# 初級情報処理技術者育成指針

(改 訂 版)

平成2年3月

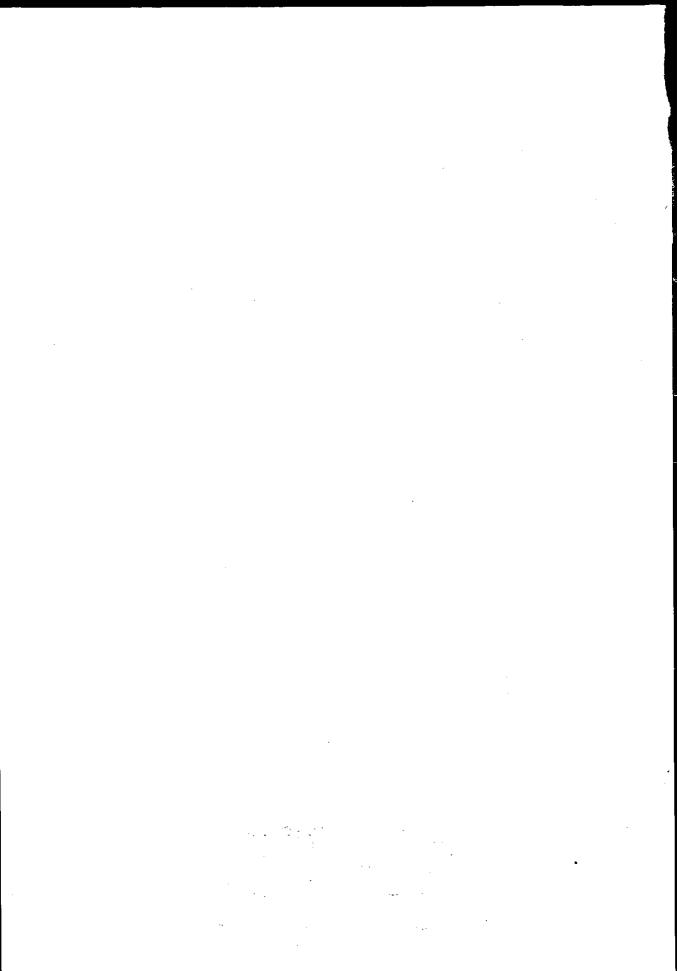


財団法人 日本情報処理開発協会 中央情報教育研究所

この報告書は、日本自転車振興会から競輪収益の一部である機械 工業振興資金の補助を受けて、平成元年度に実施した「高度情報処 理技術者育成指針等の作成」の成果をとりまとめたものです。



C 16 H1



#### 初級情報処理技術者育成指針

# 改訂委員会委員名簿

(敬称略・順不同)

委員長 西村 敏男 神奈川大学

理学部 情報科学科 教授

委 員 江村 潤朗

(財)日本情報処理開発協会 (財)日本情報処理開発協会

中央情報教育研究所 技術顧問

大津 博

学常葉学園 富士短期大学 講師

沖山 豊

学電子学園 日本電子専門学校 情報系 注了情報処理科 科長

小倉 正士

学機浜アカデミー 総合電子専門学校横浜校

副校長

耕

木村

電気通信大学 情報工学科 助教授

佐々木 秀夫

学日本電子工学院 日本工学院専門学校

情報学部長

新川 勇

日本ビジネスオートメーション(株)

人材開発部 課長

野々山 隆幸

横浜市立大学 商学部 教授

山田 統

統(学岩崎学園 情報科学専門学校 教務部長

渡辺 昭

(株)東和コンピュータマネジメント

取締役技術部長

渡辺 文雄

東芝エンジニアリング(株)

総務部 人材開発センター 技術主任

山本 欣子

**娜日本情報処理開発協会 常務理事** 

浅井 宗海

断日本情報処理開発協会 中央情報教育研究所 専任講師



# 

	2.1	·. "
grande and the control of the contro		
		. A
大块的10mm 10mm 10mm 10mm 10mm 10mm 10mm 10mm	**==	· ·
	·	1.4
se Maria (Agenta ) Agent (Agent Agent Agent A Agent (Agent Agent Ag	h qu.	٠.
	:.	
	et it	٧.
Substitute of the first of the substitute of the	•	
		. %
25、大型 1000 A	:	
Barras di Spiratifia di Sala Antonio di Solo Maria della		) .
	* · · · · · · · · · · · · · · · · · · ·	

### 一 初級情報処理技術者育成指針〈改訂版〉執筆担当 —

(敬称略)

総論	西村	敏男
一基本知識編一		
第1部 コンピュータ入門	佐々木	秀夫
第2部 ハードウェア	木村	耕
第3部 ソフトウェア	新川	勇
第4部 通信ネットワーク	山本	欣子
第5部 情報処理システム	山田	統
第6部 情報化とコンピュータ	渡辺	文雄
一実務能力編一		
第7部 プログラム・アルゴリズム	冲山	豊
第8部 ファイルとデータベース	小倉	正士

— 総

論

編一

第9部COBOL大津博第10部FORTRAN野々山隆幸第11部C言語浅井宗海第12部システム開発技法渡辺昭第13部表現技法江村潤朗

## - Angle - Angle - Angle Angl

, est est.	÷ ·
	A Committee of the second
	10 10 18 18 18 18 18 18 18 18 18 18 18 18 18
$\mathcal{O}_{\mathbf{k}} = \{ \frac{t}{c_{\mathbf{k}_{\mathbf{k}}}} \}$	
	we the second of
100 To 10	And the second second
44.14	garan ang ang ang at pagis
****	
, v	Sec. 1. 6 (5) (5)
e e Age	
\$ X \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	

# 目 次

総	論	編	
彩	<u>k</u>	hand the second of the second	
	第1章	指針の改訂について	1
	第2章	本指針の目的	···· 7
	第3章	カリキュラムの編成	17
基本	比知識編		
ᢖ	育1部	コンピュータ入門	1- 1
	第1章	コンピュータの特徴	1- 3
	第2章	コンピュータによる情報処理	1-13
	第3章	, 1	1-18
· 5		·	2- 1
	第1章	コンピュータの基本構造	2- 3
	第2章	データ表現	2- 6
	第3章	記憶装置	2-10
	第4章	制御・演算装置	2 - 14
	第5章	入出力装置	2-19
	第6章		
<b>5</b>	第3部	ソフトウェア	3- 1
	第1章		
	第2章		
	第3章		3-18
	第4章	· · · · · · · · · · · · · · · · · · ·	3-22
	, 第5章		3-26
Ì	第4部	学信令 1.00 点	4-1
	第1章		4- 3
	第2章		4- 9
	第3章		

第4章	ネットワーク・サービス	4 - 42
第5章	ネットワーク・セキュリティ	4 - 51
第5部 🏌	青報処理システム	5 <b>-</b> 1
第1章	処理方式	5-3
第2章	システムの構成	5 - 11
第3章	マルチメディア処理	5 - 20
第4章	A I システム	5 - 28
第5章	CAI/CMIシステム	5 - 38
第6部 情	青報化とコンピュータ	6-1
第1章	情報化の現状と将来	6- 3
第2章	情報化社会の課題・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	6 - 22
実務能力編		, ·.
第7部	プログラム・アルゴリズム	7- 1
第1章	アルゴリズムの表現技法	7-3
第2章	3 つの基本型とその応用	7-9
第3章	構造化プログラム論理の作成技法	7 - 13
第4章	事務処理のためのアルゴリズム	7-17
第5章	科学技術計算のためのアルゴリズム	7 - 25
第6章	共通アルゴリズム	7 - 27
第7章	アルゴリズムの検証	7 - 31
, 第,8 部	ファイルとデータベース	8-1
第1章	ファイルの基本概念	8- 4
第2章	ファイルと記憶媒体	8- 9
第3章	順編成ファイル	8 - 22
第4章	直編成ファイル	8 - 25
第5章	索引順編成ファイル	8-28
第6章	区分編成ファイル	8 - 31
	V S A M ファイル	8-33
	パーソナル・コンピュータ用ファイルの操作	8 - 37

第9章	データベースの概念
第10章	データベース管理システム (DBMS)8-54
第9部 (	COBOL 9-1
第1章	COBOL概説 ····································
第2章	順ファイル・中核(I)
第3章	順ファイル・中核 (Ⅱ)
第4章	順ファイル・中核(Ⅲ) 9-21
第5章	順ファイル・中核(Ⅳ)
第6章	相対ファイル 9-27
第7章	索引ファイル
第8章	整列併合
第9章	プログラム間連絡その他
第10部 ]	FORTRAN 10- 1
第1章	FORTRAN概説 10- 4
第2章	FORTRANプログラムの形式
第3章	プログラムの構造 10-8
第4章	直線形のプログラム10-12
第5章	選択形のプログラム
第6章	反復形のプログラム(1)
第7章	反復形のプログラム(2) 10-18
第8章	反復形のプログラム(3)10-22
第9章	反復形のプログラム(4)10-24
第10章	1 次元配列とプログラム10-26
第11章	2 次元配列とプログラム
第12章	文関数を引用するプログラム
第13章	関数副プログラムの引用
第14章	サブルーチン副プログラムを引用するプログラム 10-34
第11部 (	C言語 ····································
	C 言語の概要
第2章	数値の計算11- 9

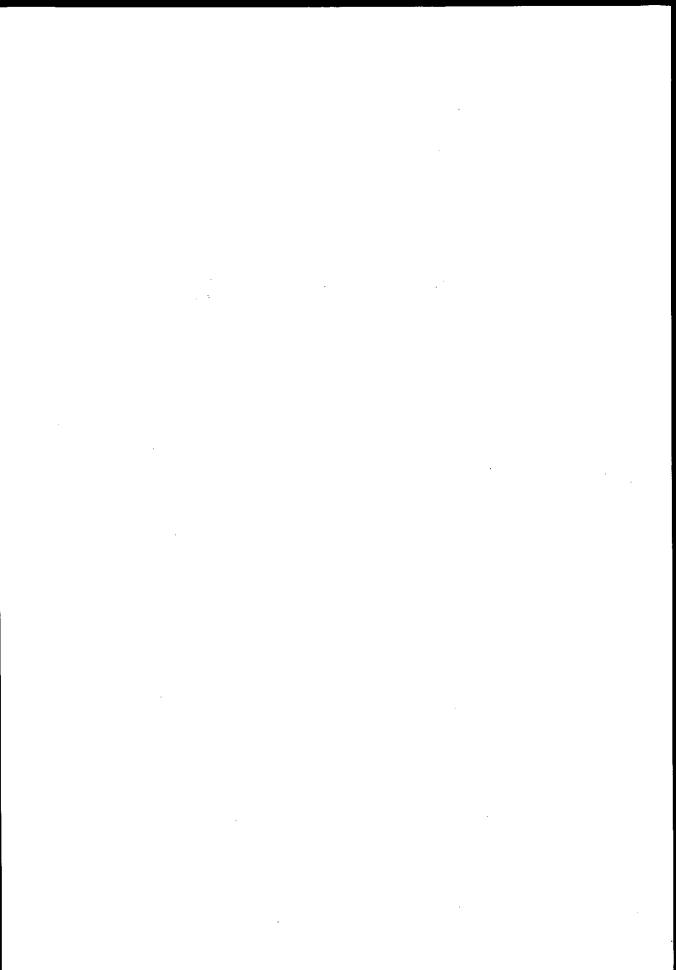
第3章	選択型のプログラム	11-12
第4章	反復型のプログラム	
第5章	ビット演算 ····································	
第6章	入出力処理	11-26
第7章	配 列	11-30
第8章	文字処理	11 – 34
·第9章	ポインタ変数	11-40
第10章	関 数	11 – 47
第11章	標準ライブラリ関数	11-51
第12章	記憶クラス	11-55
第13章	ファイル処理	11-58
第14章	特徴的なプログラム	11-64
第12部 シ	ノステム開発技法	12- 1
第1章	システム開発とシステム設計局面	12- 3
第2章	プログラム設計技法	
第3章	プログラミング技法	12-19
第4章	プログラムのテストと品質管理	
第5章	システムの導入と運用	
第6章	システム開発の進め方	12-35
第13部 表	表現技法	13- 1
第1章	文章の書き方・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	········· 13 – 3
第2章	ビジュアル表現の仕方	13-16
第3章	話し方の技術・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	13-26
·		t,

: .-.

3

· .

総 論 編



本指針が指向する情報処理技術者は、初級技術者ではあるが、企業等組織体において、情報処理を本務あるいはそれに近い形とする技術者である。

### 第1章 指針の改訂についてとなる 性点は (3)

#### 1.1 指針改訂の背景

本指針は、当研究所の前身である情報処理研修センターが、昭和61年4月に 作成した初級情報処理技術者育成指針(以下、「前初級指針」という。)に再 検討を加え改訂したものである。

まず改訂に至った背景について述べる。「前初級指針」の作成された昭和61年当時、ソフトウェア技術者の不足、殊に高度な技術者の不足は、すでに重大な問題であり、産業界で働く、時代に適応した情報処理技術者育成の指針の作成は強く要望されていた。その要望にそって、将来SEになる技術者の第一歩としての初級技術者育成のために「前初級指針」は作成された。その当時、各種の調査を通して、情報処理技術者像は、かなりよく描き出されてはいたが、概念としての未整理あるいは不統一な部分も多かった。また、情報処理技術者の種類、キャリア・パスの実情は、企業によってかなりばらつきがあり、また概念的にも未整理であった。さらに、専修学校の卒業生が、これからの情報産業で重要な役割を担うということは、かなりよく認識されてはいたが、専修学校の情報処理教育の実体は、ほとんど把握されていなかった。

「前初級指針」の完成とほぼ時を同じくして、通商産業省では、2000年に向けての情報化人材対策を検討するため、昭和61年3月に、産業構造審議会情報産業部会に、情報化人材対策小委員会を設置した。約1年間の審議を通して、昭和62年4月に、問題点の指摘とともに、当面講ずるべきいくつかの提言(以下、「提言」という)を行った。

中央情報教育研究所は、「提言」に基づき、産学共同の情報処理教育推進の中核機関として昭和62年6月に新組織として再出発した。「提言」では、技術者の職務内容、知識・技術に着目して、限定的なものとしながらも、技術者を

アプリケーション・エンジニア(AE)、プロダクション・エンジニア(PE)、テクニカル・エンジニア(TE)、デベロップメント・エンジニア(DE)に分類している。当研究所では、この「提言」の分類案を受けて、更に、テクニカル・エンジニアの高度な専門性を考慮して、これをネットワーク・エンジニア(NE)、データベース・エンジニア(DE)とナレッジ・エンジニア(KE)に分け、次の高度情報処理技術者の育成指針を作成した。

- (1) アプリケーション・エンジニア
- (2) プロダクション・エンジニア
- マミ(4) データベース・エンジニア マー・ファー・ファー・ファール
  - (5) ナレッジ・エンジニア

更に、必ずしも情報処理技術者とは言えないが、情報処理システムの健全化に は欠かせない重要な役割を持つ

- (6) システム監査エンジニア
- の育成指針をも作成した。そして、この指針作成の一貫として、これらの技術者が共通して持つべき知識・技術についての指針
- を作成した。

こうして、2000年に向けての高度情報処理技術者像、その職務、必要な知識 技術が明らかになり、その育成の歩みが始まった。初級情報処理技術者は、 やがて、ここであげられている高度情報処理技術者に成長していくべき者であ る。従って、初級情報処理技術者の育成は、高度情報処理技術者育成の前段階 として、高度情報処理技術者育成につながっていくべきものである。「前初級 指針」は、高度情報処理技術者の概念が未成熟なまま作られたものであり、こ の観点から再検討が必要になった。つぎに、情報処理技術の進歩は速く、初級 技術者といえども、それに対する社会的要請は時と共に変化する。その変化の 観点からも再検討が行われた。また、「前初級指針」は、既に社会の広範囲で 活用されてきている。そうした中からも、建設的なさまざまな要望が出されて いる。初級指針は、専修学校のカリキュラムは必ずしも同じものではないが、 この点も考慮に入れて再検討は行われた。 なおここで、前記6種類の情報処理技術者の概要を説明しておく。

- (1)アプリケーション・エンジニアとは、業務部門の立場に立って対象業務分野における最適システムを構築するエンジニアである。要求される能力を一言で言うならば、対象とする業務についての知識とそれを最適なものとしてシステム化する技能である。
- (2)プロダクション・エンジニアとは、現在および今後実用化が予想される広い範囲でのアプリケーションに関して、共通に必要となる基礎的なソフトウェア技術を、単に経験の集積や特定のシステムやアプリケーションに存在したものとしてではなく、理論的な基礎の上に立った工学として身につけたエンジニアである。
- (3)ネットワーク・エンジニアとは、ユーザのニーズを把握し、目的とする業務について情報処理と通信を統合した最適ネットワーク・システムを企画、構築、運用のできる、またネットワーク・システムに関するコンサルテーションが行えるエンジニアである。
- (4)データベース・エンジニアとは、目的に適したデータ構造を設定し、DB MSの選択や改善を行い、データベースを構築するとともにその評価・改善を行うことができるエンジニアである。
- (5)ナレッジ・エンジニアとは、知識処理技術を習得し、既存技術との最適機能分担の設定を含め知識処理システムを構築し、評価・改善を行うことができるエンジニアである。
- (6)システム監査エンジニアとは、システム監査の専門家であり、監査対象から独立した客観的な立場で、コンピュータを中核とする情報システムを総合的に点検・評価し、関係者に助言・勧告するとともに、その有効利用の促進と弊害の除去とを同時に追求してシステムの健全を図る監査人である。

#### 1.2 改訂の内容

「前初級指針」からの主要な改訂点について述べる。

まず、各部の教育目標、各章の学習目標をより明確にし、また、指導上の留意点をより詳しくかつ具体的にした。次に高度情報処理技術者へのつながりを考え、初級情報処理技術者を、

- ・アプリケーション・ソフトウェア指向
- ・基本ソフトウェア指向

に分けて、プログラム言語について選択方式を取り入れた。例えば、基本ソフトウェア指向の人は第11部「C言語」を、エンジニアリング指向の人は第10部 FORTRANを、ビジネス応用指向の人は第9部[COBOL]を選択するといったことである。また、「実務能力編」の各部・各章において、必要に応じてOJTについて触れた。

情報処理技術者には、情報処理技術のみならず、さまざまな知識・技術が必 要である。初級情報処理技術者といえども同様である。しかし、本指針には、 情報処理に直接関係した知識・技術のみを含めた。従って、「前初級指針」の「関 連知識編しは、情報処理に直接関係する部分を新「基礎知識編」に移し、かな りの部分が削除された。即ち、第1部「情報処理一般」は、情報処理に直接関 係した知識であり、また学習順序としては、新「基礎知識編」第6部「情報化 とコンピュータ」で学習するのを適当と考え、この中に含めた。第2部「エレ クトロニクスの基礎」は、一部分を第2部「ハードウェア」に移し、他は削除 した。第3部「商工業に関する一般知識」は、「経営情報システム」を第6部 「情報化とコンピュータ」に移したが、他は削除した。もちろん、削除された 部分が、初級情報処理技術者に不必要だというのではない。本指針には含まれ てはいないが、例えば、時系列、回帰分析等の統計的諸技術、在庫管理、線型 計画法、PERT、待行列、シミュレーション等は、情報処理技術そのものではな いので、本指針には含まれていないが、それらの入門的事項は、情報処理業務 では、常識的事項のように、日常きわめて頻繁に使われることである。自習し 習得しておいて欲しい。

「基礎知識編」の改訂の主要な点から述べる。「前初級指針」の使用経験から、前第1部「情報処理とコンピュータ」は、実務経験、情報処理の経験の全くない初心者には入門としては取り付きにくいので、これを、第1部「コンピュータ入門」と第6部「情報化とコンピュータ」の2つに分けた。第2部「ハードウェア」では、歴史的変遷の背後に流れるwhyを含めるようにした。また、汎用機だけでなく、パソコン、ワークステーションにも触れるようにした。また最近の急速なネットワークの普及に応じて、前第7章の「データ通信シス

テム」を「通信ネットワーク」として独立した部にした。そしてここには、前第4部「情報処理システム」の第3章「コンピュータ・ネットワーク・システム」も含めた。第3部「ソフトウェア」は、基本知識という観点から、内容をオペレーティング・システム(OS)中心にし、「ソフトウェア作成手順」を削除した。前第4部「情報処理システム」では、まず第3章の「コンピュータ・ネットワーク・システム」を「通信ネットワーク」に移した。また最新のデータベース・システムの演ずる役割の重要性を考え、第4章の「データベース・システム」を「実務能力編」に移し、この部からは削除した。しかし、一方では情報処理システムは、激しい技術進歩を直接に反映する分野である。その最近の進歩に照らして、マルチ・メディア処理を取り上げるとともに、AIシステム、CAI/CMIをも章として扱った。

次に「実務能力編」について述べる。前第1部「プログラム流れ図の作成」 は、「プログラム・アルゴリズム」と名前を改め、構造化概念を含めると共に、 フローチャートのみならず、NSチャート等をも取り入れた。前第2部「ファ イル」を、「ファイルとデータベース」とした。ファイルについては、新メデ ィアの追加、VSAMを充実し、「フレキシブル・ディスク」の章を見直し改 訂した。データベースでは、データモデル、DBMS、データベース言語を含 めた。各章の内容、指導上の留意点については、具体性をもたせるようにして 充実した。第9部「COBOL」は、文法中心でなく、例題中心に進めた。第10部「FO TRAN」には大きな変更はない。プログラム言語としては、C言語の需要が近年 高まってきている。そこで新たに第11部として「C言語」を加えた。前第5部 「システム開発と運用」は、現状では、運用はオンライン・ネットワーク管理 が主体となっている。この職務は初級技術者のレベルではこなしきれないので、 簡単に触れる範囲にとどめた。本指針では、新たに第13部として「表現技法」 を設けた。情報処理システムは、文書と口頭によって伝達され理解される。こ の文書と口頭には、システム開発と運用に際して直接に必要となる各種の文書 もあれば、ユーザへの計画の提案、説明、連絡、また、ユーザへの完成したシ ステムの説明、解説等多岐にわたるものがある。情報処理技術者としては、き わめて頻繁に直面する事柄で、これに対処する技術は必須のものであり、情報 処理技術の一環をなすものといえる。その基礎になるのが、ここで取り上げた

# 「表現技法」であり、情報処理技術者になろうとする者は初級情報処理技術者になるための初期の段階から、しっかりした訓練を積んでおくべき事である。

#### 2.1 初級情報処理技術者

一般に、初級情報処理技術者という言葉は、日常的な常識的な言葉としては、 かなり幅広く用いられている。企業に入り、情報処理の入門的訓練を受けて職 場に配置された者あたりから、高度情報処理技術者と称される者以外のすべて の情報処理技術者に対する包括的な名称のようである。日常用語としては、そ れでよい。しかし、この総論編の冒頭に書いたように、本指針が指向する情報 処理技術者は、初級であるが、企業等組織体において、情報処理を本務あるい はそれに近い形とする技術者である。そのような技術者であるための第1歩は 、本指針の内容を学習し習得することである。これは、初級情報処理技術者に なるためには、最低限必要なことである。これによって、一応、通称の初級情 報処理技術者としての第1歩を踏み出すことになるわけである。しかし、この 段階ではまだ、企業等によっては、"見習い"と称し、初級情報処理技術者と 区別するところも多い。初級といえども、技術者であるためには、所属する企 業等組織体において、生産活動の一端を担うものでなければならない。そのよ うな技術者になって、はじめて、真の初級情報処理技術者と言うことができる であろう。本指針で言う初級情報処理技術者とは、いま述べた '真の初級情報 処理技術者 のことである。この真の初級情報処理技術者になるためには、本 指針の学習・習得の後、あるいは学習と平行して、OJTによる訓練を受け、 実務経験を経ることが必要である。本指針は、OJTについても触れているが、 OJTには、指針には含みえない点も多いことを注意しておく。企業等組織体 によって、また時代によって、直面する実務は異なるが、こうした実務をも経 験して、はじめて生産活動に携わることができるものである。こうして、真の 初級情報処理技術者としての活動が可能になるのである。こうなるためには、 個人差はあるが、多くの場合、この指針の習得後2~3年を要するであろう。 そして、高度情報処理技術者に進む足場を得ることになるのである。

企業等組織体において、初級情報処理技術者はどの程度の生産活動に携わるかであるが、それはつぎのような職務を責任を持って担当することになるだろう。

- (1)上級者の指示に基づいて、業務の現状の情報(データの流れ,入出力情報,蓄積情報,業務と組織の関係など)を収集し、整理する。
- (2)上級者の指示に基づいて、システムの外部設計(概要設計)に必要な情報や資料を収集し、整理する。
- (3)上級者の指示に基づいて、画面設計、コード設計、論理データ設計などシステムの内部設計(詳細設計)を行い、システム詳細仕様書を作成する。
- (4)仕様書に基づいて、プログラムの構造化設計を行う。
- (5)プログラム・モジュールの論理設計を行い、プログラムをコーディング する。
- (6)作成したプログラムのテスト・ケースを設計し、単体テストおよび総合 テストを行う。
  - (7)プログラムの保守、改善を行う。

#### 2.2 本指針の対象

本指針は、情報処理技術者としての第一歩を踏みだそうとする者の育成指針である。企業内においては、情報処理技術者を目指す新人の教育に対する指針である。また、専修学校等において、初級情報処理技術者の育成を目的とするコースの教育、目標を与えようとするものである。従って、その対象者としては次の者を想定している。

- (1)高等学校修了程度の学力を持っている。
- 📻 (2)情報処理技術についての知識が全くない。
  - (3)企業等組織体において、情報処理に携わり始めてから日が浅い、あるいはこれから携わろうとしている。
  - (4)将来、高度情報処理技術者としての活躍が期待される。

高等学校修了程度の学力とは、国語、数学、理科、社会、英語に関する基礎学力である。特に高度な学力は求められないが、国語では、正しい日本語の読み書きの能力と発表の能力が重要である。英語では読解力が求められる。

#### 2.3 本指針の目標

情報処理技術者としてまず第一に求められることは、「何を知っているか」ではなくて、「何ができるか」である。情報処理技術者の育成もその線に沿ったものでなければならない。本指針は、この「何かができる」情報処理技術者の育成のための指針であり、「何かが」とは、2.1節で揚げた職務(1)~(7)である。

本指針は、初級情報処理技術者の育成指導に当たる講師が、「何を、どのように」教えるかについての指針を与えるものである。まず、ここで強調しておきたいのは、本指針では、基礎を重視しているということである。基礎という言葉に誤解がないように簡単にコメントしよう。基礎とは初等的な知識や技術のことではない。例えば、COBOLの細かい文法知識などは、初歩的知識・技術ではあっても、ここでいう基礎ではない。それに対して、プログラムの論理は、重要な基礎である。

まず「何を」ということから簡単に述べておく。一般に情報処理技術者には、実務を遂行していく上で、自由に駆使し実践できるように身につけておくべき知識・技術と、それらを身につけるために必要な、そしてまた、それらを活かして使うために必要な知識がある。本指針では、それらの中で初級情報処理技術者にとって必要なものを取り出し、後者を基礎知識編に、前者を実務能力編に収めた。

「基礎知識編」は、「コンピュータ入門」、「ハードウェア」、「通信ネットワーク」、「情報処理システム」、「情報化とコンピュータ」の6部から構成されている。「コンピュータ入門」の部は、オリエンテーションのコースで、コンピュータにさわってみることから出発して、コンピュータの特徴、コンピュータによる情報処理、社会でのコンピュータの効用といった、常識の概観を与える。「ハードウェア」の部では、コンピュータの基本構造、データ表現から、記憶、演算、制御、入出力の各装置について述べ、ハードウェア面からみたコンピュータの歴史と将来について述べる。「ソフトウェア」の部では、ソフトウェアの体系を概観し、言語処理系、オペレーティング・システム、サービス・プログラムについて、さらに、ソフトウェアの課題について述べる。「通信ネットワーク」の部では、その役割、基礎技術、方式、利用、そして安全性と、基本的な事柄を述べる。初級情報処理技術者なので、ハードウェア、

ソフトウェア、通信ネットワークのいずれも、内容は基礎知識であって、それらのシステム設計には立ち入らない。「情報処理システム」の部では、処理方式、システム構成についての基礎知識から最近のマルチ・メディア処理、AIシステム、CAD/CAMシステムの現状を解説する。「情報化とコンピュータ」の部では、企業、社会、行政、地域における情報化の、近未来の動向を見、情報化社会の課題について述べる。

「実務能力編」は、「プログラム・アルゴリズム」、「ファイルとデータベース」、 「COBOL」、「FORTRAN」、「C」、「システム開発技法」、「表現技法」の7部から 構成されている。ここで、プログラム言語としてCOBOL、FORTRAN、Cの3つ が取り上げられているが、この3つをすべて履修し習得することを求めている のではない。この3つの中のどれか1つを履修し習得することを求めているの である。実務能力編に含まれるものは、既に述べたように、実務を遂行してい く上で、自由に駆使し実践できるように身につけておくべき知識・技術である。 十分な演習、実習を行いながら指導しなければならない。「プログラム・アル ゴリズム」の部では、まず流れ図によるアルゴリズムの表現から出発し、NS チャートによる表現を与えるが、流れ図やNSチャートそのものが学習の目的 ではない。アルゴリズムの表現、ことに構造化されたプログラム理論をきちん と身につけることが主たる目的である。これは、プログラミングの中で生かす べきものであるが、プログラミングでは、言語特性などのため、アルゴリズム そのものが見えにくい傾向があるので、まず最初に学習すべきものとして取り 上げた。構造化されたプログラム論理も、抽象的に上滑りしたのでは意味がな いので、事務処理のためのアルゴリズム、科学技術計算のためのアルゴリズム、 表探索、内部分類等の具体的問題について、構造化されたアルゴリズムを学ぶ ようにしている。そして最後にアルゴリズムの検証を通して、構造化されたプ ログラム論理の重要性を身体で理解するようにしている。ファイルとデータベ ースは、情報処理には不可欠な基本的なものである。この部では、ファイルに ついてはその概念、格納しておく記憶媒体から、順、直、索引順、区分編成の ファイル、VSAMファイル、パソコンのファイル等の基本的事項を学ぶよう にしている。データベースについては、概念、システムの構成、DBMS、デ ータベース言語について学ぶ。プログラム言語「COBOL」、「FORTRAN」「C」の

部は、問題の処理のためのアルゴリズムを、その言語を用いたプログラムによ って表現するという観点で述べる。プログラムが文法に従って書かれるのは当 然であるが、文法が中心にあるのではなく、問題を処理するアルゴリズムを中 心に記述が進められる。そのために、文法からではなく、例題から入るという 進め方をとっている。前述したように、3つの中の1つを習得するのである。 2つ以上を同時並行的に学習することは、害はあっても益はない。まず1つを 完全に習得しなければならない。2つ目を学習するのは、1つを完全に習得し た後でなければならない。この点を強調しておきたい。「システム開発技術」 の部では、アプリケーション・システムの開発からプログラム開発、プログラ ム・テスト、システムの保守に至るまでのすべての局面についてグループ演習 ・実習を交えながら学習する。この部は、実務能力編でここまで学んできたこ とを駆使すると共に、それらを総合的に体系ずけるものとなる。また、ここで のグループ学習は、つぎの表現技法で学ぶ事柄の実践訓練にも役立つものであ る。「表現技法」の部では、文章の書き方、ビジュアル表現の仕方、話し方の 技術という表現に関する最も基本的な事柄について、演習を中心にして学ぶ。 既に述べたことではあるが、情報処理技術者にとっては、普通のビジネスマン 以上に表現技法は重要である。提案書の作成、要求定義書、システム設計書等 の各種システム仕様書の作成報告書の作成など、ソフトウェアとは文書の集合 体である。そして、提案書にもとづいての提案の説明、仕様書はマニュアルを もとに関係者にわかりやすく説明するなど、口頭表現も重要な技術になる。そ れらを円滑にわかりやすく進めるためにはビジュアル表現が重要になる。ここ では、初級情報処理技術者であるから、提案書、設計書等の具体的な内容では なく、表現技法の基本を学習することになる。尚、ファイルとデータベースは、 互いに関連の深いものであるが、概念的にはファイルの延長線上にデータベー スがあるわけではなく、両者は異なるものである。両者の関係の深さから一つ の部にしてあるが、概念上の相違から、教育上の運用としては、両者を分けて 教育することも考えられる。

本指針で学ぶことは、この受講期間だけで完全に身につくものではない。良く消化され、血となり肉となるためには、受講後の職場における〇JTが重要である。本指針には、初級情報処理技術者を育成するのに必要な〇JTのガイ

ドが実務能力編の各部の「指導上の留意点」に含まれている。現実の社会では、OJTが職場への放置と同じ意味で使われていることがよくある。もちろん、教育を受ける人間は千差万別で、一様なOJTは困難である。しかし、OJTは育成の重要な課程なのであって、目的を持って個人個人のレベルに合わせて計画的に行われなければならない。本指針で与えられるものは、どこまでもガイドであるから、さらに一段の工夫を加えていって欲しい。

「どのように」というのは、明確に完全には規定することができない。しか し基礎の重視ということは、直接の方法ではないけれども、「どのように」と いうことへの理念的指針である。そして、方法については、経験的に良い方法 として認められるものはある。教育工学的に研究された方法もある。そうした ものは、積極的に採用すべきであろう。しかし、教育の方法には、属人的な点 もあるので、細部にわたる規定は避けたほうがよい。しかし、この方法として 「の「どのように」ということについては、現場からの要望も強いので、指導上 の留意点等でも可能な限り取り入れた。また、COBOL 、FORTRAN 、C言語とい ったプログラミングの指導については、文法規則からよりも例題から導入する といった方法を取り入れている。なおここで、一般的な簡単な注意を与えてお こう。本指針が扱っているものは、職業教育である。本指針では基礎を大事に し、原理原則を大切にする。しかしこのことは、原理原則を常に表面に立てて 教育することではない。ある種の教育では、原理原則を出発点とする。これが 極めて高い教育効果をあげる場合がある。しかし、一般的な職業教育では、原 理原則は、むしろ目標なのであり、最後まで表面には出ない隠れた目標になる こともある。教育は可能な限り実際的な問題の上で行われた方がよい。そして 、その問題解決の過程の中に、原理原則を見えないように埋め込む努力も必要 である。そして、実際的な問題を扱いながら現象に振り回されず、同時に、原 理原則に立ちながら原理原則に縛られないことが重要である。

# 2.4 初級情報処理技術者育成の指導者

職業技術者にとって最も重要なことは、既に述べてきたように「何ができるか」である。育成に当たる講師も特にこの点に留意しなければならない。各教 科毎に、その教育訓練によって「何ができる」ようになるかの目標を明確にし て育成に当たらなければならない。また、単に知識、理論の切り売りではなく、 実務に即した内容を織り込むことが必要である。しかもしっかりと原理原則に 立ち、これを教育の中に埋め込むためには、講師は不断に勉強をしなければな らない。職業技術者になる自覚と意欲を持つ学習者に対しては、落ちこぼれを 作ることは可能な限り避けるべき事であり、目標に対する到達度を測りながら 育成を進めるべきである。講師は、教育方法についても評価を持ちながら、よ りよい方法を工夫し、学習者の意欲を引き出すよう努力しなければならない。

初級情報処理技術者の育成は、情報処理技術者の基礎である。育成された技 術者は更に研さんを積み、より高度の技術者に成長するが、初期に受けた教育 はきわめて大きい影響を持ち続けることが多い。講師は初級教育の重要性を充 分に認識して育成に当たって欲しい。また、情報処理技術者の持つべきモラル も、初期の時期から培うべきである。

情報処理技術の進歩・変遷は急速であり、情報処理システムも多様化してきている。また情報処理技術の適用業務も急速に広がってきている。こうした社会の進展に対応して行くためには、講師自身の自己研さんがきわめて重要である。また、育成に当たる講師には、情報処理の狭い技術だけではなくて常に広い視野をもつことが要請される。

#### 2.5 情報処理技術者試験について

まず冒頭で、平凡かつ当然のことを述べておく。「本指針で示す初級情報処理技術者たらんとする者は本指針の学習のどこかの段階で、少なくとも第二種情報処理技術者試験には合格していてほしいものである」。

本指針は、初級情報処理技術者の育成を目的とした指針である。第二種情報 処理技術者試験も、概ねそれに近い目的を持っているが、以下で概説するよう に、もう少し幅の広い目的を持っている。従って、両者の間には、範囲と深さ において違いが出ることは避けられない。従って、本指針は、この試験とは直接の関わりを持っていない。また試験に合格したからと言って、本指針を習得したことにはならない。

以下では、まず情報処理技術者試験について概説し、更に、冒頭で述べた言葉について若干の蛇足を加えておく。

情報処理技術者試験は、昭和44年から実施されてきている。当初は第一種情報処理技術者試験(シニアプログラマを対象)及び第二種情報処理技術者試験(一般プログラマを対象)の区分だけであったが、46年度から特種情報処理技術者試験(システム・エンジニアを対象)、61年度から情報処理システム監査技術者試験、63年度からオンライン情報処理技術者試験が追加されている。その目的は

- (1)情報処理技術者に対して目標を示し、刺激を与えることによって、その 技術の向上を図ること。
- (2)情報処理技術者として備えるべき能力についての基準を示すことにより、 教育水準の確保に資すること。
- (3)情報処理技術者の評価に関して客観的な尺度を提供すること。
- (4)広く受験の機会を提供することにより、国民各層の情報化に対する意識をかん養すること。

などにより、情報処理技術者を育成することである。

この試験への応募者は、平成元年度では483,052名にも及び、累積の合格者も275,899名を数える(オンライン1,311名、システム監査1,502名、特種11,220名、一種52,443名、二種209,423名)。この数字からもわかるように、情報処理技術者が、第二種情報処理技術者試験に合格していることは、当然すぎる程当然の事なのである。そして、情報サービス産業では、試験の合格者に対し、何らかの資格給を導入している企業が多い。このように、情報処理技術者試験は、制度として定着してきており、技術者の評価、モティベーションとして相応の役割を果たしてきているものと言えるだろう。

ま省機械情報産業局編、昭和62年5月)で指摘しているように、まず、情報処理技術者の意識調査によれば、30%が「技術の目標を設定する上で有効である」と一応の評価を与えているものの、「技術レベル評価に権威がある」という積極的評価は7.7%に過ぎず、「実力評価にはもの足りない」とする者が23%にのぼることからも、試験の内容等については、まだまだ改善の余地があるだろう。また、現行の試験の具体的な問題としては、まず、試験内容が必ずしも技術進歩、社会の要請に応じていない、実務とかい離している等が指摘されている。

広範囲の不特定多数の人を対象にして行われる情報処理技術者試験では、職 場や人によって千差万別な業務そのものを問題として出すことはないであろう。 抽象化を行ったり、各業務に共通するものをひろいだしたり、あるいは、業務 に近いが少し離れたものを見つけて、問題にすることになるだろう。従って、 このような試験に対処するためには、共通的、基本的な知識・技術をきちんと 身につけておくべきこと、そして、日常業務を抽象化して、それを多面的に応 用する能力を身につけておくことが要請される。試験には一般に少なくとも2 種類のもの ― 成績を他人と競う試験と、一定の水準に達していることだけが 問われる試験 ― がある。入学試験は前者の代表例で、医師の資格試験、自動 車の運転免許取得のための試験、そして情報処理技術者試験などは後者の例で ある。入学試験では、人と人との競争は熾烈に、問題は難しくなり、受験戦争 と言われるまでになる。水準達成のみが問われる試験ではこうしたことはない。 成績は良いに越したことはないが、満点を取る必要がなく、求められている水 準に達していればよい。だからといって、受験勉強が全くなくなるわけではな い。例えばつぎのような形の受験勉強は必要であろう。そうして、これらは、 入学試験の受験勉強にからまるような弊害を伴うことはない。

- 1. 試験の前に、たとえ知っていると思われることであっても、基礎的事項 を復習的に見直す。
- 2. 知識・技術に偏たりがある場合には、不十分な事を勉強する。
- 1.の受験勉強は、受験をする限り必要な事であろう。また、2.で述べられていることは重要である。一般に技術者には専門がある。しかし、特化された専門の中だけに閉じ込もると、その専門の能力さえもが成長をとめてしまう。人間の能力は、特定の専門の中だけで純粋培養をすることはできないもののようである。論理的思考といえども、幅広い生活体験の中で、知らず知らずのうちに培われていく部分が多い。その意味で、ただいま直接に使われないことでも、知識・技術の幅を広げることは、技術者としての成長の基盤になるものである。

第二種情報処理技術者試験が対象とする「一般プログラマ」には、明確な性格ずけが与えられていないが、初級情報処理技術者の入口あたりに立つ者を指すと考えられる。そして、目的の(4)にあるように、広く受験の機会を提供する関係上、受験資格に実務経験を課していない。試験問題も、技術以前の基礎的

事項が中心で、例えば学生達にも、気軽に受けられるようになっている。そして、在学中の学生の合格者も多く、目的(4)にあるように、国民各層の情報化に対する意識のかん養に貢献しているわけである。このように、第二種情報処理技術者試験では、その合格者は、本指針のように、企業等組織体において、情報処理を本務あるいはそれに近い形とする技術者を指向するとは限らない。従って、本指針はより掘り下げたものになる。その意味でも、第二種情報処理技術者試験の合格が、本指針の習得を示すことにはならないわけである。また、この試験は実務経験を課していないのであるから、これに合格しても、情報処理技術者への道はまだ平坦ではないことを認識しなければならない。

最後に、つぎのことを注意しておく。それは、「本指針の学習か、第二種情報 処理技術者試験を目標にした勉強か」といった、あたかも二者択一のような命 題は意味がないということである。受験を目標にして教育すれば合格者が一気 に増えるというのは、一種の錯覚に過ぎないだろう。20年間も試験を継続して 行っていれば、出題にある程度の傾向が出るのは当然である。しかし、この傾 向への対策に明け暮れてみても、合格者が急増することはないであろう。基礎 のないところでは、いかなる受験対策も効果を期待することは困難であろう。 すべてに先立つものは、しっかりと基礎を養うことである。試験には、しっか りした基礎に立って、自らの知識・技術を整理し、不足が有ればそれを補って 対応すべきであろう。

#### 第3章 カリキュラムの編成

#### 3.1 本指針の使い方

本指針は、企業における新人教育、あるいは専門学校において情報処理を専攻する課程の教育の指針である。企業での教育方法と学校での教育方法が違うように、教育方法は、その対象、その環境によって違ってくる。したがって、実際の教育にこの指針をどう活用するかは、教育企画者の判断に委ねられる。

第2編の基本知識編と第3編の実務能力編の編、部、章、節の順序は、カリキュラム編成上にも十分に意味のある順序である。すなわち、まず基本知識として、コンピュータ入門から入り、ハードウェア、ソフトウェア、通信ネットワークの基本知識、さらにそれらを総合した情報処理システムを学ぶ。そして、情報化とコンピュータで、これらのシステムが、社会的にどのように活用されているかについてのしっかりした認識を持つようになる。それに続けて実務能力編で実務能力を身につけることになる。まず、プログラム・アルゴリズムでしっかりとプログラムの論理を身につけ、ファイルとデータベースの一般的知識とこれらの扱いを習得した後、プログラミングをしっかりと身につけ、システム開発技法を学び実習することによって、既習の知識・技術をまとめるわけである。表現技法については、実習の比較的初期の段階から並行して行い、演習や実習のレポートの作成、結果の発表等でも実践的に訓練させることが望ましい。

しかし、学校教育ではこうした方法が必ずしも良いとはいえないし、また現実的ともいえない。学校教育ではそれぞれの科目を短期間に集中して学習することは希であり、さまざまな科目を並行して、学期あるいは学年を通して学習するのが普通である。この場合には、科目間の前後関係、どの科目とどの科目を平行させるかなど、工夫が必要である。

尚、カリキュラムとは直接関係はないが、本指針の末尾に本文中に現れる用語の索引がある。これらの用語はその意味を含めて、初級情報処理技術者として知っておくべきものであり、また、講師が教える際の参考にもなるものである。これらは、日本語の用語(タスク、ジョブ制御言語、仮想記憶等)の部と、略語(OS、JCL等)の部に分かれている。日本語はアイウエオ順、略語は

A、B、C順に配列されている。用語には、もともと英語であったものについては英訳をつけてある。また、JIS情報処理用語では、例えばオペレーティングシステムであるが、本指針ではオペレーティング・システムとしてあり、JIS情報処理用語との違いが若干ある。

#### 3.2 標準的な時間配分

基本知識、実務能力編の各部に割り当てられる標準的な時間配分を次に示す。 各章、節への時間配分は各部に記されている。

基本知識編	講義	演習	計
第1部 コンピュータ入門	8	1 2	2 0
2 ハードウェア	3 6	4	4 0
3 ソフトウェア	3 3	4	3 7
4 通信ネットワーク	6 6	1 5	8 1
5 情報処理システム	4 0		4 0
6 情報化とコンピュータ	3 2	8	4 0
·		小計	2 5 8
実務能力編			
7 プログラム・アルゴリズム	3 8	5 5	9 3
8 ファイルとデータベース	39.5	5. 5	4 5
9 COBOL	7 8	1 2 0	198
10 FORTRAN			
1 1 C言語			
12 システム開発技法	29	4 4	7 3
13 表現技法	9. 5	34.5	4.4
		小計	4 5 3
		合計	7 1 1

時間配分は講義と実習に分けてあるが、テーマによっては、講義の中で演習問題を出し自宅学習をさせるとか、講義に机上演習を取り入れるなどの方法も考えられる。またこの時間配分は標準的な目安を示しているに過ぎない。実施時には学習者の種類、教育環境に応じた時間配分を考えることが望まれる。

### 3.3 企業内教育での編成例

基本知識編と実務能力編の合計時間は約700-750時間であり、これは企業内 教育としては約4.5ヶ月の時間に相当する。基本知識編と実務能力編だけのカ リキュラム編成の1例としては次のものが考えられる。

基本知識編	日 数	講義順
コンピュータ入門	2日	1
ハードウェア	5	10
ソフトウェア	5	3
通信ネットワーク	5	1
情報処理システム	1	7
情報化とコンピュータ	2	2
実務能力編		
プログラム・アルゴリズム	5	<b>(4</b> )
ファイルとデータベース	5	(5)
COBOL	)	
FORTRAN	22(いづれか1	つ) ⑥
C言語	J	
システム開発技法	5	8
表現技法	3	9
合 計	6 0	

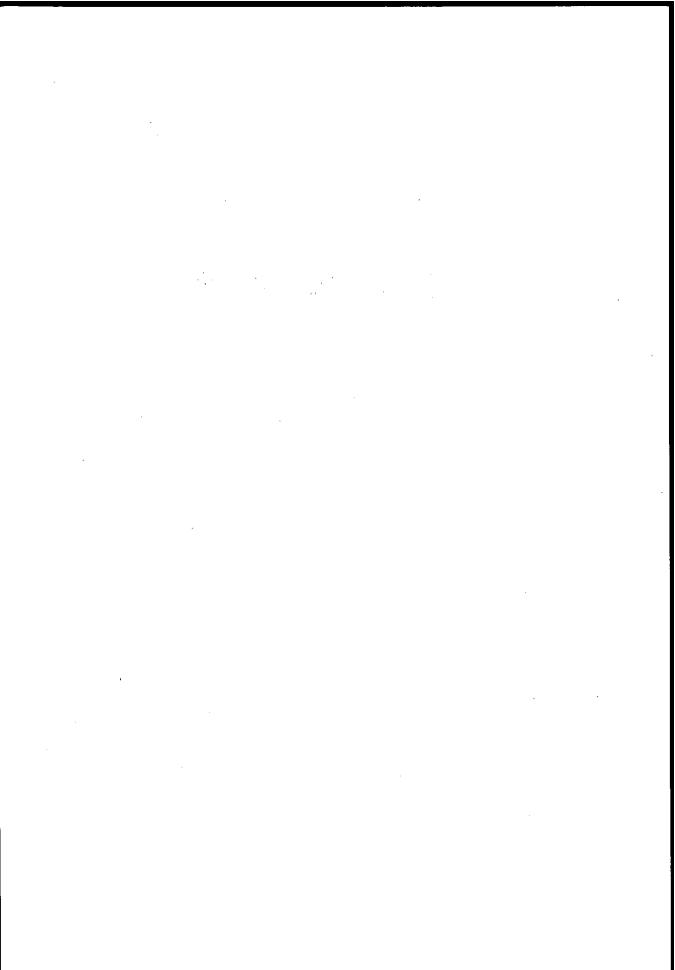
### 3.4 コンピュータ専門学校におけるカリキュラム編成例

カリキュラムの編成については、次のことに留意しなければならない。

- (1)プログラムの作成においては、複数の言語を曖昧に理解させるのではなく、 1 つの言語について確実に習得できるように図るべきである。
- (2)講義の進め方は、単に知識、技術の切り売りではなく、できるだけ実務に即した内容を織り込むことが必要である。技術革新が激しいので、常に最新技術動向を講義の中に取り入れていく姿勢が必要である。

 $\frac{1}{2(1+\epsilon)^{2}} = \frac{1}{2} \left( \frac{1} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}$ 

# 基本知識編



### 第1部 コンピュータ入門

### 教育目標

それぞれの知識や技術に深く立ち入って理解したり、あれこれ総花的に知識 を得るのではなく、飽くまでも基本を知ることに目標をおく。

「コンピュータとは何か」、「コンピュータ・システムの機能とはどのようなものか」、「情報処理はどのような仕組みか」、「それらはどのような役割をもち、どのような利点があり、どのような効用があるか」などについて、それぞれを把握する。

指導にあたっては、教育メディア(ビデオ教材、サブテキストなど)を活用 し、特にコンピュータの動作原理、構造、仕組みなどは視覚的にとらえる。大 量記憶、高速処理などは、それぞれの単位を正しく知ることによりその感覚を 養う。

実習にあたっては、「体験させる」、「成功させる」、「できる」、「肌で感じる」など、パッケージ・ソフトを使った方法を多く取り入れる。

また、現代社会ではコンピュータが、大変役に立っていることを認識するとともに、その使い方に限界があることも正しく理解する。

# 内容の構成

内容	時間	到 数
	講義	演習
第1章 コンピュータの特徴 1.1 コンヒュータの動作原理・構造・仕組み 1.2 ハードウェアとソフトウェア 1.3 コンピュータの機能と特徴	6	8
第2章 コンピュータによる情報処理 2.1 パッケージソフトの利用	3	6
第3章 コンピュータの効用 3.1 一括処理 3.2 オンライン処理の躍進 3.3 高度情報化社会	5	2
숌 計	. 14	1 6

### 第1章 コンピュータの特徴

### 学習目標

学習者が、この章を修了した時点で、次の事項ができるようになることを目標とする。

- 1. コンピュータの原理、構造、仕組みに関する基本を理解し、道具としてのハードウェアの性格が明確に説明できる。
- 2. コンピュータを汎用データ処理機械として成立させるために必要なソフト ウェアの役割や特徴について説明できる。
- 3. 大量記憶、高速処理に代表されるコンピュータの機能と特徴について概説 できる。
- 4. コンピュータの適用範囲の拡大に、データ通信ならびにマン・マシン・インタフェースの向上が大きく寄与することを理解している。

### 内 容

- 1.1 コンピュータの動作原理、構造、仕組み(420分)
  - (1) 基本構成
    - ① コンピュータの基本構成

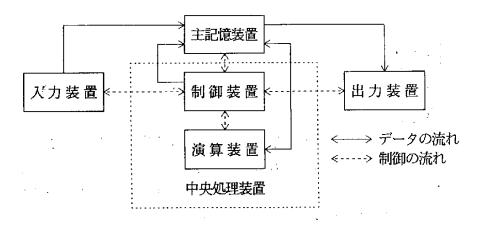
コンピュータは、「計算する」、「データを処理する」機械である。 これらの役割に注目し、演算、記憶、入力、出力、制御の各機能を取り 上げ、それらを基本機能としてコンピュータが構成されていることを解 説する。

大量記憶、高速処理などを目的とする補助記憶装置についても触れる。

② 各装置の機能と相互関係

各装置の基本的機能および装置の相互関係について説明する。実物の 装置、あるいはビデオ教材などを利用して、各装置の動作、役割、機能 などを具体的に把握できるようにする。

図表1-1-1 コンピュータの基本構成



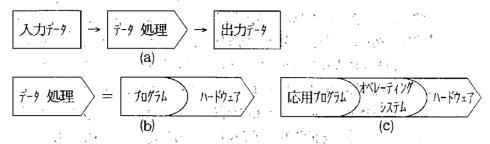
#### (2) データ処理システム

### ① コンピュータによるデータ処理

はじめに、われわれの身のまわりで使われているデータ処理の具体例を取り上げ、データ処理を身近なものとして認識させる。つぎに、コンピュータはプログラムによって様々な機能を実現できる汎用データ処理機械であることを理解させる。

コンピュータは(図1-1-2 (a))に示すように、入力データを処理し、その結果を出力するシステムとしてとらえることができる。このデータ処理システムは(図1-1-2 (b))のように、固定で汎用なハードウェアと、処理手順を記述したプログラムと呼ばれる可変で個別な部分から構成される。更にプログラムは(図1-1-2 (c))に示すように、個々のデータ処理手順を記述した応用プログラムと、ハードウェアを制御するとともに、応用プログラムを実行するオペレーティング・システムに分げることができる。したがって応用プログラムを替えることによって、同一ハードウェアのもとで、いろいろなデータ処理ができることを認識させる。

#### 図表1-1-2 データ処理システム(文献3.より)



### ② データの表現と処理単位。 スタスカーイン

コンピュータが処理する数値、文字、音声、画像などの多種多様なデータは、コンピュータ内部では数値、あるいはコードとして表現され、その取扱いは定められた大きさを基本単位として行われることを理解させる。その際、次の内容を考慮して説明する。

コンピュータでは、電気信号のオンとオフを数値の1と0に対応させることにより、2進数の1桁(ビット)を表現している。したがってコンピュータが扱うデータは基本的に2進数である。この2進数を数値として、あるいはコードとして扱うことによって、データを様々な内部表現形式で実現している。

また内部表現されたデータは、バイト(8 ビット)と呼ばれる基本単位で取り扱われる。この単位は情報の量を表現するものと考えることができる。

### ③ データ処理システムの分類

コンピュータによるデータ処理システムの主な形態について簡単に触れる。コンピュータに仕事をさせる単位をジョブといい、このジョブの 実行による方式や入力形態の相違などについて説明する。実際のデータ 処理システムは、これらを組み合わせた形態をとることに注意する。

### (3) プログラム記憶方式

### ① プログラム記憶方式

プログラム記憶方式の原理を説明し、その柔軟性や汎用性から、現在 のコンピュータによる高速で大量処理の発展と普及があったことを理解 させる。

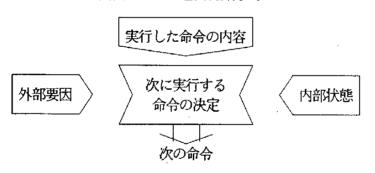
現在、一般に使われているコンピュータは、ノイマン型コンピュータと呼び、この方式は計算の手順を示すプログラムをデータとともに記憶装置に格納し、その手順を逐一読み出し、解読することによって実行する。この方式によってハードウェアの役割を明確にすることができ、一方では処理手順を記述したプログラムの柔軟な取扱いができるようになった。その結果、汎用データ処理装置としてのコンピュータの位置づけが確立した。更に記憶装置の高集積化、大容量化、中央処理装置の高速化によるハードウェアの急速な進歩と相まって、現在のコンピュータによる高速で大量処理の普及と発展が成し遂げられた。

#### ② 制御方式

ノイマン型コンピュータの制御方式を説明する。手順をわかりやすく 説明する。

命令は記憶装置に直列に記憶され、次に実行する命令は、現在実行した命令の内容と、実行した結果によって変化した内部状態と、外部要因によって逐一決定され実行に移される(図1-1-3)。

図表1-1-3 逐次制御方式



この制御方式は多くのビデオの教材があるので視覚的に理解させる。 一方、より高速な処理要求から、非ノイマン型コンピュータの開発や 実用化の動向について言及してもよい。

### (4) 言語とプログラム

### ① プログラム言語

われわれが通常用いている自然言語のあい味性や冗長性を取り上げる。 それに対してデータ処理の領域では、問題解決の方法やデータ処理の手 順から、あい味さを排除したアルゴリズムとして記述し、このアルゴリズムをコンピュータに実行させるために、一定の形式を備えた言語を用いてプログラミングする必要があることを説明する。そのためにプログラム言語が必要であることを理解させる。

### ② プログラム言語の種類

言語は社会で最も活用している言語、入門者にとって理解しやすい言語、いま流行の言語など様々である。言語の種類をあげて説明するのではなく、最も活用されている言語を2~3あげ、その個性と特徴を説明する。できれば直観的なわかりやすさなどを比較しながら説明するのもよい。

### ③ 言語処理プログラム

コンピュータが実行できる命令は機械語命令であるため、プログラム 言語は何らかの方法で機械語に翻訳・変換しなければならない。そのため の自動翻訳システムとして、それぞれの言語ごとに言語処理プログラム とよばれるソフトウェアがあることを解説する。

### 1.2 ハードウェアとソフトウェア (240分)

- (1) ソフトウェアの必要性
  - ① 汎用データ処理システムの利用技術としてのソフトウェア
    JIS(日本工業規格: Japanese Industrial Standard)定義(JIS C 6230)
    ハーヴェア・・・・・データ処理において、用いられる物理的装着
    ソフトウェア・・・・・データ処理システムの運用に関係する計算機プログラム、
    手順、規則およびそれらに関連する文書

とある。この定義から、ハードウェアは固定的、汎用的、物理的な機器であり、これを個別の機能をもったデータ処理システムとして使用するためには、処理手順を指定したソフトウェアが必要であることを説明する。

例えば、実際のコンピュータを用いて、

- (a) 裸のコンピュータ
- (b) 裸のコンピュータ+オペレーティング・システム

- (c) 裸のコンピュータ+オペレーティング・システム+応用プログラム という状態を具体的に示し、コンピュータの動作状態の変化を体験させ るとよい。
- ② ハードウェアの利用技術としてのソフトウェア ハードウェアを制御し、利用するためのソフトウェアが、家電製品や 自動車などの身のまわりの機器に組み込まれ、使われていることを解説 する。つまり、ハードウェアを利用するためには、ソフトウェアの役割 が必要であることを理解させ、強調する。
- (2) ハードウェア
  - ① 命令実行系としてのハードウェアの役割
    - 一 演 習 一

教育用の1ボード・マイコン、あるいは仮想計算機のシミュレータななどを用いて、簡単な機械語命令群のステップ動作を実演し、命令の読出し、解釈、実行記憶内容の変化の流れを確認する。

② 周辺装置の仕組みと役割 -

代表的な入出力機器や補助記憶装置の仕組み、性能、適用領域などを 説明する。またこれらの装置が非常に簡単な命令(コマンド)になって 動作することを解説し、このコマンドを組み合わせて複雑な動作を実現 させるためには、ソフトウェアの存在が必要不可欠であることを認識さ せる。

- (3) ソフトウェア
- ① ソフトウェアとは

固定的で物理的なハードウェアと異なり、ソフトウェアは可塑的で論 理的な無形の知的生産物であり、プログラムと関連文書から成り立って いる。

プログラム……データ処理手順をプログラム言語で表現したもの。 関連文書……データ処理システムの計画、設計、試験、保守、利用お よび運用に関する文書。

この概念を理解させるために、実際のプログラム、マニュアル類など を示し、ソフトウェアという、いわば無形の存在に具体性をもたせて解 説する。

### 一演習一

簡単なプログラムを示し、それを実際に翻訳実行することによって、 命令文の集合体である静的なプログラムが、実行可能なプログラムに 変換され、実行され、動的なものに変化するプロセスを体験させる。

### ② ソフトウェアの分類

オペレーティング・システムはハードウェアをブラック・ボックス化し、応用プログラムに対して統一のとれた、使いやすいサービスを提供する。

応用プログラムは、オペレーティング・システムの提供するサービス を利用してデータ処理を行う。

ツール、パッケージ類を含むユーティリティ・プログラムについても触れる。

### ③ ソフトウェアの特質.

ソフトウェアの特質(ライフ・サイクル、品質と信頼性、生産性など) について説明し、ソフトウェア工学の概念について言及することを望み たい。

### 1.3 コンピュータの機能と特徴(180分)

### (1) 大量記憶

### ① 記憶装置の高集積化

記憶装置の容量拡大、コストの低下などは、ノイマン型コンピュータ を発展普及させた要因の一つである。大量記憶はコンピュータの主要な 特徴であることを理解させる。

実習に用いるコンピュータを例にとって、各種の補助記憶装置の性能 として記憶容量やアクセス時間を具体的に説明する。

### ② 大量記憶の有効性と用途

ファイルおよびファイル処理について説明し、補助記憶装置の大量記 憶能力をファイルとして利用することを解説する。ファイルの内容は数 字や文字だけではなく、音声や画像情報まで拡大し、マルチメディア化 の傾向にあり、その用途も拡大していることを理解させる。

#### (2) 高速処理

① 処理装置の高速化

実習に用いるコンピュータやワークステーションなどの平均命令実行 回数(MIPS値)、浮動小数点演算実行回数(FLOPS値)などの CPUの性能を具体的に説明する。

② 高速処理の有効性と用途 処理速度の向上は、精度の高いシミュレータ、リアルタイム処理など、 コンピュータの適用領域を拡大することができる。

(3) マン・マシン・インタフェース

マン・マシン・インタフェースは、コンピュータとそれを操作する人間 との仲介をつかさどり、コンピュータの操作性および使いやすさに関わる 機能である。近年、コンピュータのパーソナルな利用の拡大によって、マン・マシン・インタフェースの役割の重要性が増大していることを認識させる。

① マン・マシン・インタフェースの必要性

従来、コンピュータは専門的知識をもった限られた範囲の人々によって取り扱われる傾向にあったが、高度情報化社会へ進むにしたがって、コンピュータに特別な知識をもたない人々も、コンピュータを利用する機会が多くなってきた。

データ処理という観点から見ても、今までの数値処理中心から、最近 は画像処理、知識処理などの非数値処理の要求も多くなっている。

こうした状況の中で、コンピュータにはイメージ処理を多く取り入れ、 操作性のよい、一般に使いやすいマン・マシン・インタフェースが要求 されている。

具体的には次のような実用化が進んでいる。

- (a)専門知識を必要としないコマンド・インタフェースへ改善する。
- (b)わかりやすい、扱いやすい入出力機器の開発とその操作環境を確立 する。
- (c)装置の操作方法の簡素化や標準化を図る。

(d)操作ミス、誤入力に対して柔軟に対応する。

② マン・マシン・インタフェースを向上する入出力機器と操作環境 入力機器、出力機器、操作環境などについて、実物や映像(A V 教材) などを用いて、使い方、用途、効果を具体的に説明する。

#### 一実習一

ワークステーションを中心にした、マルチ・ウインドウ+アイコン +マウスによる最近のマン・マシン・インタフェースを体験させる。

#### (4) 通信との融合

### ① データ通信の有効性と用途

コンピュータと通信の融合によって、遠隔処理、分散処理などが図れ、 コンピュータの適応領域や機能が飛躍的に拡大することを解説し、デー タ通信システムの基本的構成やその基礎技術とデータ通信の重要性につ いて解説する。またデータ通信の役割についても理解を深める。

産業分野、行政分野、社会生活分野などのデータ通信システムの中から代表的なシステムを取り上げ、具体的な形態や使われ方を解説し、データ通信システムが日常の生活に浸透していることを認識させる。

### ② データ通信の将来

高度情報化社会の基礎技術として重要な役割を担い、普及発展しているデータ通信システムの動向と将来性について、国、社会、産業、企業、団体、個人のレベルから、実例を取り上げて展望してみる。

### 育成方法および指導上の留意点

解説は細部には立ち入らず、ハードウェア、ソフトウェアの詳細は、後述の 各論に譲ることとする。本章は飽くまでコンピュータの性格、役割、特徴の基本を理解し、以降の学習の中でのコンピュータの位置づけを明確にすることを 目的とする。

したがって、実習、実演、演習などは、市販されているビデオ教材を多用し、 視覚的構成で指導することが望ましい。

### 用語

制御装置、演算装置、記憶装置、入力装置、出力装置、中央処理装置、補助記憶装置、周辺装置、大量装置、高速処理、データ処理、応用プログラム、ハードウェア、ソフトウェア、オペレーティング・システム、ビット、バイト、ジョブ、プログラム記憶方式。ノイマン型コンピュータ、処理手順、制御方式、非ノイマン型コンピュータ、自然言語、あい昧性、冗長性、アルゴリズム、プログラム言語、機械語、翻訳変換、自動翻訳システム、言語処理プログラム、JISコード、1ボード・マイコン、仮想計算機。コマンド、プログラム、関連文書、マニアル、ブラック・ボックス、パッケージ、ユーティリティ・プログラム、アクセス時間、ファイル、マルチメディア、ワークステーション、CPU、シミュレーション、リアルタイム処理、マン・マシン・インタフェース、画像処理、知識処理、ユーザ・フレンドリ、マルチ・ウインドウ、アイコン、マウス

### 第2章 コンピュータによる情報処理。

### 学習目標

学習者が、この章を修了した時点で、次の事項ができるようになることを目標とする。

- 1. パッケージ・ソフトを使う実習により、「入力一処理一出力」といった一連の動作を感覚として会得している。
- 2. JISキーボード配列による操作に馴れる、マウスの操作に馴れるなど、 利用ソフトをコンピュータで実現するための必要な要素が自然に身について いる。
- 3. コンピュータを道具として気軽に使え、「コンピュータによる情報処理とは何か」、その目的意識や動機づけが整然と確保される。

### 内容

### 2.1 パッケージ・ソフトの利用(540分)

様々な分野で活用されているパッケージ・ソフトを使って、情報がどのよう に処理されるかを理解することは、学習者の段階的育成にとって、きわめて有 効である。

センスのよい完成されたパッケージ・ソフトを操作させることは、コンピュータの基本的な動作が感覚的に会得でき、コンピュータによる情報処理を理解させるための契機となる。したがって、多くの分野で使われているパッケージ・ソフトの中から、ゲーム、ワープロ、表計算などを選別し、段階を追って使わせ、コンピュータの正しい知識を理解させる。

### (1) ゲーム・ソフトによる体験

ゲーム・ソフトは、ゲーム・センタ、ホビー用ゲーム機、パソコンなどで広く使われている。入力は専用のスイッチやレバーを用いたり、キーボードから指示したりする。体感ゲームを除いては、出力は画面がほとんどであり、音楽や効果音を付加し、飽きがこないようにゲーム進行させてい

る。

ここで取り上げるゲームは、パソコンや端末を通じてのゲームである。 キーボード操作やマウスの操作に馴れる必要があり、それを考慮したゲームを選別して、JISキーボード配列の基本操作系を習熟させる。

- ゲームを楽しむ。
- ② "入力一処理一出力"の感覚を養う。
- ③ 気軽に使える道具として意識する。
- ④ 入力のタイミングと出力の種類を考える。

日常のレベルの思考であり、ソフトを分析することによって、コンピュータ上で実現するために必要な要素や、その要素の構成などを身につけさせる。

(2) ワープロ、表計算ソフトの実習

コンピュータによる情報処理の典型的な例として取り上げる。それぞれ については基本的な使い方の実習に留める。

ワープロはフロント・プロセッサ部分の扱いが、ソフトによって異なるから、いくつかの種類について実習させる。基本的な編集機能をひととおり行い、簡単な文章を出力した後に、どのような処理で構成されていたかを考えさせる。

### 基本的なワープロ操作実習

ソフトの起動、終了、文字入力、訂正、挿入、削除、下線引き、罫 線枠取り、作成文書の保存と取出し、印字

コンピュータ処理について細部に立ち入らず、ゲーム・ソフトの体験の 延長上にあって、キー操作が複雑になることを理解させる。

- ① ワープロ・ソフトが作成される理由を考えさせる。
- ② ワープロのキー操作が複雑になる理由を考えさせる。
- ③ 他のパッケージ・ソフトと比較して、コンピュータ・ソフトの存在を 考えさせる。
- ④ コンピュータによる情報処理は、必要なソフトがあり、いろいろなバリエーションをもつであろうことを理解させる。

表計算ソフトの実習では、典型的なクロス集計を行い、効果的なコンピ

ュータの使い方を理解する。表計算ソフトは、数式を入力しておくことによって、計算したいデータに変更があれば、自動的に再計算してくれる。

#### 基本的な表計算操作実習

表計算ソフトの起動、終了、データ入力、訂正、数式入力、訂正、 作成した表の保存、印字、ソート

手作業で行うクロス集計の煩雑さと、表計算ソフトを使ってのクロス集計の作成を比較して、正確で俊敏なソフトがいかにすばらしいか、その利点を認識させる。

### (3) 簡易言語の実習

初期段階の学習では、自分でプログラムを作成する力は養成されていない。そこで簡易言語によるパッケージ・ソフトを使って、ある程度の処理を組み立てて実行させる。

簡易言語はディスプレイ表示に基づき、必要なデータを入力する。処理 を選択するか、または簡単な指示を与えることにより、処理が実現できる ようになっている。大まかな処理が組み立てられ、成功体験させるとよい。

- データを用意する。
- ② 並べ替えて表示する。
- ③ データの合計、平均を求める。
- ④ データの中の最大値、最小値を求める。

簡易言語であれば、以上の処理は一つの指示か数行の命令で行える。簡易言語を通じて、コンピュータによるプログラミングのイメージを把握し、プログラム記憶方式とはどのような方式か、少しでも理解を深める。

### (4) 新しい言語(第4世代言語)の実習

データベース検索、グラフ・表計算統合ソフト、プレゼンテーション用 ソフトなど、第4世代言語と呼ばれるものがあり、これらも簡易言語と同 様に、複雑なプログラムを作成することなく、会話方式によって簡単な指 示をコンピュータに与えるだけで、処理が実現できる。

2~3のソフトを実習させ、便利に使えるだけではなく、便利に使わせるためにはどうすればよいか思考させる。

### 育成方法および指導上の留意点

学習者が理解しやすく、操作が簡単なパッケージ・ソフトを使うことがポイントである。

指導者が慣れ親しんでいるソフトが、学習者にとって本当によいソフトか、 また、育成にあたって効果があるかなど、注意深く実習ソフトを選定しなけれ ばならない。

基本的な機能を知るための操作実習であり、ソフトが維持している機能を全部操作させる必要はない。代表的な機能を十分に学習させるためには、目的に合ったソフトを選定し、具えている機能をどのような順序で学習させるか、よく判断して使わせなければならない。(1)~(4)まで、段階的にコンピュータの機能が理解されるように配慮する。

広範なコンピュータによる情報処理を理解するための足がかりとなるように 動機づけ、結果として、学習者が興味をもち、以降の学習が意欲的に行われる ように指導し、ゲームだけに没頭しないよう注意を払う。

### 〔ゲーム・ソフト実習〕

キー入力操作や表示は簡単で明決なソフトを選び、ある程度の機能分析を 行う。

- (a) シューティング・ゲーム
- (b) 思考型ゲーム
- (c) ロールプレイング・ゲーム

### (ワープロ実習)

基本機能を習得させ、ワープロ実習によって、コンピュータの便利さを意 識させる。

- (a) 簡単な文章を作成する。
- (b) 保存文章を修正する。
- (c) 読む、見るなど、わかりやすい構成や編集の作業を行う。
- (d) ワープロの効果的な使い方を知る。

### 〔表計算ソフト実習〕

(a) 数式が与えられることにより、自動的に計算される仕組みについて、注

意させる。

(b) 計算速度や処理速度に注意を払う。

### (簡易言語実習)

- (a) データとデータ処理の相違を理解させる。
- (b) コンピュータが学習者の指示(命令)で作動する様子を理解させる。
- (c) プログラム記憶方式の概念を理解する。

### 〔新しい言語の実習〕

- (a) 統合化ソフトを使用させる。
- (b) マウスを使用し、マルチ・ウインドウ、マルチ・タスクなどを経験させる。
- (c) 専門的用語がたくさんあることを認識させる。

### 用語

パッケージ・ソフト, ワード・プロセッサ (ワープロ), 表計算ソフト, マウス, JISキーボード, ディスプレイ, データベース, 表計算統合ソフト, 第4世代言語, 簡易言語, ゲーム・ソフト, マルチ・ウインドウ, マルチ・タスク

### 第3章 コンピュータの効用

### 学習目標

学習者が、この章を修了した時点で、次の事項が理解でき、高度情報化社会 に深くかかわっている意識が備わることを目標とする。

- 1. 現代社会で多用されている基本的な処理である大量一括処理、即時処理、 オンライン処理などの概要が理解されるとともに、それぞれの処理について 限界があることを認識する。
- 2. 社会生活の中で、多くの便益が享受されているコンピュータ処理システム を正しく認識し、また、コンピュータが社会において大変役に立っていることを具体的に記述できる。
- 3. コンピュータと通信が融合した高度情報化社会がどのような社会か、具体的に説明できる。

### 内 容

#### 3.1 一括処理(60分)

- (1) 大量データの高速一括処理
  - ① 一括処理

コンピュータの特質である大量記憶、高速処理、正確性、安全性など に着目し、一括処理の利点や大量データを取り扱う場合の弊害などにつ いて解説する。

一括処理は、日、週、月など、仕事を基準にしてデータを収集し、処理の単位を形成する場合と、コンピュータの特徴である高速性を利用し、その処理速度を基準にして処理の単位を形成する場合があり、それぞれについて具体例をあげて説明する。

また、人間の判断や思考力が伴う処理と、記録、計算、分類、集計な ど、思考力がそれほど伴わない単純な処理を明確に区別し、適切で経済 的な処理が必要であることを強調する。

### ② 処理形態の広がり

通信技術の発展によって、従来の手作業によるデータ収集から、通信 回線を利用したデータ収集へと発展している現状を把握する。その結果、 オンライン処理、リアルタイム処理、TSS処理(Time Sharing System) などが広がったことや、また、分散処理も広まった背景について概説す る。

### (2) コンピュータ普及のきっかけ

現代社会において、これだけコンピュータが活用されるようになったのは、限りないハードウェアの進歩と、このハードウェアを合理的に使う利用技術、つまりソフトウェアの開発であった。そのきっかけはコンピュータ利用技術の発展をさかのぼることによって解明される。

ホレリスの統計機械からP. C. S. (Punch Card System) へ、P. C. S. からカードを入力媒体とするコンピュータの一括処理へ、通信回線を利用した即時処理から、オンライン・リアルタイム処理へと発展した。この目ざましい発展の背景には、

- (a) 大量記憶(安全で大量なデータの保存)
- (b) 高速処理(安定とサービスの向上).
- (c) 高性能化(信頼性の向上)
- (d) 小型化(取扱いの便利さ)
- (e) 価格の低廉 (需要の拡大)

などハードウェア技術の革新があり、更に利用を促進させたのは、ソフトウェア、特にオペレーティング・システムの発展と使いやすさを向上させたマンマシン・インタフェースの開発であった。そしてデータ通信の近代化がコンピュータの利用を更に身近なものにした。

利用するコンピュータは、汎用コンピュータからマイクロ・コンピュータまであり、中でも安価で高性能なパーソナル・コンピュータの需要の伸びに注目させ、コンピュータの普及のきっかけから今後の動向について解説するとよい。

また、一般的なコンピュータの呼称にも触れ、若干の機能や使い方を簡単に説明する。

### 3.2 オンライン処理の躍進(180分)

旅行する、出張する、天気予報を聞く、演劇などを観賞する、幹線道路を車で走る、納税する、買物でカードを使う、郵便局や銀行でお金を扱う、図書館で本をさがす、バスに乗る、証明書を発行してもらう……。

こうした日常の何気ない生活行動は、すべてコンピュータのオンライン処理 に関わっている。まさに高度情報化社会のまっただ中である。いかに、私達が コンピュータによる便益を享受されているか、いかに、オンライン処理と深く 関わっているかについて解説する。また、この処理の躍進が大きく社会を変え ていることや、大変役に立っていることについて説明する。

### (1) 預貯金システム

金融機関によるコンピュータの利用は、常に最先端の利用形態をとってきた。

1959年から始まったオフラインによる業務処理、1970年初頭の第1次オンライン・システムによるサービスの開始、1970年代後半から1980年代にかけての第2次オンライン・ネットワーク時代、そして、第3次、現在の巨大化したオンライン・システムの近代化へと進んでいる。

オフラインの時代は、大量業務の機械化と人員削減に主眼がおかれた。 第1次オンラインではタイム・シェアリング方式を前提に、リアルタイム 処理による業務の迅速化、顧客の待ち時間の短縮化が図られ、利用者サー ビスの向上をねらいとした。

第2次オンラインでは、いろいろなサービスを提供するために、全銀システムによる全国規模のシステムとして、地方銀行へとネットワークを構築していった。

第3次オンラインでは、業務の広域化、国際化が図られ、システムの大 規模巨大化が進んでいる。年間を通じてノンストップ運転と通信システム の充実が必須条件であり、デュアル・システム、複数センタの設置などで 信頼性の向上に努めている。

郵便貯金にも触れ、年金や保険、その他の業務についても解説する。また、異業種間のネットワーク・システムの概念についても触れる。

#### (2) 予約システム

① JRの「みどりの窓口」

JR旅客鉄道会社の「みどりの窓口」は、全国でおよそ 1,600カ所、 2,600台以上の端末が接続されており、更に各旅行会社のオンライン・ネットワークが接続されていて、座席予約をはじめとする様々なサービスを提供している。

マルス (MARS: Multi Access Reservation System)と名づけられたこのシステムは、1960年に初のオンライン・リアルタイム・システムとして開発され、利用者のニーズに合わせ、様々な機能を追加し、指定券発売(マルス105)、団体予約(マルス202)、電話予約(マルス150)の3システムにより運用されている。

一日に 100万座席の予約ができ、全国の端末に対し6秒という高速レスポンスが得られていることや、2億座席のデータ量と、年間3億件の問合わせがあることなどを誇っている。

座席予約システムは高信頼性、JR独自の専用ネットワーク(JR-NET)を保持している、高速パケット交換網の付加価値通信網(VAN: Value Added Network)であるなどを概説する。

② 旅行会社の総合予約システム

旅行会社は宿泊予約をはじめ、他の交通機関に対する座席予約などの 手配が主な業務である。

季節や時間によって処理量の増減が激しく、人手による作業では対応 しきれないため、オンラインの会話式予約システムとして構築された。

このシステムは、JRや航空会社などの予約システムとの接続が不可欠であり、接続されてはじめてその価値が生きる。

顧客登録、予約発券、代金精算など、このシステムの特徴や限界について簡単に説明する。

③ チケット予約販売システム

各プレイガイドでは、劇場の予約状況を知ることができ、会話方式に よって顧客が希望する席を予約できる。

プレイガイドに設置してある端末はセンタとオンラインで直結されて

いて、予約、発券、支払い、精算などの処理が即時に行われる。提携の 信販会社を経由し、決済することもできる。

#### (3) 検索システム

高度情報化社会ではたくさんの情報があり、必要な情報をタイムりに提供されることによって生産性が高まる。情報を貯え、利用者に情報を提供しようとするシステムを検索システムといい、その規模も種類も、非常にたくさんある。一定の約束にしたがって構築した情報の集合体を「データベース」と呼んでいる。

このデータベースの利用形態はオンライン即時処理が多く、自分の端末 から通信回線を経由してアクセスすることができる。

検索できる内容は、個人の趣味から経済、産業、社会、医学、文化、教養、文献など、非常に多くの種類のデータベース(内容)があり、情報化社会の象徴的なサービス支援である。

データベースと検索システムの概要を整理して説明する。

#### (4) 交通管制システム

安全で快適な交通を確保するためには、道路環境の整備が必要であり、 道路構造、交通信号、交通規制などが深く関わっている。

このシステムは、交通の円滑化を図るために交通量に即応した信号制御を行ったり、地上約5メートルに設置した車両感知器によって自動車の有無を判定し、交通量を把握して即時処理を行い、総合的に交通状況をタイムリに情報板に表示したり、ラジオ放送を通じてドライバに知らせ、高速道路や幹線道路の一層の安全と円滑化を図っている。

このシステムは刻々と変化する状況を把握し分析して、その結果をフィード・バックするシステムである。変化に対応する即時性や公共性を訴え、 管制システムの特徴を概説する。

### (5) 気象情報システム

私達の生活の中に溶け込んでいる気象情報は、四季を通じ経済社会に極めて大きく影響している。変化の多い大自然の情報だけに、大勢の人々に使われ、その利用価値も高い。

このシステムは、気象衛星で観測したデータ、アメダス(AMEDAS: Autom-

ated Meteological Data Acquisition System)からの情報、各地気象台からのデータなどを収集し、計算し、分析し、画像処理して、予報のサービスを提供するものである。しかも刻々と変化する状況を24時間観測し実時刻でスケジュール管理している。

予報精度を高めるために、収集した観測データをスーパー・コンピュータによって分析しているが、それでも日本全国を50km格子状に区切って計算しているにすぎない。

高性能な、計算速度の速いスーパー・コンピュータを使っていることや、 実時刻スケジュール管理など、このシステムの特徴や、予報のむずかしさ について触れ、気象情報の重要性を説明する。

### 3.3 高度情報化社会(180分)

コンピュータと通信が融合した社会では、見えない速さでデータが処理され、 結果としてたくさんのサービスが提供される。情報とサービスは不可分にあり、 利用者の賢い選択が役に立つ時代、それが高度情報化社会であろう。

個人が利用しているカード、パーソナルなコミュニケーションのパソコン通信、小規模小売業が利用する小売店端末などについて触れる。

### (1) カード化社会

普及しているカードには、キャッシュ・カード、クレジット・カード、 プリペイド・カードなど、様々な用途のカードがある。これらのカードの 共通点は情報あるいは現金を、安全に、かつ簡単に持ち運べることから普及したものである。

日本では一人平均5枚のカードを持っているといわれ、これがカード化 社会と呼ばれるゆえんである。

今後はそれぞれの機能を1枚のカードにまとめようと、ICカードや光 カードが出回るものと思われる。

#### 磁気カード

クレジット・カード(後払い式)、デビット・カード(即時払い式)、 プリペイド・カード(前払い式)の3種類がある。

クレジット・カードは銀行系、信販系で、国内国外を問わず利用でき

る。デビット・カードは銀行などでCD (Cash Dispenser)や、ATM (Automated Teller Machine)を使って、現金の出入れに使っている。 プリペイド・カードは通貨の代用として使っているが……

それぞれの利点を説明し、法との関連についても触れることが望ましい。

### ② ICカード, 光カード

I Cカードは、多用されているカード・サイズに I C (Integrated Circuit)を内蔵させたものと、CPU(Central Processing Unit)とメモリ機能を併用したものがある。

光カードは、カードそのものに大記憶容量機能を具えたものである。 図表3-3-1 カード仕様比較表

	磁気カード	ICカード	光カード
記憶容量	72 バイト (プリペイド) 216 バイト	8Kバイト ~64Kバイト	1 M~4 Mバイト
処理機能	なし	あり	なし
価 格	100 円	10,000円 ~15,000円	500 円
標 準 化	銀行系	銀行系	なし
用 途	身分証明 入退室管理 出退動管理	銀行預金記録 医療情報記録 電子手帳 電子出版	銀行預金記録 医療情報記録 データ運搬 電子出版 画像データ記録

### (2) パソコン通信

パソコン通信は、パーソナル・コンピュータ(パソコン)やワード・プロセッサ(ワープロ)を端末として、提供されているネットワークを介在し、いろいろなデータベースから情報を入手したり、仕事や趣味で収集した情報を、メールボックス(ファイル)を通じて変換する個人の利用形態

#### である。

パソコン通信を行う場合には、個人、グループ、クラブ会員制などがあり、国内はもとより国外に至るまで、データベースを利用したり、メール 交換をしたり、プライベートの利用に大きな特色がある。

パソコン通信の実現にはパソコン通信ネットワーク・システムが必要であり、利用形態も様々ある。実現のための機器構成、通信ソフト、費用などを含めて紹介する。

図表3-3-2 パソコン通信ネットワーク

### <国 内>

名 称	内容
ASCII-NET	<b>趣味・</b> スポーツ・コンピュータ <b>関連情報、</b> ニュース・電子 メール
EYE-NET	電子メール、電子掲示板、ニュース
JALNET	フライト・インフォメーション、ショッピング、ツアー情 報、電子メール、天気予報
NTT-PC NETWORK	ニュース、天気予報、ショッピング、催し物案内、その他 各種情報
日経 TELECOM	経営・投資・景気・金融為替情報、株情報
PC-VAN	新聞情報、株情報、旅行・趣味情報
TELESTAR	ニュース、株情報、中古車情報、ショッピング

