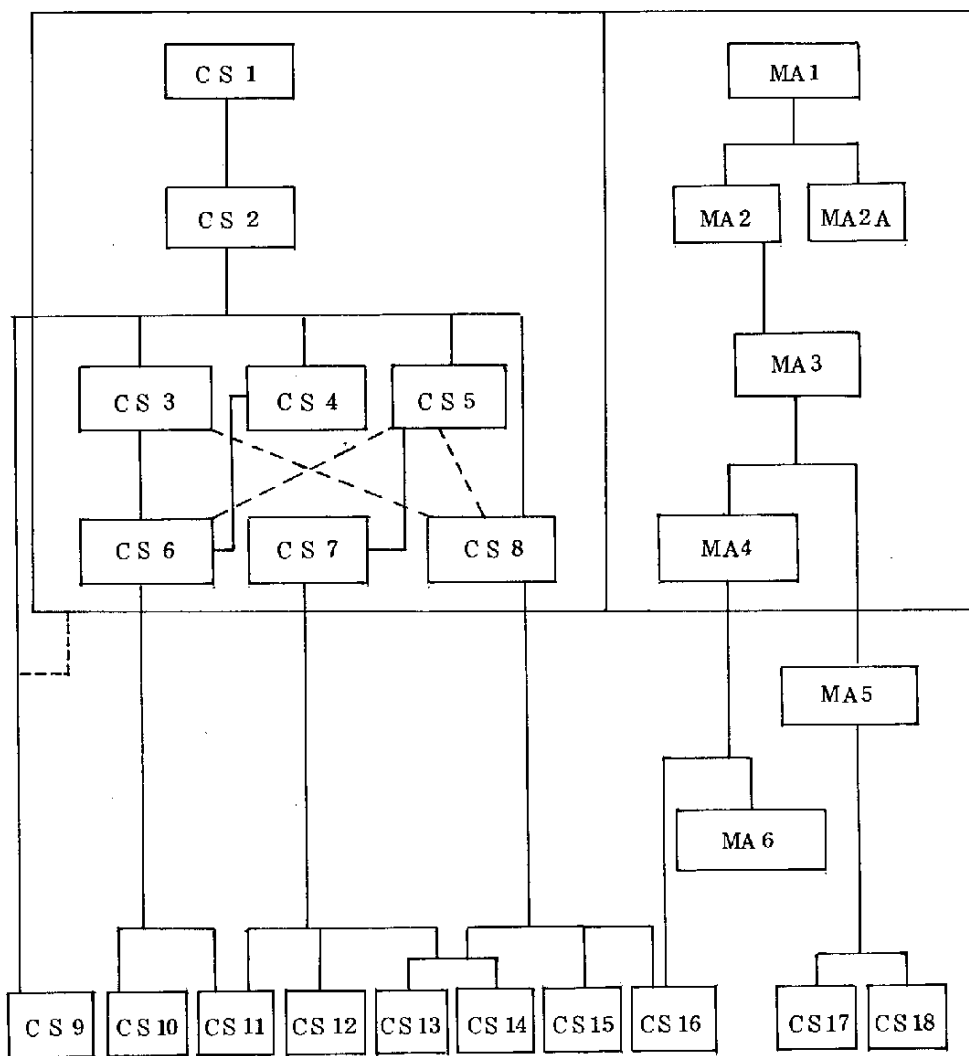


図2 推奨される計算機科学と数学のコース

(4.2 節以下略)

索

引

[ア]

RPG	137
RPGII	137
RAS	145
RASIS	150
アーキテクチャ	58
INGRES	136
IMS	133, 138
IDS	134, 138
アイドル・タイム	149
アカウンティング	29
アクセス経路	134, 135
アクセスコスト	30
アクセス時間	30
アクセス法	80, 146, 155
アクセス法式	30
ADABAS	139
圧縮	76
アドレッシング	59
あふれ	84, 142
アベイラビリティ	38
網	133
誤り検出	30
アルゴリズム	39

[イ]