### <国 外>

CompuServe	A Pニュース、各新聞紙、仕事・趣味での情報交換
DELPHI	6か国語翻訳サービス、百科事典
TheSource	UPI・APニュース、ビジネス・金融関連情報
DIALOG	文献情報、米国の新製品、企業情報、各分野のテータヘース
DowJones	ニュースの速報と検索、株式情報、百科事典、ショッピング

### (3) 小売店端末 (POS)

別名、販売時点情報管理システムと呼んでいるこの方式は、従来はキー イン方式のレジスタを使っていたが、現在は自動読取り方式のレジスタに 変わって、金額以外の情報も、同時に収集することができるようになった。 小売店で販売する商品にはバー・コードがついていて、これをレジスタ のペン・リーダ、レーザ・スキャナ、タッチ・スキャナといった読取り装 置で入力する。

日本で使用しているバー・コードはJAN (Japanese Article Number) コードと呼び、欧米と互換性のある国際的なコードである。このコード入力の特徴は、キーイン方式に比較して読取りが速く、正確で、機器コストも安い。

キーイン入力のレジスタでは金額情報だけしか把握できなかったが、現在の小売店端末ではバイプロダクト方式といい、商品名、価格、販売日、時刻、レジナンバなどが入力できる。このデータはストア・コントローラと呼ばれるパソコンやオフコンで蓄積され、売上管理や在庫管理などに使われる。

JANコードに登録されているメーカは約4万社、コード化されている 商品は約200万品目、設置されている小売店端末は約12万台と推定されて いる。

ますます需要が高まるこのシステムは、流通と商業特に小売業、とりわけスーパー・マーケット、コンビニエンス・ストア、ディスカウント・ストア、ボランタリやフランチャイズに活用され、商品開発、品揃え、売れる商品、客の動向など、きめ細かい分析ができることに特徴がある。

流通の仕組みなども紹介するとよい。

### 育成方法および指導上の留意点

それぞれの処理システムについて、様々な分野から実用例や応用例を取り上 げ解説する。その際、コンピュータの発展段階を順を追って説明し、深く技術 面に立ち入るのではなく、その概略が得られればよい。

オンライン処理では、私達の生活に関わっているシステムを取り上げている が、システムそのものを理解させるのではなく、それぞれの処理が生活と密着 した、社会に溶け込んでいる様子を強調する。 高度情報化社会の象徴的な通信、処理方式、カードなどを取り上げたが、いずれも概説に留める。

コンピュータを道具とする多くの処理システムは、便利で社会生活に豊かさをもたらせているが、それらにはおのずから限界があり、むしろこの限界を正しく理解させることに注意を払わなければならない。

この章を指導するにあたっては、教育用ビデオ教材があるので、適切なもの を選び、教育効果を高めるとよい。

### 用語

オンライン処理、T.S.S.処理、分散処理、P.C.S.、一括処理、オンライン・リアルタイム処理、データ通信、汎用コンピュータ、マイクロ・コンピュータ、パーソナル・コンピュータ、オフライン、ネットワーク、デュアル・システム、レスポンス、座席予約システム、パケット交換網、付加価値通信網、検索システム、データベース、通信回線、アクセス、交通管制システム、情報板、フィード・バック、気象情報システム、気象衛星、即時性、アメダス、画像処理、スーパー・コンピュータ、パソコン通信、小売店端末、キャッシュ・カード、クレジット・カード、プリペイド・カード、カード化社会、ICカード、光カード、磁気カード、デビット・カード、CD、ATM、パソコン通信、メール・ボックス、通信ソフト、キーイン方式。レジスタ、バー・コード、ペン・リーダ、レーザ・スキャナ、タッチ・スキャナ、JAN

### 推薦教材

(1) 「情報処理システム入門」

浦 昭二・市川照久 共編, サイエンス社, 1989年

(2) 「情報通信概論」(経営情報講座)

定道 宏 著, オーム社, 1985年

### -AV教材-

#### •第1章関連

- (1) コンピュータ入門(日本ディジタルイクイップメント株式会社)
- (2) コンピュータの基礎 (トッパン・ムーア・ラーニング株式会社)
- (3) コンピュータシステム入門(日刊工業新聞社)
- (4) 情報処理システム入門(日刊工業新聞社)
- (5) 社員教育用または貸出用がある(メーカー各社提供)。
- (6) 情報処理技術者試験講座(種類が多い)

### ・第2章関連

- (1) ワープロタイピング (オーム社)
- (2) 入門一太郎Ver.4 (日本ソフトバンク)
- (3) 表計算基礎編および実践編(CBSソニーグループ)
- (4) ビデオでわかるLotus1-2-3入門編(CBSソニーグループ)

### ーソフトウェアー

### ・第2章関連

- (1) ワープロ:一太郎(ジャストシステム)
- (2) 表計算:マルチプラン(マイクロソフト)
- (3) 統合型:ロータス1-2-3 (ロータス)

### インストラクタ必読書

(1) 「情報工学」(放送大学教材)

都倉 信樹 著,日本放送出版協会,1987年

(2) 「コンピュータダイジェスト」-月刊-(株)テー・エー・シー企画出版部

(3) 「パッケージソフト」-季刊- (株)リクルート

- (4) 「日経コンピュータ」, 「日経バイト」, 「日経パソコン」 日経BP社
- (5) 「新・情報処理システム」牧野 勝 著,日刊工業新聞社,1988年
- (6) 「I Cカード」(ニューメディア技術シリーズ) 電子情報通信学会 編, オーム社, 1988年
- (7) 「バーコードでここまでできる」 日本経済新聞社 流通システム開発センター 編

### 参考文献

- ・第1章関連
  - (1) 「計算機の機能と構造」(岩波講座情報科学14) 矢島 脩三 著、岩波書店、1989年
  - (2) 「計算機アーキテクチャ」(岩波講座情報科学15) 相磯秀夫・飯塚肇・元岡達・田中英彦 共著,岩波書店,1982年
  - (3) 「情報工学」(放送大学教材) 都倉 信樹 著,日本放送出版協会,1987年
  - (4) 「計算機システム入門」(岩波講座ソフトウェア科学1) 所 真理雄 著. 岩波書店、1988年
- ・第2章関連
  - (1) 「コンピュータ白書」

### 日本情報処理開発協会 編, 1987年版

- (2) 「月刊アスキー」(株)アスキー
- (3) 各種操作解説書…数多くあるので必要なものを選ぶ
- ・第3章関連
  - (1) 「情報化白書」

日本情報処理開発協会 編, コンピュータ・エージ社, 1987年

- (2) 「ニューメディア白書」 日本情報通信振興協会 編, 日刊工業新聞社, 1988年
- (3) 「オンライン・データベース・ディレクトリ89」 東洋経済社 日本情報通信振興協会 監修
- (4) 「コンピュータと地方自治」地方自治情報センター
- (5) 「パソコン通信」 栗原 章浩 著, オーム社, 1987年
- (6) 「POSシステム流通業の情報化戦略」 流通システム開発センター 編、日本経済新聞社
- (7) 「パソコン通信の常識読本」安田 幸弘 著、日本実業出版社、1987年

### 第2部 ハードウェア

### 教育目標

ハードウェア全般にわたっての基礎知識を修得させることを目標とする。ただし、データ通信に関する事項は第4部で述べるものとする。

高級言語が普及し、一方ではコンピュータ・システムが複雑化した今日、利用者はもとより専門職のプログラマとしても、ハードウェアに関する知識は必ずしも必要でないとも考えられる。しかしコンピュータをより有効に活用するためには、また中級・上級技術者を目指すのであれば、コンピュータを単なるブラック・ボックスとして取り扱うのではなく、ハードウェアについてもある程度の基礎知識を持つことが要求される。

さらに、今日では高性能なパーソナル・コンピュータやワークステーションによるシステムの分散化、これに関連した周辺機器の普及がハードウェアをより身近なもにしている。このような新しい形のシステムの構築と運用に従事する技術者にとって、個々の装置の持つ機能、性能、あるいは装置相互の接続技法を理解するためにハードウェアの基礎知識は不可欠なものとなろう。

# 内容の構成

rit	時 間 数		
内容	講義	演 習	
第1章 コンピュータの基本構造 1.1 コンピュータの特徴 1.2 コンピュータの構成 1.3 コンピュータの種類と構成例	3	0	
第2章 データ表現 2.1 2進法 2.2 文字の表現 2.3 バイトの概念 2.4 数値の表現	6	2	
第3章 記憶装置 3.1 記憶装置の種類 3.2 記憶の原理 3.3 記憶装置の仕組み	6	0	
第4章 演算・制御装置 4.1 コンピュータの命令 4.2 中央処理装置 4.3 プログラム実行の仕組み 4.4 演算の仕組みと論理回路 4.5 半導体素子と集積回路	1 0	2	
第5章 入出力装置 8 5.1 入出力機器の種類 5.2 入出力制御 5.3 入出力インタフェース	8	0	
第6章 コンピュータの歴史と将来 6.1 ハードウェアの歴史 6.2 コンピュータの現状と将来	3	0	
合 計	3 6	4	

### 第1章 コンピュータの基本構造

### 学習目標

ハードウェアの序論として後続の章が展望できるようになり、さらにつぎの ことができるようになる。

- 1. コンピュータとそうでないもの(電卓、そろばん、ファミコン、脳など) との類似点、相違点がわかるようになる。
- 2. コンピュータ・システムの具体的な構成図が見れるようになる。

### 内 容

#### 1.1 コンピュータの特徴(40分)

ウイナーおよびノイマンの提言に言及しながら、ハードウェアの機能的側面 からみたコンピュータの特徴がつぎの4項に集約されること、これらが互いに 不可分の要件であることを理解させる。

- (1) 高速演算
- (2) 大量記憶
- (3) 2 進ディジタル方式
- (4) プログラム内蔵方式

#### 1.2 コンピュータの構成(60分)

つぎの項目の順序で、コンピュータの大まかな構成について述べる。

- (1) コンピュータの 5 大機能 人間の知的活動と対比させながら、コンピュータが記憶、演算、制御、 入力、出力の 5 大機能をもつことを述べる。
- (2) 各機能の具体化

各機能が記憶装置、演算装置、制御装置、入力装置、出力装置として実現され、これらは電子部品および機構部品から構成されることを述べる。 また主記憶装置、補助記憶装置、中央処理装置、本体、周辺装置という用 語にも言及し、フレキシブル・ディスク装置など具体的な装置の例を示し て解説する。

(3) データと制御の流れ

各装置が制御信号によりシステムとして統合される様子とデータの流れ を理解させる。データの入出力に関して、チャネル結合とバス結合がある ことに触れる。

### 1.3 コンピュータの種類と構成例(80分)

前節を受けて、代表的な機種の構成例を図示する。とくに入出力制御に着目すると、つぎの(1)、(2)、(3) の各例を提示するのが望ましい。同時に、ハードウェアの側面からみたコンピュータの種類についても言及する。スーパー・コンピュータ、ミニ・コンピュータ、制御用のコンピュータ等は(4)で述べる。

- (1) 汎用コンピュータ
- (2) パーソナル・コンピュータ
- (3) ワークステーション
- (4) 専用コンピュータ

### 育成方法および指導上の留意点

- 1. 1.1では、今日のコンピュータのもつ特徴がいかにして生れてきたかについて、思想的な側面(ウイナーやノイマンの提言)と技術的な側面(電子部品の驚異的な進展)を簡単に述べる。
- 2. 1.3で示すコンピュータの具体的な構成図は、実習等で使用するものが親しみやすくてよい。
- 3. 具体的な構成図には、各装置の性能を示す値(MIPS、バイト、行/分など) がある方がよい。これらは、本章で簡単に説明しておくとよい。

## 用語

記憶装置、演算装置、制御装置、入力装置、出力装置、主記憶装置、補助記憶装置、中央処理装置(CPU)、周辺装置、処理速度(MIPS)、記憶容量(バイト)、汎用コンピュータ、パソコン、ワークステーション、チャネル結合、バス結合

## 第2章 データ表現

## 学習目標

- 1. 2 進数、8 進数、16進数の取扱いが出来るようになる。これらの基数と 10進数との相互変換が出来るようになる。
- 2. コード表を見て、文字や記号とコードの相互変換ができるようになる。
- 3. 文字や数値の具体的な内部表現を見てその意味が分るようになる。

# 内 容

### 2.1 2 進法(120分)

コンピュータ内部で2進法が採用される理由、2進数の取扱い方について、 次のの要領で解説する。

## (1) 数の表現方法

数値の表現方法にアナログとディジタルがあり、それぞれの短所と長所がある。とくにディジタルの場合、2進法を用いると技術的に有利な点が多いため、コンピュータのハードウェアにおいては2進法が基本となっていることを理解させる。

- (2) 2 進法
  - 2進法による数の表現、加減乗除のやり方を理解させる。
- (3) 8進数と16進数
  - 2 進数の延長上の表現法として、われわれにとって取扱いやすい 8 進数、 16進数を取上げる。 3 者の関係を明らかにする。

# (4) 基数変換

われわれが日常使い慣れている10進数から、コンピュータの内部表現である2進数、8進数、16進数への変換、およびその逆の変換の方法を習熟させる。

### 2.2 文字の表現(60分)

英字や記号、仮名や漢字が何桁かの2進数によって表現できることを解説し、 コードの重要性を理解させる。

#### (1) コード

n桁の2進数(nビットの2進数)によって 2n 種類のコード(符号)がつくられ、これと1対1の対応をつけることによって、2n 種類の文字(英字、仮名文字、漢字、0から9までの10進数字)、種々の記号を表現できることを理解させる。

#### (2) BCDコード

コードの例として 0 から 9 までの 1 0 進数字を表示する B C D コードを 説明する。誤り検出符号、誤り訂正符号についても言及する。

#### (3) コードの標準化

コードの互換性のために規格化が必要であることを認識させる。標準コード体系としてISOコード、JISコード、ASCIIコードが定められていること、EBCDICコードも広く用いられていることを説明する。 改行やエスケープのように表示されない機能コードについても言及する。

#### (4) 日本語コード

わが国独特のカナ文字と漢字を表現するコードについて解説する。とくに漢字コードには、JIS漢字コードとKSI(Knji Shift In)、KSO(Kanji Shift Out)を併用するものと、これを変形したもの(シフトJISコード等)があることを説明する。

#### 2.3 バイトの概念(60分)

ビットがいくつか集ったものをバイト(byte)と呼ぶが、今日では8ビットを 1バイトとして取扱うように固定化されたこと、また、ビットあるいはバイト の集合としていくつかのデータ単位が表されることについてつぎの順序で解説 する。

## (1) データの単位

データを表現する単位としてビット、ディジット(digit)、字(characte-r)、バイト、語(word)の概念が用いられること、それはまたハードウェア

で直接演算の対象とするデータの単位でもあることを説明する。複数のバイトまたは語から構成されるレコードの概念にも言及する。

(2) ゾーン形式とパック形式

1 バイトに 2 ディジットを入れるパックされた10進数と 1 ディジットを 表すアンパックされた10進数またはゾーン付き10進数を説明する。

### 2.4 数値の表現(120分)

コンピュータ内部で数値がどのような形で記憶され、演算に用いられるかに ついてつぎの順序で解説する。

(1) 語 長

語を単位とした一定の2進数の数として取扱う場合と、可変長のパック された10進数の形で取扱う場合があることを説明する。

(2) 符号と補数

負数の表現方法として、絶対値に符号を添えるものと、補数表示による ものがあることを説明する。補数は2進演算の核となるものなので確実に 取扱えるようにする。

- (3) 固定小数点表示 n ビットで表せる整数および小数の範囲を示す。
- (4) 浮動小数点表示

正規形の浮動小数点表示で表す数値の範囲は基数、仮数部の桁数、指数部の桁数によって定まることを示す。

# 育成方法および指導上の留意点

- 1. この章はやや数学的であるため難解に感ずる学習者もあると思われるが、 実務で役立つ基礎的な事項を含んでいるので十分に理解するよう指導する。
- 2. 2.1(4)の基数変換、2.4(2)の補数では具体的な数値を与えて多くの演習問題を課すことにより理解の助けとする。
- 3. 2.2のコードの解説では具体的なコード表を見やすい形で各種提示する。

# 用 語

アナログ量、ディジタル量、2進数、8進数、16進数、基数、コード(符号)、BCDコード、パリティ・チェック、誤り検出符号、誤り訂正符号、ISOコード、JISコード、ASCIIコード、EBCDICコード、SI(Shift In)、SO(Shift Out)、JIS漢字コード、ビット、バイト、ディジット、キャラクタ(字)、ワード(語)、レコード、パックされた10進数、アンパックされた10進数、ゾーン付き10進数、可変長語、固定長語、固定小数点表示、浮動小数点表示、浮動小数点表示の正規形、仮数部、指数部

## 第3章 記憶装置

#### 学習目標

- 1. 記憶装置の階層構造を理解し、効果的なプログラムやシステムを構築するにはこれらをいかに組合せればよいかを考えることができるようになる。
- 2. 記憶装置の性能(記憶容量、アクセス時間)を見て、作成するプログラムやシステムの効果を推定できるようになる。
- 3. 主記憶装置および補助記憶装置について、電源の接・断、媒体のセット、 保管等に関する適正な取り扱いが出来るようになる。

## 内容

## 3.1 記憶装置の種類(120分)

記憶の階層化を話題の中心におき、つぎの順序で主記憶装置、補助記憶装置 に関する広い知識をあたえる。

(1) 主記憶と補助記憶

バイトまたは語を単位としてアドレスを付け、これによって特定の記憶位置を指定し、そこにデータを格納したり取り出すことができるものを主記憶装置と呼び、その他の記憶装置を補助記憶装置と呼ぶと定義した上で、それぞれに用いられる具体的な記憶装置の種類を列挙する。ここで再び、プログラム内蔵方式に言及しておく。

(2) 記憶容量とアクセス時間

記憶装置の性能を表す尺度として、記憶容量とアクセス時間が用いられることと、各々の定義を正確に理解させる。主記憶装置にはアクセス時間が短く、直接アクセスの可能なものが適し、補助記憶装置にはアクセス時間は長くとも大容量の装置が適していることを説明する。前者の代表として半導体記憶装置、後者の代表として磁気ディスク装置と磁気テープ装置を上げる。なお、アクセス時間とサイクル時間の違いについても言及する。

#### (3) 記憶装置の階層化

高速でしかも大容量の記憶装置を実現することは技術的経済的に困難であるため、高速小容量の記憶装置と低速大容量の記憶装置を併用し、使用頻度の高いデータを高速記憶装置に移しておく記憶階層の技法が用いられることを説明する。なお、このような操作(スワッピング、ロールイン、ロールアウト)は主にソフトウェアによって行なわれることに注意しておく。また、プログラムの実行を高速化するために主記憶装置よりさらに高速のキャッシュ・メモリを設ける場合のあることにも言及する。

#### 3.2 記憶の原理(120分)

情報処理の原点ともいうべき記憶について、その媒体となる素子と装置の記憶原理を解説する。ここでは半導体、磁性体、新素材から代表的な素子を選び、1ビットの書き込みと読み出しの原理を中心に述べる。これを用いた記憶装置全体の構成やアクセス手法については3.3で述べるものとする。

#### (1) 記憶回路

電気回路に記憶機能をもたせる仕組は理解しにくい面があるので、たと えば、つぎの順序で説明する。

コンデンサが充電されているか否かによって1ビットの記憶セルができる。これにスイッチとしてトランジスタを付けたものがDRAM(Dynamic RAM)になる。この種の記憶素子は破壊的であり、再書込が必要であることやリフレッシュについても触れる。

シーソーの原理によりフリップ・フロップ(双安定回路: flip-flop)と呼ばれる記憶セルができる。これには2個のトランジスタが使われ、一般にSRAM(Static RAM)として使用される。揮発性のものと不揮発性のものがあること。また後者の技術的手法にも言及する。

#### (2) 磁気記憶

カセット・テープやビデオ・テープ等は学習者にも馴染みのある記憶媒体である。このような磁性面を利用した記憶素子では通常、媒体が相対的に移動することによって読み書き操作が可能となる理屈を理解させる。なお、磁気コアのように媒体を移動させずに読み書き操作を行なうものにつ

いても言及する。これに関しても、破壊呼出と再書込について触れる。

# (3) その他の記憶素子

光磁気ディスク、コンパクト・ディスク(CD-ROM)、光カードなど実用段階にある光メモリの種類と記憶原理について解説する。余裕があれば磁気バブル、CCD、ジョセフソン・メモリ、ホログラフィック・メモリ、PHD(光化学ホールバーニング)など特異な記憶素子について簡単に触れておく。

# 3.3 記憶装置の仕組み(120分)

3.2で示した記憶セルを具体的な記憶装置として利用するための仕組み、すなわち、語、アドレス、レコードへの編成、アクセス論理の概要等について解説する。

## (1) 主記憶装置

主記憶装置の主流をなす I Cメモリーについて、3.2(1)で述べた記憶セルをマトリクス配置することによってバイトや語が構成されること、またアドレス単位にアクセスする機構を論理レベルで示す。余裕があれば、S R A M、D R A Mの代表的な性能、実装技術にも触れる。

# (2) 補助記憶装置

具体的な装置として、磁気ディスク装置、磁気テープ装置、磁気ドラム装置の基本的な構成を示す。これらの装置は、アクセス時間が長く、かつ媒体上で物理的に連続してデータの読み書きを行なうことができるため、一度に数十数千語をまとめて読み書きをする使い方が良く、もっぱら補助記憶装置として用いられる。物理的な対象であるトラック、シリンダ、セクタの概念はレコード、ブロック、ファイル、ボリュウムの概念と関連させて説明する。フレキシブル・ディスクとハード・ディスクについては磁気ディスクと比較しながら述べ、装置よりも媒体を中心に、その種類、取扱い方等、実用面を重視して解説し、代表的な性能一覧表を示す。光磁気ディスク装置の構造と記憶形式についても解説する。

# (3) 特殊な記憶装置

ここでは(1)(2)で述べられなかった大容量の記憶装置や特殊な目的に利

用する記憶装置について解説する。具体的なものとして磁気カートリッジ・テープ装置、大容量記憶システム、小型のバックアップ・メモリとして利用価値の高いストリーマ・テープ、簡単に主記憶装置の延長として利用されるRAMボードとEMS(Extended Memory Specification)等について解説する。

## 育成方法および指導上の留意点

- 1. 本章では、物性、電磁気の理論、回路技術・実装技術の詳細に深く立入る 必要はないが、ある程度、記憶の原理と仕組み、それぞれの装置の特性を理 解させる。
- 2. 単に素子や装置の羅列にならないように工夫することが肝要である。余裕があれば、記憶装置の歴史的な開発経過をまじえたり、映像や実物を教材として利用するのがよい。
- 3. 第8部第2章の「ファイルと記憶媒体」との関連に留意すること。

# 用語

主記憶装置、補助記憶装置、アドレス、記憶容量、アクセス時間、サイクル時間、記憶階層、スワッピング、ロール・イン、ロール・アウト、キャッシュ・メモリ(バッファ・メモリ)、半導体記憶装置(I C メモリ)、記憶セル、揮発性、不揮発性、読取専用記憶装置(ROM)、R A M、S R A M、D R A M、リフレッシュ、磁気ディスク装置、磁気ドラム装置、磁気テープ装置、トラック、シリンダ、セクタ、フレキシブル・ディスク(フロッピー・ディスク)、ハード・ディスク、磁気カートリッジ・テープ、大容量記憶システム、ストリーマー・テープ、光メモリ、光磁気ディスク、コンパクト・ディスク(C D ー R O M)、光カード、磁気バブル、C C D、ジョセフソン・メモリ、ホログラフィック・メモリ、P H D (光化学ホールバーニング)

## 第4章 制御・演算装置

### 学習目標

- 1. 機械語の構成を理解し簡単なダンプ・リストを読むことができる。
- 2. 機械語レベルでプログラムの実行を部分的に追跡できる。
- 3. CPUの仕様を見て主要なアーキテクチャを知ることができる。
- 4. 簡単な論理回路を見てその機能を知ることができる。
- 5. 積極的な学習者であれば市販のICやボードを用いて、手引書にしたがって簡単な増設作業を行なうことができるようになる。

## 内 容

### 4.1 コンピュータの命令(120分)

ソフトウェア側から見てコンピュータの構成要素となるのは個々の命令によって引き起される動作であることを認識させ、命令に関してつぎの順序で解説する。

#### (1) プログラムと命令

一つのアルゴリズムを実現するために必要な命令を配列したものがプログラムである。また、コンピュータに用意されている命令の種類は機種ごとに異なり、少ないもので数十種類、多いものでは2~3百種類にのぼる。

#### (2) 命令語の構成

命令は大別すると即値命令、直接命令、間接命令に分類される。また演算の種類によって、算術命令、論理命令、分岐命令、特権命令などに分類される。

命令語は命令コードと番地部から構成される。さらに番地部の数によって1アドレス方式、2アドレス方式、3アドレス方式等がある。命令コードと番地部の構成方法によって命令の長さが異なる。

#### (3) アドレス修飾

実効番地は直接番地、間接番地、インデックス修飾、ベースアドレスな

どにより決定される。また仮想番地方式により主記憶領域を意識せずにプログラムをつくることができる。ページやセグメントの概念についても言及する。

#### (4) 命令セット

具体的なコンピュータの命令セットについて一覧表を提示して解説する。

## 4.2 中央処理装置 (90分)

演算および制御装置をまとめて中央処理装置(CPU)と呼び、コンピュータの仕様はCPUの設計思想によりほぼ決定されることを説明する。CPUの基本的な仕組みを以下の項目で解説する。

#### ·(1) CPUの構成

CPUを構成する要素として、算術論理演算ユニット(ALU)、制御部、演算用レジスタ群、制御用レジスタ群、命令デコーダ等を主体とするブロック図を示しCPUの全体像を解説する。

#### (2) レジスタの種類と役割

代表的なレジスタとしてアキュムレータとインデックス・レジスタの役割を理解させた後、汎用レジスタが演算用あるいはアドレス修飾用として任意に利用されることを例を挙げて解説する。制御用のレジスタとしてプログラム・カウンタ、命令レジスタ、割込み状況レジスタ(PSW)の役割を述べる。また、メモリ・レジスタ、アドレス・レジスタ等ソフトウェア側からは見えないレジスタがあることにも言及する。

## (3) CPUの基本動作

CPUの基本動作がフェッチ・サイクル、実行サイクル、割込みサイクルから構成されることを説明し、その動作を概説する。なお詳細は4.3で解説することにする。

# 4.3 プログラム実行の仕組み (90分)

命令制御はコンピュータの動作原理を集約する部分であり、自動車のエンジンに相当する部分である。したがって、その仕組みと動作については、下記の順序でやや詳細な解説を行ない十分理解させる。

#### (1) 逐次制御

4.2で説明したCPUの動作をさらに詳しく解説し、逐次制御の方式を 理解させる。

### (2) 先行制御

逐次制御の改良法として先回り制御、後回り制御、パイプライン制御等 の方式が用いられることを解説する。

(3) 割込み制御 割込み処理の方法について解説する。

(4) マイクロ・プログラム

命令の実行にあたって、従来からの配線方式に対してマイクロ・プログラム方式が広く採用されていることを述べ、その仕組みの概要について解説する。その応用の一つとしてエミュレータについても言及する。

## 4.4 演算の仕組みと論理回路(180分)

コンピュータにより実行される演算の種類とその仕組について、下記の手順 で段階的に解説し、最終的に論理回路の初歩を習得させる。

## (1) 演算の種類

固定小数点演算、浮動小数点演算、10進演算、論理演算について2進数のレベルで具体的に解説する。

# (2) 加算器と真理値表

半加算器の入力と出力を表の形で示し、真理値表の意味と書き方を理解 させる。

# (3) 論理回路とブール代数

コンピュータの内部で実際に演算を実行するのは半導体素子で組み立てられた論理回路であり、その機能、動作を記述するにはブール代数を用いるのが効果的であることを理解させる。AND、OR、NOT、NAND、NORなどのゲートの機能を理解させ、ブール代数の基本的な定理とその応用を簡単に解説する。

# (4) 論理回路の簡単化

素子の節約、簡明な回路設計のために論理回路の簡単化が不可欠である

ことを理解させ、ベイチ図表等による簡単化の例を示す。

#### (5) 基礎的な回路の設計

全加算器、補数器、カウンタ、デコーダ等の簡単な論理回路の動作を理解し、また、自分で書けるようにする。

## 4.5 半導体素子と集積回路(120分)

本節はディジタル回路の中核となる半導体素子と集積回路についての基礎知識を与えるためものである。対象となる学習者の興味と志向によって取捨選択するのがよい。

### (1) 半導体の物性

半導体とは何かを、導体、絶縁体、電子の移動、ホールの移動などと関連づけて解説する。

#### (2) 半導体素子

ダイオード、トランジスタ、その他の半導体素子の機能と動作原理を理 解させる。

### (3) 集積回路

集積回路の発展過程を解説しながら、その構造、機能、製法についての概念を与える。代表的な製品の特性を示し取り扱いの手ほどきを行なう。

# 育成方法および指導上の留意点

- 1. 4.1~4.3では基本的な概念(とくにアドレス修飾、命令制御、割込みの概念)をよく理解させる。ここは、一般論のみでは把握が困難なので、実習に使用するコンピュータまたは仮想のマシンに則して大筋を解説するのがよい。
- 2. 想定するマシンは、1種類に限らず、方式の異なる2種類のコンピュータを対比させて解説することが望ましい。この際、1種類は大型汎用機、他はパソコンやワークステーションのアーキテクチャを採用するのも1つの方法である。
- 3. 4.4では多くの演習問題を提示し、理解を深め確実な作業が出来るように

するのがよい。

## 用語

算術命令、論理命令、分岐命令、特権命令、命令コード、番地部、アドレス部、1アドレス方式、2アドレス方式、3アドレス方式、直接番地、間接番地、インデックス修飾、ベースアドレス、仮想番地、ページ、セグメント、算術論理演算ユニット(ALU)、アキュムレータ、インデックス・レジスタ、汎用レジスタ、プログラム・カウンタ、命令レジスタ、割込み状態レジスタ、メモリ・レジスタ、アドレス・レジスタ、命令デコーダ、フェッチサイクル、実行サイクル、割込みサイクル、逐次制御、先回り制御、後回り制御、パイプライン制御、割込み、マイクロ・プログラム、エミュレータ、論理回路、ブール代数、AND、OR、NOT、NAND、NOR、ゲート、ベイチ図表、カウンタ、全加算器、補数器、デコーダ

## 第5章 入出力装置

## 学習目標

- 1. 処理の目的に応じて適正な入出力装置を選択できるようになる。
- 2. 各入出力機器を動作原理に基づいて適正に取扱うことができるようになる。
- 3. 入出力制御の方式を理解し、それに応じて効率の良いプログラムを設計 することができるようになる。
- 4. 入出力インタフェースの知識に基づいて、システムに周辺装置を接続することができるようになる。

# 内 容

## 5.1 入出力機器の種類(180分)

主な入出力機器について、つぎの3種類に区分して解説する。必要に応じて、 その機器で使用する媒体の形式についても例を示して解説する。

(1) 対話形式の入出力機器

特別の媒体を介さず利用者と直接やり取りを行なうことができる機器について述べる。具体的にはキーボード、マウス、ディスプレ装置、グラフィック・ディスプレイ装置とライトペン、音声入力装置、音声出力装置、データ端末装置などが上げられるが、さらにジョイスティックやコントロール・ボールなどゲーム関連の機器もマウスとの類似性から言及できよう。

# (2) シート形式の入出力機器

人間にも識別できる表示方式の記録用紙を媒体として入出力を行なう機器について述べる。この種の入力機器としては光学式文字読取装置(OCR)、マーク読取装置、磁気インク読取装置などがあり、さらに図形入力のためのタブレット、ディジタイザ、イメージ・センサ等がある。出力装置としては各種のプリンタとプロッタがある。プリンタについては、インパクト型とノンインパクト型、あるいはシリアル・プリンタ、ライン・プ

リンタ、ページ・プリンタなどに区分して解説する。

#### (3) その他の入出力機器

上記(1)、(2)に含まれないものをこの項で述べる。カード読取装置、紙テープ読取装置、カード穿孔機、紙テープ穿孔機などはいずれも紙を媒体とするが、一般に、記録されたものの読取りが人間にとって困難であるため(2)のグループには入れ難い。POS端末装置や磁気カード読取装置等もこのグループに入る。

### 5.2 入出力制御(120分)

上に述べた機器がCPUあるいは主記憶装置とデータのやり取りを行なう仕 組について下記の項目を解説する。

### (1) 入出力制御装置の役割

入出力装置はそれぞれ入出力のための比較的単純な動作をするもので、 一連の入出力処理のためには制御が必要である。また、入出力装置の多く は機械的動作を伴うため、演算装置、記憶装置と比べて格段に動作が遅い ため両者の整合をはかる必要がある。この作業を担当するのが入出力制御 装置である。

# (2) 入出力制御の種類と方法

入出力制御は、CPUで行なう直接制御方式と、独立の入出力制御装置を設けて行なう間接制御方式がある。

直接制御方式では、入出力の動作をCPUの命令で制御するプログラム I/Oが基本であるが、CPUの負担を軽減し利用効率を高める目的で、 入出力割込みが生じるまでCPUの演算処理を並行して実行させる方法や すべての入出力装置に共通のバスラインを設ける方法がとられている。小型コンピュータの多くは直接制御方式が用いられる。

間接制御方式の簡単なものに、DMA (Direct Memory Access)方式がある。これはCPUから起動されたDMAが、主記憶装置と入出力装置の間で連続的にデータの受け渡しを引受けるものである。また、入出力制御の中から個別の機器に固有の部分を切り離し、共通の機能を取り出してまとめたのがチャネルである。チャネルはCPUからの命令を受けると、以

降はCPUと独立してチャネルプログラムを実行し、主記憶と入出力機器の間のデータ転送と機器の制御を行なう。汎用コンピュータの多くはチャネル方式により入出力制御を行なう。

間接制御方式では入出力データを一時的に蓄えておくバッファが必要である。バッファは入出力装置側に設ける方式と主記憶装置内に設ける直接 バッファ方式があるが、後者がより効率的で柔軟性がある。

## (3) チャネルの種類

チャネルは個々の入出力装置とのインタフェースを標準化し、同一チャネルに多様な機器の接続を可能とする。接続する機器の速度によって、セレクタ・チャネル、ブロック・マルチプレクサ・チャネル、バイト・マルチプレクサ・チャネルに分類される。また機器の接続関係によって固定チャネルと浮動チャネル、あるいは専用チャネルと汎用チャネルに分類される。

## (4) 接続の具体例

上で述べた各種の入出力制御方式に基づく具体的な接続の例を示し、理 解を確実なものとする。

# 5.3 入出力インタフェース(180分)

分散化志向が広がり、パソコンやワークステーションが普及するにつれて、利用者自らが必要に応じて周辺機器を調達し、システムへ接続するケースが多くなっている。このとき問題になるのが、システムと周辺機器の整合性と接続方法である。本節ではこの点に着目し、標準インタフェースの利用の仕方、比較的利用の多いA/D、D/A変換の考え方、インタフェース自作のための代表的IC等の基礎知識を与え、利用者が上記のケースに積極的に取組めるようにする。

# (1) インタフェースの基礎

インタフェースの機能と構成について、制御方式、信号レベル、タイミング等の基礎的な概念を理解させる。さらに、入出力インタフェースについて、直列方式と並列方式、同期式と非同期式、入出力データの転送制御方式等の解説を行なう。

### (2) 標準入出力インタフェース

主としてパソコンやワークステーションに採用されている標準的な入出 カインタフェースについて解説し、これに基づいた入出力機器を利用者が 独力でシステムに接続できるようにする。具体的には、直列伝送用のRS-2 32C、並列伝送用のGPIS、標準プリンタ用のセントロニクス規格、ハード ・ディスクに対するSCSIとSASI等の仕様説明をする。

(3) アナログ・ディジタル変換の基礎

自然界の物理量はほとんどがアナログ量であり、人間の視覚や聴覚もアナログ量を識別するため、コンピュータが外界と直接データのやり取りを行なうには、アナログ・ディジタル変換(A-D/D-A変換)が必要である。ここでは、A-D/D-A変換の役割と動作原理、代表的なA-D/D-A変換器の特徴と使い方等の解説をする。

(4) インタフェース用集積回路

上記の標準インタフェースに対して、利用者のオリジナルなインタフェース・ボードの作成を支援する代表的なインタフェース用 I Cを紹介する。ただし本項は4.5(3)の知識が必要となるため、学習対象者に応じて取捨選択すべきものである。

# 育成方法および指導上の留意点

- 1. 各種の入出力機器の解説は映像、実物を用いて理解を深める工夫をすると 同時に、解説する機器の選定と内容が陳腐化しないように気を配る。
- 2. 分散システムにおける入出力機器はほとんどの場合、利用者自身が操作・ 管理することになるので、指導内容にはその点を十分に考慮する必要がある。
- 3. 入出力制御は初心者にとって難解な課題の一つであるから、あまり詳細に 入る必要はない。しかし、プログラムの効率に関係する部分なので概要は正 確に把握させること。
- 4. 入出力インタフェースについては対象外の学習者も多いと思われるが、システムの複雑化と処理内容の多様性、さらにはパソコンやワークステーションの普及状況を考えると、ある程度の知識・技術が必要になると予想される。

### 用語

キーボード、マウス、ディスプレ装置、グラフィック・ディスプレイ装置、ライトペン、音声入力装置、音声出力装置、データ端末装置、ジョイスティック、コントロール・ボール、光学式文字読取装置(OCR)、マーク読取装置、磁気インク読取装置、タブレット、ディジタイザ、イメージ・センサ、プリンタ、プロッタ、インパクト型プリンタ、ノンインパクト型プリンタ、シリアル・プリンタ、ライン・プリンタ、ページ・プリンタ、カード読取装置、紙テープ読取装置、カード穿孔機、紙テープ穿孔機、POS端末装置、磁気カード読取装置、入出力制御装置、直接制御方式、間接制御方式、入出力割り込み、バスライン、DMA、バッファー、直接バッファ方式、間接バッファ方式、チャネル、チャネル・プログラム、セレクタ・チャネル、ブロック・マルチプレクサ・チャネル、バイト・マルチプレクサ・チャネル、固定チャネル、浮動チャネル、専用チャネル、汎用チャネル、入出力インタフェース、信号レベル、タイミング、直列転送方式、並列転送方式、同期式、非同期式、RS-232C、GPIS、セントロニクス規格、SCSI、SASI、アナログ・ディジタル変換(A-D/D-A変換)、インタフェース用集積回路、インタフェース・ボード

# 第6章 コンピュータの歴史と将来

## 学習目標

コンピュータの歴史、現状、今後の発展の方向について展望し、広い視野に 立って事に対処できるようになる。

# 内 容

# 6.1: ハードウェアの歴史(90分)

算盤、計算尺からノイマン型コンピュータに至るまでの計算機器の発達史について、関連する科学技術の発展と結びつけて解説する。歴史を飾るコンピュータを列挙すると切りがないが、今日のコンピュータのルーツとも言えるバベッジの解析機関、初の電子スイッチングを採用したENIAC、プログラム内蔵方式を採用したEDSACについてはハードウェアの歴史を語る際には欠かかせないであろう。

# 6.2 コンピュータの現状と将来(90分)

電子管式ディジタル・コンピュータの出現以降、ハードウェアは半導体技術の発展に支えられた演算速度の向上と記憶装置の大容量化を主軸として進歩してきた。さらに、パイプライン処理、並列処理、記憶の階層化などによる方式設計上の工夫が大きく貢献してきたことを解説する。また、時分割方式、実時間処理、多重プロセッシング、分割処理など処理方式の多様化、コンピュータ・ネットワーク、周辺装置の発展がハードウェア技術に与えた影響について言及する。

次に今後の発展の方向について、素子の面では光素子、バイオ・エレクトロニクス等が、方式設計の面ではノイマン型を脱却したものとしてデータ駆動型、要求駆動型のコンピュータ、さらにわが国の第5世代コンピュータ・プロジェクトに見られるような推論機構等が研究課題となっていることを解説する。

### 育成方法および指導上の留意点

- 1. 前半の歴史編では、計算機器の発達と科学技術の進歩との相互関係に興味をもたせ、思考させることが望まれる。後半の現状と将来に関しては、技術動向に注意し最新の知識を与えるように気を配ることが肝要である。
- 2. この章は初級情報処理技術者にとっていわば教養であり常識である事柄を まとめたものである。場合によっては、それぞれの関連ある章に分散して講 義をすることも可能である。この方が講義に潤いをもたせることができるや も知れない。

### 用語

ノイマン型コンピュータ、バベッジの解析機関、ENIAC、EDSAC、パイプライン処理、並列処理、時分割方式、実時間処理、多重プロセッシング、分割処理、コンピュータ・ネットワーク、光素子、バイオ・エレクトロニクス、データ駆動型、要求駆動型、第5世代コンピュータ、推論機構

### 推薦教材

1. 「ハードウェア」(初級情報処理技術者テキストシリーズ) 脚日本情報処理開発協会中央情報教育研究所,1987年

## インストラクタ必読書

1. 「計算機工学レクチャーノート」 市川忠男・阿部忠 著, 日刊工業新聞社

## 参考文献

- 1. 「ハードウェア」(初級情報処理技術者テキストシリーズ) (脚日本情報処理開発協会中央情報教育研究所, 1987年
- 2. 「電子計算機への手引き」 森口繁一・筧捷彦・高沢嘉光 著,岩波書店,1982年
- 3. 「計算機の機能と構造」

矢島脩三 著,岩波書店,1982年

- 4. 「電子計算機の方式設計」
  - 猪瀬 博 監修, 萱島興三 著, 産報出版, 1972年
- 5. 「電子計算機基礎論」 手塚慶一 著, 昭晃堂, 1978年
- 6. 「入出力制御とシステム構成」 田丸啓吉・松本吉弘・浪本敬二 著、オーム社、1982年
- 7. 「ハソコン周辺機器インタフェース詳細」 トランジスタ技術 (SPECIAL NO. 9) C Q 出版, 1989年
- 8. 「記憶特集」(電子情報通信学会誌) 電子情報通信学会, 1990年 4 月

# 第3部、ソフトウェア

# 教育目標

ハードウェアと対等な情報資源としてのソフトウェアについて、現状の課題 や今後の動向など、技術者としての常識的知識を与えるとともに、実務能力の 学習ではカバーしきれぬソフトウェア技術の背景や、進歩の経緯につき理解さ せる。

## 内容の構成

	時 間 数	
内 容	講義	演習
1. ソフトウェアの体系 1.1 ソフトウェアとは何か 1.2 基本ソフトウェア 1.3 応用ソフトウェア	5	0
2. プログラム言語と言語処理プログラム 2.1 プログラム言語の分類 2.2 高水準言語の種類と特徴 2.3 簡易言語の特徴と種類 2.4 言語処理プログラムの役割	11	
3. オペレーティング・システム 3.1 ポレーティング・システムの目的 3.2 ポレーティング・システムの基本機能 3.3 実OSの機能と利用実習	1.0	2
4. サービスプログラム 4.1 サービスプログラムの目的と課題 4.2 テキストエディタとその利用実習	. 4.	2
5. ソフトウェアの課題 5.1 生産性と品質の向上 5.2 開発環境と開発支援ツール 5.3 共通ソフトウェアの活用	3	0
合 計	3 3	4

## 第1章 ソフトウェアの体系

## 学習目標

ソフトウェアという言葉の意味する内容を説明でき、ソフトウェアを開発するとは、どういうことなのかを作成物の立場から説明できる。

ハードウェアとソフトウェアの役割分担に関する変化を認識し、ソフトウェア開発に要求される技術についての大まかな見通しを立てられる。

ソフトウェアの分類を正しく基本ソフトウェアと応用ソフトウェアに分ける ことができ、基本ソフトウェアと応用ソフトウェアのそれぞれが何を意味する かを説明できる。

# 内容

## 1.1 ソフトウェアとは何か(120分)

## (1) 語源と役割

ソフトウェアという言葉は、ハードウェアに対する言葉として1960年代 に作られた。

ハードウェアは英語の「金物」から来ており、コンピュータの機械としての実体を指す。ソフトウェアは、そのハードウェアに仕事をさせる場合の利用技術の総称であり、広義には、

プログラム:コンピュータに処理させたい一連の手続きをプログラミング言語で表現したもの

ドキュメント:ソフトウェアの計画・作成過程および保守用のいっさ いの書類

使用説明書:利用者マニュアル、操作マニュアル、保守用マニュアル などの総称である。

また、狭義には、プログラムと同義に使われることがある。

まず、大枠としてのソフトウェアという概念を理解させ、たとえばどの ようなものがソフトウェアかを例をあげて説明することによって会得させ る。

次に、ソフトウェアの範囲を明らかにし、ソフトウェアを生産するとは どういうことをさせるのか、そして、何をアウトプットとするのかを具体 的な例をもとに説明する。

## (2) 機能分担の変化

コンピュータの利用範囲の拡大と、ハードウェア、ソフトウェア双方の 技術的進歩によって、両者の機能分担は、次のように徐々に変化してきて いる。

- ・効率に問題がある部分(浮動小数点演算や仮想メモリ制御)のハー ドウェア化
- マイクロ・プログラムの使用によるファームウェア化
- ・高水準言語を一種の機械語として直接処理するコンピュータの出現
- ・通信制御、画像処理などに適した専用コンピュータのシステム構成 要素としての利用

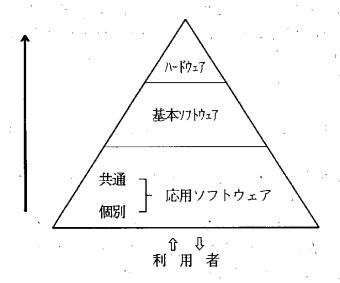
このように従来ソフトウェアで実現していた多くのことが、ハードウェア化、ファームウェア化され、ソフトウェアはもう少し高いレベルの機能を実現するのに使われるようになった。そういう意味では、ソフトウェアの開発は楽になった(生産性が上がった)といえるが、一方、ハードウェアの発展とともに、そのインタフェースの部分がますます複雑化し、高い技術力が要請されるようになった。

このような現状を踏まえて、ソフトウェアを開発することのむずかしさ、 また楽しさを会得させるようにする。

#### (3) ソフトウェアの構成

ソフトウェアは、大きく分けると、図表3-1-1に示すように基本ソフトウェアと応用ソフトウェアとに分けられる。

図表3-1-1 ソフトウェアの階層



基本ソフトウェアとは、ハードウェアとソフトウェアの中間にあって、コンピュータの種々の応用に対する共通的な機能を果たし、応用ソフトウェアの負担を軽減する役割を果たす。基本ソフトウェアがなくては、通常のプログラムの開発もできず、応用ソフトウェアも動かない。このように基本ソフトウェアは、コンピュータ利用の要となるプログラムである。

ソフトウェア技術者が開発するプログラムは、ごく一部のプログラムを除いて、この基本ソフトウェアの存在を前提として、基本ソフトウェアの上に乗せるように開発するのが普通である。

応用ソフトウェアは、コンピュータのすべての応用に対応する多様なプログラムの総称である。

利用者は、おもにこの応用ソフトウェアを通して、コンピュータを使う。 今日では、すべてのプログラムを自作するという時代ではなく、ソフト ウェア技術者といえども、この応用プログラムの恩恵を被らなければなら ない。

したがって、ソフトウェア技術者は、基本ソフトウェア、応用ソフトウェア双方に精通していなければならないことを理解させ、そのような技術

を総合的に身につける訓練をすることがシステム開発には欠かせないもの であることを認識させる。

# 1.2 基本ソフトウェア(120分)

(1) 基本ソフトウェアの機能

基本ソフトウェアの果たす機能を次の各項目に従って例を交えて説明する。

- ① ハードウェア機能の効果的な利用を支援する
- ② 大量データの処理や通信処理を始め、種種の応用に共通なサービス機能を提供する
- ③ コンピュータ・システムの性格・安全かつ効率的な運用を支援する
- ④ ソフトウェアの作成を容易にする

たとえば、「ハードウェア機能の効果的利用」に関しては、コンピュータの発展初期の頃、ハードウェアの価格が高かったため、その利用効率を上げることが、最大の課題だった。そして、基本ソフトウェアは、利用効率を上げるために開発・発展してきた。その究極的な姿がオペレーティング・システムである、というような内容を述べる。

# (2) 基本ソフトウェアの種類

基本ソフトウェアに含まれるものとして、以下の分類を示し、各々の基本的な役割を説明する。特に制御プログラムは、今日、オペレーティング・システム(の中核部分)として重要である。そしてまた、通信制御プログラムも、通信回線でコンピュータ同士を結び付けるようなシステムが中心となっている現在では、重要かつ中心的な役割を果たすプログラムであることを説明する。

- ① 制御プログラム
- ② 通信制御プログラム
- ③ 言語処理プログラム
- ④ データベース管理プログラム
- ⑤ ソフトウェア開発支援プログラム

### 1.3 応用ソフトウェア(60分)

応用ソフトウェアには、業務・業種にかかわらず、ほぼ共通して利用できる もの(共通応用ソフトウェア)と、個別のもの(業務・業種別応用ソフトウェ ア)とが存在する。

ここでは、図表3-1-2を参考にして、共通応用ソフトウェアと業務・業種別 応用ソフトウェアの全体像を把握させる。

業務・業種別応用ソフトウェアは、ここに示したもの以外にも、多種多様なものが開発されているし、また、これからも開発されていくであろう。そして、共通応用ソフトウェアも、時代が移り変わって人間とコンピュータのかかわり方が変わるとともに、その形を変えていき、新しいものが後から後から出てくるし、古いものは廃れていく運命にある。コンピュータの利用が拡大するとともに、この表の内容は変わっていくものだということを印象づける。

# 図表3-1-2 応用ソフトウェアの種類

		· · · · · · · · · · · · · · · · · · ·
共通応用ソフトウェア	□数値処理 ①数値解析 ②統計解析 ②統計解析 ②数理計画 ③予測・シミュレーション ④線型計画法 ⑤日程計画 □エンジニアリング ⑥設計支援 ①構造解析 ⑧数値制御	□共通技術 ②文書処理 ①図形像理 ①図形像型理 ②情報機械計 ②機械計 ②と ②と ②は のので のので のので のので のので のので のので のの
業務・業種別応用ソフトウェア	□教 育 ①CAI ②CMI □医療・保健 ③医療・保健 ③医療診断 ④病院運営管理 ⑤救急医療支援 ⑥病床管理 ⑦予防医療支援 □公 共 ⑧防災・防犯 ⑨交通制御 ⑩環境管理	□公務 ①政策支援 ②自治情報サービス ③社会保険管理 ④財務統制 □業種別 ①金融・保険 ②流通 ③製造 ④出版報 ②流通 ③製造 ④出版報 ⑤ユーティリティ ⑥土木・建築 ⑦化学工業 ③輸送 ③サービス ②原・教 ①明宗教 ①学術・研究 ②情報産業

# 育成方法および指導上の留意点

この章では、ソフトウェアの基本概念を理解させることに重点をおくが、つ ねにソフトウェア生産者としての立場から、ソフトウェアの全体像をとらえる ように指導する。

いま、自分が開発しようとしているプログラムがこのソフトウェア体系の中でどこに位置するものなのか、つねに考えながら仕事をするという態度が、ソフトウェア技術者にとっては重要である。

また、新たにソフトウェアを開発するときでも、基本ソフトウェアなど、すでに作成されていてコンピュータの上で稼働しているプログラムの助けを借りなければならない。その概念を理解し、使い方を学ぶことが、ソフトウェア開発にとって必須条件である。

こういう点について理解させるとともに、このあと説明する各構成要素と自 分とのかかわり合いが正しくとらえられるように、ここで十分に説明しておく。

# 用語

ソフトウェア, プログラム, ドキュメント, 使用説明書, 基本ソフトウェア, 応用ソフトウェア, 制御プログラム, 通信制御プログラム, 言語処理プログラム, データベース管理プログラム, ソフトウェア開発支援プログラム

# 第2章 プログラム言語と言語処理プログラム

# 学習目標

プログラム言語の種類を機械語、アセンブリ言語、高水準言語、簡易言語に 分け、そのそれぞれの特徴を説明できる。特にFORTRAN やCOBOL など、代表的 な高水準言語を列挙し、その特徴と主な適用分野を説明することができる。

A Commence of the State of the

言語処理プログラムの役割と基本的機能を説明することができ、実作業に沿った形でコンパイルなどの実行をさせることができる。

# 内 容

# 2.1 プログラム言語の分類(120分)

プログラミング言語の流れを歴史を追って説明することにより、低水準から高水準になるに従って、機械側から人間のほうに近づいてきたことを理解させる。また、最近では、処理を記述する代わりに機能を直接記述する非手続き型言語や宣言型言語を使用する機会も多くなっていることを体験させる。

# (1) プログラム言語の概要

機械語やアセンブリ言語のような低水準言語とコンパイラ言語のような 高水準言語がある。この水準とはコンピュータより人間に近いかどうかで 決められる。コンピュータ言語の発達の歴史を通して、低水準言語から高 水準言語に移り変わっていった様子を機械語、アセンブリ言語、高水準言 語の機能を説明することにより理解させる。

また、次のような観点からプログラム言語の重要性について理解させる。

- 1. プログラム言語のあり方(言語仕様やレベル)が、ソフトウェア作成の生産性や信頼性に大きく影響する。
- 2. プログラム言語の進化がコンピュータの利用技術進歩の源泉となり、 コンピュータの利用拡大に大きく貢献した。
- 3. 機械語やアセンブリ言語はハードウェアに固有のものであるが、高 水準言語の出現によって、機械の違いを越えたプログラムの共通化か

可能になった。また、FORTRAN 、COBOL 、Pascal、Cなどは国際的な標準が作られ、移植性の点でほとんど問題がなくなった。

4. プログラム言語の分類には、手続き型言語と非手続き型言語という 分類基準もある。手続き型言語とは「いかに」処理するかを記述する 言語で、処理手順やアルゴリズムが重要な要素となっている。非手続 き型言語とは「何を」行いたいかを記述する言語で、処理手順などよ り機能を定式化する技術が重要な要素となっている。

一般に非手続き型言語のほうが水準は高いとされており、生産性増 大などの理由から、使用する割合が多くなってきている。

## (2) 機械語

ハードウェア固有の言語。その機械が理解することのできる言語であり、 具体的にはビットの列である。

### (3) アセンブリ言語

機械語と1対1に対応する命令を覚えやすいように表意記号(ニーモニック)で表し、アドレス部も記号で書けるようにした言語である。今日のほとんどのアセンブリ言語には、いくつかの命令を集めて、それに名前をつけて呼び出せるようにした、マクロというものが付属している。

#### (4) 高水準言語

利用者がハードウェアを意識しないで、自分のやりたい処理がそのまま書けることを目標に設計された言語であり、目的によって事務処理用、科学技術計算用などに分けることもある。コンパイラで処理されるので、コンパイラ言語と呼ばれることも多い。

初期の言語は行単位に処理することを対象として設計されたものが多かったが、最近ではフリー・フォーマットのものが多くなっている。

また、最近のほとんどの言語はいわゆる構造化プログラミング用の構文 (IF THEN ELSE、WHILE )をサポートしている。

### (5) 簡易言語

コンピュータの普及につれて、エンド・ユーザーがコンピュータに触れる機会が多くなった。そこで、プログラミングの初心者でも簡単に業務プログラムが作成できる簡易言語が出現した。

簡易言語には、データベースとか表計算に付属した処理記述用言語(マクロ)と、本格的な業務においてプログラミングの手間を大幅に減らせることを目的とした第4世代言語とがある。

### 2.2 高水準言語の種類と特徴(360分)

#### (1) 言語体系と水準

プログラム言語の歴史を図表3-2-1に沿って説明し、どのように進化してきたかを理解させる。

年 関数型 手続き型 論理型 1960 **FORTRAN** ALGOL Lisp COBOL ALGOL68 BASIC InterLisp 1970 Pasca1 **PROLOG** Simula FORTRAN77 1980 ModulaI Smalltalk-80 CommonLisp 1990

図表3-2-1 言語の歴史

# (2) 汎用言語の特徴

汎用的に使われる言語の例をあげ、その特徴と主な適用分野を図表3-2-2に従って説明する。

同一の問題を各言語で書いたプログラム(約20行)を用意し、言語ごとの特徴や違いを認識させる。特に次の点に注意して指導する。

- ・フリー・フォーマットか固定フォーマットか
- ・使用文字は大文字だけか小文字も使えるか
- ・文の終わりは行末か、セミコロン(;)ピリオド(,)などの区切り文字が必要か。
- ・手続き型か関数型か論理型か
- ・モジュール化は可能か、そのときモジュールの識別には何を使うか
- ・バッチ志向か会話型志向か

# (3) 応用向き言語の種類と特徴

図表3-2-3を参考に、代表的な応用向き言語の例をあげ、その特徴を説明する。

図表3-2-2 汎用的言語の例

	· · · · · ·	
言語名	主な適用分野	<b>数</b>
FORTRAN	科学技術計算	1957年に実用化され最も歴史が古い。IBMにより 開発され科学技術計算用フロクラムの蓄積は大である
ALGOL	科学技術計算	1960年に出現し、当初はアルゴリズムの記述を主目的とした。構造化の概念を持っており、大学・研究機関で多用されていた。
COBOL	事務処理	1960年代初期に米国防省の標準的言語として開発 され、現在世界的に最も普及している。データ記 述と処理手続きを完全に分離したのが特徴である
PL/ I	汎 用	FORTRAN, COBOL, ALGOLの特徴を取り入れ、さらに アセンブリ言語に代わる機能をも包含し、システム・プログラム用にも利用可能とした。IBM により開発され、現在同社の標準的言語となっている
BASIC	汎 用	1960年代にTSS用会話型言語として開発されたが 1970年代後期にハーソナル・コンヒュータ用に再び返り咲き、 初心者向きの言語になっている。
PASCAL	科学技術計算システム・ブログラム	教育用構造化プログラミングに適した言語として 大学関係を中心に普及した。
LISP	人工知能 システム・フロクラム	関数型言語の代表格で、本来はリスト処理用である。1960年代に出現した歴史ある言語であるが、 近年人工知能研究の活発化にともなって多用されるようになった。
PROLOG	人工知能システム・プログラム	757スおよび英国で生まれ育った述語論理型言語であり、今後の知識情報処理分野での利用が期待されている。第 5世代コンヒュータの核言語のベースとなっている。
С	システム・ブログラム	1972年UNIXポレーティング・システムの下で働く言語として 誕生し、バーソナル・コンピュータを含むUNIXの普及によって 多用されるようになった。構造化機能を持つ。
APL	科学技術計算システム・プログラム	1950年代にすでに発想されていたと言われるが、 その後機能追加やオンライン環境での利用が主として IBMによって進められてきた。論理演算や行列演 算が強力である。

(続く)

 $(x_1,x_2,\dots,x_n) = (x_1x_1,\dots,x_n)_{n \in \mathbb{N}} (x_1,x_2,\dots,x_n) = (x_1,x_2,\dots,x_n)_{n \in \mathbb{N}} (x_1,\dots,x_n)$ 

言語名	主な適用分野	特 徵
SNOBOL	文字処理 システム・プロクラム	文字ストリングの処理を目的とした言語で1960年 代初期に出現し、その後種々の機能拡張が行われ 対スト処理、言語処理などに多く用いられる。
Ada	汎用 リアルタイム処理	1980年米国防省の新しい標準言語として出現した リアルタイム処理機能を含む、大型の言語である。 従来の各種言語の特質を集大成したものとも言われ、プログラム部品の概念も持つ。
Small talk	研究用 システム・フロクラム	処理の対象となる一切をガシェクトを主体としたガシェ クト指向という新しい概念に基づく対話型言語で、 心理学的検討成果が加味されている。1980年代に 開発された。

図表3-2-3 応用向き言語の例

シミュレーション言語	GPSS, SIMSCRIPT, DYNAMO, SIMULA
統計処理言語	SPSSP, BMD
数式処理言語	FORMAC, MACSYMA
数値制御言語	APT
構造解析言語	COGO, STRESS, NASTRAN
図形処理言語	GLIDE, GPL/I
プロセス制御言語	DACS-AUTRAN, PM/C
CAI言語	ELIZA, TUTOR

なお、各言語の説明にあたっては、次のような点に注意する。

- 1 各言語の詳しい説明はいらないが、それぞれの特徴が理解できるように例題などを使って説明する。
- 2 汎用言語のうち、特色のあるもの(Smalltalk 、Ada 、Lisp、Prologなど)は言語の歴史とともに、その設計思想を理解させるようにつ

とめる。

3 応用向き言語は、その対象を特化することにより、一般に汎用言語 よりさらに高水準化が可能であることを実例で理解させる。

#### 2.3 簡易言語の特徴と種類(60分)

#### (1) 第 4 世代言語

エンド・ユーザが業務処理プログラムを作成する手間を軽減することを目的に作られた言語で、高水準言語のプログラムを生成できるようにしたものもある。データベースや表計算をその枠組みの中に持っていて、定型業務処理についてはパラメタの指定だけですむことが多い。

機械語を第1世代、アセンブリ言語を第2世代、コンパイラ言語を第3 世代言語というのに対して、コンパイラ言語の次世代ということから第4 世代言語と名付けられた。

レポート・ジェネレータとして名高いRPG などもパラメタ指定でCOBOLのソース・プログラムを出力することからこの分類に入れる。

代表的な第4世代言語を1つ選び、その機能を説明するとともに、入力 形式と出力された第3世代言語のソース・プログラムを見せて、両者を比 較対照させることにより、入力の簡便さを理解させる。

#### (2) 宣言型言語

論理型言語のProlog、オブジェクト指向言語のSmalltalk などでは、処理手順を書くのでなく、機能だけを書けばいいような仕様になっている。

宣言をするようにプログラムを書くことができることから、宣言型言語と呼んでいる。しかし、現在の汎用言語では完全に機能だけを書くことにより全体を作れるようにはなっていなくて、一部、手続きも併用して書くというのが普通である。

#### 2.4 言語処理プログラム(120分)

#### (1) 言語処理プログラムの役割

ハードウェアは機械語しか理解できない。それ以外の言語は機械語に翻訳する必要があり、その翻訳プログラムが言語処理プログラムである。プ

ログラム言語で書かれたものをソース・プログラム、機械語に翻訳されたものをオブジェクト・プログラムという。通常、オブジェクト・プログラムは、サブルーチンなどのアドレスが未定のままであり、そのままで実行できる形式になっていない。いくつかのオブジェクト・モジュールとライブラリの中のルーチンとを組み合わせて実行できる形式になったものをロード・モジュールという。

#### (2) アセンブラ

アセンブリ言語で書かれたプログラムを機械語に翻訳する記号処理系。 命令部、アドレス部には記号を使用することができるため、記号を置き 換えることが処理の中心である。

いくつかの命令をまとめてひとつの命令で呼び出せるようにした、マクロの展開、命令部の解読と機械語命令への置き換え、アドレス部の解釈とアドレッシング・モードの判定、アドレス式の処理などを中心にアセンブラの基本機能を説明する。

#### (3) コンパイラ

FORTRAN、COBOL などの高水準言語で書かれたプログラムを機械語に翻訳する処理系がコンパイラである。そのことからFORTRAN などはコンパイラ言語と呼ばれる。

コンパイラの基本的な機能を、字句解析、構文解析、意味解析、コード 生成の各段階を追って説明する。また、コンパイルの前段階の処理を受け 持つプリプロセッサ、コンパイルの結果生成した機械語コードの最適化を するオプチマイザについても多少触れ、現在ではこういう処理も含めてコ ンパイラ全体を考えることが多いことを理解させる。

自分の作ったプログラムを実際にコンパイラに通してみることにより、 コンパイラの持つエラー検出機能が、人間の犯す文法エラーを指摘するの に、いかに役立っているかを認識させる。

#### (4) インタプリタ

プログラムのソース・コードを1行ずつ解釈し、直ちに実行する方式の 言語処理プログラムがインタプリタである。インタプリタ方式はTSS や対 話型処理に適している。たとえば初期のBASICやLISPはインタプリタ方式 で実行していた。

インタプリタとコンパイラとの違いを中心に説明する。特に、実行速度やプログラミング環境としての即実行形式とデバッグのしやすさの側面から理解させる。できればコンパイラとインタプリタの両方がある言語で、 具体的な例題を実行してみることにより、両者の違いを比較するとよい。

#### (5) ジェネレータ

処理内容をパラメタで与えることにより、予め用意されている部分的な プログラム群の中から必要なものを選択し、組み合わせて完成したプログ ラムを生成する方式がジェネレータである。

RPG をその例として処理内容を理解させる。

# 育成方法および指導上の留意点

プログラミング言語の概要をつかみ、その適用範囲を認識させることがここ での課題である。

したがって、演習はないが、次のことについてレポート形式で答えさせると よい。

- ・機械語、アセンブリ言語、高水準言語のそれぞれに対し、同じ問題を与 えたときの作成プログラムの量、作成の手間を比較させる
- ・言語処理プログラムでのエラーのチェックにより見つかる文法エラーと それでは見つからない論理的なエラーの経験的割合、両者のデバッグの 容易さを論じさせる

# 用語

機械語, アセンブリ言語, 高水準言語, 簡易言語, 手続き型言語, 非手続き型言語, 宣言型言語, 第4世代言語, 汎用言語, 応用向き言語, ソース・プログラム, オブジェクト・プログラム, アセンブラ, コンパイラ, インタプリタ, ジェネレータ

# 第3章 オペレーティング・システム

#### 学習目標

オペレーティング・システムの目的と構成、および主要機能について説明できる。

具体的なオペレーティング・システムをJCLなどの操作言語で指示することによって使用できる。

# 内 容

#### 3.1 オペレーティング・システムの目的(120分)

オペレーティング・システム(OS)は、ハードウェアと利用者との間にあって、以下のような目的を持った複合システムである。

- 1. ハードウェアの能力の有効な活用
- 2. 多様な処理形態への対応
- 3. 正確・安全なコンピュータ処理の実行
- 4. 利用者プログラムの負荷の軽減
- 5. コンピュータ操作やその運用の支援

OSの目的に関連して、OSがなくて裸のマシンをユーザーが使っていたら どんなに不便か、OSの機能をユーザがその都度いちいち書いていたら、開発 量は膨大になり、開発効率も悪くなることを容易にイメージさせることができ るであろう。

また、OSがなかったら、ユーザのプログラムにバグがあった場合、いちいち機械を止めて再度立ちあげ直さねばならない。このようにOSは、ユーザのプログラムを安全かつ効率よく実行させるための環境を提供しているということを十分に認識させる必要がある。

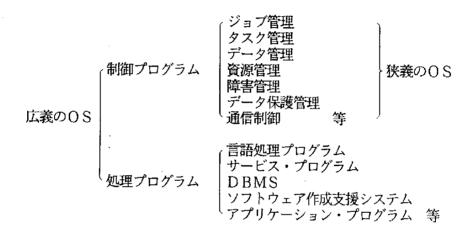
# 3.2 オペレーティング・システムの基本機能(240分)

#### (1) 構成

オペレーティング・システムは図表3-3-1に示すように制御プログラムと処理プログラムを含んだ広い概念である。

しかし、狭義には制御プログラムのことをOSと呼ぶ。

図表3-3-1 オペレーティング・システムの構成



# (2) 制御プログラムの機能

制御プログラムの機能を次の各項目に沿って説明する。

- 1. ジョブ管理
- 2. タスク管理
- 3. データ管理
- 4. 資源管理
- 5. 障害管理
- 6. データ保護管理
- 7. 通信制御
- 8. 運用管理

ここではそれぞれの機能がOSを使用するユーザおよびユーザの作成したプログラムとどのようなかかわりを持ってくるかを説明する。具体的な

JCL (Job Control Language) の例をあげ、それぞれの機能はこの言語のどの部分に現れるのかを関係づける。

#### 3.3 実OSの機能と利用実習(240分)

- (1) OSの種類
  - OSの実行形態として次のような種類がある。
    - ・バッチ処理システム(MVS など)
    - オンライン処理システム
    - リアルタイム処理システム
    - ・タイムシェアリング・システム(UNIXなど)

これらの実行形態の違いをOSを利用する立場から説明し、それぞれどのような特徴があるか、要求される性能はどの程度かを理解させる。

#### (2) 演 習

代表的なOS(UNIXとかMVS)を1つ選択し、その操作を覚えることを目標にする。OSに対する指令をJCLやコマンドの形で与える。JCLやコマンドの機能を説明し、実際に課題を与えて生徒に操作させる。

# 育成方法および指導上の留意点

OSの存在はふだんユーザの目に見えないところで動いているため、顕著には現れてこない。ユーザのプログラムの下で動いており、ハードウェアとのインタフェースを司る部分であることを理解させなければならない。そして、どういう働きをしているのか、積極的に調べてみるような気にさせるところまでいけば、しめたものである。

OSでどれだけの処理がされているかを目で見えるような形にするには、実行時間をはかるのがよい。あるコマンド(言語処理プログラムなど)を起動し、そのユーザレベルでの実行時間とシステムでの実行時間をはかる。この値から、ユーザ・プログラムの実行にどの程度OSが関わってくるかということを理解させる。

ファイルを作成したり、ファイルの使用領域を確保したりは、OS固有の機

能に依存するが、これらをJCLやコマンドを入力することで実際に体験させ、 利用のしかたを覚えさせることは、プログラムを開発していく上で重要であり、 開発効率を高める結果となるので十分に演習させる。

#### 用語

オペレーティング・システム,制御プログラム,処理プログラム,ジョブ管理,タスク管理,データ管理,資源管理,障害管理,運用管理,データ保護,ジョブ制御言語(JCL),コマンド,バッチ処理,オンライン処理,リアルタイム処理,タイムシェアリング・システム(TSS)

#### 第4章 サービス・プログラム

#### 学習目標

代表的なサービス・プログラムの例をあげ、それぞれの役割を説明すること ができる。

サービス・プログラムの例として、テキスト・エディタの使い方を覚え、実際にテキストの入力や更新などの編集ができる。

# 内容

#### 4.1 サービス・プログラムの目的と課題(120分)

(1) サービス・プログラムの種類

サービス・プログラムとは、制御プログラムや言語処理プログラムと連携をとりながら、プログラムの実行や大量データの操作などにおいて共通のサービス機能を提供するプログラム群である。ユーティリティ・プログラムと呼ばれることもある。

JISでは「サービス・プログラムとは、コンピュータによる処理を、 一般的に支援するプログラム」と定義している。

次のような代表的なサービス・プログラムの例をあげ、その機能を説明する。

- リンケージ・エディタ 複数のオブジェクト・モジュールを組み合わせて1つのロード・モ ジュールに編集する。
- 2. ローダ プログラムの実行に先立ってロード・モジュール内の相対アドレス を実アドレスに変換し、メモリ内にロードする。

3. テキスト・エディタ

プログラムや他のテキスト形式のファイルの入力や更新を含む編集 機能を持つ。

文字単位、行単位の編集のほかに、プログラムの構造に対応した編 集機能を持つ、構造エディタがある。

4. 整列(ソート)・併合(マージ)プログラム

一連のデータのグループ(ファイル)を、あるデータ項目について、数の大きさの順、アルファベット順などに配列し直すのが整列であり、整列されている2つ以上のデータ・グループを1つの順序正しい配列に統合するのが併合である。

整列プログラムのアルゴリズムには挿入法、交換法、クイックソ・ ートなどがあり、それぞれの特徴を説明し、実際にそれらのプログラムを実行してみせ、実行時間をはかることにより、アルゴリズムによって効率が違うことを理解させる。

- ファイル媒体変換プログラム
   異なる形式のファイルの間でデータを変換するプログラムである。
- 6. ライブラリ管理プログラム

ユーザの作ったコードをユーザ・ライブラリに登録したり、削除したりするプログラムである。ソースコードをいれるソースライブラリと、オブジェクト・コードをいれるオブジェクト・ライブラリがある。

7. システム生成プログラム 利用者ごとに最適なOS機能を編集し、システムを再構築するプログラムである。

(2) OSによる使い方の違い

サービス・プログラムは、基本ソフトウェアであり、OSに付属しているものがほとんどである。したがって、同じ機能を持っていてもその使い方は、OSによって異なるのが普通である。

#### 4.2 テキスト・エディタとその利用演習(120分)

#### (1) テキスト・エディタの機能

テキスト・エディタには、文字単位のデータ編集を行う文字エディタと 行単位の編集を行う行エディタがある。一般に初心者は文字エディタ、そ れもCRT画面で見ながら編集できる、画面エディタ(スクリーン・エディ ィタ)が使いやすい。

テキスト・エディタの基本的な機能を次の各項目に沿って説明する。

- 1. 文字・文字列の入力
- 2. 文字・文字列の削除
- 3. 文字・文字列の置換
- 4. 文字・文字列のコピー
- 5. ファイルへの書き込み

#### (2) 演習

市販されているパソコン上のスクリーン・エディタを使用し、プログラムテキストを作成させることによってスクリーンエディタの機能を理解させ、同時にその使い方を覚えさせる。

# 育成方法および指導上の留意点

サービス・プログラムの種類は多く、その用途もまちまちである。しかも、 現在サービス・プログラムにないものが将来入ってくるかもしれない。MMI (Man Machine Interface )の向上はめざましく、より便利になったサービス ・プログラムが取り込まれることはまちがいない。

ここでは、そのことを体験させるため、演習でテキスト・エディタを使用させる。エディタとしては市販のもっとも新しいものを使用するとよい。テキストひとつ作るにも、MMIの良否が重要であることが体験できるであろう。過去の経験にとらわれることなく、新しくてよいものを選び、積極的に利用していく態度を養うことも育成の目的の一つである。

演習にあたっては、できるだけ機能の説明は基本的なものに絞り、あとは、 質問がでたときに説明するぐらいで、生徒自身に探究させるほうがよい。 用語

サービス・プログラム、ユーティリティ・プログラム、リンケージ・エディタ、ローダ、テキスト・エディタ、整列・併合プログラム、整列アルゴリズム

3 - 25

#### 第5章 ソフトウェアの課題

#### 学習目標

ソフトウェアの持つ価値に対しての期待感がますます強くなってきている。 その期待に応えるためにソフトウェアの開発や利用での課題について実務の場 で指摘できるようになる。

この章では古くて新しい「品質」についての理解を深めてもらう。具体的にはソフトウェアの品質とは何か、品質の作り込みとは何をするのか、それを遂行するにはどんな方法が有るのかなどが列挙できる。

また、高品質なソフトウェアを構築するために必要な環境はどんな状態・状況を指すのかなどが列挙できるようになる。

品質を保ちつつ、生産性を向上する再利用技術や共通ソフトウェアの活用方法についても理解する。

ソフトウェアは「目に見えない」ものを制作、利用するので問題発見から解 決まで、それに関わる担当者の気構えが「品質」に大きく影響することが認識 できる。

# 内 容

# 5.1 生産性と品質の向上 (90分)

ソフトウェアは工業製品のように個々の製造中のバラツキや経年摩耗という事はない。不特定多数が利用するソフトウェア(パッケージ・ソフトウェア)と特定ユーザのために作られたソフトウェア(カスタマーズ・ソフトウェア)も利用者に渡る前では原本唯一の製品である。パッケージ・ソフトウェアを多数の利用者に頒布する場合は、これを電子的に複写して渡すことになる。

従って、原本のもっている性質(特に誤りの部分)が、そのまま利用者に渡ることになる。

また、工業製品では使ってゆくうちに部分的に壊れたり、摩耗して使えなくなる状態になる。ソフトウェアでは、このような事は無い。代わりにハードウ

ェアの進歩、ソフトウェア開発環境の整備により、ソフトウェアの価値が陳腐 化する。 例えば、利用者とのデータ入出力で使う画面制御ソフトウェアをみ てみる。昔は1行単位で制御していたが、現在は文字単位で制御できる。ソフトウェアの機能そのものに誤りはないが、ハードウェアの進歩を取り込めない ので陳腐化を免れない。また、ソフトウェア機能も利用者に便利なものに変わってきた。

この様に技術の進歩を前提にして、ソフトウェアを作り、利用するにはその ソフトウェアのもつ価値を十分理解する必要がある。その価値の第1条件は品 質であることを認識させる。

#### (1) 品質の定義

品質とは何であろうか。特にソフトウェアの品質とは。日本流に言えば品質とは「お客様の満足度」であろうか。

規格書では次の様に言っている。

・日本工業規格(JIS)

「品質とは、品質またはサービスが使用目的を満たしているかどうかを決定するための評価の対象となる固有の性質・性能の全体」

・アメリカの規格(IEEE/ANSI)

「ソフトウェア品質とは、与えられたニーズを満足させる能力に影響 を与える、ソフトウェア製品の総合的な特徴・特性である」

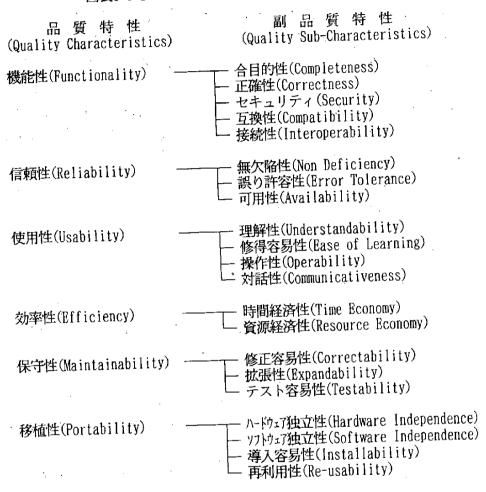
どちらも全体とか総合的なという言葉で性質・特徴、性能・特性をとらえている。

その特徴・特性とは具体的にはなにを指すのか。国際標準では図表3-5-1 で示す6項目を品質特性とその副品質特性としている。

総合的とは、個々の特性だけに注目するだけでなく、他の特性とも関連を もっている事をさしている。

この中で「機能性、信頼性、効率性」は古くから言われおり、ソフトウェ アの本質のようになっている。残り3つは少しないがしろにされてきている。

# 図表3-5-1ソフトウェア品質特性・副品質特性



使用性はMM I (Man-Machine-Interface )を含め、重要な課題となってくる。操作マニュアルの分かり易さ、操作の使い易さ、トラブル発生時のソフトウェア自身による回復処理が要求される。

保守性は信頼性や効率性に関連していて、各種のトラブルに対応する体制、 応答時間、担当者の教育と巾広い課題を抱えている。保守とは、単にトラブ ルを解決するだけが重要ではない。トラブル解決を通して、そのソフトウェ アの進歩に関与している点を認識させる。

移植性とはハードウェア・アーキテクチュアやオペレーティング・システ

ム(OS)が変わった場合に一部の修正で新しいシステム環境に移すことができるか否か。移植時、ソフトウェア全体を見直して変更部分を出すときアチコチを変更するのでなく、ある固まり(モジュール)単位でできるか否かが問われる。

以上のように品質は6項目の全てに渡って要求されていて、機能・効率性 だけの追求では不十分である事を銘記させる。

品質をバランス良く向上する事は難しいが、品質の追求が全体のコスト低減に役立つ事も認識させる。

#### (2) 工程と品質の作り込み

では品質をソフトウェア開発の、どの箇所、タイミングで評価するのであ ろうか。欧米では製品が完成した時点で検査を行い評価を下している。

日本の工業品製造では1970年前後から工程で品質を作り込む手法が提唱されてきた。これと似た考えをソフトウェアの品質に適用している。

ソフトウェアの問題を要求から解決するまでの方法は幾つかあるが、次の 2つが代表的な技法である。

#### a. 局面化技法

# b. プロトタイピング技法

局面化技法とは問題提起から解決までを計画、要求定義、外部設計、内部設計、開発実施、システム・テスト、導入・移行、保守のように工程に分けて対処する方法である。プロトタイピング技法は試作品をもとに利用者と情報交換をしながら不具合を改良して要求に応えてゆくやり方である。

この工程毎に先の6特性を意識して「誤りのないもの」、「より良いもの」 を達成して、次の工程に渡して行く。デザイン・レビューで利用者の了解や 要求定義との適合性の判定を行なうのが確実な方法である。

このように良い計画、設計、工程を通して作り込んでいかなければならないとする考え方が必要である。

#### (3) 再利用技術

ソフトウェアの品質を落とす事なく生産性を上げる各種の手法・技術が提案されている。しかし、一長一短があり決定打がないのが事実である。

システムやソフトウェアの仕事は全く新しい開発であるかと言えば、その

実態の大部分は機能追加もしくは改善である。この事実に注目して既成のソフトウェアを再利用する技術が生産性と品質の向上に役立つ事を意識させる。 生産性は次の式で定義する事ができる。

=品質(価値生産性)×増幅率×開発効率

ここで、品質はソフトウェアの価値である。増幅率は既存ソフトウェアを どれだけ利用したかによって決まる。この再利用率が多ければ多いほど生産 性は向上する。

ただし、既成ソフトウェアを何の規制もなく収集して使わせる事ではない。 再利用する事を予め計画しておいてソフトウェアを開発することが重要であ る。その際、考慮する点は以下のとおりである。

- ① データ構造、機能、制限が明文化されていること
- ② それが一定の形式で表現されていること
- ③ そのソフトウェアはテストされ、品質が保証されていること

開発効率は新規に作成する部分の効率である。短い期間、少ない人・費用で多くの開発量を生み出す割合である。

再利用は「作った振りをして、ソフトウェアを完成する」と言っても良く、 新規に作る部分が少なければ、人手・時間も少なくて済む。しかし、実効あ るものとするには解決すべき点が多いことも説明する。

# (4) 文書化、可視化

我々はソフトウェアを直接見ることはできない。できないが、それを作ったり、利用している。作ったり、利用する際にソフトウェアが持つべき機能や使い方などの情報を我々に分かるものに変換する必要がある。これを文字や図表で表わし、文書として情報交換をする。

ソフトウェアの問題を要求から解決するまでの方法の内、局面化技法とプロトタイピング技法での文書について考えてみる。

局面化技法では工程毎に、その工程での成果を文書化している。先の工程

分けからは、次の文書が考えられる。

- ・計画書
- ・要求定義書
- 外部設計書
- 内部設計書
- ・プログラム・リスト
- ・検査仕様・成績書
- ・運用・保守書
- ・操作書など

プロトタイピング技法では利用者からの最初の要求や不具合点の指摘に対しては要求定義書、外部設計書、それを試作品に反映する際は内部設計書に 相当する文書を起こしている。

これらの文書で大事なのは、次の点である。

- ① 関連する事項が洩れなく正確に、かつ過不足なく記述されていること
- ② 文書化基準や様式指定で、個人差が出ない工夫がされていること 操作書やユーザ・マニュアルでは上記以外に、分かり易く、誤解されないことも大事な要点となる。

このように、多くの文書でもって利用者と開発者、開発者と開発者間の情報交換を行っている。

ここで注意する事は、開発は要求定義(上流)から運用・保守(下流)への方向がある。上流から下流への流ればかりで開発は進まない。逆の場合もあり、この時は内部設計での変更が外部設計や要求定義に矛盾が起きないかチェックする必要がある。

もう一つ大事な事がある。それは、工程が順調に進展しているか否かが判る仕組みである。「計画通りです。」という事を裏付ける仕組みが必要となってくる。その仕組みとは、進捗状況を「見える物」に置き換える、即ち可視化する事である。

そのための第一歩は、いま担当している仕事のデータをとることから始める。 例えば、プログラム完成までの総時間数、仕様書の枚数、仕様決定にかかった時間、モジュール数、ステップ数、デバッグ回数・時間、異常箇所

の原因・対策などを記録する。この記録が、次の仕事の計画・見積をする際 の有効な情報となる。この記録をうまく文書化するのも大切である。

このように、ソフトウェアは文書の集合体である、文書作成力がソフトウェア品質を決めると言っても過言ではない。

#### 5.2 開発環境と開発支援ツール(60分)

ソフトウェア開発は膨大な量の知識を必要とする高度な頭脳作業である。 しかも、この作業には品質向上とともに生産の効率アップという事も含んでいる。

品質の重要性については前節で述べてあるので、この節では品質の確保と 生産を裏付ける環境、支援ツールについての理解を深めてもらう。

#### (1)環境と人間

品質・生産性を追求する重要なファクタの一つは人間である。人間は環境の動物であると言われている、物理的な環境も大事な要素である。それ以上に大切な、その環境を使う開発者・利用者自身の動機付けやチーム作業する場合の人間的側面について理解してもらう。

良いメンバ、良い作業環境、良い職場が揃って始めて「良い製品」が出来る。ソフトウェアの開発を一人で行う事は希である。一般的には開発チームで行うようになる。その際、チームに特有なニーズを対象にした専用の支援システムが近年発表・発売されている。これををグループウェアと呼んでいる。

グループウェアのための具体的なツールとして電子会議ソフトウェア、ワークグループ管理ソフトウェア、共同執筆ソフトウェアなどがある。これらは担当者同士の意思疎通・情報交換を行うツールである。このグループウェアで大切な要件は、このツールを使い活用するのは人間である、という点である。

チームで作業するという行為そのものがメンバ全員の潜在能力を発揮できる考えれるからである。

#### (2) 支援ツールの機能と種類

開発支援ツールの目的は開発中のソフトウェアの品質を平準化する以外に、

次の3点も重要である。

- ① 開発量を減らす
- ② 開発期間を短縮する
- ③ 開発中の情報を残す

この目的を達成するための環境として以下のような要素が考えられる。

- (1) 各種の工程に見合った開発支援システムやツール
- ② 操作性に優れたワークステーション
- ③ 利用者同士を有機的に繋ぐ、ネットワーク
- ④ 対話型で各種の支援ツールを管理・制御できるオペレーティング・システム

このように、ソフトウェア開発支援ツールとはソフトウェア開発の作業を 支援するコンピュータを使ったシステムであり、要求定義から設計、プログ ラミング、テスト、保守など全ての工程に対する多くのツールがある。

近年は工程毎のツールだけでなく、要求定義からプログラミングまでを一貫して開発を助けるシステムが発表されている。これをCASE (Computer Aided Software Engineering) と呼んでいる。

CASEを理解するには、構造化分析、構造化設計、DFD (Data Flow Diagram ) などの技術・手法の知識が必要である。

また、グループウェアで使う支援ツールも徐々に整備されてきている。 図表3-5-2にソフトウェア開発支援ツールの分類と例を示す。

#### 図表3-5-2 ソフトウェア開発支援ツールの例

# □要求定義要求定義支援システム要求仕様記述言語

- 設 計システム構成設計支援システムソフトウェア設計支援システム
- □プログラミング 各種高水準プログラム言語 構造化プログラミング・ツール プログラム編集ツール
- □テスト テストデータ・ジェネレータ デバック・ツール カバレージ・テスト
- □保 守 プログラム分析ツール データ分析ツール 修正波及分析ツール 回帰テストツール
- □文書化 ドキュメント作成支援ツール ドキュメント変更支援ツール

#### 開発管理

見積り支援ツール 工程管理支援ツール システム検査支援ツール システム性能評価ツール

- □部品化・再利用 部品化支援ツール 部品連結言語 部品検索プログラム 部品デ-タヘース・マネシメント
- □グループウェア 電子会議ソフトウェア ワーククルーク管理ソフトウェア 共同執筆ソフトウェア

# 5.3 共通ソフトウェアの活用(30分)

共通ソフトウェアとは、基本ソフトウェア、応用ソフトウェアの全域にわたり、多数の利用者が共通に利用し得るよう整備したものである。独立したソフトウェアからプログラムの部分に至るまで種々のレベルがある。

#### (1)活用の利点

共通ソフトウェアの活用により次のような利点がある。

- ① ソフトウェアの生産性向上
- ② 良質の既存ソフトウェアを使うことによる品質向上
- ③ ソフトウェア全体としてプログラムコード容量が少なくて済む

#### ④ ソフトウェアの価値評価が高まる

#### (2) 利用面からみた種類

共通ソフトウェアは、その利点を追求できるものであれば身近に存在している。一番小さい共通ソフトウェアは開発者自身が経験を重ねる内に集めたものを、別の開発時に利用する。次にくるものとして、開発チーム内で共通化を図り、利用する。

大きなものとしては、ハードウェア、ソフトウェアの実行やファイル管理・制御を共通化して利用者の便に共する目的のものもあり、基本ソフトウェアのなかに組み込まれている。

基本ソフトウェアには標準でないが、分野特有の共通ソフトウェア群を 作り、これらを組み合わせて使えるようにした各種ライブラリもある。

一番大きい共通ソフトウェアは業務分野別に開発されて、その業務関係者 がそのままで利用する。通常、このソフトウェアは応用ソフトウェアと言わ れ商品化されている。

図表3-5-3は基本ソフトウェア、分野別ライブラリと応用ソフトウェアなどにある共通ソフトウェアの例である。

図表3-5-3 共通ソフトウェアの例

基本ソフトウェア	<ul> <li>制御プログラム ファイル管理 通信制御 アクセス制御 システム運用 ・言語処理プロセッサ ・サービス・ラム・データ変換 整列・ヴラム プログラム・データ変換 整列・ブラリ管理 テキスト フードプロセッサ ・データベース管理 ・データーでエア開発支援</li> </ul>
分野別	・数値処理用ライブラリ ・シミュレーション・ライブラリ ・コンピュータ・グラフィックス・ライブラリ ・画像処理ライブラリ ・ウインドウ・ライブラリ ・日本語処理
応用ソフトウェア	<ul> <li>・共通応用プログラム 統計計画 意思決定支援 予測ション 工程管検理 「情報を表現のである。</li> <li>・業の人で、人のである。</li> <li>・業の会別である。</li> <li>・業の会別である。</li> <li>・業の会別である。</li> <li>・業の会別である。</li> <li>・業のである。</li> <li>・業のである。</li> <li>・業のである。</li> <li>・業のである。</li> <li>・業のである。</li> <li>・まれて、</li> <li>・まれては、</li> <li>・まれては、</li> <li>・まれては、</li> <li>・まれては、</li> <li>・まれては、&lt;</li></ul>

共通ソフトウェアの中で、良く使われているものとしてOS(MS-DOS, UNI-X、OS/2など)のもっているコマンド群がある。これらはシェル(Shell)・ツールと呼ばれ、個々あるいは組合わせて使える。

#### (3)活用の留意点

共通ソフトウェアを活用するには、そのソフトウェアを作る時と実際に利用する場合の両方で幾つかの留意点がある。

それは以下のような点をあげることができる。

- ① 信頼性、効率性、精度など総合品質に優れていること
- ② 利用時のインタフェースが簡単で使用が容易であること ソフトウェアの名前、入出力データの渡し方など
- ③ 誤りに対しての指摘、警告が出されること
- ④ 記憶スペースやデータ構造が明確になっており、その量が少ないこと
- ⑤ 分野毎(数値解析用、画面制御用など)にライブラリ化してある場合 の使い方を含めて管理・保守が簡単なこと
- ⑥ パッケージの場合は一部の機能のオプション化や選択性が可能なよう にモジュール化されていること
- ⑦ 必要十分な説明が、分かり易く記述してあるマニュアルなどの文書があること

ただし、次のような場合もあるので注意が必要である。

- ① 必ずしも万能ではなく精度や制限によっては使えない場合
- ② 使用環境(計算機、OS、言語など)との適合性が良くない場合 共通ソフトウェアを作る・利用するにも計画や調査が必要である。品質管 理の中で使われるPDCA(Plan-Do-Check-Action)は、5章全体の課題で あることを認識させる。

# 育成方法および指導上の留意点

ソフトウェアに限らず「品質」と「生産性」の向上は永遠のテーマである。 初級から、これらを取り上げた理由は最初に手がけたやり方が将来にも影響を 及ぼすと思えるからである。

ソフトウェアの開発は、顧客の言われるままに「何か」実行できるものを納 期までに作れば良いという面が強調されている。確かに、開発期間も短く、担 当者も少ない現実を避けては通れない。納期を守る為に目先の生産・開発効率 が先行してしまい、品質への取り組みが疎かになっている。その結果、費用効果が悪くなり、真の生産向上にはならない。それ故に「品質」を根底とした「 生産」が重要になる事を認識させる。

特に品質追求がコスト低減に結びつくという下りは5章全体の関連のなかから生まれてくるものであるので、指導は難しいものである。これは、納期は間に合ったが、経費的には失敗した開発例と反対に納期、経費とも充分に余裕があった成功例を引いて、何処に差があったかなどグループ討論させる。

品質、再利用技術には、品質表、TQC、部品化のレベルなど多種多様の道 具や手法がある。それらを知識として教える事も大切である。

品質や再利用、共通ソフトウェアの有用性の本質が理解できたら、実務で使う様式、書式を提示して実際に書く演習をする。この積み重ねが、良い品質への最短距離であることを教える。

#### 用語

品質、作り込み、生産性、共通ソフトウェア、パッケージ・ソフトウェア、カスタマーズ・ソフトウェア、評価特性、機能性、信頼性、使用性、効率性、保守性、移植性、MMI、モジュール、局面化技法、プロトタイピング技法、工程デザイン・レビュー、再利用技術、増幅率、開発効率、文書化、可視化、開発環境開発支援ツール、グループウェア、CASE、インタフェース、品質表、TQC、部品化

#### 推薦教材

「マイコンのためのオペレーティングシステム入門」 大原 茂之・倉田了一 著. オーム社, 1981年

### インストラクタ必読書

- (1) 「プログラミング言語処理学」 佐佐 政孝 著、岩波講座ソフトウェア科学 5. 1989年
- (2) 「UNIX」 石田 晴久 著, 共立出版, 1983年
- (3) 「ソフトウェアの生産技法」 菅野 文友 著, 日科技連刊, 1987年
- (4) 「ソフトウェアの品質管理」 菅野 文友 著, 日科技連刊, 1987年
- (5) 「CASE」

Carma McClure 著,三井銀総合研究所訳,日経BP社,1990年

# 参考文献

- (1) 「コンヒュータサイエンス大学講座 アセンブラプログラミング入門―IBM/370のための ―」 金山 裕 著, 近代科学社, 1977年
- (2) 「ソフトウェアの生産技法」 菅野 文友 著, 日科技連刊, 1987年
- (3) 「ソフトウェアの品質管理」 菅野 文友 著, 日科技連刊, 1987年
- (4) 「グループウェア」
  Robert Johanson 著, 会津 泉訳, 日経BP社, 1990年
- (5) 「ピープルウェア」 Tom DeMarco, Timothy Lister 共著 日立ソフトウュアエンシニアリング (株) 生産性研究会訳, 日経BP社, 1990年
- (6) 「品質機能展開による高品質ソフトウェアの開発手法〈解説編〉」 情報処理振興事業協会技術センター 編
- (7) 「情報処理技術者のためのプレゼンテーション技法」 三浦大亮 著,通産資料調査会刊, 1987年

- (8) 「CASE」 Carma McClure 著,三井銀総合研究所訳,日経BP社,1990年
- (9) 「プログラミング言語処理系」 佐佐 政孝 著, 岩波書店, 1989年
- (10)「オペレーティング・システム」 前川 守 著,岩波書店,1988年

#### 第4部 通信ネットワーク

#### 教育目標

情報処理システムの基本要素は、ハードウェア、ソフトウェア、データベース、および通信ネットワークである。このうち通信ネットワークに関する知識は、他の3つに比べて必ずしも十分でない情報処理技術者が従来は多かった。

しかしながら現在、情報処理と通信は完全に融合し、通信ネットワークに関する知識・技術は、情報処理システムを構築し運用する全ての情報処理技術者にとって必要不可欠なものとなった。

本部では、初級情報処理技術者と言えども、日常の業務において通信ネットワークを利用し、あるいはその構築に参加する立場として理解しておくべき通信ネットワークの基本的知識の修得を目指すとともに、ネットワーク社会といわれる情報化の現状を正しく認識しうるプロの技術者としての常識を得ることを目標とする。

# 内容の構成

	rin viz	時間数	
	内	講義	演習
第1章 1.1 1.2 1.3	通信ネットワークの役割 通信ネットワークの歴史 通信ネットワークの機能 通信ネットワークの利用	6	. 0
	通信ネットワークの基礎技術 データ伝送の基礎 ネットワークの基礎	1 6	6
3. 1	ネットワーク・アーキテクチャ ネットワーク・アーキテクチャの概念とその意義 ネットワーク・アーキテクチャの標準化 OSIのプロトコル階層	8	3
4. 1	ネットワーク・サービス ネットワーク・サーヒスと電気通信事業法 主なサービス ネットワークを用いたサービス LAN製品	9	2
5. 1 5. 2	ネットワーク・セキュリティ ネットワーク・セキュリティとは ネットワーク・システムの信頼性対策 ネットワーク・システムの安全性対策 ネットワークの種類とセキュリティ	5	0
	合 計	4 4	1 1

#### 第1章 通信ネットワークの役割

#### 学習目標

第4部の導入部として、通信ネットワークの全体像をまず把握させる。

とくに通信ネットワークは、ハードウェア、ソフトウェア、データベースと並んで、情報処理システムの基本的構成要素の一つであること、また当初は全く異なる発生起源を持つ通信技術と情報処理技術が、今日のように完全に融合するに至った経緯を通して、ネットワークの役割や機能のあらましを理解させる。

#### 内容

- 1.1 通信ネットワークの歴史(120分)
  - (1)通信ネットワークとは 以下の点を中心に説明する。
    - ① 情報処理の基本的機能には、

情報の記憶

情報の演算

情報の制御

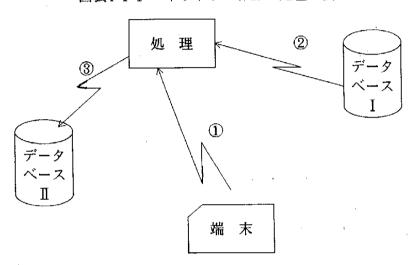
情報の移動

などがある。この中の情報の移動をやや離れた距離間で行なうために通 信回線を用いる。

- ② 通信ネットワークとは、複数地点間で通信回線がネットワーク状(網目状)に相互に接続された状態をいう。
- ⑧ 通信ネットワーク・システムは、ネットワーク化された通信機能を、 通信回線、ハードウェア、ソフトウェアで実現したシステムである。
- (2) オンライン・システムの出現 以下の点を中心に説明する。
  - ① 情報を遠距離に運ぶ通信回線を、情報処理システムの中にとり込むこ

とが可能となったことにより、かつてのようにコンピュータ・システムが存在する場所に行かなくとも、居ながらにして遠くのコンピュータへ情報を送ったり、処理を行なったりするオンライン・システムが可能となった。

例えば、図表4-1-1 に示すような例によりその効果を示す。



図表4-1-1 ネットワーク上の処理の例

- ① 処理の依頼
- ② データベース [ よりデータを入力して処理
- ③ 処理結果をデータベースⅡに出力

# (4) 情報処理と通信の融合

以下のような点を中心に説明する。

① 当初は、電話や電信を主たる目的として発達して来た通信技術を、コンピュータで扱うデータの通信に流用する形で進んで来たが、その後のオンライン・システムの普及により、情報処理技術と通信技術とが徐々に融合し、現在では通信ネットワークは完全に情報処理システムの一構成要素となった。

そして情報処理システムは、時間と距離の制約を克服したダイナミックな機能を持つものとして質的変化を遂げた。

② 情報処理と通信の融合を実現する代表的技術は、情報のディジタル化である。これは単に文字や数字のみならず、画像・図形・音声などもディジタル化することにより、これら多様な形態の情報が統合的に且つ効率的に通信できるようになった。

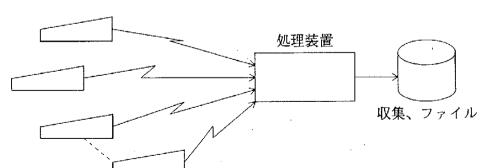
# 1.2 通信ネットワークの機能(180分)

- (1) 通信ネットワーク・システムの構成要素 通信ネットワーク・システムを構成する基本的な要素として以下のもの があることを説明する。
  - ① ネットワーク処理あるいはネットワーク上でのアプリケーション処理 を行なう処理装置(コンピュータ)
  - ② ネットワークを介して利用されるデータを蓄積したデータベースやファイル
  - ③ データの入出力やネットワーク上の処理を起動する端末機
  - ④ それらを相互に接続する通信回線
- (2) 機能の例

抽象的な説明ではなく、幾つかの具体例により代表的な機能を理解させる。

(例1) デーダ収集システム

気象データや、POSデータなど、多数のデータ発生源がある場合に、 それらのデータを集め分析・処理を行う。

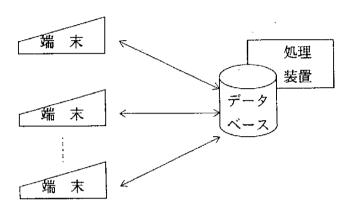


図表4-1-2 データ収集システムのイメージ

#### (例2) データ検索/問い合わせ応答システム

データベースから目的のデータを検索したり、各地の端末から商品の在 庫状態を問い合わせたりする。

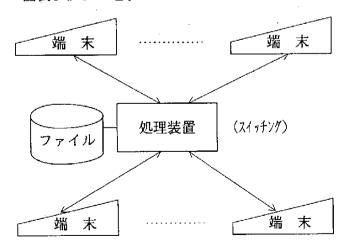
図表4-1-3 データベース検索/問い合わせ応答システムのイメージ



#### (例3) 電子メール・システム

端末相互の交信を行うシステムで、この場合、処理装置は、どの端末からどの端末にメールを送るかのスイッチングを行なう。またファイルはメールの内容を一時的に蓄請・保管する。

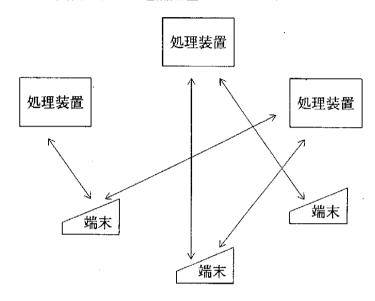
図表4-1-4 電子メール・システムのイメージ



#### (例4)遠隔処理システム

端末からコンピュータ・システムに処理の依頼を行なう。処理の即時性 を要求しないリモート・バッチ処理と、即時処理とがある。

図表4-1-5 遠隔処理システムのイメージ



#### 1.3 通信ネットワークの利用(120分)

ネットワークの利用目的は多様であるが、ここでは企業・組織内における利用と、企業・組織間における利用の2つの立場から、利用上の特徴を説明する。 各システム毎に以下の点を指摘する。

- ① システムの機能の概要と特徴
- ② ネットワーク化の効果
- ③ 同種システムの普及の現状
- (1) 企業・組織内ネットワーク・システム

(例1) 集中化されたデータベースを中心とした列車の座席予約システムなどを対象とし、ネットワークを通した集中データベース利用の効果に 焦点をあてる。

(例2) 販売・在庫管理システムなど、企業内分散処理システムを対象に、分散システムにおけるネットワークの効果に焦点をあてる。

- (例3) OA(Office Automation)ネットワークやFA(Factory Automation)ネットワークを対象に、LANシステムの効果に焦点をあてる。
- (2) 企業・組織間ネットワーク・システム
  - (例4) 銀行間の為替交換システムなどを対象に、メッセージ・スイッチング機能を中心としたネットワークの効果に焦点をあてる。
  - (例5) EDIシステムを対象に、複数企業間の新しいネットワークの 効果に焦点をあてる。

# 育成方法および指導上の留意点

- (1) 通信ネットワークに関する導入部として、技術的細部には立ち入らず、ネットワークの役割や効果の概要が把握できればよい。
- (2) ネットワーク化時代といわれるゆえんを認識させる。即ち、通信ネットワークは単に情報を遠くへ伝送するという機能を果たす媒体であるというだけでなく、情報処理システムに質的変化をもたらすとともに、ネットワーク化の進展によって、社会機構や産業構造の変化を引きおこす原動力となりつつあることなど、その重要性を認識させる。
- (3) 通信ネットワークの機能や利用の説明においては、出来る限り実例を示す と共に、VTRによる実写や、デモンストレーションなどが行われることが より望ましい。
- (4) ネットワーク・システム同志が相互に接続されることによって、ネットワーク化の効果が更に高まることを理解させる。

# 用 語

通信ネットワーク、通信ネットワーク・システム、オンライン・システム、オンライン・リアルタイム・システム、データ収集システム、データ検索システム、問い合わせ応答システム、電子メール・システム、スイッチング、遠隔処理システム、リモート・バッチ処理、分散処理システム、LAN、EDIシステム

#### 第2章 通信ネットワークの基礎技術

#### 学習目標

情報処理技術者にとって、通信技術がわかりにくい最大の原因の一つは、その 基礎を学ぶ機会を持たぬものが大多数であったことである。

通信技術はコンピュータ技術と共通点もあるが、その基礎の部分はその発生が全く異なる。本章における学習目標は、その技術ギャップを少しでも埋めることにある。

この章を修了した時点での学習目標は、以下のとおりである。

- 1. 種々の情報を通信回線を通して伝送するという基本的メカニズムを、十分 理解できる。
- 2. コンピュータと通信システムを組み合わせるために、どのような機能が基本的に必要かを説明できる。
- 3. 情報処理システムの一要素となった通信ネットワークの基本的機能を、十分把握する。
- 4. 通信技術に対する基本知識が、コンピュータ技術に対する基本知識と、ほぼ同等のレベルに引き上げられること。

# 内 容

#### 2.1 データ伝送の基礎(720分)

(1) データ伝送とは

銀行オンライン・システムや座席予約オンライン・システムなどのデータ 通信システムについて、その構成、目的、留意点について説明する。

- ① データ通信システムとデータ伝送システム データ通信(Data Communication)とデータ伝送(Data Transmission) の 違いを明らかにしておく。
  - (a) データ伝送システムとはデータを確実に伝送する機能を実現するもの であり、データ伝送システムにはその為の通信回線やモデム等の装置が

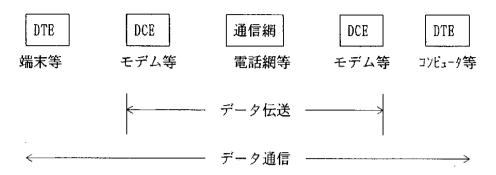
含まれる。

- (b) データ通信システムとはデータ伝送機能とデータ処理機能を合わせも つものであり、データ通信システムにはデータ伝送機器以外にコンピュ ータや端末などが含まれる。
- (c) 図表4-2-1をイメージして説明する。ただし、ここではシステム構成の概要について述べるに留め、詳細な説明をすることは避ける。また、 DCEやDTEなどの通信用語は、できるだけ使用しないで説明するのが望ましい。
- ② 通信回線利用に関する留意点

コンピュータや端末間の通信に通信回線を利用する場合に考慮しておか ねばならない事項について説明する。これらが伝送制御手順などの様々な きまりや通信を制御するための装置が必要な基本的理由であることを述べ る。

- (a) 通信回線は定速のベルト・コンベアのようなものであり、コンピュータや端末の都合にあわせてスピードを変えたり停止したりはできない。
- (b) 通信回線でデータを送る場合、データ化けやデータ落ちが起こること ことを覚悟せねばならない。
- (c) コンピュータや端末内部で使われている信号の形と通信回線上の信号 の形は異なるので、変換が必要である。
- (d) 通信回線は通常1ビットずつデータを運ぶので、データの送り側は回線のスピードにあわせてデータを分解し1ビットずつ送り出すとともに、 受取側では受け取ったビット列から正しく元のデータを復元する必要がある。

図表4-2-1 データ通信システムの構成



DTE:Data Terminal Equipment
 DCE:Data Circuit terminating Equipment

#### (2) 情報の表現と符号

データ伝送とは情報をデータとして、送受信者の間で受け渡すことである。情報を正しく伝送するには受け渡される情報の表現方法が送受信者間で共通でなければならない。情報表現を共通にするための符号についてその種類ならびにその表現できる情報量等について説明する。

# ① 情報の符号化と情報量

情報をどのような形で表現して伝送するのか、また情報の量の表し方 について述べる。

- (a) 情報は0と1の組み合わせで表現される。これを符号(コード)化という。
- (b) 符号化された情報の量はビット(bit)という単位で表す。100 ビットの情報とは、100個からなる0または1の組み合わせで表現で きる情報の量である。・

#### ② 符号の標準化

多くの装置間で情報交換を円滑に進めるためには、情報交換のための 符号は共通化しておくことが望ましい。このような標準化符号について 説明する。

(a) 符号の国際的な標準化はISOやCCITTで検討が進められ、7

単位情報コード「国際アルファベットNo.5」として制定された。

- (b) 7単位情報コードとは、1文字を7ビットで表現する方式であり、 128種の文字や記号を表現できるものである。この単位数が多いと表 現できる文字数は増えるが、伝送の効率は悪くなる。
- (c) わが国でも、国際勧告に準じて、7単位と8単位のコード体系(JIS X 0201)と2バイトを単位とする漢字用コード体系(JIS X 0208)が定めらており、目的に応じて使い分けできるようになっている。
- (d) パソコン・ネットなどでは、シフトJISなどJISコード体系以外のコード系も広く用いられている。このようなものを事実上の標準という意味でデファクト・スタンダードと呼ぶ。
- ③ アナログ情報の符号化

情報にはコンピュータ内のデータのようなディジタル情報と我々の声のようなアナログ情報がある。このようなアナログ情報をディジタル情報に符号化して送ることも多い。その方式について説明する。

- (a) アナログ信号の振幅を一定の時間間隔で読み出す(標本化)、その 時の振幅の値を2進数に変換(量子化)して伝送する。標本化のタイ ミングはシャノンの標本化定理によって決定できる。
- (b) 電話を例にとると、我々の声の帯域幅は4 KHz弱である。よって標本化は8 KHz (毎秒8,000回)でよく、その時の振幅の値を8 ビット (0 から255の値)で表すとしたら、1 秒あたり64 K ビットの情報量ということになる。
- ④ 情報の圧縮

情報量が多くなるとその伝送コストも膨大となる。このコストの削減 と伝送時間の縮小のため情報を圧縮して送ることが効果的である。この 圧縮の考え方についてラングレスやハフマンを例として説明する。

- (3) 伝送技術の基礎
  - ① 直列伝送と並列伝送

情報を伝送する方式は2つある。符号を構成するビットを並列に同時に送る並列伝送と1ビットずつ順番に送る直列伝送である。それぞれの方式と利害得失について説明する。

- (a) 一般のデータ伝送では直列伝送が用いられる。並列伝送は伝送時間 が短くて済むが、距離が長くなるとコストがかさむので、工場内など 狭い範囲で用いられることを述べる。
- (b) 直列伝送方式の場合、送られる符号の区切りかた、符号のビットの 送りだしかた(高位側・低位側)について送受信の両者で合意してお く必要があることを述べる。

### ② 伝送速度

情報を伝送する速度は、単位時間に伝送される情報の量で表す。その尺度である変調速度、データ信号速度について説明する。

- (a) 変調速度はアナログ伝送に対してのみ用いることを述べる。
- (b) 変調速度(単位はボー)とデータ信号速度(単位はbps)は、いつも同じ値を持つと誤解することも多いので、1変調で4つの値を表現できる例(2ビット分の情報)などあげ、誤解が生じないようにする。

(データ信号速度)=(1変調で表現できる状態数)×(変調速度)

(c) 演習としては、伝送情報の総量や伝送時間の要求を与えて、必要な 伝送速度を具体的に計算させるとよい。

### ③ 変調

アナログ伝送路を用いてディジタルデータを送る場合は変調を行う。 変調が必要な理由と代表的な変調方式(振幅、周波数、位相)について 説明する。

- (a) 各変調方式のメリット、デメリットを実現の容易性、ノイズに対する強さ、効率性などの点から対比させて説明するとよい。
- (b) 演習としては、ディジタル・データ列を与えて、各変調方式ではどのような波形になるかを示させるとよい。また、多位相変調の場合の変調速度とデータ信号速度の値を計算させるとよい。

# ④ 同期

送受信を正しく行うためには、データ送信側と受信側でタイミングを 合わせること、すなわち同期をとる必要がある。

(a) 同期にはビットごとのタイミングをとるビット同期と、符号の先頭

位置を知るためのブロック同期があることを述べる。

- (b) 同期をとる方式としては非同期式と同期式があることを述べる。
- (c) 非同期式は調歩式ともよばれ、文字の前後にスタート・ビットとエンド・ビット(1 ビット以上の場合もある)をつけ、文字ごとに同期をとる方式であることを述べる。これはビット同期とブロック同期の両方をかね、方式は簡単であるが、伝送効率が悪く(1 文字ごとに 2 ビット以上の冗長がある)低速向きである。
- (d) 同期式には文字同期式(SYN同期)とフレーム同期式があり、それぞれベーシックおよびHDLC手順とともに用いられる。同期式ではブロック同期はDTEで行い、ビット同期はDCEが受持ちDCEよりDTEに対してビット送受信のタイミングが通知される。同期式は、中高速の伝送に用いられる。ただし、ここで伝送制御手順やDTE、DCEについて詳しく説明する必要はない。
- (e) 同期という言葉が煩雑にでるので、各々の意味を正確に理解させておく必要がある。また同期方式に非同期式があるが、これも混乱を招く要因であるので要注意である。
- (f) 演習としては、文字列を与えて各同期方式で伝送上どのような信号 になるか記述させるとよい。

### ⑤ 多重化

1本の伝送路を複数の通信路で共有して効率化を図ることを多重化という。多重化の方式として、周波数分割多重化方式(FSM)と時分割 多重化方式(TDM)をあげ、その基本的な考え方を説明する。

#### ⑥ 誤り検出方式

データ伝送において、伝送誤りは確実に発生するといってよく、誤り の検出は不可欠な要素の一つである。誤り検出方式について説明する。

- (a) データ伝送における誤りの発生原因、発生率ならびに誤りパターンは、バースト誤りとランダム誤り分けられることを述べる。
- (b) 誤り検出方式として垂直パリティ(VRC)、水平パリティ(LRC)、群 計数チェック(CRC)をあげる。
- (c) VRC、LRCはともに用いられることが多い。VRCやLRCに

比べてCRCは信頼度が高いことを述べる。

- (d) VRCおよびLRCはパリティのつけかたに奇偶の2方式があること、CRCの場合は生成多項式に様々のものがあることを述べる。
- (e) 演習としては、文字列を与えてVRC、LRC、CRCを計算させるとよい。

#### (4) 通信方式

① 単向、半二重、全二重

通信方式にはデータの流れる方向によって単向、半二重、全二重に分けられる。各々の特徴と使われ方について説明する。

- (a) 半二重端末、全二重端末を使用させて実感させるとよい。
- ② 2線式と4線式

物理的な伝送路には、2線式と4線式がある。それぞれの方式と特徴 について説明する。

- (a) 伝送路は電気が流れるものであり、2本の通信媒体を対にして1本 の伝送路が構成される。すなわち、2線式は電車の線路にたとえると 単線であり、4線式は複線であることを述べる。
- (b) 全2重通信を行うには、基本的には4線式を使用するが、多重化の 技術で2線式で全2重通信は可能である。一般の電話線は2線式であ るが、この方法によって全2重を実現していることを述べる。
- (5) アナログ伝送とディジタル伝送

伝送網の種類には、電話網などアナログ網とDDXなどのディジタル網がある。これらの網を利用した伝送について、両方式を対比させながら説明する。

- ① アナログ伝送
  - (a) アナログとは、音声のように連続的に変化するものであり、どのような微細な変化にも対応できることを述べる。
  - (b) アナログ伝送ではノイズが侵入すると除去不可能であり、伝送距離 が長くなればなるほど劣化が激しくなり、デジタル網に比べて品質が 劣ることを述べる。
  - (c) ディジタル情報をアナログ網で伝送するときには、変調ならびに復

調が必要であることを述べる。

(d) アナログ網の伝送容量と関連が深い周波数帯域について説明する。 ちなみに電話網の場合は 0.3から3.4KHzであり、1.2 あるいは 2.4Kbps のモデムが現在多用されている。また、20Kbps位までの製品が出ている。

#### ② ディジタル伝送

- (a) ディジタルの語源は指(Digit)であり、離散的な量なので、アナログのように音声などを正確に表現することはできないことを述べる。
- (b) ディジタル網は伝送の途中で信号を再生することが可能であり、アナログ網に比べて高品質な伝送が可能であることを述べる。
- (c) ディジタル網だからと言って、計算機の出力をそのまま送るわけでなく伝送路を流れやすい形に変えたり、速度を調整したりして網に流すことを述べる。
- (d) ディジタル情報の取扱いやすさのゆえに、情報はますますディジタル化されていく傾向がある。しかし、アナログ網に比べてディジタル網が高級というわけではない。それぞれ長所、短所があり用途によって選択すべきものであることを述べる。

# (6) 有線通信と無線通信。

これまで伝送路の主体は有線であったが、技術の進歩と法規制の緩和に ともない無線を使うことも多くなっている。有線通信と無線通信の特徴に ついて説明する。

# ① 有線通信

- (a) 通信ケーブルを使用して行う通信であり、各種のハードウェア、ソフトウェアや手順などもこれを対象に発展してきた。電話をはじめ現在でも、ほとんどの公共的通信網は有線系であることを述べる。
- (b) 伝送媒体によって伝送容量は大きく左右されることを述べる。
- (c) ノイズには強いが、ケーブル線を敷設しなければいけないことを述べる。
- (d) 長い歴史があり、信頼性があり選択の幅が広いことを述べる。

# ② 無線通信

(a) 電波や光による通信である。衛星通信と地上無線通信とがあること

を述べる。

- (b) 無線通信の用途として次のような例があることをを示して説明する。
  - 多数の地点(利用者)に対する同報送信
  - ・自動車、船舶などの移動体通信
  - ・僻地、離島などへの通信
  - ・工場や倉庫内など配線がしにくい場所での通信
  - 有線回線のバックアップ用
  - ・特に衛星通信では外国など遠隔地との通信
- (c) 有線と違ってケーブルを張る必要がなく、広帯域での伝送が実現しやすい。しかし、盗聴の危険が強く、電波の場合を例に取れば電磁ノイズに弱いなど技術的にも今後に期待する面も多々あり、選択の幅もまだ狭い。また、利用が増加するとともに、周波数帯域の割当が大きな問題となっているなどを述べる。
- (d) 衛星を介した地球局間および無線局を介した端末間は、全て完全相 互接続のネットワークを構成していることを述べる。
- (e) 地球局から衛星へ(端末から無線局へ)の上り方向のチャネルと衛星から全ての地球局へ(無線局から全ての端末へ)の下り方向のチャネルとは異なった周波数が割当てられこと、上り方向のチャネルは複数の地球局(端末)が1つのチャネルを共有することを述べる。
- (f) 衛星通信は地上約3万6000キロにある衛星を介した通信であるため、 約0.27秒の遅延が生じることを述べる。

### (7) 通信媒体

伝送路に使用する媒体について用途、無中継で伝送できる距離、伝送容量、長所及び短所の観点から説明する。

- (a) 有線で使用される媒体としてペアケーブル、同軸ケーブル、光ケーブルを取り上げる。光ケーブルについては電磁ノイズの影響を受けないこと、コヒーレント通信技術などによりどんどん大容量化が進んでいることなどにより有線用の媒体としては急速にその利用が広がっているので重点を置いて説明する。
- (b) 無線用としてはミリ波や赤外線を取り上げる。これらは利用環境に大

きく左右されることを述べる。

(c) 通信容量と通信距離に関して数字をあげて述べるときには、最新の技術状況に十分留意する。とくに光ケーブルは変化が激しいので要注意である。

#### (8) 伝送制御手順

伝送制御とは送受信者間でデータを効率よく確実に伝送する為の制御機能であり、伝送制御手順とは伝送制御を行うための一連のルールである。 伝送制御の役割と代表的な伝送制御手順を取り上げ手順内容、特質、用途について説明する。

#### ① 伝送制御

- (a) 伝送制御はデータ伝送回線の接続・切断などの回線制御、送受信データのフロー・コントロールなどの同期制御、誤りデータの検出訂正などの誤り制御、データ・リンクを設定し手順に従ってデータ伝送を行うことからなることを述べる。
- (b) 伝送制御手順は伝送制御のうち、同期制御、誤り制御、手順に従ったデータ伝送について定めたものであることを述べる。

### ② 伝送制御手順

- (a) 伝送制御手順としては基本形伝送制御 (ベーシック)、ハイレベル・データ・リンク制御手順 (HDLC)、無手順の説明を行う。
- (b) 各手順について適応通信速度と伝送効率、制御符号あるいはコマンド、同期方式、誤り検出・訂正方法、メッセージ・フォーマット、データ伝送方式、伝送監視方式、透過モードの可否、データ・リンクの確立・解放方法について説明する。
- (c) データ・リンクの確立方法には概念的にコンテンションとポーリング・セレクティングがあることを述べる。
- (d) ベーシックの特徴としては、データは8ビットの整数倍、1データ ごとに応答確認が必要、伝送制御手順中に装置制御機能が混在するこ となどを述べる。
- (e) HDLCの特徴としては、どんなビットパターンも伝送可能、受信 側からの応答を待たずにデータを8個まで連続して送信可、誤り制御

が厳密(制御符号も誤り制御の対象)、計算機間伝送に向いていることなどを述べる。

- (f) 無手順の特徴としては、実現コストが低い、信頼度が低い、伝送効率は低いことなどを述べる。
- (g) データ・リンクの確立、データの伝送、応答の伝送、再送、データ・リンクの解放までを手順の流れ図を用いて説明するとよい(図表4-2-2参照)。
- (h) 各手順の機能や特徴の比較表を用いて説明するとよい。
- (i) ベーシック手順とHDLC手順ではブロック同期方式としてそれぞれSYN同期とフレーム同期が用いられることを述べる。

端末 ホスト ホスト 端末 ENQ 相手アトレス SNRM UA ACK j-9(1) ETB F-91 STK F-92 F-9(3) ACK j-94 ETB F-92 STK データ(5) P PR 5. F ACK DISC EOT ŪA.

図表4-2-2 基本形・HDLC

# 2.2 ネットワークの基礎(600分)

(1) 交換方式

通信回線の利用法には専用と共用がある。前者は通信を行う特定の利用者間で固定的な回線を設定するものであり、他の利用者はその回線を使用することはできない。後者は電話回線のような共用の回線を利用するものであり、通信を開始する前に通信する利用者間で伝送路を設定する必要がある。すなわち網制御機能が必要である。

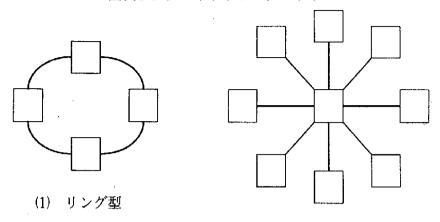
共用回線を利用してのデータ交換方式は回線交換方式と蓄積交換方式に に分けられる。各方式の特徴、適応性について説明する。

- (a) 回線交換方式は通信に先だって伝送路を設定し通信が終了するまでそれを保持する電話的な方式である。蓄積交換方式は伝送路は設定せずデータをバケツリレーの形で相手方に届けるものであり、手紙的なサービスであることを述べる。
- (b) 蓄積交換方式にはメッセージ単位で伝送するメッセージ交換方式とメッセージを交換網内で取扱い易いようにパケットに分解して送るパケット交換方式があり現在ではこれが主流であることを述べる。
- (c) NTTなどの通信事業者が提供しているアナログ交換網では回線交換のみがサービスされていることを述べる。
- (d) 通信量が多く規模の大きな会社では通信事業者から高速の専用回線を借りて、自社内で交換回線網を構築しているところも多いことを述べる。
- (e) 説明にあたっては、回線交換は電話的なサービス、蓄積交換は手紙的なサービスに例えると分かりやすい。すなわち前者の特徴はリアルタイム的であり、通信コストは距離と接続時間に比例し、ファイル転送など回線利用密度が濃い一括データ伝送処理に向く。後者の特徴は、時間遅れがあり、通信コストは伝送したデータ量に比例し、トランザクション処理などの回線利用密度の低い単発的に発生するデータ伝送処理に向く。また、後者の場合、手紙に書留などのサービスがあるように様々な付加的なサービスが実現できることも述べる。
- (2) ネットワーク・システムの形態 ネットワークは通信回線によってコンピュータや端末などの情報処理要

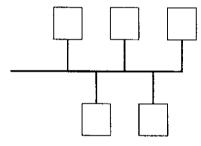
素が接続されることによって構成される。代表的な接続形態(トポロジー) を示し、その効率性、信頼性、経済性について説明する。

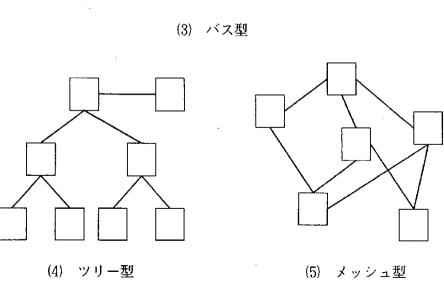
- ① 接続形態としてはポイント・ツー・ポイント、ツリー、リング、スター、バス、メッシュを説明する。最初のものは1対1接続であり、他は1対n接続である。(図表4-2-3参照)
- ② ポイント・ツー・ポイントおよびツリーはWANにおける典型的な構成形態である。リング、スター、バスはLANで用いられる典型的な形態であることを述べる。ただし、ここではWANおよびLANについて詳しく説明する必要はない。
- ③ これらのものが組み合わされて実際のネットワークが構成されていることを述べる。
- ④ 実際の回線接続形態の例として、WANを利用した典型的なオンライン・システムすなわち1つのコンピュータ・センタと1以上の端末から構成されるシステムの回線接続形態を取り上げその構成、特徴、適用性について説明する。
  - (a) 接続形態としてはポイント・ツー・ポイント、マルチ・ポイント ( 分岐方式)、集線方式を取り上げる。なおマルチ・ポイントには片方 向分岐と両方向分岐があることを述べる。
  - (b) ポイント・ツー・ポイントは端末とセンタ間の距離が短く通信量が 多い場合に適す。マルチ・ポイントは通信量が比較的少ない場合に適 する。集線方式は端末群とセンタ間が遠く離れている場合に適するこ とを述べる。

図表4-2-3 ネットワーク・トポロジー



(2) スター型

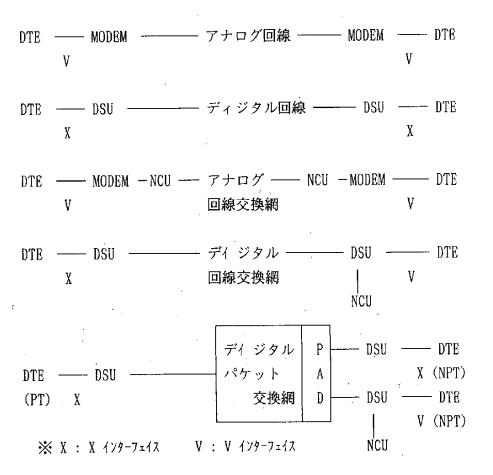




- (3) ネットワーク・システムの構成要素
  - ① ネットワークの種類と各種装置 図表4-2-4に示すようなデータ通信ネットワーク・システムの構成を 示し、各構成要素の役割を説明する。
    - (a) まず最初にDTE(Data Terminal Equipment)、DCE(Data Circuit terminal Equipment)、データ伝送回線からなる基本構成を示し、各構成要素の説明を行う。
      - ・ DTEとは具体的には端末装置やコンピュータ・システムである。 端末装置には端末制御装置があり、コンピュータ・システムは通信 制御装置を含むことを述べる。組み合わせとしては必ずしも一方が 端末装置で他方がコンピュータ・システムである必要はなく、両方 とも端末装置、両方ともコンピュータ・システムの場合もあること を説明する。
      - ・ DCEは実際には、伝送路がアナログ網の場合はモデム(MODEM) がディジタル網の場合はDSU(Digital Service Unit )が用いらることを述べる。
      - ・ 直列、並列、変調、同期など前節で示した内容とこれらの装置を 分かりやすく対応づける。
      - ・ DTE-DCE間は端末インタフェースと呼ばれ、ISOやCCITTで国際規格が定められている。端末インターフェスにはVシリーズとXシリーズがあり、前者はアナログ網用、後者はデイジタル網用であることを述べる。現在、モデムやパソコンの入出力インタフェースとして広く使われているRS-232Cについても言及する。これは、米国のEIA(Electronic Ind ustries Association)がCCITTV.24とV.28勧告に基づいて定めた端末インタフェース規格である。
    - (b) 次にNCU(Network Control Unit )を加えたものを示し交換網の場合は 網制御機能が必要なことを説明する。
      - ・ アナログ網 網制御のためにNCU(網制御装置)が必要 であるが、最近はNCU機能を内蔵するモデムを使うことが多いことを述べる。

- ・ ディジタル網の場合、ディジタル網用のDTE(Xンハ)ースの端末)の 場合は、網制御機能はDTE側で行うが、アナログ網用のDTE(V シハ)ースの端末)を接続する場合にはNCUが必要となることを述べる。
- ・ NCUはDCEとネットワーク回線の間に位置し、発着信動作の 手動・自動の別でMM型、MA型、AA型に分かれることを述べる。
- (C) 最後に、パッケト交換網の例を示し、DTEとしてはパッケト形態端末(通常はコンピュータ・システム)と非パケット形態端末(通常は端末装置)が あることを示し、両者の機能上の違いおよび後者を収容するために必要な網のPAD機能について説明する。

図表4-2-4 データ通信ネットワーク構成



#### ② 端 末

データ通信システムにおいて人間とのインタフェースの役割を担う端末装置の種類および用途について説明する。

- (a) 端末装置は用途別に分類すると情報伝送用、情報処理用に分けられる。情報処理用は、さらに汎用端末と専用端末に分けられることを述べる。
- (b) 情報伝送用とは、端末相互間でデータ伝送を行うことを主目的とする端末でありファクシミリ、テレライテイングなどがこれに当たることを述べる。
- (c) 専用端末は特定のアプリケーションに対応した端末であり、キャシュ・ディスペンサーなどの金融端末やPOS端末などがあることを述べる。
- (d) 汎用端末は一般端末と多機能端末に分けられることを述べる。前者は情報処理機能をほとんど持たないものであり、キーボードとディスプレイの最小構成から構成される。後者はパソコンやワークステーションなどを中心に構成され情報処理機能を持ち、以下のような機能を実現する。
  - 1) 中央のコンピュータから自立して情報を処理できる。
  - 2) ワープロ機能、図形処理機能、データベース機能、電子メール機能などを持ち、中央のコンピュータの負荷を軽減するとともに、協調して処理を逐行する。
  - 3) 画像、音声入出力装置などを接続し、マルチメデアを統合的に取り扱う。
  - 1)の一つの実現形態はマイクロ・メインフレーム・リンク(MML)であり、3)の一つ実現形態はAV端末と呼ばれるものである。
- (e) 端末はかってはワイヤード・ロジックで機能が実現されていたが、 今は大小はあるがほとんどのものがCPUを持ち、プログラマブルと なっていることを述べる。
- (f) 近年は、専用ワープロをはじめほとんどの情報処理機器が通信機能 を持つようになってきており、総端末化の傾向にある。またファック

シミリ、電話、パソコンなどの機能を持つものなど統合化も進んでおり、機能面での境界が不明瞭になりつつあることを述べる。

### (4) LAN

① LANEWAN

ネットワークをその地理的な広がりから分類すると、ワイド・エリア・ネットワーク(WAN)とローカル・エリア・ネットワーク(LAN)に分けられる。WANとLANについて実際のネットワークも引用しながら、特徴を比較、説明する。

- (a) 比較項目としては地理的範囲、伝送速度、伝送対象、費用、取扱い 易さなどを取り上げる。
- (b) WANはNTTをはじめとする通信事業者が提供する通信回線によるネットワークであり、LANは事業所内や大学構内などでの私設の回線によるネットワークである。
- (c) WANとLANは相互に接続される可能性が高い。例えば、企業システムでは各地の事業所や支店内のLANが相互にWANで接続されている。また大学間ネットワークでは各大学内のLANが相互にWANで接続される。
- (d) LANは、OAやFAの普及によって急速に利用が増大し、利用者にとってその高速性や装置接続の容易性など、その効用についてWAN以上に関心が持たれるようになっていることを述べる。
- (e) LANに対する一般的な要求は、次のようなものであることを説明 する。
  - ・コード化情報のみならず、画像や音声などの情報も伝送できること
  - ・1 Mbps以上の伝送速度を持つことが望ましい
  - ・コンピュータ、各種端末、ワークステーション、ファクシミリ、ワード・プロセッサ、電話機、各種センサ、AV機器など多様な機器が容易に接続できること
  - ・規模の拡張や変更が容易なこと
  - ・WANとの接続が容易であること
  - ・安価で信頼性が高いこと

- (f) LAN導入目的としては、資源の共有という観点のみが強調されが ちだが、配線の整理手段としての側面も大きいことを述べる。
- ② LANの技術的特徴

LANは、電話網などの従来の通信網を用いた通信とは制御手順などでかなり異なった技術的側面を持っているので、そのことを説明する。

- (a) LANのトポロジーとしてリング、スター、バス型があることを述べる。
- (b) LANにはベース・バンド方式とブロード・バンド方式がある。ただし、現在のところ主流はベース・バンド方式だからこれを主体に説明する。
  - (c) LANの通信制御方式として、CSMA/CD、トークン、TDM Aを取り上げ、トポロジーと絡めてその方式ならびに特徴を説明する。これらはOSIの第2層のデータ・リンク層に位置づけられメディア・アクセス制御(MAC)と呼ばれている。
  - (d) LANの伝送路を構成する媒体としてペアー線、同軸ケーブル、光 ケーブルが使われていることを述べる。
  - (e) LANの代表的なものとしては、10Mbpsの伝送速度を持つイーサネットが代表であるが、近年幹線用LANを実現するものとして光ファイバを用い100Mbpsを実現するFDDI(Fiber Distributed Data Interface)が普及しつつあることも述べる。
  - (f) FDDIと同様に国際標準化が進められているマルチ・メディアの 一つであるIVDLAN(音声・データ統合LAN)など最近の話題 にも触れる。
  - (g) LANについては、プリント・サーバ、ファイル・サーバなどを含んだシステム構成例をあげて説明するとよい。

# 育成方法および指導上の留意点

1. 細部には深入りせず、なぜそのような機能が必要であるかを納得させることをめざす。

- 2. 非常に多くの要素が含まれており、またそれらが互いに関係しているので 内容を重複させないで説明することは困難であるし、かえって分かりにくく する。よって、詳細は後述と言う形で重複説明の回避に神経質にならないよ う留意する。
- 3. 光媒体、ディジタル網およびLANは技術革新が激しいので、常に新しい 知識を蓄えるよう努める。
- 4. アナログ網とディジタル網の違いを的確に説明し、混乱させないようにする。
- 5. パソコンとモデムを利用して、パソコン間通信など簡単なプログラムを組ませたり、LAN環境があれば、その上で既存のプログラムを使わせるなどの実習が行われることが望ましい。

# 用 語

データ伝送、データ通信、DTE、DCE、モデム、DSU、NCU、ISO、CCITT、符号、bps、ディジタル、アナログ、標本化定理、変調、ボー、同期、VRC、LRC、CRC、単向、半2重、全2重、2線式、4線式、有線通信、無線通信、ペア・ケーブル、同軸ケーブル、光ケーブル、伝送制御手順、ベーシック、HDLC、無手順、専用回線、交換回線、回線交換、パケット交換、ネットワーク・トポロジー、Xインタフェース、Vインタフェース、RS-232C、LAN、WAN、MAC、OSI、CSMA/CD、トークン、TDMA、ベースバンド、ブロードバンド、端末、デファクト・スタンダード、情報の圧縮

### 第3章 ネットワーク・アーキテクチャ

### 学習目標

通信とは、発信側の意志を受信側に伝える手段であり、そのためには両者間の共通の言葉を始めとする事前の取り決めが必要不可欠である。

通信ネットワークを介して、コンピュータ同志、あるいは、端末同志が相互 交信を行うには、各種多様な取り決めが必要であり、その取り決めをできるだ け多くのコンピュータや装置間で共通なものとするための体系が、ネットワー ク・アーキテクチャの考え方である。

本章を修了した時点での学習目標は、以下の通りである。

- 1. ネットワーク・アーキテクチャの概念とその目的、その標準化の意義と 必要性などが十分理解できる。
- 2. ネットワーク・アーキテクチャの階層化の意義と効用を、具体的な例で 説明できる。
- 3. 各階層の通信プロトコルと、前章で学んだ基礎技術との関係付けを明確 にすることができる。

# 内 容

3.1 ネットワーク・アーキテクチャの概念とその意義(180分)

ネットワーク・アーキテクチャ出現の背景、考え方、種類について説明する。

(1) 背 景

ネットワーク・アーキテクチャ出現以前のデータ通信システムの作り方には、次のような特徴が見られたことを述べる。

- アプリケーションごとに、その要求に合わせてシステムが作られていた。
- ② その構造をながめてみると、機能的には通信回線を制御する回線制御部、端末などの入出力装置を制御する装置制御部、当該アプリッケーション特有の機能に関連する応用部などに分かれるが、これらの境

界が不明確であった。

データ通信システムの需要の拡大にともない、その形態や利用方法が多様化するにつれて、次のような問題が生じてきたことを述べる。

- ① 端末種類の増加などに通信制御ソフトが対応しきれない。
- ② 旧来の端末制御と業務処理が強く結び付いたプログラム構造では、 端末は特定業務プログラムに結びついており、他の業務に利用できな かった。
- ③ システムの巨大化や複雑化に伴う開発コストが増大する。
- ④ バグがなかなかとれず安定しない。
- ⑤ 機能が絡み合っているため、システムの改造や拡張が困難である。
- ⑥ 一メーカの製品で作成されているシステムであるにも関わらず、相 互接続や統合がきわめて困難である。

このため、通信に関わる部分を分離して伝送制御手順、ハードウェア、ソフトウェア等通信に関わる要素全体にわたって標準化・統一化を行い、管理が容易でかつ拡張性と柔軟性に富む概念ならびに仕様をつくり、これに基づいた通信システムの部品を用意しておき、これらを組み合わせてアプリケーションの要求にあった通信機能を作りあげるという考え方が生まれた。

(2) ネットワーク・アーキテクチャの意味

ネットワーク・アーキテクチャという言葉の意味について説明する。

ネットワーク・アーキテクチャの一般的な定義は、「通信ネットワークの論理構造・機能・プロトコルの体系」である。

ネットワークおよびアーキテクチャの意味は辞書によると、次の通りで、 である。

ネットワーク = 網状組織(網細工)

アーキテクチャー 構成方式 (建築様式)

建築物はいろいろ外見は異なっていても、土台・屋根・壁・柱などその 構成要素は共通的である。プレハブ住宅では、建築物はいくつかの要素に 分割されており、これらを組み合わせて客の要望にあった住宅を作り上げ る。屋根等の構成要素は規格化され工場で生産されている。これにより、 高品質の住宅が安価で早く建つようになった。

この考え方をデータ通信システムの世界に持ち込んだのがネットワーク・アーキテクチャであり、通信ネットワークの構成様式ならびに構成要素であるハードウェアやソフトウェアを機能面から整理し論理的に定義したものである。

ネットワーク・アーキテクチャの基本的な概念は、米国のARPAネットワークがその始まりであり、その後IBMがSNAを発表するに至って急速に広まった。

#### (3) 効果

ネットワーク・アーキテクチャを定め、それに従って通信システムを作成することによって得られる効果について述べる。

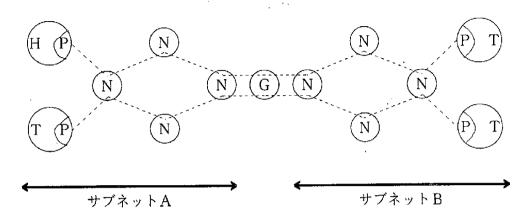
- ① 標準インタフェースを持つ装置の利用やデータ形式の統一により、 システムの開発、拡張、保守が容易となる。
- ② 論理的にネットワークを構成するので、物理的なネットワーク構成 やその変化に処理が影響されない。一貫したネットワーク管理機能に よるシステムの統一した管理の実現。伝送路をはじめとする資源の有 効活用の実現。これらによって分散処理に対して柔軟に対処できる。
- ③ 独立に開発されたシステムを相互に接続して統合システムを作ることが容易にできる。
- ④ ファイル転送、リモート・アクセスや電子メール機能などアプリケーションのレベルまで、ある程度標準化した機能部品を提供できる。
- ⑤ 異なったメーカの製品でも、同じネットワーク・アーキテクチャに 準じて作られた製品ならば、互いに利用してシステムを構築できる。 また、アーキテクチャが異なっていたとしても、アーキテクチャ・レ ベルでインタフェースの調整がとれるので、従来に比べてはるかに相 互接続しやすくなる。さらに、アーキテクチャを公開することにより サード・パーティーによる製品の供給が期待でき、選択の幅が広がる。

#### (4) 論理ネットワーク

ネットワーク・アーキテクチャにおいては、ネットワークは統一的な取 扱いを容易にするため、実際の物理的なネットワーク構成要素をモデル化

- し論理的に考える。論理ネットワークの考え方及び構成について説明する (図表4~3~1参照)。
  - ① 論理ネットワークは、いくつかのサブ・ネットワークをゲートウェイで接続したものである。これをインターネット・システムと呼ぶ。サブネットには複数のノードが含まれる。
  - ② サブネットワークとは、実際にはWANやLANで構成される大小 さまざまな個別のネットワークである。
  - ③ ノードとは、ホスト(コンピュータ)、端末、電子交換機などデータの送信・受信・中継及びそれにかかわる処理を行う装置の総称。ゲートウェイもノードである。
  - ④ ネットワークを介してデータを通信したいプロセスが存在するノードをエンド・ノードと呼び、データを中継するノードを中継ノードと呼ぶ。
  - ⑤ プロセス間の通信方法には2種類ある。互いにコネクションを設定し、それを介してデータを送受する方法(コネクション・オリエンテッド:電話的)とコネクションを設定せずに行う方法(コネクションレス:手紙的)である。
  - ⑥ コネクションとは、ノード間の双方向の伝送路。実態は、LAN、 電話網など。
  - ⑦ 各ノードはネットワーク内で一意に識別できるようネットワーク・ アドレスがつけれる。
- ⑧ プロセスとは、ノード上に存在してデータ送受信を実行するもの。 実態は、アプリケーション・プログラムや通信管理プログラムなど。 ここで述べた論理ネットワークは一例であり、各ネットワーク・アーキ テクチャで異なることに留意されたい。

図表4-3-1 論理ネットワーク



T:端末 N: 中継ノード … : コネクション H: ホスト・コンピュータ G: ゲートウェイ P: プロセス

(5) 階層化とプロトコル(通信規約)

通信を行う際の規約は、階層化されて決められている。その理由および 各層の規約を定めるプロトコルとサービスについて説明する。

- ① 階層化の理由は、次の通りである。
  - (a) あるレベルのプロトコルの変化を、他のレベルのプロトコルに影響を与えない。
  - (b) 下位のレベルのプロトコルを、ブラック・ボックスと見なせる。
- ② プロトコルの決め方の規則は、次の通りである。
  - (a) プロトコルは、同一層間で取り決められる(水平性)
  - (b) ある層のプロトコルの変更が、他層のプロトコルに影響しない(独立性)
- ③ サービスとは、下位層が上位層に提供する機能及びインタフェース( 垂直性)である。
- ④ 上位層(N+1層)から渡されたデータ(N+1層のプロトコル・データ単位)は、下位層(N層)でヘッダ(N層のプロトコル制御情報)を追加されてN層のプロトコル・データ単位として N-1層に渡される。

#### (6) サーバ・クライアント・モデル

種々のネットワーク機能の代表的な実現形態であるサーバ・クライアント・モデルについて説明する。

サーバ・プロセスは、サービス機能(ファイル転送など)を提供する為に窓口を開けて待っている(Listenしている)。クライアント・プロセスは、サービスを受けたいときにサーバ・プロセスと接続する。

図表4-3-2 サーバ・クライアント・モデルの手順

クライアント・プロセス

ネットワーク

サーバ・プロセス

サービス要求待ち(Listen)

サービス開始要求 (ファイル転送など)

サービス **◆** 

サービス終了要求

# 3.2 ネットワーク・アーキテクチャの標準化(120分)

(1) 標準化の必要性とその動向

ネットワーク・アーキテクチャの標準化の必要な理由、OSIの登場までにTCP/IPが果たしてきた役割、現状におけるOSI標準化状況とネットワーク環境ついて説明する。

- ① 1970年代にIBMのSNAをはじめとし各コンピュータ・メーカや通信業者が次々と自社製品に対するネットワーク・アーキテクチャを発表した。これらによって、それぞれの自社製品は統合できたが、異なったメーカの製品同士は依然として接続できず、ソフトやハードを有効に活用できない状況は依然として残ったことを述べる。
- ② TCP/IPは、最初のコンピュータ・ネットワークであるARPA ネットワークの第2世代のネットワーク・アーキテクチャであり、米国

の大規模なWANで採用されるとともに、ワークステーション(WS)の標準OSであるUNIXでもサポートされている。これにより、メーカの枠を越えた事実上の業界標準(デファクト・スタンダード)として急速に普及した。現在では、TCP/IPのネットワーク網は世界中に広がっており、ノードのアドレスをつける体制も整っていることもあり、TCP/IPを利用できるコンピュータがあれば、世界規模で電子メール交換やファイル転送などが可能となっていることを述べる。

- ③ TCP/IPは広く普及してはいるが、対象はWSなどの小型機が多く規格として厳密性に欠ける所もあり、また、機能の範囲が限定されている。そこで、あらゆる通信技術を包括し小型から大型まで広い範囲のコンピュータや端末で利用可能かつ相互接続が容易な国際的標準を作ることが必要とされ、1970年代後期からCCITTとISOで標準化作業が開始された。これがOSI(Open Systems Interconnection)であることを述べる。
- ④ OSIの制定状況と製品化の現状ならびにSNAなど、メーカ独自のアーキテクチャとの整合性について述べる。
- ⑤ OSIの国際標準化活動は強力に推進されてはいるが、ネットワーク・アーキテキャとして整備されそれに対応した製品が出揃い、OSIがTCP/IPにとってかわるまでにはまだ時間が必要であろうことについて述べる。

# (2) 標準化の手順

OSIを構成する各種の規約の案が出され、それが審査され国際標準となった後、それに対応した製品が世の中に出るまでの手続きと関連する組組織について説明する。

- ① 規約のドラフトから実装規約作成までの流れについて述べる。
- ② ISOとCCITTの説明を行いOSIとの関連について述べる。
- ③ SPAG、INTAPなどのフィーダーズ・フォーラムの活動内容について述べる。
- ④ コンフォーマンス・テストについて述べる。

## 3.3 OSIのプロトコル階層(360分)

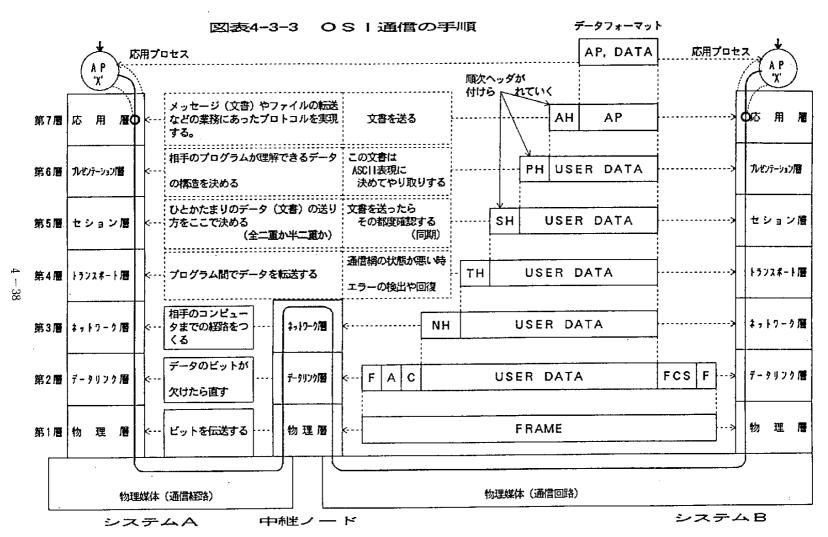
(1) OSI

図表4-3-3 および図表4-3-4 を参考にし、OSI基本参照モデルについて説明する。

- ① OSIの7層のプロトコル階層について、各層の役割を述べる。
- ② 7層のうち、下位4層は上位層に対し要求された品質の透過的なデータ伝送機能を提供し、上位層はアプリケーションの作成を容易にする機能を提供することを述べる。
- ③ 伝送制御手順は、第2層、半2重・全2重などの制御は第5層に属するなど、第2章で述べた各項目がOSIのどこの層に属するかを適切に示し、第2章の項目とのつながりを持たせる。
- ④ 説明にあたっては、「会社間での契約書の取り交わし」などを例にあ げると分かりやすい(図表4-3-5参照)。
- ⑤ 応用層の例としては、FTAM(ファイル転送)、MHS(電子メール)、VT(仮想端末)をあげる。

図表4-3-3 OSI各層の役割

,	•	
層の名称	役	割
応 用 層 (第7層)		ロセス(データ通信システムの 用プログラム部分)や端末の利 供する。
ブレゼンテーション 層 (第6層)	ロセスで取り扱っているデ ためこれらを共通なものに その変換および逆変換機能 ロセスは符号化を意識しな	信を前提としており、個々のプータの型や符号は異なる。この変換して通信する必要がある。を提供する。これにより応用プいですむ。転送されるデータのしておき、その構造に従って共送構文と呼ぶ。
セション層 (第5層)	全二重・半二重、優先デー を提供する。	夕、送信権、同期、再送機能等
トランスポート 層 (第4層)	たり、逆に過多な場合に多 行って、セション層が要求 種類)をもつ全2重の透過	網の品質や速度が不十分であっ 重化、分流、連結、誤り制御を する品質と速度(①から4の5 的な伝送路を提供する。また、 別するトランスポート・アドレ
ネットワーク層 (第3層	ートを選定し、データを中 パケット交換や回線交換な	どの交換機能、例えば、DDX トコル等が対応する交換機など
デ-タリンク層 (第2層)		位での伝送を保証する。WAN に対するMAC(CSMA/C が対応する。
物理層(第1層)	を介してビット単位の伝送	WANやLANなどの通信回線 を行う。DTE-DCEインタ でありV. 24、X. 21やRS-232など



AH:Application Header SH:Session Header NH:Network Header FCS:Frame Check Squence PH:Presentation Header TH:Transport Header F:Frag A:Address C:Control

| 図表4-3-5 | 契約書の取り交わしとOSIの階層の対応

OSI	日本の石油精製会社	サウジの石油生産会社
応用がなる	(社長) 石油の輸入契約を 早急に行いたい	(社長) 契約に基ずき石油の 輸出を指示 介
応用層	(秘書) 書式に従い日本語 で契約書を作成	・ (秘書) 契約書の内容をチェック ☆
ブレゼンテー ション層	◆ (通訳) 契約書を共通言語 【英語】に翻訳	・ (通訳) 契約書の内容を アラビア語に翻訳 へ
セション層	↓ (総務) 契約書を書留速達 で発送するととも にそのことを相手 会社の総務に通知	¥ (総務) 契約書を確かに受け 取ったことを送り元 の総務に通知
トランスポート層	◆ (郵便局) 信頼性の高い航空 会社を選び他の荷 とともに輸送依頼	☆ (郵便局) 荷物を分解し当該 会社宛の書類を配達
ネットワーク層	◆ (航空会社) (航空会社) 配送ルートを決定 ↑ ↓ し相手国まで運ぶ	∬ (航空会社) 日本からの荷物を 郵便局に渡す
データリンク層	(バロット) 隣接地まで安全 確実に輸送	(パロット) (パロット) 隣接地から安全確実 に輸送
物理層	◆ (操縦システム) 航空機を隣接地 まで操縦 1	で (操縦システム) 航空機を隣接地から 操縦 介
物理媒体	→ (航空機A) → → (航 日本 インド	

- (2) TCP/IPとの比較
  - 現実的な業界標準となっているTCP/IPとOSIを対比させる(図表4-3-6)。
  - ① TCP/IPの4層のプロトコル階層について、各層の役割について 述べた後、両アーキテクチャの階層分けの違いを説明する。
  - ② TCP/IPの応用層の例としてはFTP(ファイル転送)、TEL NET(リモート端末プロトコル)、SMTP(電子メール)をあげる。 また、LANの高度な利用例として、NFS(ネットワーク・ファイル・システム)やXウ インドウを示すのもよい。
  - ③ OSIの課題について、TCP/IPと対比しながら述べる。
    - (a) 層分けが多いことによるオーバヘッドの問題。
    - (b)アドレスの登録、障害の切りわけなど運用管理関係の充実度の不足。
    - (c) 規格制定に時間がかかるため、現実の技術や利用環境の変化に、適切 に追従しにくい。

図表4-3-6 TCP/IPとOSIの比較

### OSIプロトコル

7	応 用 層
6	プレゼンテーション層
5	セション層
4	トランスポート層
3	ネットワーク層
2	データリンク層
1	物 理 層

### TCP/IPプロトコル

T	С	P	/	I	P	応用
---	---	---	---	---	---	----

FTP TELNET

· SMT Petc

TCP

IP-ICMP

ネットワーク依存部

TCP : Transmissin Control Protocol

I P : Internet Protocol

ICMP: Internet Control Message Protocol

#### 育成方法および指導上の留意点

- 1. 「第2章通信ネットワークの基礎技術」で述べた内容と関連をもたせる。
- 2. ネットワーク・アーキテクチャ概念の説明が抽象的にならないようにする。
- 3. OSIは多くの要素を持っているので、本質的なところを抽出して説明する。
- 4. 現実のネットワークは、LANをWANで結ぶインタ・ネットワーキングがこれからの主なテーマとなりつつあるので、その動向に留意する。
- 5. ネットワークの世界は、動きが激しいので常に新しい知識をいれて、教育 内容が陳腐にならないように留意する。
- 6. パソコン・ネットやEWSを利用して電子メールやファイル転送を行わせ、 機能を実感させることが望ましい。

### 用 語

ネットワーク・アーキテクチャ、インタ・ネットワーク、ARPA、SNA、ファイル転送、リモート・アクセス、電子メール、プロトコル(通信規約)、階層化、論理ネットワーク、ホスト、ノード、ゲートウェイ、コネクション、プロセス、サーバ・クライアント・モデル、TCP/IP、OSI、CCITT、ISO、サブ・ネットワーク

### 第4章 ネットワーク・サービス

# 学習目標

ネットワーク・サービスの目的、その種類とそれぞれの特徴を理解し、多様なサービスの区別ができるよう、情報処理技術者としての常識を持たせる。

本章が修了した時点での学習目標は、以下の通りである。

- 1. ネットワーク・サービスの意義や目的が十分理解できる。
- 2. 多様なサービスを選択する場合のポイントを説明できる。
- 3. 身近な情報処理システムで使われているサービス機能の長所・短所について、ある程度の評価ができる。

### 内容

### 4.1 ネットワーク・サービスと電気通信事業法(60分)

- (1) ネットワーク・サービスとは 以下の点を中心に説明する。
  - ① ネットワークは情報処理システムの一要素であるが、情報処理システム毎にそれぞれネットワーク・システムも構築するのでは大へん負担が大きくなる。そこで特別の要求がある場合を除き、既存のネットワーク・サービスを利用することができれば大へん便利である。そしてそのために現在、多様なサービスが存在する。
  - ② 多数の利用者が共同で利用するネットワーク・システムは、信頼性・ 安全性などの点で十分な機能や品質が確保されていなければならない。 そこでそのサービスは国の規制を受ける。即ち電気通信事業法という法 律のもとで通信事業者の行なうサービスの枠組みが設定されている。
- (2) 電気通信事業法 以下の点を中心に説明する。
  - ① 最新の電気通信事業法は、昭和60年4月に施行された。この法律では 通信サービスを行なう事業者を種類分けし、それぞれの規制内容を定め

ている。

② 電気通信事業者には図表4-4-1 に示すような区別がある。 図表4-4-1 電気通信事業法の区分

種	別	事 業 内 容	認可方法
第一種電気	通信事業	電気通信回線設備を設置して電気通信役	許可制
		務を提供する事業	
第二種電気	通信事業	第一種以外の電気通信事業	
特別	第二種	・電気通信設備を不特定かつ多数の者の	登録制
		通信の用に供する。	(許可と
		・政令で定める基準を超える規模。	届け出の
		・本邦外の場所との間の通信を行う。	中間)
一般	第二種	特別第二種以外の第二種電気通信事業	届け出制

- ③ 第一種電気通信事業者としては、かつては国内通信はNTT、国際通信はKDDがそれぞれ独占していたが、昭和60年のこの新事業法によって他社の参入が認められた。これは、長いわが国の通信事業にとってたいへん大きな変化であった。
- ④ 第二種電気通信事業者は、第一種事業者から基本的な回線サービスを受け、それに種々の付加機能を付け加えたり、大口の回線を小口に分けて中小の利用者に再販を行なったりする。前者は付加価値通信ネットワーク(Value Added Network略してVAN)と呼ばれる。
- ⑤ それぞれの時点での各事業者数の概数を紹介し、全体的な傾向を把握させる。

# 4.2 主なサービス (300分)

- (1) 専用回線サービス 以下の点を中心に説明する。
  - ① 専用回線サービスの種類には以下のものがある。それぞれの概要を説明する。

- (a) 帯域品目サービス
- (b) 符号品目サービス
- (c) 高速ディジタル伝送サービス
- ② これらの専用回線を用いて、利用者は独自のネットワークを構築することができる。
- (2) 交換サービス

ネットワーク機能が既に実現されており、多くの利用者がそれを共用する。情報処理に用いられる代表的なものとして以下のサービスがある。

- ① 電話交換サービス
- ② 回線交換サービス
- ③ パケット交換サービス それぞれの概要を説明する。
- (3) VANサービス以下の点を中心に説明する。
  - ① 第二種電気通信事業者は図表4-4-1 に示すように、更に特別第二種と一般第二種に分かれる。VANに対応させると、特別第二種の事業者によるものを汎用VAN、一般第二種の事業者によるものを個別VANとそれぞれ通称する。
  - ② 汎用 V A N は、利用者がこれを用いて、後述のコンピュータ・ネットワーク・サービスやパソコン通信サービスを行なうこともできる。
  - ③ 個別VANはある目的に特化したVANで、ネットワーク上のアプリケーション機能まで含むことが多い。具体例をあげて概要を説明する。
  - ④ ネットワークの付加価値機能の例として図表4-4-2 に示すようなものをあげ、その目的や機能の概要を説明する。
- (4) サービスの選択

専用回線サービス、交換サービス、汎用VANサービスなど、それぞれの利用上の要件を、トラフィック量、安全性、価格などの点から特徴付け、それらの選択のポイントにつき本質的なところを理解させる。

簡単なモデルと、トラフィック量、料金表を与え、DDXと専用回線利用の料金比較を計算させる演習を行うのもよい。

図表4-4-2 ネットワークの付加価値機能の例

	機能	説明
1	パケット交換	パケット組立・分解、蓄積交換、ルーティング、フロー制御、エラー制御、パケット多重等
2	ネットワーク 制 御 管 理	ネットワーク内の各構成要素の障害管理・制御, ト ラフィックの監視・制御, アドレス管理等
3	各種端末の接続	端末手順の変換,仮想端末制御
.4	各種コンピュー タ の 接 続	同機種、異機種コンピュータのネットワーク接続
5	ファイル転送	ホスト・コンピュータ間のファイルの転送. コード、記録媒体、構造等の変換
6	リモート・ログ・イン	ホスト・コンピュータにおけるリモート・ハッチ 処理
7	会話型処理	ホスト・コンピュータにおける会話型処理
8	電子メール	端末間でのメールの交信、不在時のメール・ホックス機能
9	電子揭示板	多数の利用者間の情報交換
10	同 報 通 信	同一メッセージを複数サイトに一斉に転送する
11	代 行 受 信	受信サイトがメッセージを受け取れぬ場合に代りに 受信する
12	閉域接続	ネットワーク上の限られたユーザ・グループ間のみ に通信を限定する
13	固定接続	頻度の高い通信相手間の接続を常時固定化する
14	相 手 通 知	交信開始以前に相手を知らせる
15	暗号化通信	メッセージを暗号化して伝送する
16	マルチメディア	音声・画像・コード等の各種形態の混合情報の伝送
10	機能	およびメディアの相互変換
17	文 書 処 理	文書情報の転送
18	ネットワーク 操 作 言 語	同一コマンドでネットワーク内のすべてのコンピュ ータを使用できる(NJCL)
19	データベース	

#### 4.3 ネットワークを用いたサービス(180分)

4.2 で取り上げたもの以外のサービスのうち、以下のものを中心に、それぞれの特徴を把握させる。

- (1) コンピュータ・ネットワーク・サービス 以下の点を中心に説明する。
  - コンピュータ・ネットワークとは、通信ネットワークと、そのネットワークに接続されたホスト・コンピュータ群とからなる。各ホスト・コンピュータのハードウェア、ソフトウェア、データベースなどのリソース(資源)を、ネットワークを通して多くの利用者が共用することが目的である。
  - ② コンピュータ・ネットワークは、企業組織内のシステムとしても構築 されるが (例 1.3(1) 企業内分散処理システム) ネットワーク・サー ビスとしても多くのシステムが存在する。
  - ③ コンピュータ・ネットワーク・サービスは、第一種事業者のサービス を用いるもの、第二種の汎用VANサービスを用いるものもある。また 第一種事業者や第二種事業者自身によるサービスもある。
  - ④ コンピュータ・ネットワークを用いる代表的なサービスは、情報処理 サービスとデータベース・サービスである。
  - ⑤ 情報処理サービスについては以下の点にふれる。
    - (a) 異なるホスト・コンピュータの多様なリソースが、1 つの端末から 利用できる利点
    - (b) 会話型処理やリモート・バッチ処理、ファイル転送などのサービス の種類がある。
    - (c) ホスト・コンピュータの差異を意識させない共通のネットワーク・ アクセス言語や、あるホスト・コンピュータの障害時に、他のホスト で代替する機能などの方向
  - ⑥ データベース・サービスについては以下の点にふれる。
    - (a) データベースはコンピュータのリソースの中でも最も共通性の高い ものであり、多様なデータベース・サービスが行われている。主な例 をあげて概説する。

- (b) 国内のみならず、国際的なサービスも多い。
  - (c) DBMSの差異や、データベースの存在場所を意識させない、分散 型データベースの方向
- (2) パソコン通信サービス 以下の点を中心に説明する。
  - ① コンピュータ・ネットワークがリソースの共用を目的とするのに対して、パソコン・ネットワークは、利用者相互の情報交換が主な目的である。
  - ② パソコン通信の主な機能として、電子メール、BBS、ソフトウェア 配布などをとりあげ概説する。
  - ③ パソコン通信サービスの現状や、内外の主なサービスにつき紹介し、 動向を把握させる。
- (3) ビデオテックス・サービス 以下の点を中心に説明する。
  - ① ビデオテックス・サービスとは、端末機としてテレビ受像機やパソコンなどを用い、電話交換ネットワークを介して中央の情報センターに蓄積された画像情報を提供するサービスである。
  - ② ビデオテックス・サービスには以下の二種類がある。
    - (a) ビデオテックス通信ネットワーク・サービス
    - (b) キャプテン・サービス このうち(a)のサービスを用いて利用者が独自のビデオテックス・システムを構築することができる。また(b)はNTTが行なっている固有のシステムである。
  - ③ ビデオテックス・システムの利用方法として、企業内や、地域グループなどの例をとりあげて、その特徴を概説する。
- (4) ファクシミリ・サービス以下の点を中心に説明する。
  - ① ファクシミリには、アナログ網である電話交換網を使用するサービスと、ディジタル網であるファクシミリ網を使用するサービスとがある。
  - ② ファクシミリ通信にディジタル網を使用することによって以下のよう

なサービス機能が可能となる。

- (a) 送受信エラーの対策
- (b) パスワードの使用
- (c) 暗号化通信
- (d) 同報通信
- (5) マルチメディア通信サービス

コンピュータ処理やデータベースにおいて、文字、画像、音声などの多様な形態の情報の統合処理、すなわちマルチメディア処理が求められているのと同様に、通信においてもマルチメディア情報が混合して伝送できることが求められるようになった。すなわち、従来は

データ・ネットワーク

電話ネットワーク

ファクシミリ・ネットワーク

などがそれぞれ独立して存在していたが、1つのネットワークで、どの目的も達することが出来れば便利であり経済性も高い。

マルチメディア通信に関しては、以下の点を中心に説明する。

- ① マルチメディア情報の伝送には、相対的に大きな伝送容量が必要となる。
- ② 各種の形態の情報を全てディジタル化することにより、その混合や統 合通信が容易となる。
- ③ 現在行われている主なサービスには以下のようなものがある。それぞれの特徴を概説する。
  - (a) 高速ディジタル専用回線サービス
  - (b) ISDN (International Services Digital Network)サービス
  - (c) 衛星通信サービス

### 4. 4 LAN製品(120分)

- LAN製品には、以下の2種類がある。
  - ① ネットワーク製品
  - ② システム製品

(1) ネットワーク製品

使用目的により、以下のような点で、それぞれ特色を持つ、多種類の製品が存在する。

- 伝送速度
- ② ネットワーク形状
- ③ アクセス制御方式
- ④ ベース・バンドかブロード・バンドか
- (2) システム製品

ワークステーション、パソコンあるいは汎用コンピュータなどをLAN で相互に接続した、ハードウェアや基本ソフトウェアを含むシステム化さ れた製品である。

OA、FAなどの利用例をあげ、システム製品の機能の特徴を説明する。

## 育成方法および指導上の留意点

- (1) わが国におけ通信自由化の経緯を説明し、諸外国との比較、情報化の進展 に与える影響などについてもふれることが望ましい。
- (2) 各種サービスの細部にわたる説明は必要ないが、どのようなサービスの種類があり、それぞれの特徴を理解し、サービス間の区別が出来る程度の、情報処理技術者としての常識的知識を与えることは必要である。
- (3) 物理的な1つのネットワークの上に、仮想的な多数のネットワーク・システムを構築できること、および1つのネットワーク・システムの上に、多様なアプリケーションをのせることが出来ることを理解させる。
- (4) VAN機能のデモンストレーションが行われることが望ましい。
- (5) ネットワーク・サービスは種類も機能も変化が激しい。それぞれの時点で常に現状を把握し、時代遅れにならぬ知識の提供に務めねばならない。

### 用 語

ネットワーク・サービス, 電気通信事業法, 第一種電気通信事業, 第二種電気通信事業, 特別第二種, 一般第二種, 付加価値通信ネットワーク, VAN, 専用回線, 交換サービス, 回線交換サービス, パケット交換サービス, 汎用VAN, 個別VAN, コンピュータ・ネットワーク, ホスト・コンピュータ, リソース, 情報処理サービス, データベースサービス, 会話型処理, ネットワーク・アクセス言語, 分散型データベース, パソコン通信, BBS, ビデオテックス, キャプテン・サービス, ファクシミリ・サービス, 同報通信, ISDN

### 第5章 ネットワーク・セキュリティ

### 学習目標

ネットワーク社会における大きな課題の1つとして、ネットワーク・セキュリティ問題がある。

初級情報処理技術者としても、ネットワークを利用し、あるいは構築に参加する立場として、セキュリティに対する基本的知識を持つことは必要不可欠であり、日常の業務において、ネットワーク・セキュリティに対し適切な判断ができることを目標とする。

### 内容

### 5.1 ネットワーク・セキュリティとは(30分)

情報処理システムに対するセキュリティを総称してコンピュータ・セキュリティと呼ぶ。

ネットワークは情報処理システムの一要素であるが、ネットワーク特有のセキュリティ問題に焦点をあてる場合にネットワーク・セキュリティという言葉を使う。

ネットワーク・システムは設備的にも長い距離にわたって設置され、監視も 困難でありかつさまざまな外的な悪影響も受ける。また遠隔地にいる通信相手 は目に見えず不安も大きい。したがって情報処理システムの中でもネットワー クは最もセキュリティ対策が重要となる部分である。

ネットワーク・セキュリティには他と同様に信頼性と安全性の2つの側面があり、いずれも設備的、運用的、技術的な対策が存在する。

### 5.2 ネットワーク・システムの信頼性対策(90分)

以下のような信頼性対策につき説明する。

- (1) 伝送エラーの検出や再送機能
- (2) ネットワークのどこの部分に障害が起きたのかを検出する切り分け機能

- (3) 回線障害に対する回線の二重化機能
  - ① 無線回線によるバックアップ
  - ② 交換サービス・ネットワークによるバックアップ
  - ③ 専用回線の二重化
- (4) 障害箇所を避けるネットワークの迂回路の設定
- (5) ネットワークを構成する各種ハードウェアの二重化や代替機能
- (6) 過度のトラフィックを検出し、障害を未然に防止する機能
- (7) ネットワークの障害監視・管理機能