意志決定	8
位相変調	166
一階の述語論理系	135
一括処理	38
1の補数	59
イメージ	142
入れ子	59

インタフェース	29, 58
インタブリタ	29, 59, 106

[ウ]

受入れテスト	107
--------	-----

[エ]

AVL-木	77, 88
エコー・チェック方式	167
エクステント	154
エディタ	29
SRR	160
MTTR	38
MTBF	38

[オ]

オーバーレイ構造	106, 162
オープン	156
応答時間	38
置換えアルゴリズム	162
オフライン装置	38
オペレーティング・システム	58, 76, 135, 149, 150
親言語方式	129, 138
親子関係	132
親子集合	134
親子集合型	133
親子集合の順序	139
親子集合の選択	139
親子属性	139
オンライン・システム	38

[カ]

回線使用時間	167
回線使用率	167

回線網の設計	31	記 憶	27
階層記憶方式	30	記憶域の割振り	30
階層構造	3, 86, 132	記憶管理モジュール	29
階層モデル	124, 132	記憶スキーマ	138
解釈ルーチン	59	記憶素子	58
概念スキーマ	128	記憶媒体	75
開発の動機	15	記憶保護	58
外部からの信号	157	記憶容量	30
外部スキーマ	128	木構造	39, 86, 88, 132
外部ファイル	155	擬似言語	107
外部分類	39	基数増し法	59
会話型処理	166	木探索	99
学 習	27	基本アクセス	81
下降型プログラミング	101	基本アクセス法	155
仮想記憶	150	基本演算	134
仮想記憶システム	29, 147, 148, 161, 162	機密保護	60, 127
カタログ・プロシージャ	152	機密保持	29
かち合い	83	逆見出しファイル	30
可用性	150	強 度	107
環 境	14	局所性	162
環境条件	2	均衡化	77, 88
関 係	134		
関係代数	135	〔ク〕	
関係代数系	124	空間分割	158, 160
関係モデル	124, 132, 142	鎖	88
関係論理系	124, 135	具体値	132
監視プログラム	106	区分編成	156
感 情	27	区分編成ファイル	76
関数従属性	136	クラス内の優先順位	153
間接アドレス	59	繰 返	39
間接効果	20	クローズ	156
管理サイクル	8	群計数チェック方式	167
管理プログラム	44		
		〔ケ〕	
〔キ〕		ゲート	59
木	98	経営活動	3
キー	82	経営活動の階層性	4
キー比較による探索法	99	経営活動の過程	4

計算	147, 157	[サ]	
計算機上のファイル	126	サイクリック・リダンダンシー	167
計算機ネットワーク	150	・チェック方式	
系 列	39	再帰ルーチン	59
結合演算	135	再使用可能	106
結合度	107	再入可能	106
現在指示子	139	再入可能ルーチン	59
原子プログラム・リスト	107	再配置	59, 155
		再配置ローダ	59
[コ]		再編成	76, 85, 142
子	133	財務システム	9
COBOL	138	作業レベル	4
CODASYL	134	索引順アクセス	30
データベース作業グループ		索引順編成	156
CODASYL方式	125	索引つき順編成	142
交換接続方式	166	索引つき順編成ファイル	76, 82
交換による整列化	99	索引による方法	125
降 順	39, 82	雑 音	30
高信頼性	58	サブスキーマ	138
構造化流れ図	107	サポート・ツール	58
構造化プログラミング	107		
構造的プログラミング	92, 107	[シ]	
後退復帰	131	JIS規格	38
行 動	28	ジェネレータ	29
行動様式	28	支援プログラム	44
交流伝送	166	磁気ディスク	80
互換性	38	磁気テープ	80
個 性	27	磁気ドラム	80
個別テスト	38	識 別	29
コマンド	152	資 源	158, 160
コミュニケーション	5, 14, 27	資源の割り当て	160
ルーチン	59, 106	資源の管理	147
コンテンツ	31	資源割り振り	60
コンテンツ方式	167	試験仕様書	107
コンパイラ	29	思 考	27
コンパクション	162	システム	2
コンピュータ・システム	58	システム・テスト	107
		システム仕様書	21