### 5.3 ネットワーク・システムの安全性対策(150分)

- (1) ネットワーク・システムにおける安全上の脅威 ネットワークが脅威を受けやすい理由を以下のような点から説明する。
  - ① 利用者や通信相手が遠隔の見えない場所にいる(不正が行われやすい)
  - ② 不特定多数の非専門家が利用者である可能性が高い(セキュリティ意識に欠け、訓練も行われていない)
  - ③ パソコンを始めとする端末類は、何等ガードされていない場所に設置されている(誰でも使用でき不正がおきやすい)
  - ④ パソコンの端末化により不正が容易となる(例:多数のパスワードの 試行の自動化)
  - ⑤ 伝送データの盗聴が行われやすい
  - ⑥ 遠距離にわたる通信施設は、悪意の妨害を受けやすい
- (2) 安全性対策機能 以下のような機能につき、技術的、運用的な両面から説明する。
  - ① 個人識別 利用者が正当な利用者であるか否かを識別することが目的である。以 下のような手段がある。
    - (a) IDカードなど本人のみが所持している持ち物
    - (h) I D コード、パスワードなど本人のみが知っている情報
    - (c) 指紋、声紋、網膜パターンなど本人自身のフィジカルな特徴
    - (d) 手書きサイン

以上のうち、パスワードについては、その選び方、適宜の変更など運用上の要件も指適する。

② アクセス・コントロール 情報処理システム内のファイルやデータを保護することが目的である。

以下の点を中心に説明する。

- (a) 全てのファイルに対し、利用者のアクセス資格を予め設定しておき、 実際のアクセス時にチェックする。
- (b) 対象となるファイルは以下のような全てのファイルである。

#個プログラム
プログラム・ファイル #通プログラム
個別アプリケーション・プログラム ボータ・ファイル 共通ファイル 個別アプリケーション用ファイル

(c) ファイルアクセスの資格としては、以下のようなものがある。

READ可能

WRITE可能

APPEND可能

UPDATE可能

DELETE可能

EXECUTE可能

資格の変更可能

#### ③ 暗号化

不正なアクセスや盗聴が行われても、内容の理解を不可能にすることが目的である。以下の点を中心に説明する。

- (a) 暗号化の対象は、ファイルと伝送データの双方がある。
- (b) 暗号化には、暗号アルゴリズムと鍵(キー)とがあり、双方を秘密 にする方式と、アルゴリズムは公開し、キーのみを秘密にする方式と がある(後者の例:DES)。
- (c) 暗号化アルゴリズムを公開することにより、製品化が推進され、暗

号化の普及に貢献した。

- (d) 秘密鍵暗号系と公開鍵暗号系のそれぞれの概要と、両者の比較
- ④ 通信メッセージの認証

ネットワークで交信している相手が正当な相手であるが、あるいは受信したメッセージが正当なものであるか、途中で改ざんされていないかなど、ネットワーク上での交信の安全性を確認する手段として、以下のような機能があることを説明する。

- (a) 端末確認
- (b) コールバック
- (c) メッセージ認証
- (d) ディジタル署名

#### 5.4 ネットワークの種類とセキュリティ(30分)

セキュリティの面から利用するネットワークを選択する場合の基本的視点に つき説明する。

- ① 専用回線を借り、自らの専用ネットワークを構築する場合には、目的に 合わせたセキュリティ機能を設定しインプリメントする事ができる。
- ② 交換ネットワーク・サービスでは不特定多数の利用者があり、安全性上の脅威は比較的大きい。またバックアップ機能や、回線の品質などは、サービスにより若干異なり、利用時に信頼性上の機能につき検討し、サービスの選択を行うことが望ましい。
- ③ 交換ネットワークでは、閉域接続、相手通知などセキュリティ上の付加 機能を必要に応じ利用することが望ましい。
- ④ 無線回線は基本的に盗聴に対し大へん弱い。目的に応じ何等かの暗号化 手段が必要である。
- ④ 暗号化やメッセージ認証など、最近はモデムにこれらのセキュリティ機能を内蔵した製品も出現している。

### 育成方法および指導上の留意点

- (1) 起り得る脅威と、その対策との関係を明確にするよう、適宜、具体例を示すことが望ましい(例:盗聴の例、不正端末からのアクセスの例、メッセージ改ざんの例など)。
- (2) ネットワーク化により、ウィルスやハッキングなど、情報処理技術者自身によるセキュリティ侵害の脅威が大きくなっており、技術者の職業的倫理感の確立が極めて重要であることにふれる。

### 用 語

ネットワーク・セキュリティ, コンピュータ・セキュリティ, 切り分け機能, 迂回路盗聴, 個人識別, IDカード, IDコード, パスワード, 指紋, 声紋, 網膜パターン, アクセス・コントロール, アクセス資格, 暗号, 暗号アルゴリズム, 鍵, 秘密鍵暗号系, 公開鍵暗号系, メッセージ認証, 端末確認, コールバック, ディジタル署名

# インストラクタ必読書

1. 「データ伝送の基礎知識」

電気通信協会 編、オーム社、1985年

2. 「コンピュータネットワーク」(bit増刊) 石田晴久ほか 著、共立出版、1986年

# 参考文献

- 「オンラインシステムの設計」
   甘利直幸 著、オーム社、1989年
- データ伝送ハンドブック」
   電子通信学会 編、オーム社、1885年
- 3. 「JISハンドブック情報処理」 データ通信 編、日本規格協会、1989年
- 4. 「ユーザのためのOSI Q&A」OSIユーザー懇談会 編、日本規格協会、1987年
- 5. 「JISハンドブック情報処理」 データ通信 編、日本規格協会、1990年
- 6. 「TCP/IP」 西田竹志 著、ソフト・リサーチ・センター、1989年
- 7. 「TCP/IPによるネットワーク構築」(bit別冊) Doulas Comer 著、村井 純ほか 訳、共立出版、1990年
- 1コンピュータ・ネットワーク」
   八木 聴・勅使河原可海 共著、朝倉書店、1990年
- 9. 「通信白書」

郵政省 編

### 第5部 情報処理システム

### 教育目標

情報処理システムは、コンピュータのハードウェア、ソフトウェアを用いて、 情報を加工、伝達、蓄積するためのしかけということができる。

マイクロ・エレクトロニクス技術の発達およびニーズの多様化などの背景をもとに、システムの総合化技術も発達してきた。

そのような背景のもとで、第5部では、情報システムに関する基本的な処理方式、システムの構成方法、およびそこで使われている技術について学ぶ。さらに、発展、応用形として、マルチメディア処理、AI(人工知能; Artificial Intelligence)システム、CAI(Computer Asisted Instruction)/ CMI(Computer Managed Instruction)システムをとりあげ学習する。

当部の学習を行うことにより、学習者は、情報処理システムという考え方、および情報処理システムをいくつかの基本要素に分解し理解できる力をつけるごとができる。

# 内容の構成

内容	時 間 数	
	講義	演習
第1章 処理方式 1.1 処理方式の種類と目的 1.2 処理方式とアプリケーション	5 ^	
第2章 システム構成 2.1 システム構成の種類と特徴 2.2 分散処理 2.3 システム統合とその目的	5	
第3章 マルチメディア処理 3.1 コンヒュータにおけるマババイ処理 3.2 図形処理の技術と応用 3.3 画像処理の技術と応用 3.4 音声処理の技術と応用	1 5	
第4章 AIシステム 4.1 AIとは 4.2 AIの応用 4.3 AIシステムの技術動向	1 0	
第5章 CAI/CMIシステム 5.1 システムの機能と目的 5.2 現状および今後の動向	5	
合 計	4 0	

### 第1章 処理方式

### 学習目標

コンピュータの利用分野の拡大、通信技術との融合により、コンピュータに よる処理方式、システムの構成形態は、多様化している。この章では、処理方 式の基本であるバッチ処理方式、リアルタイム処理方式、分散処理方式の考え 方、適応業務について学ぶ。

この章の内容を終了した時点で、次のことができるようになることを目標と する。

- 1. 処理方式の基本、バッチ処理方式、リアルタイム処理方式、分散処理方式について、その基本的な考え方を明確に述べることができる。
- 2. システム化のテーマを提示されたとき、そのシステムを実現するために どの処理方式が適切か判断できる(ただし、この章では基本的なことがら が理解できていればよいものとする)。
- 3. 上記2つの観点を理解した上で実際のシステムはいくつかの処理方式の 組合わせで実現されていることを知る(たとえば、銀行オンライン・シス テムでは、リアルタイム処理だけでなく、リアルタイム処理と密接な関係 をもつバッチ処理とで構成されていることを知る)。

# 内 容

# 1.1 処理方式の種類と目的

情報処理システムの処理方式を決定する場合、重要な要素として適応業務の特徴、とくにデータの発生パターン、処理結果の取り出し方がキーとなることを指導する。図表5-1-1に例を示す。

図表5-1-1 適応業務の特性と処理方式の例

適 応 業 務 の 特 性 等	処 理 方 式
<ul> <li>・データはまとまったタイミングで発生する         <ul> <li>(例)給与計算業務</li> <li>大量データの統計処理業務</li> </ul> </li> <li>・データ量が多い</li> <li>・データの発生場所が比較的まとまっている</li> <li>・即時性は低い</li> </ul>	バッチ処理 (一括処理)
<ul> <li>・データの発生がランダムである</li> <li>・データが発生するたびに即時処理が必要である</li> <li>・1回のデータ量は多くはない</li> <li>・データの発生場所がまとまっていない</li> <li>(例)銀行の窓口業務</li> <li>座席予約業務</li> </ul>	リアルタイム処理
・リアルタイム処理の規模が大きくなり1ヶ所の センタで処理できない(負荷分散、信頼性の向上) ・システムに対する要求が多様化して、1ヶ所の センタでは実現できない(機能分散) (例)総合化オフィス業務 大規模リアルタイム処理業務の分散処理化	分散処理

# (1) バッチ処理方式

バッチ処理方式には通常のセンタ・バッチ処理方式とリモート・バッチ 処理方式のあることを理解させる。

# ① センタ・バッチ処理方式

バッチ処理の目的は、データを一括処理することにより、コンピュータ・システムの資源を有効に利用しスループットをあげることにある。 データを一括処理するために、通常は発生したデータをまとめ、運送手段等を用いてセンタに送る方式がとられる。その他、データ収集システムを利用しデータをセンタに蓄積した後、一括処理する形態もある。この場合データ収集の部分は通信回線を用いる。

# ② リモート・バッチ処理方式

データの入力と結果の取り出しをセンタから離れた端末で行い、コン ピュータ処理は、バッチ処理で行う処理方式である。 当然のことながら、データの入力と結果の取り出しは、通信回線を用いて行うことになる。

### (2) リアルタイム処理方式

リアルタイム処理方式は処理すべきデータが発生するつど直ちに処理を 行う方式である。

リアルタイム処理を行うためには、

- ① センタはデータが発生したら即時処理しなければならないので、1 つのデータ処理に要する時間は短時間であることが要求される。
- ② データの発生からセンタまで情報を伝達するためおよび結果を即時に受け取るために、通信回線の活用が必要となる。

という特徴をもつ。

通信回線を用いた処理をオンライン処理と呼ぶ。バッチ処理方式のところでふれたリモート・バッチ処理も通信回線を用いるので、オンライン処理であるが、(2) ①の条件は満たさなくてもよいのでリアルタイム処理ではない。学習に当たっては、この部分をしっかりととらえることが大切である。

リアルタイム処理方式はさらにオンライン・リアルタイム処理とタイムシェアリング処理に分けることができる。いずれも、即時性、通信回線を用いるといった点では共通であるが、システムの処理方式としては根本的に異なる。

オンライン・リアルタイム 処理:センタは処理すべき業務の指示をあたえられ、端 末から処理すべきデータを与えられて処理を行う。

9イムシュアリンク処理:センタは端末から処理すべき業務とデータを与えられて処理を行う。したがって、端末側からみるとあたかもセンタを独占しているような形態でセンタを用いることができる。

### (3) 分散処理方式

分散処理方式については、以下のポイントが指導の中心となる。なお、 分散処理については第2章で詳しくとりあげる。

① 分散処理方式は複数のコンピュータ・システムをネットワークで結ん

で実現する処理方式である。

- ② 分散処理方式のシステムでは、何をどのように分散するかがポイントになる。一般的には、それぞれのコンピュータで行うべき機能を階層化した垂直分散型システムの形態とそれぞれのコンピュータを論理的に対等に結んだ水平分散型システムの形態がある。
- ③ 分散処理方式の実現は、ネットワーク技術と高性能ワークステーションの出現によるところが大きい。

#### 1.2 処理方式とアプリケーション

各処理方式ごとにアプリケーションの例を示し、理解させる。

- (1) バッチ処理方式によるアプリケーションの例
  - ① 給与計算システム

人事関係のデータ処理で最初に機械化されたのが給与計算システムであり、図表5-1-2に概略図を示す。

本システムがバッチ処理で構築される理由は、

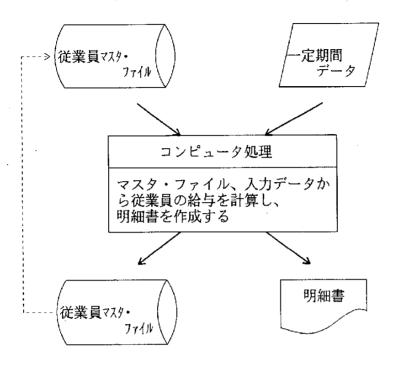
- (a)データ処理はある特定の期間に集中する
- (b)大量データを一括処理する

などによる。ただし、この給与計算システムも、人事の情報を蓄積する ような形に発展すると、リアルタイム処理が必要となってくる。

- ② リアルタイム処理方式に付帯したバッチ処理
  - ①のように業務自体が、バッチ処理に適したケースの他に、リアルタイム処理に付帯したバッチ処理がある。

リアルタイム処理は、即時性を保つため、1つの取引データを処理するための時間に制約がある。したがって、リアルタイム処理を一定期間 実行した後、例えば日単位、週単位、月単位などで、取引データの集計、 累積を行うことが多い。このための処理方式はバッチ処理となる。

図表5-1-2 給与計算システム概略図



### (2) リアルタイム処理方式のアプリケーションの例

リアルタイム処理方式は取引データが発生するたびに処理を行い、結果を出力する方式である。現在の情報処理システムの多くは、バッチ処理方式を併用したリアルタイム処理方式を用いて実現されている。

リアルタイム処理方式のアプリケーションはさらに次のように分類できる。

① 照合応答形式のアプリケーション

端末から入力されたデータにもとづいて、データベースを検索し応答 に必要な情報を取り出して問い合わせに応じる。

例: 文献検索システム

座席予約の空き状況の問い合わせシステム

② データベース更新形式のアプリケーション

端末から入力されたデータにもとづいて、そのデータに対応するデータベースの更新や追加削除を行い端末側で要求する情報を送り返す。

例:座席予約システム 銀行預金システム

③ メッセージ交換形式のアプリケーション

端末から入力された、データ(メッセージ)を他の端末に送信する機能をもつ。送信の方法としては、1つの端末からのデータを他の1つの端末へ送信する場合と、1つの端末からのデータを他の複数の端末へ送信する場合がある。

例:電子メール・システム

- ④ タイムシェアリング形式のアプリケーション
- (3) 分散処理方式のアプリケーションの例
  - ① 役員室OA(Office Automation)システム

システム構成は図表5-1-3のようにホスト・コンピュータと分散処理 用コンピュータをネットワークで結び、役員室での分散処理用コンピュータを用いることにより経営状況が即時に入手できるシステム構成になっている。

分散処理方式として構築する背景としては、

- (a)技術革新のスピードが早く新製品の開発サイクルが早くなっている。
- (b)経営者は必要な情報を迅速に入手する必要にせまられている。
- (c)非定型の処理が多い(共通データはホスト・コンピュータで収集しておき、必要に応じ分散処理用コンピュータで加工し情報を取り出す)。

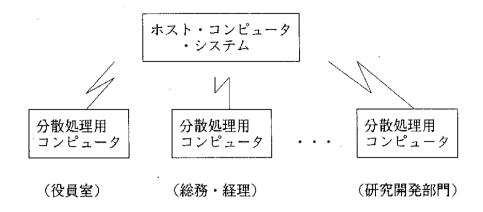
などがある。

② 地域分散処理システム

地理的に分散配置されたコンピュータ・システムを通信回線で有機的に結んだシステムであり、特定の企業、機関に限定したシステムと不特定の機関に開放されたシステムがある。

例:大規模オンライン・システム(銀行システムなど)を複数のセンタで運用するシステム

図表5-1-3 役員室() A システムの構成例



### 育成方法および指導上の留意点

第1章の内容を指導する場合、次の点に注意して指導することが大切である。

- 1. 基本的な処理方式(バッチ処理、リアルタイム処理、分散処理)を理解させる場合単なる暗記ではなく、なぜ、その方式が必要であるか、対象とする 業務の特性と併せてその必然性を理解させる。
- 2. 上記の処理方式と通信回線のかかわり合いをしっかりと理解させる。たと えば、バッチ処理でも通信回線を用いる場合があることなども、整理して理 解させることが大切である。
- 3. 実際のシステムを例として、適切な図表またはVTRを用意して、視覚を 用いて教育すると効果がある。ただし、実際のシステムは本文中でも述べた が、各基本処理方式が組み合わされて実現されていることが多い。指導に当 たっては、この点を整理することが必要である。

# 用 語

バッチ処理方式(一括処理方式), リアルタイム処理方式(即時処理方式), 分散処理方式, オンライン・システム, オフライン・システム, センタ・バッチ処理方式, リモート・バッチ処理方式, オンライン・リアルタイム処理方式, タイムシェアリング処理方式, コンピュータ・ネットワーク, データ通信, データベース, 照合応答形式, メッセージ交換形式, 対話形, 会話形, 電子メール

### 第2章 システムの構成

### 学習目標

この章ではシステムの構成方法について学ぶ。この章を修了した時点で、次 のことがでることを目標とする。

- 1. コンピュータの信頼性を向上させるために多重化が必要なことを理解し、 簡単な稼働率の計算ができる。
- 2. 分散処理のねらいと効果について記述でき、実際の分散処理システムに ついて、その構成要素を簡単な図表で表現できる。
- 3. システム統合化のねらいについて簡単な実例をもとにして述べることができる。

# 内 容

### 2.1 システム構成の種類と特徴(分)

コンピュータ・システムのハードウェア構成は、信頼性、経済性、運用のし やすさなどの観点から、いろいろな形態が存在する。ここではとくに信頼性の 観点から、システムの多重化構成について指導する。

(1) シンプレックス構成

シンプレックス構成のシステムは、システムを構成する各ユニットが一 重で構成されている形態である。

したがって、CPU、周辺装置をそれぞれユニットと考えれば、いずれ かが故障してもシステム全体は機能しなくなる。

いいかえれば、各ユニットは直列に接続されていることになる。

MTBF(平均故障間隔;Mean Time Between Failure)と

MTTR (平均修理時間; Mean Time To Repair ) を用いて

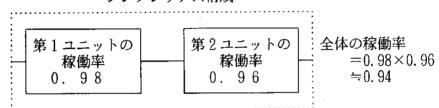
このシステムの稼働率を表すと、

稼働率 = 
$$\frac{\text{MTBF}}{\text{MTBF+MTTR}}$$
 となる。

故障は必ず発生するので、稼働率は必ず1以下となる。

たとえば図表5-2-1で示すように2つのユニットからなるシンプレック ス構成で考えた場合、稼働率をそれぞれ0.98、0.96と仮定した場合は全体 の稼働率は0.98と0.96を乗じた値(約0.94)となる。

図表5-2-1 2 つのユニットからなるシンプレックス構成の稼働率 シンプレックス構成



### (2) 多重化並列化の必要性

シンプレックス構成で稼働率についてふれた。実際のシステムでは所定 の稼働率を保持するため必要に応じて各ユニットを多重化する。

シンプレックス構成の部分で、2つのユニットを直列に結んだ場合の稼働率を試算した。一般に直列に接続された場合は、全体の稼働率は、各ユニットの稼働率を乗じた値になる。

これに対し並列に接続された場合(多重化した場合)は、並列に接続されたユニットのうち1つでも正常であれば、稼働しているとみなすことができる。したがって、全体の稼働率は並列に接続されたユニットがすべて故障した場合を除いた確率を求めることと等価になる。

直列接続、並列接続(多重化)の場合の稼働率は各ユニットの稼働率を それぞれAi ( $i=1,2,\cdots,n$ ) とすれば、

(a) 直列接続の場合

全体の稼働率 A=A1 ×A2 × ··· ×An

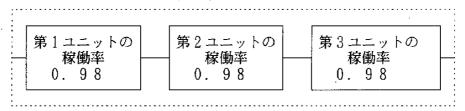
(b) 並列接続の場合

全体の稼働率 A=1-(1-A1)×(1-A2)× ··· ×(1-An)

となる。図表5-2-2 に、A1 = A2 = A3 = 0.98の例を示す。

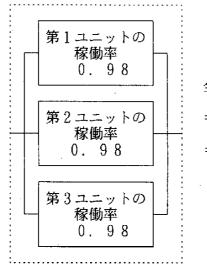
図表5-2-2 稼働率の計算例

### 直列接続



全体の稼働率 A=0.98×0.98×0.98=0.94





### 全体の稼働率

# (3) CPUの多重化

システムの信頼性、可用性、効率性を高めることが主とした目的でCPUを多重化する。とくに、リアルタイム処理方式では、信頼性、可用性を目的としてCPUを多重化することが多い。

# ① デュプレックス構成

CPUの構成を2重化しておき、一方が故障したときは、他方に切り 換え運転する。通常、主系はリアルタイム処理、従系はバッチ処理を行 っており、主系に故障が発生した場合は、従系でリアルタイム処理を続 行する。

### ② デュアル構成

2台のCPUで同一処理を行い、常に結果を照合しながら進める。故 障が発生した場合は1台で処理を続行する。

デュプレックス構成は、故障が発生してもできるだけシステムの稼動 率を上げようという目的であり、デュアル構成は、処理の正確性に重点 をおいている。

③ マルチ・プロセッサ構成

科学技術計算の処理などでは複数のプロセッサが負荷を分散しながら 並列処理を行う。目的は高速な処理を行うことにある。

### (4) 周辺装置の多重化

CPU部分と異なり、周辺装置は、機械的な要素も加わるのでCPUに 比べて故障の率も高い。ディスク、磁気テープ、プリンタなどをはじめ、 各周辺装置の多重化が必要となる。

周辺装置の多重化は、必ずしも同一装置どうしだけではない。たとえば、 ラインプリンタの代替として、磁気テープや磁気ディスクが用いられる。

### 2.2 分散処理(分)

分散処理について指導する場合は、なぜ分散処理を必要とするかという点と、 何をどのようにするかという点について整理し指導する。

- (1) 集中処理から分散処理へ
  - ① 集中処理システム

第3世代コンピュータの時代は高性能プロセッサ、大容量記憶装置、 オペレーティング・システムの発達により大型機のコスト・パフォーマ ンスが高まった。これにともない、情報を集中管理する形態が好まれ、 コンピュータやデータベースをセンタに集中させデータ通信技術により 遠隔地からサービスを受ける集中処理システムが発達した。

② 集中処理システムの問題点 集中処理システムは、コンピュータの性能が価格の2乗に比例すると いうグロッシェの法則があてはまる間は積極的に支持されていた。しかし、システムの規模が大きくなるにつれ、次の問題が発生しはじめた。

- (a) 通信回線の利用量の増大
- (b) CPU負荷が大きくなりすぎ、所定の時間内に処理できない。
- (c) 業務が大きくなりすぎて、エンド・ユーザの要求に対処できない。
- ③ 分散処理への移行

集中処理システムの問題点が明らかになりはじめた頃、高性能ワーク ステーションが出現し、必ずしもグロッシェの法則が成立しなくなった。 これを機会に、集中処理から分散処理への移行がはじまった。

- (2) 分散処理システムのねらいと効果
  - ① 経済性の向上
    - (a) 高性能ワークステーションの活用により、コスト・パフォーマンス を向上させる。
    - (b) 分散処理用コンピュータで処理を分担することにより、ネットワーク系の通信コストの削減をはかる。
  - ② 利便性の向上
    - (a) 分散処理用コンピュータを利用者側に近づけることにより、マン・マシン・インタフェースを向上させることができる。
    - (b) 機能の追加変更が容易になり、利用者の要求にこたえやすくなる。
  - ③ 信頼性の向上
    - (a) 適切な構成を設計することにより、一部の故障に対しても他の部分で処理を代行させることができる。
    - (b) やむを得ず一部の機能を縮小して実行しなければならない場合でも、 集中処理システムに比較して影響を最小限にとどめることができる。
  - 一方、留意点として、適切な分散処理システムの構成には高度な技術を 必要とすること、および安易に構築するとシステムが不統一になる危険性 がある。
- (3) 分散処理システムの分類

分散処理システムの分類の仕方には、いくつかの方法がある。ここでは、 何を分散するかという観点と、接続形態から見た観点からの分類について ふれる。

- ① 機能分散と負荷分散
  - (a) 機能分散

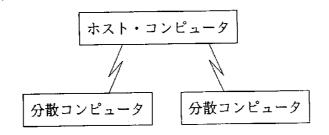
システムの機能を複数のコンピュータが分担して実行できるように システムを構築する。たとえば、大型ホスト・コンピュータとそのコ ンピュータの下で制御をうけ、特定の業務を担当する分散コンピュー ータとの関係など。

### (b) 負荷分散

同一機能を複数のコンピュータ・システムで分担して実行できるようにシステムを構築する。それぞれのコンピュータ・システムは対等 の関係で、相互の負荷を分担する。

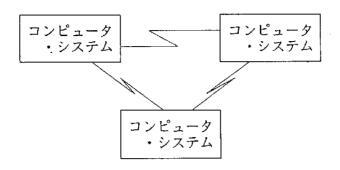
- ② 垂直分散と水平分散
  - (a) 垂直型分散システム:図表5-2-3のように、複数台のコンピュータ・システムを上位レベルと下位レベルに分け、 階層的にネットワーク化したもの。 上位レベルが全体を制御し下位レベルが特定の機能を行う。

図表5-2-3 垂直型分散システム



(b) 水平型分散システム:図表5-2-4のようにそれぞれが単独でも機能できる複数のコンピュータ・システムを規模の大小ではなく論理的に同一のレベルでネットワーク化した形態である。

図表5-2-4 水平型分散システム



### 2.3 システム統合とそのねらい(分)

システム統合については、システム統合をささえる技術と統合の必要性について指導する。

(1) ネットワーク技術、データベース技術の発達

システムが多様化、複雑化するにしたがって、いくつかのシステムを統合し利用することが多くなった。統合システムの実現に貢献したのが、ネットワーク技術とデータベース技術の発達である。

1つの例として、統合化OAシステムをとりあげてみる。図表5-2-5は、 統合化OAシステムの発達段階を3つの部分に分けて表わしたものである。

図表5-2-5 総合化〇Aシステムの発達段階

	利 用 状 況	使われているいる技術
第1段階	オフィス・コンピュータ、ワード・プロセッサ、ファクシミリなどを単体で 導入し、活用する。	(スタンドアロン)
第2段階	LAN(Local Area Network)などにより各機器を接続し活用する。一部ホスト・コンピュータ等の接続も可能となる。	(データ形式の統一) ネットワーク化
第3段階	全社データを総合的に利用できるデータベースを構築し、ネットワークを組む。社外へのアクセスも可能とする。	ネットワーク化 データベース化

図表5-2-5からわかるように、第1段階から第2段階への移行はネット ワーク技術がキーポイントであり、第2段階から第3段階へはデータベー ス技術がキーポイントである。

第2の例として、研究部門の情報を統合化する目的のラボラトリ・オートメーションを取り上げてみる。ラボラトリ・オートメーションに要求さされる機能としては、

- ① 研究開発部門の情報を蓄積するエンジニアリング・データベースをもっている
- ② 各部門のシステムを有機的に結びつけるネットワークがある。
- ③ 研究所のオフィス業務を効率よく実現できる。

などがあり、ラボラトリ・オートメーションの場合もネットワーク技術、 データベース技術が重要なポイントであることがわかる。

(2) システム統合化のねらい

総合化OAシステム、ラボラトリ・オートメーションの2つのシステムからわかるように、システム統合化の狙いは、

- ① 企業間の競争激化にたえる情報を得る。
- ② 新技術、新製品の情報を適切に得る。
- ③ 業務処理量の増加に対処する。

などにある。とくに最近は、大規模な統合化システム(戦略情報システム) の構築が進んできている。

# 育成方法および指導上の留意点

第2章の内容を指導する場合、次の点に注意する必要がある。

- 1. システムの構成の種類(シンプレックス、デュプレックス、デュアル等) を指導する場合、暗記方法ではなく、必ずその理由を理解させることが必要 である。このために、簡単な稼動率の計算を行わせることは、よいことであ あり、是非演習をさせたい。
- 2. 分散処理システムは、歴史的な事実(集中処理、集中処理の問題点、分散処理へ)から入ると理解しやすい。ただし、分散処理システムは必ずしも垂

直型、水平型という単純な形で存在するわけではなく、両者が複雑ににから み合ったシステムも多い。指導にあたってはこの点についても整理しておく こと。

3. システムの統合化については、その必要性から理解すべきである。しかし、 学習者に実務経験がないと、なかなか必要性という点を理解してもらうこと はむずかしい。したがって、学習にあたっては、本節の順序のように、技術 の発達から入った方がよい場合もある。

### 用 語

集中処理システム、分散処理システム、シンプレックス構成、デュプレックス構成、デュアル構成、マルチプロセッサ構成、多重化、結合化、稼動率、MTBF(平均故障間隔)、MTTR(平均修理時間)、水平型分散処理システム、垂直型分散処理システム、直列接続、並列接続、ネットワーク技術、データベース技術、機能分散、負荷分散、オフィス・オートメーション、ラボラトリ・オートメオション

# 第3章 マルチメディア処理

# 学習目標

コンピュータ利用技術の発達により、情報処理システムで扱う情報も文字だけから、図形、画像、音声、映像へと多様化してきた。

この章では、これらの情報を統合してとり扱うマルチメディア処理について 学習する。この章を終了した時点での到達目標は、次の通りである。

- 1. マルチメディア処理の種類及びその特徴を列記できる。
- 2. マルチメディア多重化装置について理解できる。
- 3. 図形処理、画像処理、音声処理の簡単な原理、応用例がわかる(技術用 語を理解できる)。
- 4. マルチメディアの統合について、その必要性を述べることができる。

# 内 容

3.1 コンピュータにおけるマルチ・メディア処理(分)

技術の発達にともない、できるだけ人間に近づけたメディアの開発とメディアの統合が大変なテーマであること、および、そのために解決しなければならない問題点を整理して指導する。

(1) マルチ・メディアとは

コミュニケーションの手段としてメディアを用いる。メディアの例としては古くから音声や文字が用いられてきたが、情報処理技術の発達により、 画像なども加わり、多様化してきた。

- ① メディアということの意味として2つの観点がある。
  - (a) 意味メディア : 人間が知覚できる表現形態としてのメディア
  - (b) 通信メディア : 伝送する方式としてのメディア
- ② ①の観点でマルチ・メディアを分類すると次の2つに分けられる。
  - (a) 複数の意味表現を持つ通信メディア

例:ビデオテックス(文字、画像)

(b) 複数の通信メディアの複合したマルチメディア通信

例:パソコンとファクシミリ間の通信 音声装置付データ端末間の通信 ビデオテックス間の通信

(2) マルチ・メディア処理

マルチ・メディア処理を実現するためには、マルチ・メディア装置とマルチ・メディア・データベースが重要となる。なお最近は、マルチ・メディアの統合も重 要なテーマとなっている。

- ① マルチ・メディア装置マルチ・メディア情報を扱うネットワークを構成するためのハードウェアの条件としては、
  - (a) 高速ディジタル回線
  - (b) マルチ・メディア多重化装置
  - (c) マルチ・メディア複合端末 が必要となる。
- ② マルチ・メディア・データベース マルチ・メディア情報を記憶管理するためには、種々のマルチメディ ア情報に対応すること、大容量記憶が可能であること等が要求される。
- ③ マルチ・メディアの統合 コンピュータを人間に近づけるためには、文字、図形、画像、音声、 映像などを統合した情報伝達がポイントとなる(マルチ・メディアの統合)。マルチ・メディアの統合を実現するためには、これらのメディア を多角的、統一的に扱う必要があり、次の2つが研究の中心となってい
  - (a) どのようにして人間の目、耳、口に近づけたメディアを実現するか。
  - (b) それらで伝えられる情報をどのようにコンピュータの中にたくわえ るか。
- 3.2 図形処理の技術と応用(分)

る。

コンピュータ・グラフィックスの全体像とCAD(Computer Aided Design)

システムについて指導する。

- (1) コンピュータ・グラフィックス
  - ① 歷 史

コンピュータ・グラフィックスの歴史は1960年代にCRT(Cathode Ray Tube)とライトペンを用いて、コンピュータと対話しながら図形を処理するステムからはじまる。以降、1975年頃までコンピュータ・グラフィックスに関する研究が盛んに行なわれ、技術が確立してきた。

1979年、ACM(American Computer Machinary)が標準化されたCO REシステムを提案した。更に1984年、国際規格としてGKS(Graphical Kernel System) が認められた。

- ② ハードウェア
  - (a) 中央処理装置

大型コンピュータ、パソコン、高性能ワーク・ステーションが使用 されるようになった。

- (b) 入力装置 ライトペン、タブレット、マウス等
- (c) 出力装置 グラフィック・ディスプレイ、X-Yプロッタ等
- ③ コンピュータ・グラフィックスの適用分野
  - (a) CAD/CAMシステム

CAD(Computer aided design)はコンピュータ支援による設計であり、CAM(Computer aided manufacturing)はコンピュータ支援による装置のことである。

CAD/CAMシステムは、2次元、3次元図形をコンピュータで 取り扱い、同時に必要な計算も行なうことができる。

- (b) ビジネス・グラフィックス コンピュータで処理したデータを、グラフやチャートの形式でわか りやすく表現する方法で、OAシステムなどで積極的に利用される。
- (c) 科学技術計算、シミュレーション結果の表示 地震の解析、流体の解析などの結果を図形化して表示する。

(d) その他 コンピュータ・アニメーションなどの分野もある。

# (2) CAD

CAD(Computer aided design) はコンピュータ支援による設計を示し、工業製品、建築構造物などの設計過程にコンピュータを利用することにより効率化を図る技術である。

- ① CADとして確立している領域は
  - (a) 計算量が多く計算技法が確立していて、コンピュータ処理が有利な 分野

例:建築構造物の解析

- (b) 細密で膨大な量の要素から構成されるパターンの生成、修正作図 例: VLSIなどの回線設計や実装設計
- (c) 多量の図面とそれにもとづく数値制御加工のような、生産設計の関わるもの

例:航空機、自動車などの形状設計、図面管理など

- ② CADシステムの基本技術
  - (a) 図形の入力

入力は、ライトペン、マウス、タブレット等を用いて対話的に入力 する方法と言語によって一括記述する方法がある。

(b) 操作・変換

入力された図形に関するデータは、適切な座標表現と座標変換マト リックスを用いると拡大、縮小、回転、平行移動などの操作を行なう ことができる。

(c) 表 示

表示部分はできるだけハードウエアから独立、標準化しパッケージにしたい。現在、ISO(International Organization for Standardization) でも検討され、GKSという国際規格として採用されている。

- (d) 幾何モデル
  - 3次元モデルとして次の3種類のレベルがある。
  - ・ワイヤフレーム・モデル
  - ・サーフェス・モデル

### ・ソリッド・モデル

### 3.3 画像処理の技術と応用(分)

画像情報を加工し、人間の目で識別しやすい画像に変換する処理を画像情報 処理と呼ぶ。画像情報処理について指導する場合は、基本的な考え方と応用例 について理解させる。

- (1) 画像処理の考え方と技術
  - ① 画像処理の種類

画像情報処理には、アナログ式(写真、光学的、電子的に処理)とディジタル式(コンピュータで処理)がある。最近はディジタル式が多い。

② 画像の取り込み 画像の入力は、イメージ・スキャナ等の画像入力装置でシステムに取 り込む。

③ 画像の処理方式

ディジタル式で画像処理する場合、画像を画素に分解したあと、画素に対してコンピュータ処理することにより、画像全体を処理する。わかりやすい画像を作るための画像処理技術としては、次のものがある。

- ・カラーおよびコントラストの調整
- ·拡大処理
- ・3次元画像の作成
- (2) リモート・センシング

人工衛星や航空機あるいはレーダを用いて、センサにより、地球を調査 する技術がある。リモート・センシングのシステムとして、

- ・人工衛星ランドサットからの情報で地球の状況を調査する。
- ・気象衛星からのデータをもとに雲の動きを知る。

などがあり、いずれも観測データをもとに画像処理を行ない求めるデータ を画像として手に入れる。

(3) 医療画像処理

医療関係の画像処理として、次の2つのステップがある。

① 人体内部における各臓器の形態、生理機能などの情報や人体の表面温

度分布などを画像化する。

② 画像化されたものに対し更にコンピュータ処理を施し医療に役立てる。 医療画像処理装置としてはCT(Computeriged Tomography) スキャナが ある。CTスキャナはX線、超音波などとコンピュータを組み合わせ人体 断面図を画像として表現する装置である。

### 3.4 音声処理の技術と応用(分)

音声処理の考え方、応用例について指導する。この場合、原理的なことにまで深く指導する必要はない。

(1) 音声処理の考え方と技術

音声処理は、すぐれたマン・マシン・インタフェースの1つであり、音声合成と音声認識の2つの面をもつ。

① 音声合成

コンピュータにより、人工的に音声を合成する方式であり、電話番号 案内、交通案内、さらには家電品にまで応用されてきている。

② 音声認識

音声合成と異なり、音声の認識は技術的にむずかしい。AIを用いた研究開発が進められている。

現在、音声認識のための条件としては、

- (a) 話し手に関する条件
  - ・特定の話者(話者をあらかじめ登録する)
  - ・不特定の話者
- (b) 発音方式についての条件
  - ・単語を1つずつ区切って発音する
  - ・単語を区切らなくてよい(文章、会話と同じ形式)
- (c) その他 認識できる語の数の条件 などがある。

現在、実用化されているのは特定の話者が単語を区切って発音する方式で、物の仕分け、工場検査システムなどで用いられている。

不特定の話者を対象としたシステムは、個人差を取り除いて理解する

必要があるため、現時点では数10から数100語位のものが認識できているにすぎない。

(2) 音声入出力システムの応用例

音声入出力システムは、知能ロボットのように音声を認識し、情報処理をした後音声合成で応えることが最終的な目的である。しかし、技術的には未開発の部分が多いので、当面は入力装置、出力装置の一部として、適切な音声認識装置や音声合成装置を用いて、良好なマン・マシン・インタフェースを実現することにおいている。

### 育成方法および指導上の留意点

第3章の内容を指導する場合、次の点に注意する必要がある。

- 1. この章の内容は、それぞれの分野が大きな研究分野である。したがって、いたずらに各論に入りすぎないように指導することが大切である。
- 2. とくに、図形処理、画像処理、音声処理の分析については、初級情報処理 技術者としては基本的な考え方、実際の応用例を理解するだけで十分である (専門分野として選んだ場合は、その分野について別途指導すればよい)。
- 3. 指導にあたっては、できるだけスライドやパソコンなどを利用し、具体的に示して指導することが大切である。とくに、図形処理分野は、パソコンで動作するシステムが多数存在する。実習を通して教育したい。

# 用語

マルチ・メディア、図形処理、画像処理、音声処理、コンピュータ・グラフィックス、CAD/CAM、リモート・センシング、コンピュータ・アニメーション、ビジネス・グラフィックス、パターン認識、医療画像処理、音声合成、音声認識、意味メディア、通信メディア、ビデオ・テックス、高速ディジタル回線、ドット、CRTディスプレイ装置、ウインドウ機能、ワイヤフレーム・モデル、サーフェイス・モデル、リソッドモデル、X線CTスキャナ、マルチ・メディア・データベース、イメージ・スキャナ、マン・マシン・インタフェ

# 第4章 AIシステム

# 学習目標

AI(人工知能;Artificial Intelligence)の研究は、ゲームの研究に始まり現在に至っている。エキスパート・システム、ファジィ・コンピュータの実現で、AIシステムが身近になってきた。

この章では、AIについての学習を行い、修了時点で次のことができるようになることを目標とする。

- 1. A I の基礎理論(探索、知識表現、推論、学習)についての基礎的な考え方を理解し、探索木などの簡単な技術用語について説明できる。
- 2. A I の応用としてのエキスパート・システム、自然言語処理などに関して簡単に説明できる。
- 3. ファジィ・コンピュータ、ニューロ・コンピュータなどの、最近の動向 についての雑誌や新聞記事が理解できる。

# 内 容

# 4.1 Alとは(分)

この節では、AIの基礎理論について指導する。

(1) 考えるコンピュータ

AIの研究は、人間の知識をモデル化して、コンピュータ・システム上で実現することである。

AIの研究は、

- ① チェッカ・プログラムに関する研究、神経モデルの提案、LISPの 開発が行われた1950年代~1960年代の第1段階。
- ② 導出原理の提唱、自然言語理解システム、化学構造式を推定するエキスパート・システムの開発、Prologの開発などが行われた1960年代~1975年代の第2段階。
- ③ 1975年以降実用化に向けて動き始めた第3段階。

の3つの段階に分けて考えることができる。

## (2) A I の基礎

# ① 探索

探索という言葉は、推論、プランニングおよびそれらに関した処理方法の概念として用いられる。例えば、ゲームにおいて、最善の手を求めることや、現時点から目標点までの最適経路を求めることなどは、代表的な探索の適用領域である。探索の技術的なポイントは、次の通りである。

- (a) 対象となる問題を表現する方法として、状態空間表現法と問題分割 表現法がある。
- (b) 問題は探索木の形で表現できる。探索木には、節があり、AND節、OR節がある。AND節、OR節で構成される探索木をAND/OR木と呼ぶ。
- (c) 探索木をたどりながら推論するが、推論には、前向き推論と後向き 推論がある。
- (d) 探索木が大きくなると、探索に要する組合せの数が実用の範囲を越 えてしまうことが多い。これを避けるため、ヒューリスティクスの手 法を用いる。

# ② 知識表現

A I システムでは、知識を何らかの形で形式的に表現し、その知識を用いて推論する。知識表現は、A I システムのかなめであり、次の点が大切である。

- (a) 目的を明確にした知識表現形式
- (b) 知識表現は、知識とアクセス法から成る。
- (c) 知識表現は、記号と意味の中間媒体である。

以上のことから、記憶の効率やアクセスの容易さ、推論における使い やすさ、明確な意味論などが知識表現の条件となる。

現在のところ、上記の要求をすべてみたす知識表現はないが、代表的な知識表現として、

(a) 述語論理による知識表現

- (b) プロダクション・ルール
- (c) 意味ネットワーク
- (d) フレーム

などがある。

## ③ 推論

ある論理に基づいて、前提から結論を導き出す過程を推論と呼ぶ。

(a) 推論の種類

AIの分野では、演繹推論、帰納推論、仮説推論が知られている。

- 演繹推論
  - 一般的な法則から個別事実を導く推論であり、三段論法はこれにあたる。
- ・帰納推論 個別事実の集まりから一般法則を導く。
- 仮説推論

観測された事実を説明できる仮説を、仮説集合の中から選択、推 論する。故障診断などはこれにあたる。

- (b) 知識表現と推論
  - ・述語論理による推論

 $\forall x \ \forall y \ \forall z \ [ 上(x,y) \land L(y,z) \rightarrow L(x,z) ]$  というルールがあるとする。このルールは「xがyの上に、yがzの上にあればxはzの上にある」という位置についての知識を示し、x, y, zが与えられたとき、xとzの間の関係を推論する。

・プロダクション・ルールによる推論

プロダクション・ルールは、一般にIF A THEN Bの形式で表現する。ここでAは条件を示し、Bは結論を示している。推論の手順としては、プロダクションルールの条件部にマッチするものを選び、ルールに含まれる条件を推論し、結論を導き出す。

プロダクションルールの特徴は

人間にとってなじみやすい型である。

- ・部分的に表現できる(メンテナンスが楽である)。
- ・ルールの適用法を工夫すると、効率的な推論を行なうことが できる。
- ・その他

### ④ 学習

A I システムにおける学習とは、次の2点を意味する。

第1は、入力情報に基づいて推論した結果をシステム内部に取り込む こと。第2は、経験したことを知識表現を用いて表現、改良することで ある。

A I システムにおける学習は、次の3つに分類される。

- (a) 暗記学習
- (b) 帰納的学習
- (c) 類推による学習

#### 4.2 Alの応用(分)

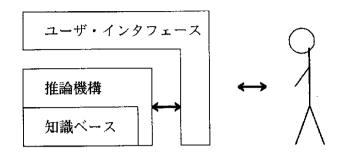
この節ではAIを応用したシステムについて指導する。

- (1) エキスパート・システム
  - ① 概 要

特定の問題領域において、その分野の専門家の知識を取り込むことにより、専門家と同等の問題解決能力を実現させようとするシステムをエキスパート・システムと呼ぶ。

エキスパート・システムは、従来の情報処理システムと異なり、知識ベースと推論機構が明確に区分される(図表5-4-1)。両者が、明確に区分されるので、知識ベースを入れ換えることにより、多様な分野の問題にも対応できるようになる。

## 図表5-4-1 エキスパート・システムの構成



## ② エキスパート • ステムのタイプ

エキスパート・システムは、問題解決の手法に対応して解析型と合成型に分けることができる。

### (a) 解析型

与えられた事実やデータからうまく説明できる仮説を選ぶ型のシステムであり、病名診断、法律相談などのシステムがある。

## (b) 合成型

一定の拘束条件のもとで与えられた要求を満たす最適な解を見つけようとする型のシステムであり、解析型のように考えられる結論を用意しておくことはできない。解答は、要素を組み合わせて合成することになる。

例としては、LSIや電子回線の設計システムがある。

# ③ 知識の獲得

エキスパート・システムの問題解決能力は、知識の質と量で決まる。 大量に効率よく質のよい専門知識を獲得しなければならない。

知識の獲得のためには、

- (a) 対象になる分野から知識を抽出、分析する。
- (b) 機械が実行できる形に変換する。
- (c) 得られた知識を知識ベースに組み込む。

といった作業が必要であり、知識獲得のためのエキスパート・システム も開発されている。

## (2) 自然言語処理システム

マン・マシン・インタフェースという観点で、自然言語処理システムの 意義は大きい。本項では、自然言語処理の手順と機械翻訳システムの研究 経過について学習する。

# ① 自然言語処理の手順

## (a) 形態素解析

1つ1つの単語にかかわる処理を形態素解析という。英文と異なり 日本文の場合は、わかち書きを行わないので最長一致法、最少文節法 などで文から単語を切りはなす処理が必要である。

## (b) 構文解析

与えられた文を解析し文法的に正しいかどうかを調べる。正しい文の場合は、その文の構造を求める。コンピュータによる解析には、文脈自由型句構造文法(CFG;Cntext-Free phrase structure Grammar)を用いることが多い。

# (c) 意味解析

意味解析の手順としては、次の方法がある。

- ・単語の持つ意味を詳細に記述する。
- ・単語の持つ意味を階層関係で記述する。
- ・動詞を中心とした格文法を用いる。

# (d) 文脈解析

前方照合スクリプト理論などにより行間の脈絡の推定を行う。

# ② 機械翻訳システムの研究経過

機械翻訳システムの研究は、

# (a)第1世代

原始言語文字列を直接目的言語に変換する

# (b)第2世代

原始言語と目的言語の間に中間的な言語表現のレベルを設定した。 この方式には、中間言語方式とトランスファ方式がある。

#### (c)第3世代

AI技術を取入れ、原始言語の解析を行った後、目的言語の文章

構造を生成する方法で翻訳を行う。

以上3つの世代がある。第1・第2世代の翻訳は、文単位の翻訳であるが、第3世代の翻訳は文章単位で行えるようになった。

- (3) コンピュータ・ビジョン
  - ① コンピュータ・ビジョンとは

コンピュータ・ビジョンは、機械に人間の持つ視覚機能を付け加える ことをいい、画像処理やパターン認識と関わりあいが深い。

コンピュータ・ビジョンの目的を一言でいうと、「コンピュータによる視覚の実現」であり、「2次元画像によるシーン(3次元世界)の理解」である」

人間は一瞬にして

- ・多くの物体の中から必要とする対象を見つけだす
- ・対象の形状を知る
- ・物体の遠近感をつかむ
- ・静止物体と移動物体を区別する
- ・対象をより細かい物体に分割する

などの能力がある。これらの能力をコンピュータにより実現することは、 多くの分野で非常に有益である。

# ② 処理手順

コンピュータ・ビジョンのシステムは、大きく分けると初期処理と認 識・解釈の2段階になる。

- (a) 初期処理(信号処理のレベル): 画像から線また面を見つける(画像処理、パターン認識の技術 が使われる)。
- (b) 認識・解釈(記号処理のレベル):

初期処理の結果からシーン(3次元世界)の解釈(線の意味づけ、面の意味づけ、物体の切り出しの構成など)を行う。

解釈時には、線の見え方・面の見え方に関する知識、物体の構成に関する知識などが必要となる。このため、人工知能の応用分野として議論されることが多い。

### (4) 知能ロボット

1960年エルンスト(H. A. Brnst)は、マニピュレータと、それを制御するコンピュータ、および外部情報をとりいれる知覚センサを結合した知能ロボットを作成した。

以降、知能ロボットに関する研究は、3つの方向に向かった。

- ① 作業計画やスケジューリングなどの知能に関する研究
  - ② コンピュータによる視覚情報処理の研究
  - ③ 触覚や力覚の利用を含むマニピュレータの研究

A I と直接関連する部分は①の自動組立ての問題、および②のコンピュータ・ビジョンの部分である。

## 4.3 A I システムの技術動向(分)

この節では、AIシステムの動向としてファジィ・コンピュータとニューロ・コンピュータを取り上げる。ファジィ・コンピュータは、実用化の域に入っていること、ニューロ・コンピュータは、次の時代のコンピュータであることを指導する。

#### (1) ファジィ・コンピュータ

1965年ザデー(L. A. Zadeh)により提案されたファジイ理論に基づいて、制御することを目的としたコンピュータである。ファジィとは、「あいまい」ということであり、従来の制御方法と異なり、いくつかの制御要素にに対応するメンバ・シップ関数を定義し、それらのメンバ・シップ関数からファジィ理論にもとづいて推論する。

人間の行動は「少し…」という表現で表せることが多い。メンバ・シップ関数は、この「少し…」という表現を適切に表現できるのでファジィ理論に基づいた交通制御システムが実用化されたり、ファジイ・コンピュータを搭載した家庭電化製品が多数出現してきている。

### (2) ニューロ・コンピュータ

ニューロ・コンピュータは、人間や生物の脳、神経回路網の機能を手本 として考え出された計算モデルを、電子回路や光回路で実現したコンピュ ータである。ニューロ・コンピュータは、

- ① 従来の数値計算、記号処理中心のコンピュータ・システムでは解決できない問題を新しい素子ニューロンを用いて実現する。
- ② 多数のニューロンを並列動作させ高速演算を行う。
- ③ 各々のニューロンを適切な形で動作させるためのネットワーク、人工 神経回路網をはりめぐらせている。
- ④ 従来のコンピュータのようにプログラムで動作するのではなくネット ワークの結合形態に知識が蓄積され、神経回路網が自ら動作するシステムである。

### などの特徴をもつ。

ニューロ・コンピュータは以上のような理由で文字、音声などの認識、 自然言語理解、人工知能の分野などでの活用を期待されている。

## 育成方法および指導上の留意点

- 1. A I の基礎理論については、基本的な事項について簡単な例題演習をもとに、その考え方について指導する。この場合、各論に深く入り込ませないように注意することが大切である。
- 2. A I の応用の部分ではVTRを用いたり、時間が許せばシステムの見学など、実物にふれる機会を作って指導したい部分である。特に、エキスパート・システムはパソコン上で動作するツールが出ているので、実習を行うことをすすめる。
- 3. ファジィ・コンピュータ、ニューロ・コンピュータなどの最新技術では、 新聞の切りぬき等を提示し、学習者にまとめさせることもよい指導になる。

# 用語

人工知能(AI),探索,知識表現,推論,エキスパート・システム,自然言語処理システム,コンピュータ・ビジョン,知能ロボット,ファジィ・コンピュータ,ニューロ・コンピュータ,ナレッジ・エンジニア(KE),機械翻訳システム(自動翻訳システム),知識ベース,推論機構(推論エンジン),ニュ

ーロン、神経回路網、マン・マシン・インタフェース、メンバ・シップ関数

# 第5章 CAI/CMIシステム

# 学習目標

コンピュータ利用技術は、教育の分野まで発展してきている。この章では、コンピュータ支援による教育システムCAIと、CMIについて学習するとともに今後の方向についても学習する。

この章を終了した時点で、次のことができることを目標とする。

- 1. CAI/СMIシステムの目的を述べることができる。
- 2. CAIシステムの歴史およびその種類について説明できる。
- 3. CAIシステムの現状と課題について説明できる。
- 4. コンピュータ・リテラシィについて説明できる。

# 内 容

# 5.1 システムの機能と目的(分)

この節ではCAI/CMIシステムを中心にコンピュータ支援による教育の全体像を指導する。

- (1) コンピュータ支援による教育の目的 教育の現場でのコンピュータ利用の目的は、次のとおりである。
  - ① CAI(Computer Asisted Instruction)学習者がコンピュータと対話しながら学習を進めるシステムで、直接 教育補助を目的としてコンピュータを用いる。
  - ② CM I (Computer Managed Instruction) 教材研究や授業の設計、分析、評価を目的としてコンピュータを用いる。
  - ③ その他

コンピュータ・システムをネットワーク化し、文献の検索、情報交換などを目的としてコンピュータを利用する。

### (2) CAIシステム

CAIシステムの概要

CAIシステムは、学習者がコンピュータと対話しながら学習を進める方式で、

ステップ1:コンピュータが説明や問題などの教授情報を提示する。 ステップ2:学習者は、コンピュータからの提示をみて理解、応答 する。この場合、必要ならコンピュータあるいはインス トラクタから助言してもらう。

ステップ3:コンピュータはこの応答を評価し、適切な情報を学習 者に返す。

以上の3つの手順を繰り返すことで学習を進めてゆく。

CAIシステムは、教材の部分と教材を使って処理を進める部分とから成る。前者をコースウェア、後者をエグゼキュータと呼ぶ。

② CAIシステムの分類

1960年代にCAIシステムが研究されはじめた。これを伝統的CAIと呼ぶ。伝統的CAIは本質的にいくつかの課題を持っていた。この課題を解決するために2つの研究が発生した。第1は、AI技術を取り入れた知的CAI、第2は、ネットワーク技術とパソコンの技術を取り入れた環境型CAIである。

- (a) 伝統的CAIシステム伝統的CAIは、
  - チュートリアル型
  - ドリル型
  - ・シミュレーション型
  - ゲーム型

の4つの形式がある。伝統的CAIはそれなりに定着、活用されているが、大きな課題がある。第1の課題は、解答照合形式のためあらかじめ答を用意しておかなければならないこと、第2の課題は、学習者が質問できないこと。第3の課題は、コースウェアの作成に時間がかかることである。

(第3の課題を解決するためにオーサリング・システムが開発された が完全な解決には至っていない。)

(b) 知的CAIシステム

知的CAIシステムは、AI技術を用いて伝統的CAIシステムの 課題を解決することを目ざしている。目的としては、

- ・伝統的CAIの第1の課題の解答照合形式から脱却する。および、 第2の課題の学習者の質問に答えられないという制約をとりはず す。
- ・伝統的 C A I では実現できなかった学習者のレベルに合わせ、説明の内容や質問のレベル、進度を変えることができるようにすることを目的にしている。

## (3) CM I システム

**教師がコンピュータを用いて授業の設計、分析、評価を**行なうシステムで、次のようなプログラムがある。

- ①教材の作成、記述、分析
- ②テスト結果の分析、管理
- ③時間割作成
- ④その他

CMIシステムは、現場の先生を中心に個別に開発されたプログラムが多い。統合化された標準システムの開発が望まれる。

# 5.2 現状および今後の動向(分)

この節では、CAIシステムを中心に、現在の普及状況、研究課題等を指導する。

(1) コンピュータ・リテラシィ教育

1970年代から使われはじめたことばである。最近は、初等・中等教育の中でCAIシステムを用いて、コンピュータ・リテラシィ教育が行なわれている。コンピュータ・リテラシィ教育の目的は2つある。

① コンピュータを利用する技術を身につけるコンピュータの操作、適切なアプリケーション・ソフトを選択し、業

務を処理する力をつける。データベースを作成、保持する能力を身につけるなどの目的がある。

- ② コンピュータのしくみを理解する コンピュータ・リテラシイ教育では、必ずしも将来の専門家を育成す ることを目的とはしていない。コンピュータのしくみを理解することに より、より上手にコンピュータを使用することを目的としている。
- (2) CAROLシステム

通産省は情報大学校構想の一環として、1986年より5年計画で情報処理 技術に対するCAIシステムを開発した。これをCAROL(Computer Aided Revolution On Learning)システムと呼ぶ。CAROLシステムは、 専門学校を中心に活用がはじまっている。

(3) 知的CAIの開発

知的CAIの開発にあたっては、次の2点が主要なテーマであり、現在 研究が進められている。

① 教授すべき知識

知的CAIシステムの最大のメリットは、学習者との対話を通じて学習者に最も適切な、問題作成や誤り診断などが推論されることである。 近年、教授知識に関する基礎研究から実用化の研究もはじめられてきた。

② マン・マシン・インタフェースの改良

マン・マシン・インタフェースを改良することにより、CAIシステムとより密接に対話することができれば、より学習者のレベルを理解することができる。このための一つの方法として、自然言語での対話なども重要なポイントである。

# 育成方法及び指導上の留意点

第5章の内容を指導する場合、次の点に注意して指導することが大切である。

- 1. CAIシステムの種類を整理して指導すること。とくに、伝統的CAIに ついては、各種の形式について、特徴、応用例をしっかりと理解させる。
- 2. CAIシステムの長所と制約事項を明確に整理すること。使用にあたって

は長所を生かす形で使用することを指導する。

- 3. 環境型 CAI、知的 CAIについては、現状および今後の動向について理解させること。各論にまで入りこむ必要はない。
- 4. コンピュータ・リテラシィ教育について、その動向を整理し指導すること。

# 用語

CAI (学習支援システム), CMI (学校管理システム), 伝統的CAI, 知的CAI, コンピュータ・リテラシィ教育, CAROL計画, チュートリアル型, ドリル型, シミュレーション型, ゲーム型, コースウェア, エグゼキュータ, オーサリング・システム

## 第6部 情報化とコンピュータ

# 教育目標

90年代における国際的な環境変動の中で、日本の国内の主な潮流として、「高齢化」、「国際化」、「情報化」の3つがあげられる。この日本社会の潮流のうち最も重要なものは情報化であり、それは同時に、他の潮流と変化を促す役割と力をもっていると考えられる。そして情報化の影響は、産業分野のみならず個人、社会、地域の広範な分野に及ぶであろう。

そこで、情報化とコンピュータの部では、

- ① 企業における情報化
- ② 社会・行政・地域における情報化
- ③ 個人・家庭における情報化

を見通し、各分野における情報化の現状と近未来の動向を述べる。

そして望ましい情報化社会の達成に必要な、

- ④ 標準化とその動向
- ⑤ セキュリティとプライバシー
- ⑥ 法制度と基準
- ⑦ 人材育成

についてふれる。

以上の概論や各論の内容をただ単に理解するだけでなく、実務の中でより大局的な視点として把えられるようになることが、最大の教育目標となる。ただ知識として知っているだけでなく、実務の中で使いこなして経験を積み重ねていくことによって、体系だった知識として習得できるようになってもらいたい。その中で、情報化とコンピュータにおける潮流および変化を自分なりに把える意識を常にもっていることが、エンジニアとしての自覚に結びつく。

そのために、講義と同じくらい演習にウエイトをかけ、グループあるいは個人 ごとに課題を解いていく中で実践的な演習ができるようになっている。

# 内容の構成

	,d-, ###	時間数		
	内 容	講義	演習	
第1章 1.1 1.2 1.3	情報化の現状と将来 企業における情報化 社会・行政・地域における情報化 個人・家庭における情報化	1 4	6 .	
	情報化社会の課題 標準化とその動向 セキュリティとプライバシー 法制度と基準 人材育成	1 8	2	
	승 <b>計</b>	3 2	8	

# 第1章 情報化の現状と将来

# 学習目標

当章を受講された方々が、この章を修了した時点で、次の事柄ができること を目標とする。

- 1. 企業における情報化について理解を深めて、70年代、80年代、90年代の潮流と変化が具体的に列挙できるとともに企業システムの現状について説明できる。
- 2. 社会・行政における情報化について社会基盤の整備の立場から現在の情報化の変化について説明できるとともに、具体的社会・行政システムについて説明できる。
- 3. 情報化やエレクトロニクス化が生産の場だけでなく、コストや採算という基準が不明確な消費の世界にも、価値観、ライフ・スタイルの変化を伴い、文化として浸透していく潮流を説明できる。

# 内 容

# 1.1 企業における情報化(600分)

企業の情報化は、90年代に入って一層発展し、産業界の幅広い分野で着実に 進行すると考えられる。この情報化の進展と旺盛な設備投資および研究開発努力と相まって、わが国企業の国際競争力は一段と強まるであろう。さらに、先端的な情報技術の活用による産業の高度化、多角化、融合化が、業種間、国家間の枠を越えて、地球的規模で展開されていくだろう。21世紀へ向けて生き残りを実現するためには、各企業は絶えず環境変化を迅速に認知し、それに対応する意思決定と戦略形成を適切に行い、競争優位を持続的に確保していく必要がある。

- 1.1.1 企業における情報化の特徴
- (1) データベースとネットワーク データベースとネットワークの拡充という観点から企業の情報化を解説

する。

80年代における企業の情報化は、OAブームの洗礼で始まり、パソコンやワープロの急速な普及とともに、情報システムのパーソナル化を促進した。次いで85年の電気通信事業法の施行により通信の自由化が推進され、ISDN、VAN、LANなどが着実に実用化され、コンピュータと通信の融合によるネットワーク時代の幕開けとなった。

消費者ニーズの多様化・高級化に伴い、80年代においても既に多品種少量生産方式への転換が見られたが、90年代では、それを一層徹底させた、いわば個別受注生産方式が確立するであろう。これを実現するためには、EOS、CIM、SISなど情報システムの高度化が必要であり、それらを支える基盤としてデータベース等情報の整備と情報ネットワークの拡充が不可欠である。

# (2) EDI、企業間ネットワーク

EDIの進展という観点から企業の情報化を解説する。

80年代後半から金融、流通業を中心に企業間ネットワークが急速に進展してきたが90年代には、企業間ネットワークや業界データベースの重要性も増大し、全業種にわたって本格的に展開されていくであろう。企業間ネットワークがEDI(Eletronic Data Interchange:電子データ交換)という形で取り上げられており、コンピュータ・ネットワークを介したデータ交換を円滑に行うために幅広く合意される標準的な規約づくりが、各業界や団体によって進められている。こうした規約づくりは、業界の枠を越えて進められ、さらに国際的な整合を指向している。当初EDIは、取引量の多い企業間での受発注処理の効率化を目的とした費用節約的利用であったが、次第に、競争相手との差別化を図る戦略的利用へ、さらには企業の情報化の基盤へと進化しつつある。

# (3) オフィス生産性

オフィス生産性の向上という観点からここ10年間オフィスの生産性をどう把えられてきたか解説する。

OAブームが華やかであった80年代初頭に「オフィス生産性の向上」が よく叫ばれたが、ワープロやパソコンを導入したからといって、直ちにそ れが実現できたわけでない。OA機器の普及は、一面では"紙の洪水"をもたらした。本格的なオフィス生産性の向上は、業務改善や組織革新を行った上で、ネットワークによる分散処理が円滑に機能し、オフィスに働く誰もがOA機器を自由に使いこなせる状態が実現するまで待たなければならない。

## (4) オフィス・アメニティ

オフィス・アメニティの充実という観点から企業の情報化を解説する。オフィス生産性の向上を厳しく追求すれば、その反面、オフィスの非人間化という"影"の部分を増大させる危険性も伴う。オフィス環境も狭い、汚い、うるさいなどのマイナス・イメージから思考の場としての快適さへの転換が求められるようになる。現在建設されている多くのインテリジェント・ビルは、情報インフラストラクチャを備えているだけでなくて、オフィス・アメニティにも配慮している。特に若年労働者の不足が深刻となる90年代には、人材確保の点からも快適なオフィスの建設が必要となろう。都心に位置するビルでは、こうしたオフィス・アメニティの充実が困難な場合も多いので、情報ネットワークを基盤としたオフィスの分散化が促進されるであろう。それらの例として、サテライト・オフィス(住宅地に近い場所のオフィス)、レジデンシャル・オフィス(自宅をオフィスとする)、リゾート・オフィスなどが挙げられる。

# (5) 情報化格差

情報化格差の拡大による歪みを最小限にとどめるための対策について解 説する。

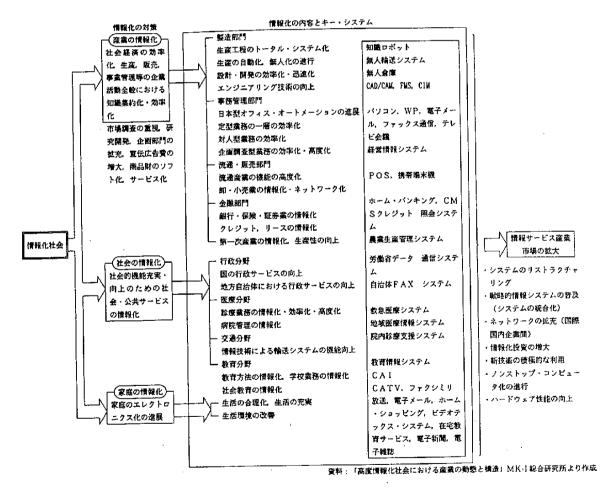
情報化の進展につれ、少なくとも現在の延長として、企業間、業種間の情報化格差は拡大傾向を示すと予想される。情報化を手段とした企業力、経営力の格差の拡大は、SISにより同業他社に差をつけ、競争優位を確保する例に見ることができる。中小企業も業界共同VANの構築を進めており、タテ型、ヨコ型のネットワークに参加する場合のメリット、デメリットを考慮して、より効果的、主体的な選択をしていく必要がある。また情報化の拡大による歪みを最小限にとどめるためには、資源としての情報の有効活用と標準化等によるハード、ソフトの技術面からの支援が不可欠

である。そのためには、インハウス・データベースの整備とともに、誰でも利用できる商用データベースの拡充を図る必要がある。一方利用側としては、真に役立つ情報は何かを見極め、ストック情報(POS; Point Of Sales)などで得られる日々の消費者データ等)がタイムリーに経営にとり入れられるような対策を講じることが肝要である。

# 1.1.2 企業における情報化研究課題(1)

主として70年代、80年代、90年代と三つの時代における企業における情報 化の特徴について調査とりまとめを行い、21世紀の情報化について各グルー プ毎に検討し、意見交換をする。

図表6-1-1 情報化と情報サービス産業市場



#### 1.1.3 戦略的情報システム

商用コンピュータが導入されて以来情報化を推進させるため、MISを始めとしてさまざまなコンセプトが導入され、現在では戦略的情報システム

(SIS; Strategic Information System) が世の中の大きな注目を浴びるようになっている。ここでは、戦略的情報システムを事例に企業の情報化を概観する。

## (1) 今、なぜ戦略的情報システムか

これまでの情報システムは、確かに業務活動を支援する役割を負っているものの、現実的には業務と情報システムとが別個のものとして存在し、 業務活動の中に両者が必ずしも融け込んでいなかった。

これに対して戦略的情報システムは、情報システムを革新的経営戦略や 組織改革などに結びつけた相乗効果を狙っている。つまり、企業の業績向 上や発展に貢献できるよう、企業全体の質的な高度化を図る手段として活 用しようという点に大きな違いがある。戦略的情報システム構築の第1歩 は、①トップや管理者を含め全社員が新しい情報や情報技術の使い方を考 えたり、②社内組織の在り方を変えたり、③トップ・マネジメントや管理 者の意識を変えるといったシステム改善の土壌づくりにあるとしている( チャールズ・ワイズマン)

# (2) 最近の戦略的情報システム 戦略的情報システムの目的を要約解説する。

# ① 競争優位の獲得

市場が成熟し、価格、製造技術、製品の品質といった面でそれほど大きな開きがなくなってくると、差別化の要件としてサービス面が重視されてくる。

# ② 低コスト化

情報技術の発達やコンピュータの低廉化といった好環境を背景に、情報システムの統合化(インテグレーション)というコンセプトが生まれた。そして、SISとかСIM(Computer Integrated Manufacturing)と呼ばれるシステムが登場し、企業活動の仕組みを変え大幅な低コスト化を図る企業も出現してきている。

## ③ 新市場の獲得、囲い込み

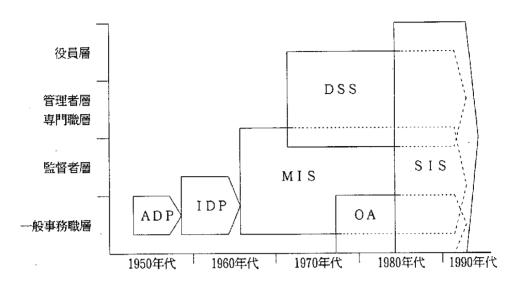
先見性や起業化意識の旺盛な企業では、情報とネットワークの結合に よって発生する多くのメリット(時間や距離の超越)を生かしてニュー ・ビジネスの創出に利用している。

例えば、金融、証券業などでは、すでに第二段階の試みがなされつつ ある。具体的には、国際化に向けてグローバルなネットワークを構築し、 24時間対応の企業活動を可能にし、国際的な企業間競争に勝つための戦 略情報システムを構築している。

図表6-1-2 企業情報システムの変遷

		#A = 1	49 ALLE 20	李母先李士祖(1951	ا بدرد فالشيونون	高を助き入りとは、またり、4 で !
概念	自動データ処理	統合データ処理	経営情報システム	意思决定支援汉孙	#74X+#-}\$-9a9	戦略的情報システム
区分	(ADP)	(1DP)	(MIS)	(DSS)	(OA)	(SIS)
目的	・省力化	• 省力化	・構造的意思決定	・非構造的意思決	・省力化	・差別化
	・コスト低減	・コスト低減	支援	定支援	・コスト低減	・既存事業の質的
ļ	1		'	}	・個人の情報処理	改善
1					能力向上	
適用素務	・トランザクショ	・トランザクショ	・機能別サブシス	・創造的・思考業	・ローカル(身辺)	- 業務問結合
·	ン処理	ン処理の連結	テム統合	₹¥	業務	• 企業間業務結合
]			・要約報告書			
処理方式	・バッチ処理	・オンライン	・オンラインリア	・タイム・シェア	・スタンドアロン	・オンライン・リ
	・非集中	• 集中	ルタイム	リング	(後のネットワー	アルタイム
l			・集中	•分散	ク化)	・ネットワーク
					・分散	• 集中/分散
IS部門	・データ処理サー	・データ処理サー	・システム開発	・情報検索支援	・業務改善・改革	・システム開発
の機能	ピス	ピス	・情報検索支援	・情報処理コンサ	・ユーザ教育	• 戦略立案支援
** 15.00		・システム開発	・アプリケーショ	ルティング	・情報処理コンサ	・コーティネート
!			ン保守		ティング	
ユーザの	<ul><li>ユーザの驚き</li></ul>	・ユーザの協力	・ユーザの失意	・ユーザ部門と情	・ユーザ主導	・18部門とユー
意識・参	・DP主導	・DP主導	・DP主導	報システム(IS)	・草の根運動	ザ部門の共同開
加				部門の共同開発		発
主な影響	・一般事務職層	・一般事務職用	・監督者暦	・上級管理者層	・オフィス・ワー	・全階層
		・監督者層	- 管理者層	- 専門破層	カー全体	
主な技術	・汎用コンピュー	・高級言語	・データベース/	・第四世代言語	・マイクロコンビ	・データ辞書/ゔ
	9	・ディジタル・デ	データ・コミュ	・リレーショナル	ュータ	ータ登録簿シス
	・機械語・アッセ	ータ伝送	ニケーション	・データベース	• 簡易言語(表計	テム
	ンプラ語		・光ファイバ	・プロトタイピン	算)	・エキスパート・
			磁気ディスク	グ技法	·LAN	システム
					·MML	・ISDN(ディジカ
	1					ル統合網)

(資料) 島田連巳、海老澤栄一編「戦略的情報システムー構築と展開」(1989)日科技連



図表6-1-3 企業情報システム概念の時代的変遷(組織階層)

<資料> 島田達巳,海老澤栄一編「戦略的情報システム-構築と展開」(1989)日科技連

## 1.1.4 統合生産システム(CIM)

CIMはもともとアメリカで提唱された生産手法の概念であるが、わが国でも工場単位の自動化や事務所のOA化が一段落した現在、さらに企業競争力を高めるためにコンピュータや通信手段を駆使して生産効率を大幅に向上させるCIMは、重要な経営戦略システムに位置づけられている。ここでは、統合生産システム(CIM)を事例に製造現場と事務管理を統合する情報化について概観する。

#### (1) 統合生産システムの概念

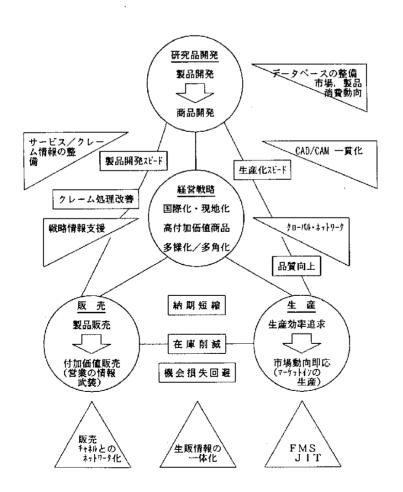
製造業を取り巻く環境の変化に柔軟に対応し顧客の満足度を高めるためのサービスの向上を図って行く必要がある。そのためCIMのねらいは、以下の点にあることを理解させる。

- ロエンジニアリング/品質の向上
- ○コスト削減(直接費、間接費)
- ○リードタイム短縮(製品開発、受注から出荷までの生産期間)
- ○生産のフレキシビリティの向上(多品種、需要変動対応)

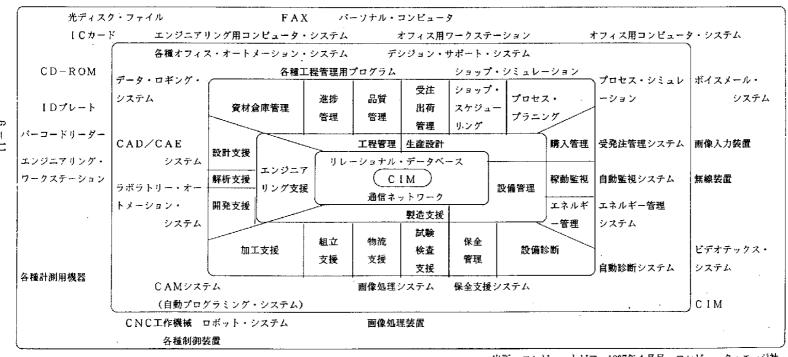
これらを実現するためのCIMの概念は、製造における販売、開発・設計、 生産・物流の大きな3つの生産活動における情報の流れをコンピュータと通 信ネットワークにより統合化した柔軟かつダイナミックに対応する生産シス テムとみなすことができる。

CIMの具体的な構築に際しては、それぞれの企業がおかれている環境に応じた統合化アプローチが必要である。1つの見方として設計・品種と製造形態の2つの面からみると、設計・品種では、個別性のある製品は、個別受注生産形態をとり、汎用性のある製品は、見込み生産形態となる。製造形態面からみると、川上製品では、素材部品生産があり、川下型であれば最終製造製品となる。

図表6-1-4 CIMのコンセプト(出典:別冊Computopia)



### 図6-1-5 СІМにおける機能ソフトウェア・ハードウェア



出所:コンピュートピア、1987年4月号、コンピュータ・エージ社

## (2) 発展経緯

- CIMに至る発展段階には、次の4段階があることを説明する。
  - 第1段階 自動機器の導入による単体機械の自動化
  - 第2段階…複数機械の接続による一工程の自動化
  - 第3段階…各工程を接続することによる製造現場全体の自動化
  - 第4段階…製造現場と事務管理システムの統合

現在、とくに先端的に進めている一部の企業を除いて、第1段階の単体の 自動化から第2段階の複数機械の接続による1工程の自動化へと進んでき ている過渡期に当る。

切削工程を例にとると、昭和50年代に入ってNC工作機械は、加工方法を数値入力するだけで自動化に加工を行うため、これにより大幅な省力化が可能であった。55年以降NC(数値制御; Numnical Cotrol)工作機械に工具の自動交換機能をもたせたMC(マシニング・センタ; Machining Center)が急速に普及してきた。MCは、60年の生産額の工作機械全体の25%を占め、一番大きくなっている。

そして現在では、FMS(フレキシブル・マニュファクチャリング・システム: Flexible Manufactuning System)と呼ばれ、数台のNC工作機械をロボット、自動搬送車などの搬送装置により接続し、ひとつの工程を完全に自動化するシステムである。

FMSは50年代後半に登場し、現在普及段階に入っている。

# (3) 技術的動向

今後の技術動向について理解させる。

今後、CIMが第3段階、第4段階へと発展してゆくためには、FA(Factory Automation)分野の標準通信プロトコルであるMAPの標準化が大きな足がかりである。第3段階において、工場内の各工程間を有機的に接続続するには、工場内LANといった大規模な情報ネットワーク綱が必要となってくる。各機器が異なった通信プロトコルを持ったままでLANを構築することは設備コスト面から、きわめて高価なものとなるため、MAPによるプロトコルの標準化は、第3段階に進むためにも必要な条件となる。

さらに、MAPは第4段階の事務・管理システムとの統合を最終目標に動き出しており、MAPとTOP(Techinical and Ohical Protocol)の接続によるCIMの実現となる。TOPとはボーイングが提唱している製造に関する技術・事務部門の標準プロトコルである。たとえば、生産管理システム、市場調査システム、CAD/CAD(コンピュータ支援による設計・生産)等設計システムなどをひとつのネットワークに乗せようとするもので、MAPが工場内LAN構築をめざすものであるのに対し、TOPは事務・管理用LANの構築をめざしている。MAPとTOPは容易に接続できるため、現在、MAPとTOPを用いたCIMの研究が進められている。

MAPとTOPといったプロトコルの標準化の問題だけでなく、さらにロボット、CAD/CAM等の機器の性能向上、生産管理等のシステム内容の充実、そしてCIM全体を効率的に動かすトータル・システムの開発等の問題を解決しなければならない。

第4段階の事務・管理システムとの有機的な結合の手段としては「情報ネットワーク」が考えられ、また統括的に情報を管理する手段として「リレーショナル・データベース」の利用が注目を浴びるようになってきている。

#### 1.1.5 CIMの事例研究課題

CIM化の動きは、電子・電機・機械・自動車・化学・食品・コンピュータなど全業種に広がり、大多数の企業がCIMを企業の国際競争力強化、生き残りに欠かせない技術と位置付けている。また一方では、システム開発要員の不足、社員の意識改革、巨額の投資といった問題に直面している。

しかし、CIMの形態については、各社の状況、背景が異なる以上、画一的なCIMはあり得ず、各社に効果的な独自のスタイルのCIMを構築している。自動車、機械産業では、CIM化が進んでいるが業種によりCIM化の速度はかなり違う。ただ、その具体的な進め方、構築方法は業種や個別企業によりまちまちだが、ねらいはかなり共通している。

CIMの基本的な狙いは、企業全体の合理化による企業の活性化、競争力の根本的な強化であり、上位の戦略情報システムの枠組みの中でCIMをとらえている。

- CIM構築の背景と進め方には3つのパターンに分けられる。
- ① 過去の技術の積み上げ長期の情報システム化、CAD化、FA化の積み上げの中で、システム・インテグレーションを達成し、今日のCIMを達成した例である。
- ② 新工場の建設から全社に拡大したパターン 新工場の建設が新しい生産ラインの整備とともに、FA化やプロセス・ オートメーション化に加えて、営業販売、生産計画管理、研究開発や設計 のシステム化を、新工場単位に統合化し、その後、全体に拡大したケース である。
- ③ 単一の工場または単一の工事パターン 単一の工場、単一の生産ライン、あるいは建設工事、プラント工事など での単一の工事を対象に営業、設計、生産などについてCIM化を達成し たケースである。

これらのことを考慮に入れて各業種CIM事例について調査をして、グループによる検討とレビューを行う。

# 1.2 社会・行政・地域における情報化(480分)

社会・行政(社会システムと行政)レベルの情報化は、90年代の情報化を左右するといっても過言ではない。というのは情報化社会において社会システムと行政の情報化とは、工業社会において上下水道システムや鉄道・道路などからなる輸送システム、電力システムなどが社会的インフラストラクチャであるように、社会基盤そのものの整備を意味するからである。

地域社会は90年代の情報化の担い手の1つとしての比重を増し、2000年には 主役としての役割を担うまでになろう。そして、これまで拡大の一途であった 地域と巨大都市の情報化格差の流れを止め、逆に格差を縮小する動きが顕著に なってくる。

1.2.1 社会・行政における情報化の流れ

社会・行政における情報化の潮流と変化について解説する。

60年代半ばに始まった銀行オンライン・システム(厳密には民間によるものであるが)や列車の予約システム(「みどりの窓口」)は、既にわれわれ

の日常生活に深く根をおろしている。その他80年代に導入され始めた住民票の管理システム、一部の公共料金(例えば、水道料金)の管理システム、さらに80年代後半に開始された学術情報システムなども実用化段階に入っている。また現在、総務庁が中心となって推進しているシステム、各省庁が独自に行おうとしている行政情報のデータベース化とその提供システム、特に中央官庁が持っている統計情報のオンライン情報検索システムなどのように、90年代になって実現されようとしているものも多い。しかし、全体としてみると、決して進んでいるとはいえない。

その1つの理由は、極めて公共的な性質を持つ社会基盤の整備は市場原理だけでは達成できないからである。このことは民営化以前の鉄道や電話のシステム等の例を見ても明白である。しかし、85年の電々公社の民営化と電気通信分野の規制緩和のように、徐々にではあるが市場原理が働く余地が広がりつつある。今後、行政レベルはもちろんのこと、社会システムの情報化(具体的には、医療情報システムや公共的な情報一般に関する検索システム等)を推進していくためには、政府の明確な方針とそれを実施していくため

等)を推進していくためには、政府の明確な方針とそれを実施していくため の施策が必要不可欠である。そして社会システムおよび行政レベルの情報化 には、省庁間の縦割り行政よりもユーザの利便性が優先的に配慮されるべき である。

### 1.2.2 地域における情報化の流れ

90年代は地方の時代といわれる情報化の潮流を解説する。

これまで拡大一途であった地域と巨大都市の情報化の流れを止め、逆に格差を縮小する動きが顕著になってくる。特に地域社会の情報化を刺激するのは通信料金の低減と遠近格差の縮小である。さらにパソコン通信やVANによる多彩なサービス・メニューが提供されることによって、これまでの注文システムや流通組織とは違った個人と個人の情報の流通が促進され、地方であれ都市であれ居住する場所に関係のない情報交換が可能になる。

80年代の地域情報化の担い手は地方自治体であった。政府各省庁の地域情報化振興政策をバックに、地域の情報が進められてきた。各地にコンピュータを核にした情報センターが建設され、ビデオテックスによる各種案内も始まり、CATV放送局などの施設、機械も続々と設置されてきた。90年代の

初頭も引き続き、こうした政府施策の助けを借りて、自治体とこれを軸にして地元の民間の力を集めた第三セクタが地域情報化の大きな推進力となるのは間違いない。

しかし、これまでの地域の情報化はハードウェアに重点を置く施策によっていた。設備指向であったが、今後はこうした設備をどう使うかのソフトウェア充実の時期となる。

また産業界のネットワークづくりも地域の情報化促進を前進させる起動力となろう。例えば情報ネットワークで「武装」されたチェーン・ストアである。現在、コンビニエンス・ストアで電力料金、ガス料金などの公共料金の支払いが可能となり、航空券や鉄道の切符も発券できるようにサービスを拡充しつつある。

同様に全国的な情報ネットワークをもつガソリン・スタンド、郵便局なども、こうした地域の情報拠点として全国に地域の情報を発信する窓口の役割を果たす。また予備校が衛星を利用して各地で一斉に同じ内容の授業を行う衛星授業を競って開始したのをはじめ、家電メーカ、乗用車、中古車販売などが衛星を使って商品情報などの映像を全国一律に伝送するなどの新サービスも始まっている。こうしたニューメディアの活用によって、全国の情報の同時性がミニコミレベルまで実現され、メディアによる地域間の情報化格差は是正に向うであろう。

# 1.2.3 EDI

ネットワークを利用した企業間取引に必要となるEDIとその動向について 説明する。

# (1) EDIの概要

従来、資材などの購入にあたっては、電話や業者への注文書渡しの方法で発注するのが当たり前であったが、今や社内の端末から通信回線に介して、業者側の端末に発注データを送り、注文するということが現実に行われてきている。このような商取引の形態を大きく変えていくものがEDIである。EDIは、異なる企業間で、商取引のためのデータを通信回線を介して、コンピュータ(端末を含む)間で交換すること。その際、当事者間で必要となる各種の取決めが、可能な限り広く合意された標準的な規約

であると定義される。

## (2) EDI発展の背景

EDIの考え方は、アメリカで1970年代の初めに提案され、実用システムが本格的に稼動を始めたのは1980年代になってからである。わが国では1982年頃から流通業界等で受発注の効率化を目標にEDIの実用化が始まっている。

## (3) EDIのメリット

1980年代に入り、先進的な情報・通信・ユーザは、多様化・複雑化する 企業環境の変化の中で、情報化、ネットワーク化を企業戦略としてとらえ、 これをいかに高度化していくかに力を注いできた成果の1つがEDIであ る。EDIは企業間の商取引をコンピュータ・ネットワークを駆使し、迅 速に、正確に、効率よく行うための最良の手段であるから、取引の範囲が 業界横断的に広がれば広がるほど各企業はさまざまなメリットを得ること ができる。

図6-1-6 標準ヒラネス・フロトコル(EDI用の標準)等の現状

	製造業の 資材発注等	流通業の 受発注等	アメリカ, ヨーロッパ	ECE/1SO (国際標準)
	(業界標準)	(業界標準)	(国内標準)	(EIWW-)
第4レベル				
取引基本規約	検討中	未検討	不明	未 検 討
(大枠の	108 41 T	不 快 削	(個別取決め -	A 17 10
取決め)				
第3レベル	例)	例)	といわれている。)	
業務運用規約	EIAJ	ベンサムの	.50 /	未 検 討
(システム運	運用ルール等	運用ルール等		A 18 61
用規約)	連州ルール寺	連用ルール寺		
第2レベル	例)	JAN コード, 日本チェーン	(米国)	• EDIFACT
情報表現規約	EIAJ標準フ	ロストナースフォースフォースフォースフォース アース アース アース アース アース アース アース アース アース ア	ANSI X 12 (欧州) EDIFACT または TDI	(=180 9735)
(コード/ フ	DIAMS 辛 ノ			• TDED
ォーマット)	1. マルレギ			(=180 7372)
第1レベル	例)		不定	
情報伝達規約	全銀手順の	J 手 順	(例: X, 400(0S1),	O S I
(通信プロト	一部借用等	全銀 手順	X. 25. SNA 等)	
コル)	(081)		4,	

注) (OSI) は、将来標準として用いる。

### (4) EDIに必要な取決め

EDIは、従来の書類ベースによる取引とは異なり、人間の介在しない 電子的な取引形態であるため、これの実施にあたっては、あらかじめ当事 者間で各種の広く合意された標準的な取決めをしておく必要がある。(取 決めの4つのレベル)

- ① お互いに使用する通信回線の種別や伝送制御手順などの通信プロトコル
- ② メッセージのフォーマットやデータ・コードに関する取決め
- ③ システム運用に関する取決め
- ④ 双方の契約に関する基本的な取決め

## (5) EDI導入の現状

流通業を中心に企業グループごと、業界ごとにEDIの導入が進み、最近は業界間のEDIも導入され、また製造業でもEDIの導入が進みつつある。さらに国際EDIの導入も始まっており、各企業ともここ当分の間は、EDIの導入とその積極的活用が最大の課題になると思われる。

# (6) 流通業におけるEDI

流通業は、ここ数年間EOSの構築などを通じてEDI導入のけん引車の役割を果たし、金融業を除けば最も導入の進んだ業界である。この業界では、大企業だけでなく中小企業にもEDIの導入が進んでおり、業務的にも受発注処理中心から関連処理を含めた総合的なEDIへの転換期に入っており、例えば、決済機能を付加したEDIや共同配送等の物流と連動したEDIの構築が始まっている。この業界の中ではEDIの導入が遅れた百貨店でも、POSの導入とともに受発注のEDIの導入が検討されており、今やEDIは必需品になっている。

# (7) EDIの円滑な推進に向けて

EDIを導入するためには、ある程度の社内システムが必要となりそれ 以上にEDIマインドともいうべき関係者の理解が必要となる。

- OED I に対応した社内システム
- ロインターオペラビリティ(相互運用性)と相互メリットEDIはインターオペラビリティを確保しつつ、相互にメリットが生

まれるように関係者の合意と協調によって進めることが大切。

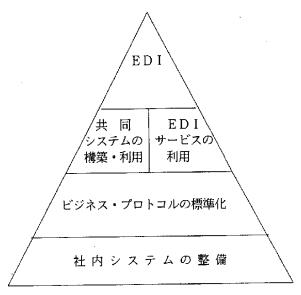
### ○標準化と共同化

ビジネス・プロトコルを中心とする標準化と業界共同システムの構築。

### ○EDIサービスの利用

VAN事業者によるEDIサービスの利用は大きなメリットがあり、 今後利用が拡大。

図表6-1-7 EDI導入の検討要件



### 1.2.4 EDIの事例研究(1)

業界ごとにEDIの導入が進んでいるがその実情について各グループ毎に 調査、検討し、意見交換をする。

## 1.3 個人・家庭における情報化(120分)

## 1.3.1 個人・家庭における情報化の流れ

90年代における個人・家庭における情報化の潮流・変化について解説する。 90年代には、情報化やエレクトロニクス化が生産の場としての産業にとど まらず、コスト採算という基準が不明確な消費の世界にも、価値観の変化、 ライフ・スタイルの変化を伴いながら、文化として浸透していくことになる う。

そのポイントとして情報利用のパーソナル化である。これは情報機器や情報サービスが情報を取り扱う主体として最小単位である個人にまでに普及していくということである。パソコン、ワープロは個人がより買いやすい価格となり、個人の利用に応えるような機能を備え、1人1台の状態に近づくと予想される。この結果、従来印刷物や放送といったメディアに頼っていた情報伝達手段にエレクトロニクスが加わり、個人や家庭にとってもコストと効用を勘案して幅広いオプションの中から情報媒体(印刷物、CD等)を選択できるようになった。

このことは次の3つの重要な変化を伴う。

その1はオンライン・データベース、パソコン通信、ビデオテックスなどに代表されるエレクトロニクスを利用した情報・通信サービスと印刷メディアとの境界が不明確になり、コストによってはエレクトロニクスを利用するようになるということである。情報利用のパーソナル化はパーソナルな情報利用を可能にするような情報提供側のサービスの変質を伴うことになる。

その2はエレクトロニクス媒体によって入手した情報の加工、再利用を可能にするパソコン、ワープロといった手段を個人が家庭に持つようになることである。90年代に完全な在宅勤務が一般化するとは考えにくいが、家庭とオフィスの境界線が不明確になり、仕事に関する情報を収集したり、収集した情報を分析、加工したりする人が増えてこよう。このような基礎の上に在宅勤務の可能性が醸成されてくると予想される。

その3は情報のパーソナル化が進み、個人が情報を電子的に入手し、知的 生産に使うようになると、情報やそのための手段を使いこなせる人と使いこ なせない人との間で情報化格差が潜在的に拡大することである。しかし、個 人の情報処理能力の差によって、情報や情報化の恩恵の偏在を招くことは望 ましくない。このような個人間の情報化格差の拡大の可能性に対しては、早 期に対策を講じて、コンピュータ・リテラシーを社会全体として高めていく 努力が大切になる。

第2のポイントは上記のような個人・家庭の分野における経済・生産活動 に関連した情報化のほかに、生活そのものに直結したホーム・エレクトロニ クスの分野での情報化が加速されることである。80年代に既にマイコンを内蔵した家電製品が大量に普及したが、90年代にそれらはさらに使いやすく、統合化の動きも出てこよう。冷暖房、防犯防災システム、給湯システムなどは各家庭においてもコンピュータによって統一的に制御され、電話回線やCATVの回線を使って外部のシステムとのインタフェースをもつようになる。一部では、機器自体が家庭内で制御されながら動き回り一定の役目を果たすロボットともいえるものが、実用化されるかもしれない。

# 育成方法および指導上の留意点

- (1) コンピュータが多用される情報化社会の現状と動向、および情報化の課題 について理解させる。90年代の潮流を個々の視座で把え、それが各自の Job とのかかわりを多少とも把握することが仕事をする上で大変重要であること を、章の始めに充分動機付けしておくことが必要である。
- (2) 技術的な解説でなく、問題解決の本質を浮き彫りにすることを目的として 説明することが受講生の情報化の理解を深め自分で考えて自分なりのスタン スをもつことに役立つと思われる。一方的な解説でなく、受講生と話し合い ながら身近な事例を参考に講義を進めるのが望ましい。
- (3) 情報化とコンピュータの章は、どちらかというと講義中心になるのは科目 の性質上やむを得ないが、教育期間を通じて事例研究やテーマ研究等の課題 を与え、グループ毎に調査をして、発表やレポートを提出させて自分達でその実状を把握させることが望ましい。

# 用 語

情報化、高齢化、国際化、データベース、ネットワーク、EOS、CIM、SIS、EDI、オフィス生産性、オフィス・アメニティ、情報化格差、戦略的情報システム、チャールズ・ワイズマン、統合生産システム、OSI、FMS、MAP、TOP、インフラストラクチャ、パソコン、VAN、インターオペラビリティー、プロトコル、ワープロ、セキュリティ

# 第2章 情報化社会の課題

## 学習目標

当章を受講された方々が、この章を修了した時点で、次の事柄ができること を目標とする。

- 1. 標準化は90年代の健全なコンピュータ・ネットワーク社会を推進していく際のインフラストラクチャであり、標準化の現状と課題について具体的に説明できる。
- 2. セキュリティについては、コンピュータ・ウイルスの出現と自然災害の 影響についての対策についての現状とパソコンの性能アップ、ネットワー クの拡大の中でセキュリティ対策をどう見直していくか説明できる。
- 3. ダイレクト・マーケティングの増加、消費者ローンやクレジットの増加 にともない、個人情報の集積拡大が進み、個人が思いがけない不利益を被 る危険が従来以上に増大している現状を説明できる。
- 4. 情報化社会における関係法令、基準などについて説明できる。
- 5.90年代の日本社会が、情報化の一層の進展によってより豊かで幸福な社会になるためには、情報化教育と情報化を推進するための人材の育成が必須の条件である。ここでは人材育成の現状と21世紀への人材育成、コンピュータ・リテラシーの重要性と意義を説明する。

### 2.1 標準化とその動向(480分)

2.1.1 標準化とその動向

ここでは、標準化についての流れについて解説する。

情報技術の進歩の速さ、情報関連産業の規模の拡大、さらにはソフトウェア資産の累積などの理由から、技術が成熟してから標準化を図るのでは、標準の意義は薄れ、また標準規格を採用するために既有のハードウェア・ソフトウェアを廃棄・変更することの損失が多大になる。こうしたことから、研究開発、技術開発の段階における先行的な標準化の必要性がより強く認識されるようになった。また、国際貿易の活発化、金融、運輸、サービスの国際

化などに伴う国際標準の重要性が再認識され、情報関連製品に対する互換性、相互接続等の要求もますます高まってきた。標準化は健全なコンピュータ・ネットワーク社会を推進していく上でのインフラストラクチャであり、特に力の弱い中小企業や消費者を護る立場に意が注がれなければならない。

## (1) OSIとビジネス・プロトコル(EDI)の標準化

現在、OSI (開放型システム間接続; Open Siytems Interconnection) の標準化は基本規格、実務規約開発、規格適合性試験、オブジェクト識別子 (コード) の登録制度、政府調達等の各レベルにおいて着々と整備されつつあり、本格的普及期へと向い始めた。

さらに、企業間等で相互接続システムを実際に構築するに当たっては、 ビジネス・プロトコル(EDI)(発注書、送り状等の取引メッセージの 書式、構文、コード等)の標準化が不可欠である。現在、各業界内および 業界間での事務処理を効率よく進めるために、ビジネス・プロトコルの整 備・検討が行われているが、90年代には、業種・業界横断的に標準化され たビジネス・プロトコルによるデータ交換が国際・国内ネットワークの発 展とともに広く進展しよう。

# (2) ソフトウェア関連の標準化

直面するソフトウェア危機を生産性の向上によっていかに克服していくかは積年の課題である。その意味でシグマ計画が推進されてきた意義は大きい。さらにソフトウェア技術の標準化により、インタフェースが統一されてソフトウェアの異機種への移植が容易になったり、部品化や再利用により生産性や品質が向上する等の効果があがり、ユーザにとってより使いやすくなることが期待される。

当面の課題としては、①プログラム言語、②データベース関連、③マンマシン・インタフェース、④OS/アプリケーション・インタフェース、⑤ハードウェアの互換性、⑥ソフトウェア評価の6分野の標準化への取り、組みが重点となろう。

#### (3) 新分野の標準化

産業の情報化は、従来実現が困難とされた分野にも浸透しつつある。高速の超LSIなどの半導体技術、光ディスクなどの大容量記憶媒体の出現、

ISDNの具体化などの要素技術の革新が統合されて大きな流れとなり、 これをユーザ・ニーズが支えて、新たなビジネス・チャンスが広がってい く。

これらを背景として①画像処理、②電子文書処理、③ CAD/CAM、④カード等の分野では急速に市場が拡大している。こうした新しい技術分野においては、わが国が技術的に高いポテンシャルを有する分野が数多くあり、先行的な研究開発/規格開発の知見に基づいて、その成果を国際標準化のステージへ積極的に提案し、国際的な役割を担うことが90年代の大きな課題となろう。

### 2.1.2 OSIの標準化

ここでは、OSIの標準化について具体的に解説する。

### ① 基本規格と実装規約

OSIに関する規格制定活動が開始されたのは、1977年3月TC97(電子計算機および情報処理)のSC16でOSIに関する標準化活動がスタートし、以後精力的な規格制定作業が続けられ、現在では40件の基本規格が制定されている。

また、わが国の実装規約は工業技術院の大型プロジェクトの一環として (財)情報処理相互運用技術協会(INTAP)を中心に開発が行なわれ ている。OSIの本格的普及期へと向うには今後 100項目に及ぶ基本規格 の制定が必要とされている。

# ② 規格適合性試験・認証制度

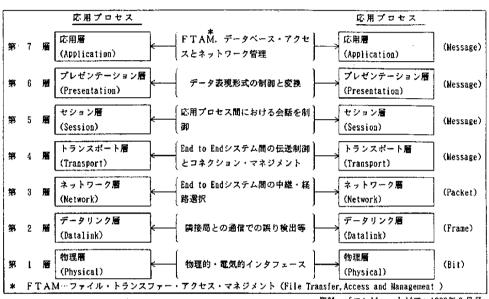
実装規約に基づくOSI製品が市場登場してくると、実際にユーザがOSIを導入する時に次のような疑問点が考えられる。

- ○実装規約に準拠して作られた製品かどうか
- ○実装規約で決められたどのような機能パラメータを選択しているのか
- ○実際に相互接続できるのかというインターオペラビリティ性

そこで、製品の検証のためには高度な技術能力をもった公正中立な第三 者機関がテスト・ツールの開発や規格適合試験を実施し、規格適合性を認 証する制度が求められる。

#### ③ 登録体制の整備

ISOでは、データ・コード等の国際規格を制定した場合、そのコードを国際的に一意に付与するために国際登録機関を指定する必要性を考慮し、1984年ISO理事会においてその旨議決し規則を定めた。さらに、必要性に応じ、順次個々のISO規格のための登録制度を整備し、国際登録機関を設立し国際標準化事業の円滑化を図っている。



図表6-2-1 ISOのOSIモデル

資料: 「コンピュートピア」1989年8月号

### 2.1.3 ビジネス・プロトコル (EDI) の標準化

EDIを実施するためには4種類の取り決めが必要でありその概要について説明する。

データのビット列のもつ意味を共通化するための規約は、今日、ビジネス・プロトコルとしてよく知られるようになり、またその重要性に対する認識も高くなっている。その標準化は、かつて純業務処理上の問題として関係業界で検討されてきたが、最近では、さらに広域の課題として注目され、ISOやCCITT(国際電信電話諮問委員会)などの国際的機関でも取り上げられ、一部の規格については、国際規格として既にISOで承認されている。このビジネス・プロトコルの標準化の具体的内容としては、現在では一般的に①シンタック・スルール、②標準メッセージ、③データ・エレメント・

ディレクトリ、④標準コードの4つの要素が議論されている。この他にも重要な要素のあることが既に知られているが優先的に標準化すべき要素としては、海外でもこの4つが取り上げられて、検討されている。

これらの要素の標準化は、業界レベルでは、主として業界団体(内部に設置された委員会)で検討されているが、実施のための専門機関を設置した業界もある(例:(社)日本電子機械工業会EDI推進センター)。また公益団体(例:(財)日本情報処理開発協会産業情報化推進センター、(財)流通システム開発センター)などでは業界横断的な検討が行われている。

### (1) 4種類の取決め規約

第1、2レベルについては標準化が進められており、第3、4レベルに ついては何らかのルールの確立が求められている。

### 情報伝達規約(第1レベル)

通信プロトコルとも呼ばれるこの規約は、EDIメッセージを通信回線で送るために必要な規約でEDIのベースということができる。

### 情報表現規約(第2レベル)

この規約は、通信ネットワークを郵便に例えた場合、封書の中の本文 の記述方法を定めたもので、発注書などの帳票の記述方法のようなも のでメッセージ・フォーマットやデータ・コードに関する取決である。 業務運用規約(第3レベル)

システム運用規約ともいわれるこの規約は、EDIシステムを円滑に 運用するために必要な取引で、どの業務処理(例、発注、納品…)に EDIを用いるかという調整から、運用時間の調整まで、取決めるべ きことは多く、さらに作業ミス(オペレーション・エラー)や自然発 生的エラーが起きたときの、後始末の方法まで取決める必要がある。

# 取引基本規約(第4レベル)

EDIを行う際に、オンライン取引契約書を締結する例が多数ある。 第1レベルから第3レベルまでの取決めが適用される範囲や有効期間、 変更する場合の手続きなどを具体的に取決めるものである。この契約 書によってEDIによる取決め法的有効性も確認することになり、E DIによる取引の有効性についての法制度上の一般則が確立されてい ない現段階では重要な取決めといえる。

### 2.1.4 ソフトウェア関連の標準化

ソフトウェア関連の標準化についての現状と考え方と課題について説明する。

ソフトウェア関連の標準化は、プログラム言語、グラフィックス、ファイル仕様、コード体系、文書管理、ユーザインタフェース、データベース等広範な部門にわたっており、ISO/IECJTCL においても活発な活動が行われている。

ソフトウェア標準化の必要性は、インタフェースの統一によるソフトウェアの異機種への移植、部品化による再利用等による生産性や品質の向上を通じて、ソフトウェア産業が抱える問題点を解決するばかりでなく、ユーザにとって大きなメリットをもたらすことが期待され、ますます重要なテーマとしてクローズアップされてきている。

しかしながらソフトウェア関連の標準化は過度な画一化は進歩を阻害する 可能性の大きい分野であるため、基本方針として次の考え方を示している。

- ① 未来像を重視した標準化
- ② 人間の多様性を尊重した標準化
- ③ ユーザの立場に立った標準化
- ④ 標準化領域と非標準化領域の見極めと技術の選択を可能にする標準化 (技術発展の障害とならず、その後の技術進歩を可能とするような標準 化領域の見極め)等

今後の技術の動向をにらみつつ、ユーザに対して何が標準化により寄与できる部分かを十分見極めて対応することが必要であろう。

ソフトウェア分野の中長期・横断的な標準化課題については、次の3点が 指摘されている。

- ① ソフトウェアの移植性、再利用性の向上 ソフトウェア技術の標準化の最大のメリットは、蓄積されたソフトウェア資産の流通性を高め、異機種への移植、部品としての再利用等をもたらすことである。
- ② 国際規格へのマルチ・バイト機能(日本語機能)の導入

コード系、データ・タイプ、グラフィックス、処理機能等の統一的な 枠組みについて、中国、韓国をはじめとするアジア諸国との調整を図り、 率先してこれの国際規格への導入に向け取り組む必要がある。

③ ソフトウェア開発・保守のトータルな生産性の向上支援 ソフトウェアの最大の課題は、直面するソフトウェア危機を生産性の 向上によっていかに克服していくかにある。このためには、ソフトウェ アの開発・保守に係る標準化が不可欠であり、また、その実効性を高め るためには、総合的にその標準化を進める必要がある。

## 2.1.5 標準化の事例研究

ソフトウェア関連の標準化について各グループ毎に調査し、プログラム開発上、コンピュータ利用上、データ交換上、教育や情報交換上の利点から検討し、意見交換を行う。

### 2.2 セキュリティとプライバシー(120分)

### 2.2.1 セキュリティ対策

セキュリティ対策について現在の状況について説明する。

80年代の最終段階になってコンピュータ・セキュリティに関する2つの重大な出来事が発生した。1つはコンピュータ・ウイルスがわか国でも発見されたことであり、もう1つはサンフランシスコ湾岸地震の発生に伴い大都市圏のコンピュータ・システムのセキュリティ対策が問われたことである。

# (1) 新しいリスクの出現

コンピュータ・システムの利用を巡って、ウイルスという新しいリスク が出現している。現在、ウイルス対策の基準づくりやワクチン・プログラ ムの開発など、いろいろな対応策が講じられているが、ウイルスの実態が 必ずしも明らかにされていないところに問題がある。

今後、ウイルスのメカニズムが解明されていくとともに、その対策も実 効性の高いものになっていくと思われる。しかし問題はウイルスそのもの への対策も重要であるがそれ以外の情報化整備(プログラムの無断コピー の横行など)までも考慮しないと、本当の対策にならない。

また、ウイルスはパソコンの問題であまり悪意が感じられず、ゲーム的

な色彩が強い。しかしウイルスが悪意をもって仕掛けられたらどうするか ということと、汎用コンピュータに関して出現する可能性について対策を 考えなければならない。

### (2) 自然災害の影響

サンフランシスコ湾岸地震の発生は、大都市における自然災害の恐ろし さを見せつけ、コンピュータ・システムにも多くの影響を与えた。その中 で、バックアップ・システム・サービス会社によるシステムのバックアッ プが効率的であったと言われている。わが国ではシステムをバックアップ するサービスを行っている企業は一社にすぎない。今後の検討課題であろう。

### (3) セキュリティ対策の見直し

パソコンの性能アップ、ネットワークの拡大など、情報化環境の変化が 著しく、それにセキュリティ対策が追いつかないという状況もあることを 知っておかなければならない。どこに、どのようなリスクが潜在している かを把握するとともに、情報化環境の変化によって影響度が変わることも 知っておかなければならない。

### 2.2.2 個人情報保護

プライバシー保護の現状と問題点および課題について説明する。

コンピュータ・システムの進展は、従来手作業で行っていた個人情報の処理を効率化するとともに、個人情報の新たな集積と利用を急速に拡大した。公的部門では、国の社会保険や貯金、運転免許、税などが電算化され、地方公共団体では住民登録を中心とする個人別の行政情報システムが導入されている。民間部門では電力、ガス、電話などの顧客管理が電算化される一方、経済成長の減速化を背景として、これまでの不特定多数の顧客を対象とするマスセールに代わる個人指向のダイレクト・マーケッティングが増加し、また金融の面では消費者ローンやクレジットが増加したことに伴い、個人情報の集積が拡大している。

90年代において、以上の傾向に対して予想される変化は、情報処理の高度 化に伴って従来比較的事業所単位で集積されていた個人情報の連携および多 角的な利用や流通が各分野に拡がることである。そのような動きは、一面に おいて対個人サービスをより緻密、適正なものとし、情報収集に当って個人にメリット(煩わしさを軽減する等)を与える一方で、個人が思いがけない不利益を被る危険を従来以上に増大させる。

すなわち、①誤情や古い情報による影響は迅速、広範に及び、②個人情報を悪用される潜在的な危険は、個人情報の分析・加工を伴う利用の高度化により、さらに深まるからである。そのような悪用の危険は、個人情報の安易な外部提供やコピーの不正持ち出し、通信回線による不正アクセス等によって高まる。

個人情報の保護に関しては、既に70年代以降欧米諸国において立法措置や 民間の自主規制が行われており、1980年にはOECDのガイドラインが公表 されている。

日本でも88年に国の行政機関の保有する個人情報について法律が成立し、 地方公共団体の保有する個人情報についても条例の制定が進んでいる。民間 部門でも86年に信用情報に関して行政通達が出されたほか、民間部門におけ る個人情報保護を目的とする自主規制を促すため87、88年にFISCとJI PDECによるガイドラインがそれぞれ公表 された。さらに後者を受けて 89年に、民間の自主規制に関する通産省のガイドラインが公表され、企業や事 業者団体の対応状況に関する登録制度が実施されている。

以上の保護措置の内容は、公的部門と民間部門の相違や、背景の違いによって必ずしも一定ではないが、そこに流れる基調として、個人情報の正確性、最新性を保つために合理的安全対策を講ずべきことと並んで、情報の主体である個人が自己の情報をコントロールできるようにすべきであろう。そのためには個人情報を利用する企業やその団体による責任ある努力が今後必要となろう。

# 2.3 法制度と基準 (360分)

情報処理に関係する法令、基準などについて基礎的な理解を得る。

# 2.3.1 ソフトウェアの権利保護

ソフトウェアが工業製品として、あるいは商品として市場に流通するよう になってきている。このように、ソフトウェアも他の工業製品と同様、企業 活動としての製品開発とその市場での販売による適正な利潤獲得を正当化していくためにも、各種の法律の整備が緊急のテーマとなっている。

ソフトウェアの権利保護の問題は、今後ますます進むであろう世の中のソフト化にともなって、企業活動としてのソフトウェア開発の先行投資に対する適正な保護を、いかに実現するかの問題であると言える。

すでに、ソフトウェアの権利をどんな法律で保護するかについては、国際 的にも様々なやり方があり、かつその適用に当っても様々な問題を提起して おり、未だ十分な法体系が整備されているとは言えない。

しかしながら、情報化・ソフト化の先進国である米国が、すでに著作権での保護を実施しており、わが国においてもソフトウェアを著作権で保護するよう、昭和61年に著作権法を改正し施行している。しかし、法改正はあくまでも原理原則の提供であり、どこまで保護されるかについては、具体的な保争での判例の動向を注視していかなければならない。

このどこまで保護されるかは、非常に厄介な問題であり、素人ではなかなか判断のつきかねる問題である。また一方で、特に最近の日米貿易摩擦などの状況ともからんで、先端技術ならびにソフトウェアなどの知的所有権の問題は、きめ細かな神経をつかわなければならなくなっている。(海外輸出物件の対応、ソフトウェア購入と組込み再販、海外技術情報の入手、リバース・エンジニアリングなどなど)

#### (1) 知的所有権の保護形態

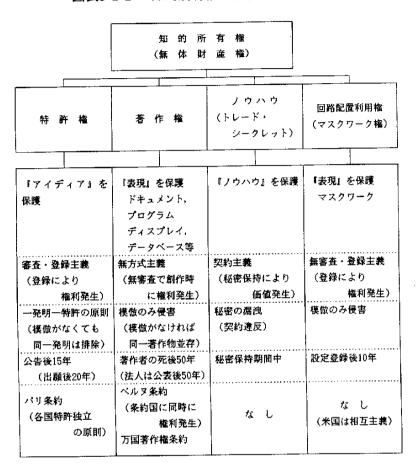
一口に、ソフトウェアの権利と言っても幅広いものがあり、無体財産を 保護対象とする権利である。知的所有権の保護形態によって保護される。

### 工業所有権

工業所有権とは、アイディア等を保護する独占排他権であり、設定登録により発生する特許権、実用新案権、意匠権及び商標権の4つの権利をいう。審査を経て権利となるものであるため、安定した権利である。

- ○特許権とは、技術思想の創作である発明を保護対象とする権利である。 特許権の存続期間は出願公告日から15年または出願日から20年のうち いずれか早く経過する日に満了する。
- ○実用新案権とは、物品に関する技術思想の創作である考案を保護対象

図表6-2-2 知的所有権の保護形態



とする権利である。「物品に関する」ものであるため、「ソフトウェア」は発明として把握すべきである。実用新案権の存続期間は、特許権のそれより短く、出願公告日から10年または出願日から15年のうちいずれか早く経過する日に満了する。

- ○意匠権とは、物品の外観である意匠(デザイン)を保護対象とする権利である。意匠権の存続期間は、設定登録日から15年である。
- 商標権とは、商品に付される商標を保護対象とする権利である。ソフトウェアの名称等は予め商標登録出願をしておき、計算機に取組んで 出荷するときには設定登録を受けられている状態となっていると積極

的に使用できる。商標権の存続期間は、設定登録日から10年である。

### ② 著作権

著作権とは、文化の発展を目的として、人間の精神の表白物を保護対象とする権利で、ソフトウェアの場合には、表現された図書類が保護対象となる。本権利は、図書類の作成の完了と同時に、無審査で発生する。著作権の存続期間は図書類の作成完了から死後50年であり、法人著作物の場合は、図書類の作成完了から公表後50年である。

### ③ ノウハウ (トレード・シークレット)

ノウハウとは、産業上利用することができる技術思想の創作又はこれを実施するのに必要な具体的な技術知識・資料・経験であって、これを創作・開発・作製または体得した者(その者から伝授を受けた者を含む)が現に秘密にしているものをいう。発明等として開示できなく、当社にとって重要であると考えられる場合には、ノウハウとしての保護を求める。ノウハウの保護期間は創作等の完成から秘密を保持しておく必要があるまでの期間であり、他社に当社のノウハウを開示する場合には、その旨の契約を締結しておく必要がある。

# ④ 回路配置利用権

回路配置利用権とは、半導体を設計製造する際の回路配置を保護対象 とする権利である。

# (2) ソフトウェア開発と保護手段(形態)

ソフトウェアの各開発工程でなされる創作に対して、工業所有権、著作権、ノウハウおよび回路配置利用権が適用され、各創作が保護される。

ここでは、ソフトウェアの開発工程での保護対象とその保護形態につい て説明する。

# ① システム設計(要求定義)段階

この段階では、客先仕様等に基づきシステムの大まかな概念が決定されるので、この段階で成立する基本的な発明は、基本特許となる。従って、発明を大局的に把え、早急に提案し、出願すべきである。また、システム要求書、システム外部仕様書、システム計画書等は、著作権の対象である著作物になる。(言語の著作物)

### ② 基本設計段階

この段階では、さらに具体的機能が定まっているので、基本発明を改良した改良発明を考え、基本発明の周辺を固める。プログラムの名称等は商標として保護を受けられないか検討する。保護を受けた方が良いと判断された場合には、商標登録出願をする。

著作権としては、方式的設計書、構造設計仕様書等が対象となる。 (言語および図形の著作物)

### ③ 詳細設計段階

この段階では、詳細機能が定まっているので、発明としては基本発明 を種々に応用した多数の応用発明となり、基本特許権の周辺を守る周辺 特許権となる。

著作権としては、プログラム(詳細)設計書、プログラム・テスト要 領書、データ定義仕様書(データベース)等が対象となる。(言語及び 図形の著作物)

## ④ コーディング段階

この段階では、詳細機能をコーディングする段階なので、発明として 成立するような技術思想として把えるよりは、具体的なオペレーティン グ・システム、アプリケーション・プログラム、あるいは別の観点とし て把ればソース・コード、オブジェクト・コード、マイクロ・コードの 創作としての意義が大きいので、特許権より著作権の対象として把える ことができる。(プログラムの著作物)

# ⑤ その他

複数のデータを収納したデータベース、CRT等のモニタに表示された画面自体等は最早特許権の対象たり得ず、著作権の対象そのものである。(図形及びプログラムの著作物)

また、工業所有権または著作権として公開して保護を受けるよりも、 会社としては秘密にしておいた方が利益があると考えられる場合には、 会社のノウハウとする。

その場合には、第三者に見られることがないよう必要な措置を講ずる。

図表6-2-3 ソフトウェア開発成果物と保護形態

保護手段 開発工程	工業所有権	著作 権	ノウハウ	回路配置利用権
要求定義	基本			
基本設計	許			ゲートアレイ
詳細設計	周	作りの多		
コーディング	辺特	作物	1	
試験	許	4物		
運用保守				

# 2.3.2 データに対する保護

データに対する保護の方法としては、著作物として保護する方法、財産として保護する方法が考えられるが、現行法では電気など法律で明示されているもののほかは有体物に限られるため、保護の対象には入らない。ただ多くの場合、媒体の財産権侵害を伴うので、媒体面から保護される。

# (1) プライバシー権の保護

従来はプライバシー権は、各個人が他から干渉されることなく「ひとりにしておかれる権利(right to let alone)」と消極的なものに受けとられてきたが、現在では「自己に関する情報の流れをコントロールする権利(individual right to control the circulation of information relating to himself)」と積極的に考えられるようになったことを説明する。そこから、自分に関する情報で他人が保管しているものがどうなっている

かを知る自己情報へのアクセス権が発生する。

### (2) データベースなどの保護

データベース、名簿などの内容については、現在のところ必ずしも保護 の根拠が明確ではないが、著作物として保護される方向にある。

### 2.3.3 電気通信事業者に関する法制

従来は電気通信政策で、基幹的な電気通信網の確保、重要通信の確保、通信の秘密の確保、安全性、信頼性の確保、合理的な料金の設定を主眼として、独占形態により日本電信電話公社、国際電信電話株式会社が担当してきたことを説明する。しかし、ニュー・メディア通信の実用化、利用者のニーズの高度化、多様化、電気通信事業の活性化、対外経済摩擦の解消など新しい要求が出てきたので、通信政策が変更され、

- ① 独占の排除と競争原理の導入
- ② 利用者サイドに立ったサービスの実現
- ③ 環境変化への柔軟な対応
- ④ 日本電信電話公社の民営化

などが考慮され、電気通信事業法が改正されて昭和60年4月に施行された。 その結果、第二電々と呼ばれる第一種業者や、第二種業者が多数事業を開始 した。

# (1) 第一種電気通信事業者

第一種電気通信事業者とは、電気通信回線を自ら設置してサービスを提供する業者であり、郵政大臣の認可を必要とする。利用者保護のため、事業開始後も技術・設備などの水準を常に一定レベル以上に維持する必要があるので、それを保証するため行政的な指導が行われる。

# (2) 第二種電気通信事業者

第二種電気通信事業者とは、第一種電気通信事業者から回線を借りてサービスを提供する事業者である。原則的には、一切の参入規制を行わず、最小限の項目を届出でればよいこととされているが、そのうちでつぎのようなサービスを提供する業者は、特別第2種として登録を必要とする。

- ① 誰れもが加入できる不特定多数向けのサービス
- ② 政令で定める一定規模以上の全国基幹サービス

### ③ 国際間の通信を取扱うサービス

これらのサービスが社会・経済的に重要であり、また国際通信の重要性が 高いため、高度の安全性、信頼性が要求されるので、郵政大臣に料金、サ ービス条件を届出て、技術基準に適合するよう諸設備を維持するため管理 運用規定を設け、電気通信主任者を選任してサービスに当らせることが要求されている。

サービス内容も規制の範囲内であれば他人使用、共同使用、相互接続が 許されることとなった。

### 2.3.4 情報処理技術者と労働者派遣法

### (1) 法律制定の背景

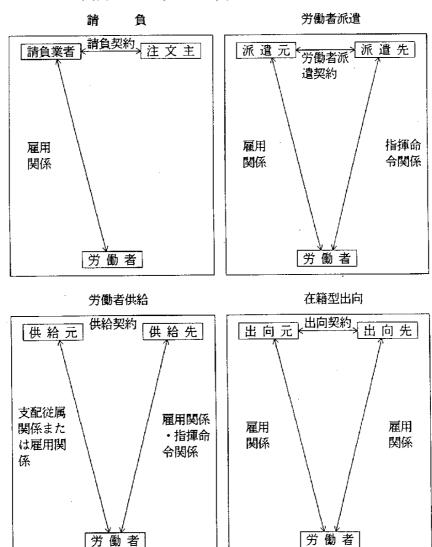
ある期間、労働者を他の企業等に派遣する事業は、これまでビル管理、 通訳などについて行われていたが、情報処理の分野でもプログラマ、シス テム・エンジニア、オペレータの他企業等への派遣という形でふえてきて いた。このような労働者の供給、派遣は強制労働や中間搾取の発生する余 地を残すため、職業安定法第44条で禁止されており、法律違反である。し かし、従来は黙認されてきたが二重派遣、条件の悪化なども目立ち始めた ため、規制する法律制定となり、昭和61年7月から施行された。法律の名 称は「労働者派遣事業の適正な運営の確保及び派遣労働者の就業条件の整 備等に関する法律」である。

### (2) 事業規制

法律では、派遣労働の対象業務について限定することとし、中央職業安 定審議会の意見を聴いて対象業務が決定されることになっている。その基 準としては、

- ① 迅速かつ的確な遂行のために専門的な知識、技術または経験を必要と する業務
- ② 就業形態等の特殊性により、特別の雇用管理を必要とする業務とされており、中央職業安定審議会労働者派遣事業等小委員会報告書によると、情報処理関係として、
  - (a) 情報処理システムの分析、設計およびこれに直接付帯する業務
  - (b) プログラムの設計、作成およびこれに直接付帯する事業
  - (c) コンピュータ・システムの操作、データの入力

図表6-2-4 派遣と請負、出向等の差異



が案として候補に挙っている。そのほか、関係がありそうなものとして、 展示会のデモンストレーションなど専門知識が必要な機械の性能、操作の 説明、紹介がある。

# (3) 労働者派遣事業の定義

法律は労働者派遣事業を「自己の雇用する労働者を当該雇用関係のもと

に、かつ他人の指揮命令を受けて当該他人のために労働に従事させる業」 と定義し、派遣元と労働者との間に雇用関係がない労働者供給事業との間 に区別を設けた。

### (4) 派遣労働者の保護

派遣元が労働者保護の観点から講ずべき措置として、

- ① 派遣先での就業条件の明示
- ② 教育訓練の実施
- ③ 責任者の選任
- ④ 管理台帳の備えつけ
- ⑤ 労働安全衛生の保護

などを義務づけている。労働基準法等労働者保護法規の適用について、使用者責任は原則として派遣元が負うものとし、労働時間、休憩、休日等の規定は派遣先を使用者とみなす特例を設けている。また派遣期間にも上限か設けられる見通しである。

2.3.5 VDT作業における労働衛生管理のあり方(労働省労働基準局長通 達)指針

### (1) 指針発行の経緯

コンピュータ、オフィス・オートメーションの普及によって、VDT (Visual or Video Display Terminal) が広く導入されてきたが、VDT 作業のもたらす健康への影響や予防に関する調査研究が各方面で進められてきた。

そのため、労働省では機器の改良や人間工学、労働生理学等の分野における知見に基づいて、VDT作業における労働衛生管理のあり方を見直し、本通達の発行となった。

# (2) 対 象

指針では、対象を事務所(事務所衛生基準規則第1条第1項に規定する 事務所)において行われるVDT作業に関する労働衛生管理を対象とする が、事務所以外の屋内作業場において行われるVDT作業およびVDT類 似作業についてもこれを参考にして労働衛生管理を行うことが望ましいと している。

### (3) 作業環境管理

作業環境管理として照明及び採光、グレア防止、騒音電波の防止、その 他を定めている。

- ① 照明及び採光
  - (a) 室内は明暗の対照がいちじるしくなく、まぶしさを生じさせないこと。
  - (b) ディスプレイ画面照度は、500 ルックス以下、書類およびキーボード面の照度は 300/1000ルックスとすること。なお、周辺の明るさも差を小さくすること。
  - (c) 直接太陽光が入射するなどの高い輝度の窓は、ブラインド、カーテン等を設けて輝度を低下させること。
- ② グレア防止
  - (a) CRTディスプレイ画面の前後の傾斜の調整を行うこと。
  - (b) 低輝度型照明器具を使用すること。
  - (c) CRTディスプレイ画面の前後の傾斜の調整を行うこと。
  - (d) CRTディスプレイにフードまたはフィルタを取り付けること、または反射防止型CRTディスプレイを用いること。
  - (e) その他グレア防止の措置を講じること。
- ③ 騒音電波の防止

プリンタ等から不快な騒音が発生する場合には、騒音電波の防止措置 を講ずること。

4) その他

換気、空気調和、静電気除去等について事務所衛生基準規則を定める 措置をはじめとする必要措置を講じること。

- (4) 作業管理
  - 作業時間等
    - (a) 1日の作業時間

CRTディスプレイ画面を注視する時間が短かくなるよう配慮し、 他の作業を組み込んだり、ローテーションを実施すること。

### (b) 連続時間及び作業休止時間

一連続作業時間が1時間を超えないようにし、次の連続作業までの間に10~15分の作業休止時間を設け、かつ一連続作業時間内において1~2回程度小休止を設ける。

## ② VDT機器等

- (a) フリッカー フリッカーは知覚されないものであること。
- (b) キーボード キーボードは位置が調整できるものが望ましく、ストロークおよび 押圧が適当で、キートップの文字は見易く、表面がつや消しになって いることが望ましい。
- (c) 椅 子 安定していて容易に移動できること。
- (d) 机または台 適当な広さがあり、脚まわり空間は窮屈でなく、調節ができない机 または台を使用する場合、65~75cmのものであること。
- (5) VDT機器等および作業環境の維持管理 日常の点検と調整、定期点検、清掃を行うこと。

### (6) 健康管理

- 健康診断
  - (a) 配置前健康診断

新たにVDT作業に従事する労働者の配置前の健康状態を把握し、 その後の健康管理を適正に進めるため、業務歴の調査、既往症および 自覚症状の有無の調査、眼科学的検査、筋骨格系に関する他覚的検査、 その他医師が必要と認める者について必要な検査を行う。

# (b) 定期健康診断

労働安全衛生規則第44条に定める定期健康診断を実施する際、併せて業務歴、既往症、自覚症状有無の調査、眼科学的検査、筋骨格系に関する他覚的検査を行う。

### (c) 健康診断結果に基づく事後措置

愁訴があるときは主因を明らかにし、視力矯正が不適切な者は適正 視力で作業が行えるよう保健指導する。また、VDT作業が適切でな いと判断される者には適切な措置を講じること。

# (d) 健康相談

VDT作業者が気軽に相談し、アドバイスが受けられるようにする。

### (e) 職場体操

就業の前後または就業中に体操を行わせることが望ましい。

### (7) 労働衛牛教育

- ① VDT作業従事者への教育 自主的に健康管理が行えるような項目について教育する。
- ② 管理監督者に対する教育 必要に応じて行う。
- ③ 新たにVDT労働につく労働者に対する教育習得及び習熟に必要な訓練を行う。

# 2.3.6 情報システムに対する監査基準

情報化の進展に伴い、情報システムが停止、悪用、有効に機能しなくなった場合の影響が深刻になってきている。情報システムの設計、開発、運用におけるこれらへの配慮を補完強化する意味で、独立したシステム監査人が情報システムを総合的に点検、評価し、関係者に助言、勧告を行う制度がシステム監査制度である。基準は一般基準、実施基準、報告基準の3分野から成立っている。一般基準は総論に当る部分で、監査の目的、対象、監査人などの監査基準を定めている。実施基準は中心的部分である。企画業務、開発業務、運用業務に分けてさまざまな監査基準を定めている。報告基準は監査を実施したあとの対策、措置を定めた部分である。

監査は、当事者以外の第三者が行うものであるから、情報処理の開発や運用の当事者に直接関係がないともいえるが、知っておくと自己チェック・ポイントとすることもでき、有益である。

### 2. 4 人 材 育 成(240分)

### 2.4.1 情報化社会の教育

情報化社会における人材育成とコンピュータ・リテラシーの問題点について説明する。

90年代の日本社会が豊かで活力に満ち、そのうえ若年者にとって希望があり中高齢者にとってはアメニティに富んだ社会であるためには、情報、通信技術の発展、なかでもユーザフレンドリな情報機器の開発と同時に、それを支える社会的な枠組みの構築と整備が必要である。そのための第一条件が、そのような情報機器の有効な活用と、学校教育・職業教育そして生涯教育ないしは生涯学習における情報化に関する教育(情報化教育)の積極的な取り組みである。つまり90年代の日本社会が、情報化の一層の進展によってより豊かな幸福な社会になるためには、情報化教育と情報化を推進するための人材の育成が必須の条件となる。

ここで情報化教育とは、ハードウェア・ソフトウェアの開発担当者のような情報・通信技術者の専門家を育てることだけではなくて、一般大衆の情報リテラシーないしはコンピュータ・リテラシーをいかにして高めるかということである。

この問題は90年代に入るとともにますます深刻化し、単に個人の問題だけでなくて、社会全体にとって大きな問題となるであろう。なぜならば、情報化が産業レベルすなわち生産現場やオフィスだけではなくて、社会システム全体から家庭の内にまで広く深く浸透すると、日常生活にまでコンピュータ・リテラシーが必要とされるようになるからである。すなわち、情報化社会においてコンピュータ・リテラシーは、「読み・書き・ソロバン」や電話・電気製品を使うための日常的な知識・自動車の運転技術などと同じように、日々の生活を営んでいくうえで欠かすことのできないものとなるからである。

このような90年代にふさわしいコンピュータ・リテラシーを人々が獲得するためには、学校教育(公的教育)の場においては、①初等・中等教育における「コンピュータを用いた教育」や「コンピュータの基礎教育」②高等教育における「コンピュータを利用した教育」と「コンピュータを有効に利用するための教育」を積極的に進める必要があろう。そのためには、教育現場

における設備装備率の向上および教育の増強・再教育が急務である。

さらに、情報化をより一層推進するための人材を育成するには、より実践的な訓練、コンピュータ要員、ソフトウェア要員を確保するための職業教育(特に専門学校、各種学校)を、これまで以上に充実していくことが必要不可欠である。

ただし、これら人材の育成は、コンピュータ・リテラシーの問題が解決されて、はじめて意味をもつ。したがって、学校教育を既に終了した人が、生涯学習の一環として、コンピュータ・リテラシーを新たに獲得しうる場ないしは機会の創設もなされるべきである。

# 2.4.2 人材育成の現状とコンピュータ・リテラシー

情報処理教育の目的と期待される人材およびコンピュータ・リテラシーに ついて説明する。

情報処理教育の目的を大別すると、情報処理技術者の養成確保とコンピュ ータ・リテラシーの涵養とがある。

# (1) 高度情報処理技術者の育成

2000年に向けての高度情報化社会を実現するための情報化投資やソフトウェア需要を充足し、情報産業がリーディング・インダストリーとして成長していくためには、情報処理技術者の育成確保が急務になっている。とりわけ提案型のシステム・エンジニア(SE)やシステム・コンサルティングできるSE、システム・インテグレーションに際してリーダとして活躍できるようなSE等、高度な情報処理技術者の育成が不可欠となっている。一方、2000年には現状のままでいくと97万人、育成の諸施策の成果をもってしても40万人のソフトウェア技術者の不足が予測されている。しかも40万人の中の31万人はSEの不足である。

このような不足を充足していくための方策の1つとして、地域格差を是正し、地域でのSEの育成確保を図っていくことである。このための施策として、1989年度より地域企業内リーダの育成研修の開催と地域ソフトウェアセンタでのSE研修の準備が開始された。また、高等教育やコンピュータ専門学校での高度技術者教育の充実施策の展開もなされている。

図表6-2-5 今後求められる人材

	分 類	職 務 内 容 等	要求される技術・知識	需要予測
		業務分野に最適の情報システム構築	適用業務知識	新規分野の広が
		要求仕様の定義。システム化の企画	要求仕様定義技術	りにより、増加
	アプリケーショ	実現可能性の検討、システム設計と	システム分析・設計手法	
	ン・エンジニア	仕様の決定 システム設計と評価	システム評価技術	
	(AE)	システムの運用・管理	システム・テスト技術	
		ユーザ・サイドに立った業務	プロジェクト・マネージメント能力	
シ			経営管理、経営工学の知識	
		ハードウェア・ソフトウェア両面で	特定のシステム資源(ハードウェア・ソフト	急増するが、全
ス		の最適システムの構築、運用、管理	ウェア・データベース・ネットワーク等) に	体に占める割合
		評価	関する専門知識	は小
テ	テクニカル・エ	コンサルテーション	コンピュータのハードウェア/ソフトウェア	
	ンジニア (TE)	情報のDB化と一元管理	技術 システム評価技術	
4		ユーザとメーカの橋渡し	データベース技術	
			情報通信技術	
			複合システム化技術	
		特定分野の最適製品やソフトウェア	特定分野の製品開発に関する専門知識	かなり増加
ェ	デベロップメン	を開発	ソフトウェア工学の専門知識	
		汎用ソフトウェア(OS, DB,	コンビュータ科学の専門知識	
ン		MS, パッケージ, 言語プロセッサ	システム工学の専門知識	
	ト・エンジニア	ー,開発支援ツール等),マイコン	製品開発管理業務	]
ジ	(DE)	組込み製品ソフト等の開発 新技	ハードウェアの専門知識	
		術の開発		
=		供給サイドに立った業務		
		ソフトウェア設計のコンサルテーシ	ソフトウェア工学の専門知識	増加
ア		ョン	プログラムの設計技術	
	• ••	プログラム開発のプロジェクト・リ	工程・品質管理技術	
	プロダクション	-ダー	プログラム技法	
	・エンジニア	プログラムの設計・開発	生産性向上技術、ツールの活用技術	
	(PE)	工程・品質・原価管理	プログラム言語の専門知識	
		結合、検査	コンピュータ科学の知識	
		保守	ハードウェアの知識	
	プログラマー	プログラム設計書に基づいてプログ	ソフトウェア工学の知識	技術進歩による
-		ラム作成	プログラムの設計技法	生産性向上で,
		プログラム・モジュール論理設計・	プログラミング技法	増加率は押えら
		製造	プログラムのテスト技法	れ、相対に減
		モジュール・テスト、結合テスト	プログラムの <del>保守技</del> 術	
		プログラム・モジュール保守改善	プログラム言語の知識	1

資料:通産省「2000年のソフトウェア人材」

### (2) コンピュータ・リテラシー

高度情報化社会の円滑な発展を支え、健全な生活を享受するためには、 企業人や社会人あるいは学生たちの情報リテラシーを啓発することが極め て重要である。ワークステーションやパソコンを駆使した情報の選択、処 理、活用および伝達能力を、誰れもが容易に備えられてこそ高度情報化社 会の開花と言えよう。それだけに、コンピュータ・リテラシーの能力開発 教育は、企業、学校、社会のすべてが取り組むべき課題である。

# ① コンピュータ・リテラシーとは

ここでいう、コンピュータ・リテラシーとは、読み、書き、ソロバン (演算)につぐ第4の能力のことである。ありあまる情報(データベース)の中から自分の問題解決や意思決定に役立つ情報をメディアを使って主体的に選択して取り出す能力、選択した情報にメディアを使って高付加価値をつけるための情報処理能力、高付加価値情報を有効に生かして問題解決や意思決定・判断に役立てる能力、さらには、高付加価値情報を自分に役立てるだけでなく、それを必要とする人達にメディアを通じて伝達する能力ーこれが情報リテラシーであり、メディア・リテラシーないしはコンピュータ・リテラシーである。

## ② 学校におけるコンピュータ・リテラシー

初等・中等教育でのパソコンの導入はかなり進展しているとはいうものの、情報リテラシー教育への利用はこれからといった状況である。パソコンは、事務処理やCAL (Computer Aided Learning)のような教育用機械としての利用のウエイトが高く、コンピュータ・リテラシーの開発の道具としての活用はまだ一般の学校では本格化していない。

一方、大学・短大・高専では一部本格化してきている。70年代に入って浸透がみられた一般情報処理教育の内容はプログラミング言語教育に終始する傾向があった。しかし、近年はパソコンの利用が容易になったことから、簡易データベースやワープロ、表計算等の学習を中心に据えたコンピュータ・リテラシー教育のカリキュラムが増加してきている。また、専修学校においても、秘書科をはじめビジネス系の学科でのコンピュータ・リテラシー教育が盛んになってきている。

### (3) 企業内教育におけるコンピュータ・リテラシー

OA化やエンドユーザ・コンピューティング(End User Computing: EUC)の推進を通じて、企業内教育でのコンピュータ・リテラシーの涵養はかなり本格化になっている。この場合OA推進室、あるいはEUC推進センター等の組織が中心となって、全社員のコンピュータ・リテラシーの教育が展開されている。最近では、戦略的情報システムを有効活用させるためのリテラシー教育も盛んである。この場合は、協力企業やグループ企業の従業員をも巻き込んだ「ぐるみ教育」として実施している場合も診しくない。

## ④ 民間団体・社会での教育

カルチャー・センタをはじめとした各種の民間団体や地域の公的機関が一般に公開しているコンピュータ・リテラシー講座やセミナーは目立って増えてきている。また、専修学校や私立の大学・短大が一般公開している講座にもリテラシーの養成を目的としたものが多く見られる。

### 2.4.3 21世紀の人材育成

21世紀へ向けての人材育成を(1)学校教育、(2)ユーザにおける人材育成、(3) ベンダにおける人材育成について説明する。

#### (1) 学校教育

### ① 初等・中等教育

文部省は、1987年12月の教育課程審議会の答申で「情報教育」がうたわれたのを受けて、1989年初頭に出された「新学習指導要領」において、各教科にコンピュータを利用する項目を盛り込んだ。

具体的には、中学校では1993年度から「技術・家庭」の「情報基礎」 の項目でコンピュータの仕組みや基本操作、情報の活用の仕方などを教 える。また、「数学」「理科」でも数量関係や実験データの処理などに コンピュータを活用することになっている。

高校では、1994年度から「数学C」で本格的にコンピュータを利用した情報処理を教える。また、「数学A」「数学B」「家庭」「物理 I B」「物理 II」の各教科でコンピュータを利用することになっている。なお小学校では「総則」の中で触れられているだけで、各教科では言及して

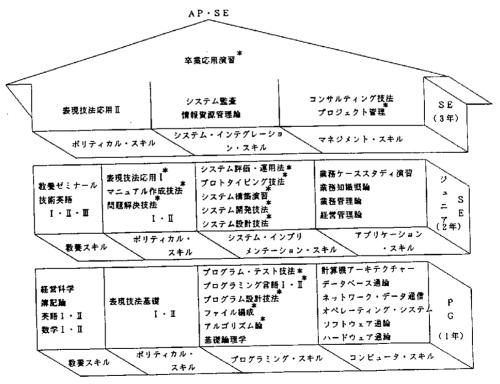
いない。

なお教育用パソコンの学校への普及は 100%導入の英、仏、米、カナ ダに比べてわが国は文部省の調査(1989年3月末現在)によると小学校 21.0%、中学校44.8%、高校96.3%と著しく遅れている。

## ② 情報処理関係専修学校

専門学校は、これまで主としてプログラマ養成を主眼に2年制ないし は1年制の学科を設置していた。しかし、これからの企業や社会ニーズ に応えていくためには、プログラマ養成よりもSE的な人材の養成が必 要になってきている。SE的な人材の育成だと、2年制では相当無理が ある。そこで、最近では3年制のSE学科を併設する学校がふえている。 今後は、SE学科を担当するにふさわしい教員の養成と、SE学科の教 育内容およびカリキュラムの充実が主要な課題となっている。

3年制学科におけるSE教育のカリキュラム編成例 図表6-2-6



(注) \*:演習・実習を含む教科

(資料) 情報処理学会研究報告 (Vol. 89, No. 73), 「専門学校におけるSE教育について」何村一樹(1989年 9 月14日)

# (2) ユーザにおける人材育成

企業にとって当面の課題は、経営に直結した情報インフラの整備とこれ を戦略的に活用しうる人材の育成であり、全社的な環境づくりによる取り 組みが必要となっている。特に、企業内での一般ユーザの情報リテラシー をいかにして高めていくかがポイントとなる。さらに、ニーズ分析やプラ ンニング等、情報システムの上流工程の仕事をこなせる管理能力を持った 技術者をユーザ企業自ら育成することが企業発展の鍵ともなる。

### ① 問題発掘能力と創造力の発揮

整備された情報インフラをベースに企業間競争が激しくなってくると、 競争上の優位性をどこに求め、かつ継続させていくかが重要となる。こ のような戦略の企画は全社員を巻き込んだ総合力発揮が鍵となる。問題 点発掘や改善を習慣化させ、業務に密着した創造力を発揮しやすい環境 を作り上げ、戦略的な仕事を創意工夫できる力を養う必要がある。

② 情報化が進むにつれてルーチン・ワーク的な業務は次々にシステム化されブラック・ボックス化されていく。このシステムの運用だけに慣れ親しんでくると、業務が持つ固有の役割に対する理解が薄れ、操作的な面だけに理解や習熟度が上がることになる。開発に参加した当初の関係者のローテーションや世代交代のつど、理解度がおちて、最低限のシステム機能しか利用できなくなってくるおそれがある。これを防止するためにも、(a) 業務処理システムをマニュアル化したりビデオ化して教育に利用したり、(b) 情報活用能力を強化するとともに、新しい仕事の仕組みを創造できる能力を養成する必要がある。

# ③ キーマン教育の推進

業務を熟知している利用部門が指導性を発揮してシステム開発を実施していくエンドユーザ・コンピューティングは、今後ますます盛んになってこよう。分散処理化によるソフトウェアの二重開発、データ資源の不整合、コンピュータ・リソースの無駄使いといった問題の発生を防ぎ、全体調整、コンサルティングや指導を行うキーマンが必要であり、その充実を図らねばならない。

### ④ 企業内SEの育成

企業内一般ユーザに対して情報技術の供給源となる情報システム部門 に今後求められるのは、高度なSE的人材の育成確保であろう。

広範な技術知識に裏打ちされた提案型、問題解決型の人材が求められる。例えば、SIS構築の企画立案ができるストラテジックSEのほか、マネジリアルSE、コンサルティングSE、アプリケーションSE、テクニカルSEなどに期待が寄せられている。

こうした人材の育成には、効果的かつ効率的な方法論に基づく徹底し た教育が必要であり、十分な教育投資を図らなければならない。

### (3) ベンダにおける人材育成

情報関連機能を提供するベンダは情報化推進の中核を担う立場であり、 ユーザ・ニーズの多様化・高度化に適応しつつ、かつ積極的な新分野の開 発努力によりユーザに高度なシステムや優れたサービスを提案していくこ とを通じて、情報関連の人材の育成・教育に寄与していくことになる。

### ① 産業分野への対応

情報システムが高度化、大規模化し、また複数の企業や異業種間にまたがるようになって、個別企業が単独で自社のシステムを構築することの難しさが増してきた。このような大規模システムの構築を一括して行うシステム・インテグレーション機能のニーズが高まっている。また、システム・コンサルティング機能、アプリケーションやシステムのパッケージ化およびネットワーク化へのニーズも急速に増大しつつある。こうしたニーズに対応できる高度な技術者の育成確保が急務となっている。

例えば、情報サービス業についてみれば、従来ややもするとユーザ企業の情報処理業務を肩代わり的に代行するきらいがあった。しかし、この産業は知識集約型、高付加価値型に転換する必要に迫られており、社内教育体制の整備、生涯キャリア・パスの策定、標準教育カリキュラムの策定等の課題を克服したうえで優秀な人材を確保し、技術力の向上に努めなければならない。

## ② 教育分野への対応

90年代の新たな局面としては、情報関連教育産業の開花が予想される。

今後、情報化教育ないしはコミュニケーション関連教育が初等・中等 教育分野にまで普及していくことは言うまでもない。

現在、国の施策として教育用コンピュータ・システムの標準化、教育ソフトの開発等への取り組みもなされており、コンピュータ・リテラシーの向上に大きく貢献しよう。

こうした動きに合わせて、既にコンピュータ・メーカのみではなくて、 出版社や教材産業、各種教育機関等が情報関連教育分野に参入し、同産 業の開花へ向けて準備を進めている。情報サービス産業も一部はその方 向に向かいつつある。この分野では、単にCAIシステムの機器やソフ トウェアを開発するだけではなくて、媒体としてCATVやパソコン通 信、FAX等のメディアミックス・システムが望まれる。したがって教 育形態としても集団教育だけではなくて、在宅または在社教育、通信教 育、さらにはサテライト教育といったさまざまな形態がとられるであろ う。こうした流れに沿った技術者やインストラクタといった人材の育成 もまた緊急の課題である。

### ③ 個人・家庭分野への対応

一般個人としての情報リテラシー向上の場も確保・充実が急がれる。 それは就業者や学生にとっては、それぞれの属する場において可能であるが、そのような場を持たない人、例えば自営・自由業の人、主婦、高齢者といった層においては、そのための社会的な環境整備を行う必要がある。既に一部で見られるように、公的機関による生涯学習カリキュラムへの取り込みやカルチャー・スクールの利用なども1つの手段であろう。あまねく広く、手軽に教育を受けられる場と需要に応じた人材、特にインストラクタが多く求められる。

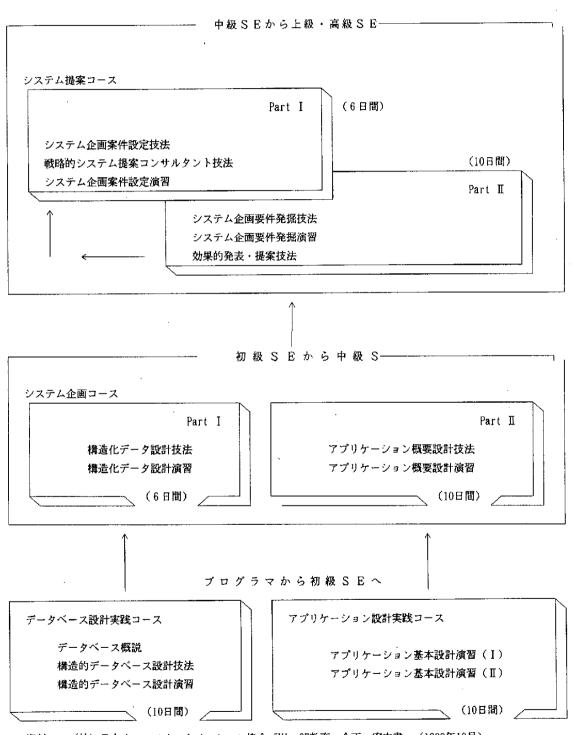
90年代には、こうした分野にもビジネスとしての取り組みが多く見られることになろう。

図表6-2-7 地域ソフトウェア供給力開発事業のSE研修カリキュラム概

部 (日)	低 要	部 (日)	銋 要
- ロシステム開発技術回	① システム開発の全過程について ② システム開発に必要な組織と要員 ③ システム分析と要求定義(基本計画) ④ 情報システムの概要設計 「情報システムの詳細設計 ⑤ かステム構築 ⑦ システム構築 ③ システム開発評価	アーキテクチャ(5)	① コンピュータ・アーキテクチャの基本概念 ② コンピュータ・アーキテクチャの一般問題と技術 ② CPUアーキテクチャ ② メモリアーキテクチャ ③ 漁信制御アーキテクチャ ② 適信制御アーキテクチャ ② 商連化方式 ② スカラ・コンピュータとベクトル・コンピュータ ① マルチプロセッサと並列処理 ② 新しいコンピュータ・アーキテクチャ
2システム構成技術(4)	システム構成技術	8システムプログラム(3)	<ul> <li>① OSの変遷</li> <li>② 各種OSの機能</li> <li>③ OSの性能評価とジョブの性能改善</li> <li>④ 開発支援ツール</li> </ul>
3プロジェクト管理(5)	① プロジェクト管理とは ② 見積り ③ 工程管理 ④ 品質管理 ⑤ 原価管理 ⑥ 外注管理	9ソフトウェア リングニアリング(5)	① ソフトウェア・エンジニアリング序論 ② ソフトウェア研究バラダイム ③ ソフトウェア・モデル ④ ソフトウェア分析・設計技法 ⑤ 形式言語表現 ⑥ ブログラミング ⑦ リアルタイム処理 ⑧ ソフトウェアの検証
4 セキュリティと システム監査(4)	<ul> <li>セキュリティ・マネジメントの方針</li> <li>リスク分析</li> <li>セキュリティの対策技法</li> <li>不測事態計画</li> <li>システム監査の基本</li> <li>システムの内部統制</li> <li>システムの信頼性、安全性、および効率性監査</li> <li>システムの企画、開発、運用業務の監査</li> </ul>	10コミュニ 大田 (5)	<ul> <li>SEにとってのコミュニケーション</li> <li>インタビュー技法</li> <li>文書化技法</li> <li>プレゼンテーション技法</li> <li>会議の進め方</li> </ul>
5データベース(5)	① データモデル ② データベース管理システム ③ データ・ディクショナリ/ディレクトリ ④ データベース・システムの構築 ⑤ データベース・システムの評価 ⑥ データベースの技術動向	11問題解決技法(5)	① SEに必要な問題解決技法 ② 問題解決の具体的プロセス ③ 問題解決のための各種技法
6 オンライン 情報処理技術(5)	① 通信ネットワークの構成と構成要素 ② 通信サービスの種類、特徴、利用方法 ③ ネットワーク・アーキテクチャの規格と実装 ④ 通信ネットワークのトラヒック設計と性能 ⑤ 電気通信関係の法制度と標準化	12効率化プログラム(0)	<ul> <li>① Σシステム入門</li> <li>② ΣΟS基礎</li> <li>③ Σシステム活用(インフラ編)</li> <li>④ Σシステム活用(事務処理分野編)</li> <li>⑤ Σシステム活用(FORTRAN編)</li> <li>⑥ Σシステム活用(利用)(マイコン分野編)</li> <li>⑦ エンステム導入と構築(FORTRAN編)</li> <li>⑥ こシステム導入と構築(FORTRAN編)</li> <li>② エシステム導入と構築(マイコン分野)</li> <li>⑥ エシステム導入と構築(マイコン分野)</li> <li>⑥ エシステム等及と標準(マイコン分野)</li> </ul>

(全66日間)

# 図表6-2-8 HI-SE (ハイレベル・システムエンジニア)教育コース



< 資料 > (社)日本オフィスオートメーション協会「HI - SE教育」企画・案内書, (1989年10月)

# 育成方法および指導上の留意点

(1) この章は前章との関連にたって情報化社会の進展の上で解決しなければならない標準化、セキュリティ、プライバシー、関係法令、および人材育成などについて講義するとともに、具体例をもって説明する必要がある。

# 用語

標準化、セキュリティ対策、コンピュータ・ウイルス、プライバシー、法令、人材育成、コンピュータ・リテラシー、コンピュータ・ネットワーク、OSI、EDI、ソフトウェア危機、インタフェース、ISDN、INTAP、インターオペラビリティー、ISO、国際登録機関、CCITT、情報伝達規約、情報表現規約、業務運用規約、取引基本規約、移植性、再利用、マルチ・バイト機能、ワクチン、自然災害、個人情報の集積、情報化教育、システム・インテグレーション、学校教育、工業所有権、特許権、著作権、発明、プライバシー権、財産、他人使用、共同使用、相互接続、労働者派遣、VDT、グレア

推薦教材

インストラクタ必読書

# 参考文献

「情報化白書1990」

(財)日本情報処理開発協会 編 コンピュータ・エージ社、1990年

2. 「EDIネットワーク化と企業単郷」

(助)日本情報処理開発協会産業情報化推進センター 編、1989年

「2010年の情報化社会とセキュリティ」 3.

(財)日本情報処理開発協会 編, 1989年

- 4. 「市場予測レポート エレクトロニクス(産業用)市場編」 日本ビジネスレポート(株)、1988年
- 5. 「エンジニアリング企業におけるCIM & SIへの対応 -エンジニアリング能力の強力に関する調査研究-」

日本機械工業連合会、(明)エンジニアリング振興協会、1989年

「情報サービス産業白書1989」 6.

(社)情報サービス産業協力 編, コンヒュータ・エーシ社, 1989年

7 「情報サービス産業白書1990」

(社)情報サービス産業協力 編。コンヒュータ・エーシ社、1990年

- 8. 「情報化白書1989」
- 編 コンピュータ・エーシ社, 1989年 9
- 的情報処理」,日本規格協会 「JISパンヤデック 10.
- 「著作権法」, 山本桂一 著, 有斐閣 11.
- 「刑 法」,藤木英雄 著,弘文堂 12.
- 13. 「現代のプライバシー」、堀部政男 著、岩波新書
- 「電気通信事業法」,官報 14.
- 15. 「労働者派遣事業の適正な運営の確保
  - および派遣労働者の就業条件の整備に関する法律」、官報
- 16. 「VDT作業における労働衛生管理のあり方」

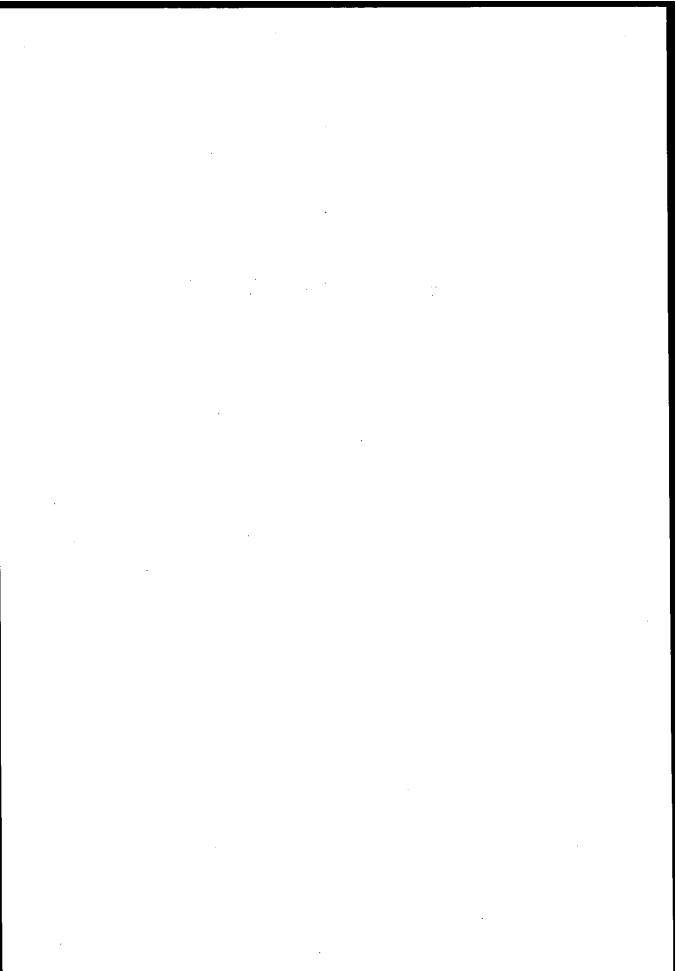
労働省労働基準局長発基発第705号,昭和60年12月20日

17. 「電子計算機システム安全対策基準」,通商産業省

# 18. 「システム監査基準解説書」

通産省機械情報産業局監修,日本情報処理開発協会 発行

# 実務能力編



#### 第7部 プログラム・アルゴリズム

## 教育目標

与えられた仕事(問題)をコンピュータで処理するためには、その問題を解くための処理手順、すなわち「アルゴリズム」を明らかにせねばならない。つまり、コンピュータのプログラムはアルゴリズムをプログラム言語で表現したものであり、アルゴリズムの構築がプログラム作成の重要な要素なのである。

当部では、このプログラム・アルゴリズムを正しくかつ適切に組み立てる能力を養成することを主眼とする。アルゴリズムを表現する技法として、現在でも「プログラム流れ図」が広く用いられていることから、流れ図によるプログラム・アルゴリズムの構築技法を教授していく。具体的には、まず、プログラム流れ図の構築技法の基本を学ぶ。その後、対象分野を「事務処理」、「科学技術計算」および両分野の「共通処理」に分け、それぞれに固有なアルゴリズムの組み立て技法を修得し、応用力を育成する。また、アルゴリズムは、単に正しいだけでなく、わかりやすいことが要求されることから、「構造化プログラム論理」にもとづくアルゴリズムの構築能力を養成することも重要な目標である。

## 内容の構成

	内	容	時間数			
			講	義	演	習
1.2	アルゴリズムを? 流れ図の種類	表現技法 ルゴリズムとは何か 流れ図で表現する 図の主要な記号と用法		3		1
第2章 2.1 2.2 2.3 2.4	3つの基本型と 3つの基本型の 直線型(連続型) 分岐型(選択型) 反復型(繰返し	整理 )の応用 )の応用		2		4
第3章 3.1 3.2 3.3	プログラム論理(構造化プログラ	ム論理の基本技法 の構造化の必要性 ミングの規則の適用 適用による構造化		5		8
第4章 4.1 4.2	事務処理のため 単一ファイル処 複数ファイル処	理	]	1 4		2 0
第5章 5.1 5.2 5.3 5.4	科学技術計算の 基本パターン 方程式の数値解 数値積分 行列式と統計計			7		9
6. 1	共通アルゴリズ 表の構造と表要 表探索の各種技 内部分類の技法	素の参照法 法		8		1 1
	アルゴリズムの 品質評価の考え 論理の正しさの	方		1		2
	合	計		4 0		5 5

## 第1章 アルゴリズムの表現技法

#### 学習目標

この章を修了した時点で、学習者が次の事柄ができるようになるのを目標と する。

- 1. アルゴリズムの表現技法として流れ図が広く使用されている理由を述べる。
- 2. 流れ図作成にさいして必要となる主要な4つの記号と、それらを用いて作成することのできる3つの流れ図の基本形を指摘する。
- 3. データ流れ図、システム流れ図とプログラム流れ図のそれぞれの主要な特 徴と目的、および概括流れ図と詳細流れ図の差を説明する。
- 4. JISで規定している「情報処理用流れ図記号」の規則にしたがって、プログラム流れ図の主要な記号の適切な使用をする。
- 5. 4.で修得した流れ図記号の書き方とそれぞれの用法を生かして、基本的な問題に関するプログラム流れ図を完成する。

## 内 容

## 1.1 プログラム・アルゴリズムとは何か

コンピュータ利用技術の総称をソフトウェアと呼ぶが、狭義にはプログラム (Program)をソフトウェアと同じ意味で使うことも多い。このプログラムは、コンピュータで処理させたい一連の手続きをプログラム言語で表現したものであり、コンピュータに与える仕事(処理)の単位ともなる。

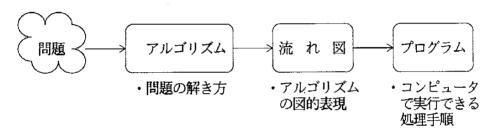
本節では、プログラムがもつ一連の手続き(以下手順と呼ぶ)の表現方法について学習する。そのため、身近かな例題を与え、次のような流れ図で演習中心に授業をすすめる。

- ① 与えられた問題(これがプログラム対象である)を解決するための作業手順を考えさせる。
- ② この作業手順を記述したものをアルゴリズム(算法: A Igorithm)と呼び、 これがプログラム作成の前提となることを理解させる。

- ③ アルゴリズムの記述方法は、文章表現でも可能(コンピュータのことを考えず手作業の場合)であるが、プログラムにより問題を解く場合は、もっと有効な表現技法があることを説明する。
- ④ プログラム向きのアルゴリズムの表現技法として、流れ図(フローチャート: flow chart)が最も多く使われていることを紹介する。

以上をまとめると、図表7-1-1 のようになるが、本節における学習のポイントを整理すると以下のようになる。

図表7-1-1 アルゴリズム、流れ図、プログラムの関係



- (1) プログラムとは
- (2) 作業手順の記述 (アルゴリズムの作成)
  - ① 問題の意味を理解する。
  - ② 作業手順を考え記述する。
- (3) プログラム用のアルゴリズム
  - ① コンピュータ向きの処理手順の表現技法
  - ② アルゴリズムの定義と特徴

## 1.2 アルゴリズムを流れ図で表現する(流れ図の基本型)

流れ図は、私達人間の行動過程や仕事の流れ等を分析して整理したり、それらを簡明に表現して伝達するのに便利な表現技法の一種である。したがって、コンピュータによる情報処理の分野だけでなく、広く一般の仕事や日常生活の中のさまざまな事柄にも活用できる手法である。そこで、本節では流れ図に関してはまったくの初心者を前提に、できるだけ身近かでなじみ易い題材をもとに流れ図の基本的な考え方や目標、一般的特徴等を把握してもらい、流れ図に親しみを持ってもらうように指導することが大切である。身近かな例示にもと

づいて、以下の内容を説明し、把握させることが肝要である。

- (1) 流れ図の目的と一般的特徴
- (2) 流れ図を表現するための基本的な記号
  - ① 端子記号
  - ② 処理記号
  - ③ 判断記号
  - ④ 線記号

以上4つの主要な記号を用いて、次項の「流れ図の基本型」を説明する。

- (3) 流れ図の基本型
  - ① 直線型(連続型)の流れ図
  - ② 分岐型 (選択型) の流れ図
  - ③ 反復型(繰返し型)の流れ図

これら3つの基本型の説明にさいしては、いずれも身近かな題材を例示 に用いること。この段階では、情報処理分野の題材にまで立ち入らない方 がよい。

#### 1.3 流れ図の種類

情報処理の分野で用いられる流れ図は、大別すると「データ流れ図」、「システム流れ図」、「プログラム流れ図」の3種類になる。それぞれの流れ図は、必要に応じ詳しさの程度をいくつかの段階に分け記述できる。特に「システム流れ図」と「プログラム流れ図」では、概括流れ図(General flow chart)と詳細流れ図(Detail flow chart) とが使い分けられることが多い。

本節では、各々の流れ図の特徴と使用目的をまず理解させることが必要である。データ流れ図は、データの経路を中心に表わせることから、事務処理システムの分析に使用できることを例示する。つぎに、システム流れ図により、事務処理のパターン(入力変換・整列併合・突合せ・出力変換など)を具体的に示す。このシステム流れ図の1つの要素(ジョブ・ステップ)が、概括プログラム流れ図に対応することを関連づけして解説する。さらに、概括プログラム流れ図は処理手順を細く展開することにより、詳細プログラム流れ図になることを把握させる。

これら一連の流れ図を関連づけて理解させるためには、事務処理分野の例題 (たとえば給与計算業務)を共通に使うことが好ましい。したがって、内容の 展開順序は次のように考える。

- (1) データ流れ図
  - ① データ流れ図とは(例示)
  - ② データ流れ図の特徴と使用目的
- (2) システム流れ図
  - ① システム流れ図とは(例示)
  - ② システム流れ図の特徴と使用目的
  - ③ システム流れ図から入手できる情報
- (3) プログラム流れ図
  - ① プログラム流れ図とは
  - ② プログラム流れ図の特徴と使用目的
  - ③ アルゴリズムとプログラム流れ図
  - ④ プログラム流れ図作成上の要点
- (4) 概括流れ図と詳細流れ図
  - ① 概括システム流れ図と詳細システム流れ図(例示)
  - ② 概括プログラム流れ図と詳細プログラム流れ図(例示)

## 1.4 プログラム流れ図の主要な記号と用法・

プログラム流れ図は、正確で高品質のプログラムを作成するための技法であると同時に、その内容を関係者に理解してもらう伝達手段でもある。したがって、そこで用いる記号や表記法には、標準化されたルールが必要である。

日本規格協会による「情報処理用流れ図・プログラム網図・システム資源図記号」(X0121-1986)は、このような主旨で定められた日本工業規格(JIS)である。この規格は、国際標準規構(ISO)による国際規格-ISO5807に準拠しており、次の10項目で構成されている。

- ① 適用範囲
- ② 用語の意味
- ③ データ流れ図

- ④ プログラム流れ図
- ⑤ システム流れ図
- ⑥ プログラム網図
- ⑦ システム資源図
- ⑧ 記 号
- 9 用流
- ① 記号用途一覧

本項では、以上の規定を前提にして、プログラム流れ図記号の具体的な用法 と流れ図作成上の規則を指導してほしい。ただし、単なる規定の解説を行うの ではなく、具体的な例題を中心にどのように適用すべきかを主体に説明を展開 するように配慮してほしい。

次の内容の順序で説明する。

- (1) その適用範囲と用法の一般事項
  - ① 適用範囲
  - ② 用法の一般事項 流れの方向
  - ③ 用法の一般事項 流れ線の交差
  - ④ 用法の一般事項 流れ線の合流
  - ⑤ 用法の一般事項 記号の変形や回転の制約
- (2) 主要な流れ図記号の種類と用途
  - ① 流れ図記号の分類(処理関係記号、ファイル操作の記号、入出力関係の記号、制御関係の記号、その他)
  - ② プログラム流れ図用の基本的な記号とその用法(端子記号、入出力記号、処理記号、線記号、準備記号、判断記号、ループ始端記号、ループ 終端記号、結合子記号、注釈記号)
  - ③ 定義済み処理記号と特定入出力記号
- (3) 流れ図作成上のその他の規則
  - ① 流れ図中の記述(本文、識別名、説明)
  - ② 結合子(出結合子、入結合子、結合)
  - ③ 横線
  - ④ 2つ以上の出口

#### ⑤ 同種類の媒体の反復表現

## 育成方法よび指導上の留意点

この章は、流れ図に関しては知識を持ち合わせていない初心者を念頭にしたものである。したがって、極めて初歩的な事柄も教授内容に含めてある。それに、できるだけ身近かな例題をもとに、分りやすく説明するように要求している。この章を通じて、学習者にプログラム流れ図を身近かなものとしてなじませ、親しみを覚えさせることが大事である。また、プログラム流れ図の適切な記号の使い方と正しい書き方を覚えさせることも大切である。この意味からすると、すでに流れ図にある程度なじみのある学習者に対しても適切な記号の使用法や正しい流れ図の書き方を徹底し、再教育することは意義がある。

学習者のレベル差が多分にあると思われるので発問を多用しながら、全員を 上手に巻き込むように工夫すること。すでに経験や知識の所有者にはそれらを 体系化するように仕向けること。初心者には興味・関心をいだかせるように配 慮すること。各節の終りでは、簡単な「練習問題」をやらせて、知識の定着を はかるようにすべきである。

## 用 語

プログラム,アルゴリズム,流れ図(フローチャート),端子記号,線記号,処理記号,判断記号,直線型(連続型),分岐型(選択型),反復型(繰返し型),データ流れ図,システム流れ図,プログラム流れ図,情報処理用記号,定義済み処理記号,準備記号,ループ始端記号,ループ終端記号,結合子記号

## 第2章 3つの基本型とその応用

## 学習目標

この章を修了した時点で学習者が流れ図の3つの基本型に関連して、次の事柄ができるようになるのを目標とする。

- 1. プログラム流れ図作成にあたっての3つの基本型を指摘し、それぞれの 使用上の主要な特徴を説明する。
- 2. プログラムの逐次制御を生かした直線型の適用、選択型の中の IF THEN ELSE型と IF THEN型の使い分け、および複合条件を駆使した選択条件の指定をする。
- 3. DO WHILE型とREPEAT UNTIL型の反復、さらにカウンタを用いた反復制御のそれぞれの適切な使い分けをする。
- 4. 与えられた問題に対して、直線型、選択型、反復型の3つの基本型を適用して、それぞれの問題に最適なプログラム流れ図を完成する。