システムの構造	2	順次的	147
システムの費用	20	順編成	156
システムの目的	3	順編成ファイル	76, 82
システムのライフサイクル	16	障害回復	124, 127, 131
システム構成	38	条件アセンブリ	59
システム資源	58, 145, 150	昇 順	39, 82
システム資源の使用率	38	上昇型プログラミング	101
システム制御機能	58	衝突	141
システム性能	38	情報検索	142
システム分析	16	情報構造	39
SYSTEM2000	139	情報システム	2
SYSTEM R	136	情報システムの開発	15
実 現	123	情報システムの導入	28
実現値	132	情報システムの役割	15
実行サイクル	58	情報代数	137
実際のシステムでの実現法	125	情報のタイプ	5
実体システム	2	情報要求	5, 14
指 標	59	ジョブ	146
時分割	158, 160	ジョブ管理	149, 150
シミュレーション	167	ジョブ・クラス	153
シミュレータ	59	ジョブ・スタック	153
事務処理上のファイル	126	ジョブ・ステップ	150
事務分析	17	ジョブ制御	29
社会的規範	28	ジョブ制御言語	146, 151, 152
社会的なシステム	10	ジョブ制御ブロック	160
集 団	28	ジョブ制御文	152
集中型	31	ジョブ連続処理	149
周波数変調	166	処理プログラム	44, 150
術 語	150	人 格	27
順アクセス	30, 76, 81	診断モジュール	29, 60
順アクセス法	155	振幅変調	166
巡 回	88	信頼性	150
循環つなぎリスト	142	信頼度	38
循環連結リスト	77	心理テスト	27
純手続	106	真理値表	59
順次アクセス記憶装置	80		
順序木	133	〔ス〕	
		スーパーバイザ	158

スーパーバイザ呼出し	157	[ソ]	
スキーマ	124, 128, 138	相互排除問題	159
スケジューラ	29	相対アドレス	59
スケジューリング	60, 146, 151	装置独立性	155
スケジューリング・アルゴリズム	147, 158	挿入による整列化	99
スタック	95	属性	135
ステート・ワード	157	組織構造	28
ストア・アンド・フォワード	31		
ストリング	39	[タ]	
スナップ・ショット	107	ターン・アラウンド・タイム	145
スプーリング	149, 153	待機アクセス	81
スループット	38, 145, 158	待機アクセス法	155
		対人関係	28
[セ]		代替案	16
生産システム	8	タイム・シェアリング・システム	149, 166
生産者-消費者問題	159	対話モード	29, 60
制御構造	91	多重タスク	106
制御プログラム	145, 150	多重つなぎ	77, 88
整合性	129	多重つなぎ編成	142
静的	153, 160	多重プログラミング	38, 147, 149, 157
正当性	124, 129	多重プロセッシング	38
整列化	92, 99	多重プロセッサ	150
セグメント	162	タスク	147, 151, 157
セグメント・テーブル	162	段階的な細分化	39
セグメント方式	162	段階的詳細化	92, 101
設計仕様書	107	単向通信方式	166
絶対アドレス	59	探索	92
設置テスト	107	単純構造	106
線形構造	86	単体テスト	107
線形リスト	92, 95	ダンプ	107
前進復帰	131	端末装置	38, 166
全体テスト	38	端末の選定	31
選択	39		
選択による整列化	99	[チ]	
センタ装置	38	チェック・ポイント	131
全二重通信方式	166	知覚	27
戦術レベル	4	逐次探索法	39, 99
戦略レベル	4	知能	27