## 内 容

#### 2.1 3つの基本型の整理

プログラム流れ図を構成する基本的パターンは、直線型(連続型)、分岐型 (選択型)および反復型(繰返し型)の3つの基本型に大別することができる。 これら3つの基本型の組合せだけでプログラム流れ図全体を完成してもらうこ とになじんでもらうのが、この章の最大のねらいである。したがって、単に基 本型のパターンを提示するだけでなく、できるだけ豊富な例題と練習問題を中 心に応用力を養わせるよう配慮することが重要になる。ただし、本節では、ま ず3つの基本的パターンを整理し、それぞれの使用上の主要な特徴を理解させ る。

- (1) 直線型(連続型)
- (2) 分岐型(選択型)
  - ① IF THEN ELSE 型

- ② IF THEN 型
- (3) 反復型(繰返し型)
  - ① DO WHILE 型
  - ② REPEAT UNTIL型

## 2.2 直線型 (連続型)の応用

直線型は、プログラムの実行順序の原則である逐次制御方式に最もふさわしいパターンである。一方、分岐型と反復型はプログラムの実行の制御の流れを変更することによって実現できる。そこで、本節では直線型の応用に先立って、プログラムの逐次制御と制御の流れの変更について説明する。そのあとで、直線型の応用について把握させる。応用に関しては、いくつかの例題を中心に説明し、最後に知識の定着と要約のために練習問題を与える。

- (1) 逐次制御と制御の流れの変更
  - ① プログラム記憶方式と逐次制御
  - ② 制御の流れの変更(飛び越し命令、無条件飛び越し命令、条件付き飛び越し命令)
- (2) 直線型の応用
  - ① 例題1-四則演算の例
  - ② 例題2-データの代入(記憶内容の置換え)
  - ③ 例題3-数字データの合計
- (3) 練習問題

## 2.3 分岐型(選択型)の応用

分岐型には、IF THEN ELSE型とIF THEN 型、および多分岐の選択が行える CASE型がある。それぞれにふさわしい例題にもとづいて説明を行い、使い分けができるようにする。また、複数個の条件の組合せで選択判断を行う場合の複合条件の指示の仕方についても、例題を通じて学習させる。

- (1) IF THEN ELSE 型
  - ① 例題1-数字データの合計
  - ② 例題2-比較判断にもとづいた選択

- (2) IF THEN 型
  - ① 例題1-比較判断にもとづいた選択
  - ② 例題2-最大値と最小値
- (3) 多分岐型(CASE型)
  - ① 例題1-種別コードの判定による選択
- ② 例題2-演算結果の数値にもとづく選択
- (4) 複合条件にもとづいた選択
  - ① 複合条件の種類(OR条件、AND条件、NOT条件)
  - ② 例題1-入力情報の組合せによる判断
  - ③ 例題2-入力情報と演算結果の数値との組合せによる選択
- (5) 練習問題

#### 2.4 反復型(繰返し型)の応用

反復型には、DO WHILE型とREPEAT UNTIL型とがある。また、REPEAT UNTIL型 の追加パターンとして、カウンタを用いた反復制御のためのDO型がある。それぞれについて、ここでも例題中心に説明を進める。

- (1) DO WHILE型
  - ① プログラム流れ図の基本的な構造
  - ② 例題1-生徒の平均身長
  - ③ 例題2-自然数の総和
- (2) REPEAT UNTIL型
  - ① プログラム流れ図の基本的な構造
  - ② 例題2-自然数の印刷
  - ③ 例題3-平方根の計算
- (3) カウンタを用いた反復制御のパターン
  - ① 反復制御用のカウンタの使い方(初期値のセット、反復処理、増加値のセット、反復条件の判定)
  - ② 例題 総和の計算
- (4) 練習問題

#### 2.5 総合演習

この章で学んだ知識を整理し、3つの基本型を組み合せた総合的な適応力ないしは応用力をつけるために、総合演習を行う。

## 育成方法および指導上の留意点

この段階では、構造化プログラミング(structured programming)を意識させない方がよい。というよりも、むしろ意識させるべきではない。どんな問題であっても、基本的な3つのパターンの組み合せで対応できることを徹底して教え込むべきである。

単に理屈として納得させるのではなく、3つの基本型だけを用いてどんな問題に対しても自然に手が動くように仕向けることである。そのためには、豊富な具体例にもとづいて説明し、各節ごとの練習問題でそれぞれの基本型の適用力をつけ、さらに章末で総合的な演習問題を与えて、3つの基本型を組み合せて流れ図を完成する能力をつけるよう指導して欲しい。

## 用 語

IF THEN ELSE型, IF THEN 型, DO WHILE型, REPEAT UNTIL型, 多分岐型, CA SE型, 条件式, 論理式。複合条件, 論理和 (OR) 条件, 論理積(AND) 条件, 否定(NOT) 条件, カウンタ, 初期値, 増加値(増分値)

#### 第3章 構造化プログラム論理の作成技法

#### 学習目標

第4章で学習予定の「事務処理のためのアルゴリズム」では、事務処理問題 (ファイル処理が中心)のアルゴリズムを扱う。事務処理の問題は、アルゴリズムはむずかしくないが、処理の内容が多く複雑なケースが多い。したがって、誰にでもわかりやすいプログラム論理の構築(構造化プログラム論理)が特に要求される。

本章では、学習者がプログラム論理の構造化に関連して、次の事柄ができる ようになることを目標とする。

- 1. プログラム論理構造化の背景とその主要な理由を述べる。
- 2. 適正プログラムの条件を理解し、不適正プログラムを適正プログラムに 修正する。
- 3. 3つの基本型をNSチャートで表現する。

## 内 容

第2章では、3つの基本型の組合せでプログラム論理を組立てるために身近かな例題を取上げ、アルゴリズムを構築した。しかしながら、事務処理を中心とする実務分野では、プログラム論理が正しいだけでなく、誰にでもわかりやすいプログラム論理であることが要求される。

本章では、わかりやすいプログラム論理の必要性および構築技法を説明する。 わかりやすいプログラム論理とは、複雑に入り組んだ論理をさけ、できるだけ 標準化した単純明快な論理の組立てで、プログラム流れ図を作成することにほ かならない。そのため、基本型の使用上に厳しい制約を設ける。つまり、1つ のパターンに対する入口は1つにし、出口も1つにするという基本原則である。 この原則を守って作成したプログラムを「適正プログラム」と呼ぶ。これは、 構造化プログラミングにおける重要な規則の1つである。

本章における指導ポイントは、適正プログラムの基本原則にしたがったプロ

グラム流れ図の作成技法であるが、NSチャート(Nassi Schneiderman Chart) による表現技法も併せて学習し、構造化プログラム論理の構築技法の理解度を深めたい。

#### 3.1 プログラム論理の構造化の必要性

ここでは、プログラム論理の構造化が必要になった背景を概観し、わかりや すいプログラム論理の重要性とその要件を明らかにする。

- (1) ハードウェア技術の進歩と価格性能比の向上
  - ① ハードウェア技術の長足の進歩
  - ② 価格性能比の飛躍的向上
- (2) 従来のプログラミング技法の反省
  - ① ソフトウェア開発のあり方の問題
  - ② プログラムの維持保守の負担増
  - ③ 人件費比率の増大化傾向
  - ④ システムやプログラムの高度・複雑化
- (3) プログラム論理の構造化の重要性
  - ① わかりやすさの必要性(信頼性の向上、保守の容易性、拡張性の保証)
  - ② わかりやすさの要件(プログラムのモジュール分割、機能の階層構造化、モジュールの独立性、モジュール内でのプログラム論理の構造化)

## 3.2 構造化プログラミングの規則の適用

本節では、まずわかりやすい論理を組み立てるさいの原則を明らかにする。 それにもとづいて、不適正なプログラム論理を適正なプログラム論理に構造化 するやり方を具体例で示す。

- (1) 分かりやすいプログラム論理の組立て方
  - ① 適正でないプログラム流れ図(入口が複数個の場合、出口が複数個の場合)
  - ② プログラム論理の構造化(基本構造化単位―順次・選択・反復、拡張 構造単位―反復・選択、適正プログラム)
- (2) 適正プログラムへの論理の構造化

- ① 入口が複数個の例題の適正構造化
- ② 出口が複数個の例題の適正構造化

#### 3.3 NSチャートの適用による構造化

これまで見てきたプログラム流れ図も、構造化の有効な手段ではあるが、表記法の自由度が高いため、ともするとプログラム論理を非構造的に作成してしまう危険性がある。これは、判断とか処理を記述する記号間の結び方に特に制限が設けられていないことに起因する。

NSチャートは、別名構造化チャートとも呼ばれるように、プログラム論理 の構造化向きの技法であり、線記号が飛び交ったりしないような表現方法をと っている。

本節では、3つの基本型をNSチャートで表現する方法を学習する。プログラム流れ図と対比し両者の関係をじっくり把握させることにより、構造化プログラム論理の構築技法をマスタさせてほしい。

- (1) NSチャートとは
  - ① NSチャートの特徴
- (2) NSチャートの記述の仕方
  - ① 直線型(連続型)
  - ② 分岐型(選択型)
    - (a) IF THEN ELSE型
    - (b) IF THEN型
    - (c) CASE型(多分岐型)
  - ③ 反復型(繰返し型)
    - (a) DO WHILE型
    - (b) REPEAT UNTIL型

#### 育成方法および指導上の留意点

個人の研究や趣味で作るプログラムは別として、企業や団体など組織体の中で作るプログラムは私的な財産ではなく組織の公的な財産である。公的なプロ

グラムとして有効に機能させるためには、個人芸や個性が強すぎるのは困りものである。誰にとっても理解しやすく、わかりやすいものであることが優先する。学習者にこのことを十分認識させたうえで、わかりやすさを実現するためのプログラム論理の構造化を徹底させる必要がある。

本章では特に1つの入口と1つの出口による適正プログラムの論理の組立てを中心に指導してほしい。これまでに示してきた例題の中で適正でない流れ図を引用して、その適正化のやり方を具体化する方法で指導するのが効果的である。

構造化プログラムの表現方法として、「流れ図」に加えて「NSチャート」を示したが、最近は「木構造チャート」がプログラムの制御構造の新しい図的表現方法として使われ始めている。

木構造チャートは、構造化プログラミングに適する用にプログラムの基本的な制御構造を木の形で表現する技法である。代表的なものとして、HCP(Hie rarchical and Compact descyiption Chart : NTT)、YACI(Yet Anot her Control chart Ⅱ:富士通)、PAD(Problem Analysis Diagram : 日立) SPD(Structured Programming Diagram : 日本電気) などがある。

注目しなければならないことは、この木構造チャートからプログラムを自動生成するシステムの開発が進められていることである。正しいアルゴリズムが構築できれば、プログラミングは不用になる時代が遠からずやってくることを学習者にも知らせておくべきである。

## 用語

構造化プログラミング (ストラクチャード・プログラミング),適正プログラム,モジュール,モジュール分割,モジュールの独立性,機能の階層構造化

#### 第4章 事務処理のためのアルゴリズム

#### 学習目標

この章を修了した時点で、学習者がファイルと流れ図に関連した次の事柄が できるようになることを目標とする。

- 1. 事務処理分野でのデータ処理の基本的な内容と主要な処理パターンを説明する。
- 2. ファイルを構成する諸要素を列挙し、また主要なファイルの種類とその 用途を説明する。
- 3. ファイル処理の主要な作業内容を説明し、ファイル処理プログラムの基本構造を構造化プログラム論理の原則に従って流れ図にまとめる。
- 4. 次の処理内容を含んだ単数入力ファイルを処理するためのプログラム流 れ図を作成する。
  - (1) 入力レコードの検査
  - (2) 明細印刷と改ページ制御
  - (3) 配列検査
  - (4) 単一レベルのグループ制御
  - (5) 多重レベルのグループ制御と合計印刷
- 5. 複数ファイル処理における突合せ型のファイル処理の4つの種類を列挙 し、それらに共通する処理技法上のポイントを指摘する。
- 6. 4つのファイル処理に関するプログラム流れ図を、構造化プログラム論 理の原則にしたがって完成する。
  - (1) ファイルの併合処理
  - (2) ファイルの突合せ処理
  - (3) ファイルの更新処理
  - (4) ファイルの維持保守

## 内 容

事務処理分野での仕事の形態は、要約すると次の2つのパターンに大別できる(図表4-1-1 参照)。

- ①元帳とか台帳とか称される帳簿を読み、集計を行い報告書を作成する作業…(a)
- ②伝票類の内容を帳簿に転記したり、追加・修正の記録などを行って帳簿 の内容を更新する作業・・(b)

図表 4-1-1 事務処理のパターン

以上2つの作業とも、帳簿(ファイル)の読み書き動作が中心になっている。 したがって、これらをコンピュータのプログラムで行わせようとする場合も、 手作業で遂行するのと同様に、ファイルの入出力が主体になってくる。このよ うに、ファイルの入出力を中心に、ファイルの内容を更新したり、修正・追加 したり、新しいファイルを作成したりするプログラム処理を、一般にファイル 処理という。

本章では、①に相当する処理を「単一ファイル処理」、②に相当する処理を「複数ファイル処理」として、それぞれのプログラム・アルゴリズムの構築方法を学習する。

#### 4.1 単一ファイル処理

(1) ファイル処理の特性

本節は、入力ファイルを一種類に限定した場合のファイル処理の基本的な論理の組立てとそれを応用した具体的なプログラム流れ図の作成技法を修得させるのが主眼である。しかし、上記のねらいに専念させるためには、ファイル処理の流れ図の説明に先立って、ファイルの構成要素や種類、ファイル処理の考え方やその基本的内容、あるいはファイル処理のプログラム論理の基本的な構造などを理解させておく必要がある。そうすることによって、次項のファイル処理のプログラム流れ図を学習する段階では、ファイルに関する余計なことに気をまわす必要がなくなり、流れ図の作成という中心課題にだけ注意を集中することができるようになる。ファイルとファイル処理の基本概念を把握させるために、以下の内容順序で説明する。

- ① 事務処理分野での基本的なデータ処理
  - (a) データ処理の基本的要素とその内容(データの発生、データの記録、 データの処理、報告書の作成、データの保管)
  - (b) 事務処理のパターン(報告書の作成事務、帳簿の内容更新事務)
  - (c) 事務処理の性格とコンピュータ処理の特徴
- ② ファイル処理の概念と基本的内容
  - (a) ファイルの基本構成(けた、バイト、フィールド、項目または欄、 レコード、論理レコード、物理レコード、ブロック化係数)
  - (b) ファイルの種類(マスタ・ファイル、トランザクション・ファイル、 ワーク・ファイルなど)
  - (c) マスタ・ファイルとトランザクション・ファイルの関係
  - (d) 順処理ファイルと乱処理ファイル
- (e) ファイル処理の基本的内容(ファイルの入力媒体変換、ファイルの 分類、ファイルの生成、ファイルの突合せ、ファイルの更新、ファイ ルの出力媒体変換)
- (f) ファイル処理プログラムの基本構造(処理の前準備、処理の本体、 後処理)

#### (2) 単一ファイル処理のプログラム流れ図

本項は、すでに修得したプログラム流れ図作成上の基本的なパターンと 構造化プログラミング技法とを応用して、事務処理分野の実務的問題のプログラム流れ図を組み上げる力をつけてもらう。一見複雑そうに見える問題であっても、そのほとんどが3つの基本型の組合せでプログラム流れ図を完成することができる。このことを本項を通じて身をもって体験させることである。

本項では、1種類の入力ファイルにもとづいて処理を行う典型的なファイルの基本的な論理と、それを応用した具体的なプログラム流れ図の作成 技法を例題中心に解説し、最後に総合的な練習問題で知識を整理し、応用 カへと高めてもらう。

- ① ファイルの入力媒体変換と入力レコードの検査
  - (a) ファイルの入力媒体変換の論理(入力媒体変換の例題説明、概括プログラム流れ図、詳細プログラム流れ図)
- (b) 入力レコードの検査の論理(入力データの検査の種類とその概略、 入力データの検査論理の組込み、入力データの検査を伴った例題)
- ② 明細印刷と改ページ制御
  - (a) 明細印刷の論理(明細印刷のプログラム論理の基本構造、明細印刷の例題説明、詳細プログラム流れ図)
  - (b) 改ページ制御の論理の組込み(改ページ制御を組み込んだプログラム論理の基本構造、明細印刷に改ページ制御を組み込んだ例題、記憶域の設定と改ページ制御のプログラム論理、サブルーチンを用いた詳細プログラム流れ図、同一の改ページ制御を共有した詳細プログラム流れ図)
- ③ 配列検査とグループ制御
  - (a) 配列検査の論理 (レコードの配列順序、配列検査の基本論理、配列 検査を含んだプログラム論理の基本構造、配列誤りの処理方法)
  - (b) グループ制御の論理(グループ制御の考え方、グループ制御を含ん だプログラム論理の基本的構造、グループ制御の例題説明、概括プロ グラム流れ図、詳細プログラム流れ図)

- ④ 合計処理と報告書作成
  - (a) 明細印刷とグループ合計印刷
  - (b) グループ合計印刷の例題と詳細プログラム流れ図
  - (c) 多重レベルのグループ制御の論理(多重グループ制御の考え方、多重グループ制御を含んだプログラム論理の基本的構造、多重グループ制御の例題説明、詳細プログラム流れ図)

#### 4.2 複数ファイル処理

前節で取り扱ったファイル処理では、入力されるファイルを1種類に限定した。本節では、入力されるファイルが複数個である場合のファイル処理の論理を取り上げる。いわゆる突合せ形(マッチング・タイプ)のファイル処理の理論である。

突合せ形のファイル処理は、ファイルの併合(ファイル・マージ)、ファイルの突合せ(ファイル・マッチング)、ファイルの更新(ファイル・アップデート)、ファイルの維持保守(ファイル・メンテナンス)の4種類のタイプに大別できる。それぞれについて、ファイル処理の基本的な論理と、それを応用発展させた具体的なプログラム流れ図の作成技法を、例題を中心に説明して行く。

- (1) 突合せ形ファイル処理の基本的なポイント 複数ファイル処理の流れ図の説明に先立って、突合せ形処理の種類とそ れぞれの処理に共通する技法上のポイントについてまず説明しておく。
  - ① 突合せ形のファイル処理とは
  - ② 突合せ形処理の種類
    - (a) ファイルの併合(ファイル・マージ)
    - (b) ファイルの**突**合せ (ファイル・マッチング)
    - (c) ファイルの更新 (ファイル・アップデート)
    - (d) ファイルの維持保守(ファイル・メンテナンス)
  - ③ 突合せ形処理の技法上のポイント
- (2) ファイルの併合処理
  - ① ファイルの併合処理の概略

- ② ファイルの併合処理のプログラム論理
  - (a) 併合処理プログラムの概略的な構造
  - (h) 併合処理の基本概念
  - (c) 併合処理部分のプログラム論理(プログラム・スイッチによる方法、 最大値の注入による方法)
- ③ 例題でみるファイルの併合処理
  - (a) 例題の説明
  - (b) プログラム流れ図の説明
- (3) ファイルの突合せ処理
  - ① ファイルの突合せ処理の概念 ファイルの突き合わせ処理とは、あらかじめキー項目について分類済 みの2つ以上の入力ファイルについて、相互にレコードのキー項目を比 較し、相手となるレコードを見つけ出して処理するやり方である。
  - ② ファイルの突合せ処理のプログラム論理
    - (a) 突合せ処理プログラムの概略的な構造
    - (h) 突合せ処理の基本論理
    - (c) **突合せ処理部分のプログラム論理**(プログラム・スイッチによる方法、最大値の挿入による方法)
  - ③1対1の突合せ処理
    - (a) 例題の説明
    - (b) プログラム流れ図の説明
  - ④1対nの突合せ処理
    - (a) 1対nの突合せ処理の基本論理
    - (b) 例題の説明
    - (c) プログラム流れ図の説明
- (4) ファイルの更新処理
  - ① ファイルの更新処理の概念

更新処理は、1対nの突合せ処理の応用であり、取引ファイルとマスタ・ファイルとを突合せて、両者のレコードのキーが一致したとき、取引レコードの変動情報でマスタ・ファイルの対応レコードの特定項目の

内容を最新のものにする処理である。

- ② ファイルの更新処理のプログラム論理
  - (a) 更新処理のプログラム概略的構造
  - (b) 更新処理の基本論理
  - (c) 更新処理部分のプログラム論理(プログラム・スイッチによる方法、 最大値を挿入する方法)
- ③ 例題でみるファイルの更新処理
  - (a) 例題の説明
  - (b) プログラム流れ図の説明
- (5) ファイルの維持保持のプログラム論理
  - ① ファイルの維持保持の概略
    - (a) レコードの新規追加
    - (b) レコードの削除
    - (c) レコードの内容変更
  - ② ファイルの維持保持の基本論理
  - ③ 例題でみるファイルの維持保守
    - (a) 例題の説明
    - (b) プログラム流れ図の説明

#### 4.3 総合演習

#### 育成方法および指導上の留意点

第1節1項の「ファイル処理の特性」に関しては、余り深入りする必要はない。当節の後半や第2節の「複数ファイル処理」の中で取扱う例題を解釈する上で支障が生じない程度のファイルに関する基礎知識を修得させておけばよい。ファイルの詳細に関しては、第8部「ファイルとデータベース」で説明する。

第1節第2項の「単一ファイル処理」では、流れ図の3つの基本型がどの場合でも適用できることを納得させ、それらが駆使できるように仕向ける必要が

ある。

第2節「複数ファイル処理」は、事務処理分野で最も重要なアルゴリズムを 含んでいる。特にファイルの突合せ処理は、ファイルの更新、ファイルの維持 保守の基本となる処理であるので、学習者にできるだけ考えさせ、納得させる ように指導法を駆使することが望ましい。

実務で使うプログラムは対象となる業務がファイルを扱う場合には、当該ファイルでアクセスエラーが発生したり、読込んだレコードに以上データが含まれていた場合に「特別な処理」を行うことがある。これは「エラー処理ルーチン」として標準化されている(個々の企業等の標準化規約など)場合も多い。特に事務処理分野では、適用業務に固有な約束ごとも多く、実務的な処理手続になじむことが大切になってくる。

#### 用 語

ファイル処理、媒体変換、分類、併合、入力レコードの検査、使用文字検査、コード検査、範囲検査(限界検査)、検査数字桁チェック、妥当性検査、ハッシュ・トータル検査、順番検査、改ページ制御、グループ制御、キー項目、制御グループ、コントロール・ブレイク(Control Break)、多重グループ制御、マスタ・ファイル、トランザクション・ファイル、突合せ、ファイルの併合(ファイル・マージ)、ファイルの突合せ(ファイル・マッチング)、ファイル更新(ファイル・アップデート)、ファイルの維持保守(ファイル・メンテナンス)、プログラム・スイッチ

## 第5章 科学技術計算のためのアルゴリズム

## 学習目標

この章を修了した時点で、学習者は科学技術計算に関連した次の要素を組み込んだプログラム流れ図が作成できるようになることを目標とする。

- 1. 基本的技法(平均と標準偏差、素因数分解、最大公約数、計算誤差、相 関係数)
- 2. 方程式の数値解法
- 3. 数值積分
- 4. 行列演算と統計計算

## 内 容

#### 5.1 基本的パターン

- (1) 平均と標準偏差
- (2) 素因数分解
- (3) 最大公約数
- (4) 計算誤差
- (5) 相関係数

## 5.2 方程式の数値解法

- (1) 二分法
- (2) ニュートン法
- (3) 掃き出し法

## 5.3 数 值 積 分

- (1) 台形則
- (2) シンプソン則

#### 5.4 行列演算と統計計算

- (1) 行列演算と行列式
- (2) 統計計算

#### 5.5 総合練習問題

#### 育成方法および指導上の留意点

この章は、全員必須にする必要はない。科学技術計算の分野の仕事に進みたい学習者やFORTRAN言語のような科学技術計算向きのプログラム言語でプログラムを作成する立場にある学習者には必須である。事務計算の分野を目指す人に対しては、本章を省略してよい。なお、科学技術計算の分野を目標にするのなら、第4章のファイル処理技法に関しては余り深入りする必要はない。ただし、第4章を無視することはできない。

他の章と同様に、当章も具体的な例題を中心に指導を進めてほしい。

## 用語

平均,標準偏差,相関関係,素数,素因数分解,最大公約数,計算誤差,方程式の数値解法,二分法,ニュートン法,数値積分,行列演算,行列式。文字列処理,ビット処理,数の基底変換

#### 第6章 共通アルゴリズム

#### 学習目標

私達の日常生活や仕事の中では、各種の料金表や税率表、時刻表などの表を使うことがしばしばある。これと同様、事務計算、科学計算の別を問わず、コンピュータによるデータ処理でも「表を操作する」ことは重要なテーマである。この章を修了した時点で、学習者が表操作と内部分類に関して、次の事柄ができるようになることを目標とする。

- 1. 次の3種類の表について、それぞれの構造とその要素を参照し処理する やり方を説明する。
  - (1) 一次元の表の場合
  - (2) 二次元の表の場合
  - (3) 三次元の表の場合
- 2. 次の表探索のプログラム論理を組み込んだプログラム流れ図を作成する。
  - (1) 順次探索
  - (2) 複合表の探索
  - (3) 区間探索
  - (4) 直接探索
  - (5) 二分探索
  - (6) 二次元の表探索
- 3. 内部分類の技法を説明し、そのうちの交換分類法のプログラム論理を組み込んだプログラム流れ図を作成する。

## 内容

データ処理の技法にはファイルの入出力処理には直接関係しない主記憶装置 内部だけでの処理技法もある。その代表的な技法が、表操作と内部分類の技法 である。

規則的に連続した同種のデータを表(テーブル)とみなして処理することに

よって、プログラムの作成を容易にしたり、コンピュータ内部での処理効率を 高めることができる。したがって、本章ではまず表操作を取り上げ、表の構造 と表要素の参照の仕方、および表探索の技法とその論理について解説する。

一方、データやレコードを昇順ないしは降順に並べ換える分類処理は、事務計算の分野では不可欠な技法である。ファイルの入出力を伴ったレコードの外部分類は、汎用プログラムや汎用ルーチンを用いるのが一般的であるので、ここでは主記憶装置内にある複数個の同一形式のデータの配列順序を整える内部分類のプログラム論理に焦点をあてて説明する。

## 6.1 表の構造と表要素の参照法

- (1) 一次元の表の場合
  - ① 一次元の表の構造(表、配列、表要素、配列要素、表名、配列名、添字、指標、限界値、表操作、配列処理)
  - ② 表要素を参照し処理する例題と流れ図
- (2) 二次元の表の場合
  - ① 二次元の表の構造
  - ② 表要素を参照し処理する例題と流れ図
- (3) 三次元の表の場合
  - ① 三次元の表の構造
  - ② 表要素を参照し処理する例題と流れ図
- (4) 表の初期値の設定方法

## 6.2 表探索の各種技法

- (1) 順次探索のプログラム論理
  - ① 表探索のための基本的な処理ステップ
  - ② 順次探索の例題とプログラム論理
- (2) 複合表とその探査技法
  - ① 複合表の探索の仕方
  - ② 複合表の探索例題とプログラム流れ図
- (3) 区間探索の手法とプログラム論理

- ① 区間探索の仕方
- ② 区間探索の例題とプログラム流れ図
- (4) 直接探査の技法とプログラム論理
  - ① 直接探索の仕方
  - ② 直接探索の例題とプログラム流れ図
- (5) 二分探索の手法とプログラム論理
  - ① 二分探索の仕方
  - ② 二分探索の例題とプログラム流れ図
- (6) 二次元の表探索技法
  - ① 二次元の表探索の仕方
- ② 二次元の表探索の例題とプログラム流れ図
- (7) 表操作の練習問題

#### 6.3 内部分類の技法とプログラム論理

- (1) 分類の基本概念
  - ① 内部分類と外部分類
  - ② 外部分類の基本ステップ(分類フェーズ、併合フェーズ、最終併合フェーズ)
  - ③ 内部分類の種類(挿入分類法、選択分類法、交換分類法)
- (2) 交換分類法の技法とプログラム論理
  - ① 交換分類法の考え方とデータ交換の論理
  - ② 交換分類法による内部分類の実行過程
  - ③ 交換分類法の例題とプログラム論理
- (3) 内部分類の練習問題

## 育成方法および指導上の留意点

前章までの学習で、学習者にはプログラム論理を組み立てる能力はかなりついてきている。それに、構造化プログラム論理で流れ図を作成することを原則にしているので、判読しやすい論理に十分なじんでおり、プログラム流れ図の

解読力はあるはずである。したがって、当章では表の構造と要素の参照の仕方や表探索の手順などをしっかり把握するように指導するとよい。あとは、例題とその処理ロジックをできるだけ各人が追跡し、自ら確認するように指導する。例題のプログラム流れ図を逐一詳細に説明する必要はない。

当章は、事務処理分野と科学技術分野ともに学習する。事務処理では、表の 内容(表要素を構成する値)は共通テーブル(テーブルファイル)として独立 させている場合が多い。業務によっては、表要素がかなり大きいものがある。 従って、表操作のアルゴリズムの良し悪しがプログラムの実行スピード(処理 速度)にも影響を与えることを十分に留意した上で正しいアルゴリズムを構築 するよう心がける必要がある。

## 用 語

表(テーブル),配列,表要素(配列要素),表名(配列名),添字(指標)限界値,表操作(配列処理),一次元配列,二次元配列,三次元配列,行列,表探索(テーブル・サーチ),表索引,探索引数。テーブル引数(表引数),順次探索,複合表,区間探索,直接探索,二分探索,分類キー,昇順、降順、外部分類,内部分類,交換分類法,挿入分類法,選択分類法、バランス型分類法

#### 第7章 アルゴリズムの検証

#### 学習目標

この章を修了した時点で、学習者はアルゴリズムの検証に関して、次の事柄ができるようになることを目標とする。

- 1. よいプログラム・アルゴリズムの条件は何かを説明でき、作成したアルゴリズムを見直し、もし問題があれば、それを指摘し、改良することができる。
- 2. 組織的なレビューの手法として、構造化ウォーク・スルーの意義を認識し 体験する。これを通して、アルゴリズムを検証する習慣を身につける。

## 内 容

本章では、でき上がったプログラム・アルゴリズムを見直し、よいプログラム論理の条件を満たしているかどうかを検証する方法を学習する。すでに学んだ構造化プログラム論理の作成技法における「適正プログラム」は、よいプログラム論理の条件に入るはずである。

ここでは、よいプログラム・アルゴリズムの条件となる品質評価の考え方お よびアルゴリズムの検証方法を学習する。

#### 7.1 品質評価の考え方

ソフトウェアの品質に対する指標として、たとえば、

- ① 働き (機能・性能・操作性)
- ② 正確さ(信頼性・可用性)
- ③ 適用範囲(互換性・可搬性・拡張性)

などがあげられる。

それでは、よいプログラム・アルゴリズムの条件は何かを考えてみよう。い ろいろあげられると思うが、結局次の3点に要約できる。

① 論理的に正しいものであること

- ② わかりやすく、簡素に表現されていること
- ③ 実際にプログラムに変換しやすく、論理的に無駄のないこと これらを実現するためには、プログラム流れ図を書いた後必ず見直しを行い、 内容の検証と表現の推敲を行うことを習慣づけることを指導してほしい。

具体的なチェックポイントとして、次の2つの場合を指摘し、演習させると よい。

- ① 共通処理が存在する場合の改良
- ② サブルーチンによる改良

#### 7.2 論理の正しさの検証

プログラム・アルゴリズムの見直しは、まず作成者が独自で行うことが必要 であるが、関係者数人が集って問題点を徹底的に洗い出す方法が効果的である。 この組織的に行うやり方として、構造化ウォーク・スルーという方法がある。

さらに、作成したプログラム・アルゴリズムを歩行可能にさせるプロトタイピングにより、アルゴリズムの正しさを検証することもできるようになってきた。たとえば、木構造チャート(tree structure chart)に基づくプログラムの自動生成がある。これらの技法を有効に活用することにより、プログラミングを行わずアルゴリズムの検証まで可能となるわけである。

本節では、構造化ウォーク・スルーの手法にしたがった簡単なレビューを実施し、その雰囲気を体験させることが目標となる。プロトタイピングについては、その考え方を紹介する程度に留めておけば良い。

## 育成方法および指導上の留意点

実務でのソフトウェア開発は、プロジェクト(チーム)により進められる。開発途上の随所でレビューが行われることを認識させることが大切である。したがって、プログラム・アルゴリズムの作成後、独自の見直しのほかに、組織的にレビューを行うことが有効であり、よいアルゴリズム作りの条件であることを理解させてほしい。

また、将来的にはプログラミングという工程をなくす(自動プログラミング)

ことにより、ソフトウェア開発の生産性を高めようとしている。したがって、 よいプログラム・アルゴリズムを作成することが、より重要な意味をもつこと を強調してほしい。

#### 用語

よいプログラム、アルゴリズムの検証、プロトタイピング、自動プログラミング、構造化ウォーク・スルー、レビュー、プロジェクト

## 参考文献

- 1.「ソフトウェア・エンジニアリング序説」 ロジャー・S・ブレスマン 著、岸田孝一監訳、TBS出版会、1983年
- 2.「プログラム流れ図の作成技法」 江村潤朗・野津昭共著,オーム社,1981年
- 3.「プログラム流れ図の作成技法」 日本情報処理開発協会情報処理研修センター, 1987年
- 4.「ソフトウェア・プロトタイピング」 有**澤誠著**,近代科学社,1986年
- 5.「アルゴリズムの作り方」 澤田晃著, 共立出版, 1989年
- 6.「システムの設計と開発」 沖山豊・河村一樹・北村公彦・菅野篤共著,ムベル出版, 1990年

## 第8部 ファイルとデータベース

## 教育目標

当部の目標は、初級技術者によるファイルとデータベースに関する知識を確 実に自分のものにしてもらうことである。なかに多少の演習があるが、それは 技術の修得よりは、知識を間違いないものにすることの意味の方が大きい。

ファイルは言語と並んでソフトウェアのすべての分野に関わりのあるテーマで、日常的にもごく親しいものである。それだけに一旦特定の専門分野なりプロジェクトなりに携わるようになると、かえって全体を見渡すことが困難になる。初級のレベルで、横断的に勉強しておくことが大切である。

- 1. ファイルの概念を理解し、各種の編成法について情報処理システムと関係 づけて特徴を理解する。コンピュータ技術は、ハードもソフトもファイルを 巡って進歩をうながされて来た面がかなり大きい。それがまた新しい応用を 生み、普及と発展を助けて来た。そのような見方でファイルを見させて欲し いと思う。
- 2. ファイルの作成、レコードの処理に必要なプログラムの仕様を説明できる。
- 3. ファイルの記憶媒体について、ファイル編成法との関係を説明し、また、必要な性能計算ができる。自分の設計したプログラムはもちろん他人の設計したもので、ファイルに関係するプログラムの仕様を、他人に説明する程度に把握できることは初級技術者にも必要である。媒体の知識についても同様である。そのためには、もちろん当部のみでなく他の勉強も必要であるが、少くとも当部の教育目標の一つでなければならない。
- 4. データベースの概念を理解し、利用者として必要なレベルでDBMS(データベース管理システム)の特徴を理解する。自ら必要な調査研究を行えるようになる。データベースは、当初考えられたようなものがその通り実現したとは言えない。しかし、情報処理システムのあるべき姿を示すものであり、環境や条件の変化はあっても方向は変らない。新しい技術や製品を適切に受け止める準備が必要である。

# 内容の構成

	rfm tiv	時間	1 数
	内容	講義	演習
	ファイルの基本概念 レコードとファイル 情報処理システムとファイル ファイルの種類	4	·
第2章 2.1 2.2 2.3 2.4 2.5 2.6 2.7	ファイルと記憶媒体 磁気テープ装置 磁気テープファイル形式 磁気テープの記憶容量と処理時間 磁気ディスク装置 磁気ディスクファイルの形式 磁気ディスクの記憶容量と処理時間 フレシキブル・ディスク装置	7	2
第3章 3.1 3.2 3.3	順編成ファイル 磁気テープ順編成ファイル作成と処理 磁気ディスク順編成ファイルの作成と 処理 順編成ファイルの代表的な応用例	5	1
4. 1	直編成ファイル 直編成ファイル 直編成ファイル	3	0.5
	索引順編成ファイル 索引順編成ファイルの構造 索引順編成ファイルの作成と処理	4	
6. 1	区分編成ファイル 区分編成ファイルの構造 区分編成ファイルの作成と処理	1	·

(続く)

(続き)

内容			時 間 数		
	<b>内</b> 容		講義	演習	
7. 1 7. 2	VSAMファイル VSAMファイルの VSAMデータセッ レコードのアクセス	トの種類	5		
8. 1	パソコン用ファイル パソコン用ファイル ファイル <b>操</b> 作の <b>演習</b>		l	1.5	
9. 1 9. 2 9. 3	データベースの概念 ファイルからデータ CODASYL型デ 関係型データベース データ構造		7. 5	0. 5	
10. 1 10. 2	データベース管理シ DBMSの機能 データベース言語 各種の実用DBMS	ステム (DBMS)	2		
	合	計	3 9. 5	<b>5.</b> 5	

## 第1章 ファイルの基本概念

## 学習目標

学習者が次のことができるのが目標である。

- 1. ファイルの概念と各種のレコード形式を説明する。
- 2. 各種ファイル編成法のちがいを説明する。
- 3. 各種ファイル編成法の情報処理システムにおける利用法のちがいを説明する。

## 内 容

### 1.1 レコードとファイル(90分)

(1) ビットからファイルまで 簡単なファイルの実例を引用して次の事項を説明する。

① a. ビット 情報の物理的に取扱える最小単位。

b. 項目(フィールド) 論理的に意味のある情報の最小単位。

c. レコード 項目の集まりで実体に対応した論理的な意味を持ち、情報処理の対象となる単位。

d. ファイル レコードの集まりで、情報処理の実行の対象となる単位。

(2) 各種のファイル記憶媒体

本指針のハードウェアの章の内容を引用し、可能ならば実物を示して次の事項を説明する。

- ① 磁気テープ、磁気ディスク、フレキシブル・ディスク、光ディスク、 印刷用紙その他の概要。
- (3) レコード形式 図表を用いて各種のレコード形式を説明する。磁気テープまたは磁気ディスク上の形式を引用して説明する。
  - ① 非ブロック化固定長レコード、可変長レコード。

- ② ブロック化固定長レコード、可変長レコード、ブロック化係数の意味。
- ③ 不定長レコード、スパンド・レコード。
- (4) 論理レコードと物理レコード 両者の差異を明確に説明する。
  - 論理レコード 情報処理の対象となる単位。
     狭義のレコードと同じ。

物理レコード 媒体上で取扱いの対象となる単位。 ブロックと同じ。

### 1.2 情報処理システムとファイル(90分)

- (1) 順呼出しファイルと乱呼出しファイル 次の事項を説明する。必要ならば図表を使用する。
  - ① 順呼出し(順アクセス) ファイルの先頭からレコードを1つずつ読んで(書いて)いく方法

乱呼出し(乱アクセス、直接アクセス) キーの値またはアドレスを 指定して目的のレコードだけを読む(書く)方法

- ② 順編成ファイル 順呼出しだけが出来る。磁気テープ・ファイル、磁 気ディスク・ファイル。印刷ファイル。
  - 特徴 a. 媒体の利用効率がよく経済的。
    - b. ファイルの大きさに制限がない。
    - c. 小規模なプログラムで処理出来る。
    - d. アクセス時間が長い。
- ③ 直編成ファイル 順呼出し、乱呼出しが出来る。磁気ディスク・ファイルのみ。
  - 特徴 a. 媒体の利用効率がよくない。
    - b. アクセス時間は短い。
    - c. 小規模なファイルに適する。
- ④ 索引順編成ファイル 順呼出し、乱呼出しが出来る。磁気ディスク・ イァイルのみ。
  - 特徴 a. 大規模なファイルを作ることが出来る。

- b. アクセス時間は短い。
- c. 比較的大規模なシステムに適する。
- ⑤ VSAMファイル 上記3種をすべて実現出来る。磁気ディスク・ファイルのみ。

特徴 仮想記憶オペレーティング・システムに固有のファイル。

- (2) 情報処理とファイル次の事項を説明する。
  - ① バッチ処理の場合 古くは順編成ファイルのみであった。一応すべて の種類のファイルが処理できるが、多量のデータを一括処理するために、 経済性、データの保全等の観点から依然順編成ファイルが多く使用される。
  - ② 実時間処理の場合 短いアクセス時間が最重要な条件であるから、乱 アクセス・ファイルが使用される。磁気ディスクの進歩によって実時間 処理が実現したということが出来る。システムの規模、ファイルの規模、 期待される性能、他システムとの整合性、その他を考慮してファイルの 種類を決定する。
  - ③ 分散処理の場合 近年パーソナル・コンピュータの普及にともない、 その専用ファイルの普及が著しい。LAN等で結ばれたコンピュータ・ ネットワークの場合、ファイル・サーバなどが研究されている。
- (4) オペレーティング・システムとファイルの関係 次の事項を説明する。OSの仕様およびジョブ制御言語(JCL)の引用 が望ましい。
  - ① ファイルの取扱いには複雑なプログラムによる処理が必要である。そのほとんどはOSの提供するサービスに依存する。VSAMが仮想記憶OSに固有のファイルであること、パーソナル・コンピュータのファイルなど著しい例である。OSのデータ管理機能がOS選定の際の要件であるのはこのためである。

### 1.3 ファイルの種類(60分)

アプリケーションの立場からファイルの種類の概要を説明する。

- (1) システム・ファイルとユーザ・ファイル
  - ① システム・ファイル コンピュータ・システム運用のためのファイル、 保守用ログ・ファイル、料金計算用ファイル、各種サービス用ワーク・ ファイルなど。
  - ② ユーザ・ファイル 特定の業務のためにユーザが用意するファイル。
- (2) 基本ファイルと変動ファイル
  - 基本ファイル 業務処理システムの基本となるファイル、マスタ・ファイル、各種テーブルなど。
  - ② 変動ファイル トランザクション・ファイル、ジャーナル・ファイル、 作業ファイルなど。
- (3) 永久ファイルと一時ファイル
  - ① 永久ファイル マスタ・ファイル、保存用ファイルなど。
  - ② 一時ファイル トランザクション・ファイル、作業用ファイル、データ用一時ファイルなど。

## 育成方法および指導上の留意点

- 1. この章の内容は初級技術者といえども徹底して知らしむべきである。 演習はないが、小テストを繰返し行う、あるいはレポートを課すなどの方 法で質問には確実に答えられるようにしなければならない。
- 2. 1.1(2) ファイル記憶媒体。深く立入る必要はない。多様な媒体の存在と そのイメージをあたえるだけで充分である。
- 3. 1.1(3) レコード形式。とかく曖昧になり勝ちな所であるから学習者に図を書かせるなどして明確なイメージを一度把握させる。
- 4. 1.2 情報処理システムとファイル。各種の編成法を知らしめることの他、情報処理システムの中のそれぞれの編成法の必要性を理解させるため、他の教科、言語、プログラムの中で取り上げた例を報告させるなどして理解を深めさせる。
- 5. 1.2(3)ファイルの種類。このような分類や名称にさほど重要な一般性があるとは思わない。名称もかなりローカルなものが多く使われていると思うの

でここに書いたものにこだわる必要はない。要は学習者に多様なファイルの 用途を認識させることである。

## 用 語

項目(フィールド),レコード,ファイル,固定長レコード,可変長レコード,スパンド・レコード,不定長レコード,ブロック,ブロック化,非ブロック化,ブロック化係数,論理レコード,物理レコード,順呼出し,乱呼出し,順編成ファイル,直編成ファイル,索引順編成ファイル,VSAMファイル,システム・ファイル,ユーザ・ファイル,基本ファイル,変動ファイル,永久ファイル,一時ファイル

## 第2章 ファイルと記憶媒体

## 学習目標

学習者が次のことができるのが目標である。

- 1. 磁気テープ装置、磁気ディスク装置およびフレキシブル・ディスク装置の 構造と機能を説明する。
- 2. それぞれの媒体に記憶されるファイルの編成法とレコード形式を説明する。
- 3. それぞれの媒体の記憶容量、他の条件を無視した場合の処理時間の計算をする。

## 内 容

### 2.1 磁気テープ装置(60分)

- (1) 構造と機能 図表を用いて次の事項を説明する。
  - ① テープリール、読み書きヘッド、テープ送りサーボ機構その他からなる磁気テープ装置の構造と機能の概要
  - ② 磁気テープの物理的形状およびBOT、EOTマーカの機能。書込み 許可リングの機能。IBG(ブロック間隔)発生の理由。
- (2) 磁気テープ上の記録方法 図表を用いて 9 トラック磁気テープ上の情報記録方法を説明する。
  - ① 磁気記録の物理的詳細には立入らない。ブロックの形式。奇偶検査ビットの(奇数)の意味。
  - ② ブロックに付加される奇遇検査用冗長列(遇数)の意味。巡回検査用 冗長列の概要。読取り検査の概要。
  - ③ MTマークがハードウェアによって記録される1桁の特殊記号である こと。
- (3) 磁気テープ装置の性能
  - (1) 磁気テープの記録密度、テープ速度およびデータ転送速度の意味とそ

れらの間の関係を数式で説明する。起動停止時間を加えた場合の1ブロックの転送所要時間を示す。

記録密度(列/mm)×テープ速度(mm/秒)

=データ転送速度(桁またはバイト/秒)

ブロック長さ (バイト) データ転送速度 (バイト/秒)

+ 起動停止時間(秒)

=1ブロック転送 所要時間(秒)

② 実際に使用されている磁気テープ装置を1例以上あげ、性能表を説明する。

### 2.2 磁気テープ・ファイルの形式(60分)

図表を用いてファイルの論理的形式を説明する。

- (1) ブロックの形式
  - ① ブロック化された固定長レコード、可変長レコード、スパンド・レコードのそれぞれのブロックの形式。ブロック化係数の意味。
  - ② ブロック化されない場合の形式。不定長レコードの形式。
- (2) ラベルとMTマーク
  - ① ボリューム・ラベル、見出しラベル、後書きラベルの形式の概要
  - ② 単一ファイル・ボリュームについて、ファイル全体のラベル、MTマーク、レコードの配置。
- (3) 複数ファイルと複数ボリューム 複数ファイル、複数ボリュームのそれぞれそについてファイル形式とラ ベル形式の概要。

## 2.3 磁気テープの記憶容量と処理時間(60分)

例題によって記憶容量と処理時の計算の演習を行う。原則として固定長レコードを取扱う。ラベル、MTマーク等は無視する。

またCPU、磁気テープ装置が処理に必要とする時間は無視する。

### (1) 記憶容量の計算

記号の意味

L:磁気テープの長さ mm (=25.4×12×長さ(フィート)

d:記録密度 バイト(列)/m

M:ブロック化係数

n: レコード長さ バイト

G:ブロック間隔(IBG) mm

N:レコード件数

1ブロック(M件のレコード)を記憶するために必要なテープの長さを 計算する式

$$\frac{Mn}{d} + G$$
 (mm)

磁気テープ1巻に記憶できるレコード件数を計算する式

$$\frac{ML}{\frac{Mn}{d} + G}$$
 (小数点以下切捨て) (件)

またN件のレコードを記憶するために必要てテープの長さを計算する式

$$rac{N}{M}$$
 ( $rac{Mn}{d}$  +G)( $rac{M}{N}$  は小数点以下切上げ)(mm)

## (2) 処理時間の計算

記号の意味

s:データ転送速度 バイト/秒

v:テープ速度 mm/秒

d:記録密度 バイト (列) /mm

M:ブロック化係数

n:レコード長さ バイト

ts:起動停止時間 秒

N:レコード総数

1ブロック(M件のレコード)の転送(読みまたは書き)に必要な時間 を計算する式

$$\frac{Mn}{s} + t_s$$
 または  $\frac{Mn}{dv} + t_s$  (秒)

N件のレコードの転送に必要な時間を計算する式

$$\frac{N}{M}$$
 ( $\frac{Mn}{s}$ +  $t_s$ ) ( $\frac{N}{M}$  は小数点以下切上げ) (秒)

### (3) 演習問題

演習問題の例を提示する。

例1.次のような磁気テープ・ファイルがある。これについて設問に答えよ。 ファイルの仕様

10,000件

## 磁気テープの仕様

9トラック磁気テープ

記録密度

6 3桁/mm

データ転送速度

320KB/秒

IBG

1 5 mm

起動停止時間

6ミリ秒

テープ長さ

約720m(2,400フィート)

設問1 ブロック化係数が5の時と10の時の必要なテープ長さ。

設問2 同じく読取り時間

# 2.4 磁気ディスク装置

(1) 構造と機能

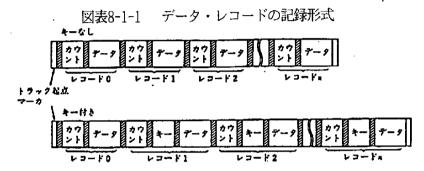
図表を用いて次の事項を説明する。

- ① スピンドル、磁気ディスク、読み書きヘッド、アクセス・アームその 他からなる磁気ディスク装置の構造と機能の概要。
- (2) 固定ディスクと取扱い可能ディスク 図表または実物を示して次の事項を説明する。
  - ① 固定ディスクとディスク・パックのちがい。固定ディスクは一般的に大容量であること。1スピンドルを1ボリュームとすること。

(3) トラック記録方法

図表を用いて次の事項を説明する。

- ① 1トラックは一定の記憶容量をもち、ビット直列に情報が記憶される。
- ② レコード (物理レコード) はキーなしまたはキー付きの形式のどちらかで書かれる。レコードO(RO)はOSがトラックの管理に使用するのでユーザはR1以降を利用する。



- ③ それぞれのブロックにはハードウェアの利用する制御情報が含まれている。カウント部は(CCHHR)シリンダ番号、トラック番号、レコード番号を含む。キー部はレコード検索用のキーを記憶する。データ部は本来の物理レコードを記憶する。
- (4) シリンダの考え方図表を用いて次の事項を説明する。
  - ① アクセス・アームの動作位置によりヘッドの数に等しいトラックから なるシリンダが形成される。シリンダの数は1000以上におよぶものがある。
  - ② シリンダ番号、トラック番号、レコード番号でアドレスを表現する。
- (5) 磁気ディスク装置の性能
  - ① 磁気ディスク装置の性能を評価する上で、記憶容量と平均アクセス時間が重要であることを説明する。

記憶容量=1トラック当り記憶容量×1シリンダ当りトラック数 ×シリンダ数

平均アクセス時間=平均位置決め時間+平均回転待ち時間 +データ転送時間 平均回転待ち時間≒1/2 ディスク1回転時間 データ転送時間=ブロック・サイズ (バイト) /データ転送速度

② 実際に使用されている磁気ディスク装置を1例以上あげ、性能表を説明する。

### 2.5 磁気ディスク・ファイルの形式(60分)

図表を用いて論理的形式を説明する。

- (1) ボリューム・ラベルとVTOC
  - ① ボリュームの先頭に置かれるボリューム・ラベルの形式と機能の概略。
  - ② VTOC (Volume Table Of Contents) の形式と機能の概略。いずれもOSによって生成、管理される。
- (2) イクステント
  - ① ファイルはユーザがあらかじめOSの助けを借りて磁気ディスク上に 用意した領域(イクステント)に作成される。イクステントはファイル 名と共にVTOCに登録されている。
- (3) ブロックの形式 磁気テープと同様に各型式のレコードのブロック(物理レコード)の形式の概略。

## 2.6 磁気ディスクの記憶容量と処理時間(60分)

例題によって記憶容量と処理時間の計算の演習を行う。原則として固定長レコードを取扱う。カウント部、制御情報等はギャップに含める。CPU、磁気ディスク装置が処理に必要とする時間は無視する。

(1) 記憶容量の計算

記号の意味

d:1トラック当り記憶容量 バイト

h:トラック数 C:シリンダ数

M:ブロック化係数

n:レコード長さ バイト

バイト

N:レコード件数

1 ボリュームの記憶容量を計算する式

dhC

1トラックに記憶出来るレコード数を計算する式

$$M = \frac{d}{Mn+G}$$
 (小数点以下切捨て)

N件のレコードを記憶するために必要なトラック数とシリンダ数を計算 する式

$$\frac{N}{M - \frac{d}{MN + C}}$$

(小数点以下切上げ)

$$\frac{N}{hM \frac{d}{MN+G}}$$

### (2) 処理時間の計算

記号の意味

S:データ転送速度

t。: 平均位置決め時間

tw: 平均回転待ち時間

M:ブロック化係数

n:レコード長さ

バイト

G:ギャップ長さ

ラバイト

平均アクセス時間を計算する式

$$t_p+t_w+\frac{Mn+G}{S}$$
 (小数点以下切捨て)

N件のレコードを記憶するために必要なトラックとシリンダ数を計算す る式

## (2) 処理時間の計算

## 記号の意味

S:データ転送速度 バイト/秒

t。: 平均位置決め時間 tw: 平均回転待ち時間

M:ブロック化係数

n:レコード長さ

バイト

G:ギャップ長さ

バイト

平均アクセス時間を計算する式

$$t_p + t_w + \frac{Mn + G}{S} \tag{4}$$

明らかに同じシリンダ内の情報を引続いてアクセスする時はt。を無視出来る。また同じトラックの次のレコードをアクセスする時はtwも無視出来る。

# (3) 演習問題

演習問題の例を掲示する。

例1. 次のような磁気ディスク・ファイルがある。これについて設問に答 えよ。

ファイルの仕様

レコード長さ

600バイト

レコード件数

10,000件

磁気ディスクの仕様

トラック/シリンダ

1 4

記憶容量/トラック

28KB

ギャップ長さ

159バイト

平均位置決め時間

18ミリ秒

平均回転待ち時間

8ミリ秒

設問1 ブロック化件数が1の時と10の時の必要なシリンダ数

設問2 同じ条件で全体を順処理する場合の総アクセス時間。位置決め 時間と回転待ち時間は無視出来ない。

## 2.7 フレキシブル・ディスク装置(60分)

(1) 構 造

図表を使用して次の事項を説明する。

- ① 200mm (8インチ), 130mm (5インチおよび90mm (3.5インチ) フレキシブル・ディスクの外観、構造および機能の概要。
- ② 駆動装置の説明はなくてよい。
- (2) フレキシブル・ディスクの種類 次の事項を説明する。
  - ① 上記の3種類においてそれぞれ次の種類がある。
    - a. 片面形、両面形
    - b. 単密度、倍密度
    - c. 通常トラック密度、倍トラック密度
  - ② このうちコンピュータに多く使用されるものの仕様をあげる。

#### 200mフレキシブル・ディスク

	トラック数 /面	セクタ数 /トラック	記 <b>憶</b> 容量 バイト/セクタ	回転数 /分
片面形	77	2 6	1 2 8	360
両面倍密度	77	2 6	256	360

## 130mmフレキシブル・ディスク

両面倍密度 倍トラック密度	. 77	2 6	256	3 6 0
------------------	------	-----	-----	-------

#### 90mmフレキシブル・ディスク

	両面倍密度	8 0	2 6	256	3 0 0
--	-------	-----	-----	-----	-------

(3) トラックとセクタ

図表を用いて次の事項を説明する。

- ① トラックは最外周を0として番号付けされる。両面形ではトラック番号についてする。またトラックはセクタにわかれ、読み書きはレコードの形式に関係なくセクタ単位で行なわれる。
- (4) 記憶容量の計算
  - ① 8 インチ、5 インチ両面倍密度の場合、ユーザの使用出来るトラックは 0、74、75、76番を除く73本であるから、記憶容量は次の通りである。 2×73×26×256 = 971,776 バイト
  - ② 3.5インチについてもほぼ同じである。

## 2.8 その他の媒体(30分)

- (1) 光ディスクの種類と特徴 図表を用いて次の事項を説明する。
  - ① 光ディスクはコンピュータ用ファイル媒体として現在次の種類が使用されている。この他90m書換え型があるがまだあまり多く使用されていない。

図表8-1-2 光ディスクの種類と特徴

	3 0 0 mu追記型	130mm追記型	130㎜書換え型
記憶容量の1例 MB/面	1, 310	300	3 2 2
平均アクセス時間 の1例 m秒	2 5 0	110	読出し 75 書込み100
転送速度の1例 MB/秒	1, 5	1, 5	1, 5

追記型(WORM: Write Once Read Many とも言う。)

1度書き込むと以降は書き直し不可、読み出しのみ可能な型。

## 書換え型

繰り返して書き換え可能な型。

- ②光ディスクには次の特徴がある。
  - a. 大容量である。 磁気ディスクの10倍以上の面記憶密度がある。
  - b. 可換型である。

カートリッジに収納したディスクを容易に交換できる。自動選択交換型ライブラリ装置を使用した大規模なファイル・システムを構成できる。

- c. ランダム・アクセスである。ただし、アクセス時間は磁気ディスクよりかなり大きい。
- d. 媒体の寿命が長く安定である。 10年以上の安定した性能が期待できる。熱、光、ほこりなどの影響を受けにくく、環境に耐える力が大きい。
  - e. 経済的である。

磁気ディスクに比較して単位情報量あたりのコストは低いといわれる。

- (2) 光ディスクの用途次の事項を説明する。
  - ①イメージ情報の媒体

文書、図面、写真などをスキャナで分解し、ディジタル化した情報を記憶する。光ディスクの特性を生かし大量の情報の保存と迅速な検索に適するシステムが多方面で利用されている。医療、特許事務分野などへの導入が進んでいる。

②コード情報への応用

非常に大きい容量を持つファイルを造り易いので磁気テープ、磁気ディスクの代替えとしての利用が進んできた。

追記型は大量の情報の保存に適するので、この目的には磁気テープに代わるうとしている。

また、書換え型は即時性のある大容量ファイルとして磁気ディスクに代 わって利用されるケースが出てきた。特にアクセス時間が大きくても差 支えない場合に適する。

(3) 印刷用紙(フォーム)

次の事項を説明する。必要あれば図表を利用する。

①コンピュータに接続される印刷機は殆どすべて文字間隔 1 / 1 0 インチ (2.54mm)、行間隔 1 / 6 インチを基準に設計されているので、印 刷用紙もこれに合わせて規格化された連続用紙が使用される。

(JIS X6195 参照)

②用紙は55Kg以上の上質紙が使われるが、印刷機の種類に応じて、感 熱紙、ノンインパクト専用紙などがある。また多層の複写用紙も使われ る。

無地または罫線入りの15インチ×11インチ(381mm×279mm)の連続用紙はストックホームと呼ばれて最も多く使用される。

### 育成方法および指導上の留意点

- 1. ハードウェアの教科との重複を気にすることなく、ファイルの理解に必要 な範囲で説明する。
- 2. 磁気テープの重要性は以前ほどではなくなったが、教育上の重要性は変らないのでこの程度の内容は理解させなければならない。
- 3. 磁気テープ上の物理的ブロックの形式にはJISが良い参考書となる。
- 4. レコードの形式、ファイル形式は適当なファイル、レコードを教材として 学習者に作図させるとよい。
- 5. 2.4磁気ディスク。トラック・フォーマットは実在の機種について物理的フォーマットを図で示し論理的フォーマットとの差異を認識させる方がよい。
- 6. 記憶容量、特に処理時間の計算は他の制約をすべて無視したもので、実際 の処理時間とは大差があることは言うまでもない。むしろ機能を性格に理解 させるための演習である。
- 7. 演習問題は過去の各種試験の出題の中から選ぶとよい。

## 用 語

1BG, MTマーク, 記録密度, テープ速度, データ転送速度, ラベル, 単一ボリューム, 複数ボリューム, 単一ファイル, 複数ファイル, トラック, シリンダ, カウント部, キー部, データ部, 平均アクセス時間, 平均位置決め時間, 平均回転待ち時間, ボリューム・ラベル, VTOC, イクステント, 片面形, 両面形, 単密度, 倍密度, トラック密度, セクタ

## 第3章 順編成ファイル

## 学習目標

学習者が次のことができるのが目標である。

- 1. 磁気テープ、磁気ディスク順編成ファイルの作成方法を説明する。
- 2. 磁気テープ、磁気ディスク順編成ファイルのレコードの更新、追加、削除の単純な例題の流れ図を書く。
- 3. 併合、突き合せ等の流れ図を説明する。

## 内 容

3 1 磁気テープ順編成ファイルの作成と処理(120分)

簡略化したプログラム流れ図および図表を使用して説明する。

- ファイルの作成 次の事項を説明する。
  - ① あらかじめ昇順または降順に配列されたレコードを用意して磁気テープに書き込むか、レコードを磁気テープにすべて書き込んだ後、分類 (ソート)プログラムを使用して昇順または降順に配列しなおす。
  - ② 磁気テープへの書き込みにはユティリティ・プログラムを使用することが多い。
  - ③ レコード型式にはすべて許されるが、レコードの最大長さ等の制限は 利用するOS(オペレーティング・システム)の仕様にしたがわねばな らない。
- (2) レコードの更新、追加、削除処理 次の事項を説明し、演習を行う。
  - ① アクセスはファイルの先頭からの順アクセスに限られ、しかも磁気テープでは同じテープ上でレコードの書き直しは出来ないので、すべての 処理は読んだレコードを別の出力ファイルに書きながら行う。
  - ② 演習問題によって流れ図作成を演習する。

#### 3.2 磁気ディスク順編成ファイルの作成と処理(120分)

磁気テープと同様に流れ図等を使用して説明する。

- ファイルの作成 次の事項を説明する。
  - ① 磁気ディスク順編成ファイルを作成するには、あらかじめファイルに 必要な大きさを持つイクステントを用意しVTOCに登録しておかなけ ればならない。この操作は利用するOSの仕様にしたがって行う。
  - ② ファイル作成の手順は磁気テープ・ファイルと同じである。
- (2) レコードの更新、追加、削除処理 磁気テープと比較しながら次の事項を説明し、演習を行う。
  - ① 磁気ディスクではレコードの書き直しが可能なので、レコード長さの 変更をともなう更新とファイルの途中へのレコードの追加の場合を除き 新しいファイルの作成は不要である。
  - ② 削除はレコードに削除文字を記入して行うことが出来る。削除レコードが多くなった時は効率の低下を避けるため、ファイルを再編成する。
  - ③ 演習問題によって流れ図作成を演習する。

#### 3.3 順編成ファイルの代表的な応用例(60分)

順編成ファイルの更新、追加、削除等の処理は実際には次のような処理において行なわれることを流れ図を引用して説明する。

- (1) ファイルの併合
  - 同じレコード形式、同じキーの昇(降)順を持つ2つのファイルを1つのファイルに併合する。
- (2) 基本ファイルの突き合せ 基本ファイルと変動ファイルを突き合せて新しいファイルを作成する。
- (3) 基本ファイルの更新 基本ファイルと変動ファイルを突き合せて基本ファイルのレコードを更 新する。基本ファイルのレコード数に増減はない。
- (4) 基本ファイルのメンテナンス 基本ファイルと変動ファイルを突き合せて基本ファイルのレコードの追

### 加、削除を行う。

## 育成方法および指導上の留意点

- 1. 順編成ファイルの重要性は少しも変っていないことを、まず念頭に置かねばならない。
- 2. 3.1(2)の演習で磁気テープの場合、すべての処理が出力ファイルを必要とすることを理解させる目的であるから変動レコードは1件でよい。
- 3. 3.2(2) の演習で磁気ディスクの場合、出力ファイルなしでできることとできないこと、また、磁気テープとの違いを明らかにするためであるから変動レコードは1件でよい。
- 4. 3.3 代表的な応用例。これらの処理はアルゴリズムの分野であり、深く立 入れば容易ならぬことになるから、順編成ファイルの処理はこのように行な われることを理解させるだけでよい。

# 用 語

昇順, 降順, 更新, 追加, 削除, 削除文字, 再編成, 併合, 突き合せ, メンテナンス

## 第4章 直編成ファイル

#### 学習目標

学習者が次のことができるのが目標である。

- 1. 直編成ファイルの構造を説明する。
- 2. 実アドレス、相対アドレスのちがいを説明する。
- 3. シノニムの発生の理由と、発生時の対策を説明する。
- 4. レコードの更新、追加、削除の流れ図を書く。

## 内 容

#### 4.1 直編成ファイルの構造(90分)

- 直編成ファイル 次の事項を説明する。
  - ① 磁気ディスクのような直接アクセス記憶装置にのみ適用できる編成法。特定のレコードを直接アクセスするので、順アクセス、乱アクセスの両方が出来る。
  - ② 物理レコードだけが対象。論理レコードはユーザのプログラムで取扱わねばならない。
  - ③ COBOL文法の相対編成ファイルは相対アドレスを用いた直編成ファイルで、順アクセスも可能である。
  - ④ 領域利用効率が低く、小規模なファイルに適する。
- (2) 実アドレスと相対アドレス 図表を用いて次の事項を説明する。
  - ① 実アドレス レコードが記憶されているアドレスを表わすのにレコードのカウント部(CCHHR:シリンダ番号、トラック番号、レコード番号)を使用する。
  - ② 相対アドレス ファイルの先頭レコードを0とする相対レコード番号 でアドレスを表わす。相対トラック番号、レコード番号を組合わせる方

法もある。

- (3) 直接アドレス指定法と間接アドレス指定法 次の事項を説明する。適宜図表を利用する。
  - ① 直接アドレス指定法 レコードの中のキー項目で直接アドレスを指定 する。キー項目の設計に制約あり、効率的でない。
  - ② 間接アドレス指定法 ファイル作成時にキー項目の値に所定の演算を 行ってレコードのアドレスを決め、読む時も同じ計算を行ってアドレス を求めてアクセスする。レコードがイクステント内に一様に分布するよ うな計算法が望ましく、除算法、折りたたみ法、基数変換法、抽出法な どが行なわれる。ランダマイジングともいう。
  - ③ 上記のキー・アドレス変換法のそれぞれの解説。
- (4) シノニム・レコードの処理 次の事項を説明する。
  - ① アドレス変換の結果、同じアドレスに2つ以上のレコードが割り当てられて発生するシノニム・レコードの対策。シーケンシャル法、チェーン法等が行なわれる。

#### 4.2 直編成ファイルの作成と処理(120分)

- (1) ファイルの作成流れ図等を利用し次の事項を説明する。
  - ① あらかじめ最大レコード数を予想しパッキング・ファクタが80%程度になるようにイクステントを用意する。シノニム発生が少いアドレス変換法をえらぶ。ダミー・レコードをあらかじめ全領域に書いておく。
- (2) レコードの更新、追加、削除 次の事項を説明し、演習を行う。
  - ① 更新、追加の処理。シノニム発生時の処理。
  - ② 削除はダミー・レコードへの置換で行う。
  - ③ 演習問題によって流れ図作成を演習する。演習問題例、相対アドレス、 除算法による間接アドレス指定法、シノニムの対策はシーケンシャル法 によって編成したファイルがある。これに対し1件のレコードの更新、

追加、削除を行う場合の流れ図。ただし除算法のサブルーチンは別に用 意されている。

## 育成方法および指導上の留意点

- 1. 比較的問題の少い章であると思う。次の索引順編成の前段として、乱アクセスを充分に理解させるのに利用する。
- 2. 練習問題の流れ図そのものは単純なものであるが、夫々の処理の内容を正確に把握するように指導する。

# 用 意

実アドレス,相対アドレス,直接アドレス指定法,間接アドレス指定法,キー・アドレス変換,ランダマイジング,シノニム・レコード,パッキング・ファクタ

## 第5章 索引順編成ファイル

## 教育目標

学習者が次のことができるのが目標である。

- 1. 索引順編成ファイルの構造を説明し、基本データ域、索引域、あふれ域の内容と機能を説明する。
- 2. キー順アクセス、乱アクセス時のファイル上の検索手順を図表を使って説明する。
- 3. 初級技術者が独自に索引順編成ファイルを企画、設計することは想定しない。
- 4. レコード追加、削除時のファイル上の変化について要点を説明する。

## 内容

- 5.1 索引順編成ファイルの構造(180分)
  - (1) 概 要

次の事項を説明する。

- ① 索引順編成ファイルは直接アクセス記憶装置にのみ適用できる編成法 である。レコードをキーによって順アクセスすることも、特定レコード を直接アクセスすることも出来る。
- ② イクステントは索引域、基本データ域、あふれ域にわかれる。
- ③ 固定長、可変長レコードに適用できる。記憶領域の利用効率も高い。
- (2) 基本データ域図表を使って次の事項を説明する。
  - ① レコード本体を収容する場所で、ここにキー順にレコードが収容される。磁気ディスク上はキー付きレコードが用いられる。
- (3) 索引域 図表および例をあげて次の事項を説明する。
  - ① トラック索引は各シリンダの0トラックにおかれ、各トラックに収容

されたレコードのキーの最大値がすべて記憶されている。シリンダ索引には各シリンダに収容されたレコードのキーの最大値が記憶される。ファイルが大きい時はシリンダ索引のトラックを検索するためのマスタ索引が自動的に作られる。

② 順アクセスの時はファイルの先頭から基本データ域のレコードをキー順に読む。特定のキーの値を持つレコードを直接アクセスする時とマスタ索引、シリンダ索引、トラック索引の順に検索し、該当するトラックの中から目的のレコードを探す。

#### (4) あふれ域

図表および例をあげて次の事項を説明する。

- ① レコードを追加すると該当するトラックからあふれ出たレコードはそのシリンダのあふれ域に収容される。これにともないトラック索引の書直しが行われる。
- ② シリンダあふれ域からさらにあふれたレコードは独立あふれ域に収容 される。
- ② レコードをアクセスする時はあふれ域にあるレコードもチェーン法でアクセスされる。
- ④ あふれ域のレコード数が増加すると著しくアクセス時間が長くなるので、ファイルを再編成して基本データ域に収容しなければならない。

## 5.2 索引順編成ファイルの作成と処理(60分)

指導上の注意、使用中のOSによる具体例を示すのが望ましい。

- ファイルの作成 次の事項を説明する。
  - ① ファイル設計にもとづいてあらかじめ、基本データ域、索引域、あふれ域を用意する。
  - ② キー昇順に配列したレコードを用意し、OSの提供するサービス・プログラムを利用して作成する。索引に自動的に作成される。
- (2) 順処理によるレコード変更、追加、削除処理
  - ① ファイルの先頭レコードからすべてのレコードをアクセスして行う場

合とファイルの途中から行う乱順処理とがある。

- ② 削除はレコードに削除文字を記入して行う。その領域は追加の時に再利用される。
- (3) 乱処理によるレコード変更、追加、削除処理 流れ図等を引用して順処理と同様の説明を行う。

## 育成方法および指導上の留意点

- 1. 学習者はかなりややこしい印象を受けるから、それを緩和する教育上の工夫が必要であろう。テキストを参考に説明図を作成させる。小テストを繰返して知識の定着を図る。VTRの利用等である。
- 2. もし実際にファイルを取扱える環境にあれば、プログラム演習の中に索引 順編成ファイルを取入れるのが望ましい。

# 用語

基本データ域。索引域。トラック索引、シリンダ索引、マスタ索引、あふれ域。シリンダあふれ域。独立あふれ域。乱順アクセス

## 第6章 区分編成ファイル

### 学習目標

学習者が次のことができるのが目標である。

- 1. 区分編成ファイルの構造と特徴について説明する。
- 2. 区分編成ファイルの作成、保守を直接行うことは想定しない。

## 内 容

## 6.1 区分編成ファイルの構造(30分)

(1) 概 要

図表を用いて次の事項を説明する。

- ① 直接アクセス記憶装置にのみ適用できる編成法である。
- ② 登録簿(ディレクトリ)とメンバーに分かれ、メンバーはそれぞれが 順編成ファイルである。登録簿によってメンバーを直接アクセスすることが出来る。
- ③ 区分編成ファイルはプログラムの登録などにしばしば用いられる。ユーザ・ファイルと使用されることは殆んどない。
- (2) 登録簿

図表を用いて次の事項を説明する。

- ① メンバーごとのメンバー記入項目がメンバー名の昇順に登録されている。
- ② メンバー記入項目の内容

## 6.2 区分編成ファイルの作成と処理(30分)

図表を用いて次の事項を説明する。

- ① 作成はユティリティ・プログラムを利用する。
- ② メンバーにレコードを追加してメンバーの領域が不足する時は新しい 領域が設定され登録簿が書き替えられる。

③ メンバーの削除は登録簿からの削除で行う。

# 育成方法および指導上の留意点

1. 大きい問題はない。

用語

区分編成ファイル、登録簿、メンバー、メンバー記入項目

## 第7章 VSAMファイル

#### 学習目標

学習者が次のことをできるのが目標である。

- 1. VSAMファイルの構造を図表を使って説明する。
- クラスタ、CI、CA、コンポーネント、RBA(相対バイト・アドレス)
   の用語の意味と内容を説明する。
- 3. ESDS、KSDS、RRDSをそれぞれ順編成ファイル、索引順編成、 直編成ファイルと比較対応させて説明する。
- 4. KSDSについては索引順編成と同様に、初級技術者が独自に企画、設計 することは想定しない。

### 内 容

#### 7.1 VSAMファイルの概要(180分)

- (1) VSAMの概要と特徴 次の事項を説明する。
  - ① VSAMは仮想記憶OSに固有の編成法で効率の良いデータ管理ができる。直接アクセス記憶装置にだけ適用できる。
  - ② VSAMではファイル、イクステントの言葉の代りにデータセット、 データ・スペースを使う。
  - ③ VSAMには3種の編成法がある。 エントリ順データセット (ESDS)

キー順データセット (KSDS)

相対レコード・データセット(RRDS)

- ④ VSAMユティリティ・プログラムを広く利用出来る。
- (2) VSAMデータセットの構造図表を使用して次の事項を説明する。
  - ① クラスタ 1つのデータセットのためにデータ・スペースの中に確保

された領域。

- ② CI (制御インタバル) 入出力の論理的単位で一定の大きさを持つ。 レコードは固定長、可変長の区別なくCIの中に左側から詰められる。 CIはさらに空きスペースおよび制御情報を含む。
- ③ CA(制御エリア) 連続した複数個のCIからなる。CAはまた空きスペースのみからなるCIを含むこともある。
- ④ コンポーネント(構成要素) 連続した複数個のCAからなる。ES DS、RRDSにおいてはクラスタはデータ・コンポーネントのみであるが、KSDSのクラスタはデータ・コンポーネント、インデクス・コンポーネントの2つのコンポーネントよりなる。
- ⑤ RBA(相対バイト・アドレス) クラスタの先頭を 0 とする相対バイト・アドレス。 VSAMはこれによって全体を論理的に連続した領域として扱う。

### 7.2 VSAMデータセットの種類(60分)

- 入力順データセット(ESDS)
   図表を用いて次の事項を説明する。
  - ① ESDSは磁気ディスク順編成ファイルに相似の編成法である。あらかじめキー順に配列したレコードを順に読込んで作成する。データ・コンポーネント内部に空きスペースは作られない。
  - ② RBAで直接アクセスすることが出来る。
- (2) キー順データセット (KSDS) 図表を用いて次の事項を説明する。
  - ① KSDSは索引順編成ファイルに相似の編成法である。データ・コンポーネントは基本データ域に相当し、作成時に内部に空きスペースを作っておくことが出来る。
  - ② インデクス・コンポーネントは索引域に相当する。最上位のインデクス・レコードを根とする木構造になっており、最下位のインデクス・レコード群を順序セット、上位を索引セットという。最上位から順に索引し目的のキーを持つレコードを直接アクセスする。

- ③ 追加レコードはデータ・コンポーネントの空きスペースを利用して収容し、インデクスも更新される。空きスペースが不足すれば空きCI、空きCAを利用しCI分割、CA分割が行なわれる。
- (3) 相対レコード・データセット (RRDS) 図表を用いて次の事項を説明する。
  - ① RRDSは直編成ファイルに相似の編成法である。レコードは固定長と見なされデータ・コンポーネントはレコード長と等しいスロットに分割され相対レコード番号(RRN)がつけられる。
  - ② RRNで直接アクセスすることが出来るが目的レコードのキーとRR Nの関係はシノニムも含みユーザが管理しなければならない。

### 7.3 レコードのアクセスと処理(60分)

流れ図等を引用して次の事項を説明する。

- (1) ESDS
  - ① 順アクセス、逆順アクセスの他RBA(相対バイト・アドレス)による直接アクセスが出来る。ただしレコードのキーの値とRBAとの対応表を別に作らねばならない。
  - ② レコード長さの変更、途中への追加、削除は出来ない。
- (2) KSDS
  - ① 順アクセス、逆順アクセス、キーによる直接アクセス、乱順アクセス が出来る。対応表を用意すればRBAによる直接アクセスも出来る。
- (3) RRDS
  - ① 順アクセス、RRN(相対レコード番号)による直接アクセス、乱順 アクセスが出来る。

# 育成方法および指導上の留意点

- 1. VSAMは仮想記憶OSにふさわしく進歩したシステムではあるが、従来からある編成法と比較して格別なものではないことを考えておく必要がある。
- 2. もしVSAMを実際に使用出来る環境にあるならば、プログラム演習の中

のVSAMファイルを取入れるべきである。

3. 流れ図の利用等ユーザ・プログラムの側からVSAMを説明することは難 しいから、学習者から見ればかなり退屈なものになる恐れがある。したがっ て索引順編成の場合と同様な工夫が必要である。

# 用語

データセット, データ・スペース, ESDS, KSDS, RRDS, クラスタ, CI, CA, コンポーネント, RBA, 順序セット, 索引セット, CI分割, CA分割

# 第8章 パーソナル・コンピュータ用ファイルの操作

## 学習目標

最近パソコン (パーソナル・コンピュータ) の普及が著しく、実務のみなら ずコンピュータ学習の手段としても大きい比重を占めるようになって来た。

本章の学習目標は学習者が次のことができるようになることである。

- 1 パソコン用ファイルの概要を説明する。
- 2. 学習に必要な範囲で、ファイルを操作することが出来る。

## 内容

## 8.1 パソコン用ファイルの概要(60分)

(1) 特徴および構造

現在、最も広く利用されているパソコン用OSのファイルについて、ファイル操作の演習に必要な範囲で特徴および構造を説明する。必要に応じて図表を使用する。

- ① フレキシブル・ディスクまたは固定ディスクに適用するファイルである。
- ② 階層化されたディレクトリ(登録簿)の下に多数のファイルを登録し、 ファイルの検索と管理が容易に行えるようになっている。

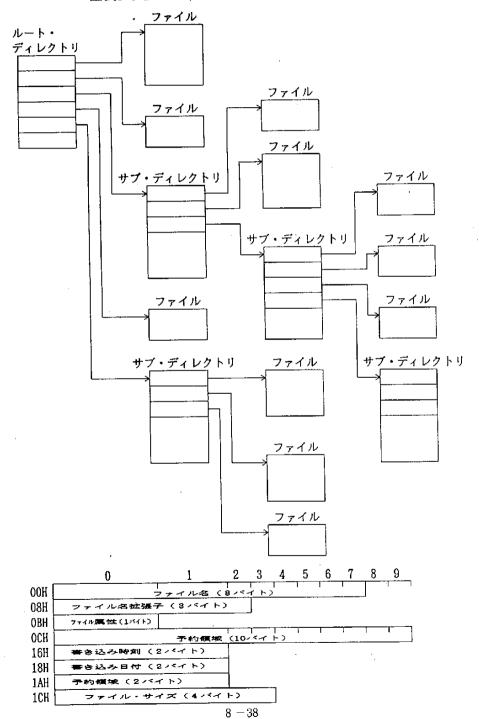
ルート・ディレクトリ:ボリュームに必ず1個ある主要なディレクトリ。

サブ・ディレクトリ:利用者がルート・ディレクトリの下に作ったディレクトリ。階層化することができる。

- ③ ファイル本体の領域はクラスタ(1024バイト)の集まりで、クラスタ 単位で増減できる。クラスタは連鎖になっている。
- ④ 1字以上の文字列をレコードとみなし、データとプログラムの区別がない。プログラムもファイルの一つとして取扱う。
- (5) 日本語を使用できる。ファイル名、データに漢字をふくむ日本語を使

用できる。

⑥ レコードは固定長である。順アクセスおよび相対レコード番号による 図表8-8-1 ディレクトリへの登録内容



乱アクセスができる。

#### (2) その他の事項

ファイル操作に関係ある事項を整理して説明する。

コマンド

パソコンのOSではプログラムの実行はプロンプト状態の時にコマンドを入力することによって始まる。コマンドのうち、実行プログラムが主記憶装置に常駐しているものが内部コマンド、プログラム・ファイルとして外部にあるものが外部コマンドである。外部コマンドにはOSに固有のものと利用者が作成して登録したものとがある。本章の演習に掲げるコマンドの大部分は内部コマンドであるが、一部に外部コマンドがある。

## ② カレント・ドライブ

外部コマンドの検索、ボリューム指定のないファイルの登録と検索を 行うボリュームである。プロンプト状態で表示される。

③ カレント・ディレクトリ

ファイルの登録と検索はその時のカレント・ディレクトリの下で行な われる。通常はルート・ディレクトリであるが、任意のディレクトリに 変更できる。

# (4) パスの指定法

目的のファイルを検索する経路がパスである。

絶対(完全)パス:ルート・ディレクトリから出発して目的のファイルにいたるパス。

# ¥SUB1¥SUB2¥ファイル名

\_\_\_\_\_\_ サブ・ディレクトリの名称。

相対パス:カレント・ディレクトリから出発して目的のファイルにい たるパス。

# SUB1¥ファイル名

\_\_\_\_ ← 下位のサブ・ディレクトリの名称。

⑤ ワイルドカード(かくし字)の利用特殊文字\*および?をファイル名の1部に使用してファイル名の一括

取扱いができる。

例 \*. LST 拡張子LSTを持つすべてのファイルを表わす。

#### 8.2 ファイル操作の演習(90分)

ファイル操作の演習テーマを以下に示す。これらはパソコン用OSを利用するための最低限のものである。この程度の操作は必ず経験させ、自在にやれるように熟練させるべきである。他の教科の演習の一部として行ってもよい。以下の記述では使用するコマンドの例をあげた。詳細はそれぞれのマニュアル等の資料によってほしい。

- (1) ファイルの作成と登録、更新、追加、削除 演習用のソース・プログラム、テスト用データ等適宜の資料をキーボー ドより入力して行う演習。
  - ① 行エディタによる方法使用コマンド EDLIN説明:1行のテキストが1件のレコードとして入力される。あまり実用されていないが、演習として行うべきである。
  - ② 画面エディタによる方法 OSに装備されている高機能の画面エディタを利用する。ファイルの 合併、分割などの機能を持つものもあるので、それらを含みできるだけ 詳細に演習する方がよい。
- (2) ファイル名表示使用コマンド DIR説明:指定されたディレクトリに登録されたファイルの名称を画面に表示する。
- (3) ファイル削除使用コマンド DEL
- (4) ファイル名の変更使用コマンド REN
- (5) ファイル内容の表示使用コマンド TYPE

説明:ファイルのレコードの内容を画面に表示する。画面エディタを使用する場合は勿論この機能は含まれている。

(6) ファイルのCOPY

使用コマンド COPY

説明:ファイルを別のボリュームにコピーする。

(7) ファイル印刷

使用コマンド PRINT

(8) サブ・ディレクトリの牛成と削除

使用コマンド MKDIRおよびRMDIR

説明;ファイルの検索、操作の便宜あるいは完全確保などの目的で用途 別、分野別などに分類して登録するためにサブ・ディレクトリを 用意する。

(9) カレント・ディレクトリの変更

使用コマンド CHDIR

説明:ファイル操作の便宜のため、カレント・ディレクトリを変更する。

(10) その他

上記の操作の他できれば演習する。

- a. パイプ機能によるコマンドの組合わせ。
- b. SORTコマンドによるレコードのソート。

# 育成方法および指導上の留意点

- 1. 現在パソコン用OSはMS-DOSとUNIXとで大部分を占める。ここではMS-DOSver.3.0 に準拠している。MS-DOSを選んだ理由は現在これが最も多数利用されていると思うからである。最近のMS-DOSはファイルに関してはUNIXに近いものになっていると言われている。したがってUNIXの環境下ならば、本章と同等以上の演習が可能である。
- 2. 本章の演習はファイルに関することだけであるが、他教科の進渉に合わせて高度な演習の一部分として取扱う方がより効果がある。

# 用 語

ルート・ディレクトリ, サブ・ディレクトリ, コマンド, 内部コマンド, 外 部コマンド

#### 第9章 データベースの概念

### 学習目標

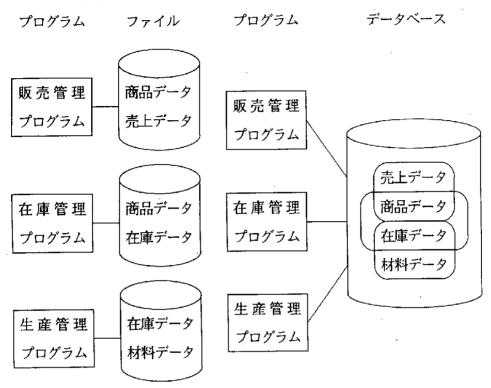
学習者が次のことができるのが目標である。

- 1. ファイルの基本的問題点を説明する。
- 2. CODASYL型データベースの構成、スキーマ、データ操作言語を説明 する。
- 3. 関係型データベースの構成、関係(表)を説明する。
- 4. 簡単な関係演算を実行する。
- 5. 3種のデータ構造を説明する。

## 内容

- 9.1 ファイルからデータベースへ(90分)
  - (1) データベースとは何か ファイルの基本的な問題点を整理し、データベースの概念を説明する。
    - ① ファイルの基本的な問題点
      - a. ファイルとプログラムの相互依存性。
      - b. 絶えず内容、仕様までも変更を迫られる。
      - c. 内容の重複、矛盾の危険が避けられない。
      - d. データの機密保持、標準化が困難。
    - ② すべての利用者の要求に応ずることができて、しかもプログラムの仕様に依存せず、独立して保守できるデータベースと管理体制を構築すれば上の諸問題はすべて解決する。

図表8-9-1 ファイル方式とデータベース方式の比較・

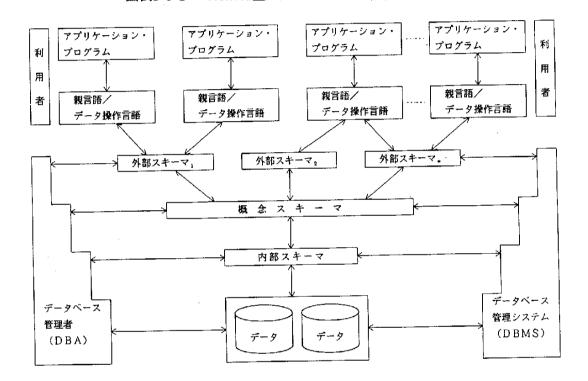


- (2) データベースの歴史 データベースの歴史を大きく次の各項にわけて説明する。
  - ① このようなデータベースの必要性は早くから認められ、1963年GEの IDSが開発された。次いで60年代の終りにIBMのIMSが発表された。
  - ② これらの開発をうけ、CODASYLにおいて研究が行なわれ、CODASYL DBTG 71が提案された。また言語についてはデータ記述言語DDL73、COBOLデータベース機能76、FORTRANデータベース機能76が提案された。これらに基づいて実用システムの開発が進んだ。
  - ③ 1980、81年にCoddによって関係型データベースが提案された。これは 従来の考えを全く捨てデータを集合論の立場からとらえたもので、デー

タベースに新しい展望を開くものとして盛んな議論をまき起し、その後 の開発に多きい影響をあたえた。

## 9.2 CODASYL型データベース(180分)

CODASYL型データベースについて、その構造や概念を説明する。 図表8-9-2 CODASYL型データベース・システムの基本構成



## (1) データ管理者

データベースはデータの集中管理と利用者へのサービスの提供を目的とするものである。そのため利用者とは独立したデータ管理者が必要である。 データ管理者は概念スキーマ以上を管理し、その職務はデータベースの設計、編成、保守、運用の監視、障害からの回復等である。

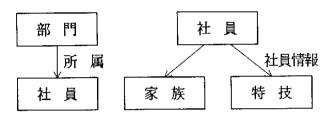
(2) 網型 (ネットワーク型) データ構造 CODASYL型ではレコードの他、レコード型の間の親子集合型 (S ET)として定義し、レコード型、親子集合型それぞれに実現値(オカレンス)を認める。

構造をデータベースの中に取組むことが出来る。

#### (3) バックマン線図

レコード型の間の親子関係を表現するための線図。それぞれ親(owner)、子(member)という。図中の所属、社員情報は親子集合型の名称である。

図表8-9-3 バックマン線図



## (4) スキーマの3層構造

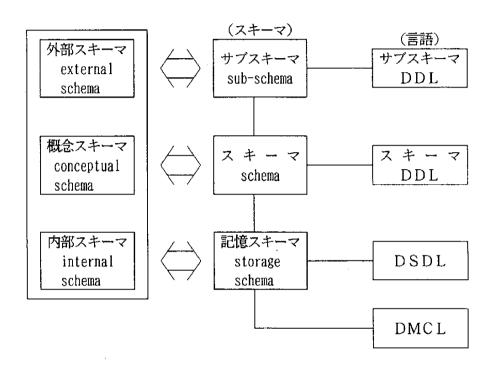
スキーマはデータベースの仕様を記述したものである。CODASYL 型では3層にわたって記述する。

- ① 副スキーマ(外部スキーマ)利用するプログラムの立場で個々に記述する。
- ② スキーマ(概念スキーマ) 現実世界をデータベースに写し取るためにデータ・モデルにしたがって記述する。
- ③ 記憶スキーマ(内部スキーマ)スキーマに記述された内容をデータベースにどのように実現するかを 記述する。
- ④ 同じものをANSIの報告では外部スキーマ、概念スキーマ、内部スキーマと呼ぶ。

図表8-9-4 ANSI/X3/SPARC と CODASYL DBTG ぜいとの3層構造比較

[ANSI/X3/SPARC]

(CODASYL DBTG]



#### (5) データ記述言語

スキーマを記述するための言語で、CODSYL案では、副スキーマ、 スキーマを記述するDDL、記憶スキーマを記述するDSDLとDMCL とからなる。

DDL : Data Description Language

DSDL: Data Storage Definition Language

DMCL: Device Media Control Language

# (6) データ操作言語

CODASYLは、COBOLデータベース機能76、FORTRANデータベース機能77を発表した。これらはいづれも既存の言語の機能を拡張してデータベースを操作出来るようにするものである。

#### 図表8-9-5 副スキーマの例

```
100100 TITLE DIVISION.
          ITLE DIVISION.
SS KYUUYO-KEISEN-BUBUN WITHIN REIDAI
ACCESS-CONTROL LOCK FOR INVOKING IS "SUKE".
100200
100300
200100 KAPPING DIVISION.
300100 STRUCTURE DIVISION.
300200
                   SECTION.
           REALM
300300
               KYUUYO-RYOOIKI CONTAINS ALL SUB-SCHEMA
406100
          SET SECTION.
400200
500100
          RECORD SECTION.
500200
                               SYAIN-KEY IS BANGOO.
PIC X(8).
                RECORD KEY
10 BANGOO
10 SIMEI
500300
500400
                                     X(8).
X(20).
500500
                    SYOZOKU PIC X(20)
                                PIC X(8).
PIC X(8).
PIC 9(10)
500700
          01 KYUUYO-REKI
                    K-BANGOO PIC
500800
                10
                    KYUUYO-GAKU
501000
                10
                     ACCESS-CONTROL
                                                FOR GET IS KYUUYO-GAKARI
FOR HODIFY IS KANRI-SYA.
                                        LOCK
501100
```

#### (7) データベース言語NDL

CODASYL案に基づきISOの勧告にしたがってJISが制定された。

JIS X 3004-1987

スキーマ定義言語、副スキーマ定義言語、モジュール定義言語、データ 操作言語よりなっている。

#### 9.3 関係型(リレーショナル)データベース(150分)

80、81年にわたってCoddが提案した関係型データベースを概説し、簡単な関係演算の演習を行う。

## (1) 関係(表)とは

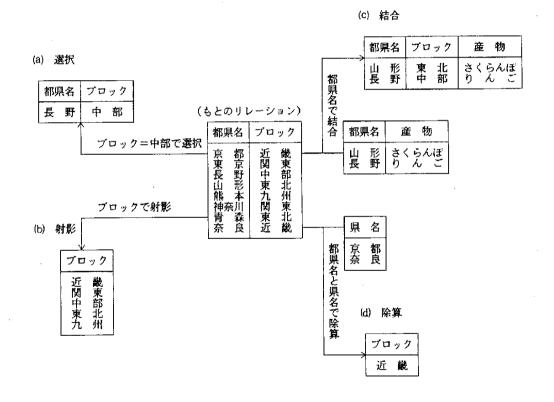
現実世界を各種データ集合の集まりと考え、それらの間の直積集合の部分集合を2次元の表の形にまとめたものが関係(または表)である。スキーマと呼ぶこともある。表全体がファイル相当するものとすれば表の1行はレコードになり、表の1列はフィールド(項目)になる。それぞれ組(Tuple)、属性(Atribute)または列(Column)という。表は第一正規形でなければならない。データベースは表の集まりで構成される。

#### (2) 関係演算および演習

表の間の演算(関係演算)によって必要な情報を得ることが出来る。各種の演算が提案されているが、基本になるのは選択(Selection)、射影 (Projection)、結合(Join)である。

- ① 選択 表の中から所興の条件に満足する組を選んで新しい表を作る。
- ② 射影 表の中から所定の属性だけを抜き出して新しい表を作る。
- ③ 結合 2つの表の組から条件に合致した組同志を結合して新しい表を 作る。
- ④ 演習問題例

図表8-9-6 関係演算の例



## (3) データベース言語SQL

IBM社の仕様を基にISOの勧告にしたがってJISが制定された。 JIS X 3005-1987

関係型データベースのスキーマ定義言語(SQL-DDL)、モジュール言語およびデータ操作言語(SQL-DML)よりなる。

図表8-9-7 SQLの問い合わせ例①

(学籍) リレーション

SELECT 学籍番号,生徒氏名 FROM 学籍 WHERE 学部=理工

学籍 番号	生徒 氏名		入学 年度	学	部
1	鈴	木	60	経	済
2	田	中	61	理	エ
3	佐	藤	60	文	学

学部=理工

学籍	生徒
番号	氏名
2	田中

図表8-9-8 SQLの問い合わせ例②

#### (成績) リレーション

学籍 番号	科	目	成績	単位
1	英	語	Α	2
1	経済	¥Ι	В	4
1	文	学	A C	2
1	歴	史	С	2
1	経	筝Ⅱ	Α	4
2	物	理	Α	4
2	数	学	Α	4
2	独	語	В	2
3	文	学	Α	4
3	歴	史	Α	4
3	仏	語	В	2
3.	経	済	В	2

SELECT 生徒氏名,学部 FROM 学籍 WHERE 学籍番号 IN SELECT 学籍番号

FROM 成績

WHERE 科目=歴史 AND 成績=A

(学籍) リレーション

科目=歴史 AND 成績=A

 学籍

 番号

 3

 学籍番号=3

学籍 番号	生徒 氏名		入学 年度	学	部
1	鈴	木	60	経	済
2	田	中	61	理	工
3	佐	藤	60	文	学
			•		

 生徒
 学部

 氏名
 大名

# 9.4 データ構造

3種のデータ構造について解説する。

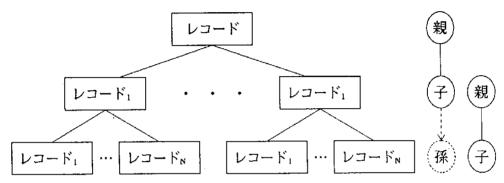
## (1) 概 要

データ構造は現実世界のデータをデータベースに構築する場合の論理構造であって、スキーマはこれにしたがって記述される。これにはおよそ3種類あって、しばしばデータベースの分類にも使われる。

#### (2) 階層型

IMSが代表的である。レコード(IMSではセグメント)型の間に一定の親子関係を定義する。実際のレコードは夫々のレコード型の実現値(オカレンス)であるが、親子関係にしたがって木構造に構成される。レコードのアクセスは根セグメントから出発して目的レコードに到達する。データベースはこのような木の集り、林(Forest)として構成される。親:子レコードの数の関係から1:N型とも呼ばれる。

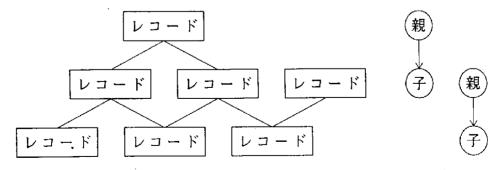
図表8-9-9 階層型モデルの基本構造



## ② 網型 (ネットワーク型)

階層型は現実世界の一部に適用できるに過ぎない。CODASYL型ではレコード型の他、レコード型の間の親子関係を親子集合型(SET)として定義し、レコード型、親子集合型それぞれ実現値(オカレンス)を認める。M: N型ということもある。

図表8-9-10 ネットワーク型モデルの基本構造



③ 関係型(リレーショナル型)

関係型データベースを構成するデータモデル。データベースは表(スキーマ)の集まりとして構成される。これらの間に関係演算を行って目的の情報を取り出す。

#### 育成方法および指導上の留意点

- 1. データベースの概念を学習者に理解させるためには、興味をもたせるよう にしむけるさまざまな工夫が必要である。教科書を輪読形式で学習者に説明 させるのも一つの方法であろう。
- 2. あるいは身近かな現実世界、たとえば家族親族、所属する職場などを取上 げて、図表、バックマン線図、文章など適宜の手段でレコード型、親子集合 型を定義して網型データベースを構成させるのも一つの方法であると思う。
- 3. 9.3(2)関係演算の演習は簡単な例で関係演算の意味を理解させれば充分である。

## 用 語

CODASYL, データ管理者, データ構造, 階層型, 網型, 関係型, レコード型, 親子集合型, 実現値, バックマン線図, 親, 子, スキーマ, 副スキーマ, 記憶スキーマ, データ記述言語, データ操作言語, NDL, 関係, 表, 関係演算, SQL

# 第10章 データベース管理システム(DBMS)

## 学習目標

学習者が次のことができるのが目標である。

- 1. データベースを利用する立場でDBMSの機能を説明する。
- 2. 実用DBMSの数例を列挙する。

# 内 容

## 10.1 DBMSの機能(60分)

次の事項を説明する。

(1) 概要

DBMS (Data Base Management System)はデータベースを効率よく構築、運用するための一群のソフトウェアである。主な機能は次の通りである。OSの下でDB/DCとして同様な機能が提供されることもある。

DB: Data Base

DC: Data Communication

- (2) データ定義機能 スキーマの記述を支援する機能。CODASYL型ではDDL、DSDL、DMCLなどの機能。
- (3) データ操作機能 データベースの構築、再構成を支援する機能、利用者 によるデータベースの操作を支援するDMLの機能。
- (4) データ制御機能 データの正しさを保証する保全機能、データの機密を保護する機能、障害回復機能その他。

# 10.2 データベース言語(30分)

次の事項を説明する。

(1) データ記述言語(DDL)

データ構造に基づいてスキーマを記述するための言語。CODASYL 型ではNDLの中でスキーマ定義言語、サブスキーマ定義言語として規定 されている。関係型ではSQLの中でスキーマ定義言語(SQL-DDL)として規定されている。

#### (2) データ操作言語

ユーザがデータベースを操作するための言語。CODASYL型のCOBOLデータベース機能、FORTRANデータベース機能。関係型ではSQL-DMLがこれに当る。

#### (3) 親言語方式

既存のプログラム言語を親言語として、これにデータベース機能を拡張する方式。CODASYLの方式はこれに当る。

## (4) 独立言語方式

既存の言語とは独立にデータベース操作専用の言語を用意する方式。ユーザ言語方式ともいう。DB/DCに例が見られる。

#### (5) コマンド方式

OSの下で動作するデータベース操作用コマンドを用意する方式。

# 10.3 各種の実用DBMS (30分)

メーカから提供されているDBMS例をあげる。

図表8-10-1 代表的な商用DBMSの概要

	h	DBMS名	国内販売	稼 働		提供してい	いる親言	115	
<i>T</i>	データモデル DBMS:		機関	開始年	COBOL	FORTRAN	PL/I	アセンブラ	エンドユーザ言語等
		LMS	日本18M	1974. 2	0		0	. 0	DL/1. IQF
階	層	전 ADM	日立	1973	0	0	0	0	VCELT
		System2000	(intel)						1
		DWS1100	日本ユニバッ	1971	0	0	0		QLP
			2						
÷.	トワーク	TOTAL	シンコム・シ	1969	0	0	0	0	MANTIS, T-ASK
~ >	717, 72	*	ステムズ						
		ALM/DB	富士通	1976末	0		0	0	Query (Interact 下のP.P.として配備)
		PDMII	日立	1980	0	0	0	0	ACEII
		ALM/RDB	富士通	1982, 12	0	0	: 0		AQL
		DB2	日本1834	1984後半	0	0	0	0	SQL, D82-Interactive, QMF
裘	リレー	SUPRA	シンコム・シ	1986, 2					SPECTRA
~	ッレー ショナル	*1	ステムズ						
	<i>2977</i> 9	SE OBE	日本IBM						
		WAPPER	日本ユニバッ						
			2				:		
型		ADABAS	ソフトウェア	1971.3	0	0	0	0	Super-NATURAL, NATURAL, ADASCRIPT+等
-	その	他	エージー		,				
i	(転置型等	MODEL 204	!   三井情報開発	1971	0	0	0	. 0	!   MODEL 204 ユーザ言語
		INQ	日花	1976, 9	0	Ō		· ŏ	EQL/EQU

## 育成方法および指導上の留意点

- 1. 実際にDBMSを利用出来る環境にある場合はその操作によって機能をた しかめることを併せて指導するのが望ましい。
- 2. 実用DBMSは出来るだけ広く資料を集めて学習者に提供する。
- 3. ここではDBMSの実例について内容を調査し、操作に習熟することは期待しない。

用語

DBMS, 親言語方式。独立言語方式。 コマンド方式

## 推薦教材

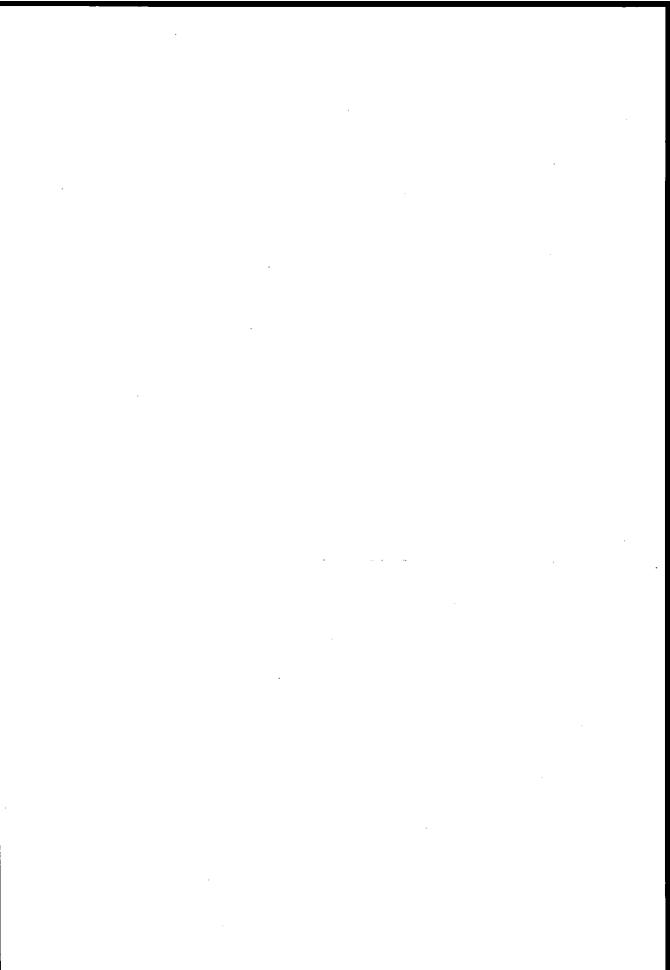
- 1. 「図解コンピュータシリーズ ファイル入門」 山谷 正巳 著. オーム社
- 2. 「ファイル」 (初級情報処理技術者テキストシリーズ Vol. 6) (脚)日本情報処理開発協会 中央情報教育研究所 編

## インストラクタ必読書

- 「データベースの基礎」
   植村 俊亮 著, オーム社
- 「図解 ファイル編成とデータベース」
   菊池 豊彦 著、コロナ社
- 「MS-DOS入門」
   北原 拓也 著, CQ出版
- 4. 「UNIX詳説―基礎編―」 井田・田中 共著. 丸善

# 参考文献

1. 「JIS 情報処理」,日本規格協会



#### 第9部 COBOL

# 教育目標

当部を受講した者は、この章を終了した時点で、プログラマとして次のような事柄ができるようになることを目標とする。

- 1. 与えられた問題をCOBOLで記述できる。
- 2. 与えられたCOBOLプログラムを読んで、その内容について説明できる。
- 3. これらに関連したテスト、デバッグ、修正、機能変更等ができる。
- 4. COBOLの基準及び水準は次の通り。

基準 電子計算機プログラム言語 COBOL JIS X 3002-1988 水準 必須機能単位の上位集合とし、選択機能単位については特に 規定しない。

必須機能単位	略 <del>号</del>	選択機能単位	略号
中核機能単位	中核	報告書作成機能単位	報告書
順ファイル機能単位	順	通信機能単位	通信
相対ファイル機能単位	相対	デバッグ機能単位	デバッグ
索引ファイル機能単位	索引	区分化機能単位	区分化
プログラム関連絡機能単位	連絡		
整列併合機能単位	整列		
原始文操作機能単位	原始文		
		La Maria National Maria	على المساور والمساور

5. 文法事項は確実に理解するが、暗記する必要はない。必要に応じて文法 書を読解できる。

# 内容の構成

		——————— 時 間	<b>数</b>
	内容	講義	演習
1. 1 1. 2 1. 3	COBOL概説 COBOL概説 機能単位と水準 COBOLプログラム記述の基本事項 COBOLプログラムの構成 演習問題	2	1
第2章 2.1 2.2 2.3 2.4 2.5	順ファイル・中核(I) 単純入出力プログラム例の読解 単純入出力プログラムの作成 単純入出力プログラムのメンテナンス 文法事項のまとめ 演習問題	4	2
第3章 3.1 3.2 3.3 3.4 3.5	順ファイル・中核(Ⅱ) 集計・作表プログラム例の読解 集計・作表プログラムの作成 集計・作表プログラムのメンテナンス 文法事項のまとめ 演習問題	2 1	3 9
4. 3	順ファイル・中核(II) マッチング・プログラム例の読解 マッチング・プログラムの作成 マッチング・プログラムのメンテナンス 文法事項のまとめ 演習問題	. 15	2 4
5. 1 5. 2	順ファイル・中核(IV) 表操作プログラム例の読解 表操作プログラムの作成 表操作プログラムのメンテナンス 文法事項のまとめ 演習問題	6	9

(続く)

	rh six		1 数
	内 容	講義	演習
6. 2 6. 3 6. 4	相対ファイル 相対ファイル・プログラム例の読解 相対ファイル・プログラムの作成 相対ファイル・プログラムのメンテナンス 文法事項のまとめ 演習問題		9
7. 1 7. 2 7. 3	索引ファイル 索引ファイル・プログラム例の読解 索引ファイル・プログラムの作成 索引ファイル・プログラムのメンテナンス 文法事項のまとめ 演習問題	6	9
第8章 8.1 8.2 8.3 8.4 8.5	整列併合 整列併合機能を用いたカグラム例の読解 整列併合機能を用いたカグラムの作成 整列併合機能を用いたカグラムのメンテナンス 文法事項のまとめ 演習問題	6	9
第9章 9.1 9.2 9.3	プログラム間連絡その他 プログラム間連絡機能を用いた プログラム例の読解 プログラム間連絡機能を用いた プログラムの作成 プログラムの作成 プログラムのパテナンス	12	18
	文法事項 演習問題 その他		
	合 計	7.8	120

## 第1章 COBOL概説

## 学習目標

この章を終了した時点で、学習者が次の事柄ができるようになることを目標とする。

- 1. COBOLの簡単な歴史、特徴について説明できる。
- 2. COBOLの規格の構成について簡単に説明できる。
- 3. 簡単なCOBOLプログラムをみて文字、語、文、句、節、部の区分ができる。
- 4. 正書法を説明できる。

# 内容

## 1.1 COBOL概説(20分)

(1) COBOL概説

事務計算用のコンピュータ言語であること、COBOL語とCOBOL コンパイラの説明およびコンパイルと実行について説明。 図を用いて、わかりやすく説明する。

- (2) COBOLの歴史
  - COBOLの生い立ちの前後、その後の発展を説明。CODASYLISO、ANS、JISの関係を説明。
- (3) COBOLの特徴

長所として、英語とほとんど同じであり、英文を書いたり読んだりする のと同じと思えばよいところがあることを説明する。数次にわたる機能改 定で便利な機能が追加されてきたこと。

短所は英語という自然言語に似せてあるため、自然言語の冗長性が持ち込まれていること。数次にわたる改定で機能が複雑になったこと。

#### 1.2 機能単位と水準(20分)

(1) 機能単位

JIS X3002-1988 の実物を見せる。見せるだけでよい。このうち11 個の機能単位の存在とその内容の概略説明をする。

(2) 水準

各機能単位の中の水準の存在を説明する。また、水準の内容について簡 単に説明する。

(3) JIS準拠

JIS準拠というのはどのようなことを意味するのか、また、JIS規格と実際の処理系で用いられる文法との関係を説明。授業で実際に使用するコンパイラについてここで説明する。

(4) コンパイラ

コンパイラと実行については冒頭で説明しているが、ここでは、実際に 実習で用いる機材を目前において説明する。

# 1.3 COBOLプログラム記述の基本事項(40分)

- (1) プログラム例 1 2 桁程度の数値の加算を行うプログラム例を提示する。
- (2) 文字 プログラム例 1 で用いた文字から COBOL で用いる文字の説明をする。
- (3) 語 プログラム例 1 を用いてCOBOLの語の説明をする。
- (4) 正書法

正書法を記した文書を配布する。JISをコピーしてもA4版2頁である。この文書とプログラム例1を対照させて説明する。暗記を強要しない。必要に応じ正書法の文書を参照するよう指導する。何度か繰り返しているうちに自然に覚えるように導く。

#### 1.4 COBOLプログラムの構成(40分)

(1) 部

プログラム例1を用いてCOBOLの部の説明をする。いわゆる4つの部を説明する。プログラム終わり見出し(END PROGURAM)についてもふれる。

- (2) 節 プログラム例1を用いてCOBOLの節の説明をする。
- (3) 句 プログラム例 1 を用いて C O B O L の句の説明をする。
- (4) 文 プログラム例 1 を用いて C O B O L の文の説明をする。

#### 1.5 演習問題(60分)

- (1) この章の演習問題はペーパーテスト形式が主体となる。ただし、コンピュータにふれる機会を設ける。
  - ① 1.1 および1.2 については記述式の問題を出題、参考書をみてよいということで答案作成をさせる。さらに、確実に理解をさせておかねばならないことについて、多岐選択型、穴埋め型の問題を出題し、参考書なしで答案作成をさせる。
  - ② 1.3および1.4についてはプログラム例1およびプログラム例1を変形したものを用いて文字、語、文、句、節、部の説明ができるかどうかをテストする。正書法についても、プログラム例1を様々に変形し、正しいかどうかの判断をさせる。
  - ③ プログラム例1をあらかじめ作成しておき、実習用コンピュータを用いて、コンパイルから実行までを実体験させる。この時点では、真の意味での理解はしていないと思われるが、実際にコンピュータに触れることで興味を引かせるようにする。

## 育成方法および指導上の留意点

- (1) COBOLの歴史、特徴は概略の説明にとどめる。
- (2) 規格と実際に使用するコンパイラとの違いについては、プログラム例1の中で具体的に指摘しながら教える。その際、できるだけ違いの理由を説明する必要がある。
- (3) プログラム例1をうまく工夫しなければならない。例を次に掲げる。入力 レコードは1件しかないものとする。(この段階ではあえてGO TO文を 用いる)

IDENTIFICATION DIVISION.

PROGRAM-ID. REII.

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT IN-FILE ASSIGN KISK-UNIT.
SELECT OT-FILE ASSIGN PRINT-UNIT.

DATA DIVISION.

FILE SECTION.

FD IN-FILE.

01 IN-RECORD.

02 ADATA PIC 9(2).

02 BDATA PIC 9(2).

FD OT-FILE.

O1 OT-RECORD.

02 CDATA PIC 9(3).

PROCEDURE DIVISION.

HAJIME.

OPEN INPUT IN-FILE.

OPEN OUTPUT OT-FILE.

KBISAN.

READ IN-FILE

AT END

GO TO OWARI

NOT AT END

COMPUTE

CDATA = ADATA + BDATA

WRITE OT-RECORD

END-READ.

GO TO KEISAN.

OWARI.

CLOSE IN-FILE OT-FILE.

STOP RUN.

END PROGRAM REII.

入力ファイルはカード・ファイルを使用せず、初めから磁気ディスクあるいはフレキシブル・ディスク・ファイルを用いる。あらかじめ入力ファイルをダンプアウトしたリストを用意して入力ファイルの内容を解説をする必要

がある。

- (4) 演習時における留意点は上述の演習問題の項を参照。
- (5) 用語は、この時点では覚えるのではなく、実例に即した理解に重点をおくこと。

# 用 語

A領域、B領域、COBOL、CODASYL、一意名、完結文、記述項、基本項目、句、区切り文字、原始プログラム、語、COBOL文字集合、作成者語、算術演算、算術式、式、実行時、実行用プログラム、集団項目、数字、数字項目、正書法、整数、節、節の見出し、節名、段落、段落の見出し、段落名、データ記述項、データ句、データ項目、データ名、手続き、手続き名、動詞、入出力節、入力ファイル、入力モード、廃要素、部、ファイル、ファイル管理、ファイル句、ファイル終了条件、ファイル節、ファイル名、部の見出し、プログラム終わり見出し、プログラム名、プログラム名記述項、文、分離符、文字、文字列、予約語、利用者語、レコード、レコード記述項、レコード名、レコード領域、レベル指示語、レベル番号

# 第2章 順ファイル・中核(I)

# 学習目標

この章を終了した時点で、学習者が次の事柄ができるようになることを目標とする。

1. 単純入出力プログラムの読解、作成、デバッグができる。 なお、単純入出力とは順編成の入出力ファイルがそれぞれ1個であって 手続き部において、個々のレコード内の演算は行っても、特定項目につい て、全レコードの合計計算を行う等のレコード間処理は行わないものとす

- る。
- 2 次の事項を理解し、使用することができる。
  - (1) 見出し部
  - (2) 順ファイルにおける環境部、データ部
  - (3) データ部におけるファイル節と作業場所節 VALUE句
  - (4) 手続き部における次の動詞
    ACCEPT, ADD, COMPUTE, DISPLAY, DIVIDE, GO TO, IF, MOVE,
    MULTIPLY, PERFORM, STOP, SUBTRACT
    CLOSE, OPEN, READ, WRITE, REWRITE

# 内 容

# 2.1 単純入出力例の読解(60分)

(1) プログラム例2

プログラム例2を解説する。OHPや黒板に示す。次いで、次項以下に示す順にプログラム例2が行おうとしている内容の説明をする。

プログラム例2の入力文は、はじめGFO TOを用いたものを示し、 次にそれを用いないものに加える。このとき内PERFORM文の導入が 必要になる。

#### (2) 入出力ファイル

COBOLにあっては入出力のデータの集合はすべてファイルと呼ばれることを説明、レコード、項目についても説明する。ファイル・レイアウトの説明をする。

## (3) 処理手続き

プログラム例2の処理手続きを流れ図を示しながら説明する。

(4) 上のファイル・レイアウトと流れ図を突き合わせながらソース・リストを読解する。

#### 2.2 単純入出力プログラムの作成(120分)

(1) プログラム仕様書 ここで作成するプログラムの仕様書を示し理解させる。 流れ図も示す。

#### (2) 文法事項

ファイル・レイアウトを参照しながらデータ部をコーディングするに必要な文法事項を説明する。

流れ図を参照しながら手続き部をコーディングするために必要な文法事 項を説明する。

## (3) 実習

- ① コーディング用紙上にソース・プログラムを作成する。
- ② ディスプレイ入出力装置に向かいテキスト・エディタを用いてソース・プログラムを入力する。
- ③ コンパイル、実行を行う。
  - (注) 以上の手順は、 下の各章にほば共通である。

## 2.3 単純入出力プログラムのメンテナンス(120分)

#### (1) 文法エラー

上の実習の様子をみて、文法エラーで代表的なもの、特徴的なもの、特 異なもの等を選び出し、コーディング例を示しながらエラーの理由、修正 の仕方などの説明をする。

#### (2) 論理エラー

プログラム例 2 のレベルまでで発生する論理エラーを含んだプログラム を例に説明と演習を行う。

#### 2.4 文法事項まとめ(60分)

- (1) 順ファイルに関する見出し部、環境部については、この章で完了させる。
- (2) データ部 ファイル節の基本項目、集団項目、作業場所節の独立項目、基本項目、 集団項目を理解させる。
- (3) 本章学習内容 2(4)の動詞について、教えるレベル、範囲をあらかじめ策定すること。レベル、範囲は受講生のおかれた環境条件によって定める。

#### 2.5 演習問題

- (1) 本章の2.2 および2.3 はそのまま演習問題であり実習問題である。
- (2) 実習
  - 2.2(3)の実習問題はプログラム例1、プログラム例2のレベルを用意す る。

プログラム仕様書を与え、プログラムを作成せよ、という形態をとる。

(3) 演習

文法エラーや論理エラーを含んだソース・プログラムリストを与え机上 デバッグの演習を行う。

# 育成方法および指導上の留意点

- (1) プログラムの作成はコーディングだけが独立して存在しているのではなく その前段階にプログラム仕様書があり、さらにその前段階がある、というこ とを理解させる必要がある。
- (2) プログラム仕様書を与えられてプログラム作成に入るが、具体的にコーディングにかかる接点はあいまいなのが現実である。プログラム仕様書に流れ図は完成したものが与えられるのか、レコード・レイアウト、出力伝票デザ

インは与えられるのか、コーディングに用いるデータ名は与えられるのか、 などである。これらについては、どこかで明確に線引きし、そのことを周知 させてから実習演習等を行う必要がある。

- (3) 実習にあたっては、平易な問題から難度の高い問題までいくつか用意し、 平易な問題で達成の喜びを、難度の高い問題で挑戦の気力を養うなど配慮す る。
- (4) 生徒がわからないところ、できないところを尋ねてきた場合、ていねいに 
  応対することは勿論であるが、①ヒントを与えて自分で考えさせる方向、② 
  直接答えを教える、の二つを使い分ける必要がある。①が目標である。②は 
  避けるべきであるが、生徒の行き詰まりを打開する糸口となることもありう 
  る。ただし、②ばかりであると、上達は望めない。

# 用語

暗黙範囲符、英字、英数字、英文字、演算符号、固定長レコード、作業場所 節、出力ファイル、順ファイル、順編成、条件、条件文、真理値、数字定数、 注記行、定数、特殊文字、独立項目、独立データ記述項、入出力文、表意定数、 ファイル管理記述項、無条件文、明示範囲符、文字項目、文字定数

# 第3章 順ファイル・中核(Ⅱ)

## 学習目標

この章を終了した時点で、学習者が次の事柄ができるようになることを目標 とする。

- 1. 集計・作表プログラムの読解、作成、デバッグができる。 集計・作表プログラムの範囲・レベルを便宜上次のものとする。
  - (1) 入力ファイルのデータはキー項目を有し、キー項目は昇順、又は降順 に整列ずみとなっている。キー項目が複数の場合も想定する。
  - (2) 出力ファイルは原則として印字装置である。
  - (3) 報告書様式は、事前に印字用紙に罫線、見出し等が印刷されている場合と、見出し等をコンピュータから印字する場合、あるいはこれらが混合する場合のすべての場合を想定する。
  - (4) 入力ファイル、出力ファイルはそれぞれ1個を原則とする。
- 2. 上項の原則、即ち、出力ファイルが印字装置、入力ファイル、出力ファイルが複数個の場合にも応用ができる。
- 3. 次の事項を理解し、使用することができる。
  - (1) データ部における次の句などUSAGE、レベル番号88(条件名条件)
  - (2) 手続き部における次の動詞
    - ① 第2章で取り上げた動詞のより高度な使用法
    - ② INITIALIZE, SET

# 内容

- 3.1 集計・作表プログラム例の解読(540分)
  - (1) プログラム例3

プログラム例3を解説する。プログラム例3は小計、総合計の処理を行うプログラムとする。OHPや黒板に示す。次いで、事項以下に示す順に

プログラム例3か行おうとしている内容の説明をする。

(2) コントロール・ブレーク

昇順または降順に整列しているキー項目を比較してレコードのグループ を見分け、小計、中計、総合計処理などを行うコントロール・ブレーク処理を理解させる。

レコード数が10~20程度のデータを用いて処理の流れを具体的に説明する。特にコントロールの切れ目に注目する。

#### (3) 帳票編集

- ① 報告書頭書き、報告書脚書き、ページ頭書き、ページ脚書き、表側表示、改ページ処理などの方法を示す。
- ② 項目編集(英数字編集項目、数字編集項目)の例を解説。

#### 3.2 集計・作表プログラムの作成(1,620分)

(1) プログラム流れ図

プログラム例1、プログラム例2およびプログラム例3について復習をかねて流れ図で制御の流れを整理する。その応用として、次の2本のプログラムの流れ図の作成およびコーディングをさせる。なお、ともに帳票編集と行わない。

- ① 合計処理のみを行うプログラム
- ② 小計、中計、総合計処理を行うプログラム
- (2) 帳票設計

上記プログラムの帳票部分について、報告書頭書き、報告書脚書き、ページ頭書き、ページ脚書き、表側表示、改ページ処理等を含んだ帳票の設計問題を出し、帳票設計を経験させる。

# 3.3 集計・作表プログラムのメンテナンス(1,260分)

(1) 論理エラー

プログラム作成中に発生した論理エラーの代表的なものをモデルにエラーを含んだプログラムを示し、エラーの発見、修正の経験をさせる。

(2) 仕様の変更への対処

上の3.2(2)で行った帳票設計の通りに、3.2(1)で作成したプログラムを変更修正させる。

### 3.4 文法事項のまとめ(180分)

- (1) 本章で新たに出てきたOCCURS、USAGE 、レベル番号88(条件名条件) INITIALIZE文、SET 文のまとめを行う。
- (2) 見出し部の動詞について本章の学習事項をまとめる。教えるレベル、範囲は、受講生のおかれた環境条件を勘案しながら、第2章とバランスを考えて策定する。

#### 3.5 演習問題

本章の3.2および3.3はそのまま演習問題であり実習問題である。

## 育成方法および指導上の留意点

- (1) コントロール・ブレーク処理と帳票編集が完全にできるようになればCO BOLプログラムはほとんどマスターできたといってもよいくらいである。 一方、COBOLプログラム学習でつまずく場合、ほとんどがこの段階のようである。それだけにここをわかりやすく楽しく学習させるようにしなければならない。
- (2) プログラム例3は次のように考える。
  - データをできるだけ単純にする。
     例 図表 9 3 1
  - ② 明細行の印字、小計と総合計を計算印字する。例 図表9-3-2
  - ③ 処理の流れ図は大きくわけて次の2つのタイプが考えられる。両者のタイプをともに理解できることが望ましい。

例 図表 9-3-3、図表 9-3-4

④ 図表 9 - 3 - 3、図表 9 - 3 - 4 中の処理内容説明。開始 プログラムの開始

ファイルのオープン、ファイル終了フラグを 0 にする 準備 小計域、総合計域の初期化 データ ファイルの読込み ファイル終了フラグを1にする 処理 a 明細行の編集出力、小計累積、総合計累積 処理b キー項目を比較域へ移す 処理 c 処理 d 小計の編集出力 小計の編集出力、総合計の編集出力、ファイルのクロー 処理e ズ ループ1 ファイル終了フラグが 1 になるまでくり返す

ループ2 現在のキー項目の内容と比較域が異なるか、またはファ

(条件a) イル終了フラグが1になるまでくり返す

条件h 現在のキー項目と比較域の内容とが等しいか

(3) 上項を十分に理解できたところで帳票編集(報告書頭書き、報告書脚書き、ページ頭書き、ページ脚書き、表側表示、改ページ処理等)に入る。 ただし、上項の理解に時間がかかりそうな場合、帳票編集をプログラム例 2 あるいは第2章で作成したプログラムを用いて行う。

(4) コントロール・ブレーク処理と帳票編集を同時並行して学習しないように する。

# 用 語

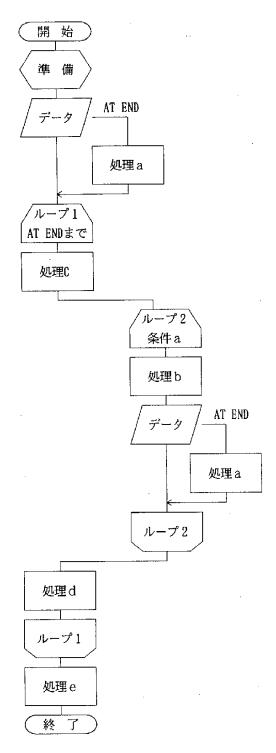
プログラム仕様書、コントロール・ブレーク、帳票編集、報告書頭書き、 報告書脚書き、ページ頭書き、ページ脚書き、表側、改ページ、条件名、条件 名条件、英数字編集、数字編集、初期化、文の範囲

図表 9-3-1 学生得点一覧ファイルのデー 合計点 学籍番号 学 年 クラス 0211 6 7 5  $\overline{(1)}$ 2 3 8 (2) 0 2 1 3 4 2 6 (3) 0 2 1 4 0 2 2 3 1 6 5 7 **(4)** 0 2 3 4 2 4 3 5 (5)  $\overline{2}$ 3 4 2 0 2 3 5 **6**) 2 637 (7)0 2 3 6 1 (8) 0 2 4 5 2 7 5 4 1 2 8 5 0 2 4 7 (9) 2 3 2 8 1 0 2 4 8 (1)0 2 4 9 1 2 6 6 3 (12) 2 3 5 8 0 1 1 1 2 2 7 4 2 (13) 0 1 1 2 5 6 7 (14) 0 1 1 3 2 3 8 5 (15) 0 1 2 3 2 7 4 3 **(16)** 0 1 2 4 (17) 0 1 2 7 2 3 6 3 6 4 3 8 (18)0 1 3 4 2 3 2 6 1 (19) 0 1 3 5

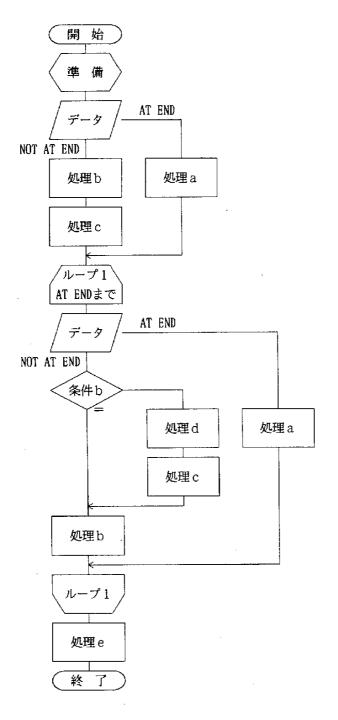
図表9-3-2 学生得点一覧表出力例

学生得点一覧表 (学年・全体)

学籍番号 0211 0213 0214 0223 0234	年 1 1 1 1 1 1	クラス 1 1 1 2 2 2	得点 675 238 426 657 435 342
0 2 1 3 0 2 1 4 0 2 2 3	1 1 1 1 1	1 1 1 2 2	2 3 8 4 2 6 6 5 7 4 3 5 3 4 2
0 2 1 4 0 2 2 3	1 1 1 1	1 1 2 2	4 2 6 6 5 7 4 3 5 3 4 2
0 2 2 3	1 1 1	1 2 2	6 5 7 4 3 5 3 4 2·
	1 1 1	2 2	4 3 5 3 4 2·
0 2 3 4	1 1	2	3 4 2
	1		
0 2 3 5		9	
0236		۷	637
0 2 4 5	1	2	754
0 2 4 7	1	2	285
0 2 4 8	- 1	2	3 2 8
0 2 4 9	1	2	663
小計			5 4 4 0
0 1 1 1	2	2	3 5 8
0 1 1 2	2	2	7 4 2
0 1 1 3	2	2	567
0 1 2 3	2	2 '	3 8 5
0 1 2 4	2	3	7 4 3
0127	2	3	6 3 6
0 1 3 4	2	3	4 3 8
0 1 3 5	2	3	2 6 1
· 小計			4 1 3 0
総合計			9570



図表9-3-4 多重条件



## 第4章 順ファイル・中核(Ⅱ)

### 学習目標

この章を修了した時点で、学習者が次の事柄ができるようになることを目標 とする。

- 1. マッチング・プログラムの読解、作成、デバッグができる。 マッチング・プログラムの範囲・レベルは便宜上次のものとする。
  - (1) マッチング・プログラムは突合せ型(マッチング・タイプ)処理を行う。突合せ型処理は次の4形態とする。
    - ① ファイルの併合処理
    - ② ファイルの突合せ処理
    - ③ ファイルの更新処理
    - ④ ファイルの維持処理
  - (2) 入力ファイルのデータはキー項目を有し、キー項目は昇順、又は降順 に整列ずみとなっている。
  - (3) 原則として入力ファイルは2個、出力ファイルは1個、このほか必要に応じ、印字出力用のファイルを1個付加する。
- 2. 入出力ファイルの個数が上記原則より増加しても応用できる。
- 3. 次の事項を理解し、使用することができる。
  - (1) プログラム・スイッチ
  - (2) 表意定数 (ZERO, SPACE, HIGH-VALUE, LOW-VALUE, QUOTE, ALL 定数)

# 内 容

- 4.1 マッチング・プログラム例の解読(360分)
  - (1) プログラム例 4

プログラム例 4 を示す。プログラム例 4 は 1 対 n の 突合せ処理を行うプログラムとする。 O H P や 黒板に示す。次いで、事項以下に示す順にプログラム例 4 が行おうとしている内容の説明をする。

#### (2) 突合せ例説明

プログラム例4の入力データのモデルを $4\sim5$ 組み1対1で用意し、プログラム例4の流れを説明する。次に、1対1のモデルデータを用意して、処理ができることを示す。また、データが1対1ならば不要なルーチンがあるので、その存在を説明する。

## 4.2 マッチング・プログラムの作成(1,620分)

- (1) ファイルの併合処理の作成
- (2) ファイルの突合せ処理の作成
- (3) ファイルの更新処理の作成
- (4) ファイルの維持処理の作成

### 4.3 マッチング・プログラムのメンテナンス(300分)

(1) 論理エラー

プログラム作成中に発生した論理エラーの代表的なものをモデルにエラーを含んだプログラムを示し、エラーの発見、修正の経験をさせる。

(2) 仕様の変更への対処

併合処理を突合せ処理に機能変更、更新処理を維持処理に機能変更等の 機能変更、あるいは1対1型を1対n型に変更するなどの経験をさせる。

# 4.4 文法事項のまとめ(60分)

- (1) 表意定数についてまとめを行う。
- (2) 見出し部の動詞について本章の学習事項をまとめる。教えるレベル、範囲は、受講生のおかれた環境条件を勘案しながら、第2章および第3章とバランスを考えて策定する。

#### 4.5 演習問題

本章の4.2 および4.3 はそのまま演習問題であり実習問題である。

### 育成方法および指導上の留意点

- (1) マッチング・プログラムの指導は、2個のファイルの突合せから入る。
- (2) 突き合わせについては次の4つの類型について易から難へ順を追って指導する。
  - ① ファイルの併合処理
  - ② ファイルの突合せ処理
  - ③ ファイルの更新処理
  - ④ ファイルの維持処理
- (3) ファイルの終了処理についてモデルデータを提示しながら注意を喚起す。
- (4) プログラム・スイッチの使い方を学習させるとき、その得失を十分に納得させる。使ったほうがよい場合、好ましくない場合について例を用いて説明する。
- (5) 突合せによるデータの出力状況の説明にはOHPの使用が効果的である。
- (6) この章では文法事項で新たなものは表意定数であるが、このうちSPACE と ZEROは前章までのどこかで学習させておく方法もある。

# 用 語

突合せ処理、ファイルの併合(ファイル・マージ)、ファイルの突合せ(ファイル・マッチング)、ファイルの更新(ファイル・アップデート)、ファイルの維持(ファイル・メンテナンス)、プログラム・スイッチ

# 第5章 順ファイル・中核(IV)

# 学習目標

この章を終了した時点で、学習者が次の事柄ができるようになることを目標とする。

- 1. データ構造の繰返しの論理的な概念、構造について理解し、表の定義、 反復句、再定義句の使用ができる。
- 2. 添字(整数、整数項目名、指標名)を理解し、添字付けができる。
- 3. 指標データ項目を使用できる。
- 4. 添字付けによる表操作、SEARCH文による表操作、および SET文の操作が できる。
- 5. PRRFORM 文と組合せて表操作ができる。
- 6. 表操作をIF文と GO TO文を用いてできる。
- 7. 次の事項を理解し、使用することができる。
  - (1) データ部における次の句など DCCURS, REDEFINES
  - (2) 手続き部における次の動詞
    - ① これまでの章で取り上げた動詞のより高度な使用法
    - ② SEARCH, SET
    - ③ 次の動詞は本章の中でなんらかのかたちで取り上げる EVALUATE, INSPECT, EXIT, COTINUE, STRING, UNSTRINC

# 内容

# 5.1 表操作プログラム例の解読(180分)

(1) プログラム例5

プログラム例5を示す。プログラム例5は表操作処理を行うプログラムとする。OHPや黒板に示す。次いで、プログラム例5が行おうとしている内容の説明をする。

OCCUERS 句を使用せずに書いた場合の冗長性を示す。また、SEARCH文を 使用せず、IF文と GO TO文を用いて表操作を行ってみせる。

(2) 表の次元

1次元、2次元の表を示す。可能性として7次元まで可能なことを示す。

## 5.2 表操作プログラムの作成(420分)

(1) データ部

OCCUERS 句、REDEFINES 句を含んだデータ部の作成。

(2) 手続き部

SEARCH文による表操作、PERFORM VARYING ・・・文による表操作、および SET文を含んだ手続き部の作成。

(3) 表探索の各種技法

順次探索、複合表探索、区間探索、直接探索、2分探索、多次元表探索、 内部ソート技法等を含んだプログラム作成。

### 5.3 表操作プログラムのメンテナンス(240分)

(1) 論理エラー

プログラム作成中に発生した論理エラーの代表的なものをモデルにエラーを含んだプログラムを示し、エラーの発見、修正の経験をさせる。

(2) 仕様の変更への対処

データ部の表の設計を工夫して、表操作をSEARCH文、SEARCH ALL文、PERFORM 文、IFと GO TO文による操作など様々な方法を経験させる。

## 5.4 文法事項のまとめ(60分)

- (1) 本章で新たに出てきたOCCURS句、REDEFINES 句、SEARCH文、SET 文のまとめを行う。
- (2) 直接表操作とは関係ないが、EVALUATE, INSPECT, EXIT, COTINUE, STRING UNSTRING各動詞のまとめを行う。レベル、範囲は前章までの状況とのバランスを考えて策定する。

#### 5.5 演習問題

本章の5.2および5.3はそのまま演習問題であり実習問題である。

### 育成方法および指導上の留意点

- (1) COBOL プログラム学習者にとって第1の難関がコントロール・ブレーク処理とすれば、第2の難関がこの表操作である。表操作は冗長性を覚悟すれば、OCCURS句、REDEFINES 句、SEARCH文、SET 文なしで処理可能である。そこで、内容的には表操作である処理をOCCURS句、REDEFINES 句、SEARCH文、SET 文なしで行う例を示し、次に、順次これら表操作用の機能を利用してみせ、これらが大変便利なものであることを示す。
- (2) この章までで、中核機能単位について一通り終了する。したがって、表操作と直接関係ないか動詞EVALUATE, INSPECT, EXIT, CONTINUE, STRING, UNSTRING について、一通り説明する。動詞全体にいえることであるが、特にこれらの動詞については使い方の詳細を記憶するのではなく、概略を理解しておき、必要に応じ文法書を自分で読んで使いこなせるように指導する。

# 用 語

表、表操作、表探索、添字付け、次元、指標名、指標データ項目、再定義、 順次探索、複合表探索、区間探索、直接探索、2分探索、多次元表探索、内部 ソート技法

### 第6章 相対ファイル

### 学習目標

この章を修了した時点で、学習者が次の事柄ができるようになることを目標 とする。

- 1 相対ファイルを使用するプログラムの読解、作成、デバッグができる。
- 2. 相対ファイルにおけるキーの役割を説明できる。
- 3. 相対ファイルと順ファイルを用途に応じて的確に使い分けできる。

# 内容

- 6.1 相対ファイル・プログラム例の解読(180分)
  - (1) プログラム例6グループプログラム例6グループはプログラム例61、プログラム例62、およびプログラム例63の3本からなる。
    - ① プログラム例 6 1データ・レコードが存在する相対ファイルの生成を内容とするプログラム
    - ② プログラム例 6 2 相対ファイルの順呼出しを内容とするプログラム
    - ③ プログラム例63相対ファイルの乱呼出しを内容とするプログラム
    - OHPや黒板に示す。次いで、プログラム例6グループのそれぞれが行 おうとしている内容の説明をする。
  - (2) 相対ファイルにおけるキー 相対ファイルにおける相対レコード番号と相対キーが、直接的に相対ファイル中のレコード項目とは関係がないことを十分に理解させる。
  - (3) 無効キー条件 どのようなとき無効キー条件がとなるかを説明する。

### 6.2 相対ファイル・プログラムの作成(420分)

(1) 環境部

ACCESS MODE 句、ORGANIZATION IS RELATIVE句を含んだ環境部の作成。

(2) データ部 必要に応じ相対キーの項目記述を行う。

(3) 手続き部

OPEN, READ, DELETE, REWRITE, WRITE, START各動詞を使用する順呼出し、乱呼出しおよび動的呼出し等を含んだプログラムの作成。

(4) プログラム内容

商品コード・ファイル、得意先コード・ファイル等のコード・ファイル の作成、維持、およびそれらコード・ファイルへのアクセス等。

## 6.3 相対ファイル・プログラムのメンテナンス(240分)

(1) 論理エラー

プログラム作成中に発生した論理エラーの代表的なものをモデルにエラーを含んだプログラムを示し、エラーの発見、修正の経験をさせる。

(2) 仕様の変更への対処

併合処理の中の更新処理や維持処理のマスタ・ファイルを相対ファイル に変更するなどの経験をさせる。

# 6. 4 文法事項のまとめ(60分)

- (1) 環境部、データ部および手続き部それぞれ相対ファイル特有の文法事項をまとめる。
  - ① 環境部 ACCESS MODE 句、ORGANIZATION IS RELATIVE句
  - ② データ部 相対キーの項目記述
  - ③ 手続き部 OPEN, READ, DELETE, REWRITE, WRITE, START各動詞
- (2) 順呼出し、乱呼出しおよび動的呼出しそれぞれにおける環境部、データ 部および手続き部の書き方の異同を整理する。このとき、図表9-6-1(OPEN 文と呼出し法の実行可能な入出力文)を用いる等の工夫をする。

図表9-6-1 OPEN文と呼出し法の実行可能な入出力文

呼出し法	文	OPEN文のモード			
		INPUT	OUTPUT	O – I	EXTEND
	READ	0			<del></del>
	WRITE	_	0	•	0
順呼出し	REWRITE	_	_	0	<del>-</del>
	START	0	_	0	_
	DELETE	_	· –	0	_
乱呼出し	READ	0	_	0	_
	WRITE	_	0	0	
	REWRITE			0	<u></u>
	START		_		
	DELETE	_		0	<del>-</del>
	READ	0	_	0	<del></del>
動的呼出し	WRITE	**	0	0	_
	REWRITE	_	_	0	_
	START	0	_	0	_
	DELETE	_	_	0	

○印が実行可能

## 6.5 演習問題

本章の6.2 および6.3 はそのまま演習問題であり実習問題である。

# 育成方法および指導上の留意点

(1) 相対ファイルの指導では、まず、プログラム例 6 1 を机上で説明、次に、 モデルデータを10~20レコード用意し、相対ファイルを生成する。 モデルデータは3種類用意する。

- ① 図表9-6-2 モデルデータ ファイル(a) 相対キーとファイルのレコードの中のキー項目とが完全に一致している 例。
- ② 図表9-6-2 モデルデータ ファイル(b) ファイルのレコードにキー項目があるが、相対キーとは一致していない。 ただし、演算式により相対キーと一致するよう換算できる。
- ③ 図表9-6-2 モデルデータ ファイル(c) ファイルのレコードの中にはキー項目に相当するものがない。相対ファイルではこのようなものでも、プログラマは任意のレコード、例えば、3 番目のレコードであるとか、8番目のレコードであるとかのアクセスができることを説明する。なお、この場合、レコード項目内容との関連付けによるアクセスはできない。
- (2) 上の各モデルについて、コンピュータで実行し、結果のファイル・ダンプ リストを示して説明する。
- (3) プログラムを作成は順ファイルと組合せた処理とする。例えば、順ファイルのトランザクション・データを読込んで、相対ファイルのコード・ファイルを索引する処理等。
- (4) 順ファイルと相対ファイルの特徴、異同について平易な例を用いて説明、 納得させる。

# 用語

相対ファイル、相対レコード番号、相対キー、相対キー項目、乱呼出し、動 的呼出し、無効キー条件

図表9-6-2 モ デ ル デ ー タ

格納順	商品3-1 商品名	商品コード 商品名	メーカー 商品名
1	01 ラジオ	03 ラジオ	MA ラジオ
2	02 テレビ	06 テレビ	SO F V E
. 3	03 レイゾウコ	09 レイゾウコ	SA レイゾウコ
4	04 ソウジキ	12 ソウジキ	SH ソウジキ
5	05 センタクキ	15 センタクキ	MA センタクキ
6	06 センプウキ	18 センプウキ	HI センプウキ
7	07 クーラー	21 7-5-	MI クーラー
8	08 ラジカセ	24 ラジカセ	SOラジカセ
9	09 ヒゲソリキ	27 ヒゲソリキ	BR ヒゲソリキ
1 0	10 1 7 1	30 1 7 1	SE F 7 1
1 1	11 スイハンキ	33 スイハンキ	HI スイハンキ
1 2	12 アイロン	36 アイロン	MA アイロン
	ファイル(a)	ファイル(b)	ファイル(c)

## 第7章 索引ファイル

### 学習目標

この章を終了した時点で、学習者が次の事柄ができるようになることを目標 とする。

- 1. 索引ファイルを使用するプログラムの読解、作成、デバッグができる。
- 2. 索引ファイルにおけるキーの役割を説明できる。
- 3. 索引ファイル、相対ファイルおよび順ファイルを用途に応じて的確に使い分けできる。

# 内容

### 7.1 索引ファイル・プログラム例の解読(180分)

- (1) プログラム例7グループ プログラム例7グループはプログラム例71、プログラム例72、およ びプログラム例73の3本からなる。
  - ① プログラム例71データ・レコードが存在する索引ファイルの生成を内容とするプログラム
  - ② プログラム例72 索引ファイルの順呼出しを内容とするプログラム
  - ③ プログラム例73索引ファイルの乱呼出しを内容とするプログラム)HPや里板に示す。次いで、プログラム例7グループのそ

OHPや黒板に示す。次いで、プログラム例7グループのそれぞれが行おうとしている内容の説明をする。

(2) 索引ファイルにおけるキー 索引ファイルにおける参照キーと相対ファイルにおける相対キーの違い を中心に説明する。また索引ファイルでは主レコードキーのほかに副レコ ードキーの使用ができることを説明する。 (3) 無効キー条件

どのようなとき無効キー条件となるかを説明する。

### 7.2 索引ファイル・プログラムの作成(420分)

(1) 環境部

ACCESS MODE 句、ORGANIZATION IS INDEXED 句、RECORD KEY句、ALTERNATE RECORD KEY句を含んだ環境部の作成。

- (2) データ部 索引ファイル中に記述するレコードキー。
- (3) 手続き部 OPEN, READ, DELETE, REWRITE, WRITE, START各動詞を使用する順呼出し、乱 呼出しおよび動的呼出し等を含んだプログラムの作成。
- (4) プログラム内容

商品コード・ファイル、得意先コード・ファイル等のコード・ファイルの作成、維持、およびそれらコード・ファイルへのアクセス。マスタ・ファイルの作成、維持、およびそれへのアクセス。主レコードキー、副レコードキーの使用。

# 7.3 索引ファイル・プログラムのメンテナンス(240分)

(1) 論理エラー

プログラム作成中に発生した論理エラーの代表的なものをモデルにエラーを含んだプログラムを示し、エラーの発見、修正の経験をさせる。

(2) 仕様の変更への対処

併合処理の中の更新処理や維持処理のマスタ・ファイルを索引ファイル に変更するなどの経験をさせる。

## 7.4 文法事項のまとめ(60分)

- (1) 環境部、データ部および手続き部それぞれ相対ファイル特有の文法事項をまとめる。
  - ① 環境部 ACCESS MODE 句、ORGANIZATION IS INDEXED 句

### RECORD KEY句、ALTERNATE RECORD KEY句

- ② データ部 参照キーの項目記述
- ③ 手続き部 OPEN, READ, DELETE, REWRITE, WRITE, START各動詞
- (2) 順呼出し、乱呼出しおよび動的呼出しそれぞれにおける環境部、データ 部および手続き部の書き方の異同を整理する。このとき、図表 9 6 1 (相対ファイル OPEN文と呼出し法の実行可能な入出力文)と同様なもの を用いる等の工夫をする。

### 7.5 演習問題

本章の7.2 および7.3 はそのまま演習問題であり実習問題である。

### 育成方法および指導上の留意点

- (1) 索引ファイルの指導では、まず、プログラム例71を机上で説明、次に、 モデルデータを10~20レコード用意し、索引ファイルを生成する。
- (2) モデルデータを用意する。
  - ① 相対ファイルで用いた例のように簡単なものとする。
  - ② 主レコードキーのほかの副レコードキーを含んだものにする。
- ③ 相対ファイルと違ってファイルのレコードの中にはキー項目に相当する ものがない必ず存在していなければならないことに注目させる。
- (3) 上のモデルデータについて、コンピュータで実行し、結果のファイル・ダンプリストを示して説明する。
- (4) プログラムの作成は順ファイルと組合せた処理とする。例えば、順ファイルのトランザクション・データを読込んで、索引ファイルのマスタ・ファイルを削除、追加、更新する処理等。
- (5) 順ファイル、相対ファイルおよび索引ファイルそれぞれの特徴、異同について平易な例を用いて説明、納得させる。

# 用語

索引ファイル、参照キー、主レコードキー、副レコードキー、レコードキー

# 第8章 整列併合

## 学習目標

この章を終了した時点で、学習者が次の事柄ができるようになることを目標 とする。

- 1. 整列併合機能を使用するプログラムの読解、作成、デバッグができる。
- 2. 整列併合キーの種類、機能、用法について説明できる。

# 内 容

- 8.1 整列併合機能を用いたプログラム例の解読(180分)
  - (1) プログラム例8グループ プログラム例8グループはプログラム例81、プログラム例82、およびプログラム例83の3本からなる。
    - ① プログラム例81SORT文の USING句とGIVING句を用いたもの
    - ② プログラム例82
      SORT文の INPUT PROCEDURE指定とOUTPUT PROCEDURE指定を用いたもの
    - ③ プログラム例83 MERGE 文を用いたもの

OHPや黒板に示す。次いで、プログラム例8グループのそれぞれが行 おうとしている内容の説明をする。

(2) SORT文と MERGE文の違いを説明する。正しく書かれた MERGE文の動詞 MERGE を動詞SORTにおきかえても、処理機能は全く同じであるが、正しく 書かれたSORT文の動詞SORTを動詞 MERGEにおきかえた場合は同じ処理機能にはならないことなどの説明をする。

## 8.2 整列併合機能を用いたプログラムの作成(420分)

- (1) 環境部 整列併合ファイル指定を含んだ環境部の作成。
- (2) データ部 レベル指示語SDによる整列併合用ファイルの記述。SORTまたは MERGE 用キーの設定。
- (3) 手続き部 次の動詞を含んだプログラムの作成。
  - ① SORT文について次の組み合わせ
    - (a) USING 指定とGIVING指定
    - (b) INPUT PROCEDURE 指定とOUTPUT PROCEDURE指定
    - (c) USING 指定とOUTPUT PROCEDURE指定
    - (d) INPUT PROCEDURE 指定とGIVING指定
  - ② MBRGE 文について次の組み合わせ
    - (a) USING 句とGIVING句
    - (b) USING 句とOUTPUT PROCEDURE句
  - ③ RELEASE, RETURN
- (4) プログラム内容

第4章の併合処理や突合せ処理で学習したプログラム処理内容をアレンジして利用する。

# 8.3 整列併合機能を用いたプログラムのメンテナンス(240分)

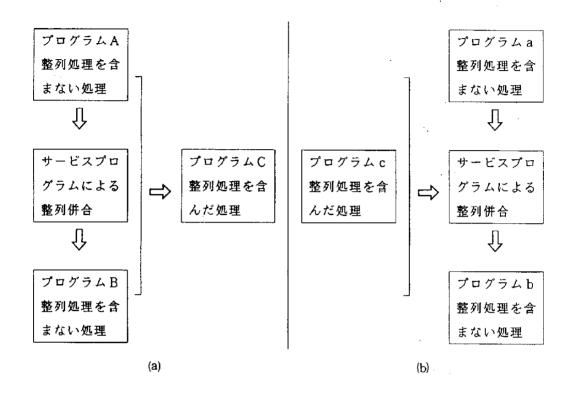
(1) 論理エラー

プログラム作成中に発生した論理エラーの代表的なものをモデルにエラーを含んだプログラムを示し、エラーの発見、修正の経験をさせる。

(2) 仕様の変更への対処

図表9-8-1(a)のように整列処理を含まないプログラムA、サービス・プログラムによる整列併合、整列処理を含まないプログラムBといった処理 過程を整列処理を含んだプログラムCに変更する、あるいは図表9-8-1(b) のようにその逆を経験させる。

図表9-8-1 整列併合仕様変更演習例



### 8.4 文法事項のまとめ(60分)

- (1) 環境部、データ部および手続き部それぞれ整列併合特有の文法事項をまとめる。
  - ① 環境部 整列併合用ファイル指定
  - ② データ部 レベル指示語 SD、SORTまたは MERGE用キー
  - ③ 手続き部 SORT, MERGE, RELEASE, RETURN 各動詞
- (2) キーの昇順、降順、キーの強さの順、文字のキー項目の大小順序についてまとめる。

### 8.5 演習問題

本章の8.2 および8.3 はそのまま演習問題であり実習問題である。

### 育成方法および指導上の留意点

- (1) SORT文については次のとおり。
  - ① 簡単な例、すなわち、SORT文の USING指定とGIVING指定の組合わせでできる処理から始め、最終的には、SORT文の INPUT PROCEDURE指定とOUTPUT PROCEDURE 指定の組合わせの処理まで進む。
  - ② 図表 9-8-1 を行わせる場合も易から難へといくつかの段階を踏んで進める。
  - ③ COBOL プログラムの中でSORTを行う場合と、サービス・プログラムとの 組合せで処理する場合との得失、長短を考えさせる。
- (2) MERGE 分については次のとおり。
  - ① SORT文で代用が可能であること。
  - ② ただし、処理系によっては同じ処理をする場合、SORT文と MERGE文では データ処理実行速度が異なることがある。
- (3) 上記の例題はあるいは実習問題等は一つのSORT文だけを服務、いわゆる、 基本整列プログラムで行う。文法仕様上、一つのプログラムに複数のSORT文 を含んだり、SORT文の INPUT PROCEDUREとOUTPUT PROCEDURE以外の手続きを 含んだ拡張整列プログラムが許されている。余裕があれば、拡張整列プログ ラムの例を示すとよい。拡張整列プログラム例をみることで、整列併合機能 を理解しやすくなることがある。

# 用 語

整列、併合、整列併合、昇順、降順、入力手続き、出力手続き、引渡し、引取り

## 第9章 プログラム間連絡その他

## 学習目標

この章を終了した時点で、学習者が次の事柄ができるようになることを目標とする。

- 1. プログラム間連絡機能を使用したプログラムの読解、作成、デバッグができる。
- 2. 翻訳単位が異なったプログラム間の連絡について説明できる。
- 3. 翻訳単位が一つのプログラムの入れ子のプログラムに関連したプログラム連絡について説明できる。
- 4. 原始文操作、報告書作成、通信機能各単位について機能の概略を説明ひることができる。

## 内容

- 9.1 プログラム間関連機能を用いたプログラム例の解読(180分)
  - (1) プログラム例9グループ プログラム例9グループはプログラム例91A・プログラム例91Bならびにプログラム例92A・プログラム例92Bからなる。
    - プログラム例91A・プログラム例91B プログラム例91A・プログラム例91Bは別の翻訳単位からなるプログラムである。ただし、プログラム例91Aはプログラム例91Bを呼ぶ機能(CALL "プログラム例91B" USING・・・)を有している。一方、プログラム例91Bは、プログラム例91Aから呼ばれるプログラムである。
    - ② プログラム例92A・プログラム例92B プログラム例92Aは原始プログラムの手続き部の最後の部分に プログラム例92Bを入れ子としている。
    - OHPや黒板に示す。次いで、プログラム例9グループのそれぞれが行

おうとしている内容の説明をする。

(2) 入れ子のプログラム この機能は COBOLとしては比較的新しい機能である。実用上具合のよい 例を考えさせる。

## 9.2 プログラム間関連機能を用いたプログラムの作成(780分)

- (1) データ部 連絡節の設定、EXTERNAL句、GLOBAL句の使用法。
- (2) 手続き部 CALL, CANCEL, EXIT PROGRAM, USE各動詞を用いた文の作成。
- (3) プログラム内容 たとえば、2組の数字を渡して、呼ばれるプログラムでその和を算出結果を引き取るといった程度の平易な例題を用いる。

## 9.3 プログラム間関連機能を用いたプログラムのメンテナンス(420分)

- (1) 論理エラー プログラム作成中に発生した論理エラーの代表的なものをモデルにエラーを含んだプログラムを示し、エラーの発見、修正の経験をさせる。
- (2) 仕様の変更への対処 翻訳単位別の呼ぶプログラムと呼ばれるプログラムを入れ子構造のプログラムに変換したり、あるいはその逆を行ったりする。

# 9.4 文法事項のまとめ(60分)

- (1) データ部および手続き部それぞれプログラム間関連機能特有の文法事項をまとめる。
  - ① データ部 連絡節、EXTERNAL句、GLOBAL句
  - ② 手続き部 CALL, CANCEL, EXIT PROGRAM, USB各動詞、およびプログラム終わり見出し
- (2) 他の機能と関連している部分にふれる。たとえば、USE BEFORE REPOR TING文。

#### 9.5 演習問題

本章の9.2および9.3はそのまま演習問題であり実習問題である。

#### 9.6 その他(360分)

- (1) 原始文操作機能単位 前章までの中のどこかで進行状況を勘案しながら、この機能を説明し、 実習の中に折りこむ方法を工夫する。
- (2) 報告書機能単位、通信機能単位 この2つの機能単位については、時間的に余裕があれば、プログラム例 を示し、内容を説明する。それぞれ機能の概略が理解できる程度にとどめ る。

## 育成方法および指導上の留意点

- (1) プログラム間関連機能を含んだプログラム例、同じくその作成問題についても、内容は極力平易なものとする。
- (2) プログラム間のデータのやり取りの方法の部分に重点をおく。
- (3) 呼ぶプログラムと呼ばれるプログラムを別々の人あるいはグループに作成 させデバッグの共同作業を行わせるなどの工夫をする。
- (4) 翻訳単位が別々のプログラム間の連絡の場合と、一つの翻訳単位の中の入れ子の内外との連絡の場合の両者について、処理内容に応じた利害得失を考えさせる。上の両者が混在するプログラムも検討させる。
- (5) 原始文操作機能そのものは平易な機能であるから、本章以前の段階に適宜 入れ込むことが望ましい。
- (6) 報告書機能単位と通信機能単位については概略理解の範囲に留めてよいと どめる。

用 語

翻訳単位のプログラム、実行単位のプログラム、データ名の有効範囲、大域名、局所名、外部属性、外部(内部)対象、外部(内部)レコード、共通プログラム、初期化プログラム、ファイル結合子、呼ぶプログラム、呼ばれるプログラム、引き数

# 推薦教材

# インストラクタ必読書

- 1. 「JIS 電子計算機プログラム言語COBOL JIS X 3002-1988」
- 2. 「実習時に使用するコンパイラにかかわるマニュアル」

# 参考文献

1. 「入門 COBOL」 西村恕彦・植村俊亮 共著, オーム社, 1988年

# 第10部 FORTRAN

# 教育目標

本指針におけるFORTRANの教育の目標は、初級情報処理技術者に求められる科学技術計算のプログラム技法について、JIS FORTRAN上位水準に準拠して、適正プログラムを作成する能力を習得することにある。

# 内容の構成

	時間数		
内 容	講義	演習	
FORTRAN FORTRANの特徴 翻訳と実行	5		
FORTRANプログラムの形式 FORTRANの文字集合 プログラムの形式	5		
プログラムの構造 事 例 アルゴリズムの定義と性質 三つの基本形と流れ図 モジュールと適正プログラム	5		
直線形のプログラム 事例解析 文 法 演習問題	. 5	5	
選択形のプログラム 事例解析 文 法 演習問題	5	5	

反復形のプログラム(1) 事例解析(DO WHILE形) 文 法 演習問題 整数型データと固定小数点方式	5	5
反復形のプログラム(2) 事例解析(REPEAT UNTIL形) 文 法 演習問題	5	5
反復形のプログラム(3) 事例解析(中間判定形) 文 法 演習問題	5	5
反復形のプログラム(4) 事例解析 (DO形) 文 法 演習問題 浮動小数点方式	5	10
1 次元配列とプログラム 事例解析 文 法 演習問題	5	1 0
2 次元配列とプログラム 事例解析 文 法 演習問題	5	2 5
文関数を引用するプログラム 事例解析 文 法 演習問題	5	1 0
関数副プログラムの引用 事例解析 文 法 演習問題	5	2 5
サブルーチン副プログラムを引用ふるプログラム <b>事例解析</b>	5	2 5

文 法 演習問題			¥	
	合	計	7 0	1 3 0

### 第1章 FORTRAN解説

### 学習目標

- 1. FORTRANの特徴、開発と標準化の歴史を理解する。
- 2. 実習計算機に即して、FORTRANの原始プログラムの翻訳と実行の仕組を理解する。

# 内 容

### 1. 1 FORTRANの特徴

(1) プログラム言語FORTRAN

FORTRANは高水準言語の一つであり、本来、数値計算を行うための言語である。又、手順を明確な算法として表現できる問題向き言語である。他のプログラム言語と比較しながら説明する。

(2) JIS FORTRAN

最初のFORTRANの開発からJIS FORTRAN制定までの過程を、特にFORTRAN77と関連付けて説明する。

FORTRAN JIS規格の利用方法を説明する。特定のデータ処理システムに依存する規定外事項は、そのシステム供給者の「FORTRA N手引書」等を参照することを説明する。

#### 1.2 翻訳と実行

(1) 翻訳と実行、JCL

実習計算機に即して、FORTRAN原始プログラムの翻訳と実行の過程、JCLの役割を説明する。ソースリスト、エラーメッセージ、プログラムとデータの区別、デバッグの大要について説明する。

実習計算機による処理の様子を見学する。計算機室の入退室、利用上の ルールについて説明する。

### 育成方法および指導上の留意点

- (1) FORTRANの開発と標準化の過程は、計算機科学、情報処理技術の発展過程や動向と深く結び付いている。歴史の説明が単に年代順に主要項目を羅列して終るというような平板なものにならないよう、「生きた歴史」として、又学習者の「よい動機づけ」となるよう、教師の経験やトピックスなど交えながら授業展開するようにする。
- (2) 電子計算機がシステムとして稼動していること、計算機実習が計算機室要員等の様々な作業に支えられてはじめて可能であることを認識させる。

### 用 語

高水準言語,問題向き言語,手順向き言語,コンパイラ言語,JIS FORTRAN,基本水準,上位水準,FORTRAN77,処理系,原始プログラム、翻訳,実行,ソースリスト,エラーメッセージ

## 第2章 FORTRANプログラムの形式

### 学習目標

1. FORTRANプログラムの形式、文字の書き方の規則に従って、コーディングの練習をす。

# 内 容

#### 2. 1 FORTRANの文字集合

(1) 英数字、特殊文字

FORTRANで使用する英数字、特殊文字及び処理系に用意されている使用頻度の高いその他の文字を説明する。

(2) 文字の書き方

誤読を防ぐための文字の書き方の規則や留意事項を守って、正確な文字を記述する。

#### 2. 2 プログラムの形式

(1) 文、文番号、行の領域

文、文番号、行の正しい領域を理解して、プログラム例をコーディング する。

プログラム単位を説明し、END文以降に注釈行(空白行)があっても、 翻訳時のエラーになることを説明する。

- (2) コントロール・シート ユーザ名、アカウント名、パスワードを覚え、正しいコントロールシートを作成する。
- (3) 誤り修正

実習計算機システムに即して、誤りの修正方法を説明する。

- 1. コーディングミスや誤読の発生は、プログラミングの生産性を低下させること、プログラムは公的な製品であり文書である、という認識を持たせることが大切である。OCRディスプレイを観察させて、誤読されやすい文字や「自分のくせ」を認識させるとよい。
- 2. 端末装置からプログラムを投入するとき、カラムずれを防ぐためプログラムの開始行として\*234のようにカラムの位置を示す注釈行を入れることもよい(最後には削除する)。

文番号が第6カラム(継続欄)にかかって間違いを生ずる学習者が多いので、文番号はすべて第1カラム(左詰めにして)打鍵することもよい。

#### 用語

コーディング, コーディング用紙, プログラム・カード, データ・カード, 制御カード, OCRシート, コントロール・シート, パスワード, 文字集合, 英字, 数字, 特殊文字, 文, 開始行, 注釈行, 継続行, 文番号, カラム, EN D文, プログラム単位



#### 第3章 プログラムの構造

#### 学習目標

- 1. 事例を完成して、アルゴリズムの意味とその基本性質を理解する。
- 2. 三つの基本形の特長を説明し、流れ図に表現する。モジュールの意味を理 解する。

### 内 容

#### 3.1 事 例

手作業、手計算によって事例を完成する。

- (1) 奇数魔法陣の作成 (ド・ラ・ルーブルの方法)
- (2) データの整列 (選択交換法)
- (3) 平方根の近似値計算(ニュートン法)

#### 3. 2 アルゴリズムの定義と性質

(1) アルゴリズムの定義

明確に定義された有限個の規則の集まりであって、有限回適用することによって問題を解くもの(JIS情報処理用語より)の意味を、事例を参照して理解する。

(2) 基本性質

アルゴリズムは正当性、一意性、停止性、汎用性を満足していなければ ならないことを、事例を参照して理解する。

事例(1)は  $n^2$  回、事例(2)は (n-1) 回の操作で必ず終了するが、事例(3) は操作の終了条件を与えなければならないことを説明する。

### 3. 3 三つの基本形と流れ図

(1) 直線形

文S₁の次に文S₂を、次に文S₃を ……、と順に文を実行する形である。

#### (2) 選択形

論理式 eが真ならば文S₁を実行し、偽ならば文S₂を実行する形で、どちらの文を実行しても実行後は必ず一つの結合点に合流する。

(3) 反復形DO WHILE形 論理式 eが真である間、文Sを繰返し実行する形である。

#### 3. 4 モジュールと適正プログラム

(1) ユークリッドの互除法 論理の組立てによって、アルゴリズムの解読性、流れ図が異なり、プログラミングに大きく影響することを説明する。

(2) 適正プログラム プログラムをモジュール化、構造化する必要性を説明する。適正プログ ラムの特長を説明する。

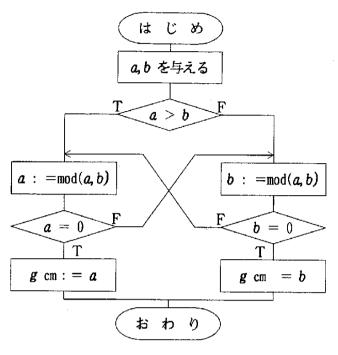


図3-5-1 基本形によらない流れ図

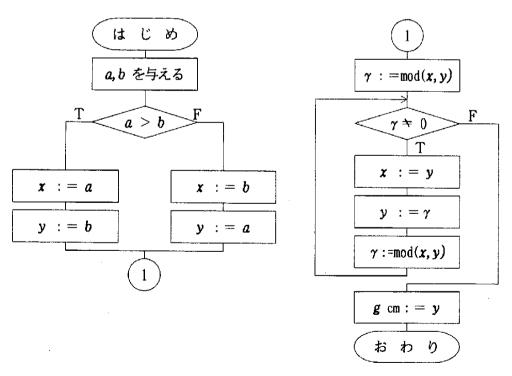


図3-5-2 モジュール化された流れ図

- (1) 本章での指導の内容は、以降のFORTRANプログラミングのスタイルを決定する。即ち、本部は第3編第1部の「プログラム流れ図の作成技法」のフィロソフィを継承し、本部の教育の目標に示すように「適正プログラムの作成」を目指している。従って「文法を小出しに紹介しながらその文法の範囲内で自由なスタイルでプログラミングする」というものではなく、「問題を解析してモジュール化、構造化された流れ図を作成し、これをプログラム化するに必要な文法を適用する」という観点にたっている。
- (2) 本章で提示する事例は、学習者が実際に手を動かし、その過程を経験的に 理解し、アルゴリズムの定義及びその性質を把握させることが重要である。 以下の事例も役立つ。

組立除法、エラトステネスの篩による素数表の作成。

- (3) 反復形はDO WHILE形の紹介だけにとどめる。ユークリッドの互除法の流れ 図中、剰余 $\gamma$ を求める文を 2 回用いてDO WHILE形にしているが、学習者から 「最初の文 $\gamma$ :=mod(x,y)に制御を戻す中間判定形の適用の可否について」 の質問が予想される。混乱しない説明を用意しておく必要がある。
- (4) 本章の選択形及び以下の流れ図において、例えば a:0 を判断ボックスの中に書込み、分岐する流れ線の傍に等号、不等号を添える表現は使用しない。
- (5) 算術代入文の等号にあたる記号は、FORTRANの様にイコールを用いてもよいが、矢印やPASCAL風にコロンイコールを用いることもよい。 文の区切り記号セミコロンを用いることもよい。

### 用語

アルゴリズム,算法,三つの基本形,直線形,選択形,IF THEN ELSE形,反復形,DO WHILE形,モジュール,論理式,真,偽,適正プログラム,奇数魔方陣(ド・ラ・ルーブルの方法),整列(選択交換法),ニュートン法,ユークリッドの互除法,素数(エラトステネスの篩),組立除法

### 第4章 直線形のプログラム

#### 学習目標

直線形の事例、演習問題を解析し、流れ図を作成し、プログラムを完成する。

#### 内 容

#### 4.1 事 例 解 析

- (1) 四則計算、剰余
  - 2 整数の値をプログラム上で定義し、四則、剰余を求めて印字する。
- (2) 2数の入れ換え
  - 2整数の値をプログラム上で定義し、値を入れ換えて印字する。

### 4.2 文 法

- (1) PROGRAM文、プログラム名、英字名、STOP文、EMD文 プログラム単位を説明する。実習計算機システムに即してプログラム名 とJCLの関係を説明する。
- (2) データの型、型宣言、型宣言文、INTEGER文、整数型データ、整定数、整数型変数、変数名、数値記憶単位 プログラム会体を把握するため、すべての変数は型宣言することを説明

プログラム全体を把握するため、すべての変数は型宣言することを説明 する。

- (3) 算術演算子、算術式、算術定数式、演算の優先順位、整数化された商、 算術代入文
- (4) PRINT文、書式識別子、出力並び、並びによる出力文 実際の出力結果例を説明する。

#### 4.3 演習問題

(1) 恒等式

(例)  $a^3 + b^3 = (a + b)(a^2 - ab + b^2)$ について、2数a、b の

値をプログラム上で定義し、左辺と右辺の値をそれぞれ計算して等しいことを確認する。

(2) 級数の和

(例)  $\sum_{i=1}^{n} i=n(n+1)/2$ について、nの値をプログラム上出定義し、式 i=1

を計算して結果を確認する。

#### 育成方法および指導上の留意点

(1) 講師は旧JIS FORTRANにこだわってはいけない。現FORTRANに含まれない旧文法を紹介することは、学習者に混乱を生じさせるだけである。

(注:FORTRANの歴史の項で発展過程例として紹介すべきである)

(2) 算術式を正しく理解させる一方法として、「算術式の(逆)ポーランド記法への変換」を指導することは有効である。

### 用 語

PROGRAM文,プログラム名,英字名,データの型,型宣言,型宣言文,INTEGER文,整数型データ,整定数,整数型変数,変数名,数値記憶単位,算術演算子,算術式,演算の優先順位,整数化された商,整定数式,算術代入文,PRINT文,書式識別子,出力並び,並びによる出力文,STOP文

### 第5章 選択形のプログラム

### 学習目標

選択形の事例、演習問題を解析して流れ図を作成し、プログラムを完成する。

### 内 容

### 5.1 事 例 解 析

- (1) 2数の大小比較 2整数の値を入力して、大きい数(小さい数)を印字する。
- (2)最大値、最小値3整数の値を入力し、最大値、最小値を印字する。

### 5.2 文 法

- (1) ブロック I F文、E L S E文、E N D I F文、I F ブロック、E L S E ブロック、字下げ効果 字下げをして選択形の構造を明確にする。
- (2) 関係演算子、関係式、論理式
- (3) 入力並びによる入力文、入力並び、READ文 入力データ例を示して説明する。
- (4)組込み関数MOD、総称名、引数 組込み関数MODは奇偶判定などに用いられることを説明する。

### 5. 3 演習問題

- (1) IF THEN形の適用 最大 (小) 値判定を、IF THEN ELSE形で表現する。
- (2) 正負、奇偶判定整数を入力して正負、奇偶判定する。

こ時点で、論理IF文とGO TO文による選択形は紹介しない。IF THEN ELSE形、IF THEN 形に徹すべきである。

用語

ブロックIF文, ELSE文, END IF文, IFブロック, ELSEブロック, 関係演算子, 関係式, 論理式, 入力並び, 並びによる入力文, REA D文, 組込み関数MOD, 総称名, 引数

### 第6章 反復形のプログラム(1)

### 学習目標

- 1. 先行判定反復形(DO WHILE形)の事例、演習問題を解析して流れ図を作成し、プログラムを完成する。
- 2. 整数型データの正負の最大値を理解し、プログラミングに役立てる。

### 内 容

#### 6.1 事 例 解 析 .

(1) 級数の和

n を入力して、l から nまでの自然数の和、偶数の和、奇数の和、階乗を求める。

(2) フィボナッチ数列フィボナッチ数列を発生して印字する。

#### 6.2 文 法

(1) 論理 I F文、GO TO文

DO WHILE形は、2つの構造で表現できることを説明する。

DO WHILË形の2つの表現

(2) READ文、WRITE文、FORMAT文、装置指定子、行送り、I 形編集、アポストロフィ編集、X形編集

入力データ形式とREAD文及びFORMAT文、出力形式とWRIT

E文及びFORMAT文の関係を事例を示して説明する。

#### 6.3 演習問題

(1) 最大公約数

2整数を入力して、ユークリッドの互除法により最大公約数を求めて印字する。

(2) 級数の和

n の値を入力して次の値を求める。

$$1 + (1+2) + (1+2+3) + \cdots + (1+2+\cdots+n)$$

### 6. 4 整数型データと固定小数点方式

事例(1)の階乗計算の出力結果を参照して、整数型データの値x は $-2^{-1} \le x$   $\le 2^{-1} - 1$ (nは 1 記憶数値単位のビット数) であることを説明する。

#### 育成方法および指導上の留意点

- (1) 事例(1)では、正順の和、逆順の和を求めるよう指導する。
- (2) 実習計算機に即して永久ループの中断方法を事前に指導しておく。

### 用語

論理IF文,GO TO文,READ文,WRITE文,装置指定子,FORMAT文,行送り,I形編集,アポトロフィ編集,X形編集,固定小数点方式、フイボッチ数列,漸化式

### 第7章 反復形のプログラム(2)

### 学習目標

後続判定反復形(REPEAT UNTIL形)の事例、演習問題を解析し、流れ図を作成しプログラムを完成する。

### 内 容

#### 7. 1 事 例 解 析

(1) 逐次近似法

方程式  $x^2 - x - 1 = 0$  の近似解を、漸化式

$$x_{n+1} = 1 + 1/x_n$$
 (n=0, 1, 2...)