知能テスト	27	データ収集型処理	166
チャネル	30	データ信号速度	166
チャネル制御モジュール	29	データ制御	124
抽象化	132	データセット管理	29
調歩同期	30	データ操作	124
調歩同期方式	166	データ操作機能	129
直後像	131	データ操作言語	129
直積	135	データ通信	166
直接アクセス	30	データ・ディクショナリ	30
直接アクセス記憶装置	80	データ伝送速度	166
直接アクセス法	155	データ伝送路	166
直接効果	20	データ独立性	124, 128, 135
直接編成	76, 141, 156	データのアクセス制御	129
直接編成ファイル	83	データの一貫性の制御	129
直前像	131	データの制御機能	129
直通接続方式	166	データの正規化	124, 135, 136
直流伝送	166	データベース	77, 123, 126, 127, 135
		データベース概念	30
[ツ]		データベース管理システム	123, 124, 137, 138
追跡控え	131	データベース管理者	128
2パス・アセンブラ	59	データベース・キー	139
通信回線	38, 166	データベース・システム	123, 124, 126
通信制御	29	データ変換装置	38
通信制御装置	58, 166	データ・モデル	123, 124, 132
突き合せ	76, 82	テープ	30
つなぎ	77, 142	ダブル駆動型プログラム	106
つなぎによる編成	125	DDL	138, 139
		DSDL	138
[テ]		DBTG '71 提案	134
データ管理	30, 149, 151, 154	DBTG の 1 の提案	138
データ管理者	30	DBMS の実例	30
データ管理の概念	146	定義域	135
データ記述	124	ディスク	30
データ記述機能	129	ディスパッチング	147, 158
データ記述言語	129	ディバッキング	29
データ構造	76, 77, 80, 91, 146	ディレクトリ	30, 84
データ構造図	134	テスト・アンド・セット	159
データ作成装置	38	哲学者の食事の問題	159

手続きの	135	内部分類	39
デッドロック	140, 159, 160		
デバッグ・エイド	107	[ニ]	
デブロック化	59	2進木探索法	39
伝送制御手順	167	2線式回線	166
伝送制御とプロトコル	31	日程計画	38
伝送制御符号	167	2度送り方式	167
転置ファイル	77, 88, 142	2の補数	59
		2分木	77, 88, 98
[ト]		2分探索法	39, 99
問合せ言語	30	入出力管理システム	59
問合せ/応答型処理	166	入出力指示ステートメント	59
問合せ機能	129	入出力制御	59
同期	30, 147, 159	入出力制御システム	149
同期方式	166	入出力制御装置	58
動機	28	入出力チャネル	29, 58
動機づけ	27, 28	人間関係	28
同義語	76, 83, 84, 141	人間行動	15
同時更新	127, 131	人間の作業特性	27
同時実行制御	124, 129	認知	27
同時性	160	[ネ]	
到着順方式	153	ネットワーク	133
動的	153, 160	ネットワーク構造	86, 142
動的アドレス変換機構	162	ネットワーク・システム	29, 58
動的再配置	106	ネットワークの形態	31
TOTAL	139	ネットワーク・モデル	123, 124
独立	127		
独立言語方式	129	[ハ]	
独立同期方式	166	ハードウェア・モジュール	29
トップダウン・プログラミング	39	媒体変換	29
トランスレータ	29, 60	排他制御	161
取り出しサイクル	58	HIPO	107
トレース	59, 107	配列	39, 92, 94
		バケット	30
[ナ]		パス	39
内部スキーマ	128	PASCAL	38
内部整列化	99	バックマン線図	104, 134
内部ファイル	155	ハッシュ関数	83, 141

ハッシュ関数の利用	125	ブール関数	59
ハッシング	83	FORTRAN	138
ハッシング法	76	フォーマット変換	29
パッチ処理システム	38	復元	29
パッチ・モード	60	復号器	59
バッファ	58	符号器	59
バッファリング	59	付帯設備	38
パラメタ	59	付帯装置	38
バリティ・チェック方式	167	物理的	134
半二重通信方式	166	物理的構造	77
販売システム	9	物理レコード	154
汎用シミュレーション	167	フラグメンテーション	162
〔ヒ〕		フリップフロップ	59
B-木	77, 88, 142	プログラム言語	92
B*-木	77, 88	プログラム再配置	106
PPBS	19	プログラム状況語	157
非手続き的	135	プログラムのスタイル	102
非同期的	157	プログラム連係	59
非ブロック化	156	プロジェクトチーム	28
表	39, 94	プロセス	147, 151, 157
評価	19	プロセス管理	149, 151, 153
評価基準	17	プロセス制御ブロック	158, 160
標準	127	プロセッサ	58, 147
標準化	38	ブロック	154
〔フ〕		ブロック化	59, 79, 156
ファイル	30, 75, 79, 135, 146 147, 154, 155	ブロテクション	29
ファイル・カタログ	81, 155	分岐接続方式	166
ファイル構成法	146	分散型	31
ファイル・システム	126	分散処理	150
ファイル処理	75, 76	分析の手法	17
ファイル制御ブロック	156	分配による整列化	99
ファイルの再編成	155	分類	76, 82
ファイル編成	29	〔ヘ〕	
ファイル編成法	76, 80, 86, 135	ページ	154
フィールド	154	ページ置換え	162
		ページ・テーブル	162
		ページング方式	162

並行的	147	[ミ]	
並行的プロセス	150	見出し付探索	100
併合	76, 82		
併合による整列化	99	[メ]	
変革推進者と受入側	28	メッセージ交換型処理	166
変調速度	166	メンバ	84
変調方式	166		
変復調装置	166	[モ]	
		目的	16
[ホ]		目的モジュール	106
ボーリング	31	モジュール	29
ボーリング/ アドレッシング方式	167	モジュール化	59, 107
ポインタ	77, 88	モデム・インタフェース	166
ポインタ配列	142	モニタ	29, 159
ポイント	142	問題仕様書	107
ポイント・ツー・ポイント	30		
保護	160	[ユ]	
保守性	150	ユーティリティ	30
POST/WAIT	159	有向木	98, 133
ボリューム	154	有向グラフ	133
		有向輪	133
[マ]		優先順位	157
MARKV	137, 138	優先順位方式	153
マイクロプログラム	58		
マイクロプログラミング	29	[ヨ]	
マクロ展開	59	予測の手法	18
マクロ命令	59	欲求	27, 28
マクロ呼出し	59	予備検討	16
マス・ストレージ	30	呼び出し手順	106
待ち行列	95, 153, 167	読み書き問題	159
マルチアクセス	29	4線式回線	166
マルチ・ドロップ	30		
マルチブлекс	29	[ラ]	
マルチプログラミング	29	ラベル	154
マルチプロセッシング	29	乱アクセス	30, 76, 81
マルチ・プロセッサ	29	ランダム化	83
マン・マシン・システム	27		

〔リ〕

リーダシップ	28
リアル・タイム・システム	149
リアル・タイム処理	166
リエントラントプログラム	29
リスト構造	39
リテラル	59
リモート・バッチ処理	166
利用者マニュアル	107

〔レ〕

レコード	30, 79, 135
レコード型	132, 133
レコード・キー	139
レコードの論理的構造	77
連系編集	29
連系編集プログラム	106
連結リスト	77, 88
連 続	39

〔ロ〕

ローダ	29
ローディング	60
ロード・モジュール	106
論理的構造	77
論理的な構造	86
論理的レコード	154

〔ワ〕

ワーキング・セット	162
割当て	147
割込み	29, 147, 157
割込み監視	60
割込み動作	106
割込みマスク	157
1バス・アセンブラ	59

本書は、日本自転車振興会の機械工業振興資金による
補助金の交付を受けて実施した「昭和 55 年度情報処理
教育に関する調査研究等補助事業」の一環として作成し
たものです。

昭和56年3月発行

財団法人 日本情報処理開発協会
情報処理研修センター

〒105 東京都港区浜松町2丁目4番1号
(世界貿易センタービル 7階)

TEL 03 (435) 6511 (代)

許可なしに転載、複製することを禁じます。

THE UNIVERSITY OF CHICAGO

PHYSICS DEPARTMENT