により求める。初期値  $x_0=1.0$  、精度  $\varepsilon=10^{-6}$ を入力し、収束の過程も印字する。

(2) 平方根 (ニュートン・ラフソン法)

平方根√2 の近似値を、漸化式

$$x_{n+1} = \frac{1}{2} (x + \frac{a}{x_n})$$

により求める。組込み関数SQRTを用いるときの値と比較する。

#### 7.2 文 法

(1) 実数型データ、実定数、基本実定数、実数型変数、REAL文、型の変換

実数型データは、実数値を処理系が近似したものであることを説明する。

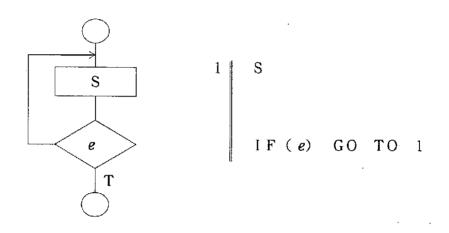
- (2) 組込み関数ABS、SQRT、個別名 組込み関数の引数に注意する。
- (3) P形編集、E形編集、斜線編集

入力データ形式とREAD文及びFORMAT文、出力形式とWRIT E文及びFORMAT文との関係を例を示して説明する。入力データ、出

力形式を設計する。

#### (4) REPEAT UNTIL形

REPEAT UNTIL形は、FORTRANでは直接表現できないので、流れ図中の論理式を否定した形にして表現することを説明する。



(a) 流 れ 図 (b) FORTRANの表現 図3-5-3 REPEAT UNTIL形の表現

#### 7.3 演習問題

#### (1) 黄金分割比

フィボナッチ数列の後項と前項の比は、黄金分割比に収束することを確める。

#### (2) 平方根表の作成

数 aを 1 から10まで0.5 きざみに変えて、平方根 $\sqrt{a}$  の近似値を、ニュートン・ラフソン法により求める。

(1) 事例 1 をDO WHILE形にして、REPEAT UNTIL形との違いを理解させる。

$\varepsilon:=10^{-6}$		
$x_n := 1.0$		
$\boldsymbol{x}_{n+1} := 1 + 1 / \boldsymbol{x}_n$		
x <sub>n+1</sub> , x <sub>n+1</sub> -X <sub>n</sub> の印字		
$ x_{n+1}-x_n \geq \varepsilon$		
$x_n := x_{n+1}$		
$\boldsymbol{x}_{n+1} := 1 + /\boldsymbol{x}_n$		
$x_{n+1}, x_{n+1}-x_n OF$	字	

$\varepsilon:=10^{-6}$
$x_{n+1}:=1,0$
 $x_n := x_{n+1}$
$x_{n+1}:=1+1/x_n$
$x_{n+1}$ , $x_{n+1}-x_n$ の印字
$ x_{n+1}-x_n \geq \varepsilon$

(a) DO WHILE形 (b) REPEAT UNTIL形

図3-5-4 2つの構造図

次のような流れ図を作らないよう指導する。

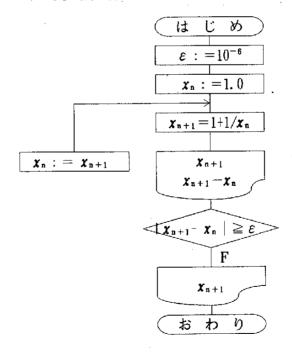
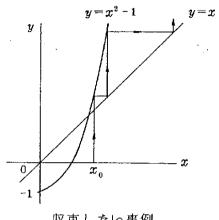


図3-5-5 基本形によらない流れ図

(2) 事例 1 で漸化式を  $x_{n+1}=x_n^2-1$  (n=0, 1, 2……) とすると、発散 することがあることを説明する。



収束しない事例

(3) 実数型同士の相等判定を用いてはならないことを事例を示して指導する。 (例えば、0.2 を 5 回加えても 1.0に等しくならないことを示す。)

実数型データ、実定数、基本実定数、実数型変数、REAL文、F形編集、 E形編集、斜線編集、組込み関数ABS、組込み関数SQRT、個別名、逐次 近似法、ニュートン・ラフソン法、黄金分割比

#### 第8章 反復形のプログラム(3)

#### 学習目標

中間判定反復形の事例、演習問題を解析し、流れ図を作成し、プログラムを 完成する。

### 内 容

#### 8.1 事 例 解 析

- (1) 総和と平均
  - エンドデータ又はデータの個数を添えたデータを入力して、総和と平均 を求める。
- (2) 2次方程式の解

ファイル終了指定子を含むREAD文により、n 個の2次方程式の係数を入力し、解の公式により実数解を求めて印字する。虚数解のときは"虚数"と印字する。

#### 8.2 文 法

(1) ファイル終了指定子、制御情報並び、書式付き入出力文 並び項目をもつREAD文を説明する。

(UNIT=)u, (FMT=)f, END=s

#### 8.3 演習問題

(1) ヘロンの公式

n 組の三角形の3辺の長さを入力し、三角形を形成するかどうかを判定 し、形成するときはヘロンの公式によりその面積を求める。

(1) 中間判定反復形は、原則的には基本形に含まれない。

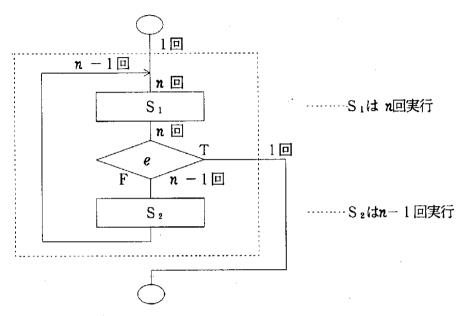


図3-5-6 中間判定反復形の流れ図

流れ図中、文 $S_1$  は n回実行され、文 $S_2$  は (n-1) 回実行され、実行 回数が 1 回だけ異なることに注意する。

### 用語

ファイル終了指定子, 制御情報並び, 書式付き入出力文, ヘロンの公式

#### 第9章 反復形のプログラム(4)

#### 学習目標

- 1. DO形反復形の事例、演習問題を解析し、流れ図を作成し、プログラムを 完成する。
- 2. 計算機の計算誤差を認識して、プログラミング、結果の評価に役立てる。

### 内 容

#### 9.1 事 例 解 析

(1) 関数表の作成

x の値を 0 から50まで 1 ずつ変えて、 $x^2$ ,  $x^3$ , 2 ,  $\sqrt{x}$  の関数表を作成する。結果を見易くするため、5 行毎に空白行を入れる。2 はオーバーフローしない間印字する。

(2) 自然対数の底 e

自然対数の底 eの値を次式により求める。

 $e_1 = (\cdots (((1)+1/1!)+1/2!)+\cdots)+1/n!$ 

 $e_n = ((\cdots (((1)+1/n!)+1/(n-1)!)+\cdots)+1/2!)+1/1!$ 

 $e_3 = ((\cdots (((0)+1/n!)+1/(n-1)!)+\cdots)+1/2!)+1/1!+1/0!$ 

#### 9.2 文 法

(1) DO文、DOループの端末文、CONTINUE文、DOループの範囲、DO変数、初期値パラメータ、終値パラメータ、増分パラメータ DOループの範囲を実行した後の変数の値を、例を示して説明する。

#### 9.3 演習問題

(1) 分子の有理化

 $ax^2 + bx + c = 0$  において、  $b^2 \ge 4ac$ のとき解の公式の分子を有理化すれば精度があげられることを確かめる。

(2) DO形化

前章までの事例、演習問題をDO形化する。

9. 4 実数型データと浮動小数点方式

実数型データと浮動小数点方式、計算誤差について例を示して説明する。

#### 育成方法および指導上の留意点

- (1) DOループを脱出した後でDO変数を不用意に引用しないよう説明する。
- (2) a=0.5、b=0.2、c=0.1を2進、16進に変換して1数値記憶単位に近似値が格納されることを確かめる。又、3 F 15. 4、3 F 15. 10 により出力して結果を確認する。

用 語

DO文, DOループの端末文, CONITNUE文, DOループの範囲, DO変数, 初期値パラメータ, 終値パラメータ, 増分パラメータ, 浮動小数点方式

#### 第10章 1次元配列とプログラム

#### 学習目標

1次元配列を用いる事例、演習問題を解析し、流れ図を作成しプログラムを 完成する。

### 内 容

#### 10. 1 事 例 解 析

- (1) ベクトルの内積
  - n 個の品目の単価と個数を読込んで、合計金額を求める。
- (2) 標準偏差
  - n 個のデータの平均と標準偏差を求める。

#### 10. 2 文 法

(1) 1次元配列、配列宣言、配列名、配列要素、寸法宣言子、上限、下限、 添字式

プログラム上で配列宣言文を書く位置を説明する。データを格納するに 必要な配列の寸法指定の仕方を説明する。配列要素の引用に際して、添字 式の値が寸法を超えないよう十分注意しなければならないことを説明する。

(2) DO形並び

入力データの形式とREAD文及びFORMAT文の関係、出力形式とWRITE文及びFORMAT文の関係を、例を示して説明する。入力データ、出力形式を設計する。

### 10. 3 演習問題

(1) 相関問題

2数を組とする n個のデータを入力して、相関係数を求める。

#### (2) 組立除法

n 次多項式の係数を入力して、組立除法により式の値を求める。

#### 育成方法および指導上の留意点

- (1) DO文と配列を結びつける手法が、特に大切であることを説明する。 配列の寸法をオーバーするとき、致命的な誤りになることを説明する。処 理系に用意されているデバッグ文を紹介して、配列オーバーをチェックする デバッグ文を説明することもよい。
- (2) 入出力設計に十分時間をとって練習することが必要である。

### 用語

1次元配列,配列宣言,配列名,配列要素,寸法宣言子,上限,下限,添字式,DO形並び,標準偏差,ベクトルの内積,相関係数

## 第11章 2次元配列とプログラム

#### 学習目標

2次元配列を引用する事例や演習問題を解析し、流れ図を作成し、プログラムを完成する。

### 内 容

#### 11. 1 事 例 解 析

(1) 奇数魔方陣

ド・ラ・ルーブルの方法により 3 次から n次までの奇数魔方陣を作成する。

(2) 減衰振動曲線

減衰振動  $y = e^{-x} \sin (x + \theta)$  のグラフを作成する。

### 11. 2 文 法

(1) 2次元配列

配列要素は列として順序付けられていることを説明する。通常の行列の行方向と2次元配列の行方向が逆順になっていることを説明し、グラフの出力等の場合、逆順にとり出さなければならないことを説明する。

(2) DO形並び

入力データ形式、出力形式とDO形並びをもつREAD文、WRITE 文及びFORMAT文の関係を例を示して説明する。入力データ、出力形 式を設計する。

(3) CHARACTER文、DATA文、文字代入文 文字型データは文字型宣言が必要であることを説明する。文字型データ と数値データの関係は、処理系によって異なることを説明する。

#### 11. 3 演習問題

(1) 素数

「エラトステネスの篩」によりできるだけ多くの素数を作る。

(2) 行列の積

2つの行列を読込んで、行列の積を計算する。

### 育成方法および指導上の留意点

(1) DO文と2次元配列との結びつきは、FORTRANプログラミングで最も有用な技法である。特に外側のDOループが内側のDOループを制御するような場合の指導に力点をおく必要がある。

#### 用 語

2次元配列, CHARACTER文, DATA文, 文字代入文, 組込み関数 SIN, 組込み関数COS, 組込み関数EXP

### 第12章 文関数を引用するプログラム

### 学習目標

文関数を引用する事例、演習問題の流れ図を作成し、プログラムを完成する。

### 内 容

#### 12. 1 事 例 解 析

- (1) 方程式の解法(二分法) 方程式の1実数解をはさむ2数を入力して、二分法により解の近似値を 求める。
- (2) 数値積分(シンプソン則) 定積分の上限、下限の値を入力して、シンプソン則により数値積分する。

#### 12. 2 文 法

(1) 文関数定義文、文関数の引用、仮引数、実引数、関数名 文関数は1行で定義される関数である。仮引数と実引数の型、順序、個 数が一致しなければならないことを説明する。

### 12. 3 演習問題

- (1) 方程式の解法(はさみ打ち法) はさみ打ち法により方程式の近似解を求める。
- (2) 連立方程式の解法(クラーメルの定理) 2元連立方程式をクラーメルの定理により解く。行列式の値はサラスの 規則を適用する。

文関数は次章の関数副プログラムに置き換えられるので、それ程力点を置く 必要はない。

用語

文関数定義文, 文関数の引用, 仮引数, 実引数, 関数名, 二分法, シンプソン則, はさみ打ち法, クラーメルの定理, サラスの規則

### 第13章 関数副プログラムの引用

#### 学習目標

関数副プログラムを引用する事例や演習問題を解析して流れ図を作成し、プログラムを完成する。

### 内 容

#### 13. 1 事 例 解 析

- (1) 行列式の値n 次の行列式の値を求める。
- (2) 進法変換2 進数を10進法に変換する。

#### 13. 2 文 法

(1) 主プログラム、関数副プログラム、外部関数名、手続き、FUMCTI ON文、RETURN文

関数副プログラムの外部関数名が値をもち、主プログラムに値を持ち帰ることを説明する。

関数名は関数副プログラム内で、変数として表れなければならないこと を説明する。

### 13. 3 演習問題

- (1) 関数副プログラム 文関数を用いた事例や演習問題を、関数副プログラムを引用するプログ ラムに変える。
- (2) パスカルの三角形 m 個の中から n個をとる組合せ計算を利用して、パスカルの三角形を作成する。

- (1) 問題のどの部分を関数副プログラム化するかを発見することが指導の要点である。
- (2) 関数副プログラム内で、例えば F(X, Y) = X + Y のように左辺に誤った記述をする学習者がいるので注意して指導する。

用 語

主プログラム、関数副プログラム、外部関数名、手続き、FUNCTION 文、RETURN文、パスカルの三角形

### 第14章 サブルーチン副プログラムを引用するプログラム

### 学習目標

サブルーチン副プログラムを引用する事例や演習問題を解析し、流れ図を作成してプログラムを完成する。

### 内 容

#### 14. 1 事 例 解 析

- (1) 連立方程式の解法(掃き出し法) n 元連立方程式を掃き出し法により解く。係数行列の逆行列を求める。
- (2) 全加算器のシミュレーション 2つの10進整数を入力して、全加算器の働きをシミュレーションする。

#### 14. 2 文 法

- (1) サブルーチン、サブルーチン副プログラム、サブルーチン名、CALL文サブルーチン副プログラムの特長、関数副プログラムとの相違を説明する。
- (2) COMMON文、共通ブロック名、無名共通ブロック 引数による統合との相違を説明する。
- (3)整合配列、定配列宣言子、仮配列宣言子、実配列宣言子 整合配列により柔軟な副プログラム化ができることを説明する。

#### 14. 3 演習問題

(1) 素因数分解

素数表を作り、これを利用して2から100までの整数を素因数に分解する。

#### (2) 進法変換

10進、2進、16進変換表を作成する。2進にはパリティビットを付加する。

#### (3) 疑似英文

英文を入力して各文字の出現頻度を整列してヒストグラムを画く。第0次、第1次、第2次近似の疑似英文を発生する。

(4) ポーランド記法

1文字の変数からなる任意の算術式をポーランド記法に変換する。

#### 育成方法および指導上の留意点

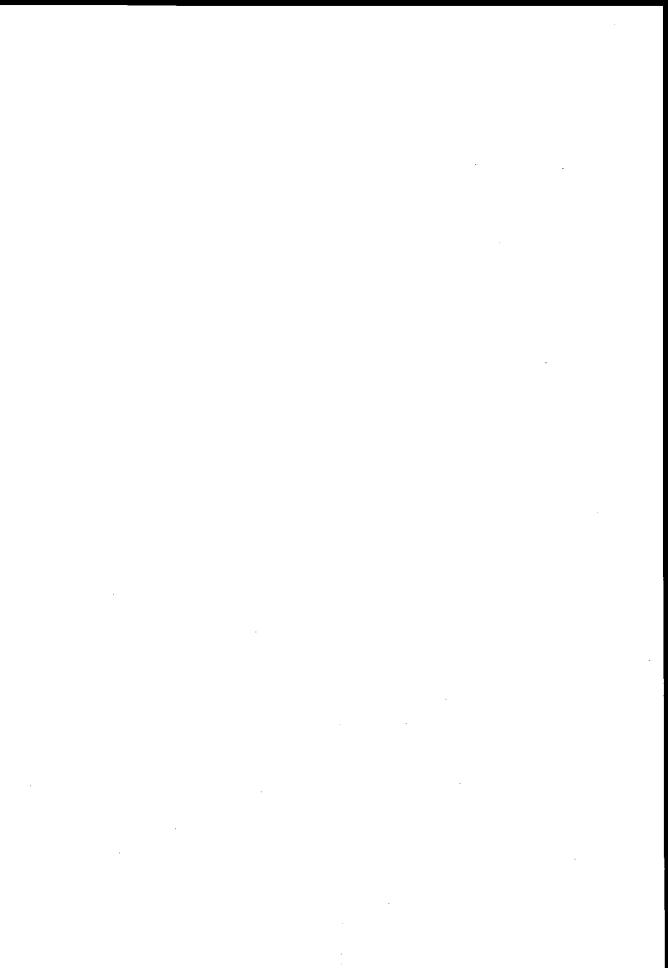
副プログラム化は、適正プログラム作成の最も重要な技法である。どの部分 を副プログラム化するか、問題の解析に力点をおかなければならない。

### 用 語

サブルーチン, サブルーチン副プログラム, サブルーチン名, CALL文, COMMON文, 整合配列, 定配列宣言子, 仮配列宣言子, 実配列宣言子, 共通ブロック名, 無名共通ブロック, 掃き出し計算, 逆行列, 全加算器

### 参考文献

- 1. FORTRAN77入門 浦 昭二編, 培風館, 1982年
- 2. けつこさこうめ77 大駒誠一, サイエンス社, 1982年
- 3. 人文科学のFORTRAN77 西村恕彦、東大出版会、1980年



#### 第11部 C言語

#### 教育目標

C言語の学習を修了した時点で、次の事柄ができるようになることを目標と する。

- 1. 数値計算、ビット操作、文字処理、ファイル処理などの範囲にC言語を 適用する。
- 2. プログラム中で宣言されたデータ構造がメモリ上にどの様にマッピング されるかをイメージし、説明する。
- 3. 構造化プログラミングを目指した読み易いプログラム作りを実践する。
- 4. C言語の文法はすべて利用する。標準ライブラリ関数は、必要なものをマニュアルから見つけて利用する。
- 5. ポインタ変数,配列,構造体を,どんな場面で利用すれば良いかを見極められる。
- 6. 記憶クラスの違いにより、変数の有効領域が異なることを意識したうえでプログラムする.

# 内容の構成

内容	時 間 数	
	講義	実 習
第1章 C言語の概要 1.1 C言語の特徴 1.2 C言語の構造 1.3 演習問題	2. 5	1. 5
第2章 数値の計算 2.1 整数の計算 2.2 実数の計算 2.3 演習問題	4. 5	5. 5
第3章 選択型のプログラム 3.1 if文を使った選択型のプログラム 3.2 switch文を使った選択型のプログラム 3.3 goto文を使ったプログラム 3.4 演習問題	5	8
第4章 反復型のプログラム 4.1 do-while文を使った反復型のプログラム 4.2 while 文を使った反復型のプログラム 4.3 for 文を使った反復型のプログラム 4.4 演習問題	5	1 0
第5章 ビット演算 5.1 ビット演算の基本 5.2 データの表現方法を調べる 5.3 演習問題	3. 5	8
第6章 入出力処理 6.1整数の入力 6.2 実数の入力 6.3 1 6進数の入力 6.4 演習問題	4.5	7
第7章 配 列 7.1 1次元配列 7.2 2次元配列 7.3 演習問題	6	1 0

(続く)

内容	時間数	
	講義	実 習
第8章 文字処理 8.1 文字の入出力 8.2 文字列の入出力 8.3 演習問題	5	8
第9章 ポインタ変数 9.1 ポインタ変数 9.2 ポインタ変数と配列 9.3 演習問題	5	8
第10章 関数 10.1 関数の作り方 10.2 ポインタ変数の引数 10.3 共通に使う変数 10.4 演習問題	4.5	1 0
第11章 標準ライブラリ関数 11.1 標準ライブラリ関数の呼び出し方 11.2 標準ライブラリ関数の活用 11.3 演習問題	4.5	1 0
第12章 記憶クラス 12.1 変数の有効領域 12.2 レジスタの利用 12.3 演習問題	4	7
第13章 ファイル処理 13.1 ファイルの開き方 13.2 順編成ファイルの作成 13.3 直編成ファイルの作成 13.4 演習問題	1 0	1 2
第14章 特徴的なプログラム 14.1 システム・コール 14.2 共用体 14.3 リカーシブ・コール 14.4 テキスト・ファイルの出力 14.5 プリプロセッサ 14.6 自作関数のリンク 14.7 演習問題	1 6	1 5
合 計	8 0	120

#### 第1章 C言語の概要

### 学習目標

学習者がこの章を修了した時点で、C言語に関連した次の事柄ができるようになることを目標とする。

- 1. C言語の機械語的特徴を2つ挙げる。
- 2. C言語の移植性の特徴を説明する。
- 3. C言語の関数型言語的特徴を説明する。
- 4. C言語の基本的な構造を説明する。

### 内 容

#### 1.1 C言語の特徴(90分)

(1) C言語の使われ方

C言語はUNIX OSの開発の過程で生まれた言語で、はじめはOSを記述するために作られた言語であることを説明する。この時、UNIX の歴史と現在の普及ぶりにも触れるとよい。

現在、C言語はパーソナル・コンピュータからメイン・フレーム、スーパー・コンピュータまで使用可能であり、とくにパーソナル・コンピュータ用のパッケージ・ソフト/アプリケーション・ソフトのかなりの部分がC言語で書かれていることを述べる。パーソナル・コンピュータのソフトウェアの開発がなぜC言語で行われる様になったのかの理由について以下のような話を述べるとよい。

- ① 以前はすべてアセンブラと機械語で書かれていた。
- ② アセンブラや機械語では生産性が悪い、移植性が悪い、習得が難しいなど問題が多い。
- ③ C言語はOSを記述するために作られていた言語でアセンブラ的使い 方も出来、ハードウェアに依存しない言語であるため移植性が良い、構 造化された言語で習得が易しいなどの特徴を持つことなどから、パーソ

ナルコンピュータのソフトウェア開発用言語として使われる様になった。 ③の詳しい内容については、あとで詳しく述べるのでここでは細かく説 明する必要はない。

#### (2) C言語の移植性

C言語の規格は最近まで、C言語の開発者であるKernighan、Ritchieが書いた『The C Programing Language』という本に沿っていた。このため方言の少ない言語であり、高い移植性を保つことが出来た。しかし上記本で定義されていない部分は処理系により解釈が分かれていた。これらの部分をANSIが規定し、あいまいさをより少なくしたものが、ANSI規格のC言語であり、これによりC言語の移植性がより一層高くなった。以上述べた様にC言語は高い移植性を持つことからC言語で書いてあるアプリケーション・ソフトウェアを、異なるコンピュータで使う場合は一度コンパイルすれば、そのまま動いてしまうことが多い。また、UNIX OSが多くのコンピュータで動いている理由は、UNIXの大半がC言語で書かれているためである。以上のことを説明し、C言語の移植性が高い理由を理解させる。

#### (3) C言語のその他の特徴

システム記述言語としての特徴として、bit ハンドリングが得意であり、ポインタ変数があるので指定したアドレスの内容を操作することができることを説明する。この時、アセンブラとC言語で同じこと行う例を比較すると良い。

構造化言語としての特徴としては、構造化プログラミングを可能とする制御構文 (if~then~else~endif, for文, while 文, do~ while文, switch文) が用意されていることを説明する。この時、各制御構文の細かい説明はしない。概要の説明と、構造化プログラミングとは何かについても触れると良い。

関数型言語としての特徴では、すべての処理が関数で行われ、最初に実行される関数が main()関数であることを説明する。この時、他の言語、FORTRAN、BASICなどと比較すると良い。再帰呼び出しが可能であることにも触れる。

#### 1.2 C言語の構造(60分)

#### (1) サンプル1

図表11-1-1で表す様な簡単なサンプルプログラムを示し、各行についての説明を行う。ここでは、主にmain関数、実行文、コメントについて説明する。main以外の関数についての説明はその機能の簡単な説明にとどめる。

#### (2) サンプル2

2番目のサンプル・プログラムとしては図表11-1-2で示す様なサンプルプログラムを示す。このプログラムの説明の中では、C言語の内部コードがすべてASCIIであり、当然、変数名もデータも大文字、小文字の区別をしていることを説明する。また、字下げの必要性についても触れるとよい。

while の条件式内の関数やelse if などについては、この様な使い方も出来るとの説明にとどめる。

図表1-1-1 サンプル・プログラム1

```
# include <stdio.h>
main()

{

    /* 変数の宣言 */
    char c; /* キャラクタ変数 */
    int i; /* ループカウンタ */

    i=0; /* 初期化 */
    do {

        c=getchar(), ++i; /* 文字入力, & カウント*/
    ) while (c!=EOF);

    printf("loopcounter=%d\n", i);
}
```

#### 図表1-1-2 サンプル・プログラム2

#### (3) 予約語

C言語で規定されている予約語をすべて示し、それらの意味を簡単に 紹介し、それらが変数、関数の名前に使えないことを説明する。

### 1.3 演習問題(90分)

(1) Cの普及

C言語がパーソナル・コンピュータのアプリケーション開発言語として 普及した理由を説明させる。

(2) Cの構造・

1. 2で示したサンプル・プログラムを示し、このプログラムはどの様な動きをするプログラムかを説明させる。また、基本的なプログラムの構造をサンプル・プログラムを使い説明させる。

# 育成方法および指導上の留意点

1. この章では、これからC言語を勉強しようとする人間に対し、C言語とは、 どの様な言語か、どの様にして生まれて来たのか、どの様な特徴があり主に どの様な用途に使われているかを理解させ、C言語に対する興味を持っても らう様に指導する。

- 2. この章で説明したC言語が普及した理由が主にMS-DOSのパーソナル・コンピュータ上のアプリケーションにおいてであり、UNIX上でC言語が普及した理由と違うことに留意する。
- 3. C言語の構造の説明では、C言語とはどんなものであるかを知らせること、 次章以降の説明が理解出来る程度で良い。

### 用 語

UNIX, OS, パーソナル・コンピュータ, メイン・フレーム, スーパー・コンピュータ, パッケージ・ソフト, アプリケーション・ソフト, 機械語, アセンブラ, 移植性, 構造化プログラミング, ANSI, bit ハシドリング, ポインタ, アドレス, 制御文, 関数、関数型言語, main(), FORTRAN, BASIC, ASCII, 予約語

## 第2章 数値の計算

### 学習目標

学習者がこの章を終了した時点で、数値の計算に関連した次の事柄ができる ようになることを目標とする。

- 1. main関数を作成する。
- 2. 間違いのない変数の宣言をする。
- 3. 定数と変数の違いを説明する。
- 4. 代入文を自由に使用する。
- 5. 2項演算子の内で算術演算に関するものを活用する。
- 6. 整数と実数のすべてのデータ型が取り扱える。
- 7. printf関数と書式指定子(整数型と実数型の全て)を使用する。ただし、 プリプロセッサ#includeについては、標準ライブラリ関数を利用するとき に必要であることを認識させる程度でよい。

# 内 容

# 2.1 整数の計算(180分)

(1) 整数の四則演算

初めてC言語のプログラムを学習するので、簡単なサンプル・プログラムを提示し、時間をかけて解説する必要がある。サンプル・プログラムは、たとえば2つの数の足し算といったものでよい。サンプル・プログラムの解説後には、実際にそのプログラムを実行させるとよい。サンプル・プログラムとしては、定数の演算、定数と変数を組み合わせた演算といったものを2~3示す。printf関数の中でも演算ができることを説明しておくとよい。ここで取り扱うデータ型は、intに止める。

(2) 色々なデータ型による整数の四則演算

サンプル・プログラムを使って、整数には色々なデータ型があることを 説明する。サンプル・プログラムはデータ型の違い(主にビット幅)を分 からせるようなものにする。たとえば、short だとあふれてしまうかけ算が、longだとあふれないといったことを実感させるサンプル・プログラムである。また、たとえば演算子%を使った若干複雑な演算も、サンプル・プログラムとして示しておくとよいであろう。

#### 2.2 実数の計算(90分)

(1) 四捨五入するプログラム

実数型の定数や変数を取り扱ったサンプル・プログラムを示す. それは、単に実数型を使った割り算やかけ算といった演算でよい。次に. 実数型の変数と整数型の変数が混在した代入文を含むサンプル・プログラムを示し、その場合の数値がどうなるかを確認させる。最後に、今までの学習範囲内で四捨五入するプログラムを学習者に考えさせるとよいであろう。また、ここでキャストについても触れておく.

(2) 精度の高い実数の演算

ここでは倍精度型の実数を説明する。特に、float との違いを分からせるようなサンプル・プログラムを示すとよい。また、浮動小数点を使った定数の表現方法についても説明しておく。

#### 2.3 演習問題(330分)

(1) 四則演算

この章で学習したデータ型と演算子を、すべて確認させるような問題を 数題出す。問題としては、単純な四則演算を色々なデータ型で計算し、出 力するといったものでよい。

(2) 余りを求める

演算子%を使わないで余りを求める方法を考えさせる。

# 育成方法および指導上の留意点

1. この章では、初めてC言語のプログラミングを経験するので、十分時間を 取って、全学習者が演習問題を解答できるように配慮する。

- 2. この章での説明は必要最小限のものに押さえて、C言語で四則演算のプログラムが組めるようになることを第一として指導する。
- 3. この章では、講義中でもなるべくコンピュータに触れさせるように努める。 それができない場合には、少なくとも学習者にプログラムの動作を確認させ られるような設備を用意するとよい。
- 4. 制御系の技術者を育てる場合には、データ型のビット幅やデータの表現方法といったことを、ハードウェアの知識と併せて説明するように心がける。
- 5. この章での演習問題は、比較的易しいものを多く出題し、自信を付けさせることを第一とする。単純な四則演算をし、出力するといった問題でよい。
- 6. int などのデータ型では、利用するコンピュータの性能(16ビット、32ビット)や基本ソフト(MS-DOS, UNIX)で、取り扱いが異なるので、コメントしておく必要がある。
- 7. 関数printfの左詰め、右詰め、印字幅指定子といた書式制御について、この章のどこかで説明しておくとよい.
- 8. 演算子の優先順位について、必ずどこかで触れておく。

# 用語

定数、変数、データ型、int , long、short , float , double、浮動小数点、接尾子, L , F , 2項演算子、+, -, \*, /, %、優先順位、プリプロセッサ、#include、標準ライブラリ関数、ヘッダ・ファイル、stdio.h , printf()、書式文字列、書式制御、左詰め、右詰め、印字幅指定子、%d、%e、%f、キャスト

### 第3章 選択型のプログラム

### 学習目標

学習者がこの章を修了した時点で、選択型のプログラムに関連する次の事柄ができるようになることを目標とする。

- 1. 与えられた問題の条件を関係演算子を使って表現する。当然、関係演算子を組み合わせての表現もできること。
- 2. 与えられた問題での場合分けに対して、どの選択型の制御文をどのよう に適応すればよいかが考えられる。
- 3. 字下げを行って、読みやすいプログラムを実践する。
- 4. 構造化プログラミングの考え方を実践したプログラムが書ける。

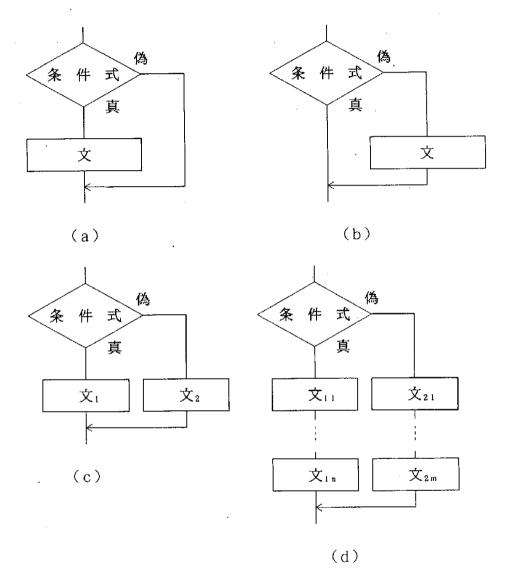
# 内 容

## 3.1 if文を使った選択型のプログラム(210分)

### (1) 基本的な分岐

たとえば、奇数か偶数の判断といったサンプル・プログラムを使って、まずプログラムの分岐と、if文の使い方について説明する。このとき、条件式の値が0のとき偽であり、それ以外の値のときは真であることを併せて説明しておく。if文の説明には流れ図を使って視覚的な対応づけをするとよい。また、ここで使うサンプル・プログラムとしては、図表11-3-1の4つのパターンを示すとよい。特に(d)のパターンについては、複文の説明をする。また、サンプル・プログラムでは字下げを実施する。また、最後にif文と対比して3項演算(たとえば、z = (a > b)? a : b; z = (a > b) といったもの)を説明しておく。ところで、ここで取り扱う図表11-3-1のような選択型のプログラムは、構造化プログラミングの基本パターンであるから、学習者に構造化の意義を説明し、理解させておく必要がある。

図表11-3-1 「選択型の基本」



# (2) 条件の組合せ

ここでは複雑な条件を表現できるように指導する。サンプル・プログラムとしては、if文が二重になったものや、else-if文を使ったものを示す。やはり、ここでも流れ図を使って説明するとよい。また、それらのサンプル・プログラムと対比して、関係演算子と論理演算子を組合せた条件式に

よる表現方法を説明するとよい。特に、if文を二重にして表現した条件は、 論理演算子のAND(&&)と同様であるといったように、できるだけ論理 をベースにして解説する。また、ここでも構造化プログラミングについて 触れておく。

### 3. 2 switch文を使った選択型のプログラム(60分)

#### (1) switch文の基本例

直接switch文を使ったサンプル・プログラムを示す前に、if文を多重に しているサンプル・プログラムを示し、それがswitch文に置き換えられる ことを説明するとよい。そのことで、どんな場合にswitch文が適用できる かを伝えることができる。ここでは、

```
switch(条件式) {
    case 定数式:
    文;
    inherent break;
}
```

という基本的な使い方をしっかりと学習させる.

#### (2) switch文の応用例

ここでは、サンプル・プログラムを使ってbreak のあるなしの違いや、caseを書き並べる場合や、default を使った場合について説明する。

# 3.3 goto文を使ったプログラム(30分)

### (1) if文との組合せ

ここではgoto文がどのようなものかを説明する。サンプル・プログラムでは、if文と組み合わせて利用し、プログラムの1部を飛び越すことができることを示すとよい。ただし、構造化プログラミングの考え方を説明し、goto文を多用しないことを強調する。たとえば、switch文と同様の処理をif文とgoto文の組み合わせでプログラムしたものと、プログラムの読みづらさをswitch文のプログラムと対比して説明する。

#### 3.4 演習問題(480分)

(1) if文の演習

ここでは、if文を理解させる簡単なものから、else-if 文を使った複雑な条件を表現するものまで多くの問題を解答させる。特に、条件が組み合わされた場合の演習は、十分時間を取って色々な問題を解かせるようにする。題材としては、数値の大きさを分類するものなどが考えられる。

(2) switch文の演習

ここでの例としては、割り算の余りの違いで演算を変化させるプログラムや、おつりの払方を求めるプログラムなどが考えられる。

### 育成方法および指導上の留意点

- 1. この章のサンプル・プログラムで、まんべんなく関係演算子と論理演算子を取り上げるように心がける。
- 2. 字下げに関しては、各学校や企業内で統一的なものを決めてから指導に当たるとよい。特に、if文が組み合わされた場合、プログラムが読みづらくなるので配慮を要する。

字下げの例としては,

といった書き方もある.

- 3. 選択型の制御文では、同じ条件でも色々なプログラムの書き方が考えられるので、色々な表現方法を考えさせるように指導するとよい。
- 4. 関係演算子と論理演算子が組み合わされた条件式の場合、たとえば、

$$x <= 100 \&\& x >= 50$$

といった場合、

$$(x \le 100) \&\& (x > = 50)$$

のように括弧を付けた方が分かりやすくなるであろう。

- 5. この章での演習問題は色々な条件を扱った問題を多く出題し、条件をプログラムで表現することに十分なれさせる。
- 6. goto文の取り扱いに関する説明には注意が必要である。学習者が安易にgoto文を使ないようにする。講習の時間が短ければ、「3.3 goto文を使ったプログラム」を飛ばしてもよいであろう。
- 7. 演算子の優先順位についても、必ずどこかで触れておく。

## 用 語

制御文、選択型、if文、if-else 文、else-if 文、switch文、case、break、default 、goto文、複文、{ } 、条件式、関係演算子、> 、>=、< 、<=、==、!=、論理演算子、! 、 & 、|| (||) 、3項演算子、? 、: 、優先順位、字下げ、構造化プログラミング

### 第4章 反復型のプログラム

### 学習目標

学習者がこの章を修了した時点で、反復型のプログラムに関連した次の事柄ができるようになることを目標とする。

- 1. do-while文, while 文, for 文の違いを説明する。
- 2. 繰り返し処理を、do-while文, while 文, for 文の各制御文を使って表現する。
- 3. 繰り返し処理に対して、do-while文, while 文, for 文のどの制御文を使って表現するのが最適かを判断する。
- 4. 繰り返し処理におけるカウンタの働きと、数値の変化を説明する。
- 5. たとえば、++a とa++ の違いを説明する。
- 6. たとえば、繰り返し処理内に"a += b"という文があったとき、変数 a の変化を説明する。
- 7. 繰り返し処理内のプログラムに対して、字下げを実践する。
- 8. 与えられた問題に対して、繰り返し処理を活用する。
- 9. 選択型と併せて構造化プログラミングを実践する。

# 内 容

# 4. 1 do-while文を使った反復型のプログラム(120分)

(1) 100になるまでの繰り返し

サンプル・プログラムによりdo-while文を説明する。流れ図を使って視覚化した説明が重要である。サンプル・プログラムとして、たとえば"a = a + b"をdo-while文で繰り返し、変数 a が100以上になったら終了するプログラムといった簡単なものでよい。変数 a の変化もしっかりと理解させる必要がある。また、字下げについてもサンプル・プログラムで示す。ただし、ここではまだ代入演算子を適用しない。次に、たとえば10回といったように、決められた回数だけ繰り返しを行う方法を考えさせ、サン

プルプログラムを示す。ここで、カウンタというものの考え方を定着させる。

#### (2) n!を求める

do-while文を使って10!を求めるプログラムを考えさせる。そのプログラムを解説後、カウンタに単項演算子を適用したプログラムを示し、単項演算しについて説明する。ここでの説明は、++a , --a , ++a <= 100といった使い方にとどめる。

#### 4.2 while 文を使った反復型のプログラム(90分)

(1) 1~100までのカウント

たとえば、 $1 \sim 100$ までカウンタの変化を出力するといった簡単なサンプル・プログラムを使ってdo-while文とwhile文の比較を示し、その違いを説明する。ここでも、流れ図により視覚化すると有効である。次に、単項演算子++a とa++ の違いをサンプル・プログラムにより示す。たとえば、b = ++a "と"c = a++ "を繰り返し、変数 $b \geq c$ の変化の違いを見せるといったサンプル・プログラムが考えられる。

## 4.3 for 文を使った反復型のプログラム(90分)

(1) Σnを求める

たとえば、 $\Sigma$ nを求めるプログラムをdo-while文とwhile 文を使って作成させる。そして、同様のプログラムをfor 文を使ったもので示す。

do-while文, while 文との違いからfor 文の特徴と使い方を説明する。 また、代入演算子についても、ここで説明する。

## 4.4 演習問題(600分)

(1) 繰り返し処理に慣れる

ここでは、do-while 文、while 文、for 文と単項演算子、代入演算子、カウンタに慣れさせるための簡単な演習問題を数多く行わせる。その例としては、サンプル・プログラムで使った、 $\Sigma n$  やn ! を求めるプログラムを違った値で行うような問題や、  $1\sim100$ までの偶数の和を求めるといった

問題でよい。

(2) 選択型と反復型の演習

選択型と反復型を組み合わせた問題を練習させる。たとえば、1つの反復型のプログラムの中で、偶数と奇数の和を求めさせる問題や、与えられた値が素数であるかを判別するといった問題が考えられる。

# 育成方法および指導上の留意点

- 1. do-while文, while 文, for 文という3つの反復型の制御文があるので、 混乱しないような指導が必要である。また、この3つの制御文が用意されて いる意味を、アルゴリズムと対応づけて認識させることも必要である。
- 2. 繰り返し処理でのカウンタの働きは、この後の配列やポインタを使った処理と密接に関わってくるので、具体的にカウンタの変化を学習者に追わせるような指導をするとよい。
- 3. この章では、単項演算子と代入演算子の学習を行うので、学習者が混乱しないように配慮する。まず、普通の代入文による表現を確実なものにしてから、順にステップ・アップをはかるような講義の展開とする。
- 4. 単項演算子と代入演算子では、特に"++a","a++","a+=1"の取り扱いで学習者が混乱する可能性がある。これらの説明には、サンプル・プログラムで取り上げ、実際に学習者に確認させるようなことをするとよい。
- 5. 演算子の優先順位について、必ずどこかで触れておく。
- 6. 演習問題では十分時間を取って、講義の内容の定着をはかる。特に、「(1) 繰り返し処理に慣れる」では講義中のサンプル・プログラム程度の問題を多 く練習させる。また、「(2)選択型と反復型の演習」では第3章との知識の複 合をはかるので、各学習者へのきめ細かな指導が必要であろう。
- 7. 繰り返しの条件は、第3章の条件式で学習済みであるが、だからといって すぐに活用するのは問題がある。やはり、復習させるような講義展開にすべ きである。
- 8. 第3章の選択型と同様に、反復型も構造化プログラミングを実践する上で 重要な技術である。よって、do-while文, while 文, for 文ともに十分時間

を取って学習させ、定着を図るように指導する必要がある。また、おりに触れて構造化を強調する必要もある。

# 用 語

反復型, do-while文, while 文, for 文, カウンタ, 字下げ, 単項演算子, ++a, --a, a++, a--, 代入演算子, a += b, a -= b, a \*= b, a /= b, a %= b

#### 第5章 ビット演算

### 学習目標

学習者がこの章を修了した時点で、ビット演算に関連した次の事柄ができるようになることを目標とする。

- 1. 論理演算の結果を求める。
- 2. シフト演算後のビット情報を求める。
- 3. 符号付きと符号無し整数データの違いを説明する。
- 4. マスク処理の意義を具体的に説明する。
- 5. 論理演算を使って、特定のビット情報を取り出すといったプログラムを 書く。
- 6. シフト演算を使って、かけ算や割り算を行うといったプログラムを書く。
- 7. 2の補数表現や浮動小数点の表現が、どの様なビットの情報で表されるかを説明する。

### 内 容

# 5.1 ビット演算の基本(120分)

(1) マスク処理

ここでは、ビット単位の論理演算について説明する。よって、ハードウェアの内容を復習するような講義展開にすべきである。サンプル・プログラムでは、たとえば任意の数値に対して16進数のFFFFとAND、OR、XOR、NOTの演算を施し、その結果を出力するといったものが考えられる。そのプログラムの説明では、演算の結果に対してのビット単位での解説と、その意味を述べる必要がある。特に、XORとNOTの結果が同じになることには注目させるとよい。マスクの概念についても、できれば触れておきたい。また、ここでは、unsignedを使ったデータの宣言についても解説する。

#### (2) ビット操作によるかけ算、割り算

ここでは、シフト演算について説明する。シフト演算については、2倍、4倍、 $\cdots$ , 2分の1倍、4分の1倍、 $\cdots$ となることと、特定の位置のビットを取り出すのに役立つといった意味を説明しておく必要がある。サンプル・プログラムとしては、任意の値を10倍するプログラムや、10分の1倍するプログラムが考えられる。また、たとえば第5ビット目の情報が1か0かを調べるために、85ビットを符号位置まで左シフトし、その結果が正負であるかによって判別するといったプログラムが考えられる。さらには、同様のことをマスクを使った処理でもできることを説明すると良い。

### 5.2 データの表現方法を調べる(90分)

#### (1) 2の補数を調べる

任意の負の整数に対して、そのビット情報がどうなっているかを取り出すというサンプル・プログラムを取り上げる。その解説では、論理演算ややシフト演算の復習と、ハードウェアで学んだ知識の復習を兼ねるとよい。また、符号有りと符号無しのデータ型の数値の表現できる範囲を説明しておくとよい。

#### (2) 浮動小数点を調べる

「(1) 2の補数を調べる」と同様に、浮動小数点のビット情報を出力するプログラムを取り上げ、浮動小数点のデータ表現を復習する。ただし、浮動小数点のビット情報は共用体を利用する必要があるので、第14章の「14.2 共用体」で取り扱ってもよい。

### 5.3 演習問題(480分)

#### (1) 論理演算の演習

特定のビット情報を取り出すとか、特定のビットを 0 や 1 にするといった問題が考えられる。また、論理演算の組合せで、足し算をさせるといった問題もハードウェアの復習となってよい。

### (2) シフト演算の演習

2つの任意の数値に対して、かけ算や割り算を行わせる問題が考えられる。数字の文字コード(ただし、文字コードについては特定の値を与える)をデータとして与え、それをパックするといった問題や、さらに、パック10進数を2進数に変換するといった問題も考えられる。

## 育成方法および指導上の留意点

- 1. この章は、C言語の特徴的な内容の1つである。しかし、この章を講義するためには、ハードウェアの若干の知識が必要である。よって、講義の展開ではハードウェアの内容を復習させるような場面を設ける必要がある。また、C言語の初歩を目的とする場合は、飛ばしてもよいであろう。
- 2. 「(1) マスク処理」の解説で、対象が制御系の学習者の場合は、マスク処理に関する具体的な応用例を話題として取り上げるべきである。また、アセンブラ言語を既に学習済みである場合には、それと対応づけて解説するのもよい。
- 3. この章では、論理演算に関する代入演算子についても、どこかで説明しておく。
- 4. 対象者が「5.1 ビット演算の基本」やそれ以前の学習に熟知しているようであれば、「5.2 データの表現方法を調べる」の内容は演習として与えてもよい。
- 5. 論理演算子と条件演算子を混同する恐れがあるので、講義中に注意を促す必要がある。
- 6. 演習問題の内容は対象者の前提学習を考慮にいれ、演習問題のレベルを考える必要がある。特に、制御系の対象者の場合は、ここでの知識が活用できるように、十分時間を取って定着をはかる必要がある。
- 7. 演算子の優先順位について、必ずどこかで触れておく。
- 8. 2の補数や浮動小数点の表現については、シフト演算を施さなくても、書 式制御文字%xを利用すれば、16進数で取り出せることも説明しておくとよい。

- 9. 右シフト演算の場合、処理系によって符号情報を引きずる場合と引きずらない場合がある。よって、その点も注意して説明しておくと良い。
- 10. OS内部のステータスの保持・変更などでは、ビット演算を多用する必要がある。そのため、ビット演算に重点を置いたOJTとしては、OSの開発・保守などが考えられる。

# 用 語

符号無しのデー型, unsigned, 論理演算, AND, OR, XOR(BOR), NOT, マスク、論理演算子, &, | (|), ^,!, 2 進数, 8 進数, 16進数, ‰, ‰x, 左シフト, 右シフト, 論理演算子, <<, >>, 代入演算子, &=, |=(|=), ^=, !=, <<=, >>=, 2 の補数, 浮動小数点, 仮数部, 指数部, イクセス 6 4

### 第6章 入出力処理

### 学習目標

学習者がこの章を修了した時点で、入出力処理に関連した次の事柄ができる ようになることを目標とする。

- 1. scanf 関数を使って、色々なタイプのデータを入力する。
- 2. ニュートン法をプログラムで表現する。
- 3. アドレス演算子の意味を説明する また、プログラム中で利用する。
- 4. ポインタ変数の意味を説明する。また、プログラム中で利用する。ただし、ここで取り扱うポインタ変数は整数型と実数型の変数に対するものに限る。

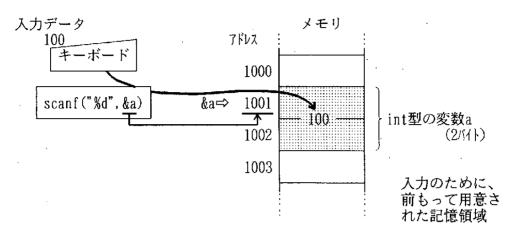
## 内 容

### 6.1 整数の入力(120分)

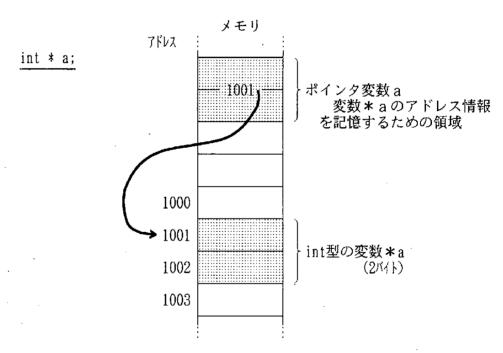
(1) 10人分の点数合計

10人分の点数を(キーボードより)入力し、合計するサンプル・プログラムを示す。ここでは、scanf 関数の利用方法についての理解を図ることが目的である。scanf 関数の書式文字列に関しては、printf関数での学習と対応づけて行う。また、scanf 関数の引数がアドレスであることを説明する。特に、アドレス演算子の意味については図解(図11-6-1参照)など使って説明する。また、アドレス演算子をポインタ変数に置き換えた場合の、サンプル・プログラムも示し、ポインタ変数とポインタ演算子について説明する。ここでも図解(図11-6-2参照)を使うと有効である。

図表11-6-1 「アドレス情報とアドレス演算子」



図表11-6-2 「ポインタ変数とポインタ演算子」



# (2) n人分の点数合計

(1) のサンプル・プログラムを変更し、繰り返しの回数を入力するサンプル・プログラムを示す。ここで、強調したいことは、繰り返しのカウンタの上限値を入力するという方法である。このことは、処理を一般化する

意味で非常に頻繁に行われる方法なので、あえて時間を裂く必要があるであるう。

#### 6.2 実数の入力(90分)

(1) ニュートン法

ここでは、ニュートン法のサンプル・プログラムを利用して、scanf 関数を使った実数の入力について説明する。ニュートン法のアルゴリズムに関しては、役立つ手法であるので、ここで紹介する。ニュートン法で扱う方程式については、プログラム中でデータとして与えておけばよい。そして、方程式を入力するプログラムについては、演習問題として利用するとよい。また、ニュートン法のような逐次近似というアルゴリズムをプログラムで表現するには、どのような論理で設計すればよいかを示すような指導にこころがける必要がある。

### 6.3 16進数の入力(60分)

(1) 4桁の16進数を10進数に変換 ここでは、4桁の16進数を10進数に変換するサンプルプログラムを示す。 そして、scanf 関数を使った8進数や16進数による入力方法と、入力幅の 指定について説明する。

## 6.4 演習問題(420分)

(1) scanf 関数の演習

ここでは、scanf 関数を熟知することが目的であるから、演習問題としては、ここまでで取り扱った問題をいくつか利用し、データを入力するプログラムに変更させる。たとえば、第5章のビット演算を使ったかけ算のプログラムに対して、2つの数値を入力するプログラムに変更する問題などが考えられる。

#### 育成方法および指導上の留意点

- 1. この章は、scanf 関数を活用できるようにすることを第一に考える。よって、第2章で紹介したprintf関数の場合と同様に、#includeやstdio.h については認識させる程度の説明でよい。
- 2. scanf 関数を利用するために、この章では、初めてアドレス演算子やポインタ変数について言及する。しかし、その範囲は整数型や実数型を対象にした範囲に止め、文字変数には触れない。よって、アドレス演算子やポインタ変数についての説明は、その範囲に止める。ただし、これらの概念は後の学習で重要な意味を持つので、基本をしっかりと踏まえた説明をする必要がある。また、ハードウェアやアセンブラ言語の知識を持たない学習者には、コンピュータの仕組みを踏まえた解説が必要である。
- 3. ニュートン法のアルゴリズムの解説では、グラフを利用して、視覚的に説明する。
- 4. アドレス演算子、ポインタ演算子と論理演算子のAND、算術演算子の乗 算を混乱しないためにも、前者は単項演算子であり、後者は二項演算子であ ることを触れておくとよい。
- 5. ポインタ変数とポインタ演算子で修飾された変数が混乱しやすいので、幾度となく講習の中で確認するとよい。

# 用 語

scanf 関数, 書式制御, 入力幅指定し, アドレス演算子, & , ポインタ変数, ポインタ演算子, \* , ニュートン法

### 第7章 配 列

### 学習目標

学習者がこの章を修了した時点で、配列に関連した次の事柄ができるように なることを目標とする。

- 1. プログラムで配列を活用する。最低でも、1次元配列と2次元配列は使える。
- 2. 漸化式で表された数列をプログラムで表現する。
- 3. 単純挿入ソート等のアルゴリズムを説明する。
- 4. 2行2列の行列に対して、逆行列を求める計算をプログラムで表現する。
- 5. 3元1次の連立方程式の解を、掃き出し法で求める。

# 内 容

## 7.1 1次元配列(180分)

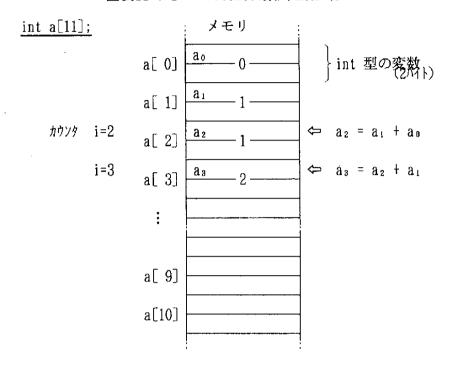
(1) 数列のa。~a」。まで計算

ここでは、数列の計算を例に配列について説明する。サンプル・プログラムとしては、漸化式で与えられた数列のa0~a10の各項を計算し、配列に記録するといったものが考えられる。ここでの数列は、漸化式

 $a_0 = 0$ ,  $a_1 = 1$ ,  $a_n = a_{n+1} + a_{n+2}$  (n=2, 3, ...)

で与えられるFiboncci数列のように、簡単なものでよい。サンプル・プログラムの説明としては、そのプログラムを実行することで、配列の中にどのようにデータが記録させていくかを図解(図表11-7-1参照)などを使って説明するとよい。このとき、配列の添字と繰り返しのカウンタの変化にも注目させる必要がある。

図表11-7-1 「Fiboncci数列と配列」



#### (2) ソート

ここでは、ソートのプログラムを使って1次元配列の学習を深める。ソートのサンプル・プログラムとしては、単純挿入ソートや隣接交換ソートなどが考えられる。対象者の理解力が期待できれば、木ソートやクイック・ソートなどをサンプル・プログラムとして追加してもよいであろう。配列の説明については、プログラム中で1次元配列に初期値を与えるといった方法に触れる。ソートのアルゴリズムの説明では、直接プログラムの説明に入らず、ソートの方法を具体的な数値で説明しておく必要がある。

#### 7.2 2次元配列(180分)

### (1) 逆行列の計算

ここでは、逆行列の計算をするサンプルプログラムを通して、2次元配列について学習させる。2次元配列の説明では、添字に注目させるような説明を行う。逆行列の計算は、たとえば2行2列といった簡単なものでよい。また、scanf 関数で行列にデータを入力することにも触れておく。

#### (2) 掃き出し法

ここでは、掃き出し法のサンプル・プログラムを通じて、2次元配列の 学習を深める。サンプル・プログラムは、3元1次の連立方程式の解を求 める程度のものでよい。掃き出し法の説明としては、プログラムの説明に 入る前に、まず掃き出し法がどのようなものであるかを実際に計算してみ せる必要がある。また、掃き出し法のプログラム説明としては、2重ルー プ内で添字が変化するので、その変化を具体的に説明する必要がある。ま た、配列の説明としては、プログラム中で、2次元配列に初期値を与える 方法について触れる。

### 7.3 演習問題(600分)

#### (1) 数列の演習

たとえば、

 $a_0 = 0$ ,  $a_1 = 1$ ,  $a_n = 2 a_{n-1} + a_{n-2}$  (n=2, 3, …) といった漸化式を与え、プログラムを作成させるといった問題が考えられる。

## (2) ソートの演習

たとえば、サンプル・プログラムとして取り上げたソートを自作させるとか、サンプル・プログラムで取り上げなかったソート(たとえば、隣接交換ソート)をプログラムさせるといった問題が考えられる。ただし、サンプル・プログラムで取り上げなかったソートの問題の場合、その内容の説明とヒントを十分に与える必要がある。

## (3) 行列の演習

たとえば、3行3列の逆行列の計算問題や、行列のかけ算の問題が考えられる。また、サンプルプログラムとして取り上げた掃き出し法のプログラムを自作させるといった問題もよい。

#### 育成方法および指導上の留意点

- 1. C言語の配列要素は添字 0 から始まることに注意させる。
- 2. C言語の学習時間に比較的余裕があり、C言語の学習を通じてアルゴリズムの考え方を定着さることを目標としている場合、ここでのサンプル・プログラムの理解と、サンプル・プログラムに関連する演習問題の実践は、重要な意味を持つ。よって、その学習を深めるように配慮するとよい。特に、帰納法的な発想やソートのアルゴリズムをプログラムで表現するといった能力は重要である。
- 3. ここで取り扱うソートや掃き出し法の考え方は重要であるが、難しい内容でもある。よって、それを適用する場合には、対象者の能力を十分考慮にいれて、嚙み砕く必要がある。
- 4. 2次元配列を取った場合でも、メモリ上ではシーケンシャルな並びである ことを触れておくとよい。
- 5. 配列の添字と繰り返しのカウンタの関係は、講義中に意識して説明することに心がける必要がある。
- 6. 配列の宣言時に初期値を与える場合、配列の宣言でサイズを省略できることについても触れておくとよい。
- 7. ソートや掃き出し法の指導では、その考え方を十分に説明する必要がある。 また、そのアルゴリズムをどのようなプログラムで実現するのが適切である かを、流れ図などを利用して示し、プログラムの論理的な作り方を実践して みせる指導が必要である。

# 用語

配列,添字,配列要素,1次元配列,a[SIZE],a[]={···},2次元配列,a[SIZE][SIZE],a[]={{···},{···},···},数列,漸化式,ソート(分類,整列),単純挿入ソート,行列,逆行列,掃き出し法

### 第8章 文字処理

## 学習目標

学習者がこの章を修了した時点で、文字処理に関連した次の事柄ができるようになることを目標とする。

- 1. 文字と文字列の取り扱いの違いを説明する。
- 2. printf関数やputchar 関数、puts関数を使って文字や文字列を出力する。
- 3. scanf 関数やgetchar 関数、gets関数を使って文字や文字列を入力する。
- 4. ASCIIコードの意味を説明する。
- 5. 配列に文字列データが記録された状態を説明する。
- 6. 配列が取られているメモリ上の番地を取り出す。
- 7. 制御文字の種類と意味を説明する。

# 內 容

# 8.1 文字の入出力(120分)

(1) 文字の出力

初期値として与えた文字(1文字)データを出力するサンプル・プログラムを通じて、文字変数、文字定数の取り扱いについて解説する。出力に関しては、printf関数を使った場合と、putchar 関数を使った場合について示す。

- (2) 文字の入力
  - (1) のサンプル・プログラムを変更し、文字データを入力するようにしたプログラムを示す。このプログラムを通して、scanf 関数とgetchar 関数による文字データの入力方法について説明する。
- (3) ASCII⊐-F

文字データを取り扱ったプログラムの応用として、ASCII コードを出力するプログラムを示す。そのプログラムは、図表11-8-1のように出力するものとし、その説明では文字変数の値を16進数として取り扱えることに触

れておく。また、ASCII コード自身についても、その意味について解説を加えておくとよい。

#### 図表11-8-1 「ASCIIコードの出力」

```
!=21
           "=22
                 #=23 $=24 %=25
 =20
                                  &=26 '=27
(=28)
     )=29 *=2a
                 +=2b, =2c
                            -=2d
                                  .=2e
                                         /=2f
                                  6=36
0 = 30
     1=31
           2=32
                 3=33 4=34
                             5=35
                                         7 = 37
                 ;=3b <=3c
8=38
     9=39
           :=3a
                             ==3e
                                  >=3d
                                         ?=3f
@=40
     A=41
           B=42
                 C=43 D=44
                            E=45
                                  F=46
H = 48
     I=49 J=4a
                 K=4b L=4c
                             M=4d
                                  N=4e
                                         0=4f
P = 50
     Q=51
           R=52
                 S=53 T=54 U=55
                                  V=56
X = 58
     Y=59 Z=5a
                 [=5b \quad Y=5c]
                            ]=5d
                                   ^ =5e =5f
                 c=63 d=64 e=66
=60
     a=61
           b=62
                                  f=66 g=67
     i=69 j=6a
                 k=6b 1=6c m=6d
                                  n=6e o=6f
h=68
p=70 q=71 r=72
                 s=73 t=74 u=75
                                  v=76
                                         w = 77
                 \{=7b \mid =7c \} =7d
x=78 v=79 z=7a
                                  =7e
```

#### 8.2 文字列の入出力(180分)

### (1) 文字列の出力

ここでは、初期値として与えた文字列データを出力するサンプル・プログラムを通して、配列と文字列定数について解説する。そのプログラムの説明では、文字列データの初期値の与え方について触れておく。また、文字列データの最後にはヌル文字(〈①)が付くことも触れておく。また、配列に初期値として与えた文字データがどの様に記憶されるかを図解(図表11-8-2参照)で示しておくとよい。文字列の出力については、printf関数とputs関数を紹介する。

図表11-8-2 「文字列定数と配列」

<pre>char a[]="CAIT";</pre>	;	メモリ	!
í	a[ 0]	С	i char型の変数(1所件)
;	a[ 1]	A	
;	a[ 2]	I	
	a[ 3]	Т	
:	a[ 4]	0	
			]

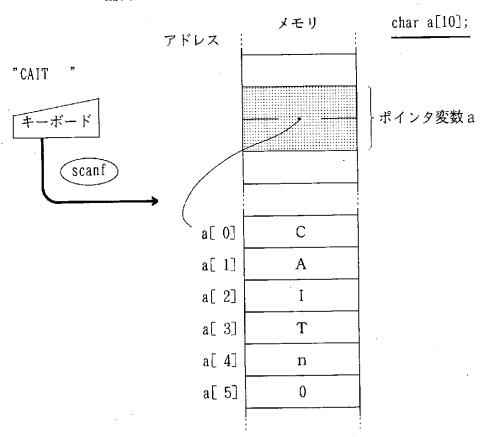
#### (2) 文字列の入力

(1) のサンプル・プログラムを変更して、文字列データを入力するプログラムを示す。ここでは、配列のアドレスについて説明する。その説明は図解(図表11-8-3参照)でおこなうとよい。また、入力された文字列の場合は、改行(<n)も入力されることを触れておく。文字列の入力については、scanf 関数とgets関数を紹介する。

## (3) 文字列のコピー

文字列データを取り扱った応用として、文字列をコピーするサンプル・プログラムを示す。ここでは、文字列の入出力と、文字列と配列の取り扱いを再度確認しながら、その学習を深める。文字列コピーのプログラムとしては色々考えられるが、図表11-8-4に示すようなプログラムを使うとよい。

図表11-8-3 「配列のアドレス」



図表11-8-4 「文字列コピーのプログラム」

```
char st[10], sf[10];
int i;

i=0;
while((st[i] = sf[i]) != '\0') {
    i++;
}
:
```

#### 8.3 演習問題(480分)

(1) 文字操作

たとえば、サンプル・プログラムで取り上げたASCIIコードの表を出力するプログラムの自作や、入力した文字のASCIIコードを出力するプログラムの作成といった問題が考えられる。

### (2) 文字列操作

たとえば、文字列を反転させて配列に記憶するプログラムや、文字列の特定の部分を取り出すプログラムや、タブ文字(〈t)をスペースに置き換えるプログラムといったプログラム作成問題が考えられる。

### 育成方法および指導上の留意点

- この章では、配列を使った文字列の取り扱いをしっかりと学習させる。この後で学習するポインタを使った文字列の取り扱いと混乱しないためにも、ここでの学習が重要である。
- 2. ヌル文字の説明では、C言語の場合、\0 という特別の文字定数が与えられていることと、これをC言語のコンパイラでは1文字として扱うことも言及しておく。また、先に学習済みである\n と併せて、エスケープ文字(\)を使って表現する文字定数(エスケープ系列)を紹介しておくとよい。特に、エスケープ系列に含まれる制御文字については、その意味も解説しておく。
- 3. 配列を宣言すると、コンパイル時にその領域が確保されることをしっかりと認識させる。ただし、ローカルな自動変数の場合は呼び出し時に領域が取られることを教師は念頭におく必要である。
- 4. 配列を宣言すると、その配列名は配列の先頭番地を表すことを認識させる。 ただし、配列名はポインタ変数とは異なる役割(配列名は定数として考える ので、配列名を直接インクリメントするような演算はできない)を持ってい るので、学習者が後の授業で混乱しないような配慮の元に解説する必要があ る。
- 5. 3.、4.は後に学習するポインタ変数との混乱を避けるために重要なことであるから、しっかりと認識して指導に当たるべきである。

6. 文字列を配列に初期値として代入した場合と、scanf 関数等でキーボード より入力した場合には、改行文字の有無の違いが生じることも確認しておく。 その差は、CRT等に出力した場合に起こり、学習者が知らないでいると誤 解を招く恐れがある。

### 用語

文字, '文字',文字列, "文字列", a[]="文字列",printf関数,putchar 関数,puts関数,scanf 関数,getchar 関数,gets関数,書式制御文字,%c,%s,ASCIIコード,配列のアドレス,配列名,エスケープ文字,\(\formalfont{\substack}),制御文字,改行,\n(\formalfont{\substack}n), ヌル文字,\0(\formalfont{\substack}n), タブ,\t(\formalfont{\formalfont{\substack}n})

### 第9章 ポインタ変数

### 学習目標

学習者がこの章を修了した時点で、ポインタ変数と配列に関連した次の事柄ができるようになることを目標とする。

- 1. ポインタ変数の役割を説明する。
- 2. ポインタ変数を使って配列の内容を取り出す。
- 3. ポインタ変数と1次元配列の違いを説明する。
- 4. ポインタ変数の配列と2次元配列の違いを説明する。
- 5. ポインタ変数及びポインタ変数の配列に、初期値を与える。また、その 初期値を取り出す。
- ポインタ変数と文字列データが、メモリ上でどの様な関係にあるかをイメージし、説明する。
- 7. ポインタ変数の配列と各文字列データが、メモリ上でどの様な関係にあるかをイメージし、説明する。

# 内 容

## 9.1 ポインタ変数(60分)

(1) ポインタ変数と変数

ここでは、ポインタ変数にアドレスを代入するということの意味について学習する。ここでのサンプル・プログラムとしては、図表11-9-1のようにポインタ変数が変数のインデックシングをする役割のものであることを、学習者に明確に伝えるプログラムを示す。特に、ポインタ変数とポインタ変数が指す変数を図解で示しながら、変数の内容の変化について説明する。ポインタ変数を利用してアドレスを知る方法は、ちょうど、ハードウェアで学習した間接アドレッシングであることにも触れるとよい。

# 図表11-9-1 「ポインタ変数を説明するためのプログラム」

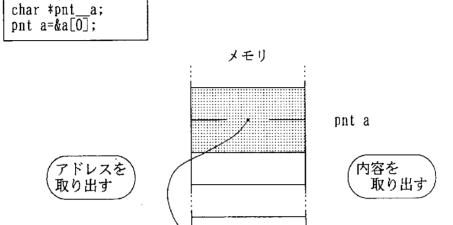
```
int a, b;
int *i_pnt;
ia = 10;
b = 100;
i_pnt = &a;
printf("a = %d , b = %d , *i_pnt = %d \n", a, b, *i __pnt);
a = 20;
printf("a = %d , b = %d , *i_pnt = %d \n", a, b, *i __pnt);
*i_pnt = 200;
printf("a = %d , b = %d , *i_pnt = %d \n", a, b, *i __pnt);
i_pnt = &b;
printf("a = %d , b = %d , *i_pnt = %d \n", a, b, *i __pnt);
i_pnt = &b;
printf("a = %d , b = %d , *i_pnt = %d \n", a, b, *i __pnt);
i:
```

#### 9.2 ポインタ変数と配列(240分)

#### (1) ポインタ変数と1次元配列

ここでは、1次元配列の先頭番地をポインタ変数に代入し、ポインタ変数の値を変化させることで1次元配列の要素を自由に取り出すといったサンプル・プログラムを示す。そのプログラムにより、ポインタ変数により配列を操作できることを解説する。そのときの説明には図解(図表11-9-2参照)を使うとよい。さらに、配列のラベルについても説明しておくとよい。それは、例えば配列名がaの場合,aとa[0]が同様のものであるということである。

図表11-9-2 「ポインタ変数と配列の関係」



	\	·		
pnt_a+0	&a[ 0]	С	a[ 0]	*(pnt_a+0)
pnta+1	&a[ 1]	A	a[ 1]	*(pnta+1)
pnta+2	&a[2]	I	a[2]	*(pnta+2)
pnta+3	&a[ 3]	Т	a[3]	*(pnt_a+3)
pnta+4	&a[ 4]	0	a[4]	*(pnt_a+4)

## (2) ポインタ変数と文字列

char a[ ]="CAIT";

ここでは、ポインタ変数に文字列データを初期値として与え、そのデータ中の各文字を配列要素のように取り出すサンプル・プログラムを示す。そのプログラムの説明では、ポインタ変数と文字列データの関係が、ちょうど配列のようなデータ構造になっていることを説明する。ただし、このプログラムを説明すると、見かけ上、配列を利用した場合と良く似ているので。学習者が配列と同様なものであると誤解する恐れがある。よって、ポインタ変数だけを用意し、scanf 等の関数でそのポインタ変数に文字列を入力して、サンプル・プログラムと同様な扱いをすることは許されないことを説明する。ポインタに初期値を与えられる場合には、文字列データを記録する領域がコンパイル時に取られる。しかし、実行時に入力する場

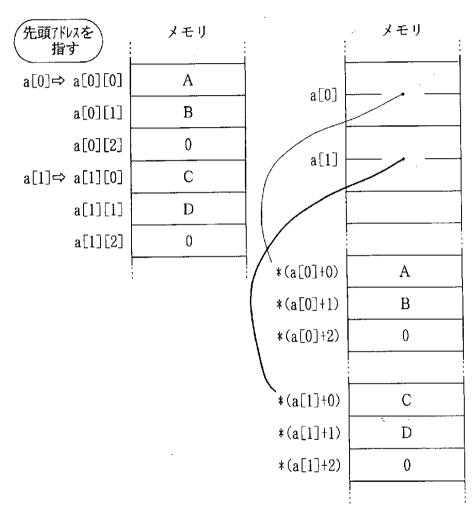
場合は、ポインタ変数に初期値が与えられておらず、また、文字列データを記録する領域も取られないことを解説する。このことは、配列とポインタ変数の大きな違いであるので、学習者が誤解しないように明確にしておく必要がある。

#### (3) ポインタ変数と2次元配列

ここでは、ポインタ変数の配列と2次元配列のデータ構造に関するそれぞれの特性を対比しながら示す。そのためのサンプル・プログラムとしては、ポインタ変数の配列および2次元配列に初期値として複数の文字列データを与え、そのデータを1文字づつ取り出したり、各文字列データ毎にに取り出したりという操作を行うプログラムを示すとよい。その説明のときには、図解(図11-9-3参照)を利用しメモリ上でのイメージを与えるとよい。また、「(2) ポインタ変数と文字列」と同様に、ポインタ変数の配列の場合も初期値を与えないと、ポインタ変数の内容が不定になり、データを記録する領域も確保さないことに注意させる。

図表11-9-3 「ポインタ変数の配列と2次元配列」





# 9.3 演習問題(480分)

## (1) ポインタ変数の演習

ここでは、ポインタ変数の役割を確認し、活用する能力を養うための演習を行う。たとえば、ポインタ変数に1次元配列の先頭番地を代入し、ポインタ変数の値をインクリメンとしながら順に取り出すプログラムの作成問題が考えられる。また、その問題を拡張して、データはscanf 等の関数

を使って行うといった問題も考えられる。また、MS-DOS上の処理系ではポインタ変数に特定の数値を与え、その番地にある内容を取り出すといった問題も考えられる。

(2) ポインタ変数の配列の演習

ここでは、ポインタ変数の配列を活用する能力を養うための演習を行う。 たとえば、2次元配列の各列の先頭番地をポインタ変数の配列要素に代入 し、ポインタ変数の配列を使って、2次元配列のデータを取り出すといっ たプログラム作成問題が考えられる。また、その問題を拡張して、データ の入力はscanf 等の関数を使って行うといった問題も考えられる。

## 育成方法および指導上の留意点

- 1. ポインタ変数だけを用意し、scanf 等の関数でそのポインタ変数に文字列を入力するプログラムを作ると、多くの処理系ではwarning が出ることを実演して見せ、その原因を考えさせると効果的である。
- 2. 1. の原因を全学習者に理解させた後、配列を利用してwarning を出さないで文字列データを入力するプログラムを見せ、解説すると効果的である。また、そのプログラムを学習者に作らせてもよい。
- 3. ポインタ変数の配列と2次元配列の説明では、
  - ・ポインタ変数の配列を取ると、記録する各文字列の長さが違っていても気にすることなく利用でき、かつメモリが有効に使える
  - ・ 2 次元配列では、配列のサイズを文字列データの最大のものに合わ せなければいけない

といったことを比較しながら解説する。

- 4. 学習対象者の理解が高いようであれば、ポインタ変数のポインタ変数について触れることが望ましい。ただし、学習対象者の理解が高くない場合には、 混乱を招く恐れがあるので説明を避ける。
- 5. ポインタ変数については、FORTRANやCOBOLといった高水準言語を学習済みの者にとっても、馴染みのないデータ構造であるので、十分解説に時間を取って、しっかりと理解させる必要がある。ポインタ変数は、便

利なものであるが、非常に理解しづらいものなので注意を要する。

- 6. この章で、ポインタ変数を全学習対象者に理解させておかないと、次の関数のデータ引渡しでさらに困難をきたすので、その点を十分認識して指導に当たる。
- 7. 学習対象者がポインタ変数になれないうちは、データを記録する領域は配列で取り、その操作をポインタ変数で行うといった、役割の切り分けをさせることも1つの指導方法として考えられる。

# 用 語

ポインタ変数, \*a, \*a = "文字列", \*(a+i), 間接アドレッシング, ポインタ変数の配列, \*a[], \*a[] = {"文字列1","文字列2",…}, \*(a[i]+j), 2次元配列の各列の先頭番地, a[添字]

## 第10章 関 数

## 学習目標

学習者がこの章を修了した時点で、関数に関連した次の事柄ができるように なることを目標とする。

- 1. main関数以外の関数を使ったプログラムを書く。
- 2. return文で値を返す。
- ポインタを引数としたプログラムを書く。
- 4. 文字列を操作する関数を作る。
- 5. グローバル変数とローカル変数の違いを説明する。

### 内容

#### 10.1 関数の作り方(60分)

(1) 変数で数値を受け渡す

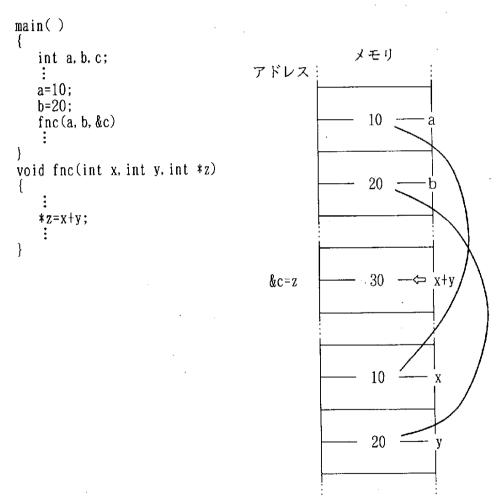
ここでは、一部の処理を別の関数として作ることを学習する。サンプル・プログラムとしては、main関数から整数型の2変数を渡して、その値を加算して結果を返すといった簡単なものでよい。このプログラムで説明しておくことは、関数の宣言の仕方と関数の呼び出し方である。特に、関数の宣言では、関数にもデータ型があり、そのデータ型を宣言してなければならないということである。また、関数が値を返すときにはreturn文を使い、返す値によって関数のデータ型が決まることを解説する。また、返す値がないときには、void型を宣言することにも触れる。

#### 10.2 ポインタ変数の引数(180分)

(1) ポインタで数値を受け渡す

ここでは、整数型の2変数をmain関数から受け取り、加算した結果と乗 算した結果を返すといった簡単なサンプル・プログラムを示す。このプロ グラムで説明することは、加算結果と乗算結果を返すために、それらの結 果を記録する場所(アドレス)をmain関数から与えることである。そして、関数で受け取る引き数は、ポインタ演算子の付いた変数であることに注意させる。この説明では、図解(図表11-10-1 参照)を用いて、ポインタで受け渡すということがどういうことであるかを、明確に伝える必要がある。また、ポインタで受け渡す場合、関数の呼ばれる前にあった情報は消えてしまうことにも注意させる。ここでは、"call by value "と" call by address "の違いを説明する。

図表11-10-1 「関数間のポインタでのやり取り」



#### (2) 文字列を受け渡す

ここでは、2つの文字列をmain関数から受け取り、それらを合わせて1つの文字列にして返すといったサンプル・プログラムを示す。このプログラムでは配列のアドレスを引数として使い、配列のアドレスでやり取りする意味を説明する。また、サンプルプログラムを変更し、main関数の中で2つの文字列をポインタ変数に初期値として与え、ポインタ変数を関数に渡すサンプル・プログラムを示してもよい。ただし、関数から受け取る文字列に関しては、配列で受け取らなければいけないことに触れる。このことは、先に学習した配列とポインタ変数の違いから類推すれば分かるので質問して考えさせるような指導もよいであろう。

#### 10.3 共通に使う変数(30分)

#### (1) グローバル変数

ここでは、関数の外で定義するグローバル変数について説明する。そのサンプル・プログラムでは、「10.1関数の作り方」で示したプログラムを変更し、引数として扱っていた2つの整数型の変数をグローバル変数とする。グローバル変数を説明するとき、必ずローカル変数(関数内で宣言した変数)と比較し、その変数の有効領域について明確にすることが重要である。

#### 10.4 演習問題(600分)

(1) return文を使った関数の演習

たとえば、2つの引数の最大公約数を求めて返すといった関数を作る問題が考えられる。また、値を返さない関数についても演習する。たとえば、1つの引数を受け取り奇数か偶数かを判断し、その結果を出力するといった関数を作る問題が考えられる。

(2) ポインタを使って数値を返す関数の演習

たとえば、2つの引数の最大公約数と最小公倍数を求めて返すといった 関数を作る問題が考えられる。また、応用問題として、配列内の点数を合 計するといった関数を作る問題も考えられる。 (3) ポインタを使って文字列を返す関数の演習 たとえば、2つの文字列の内容を入れ換えて返す関数を作る問題が考え られる。

#### 育成方法および指導上の留意点

- 1. この章では、関数を作ることを学ぶ。そして、その学習を通じて、関数を作る意義、すなわち、構造化プログラミングを意識して関数を作ることを知らせる必要がある。
- 2. 「10.1関数の作り方」では、return文を使って関数から返せる値は1つであることを説明する。そして、複数の値を返したいときには引数をポインタにして使うとよいことを説明し、「10.2ポインタの引数」に話をつなげる。そうすることで、講義の展開がスムーズに行く。
- 3. 学習対象者の理解が高いようであれば、ポインタ型の関数(データ型 \*関 数名(…)) についても説明すると良い。
- 4. 文字列を受け渡すときには、配列のアドレスでやり取りするように進める。 なぜなら、関数からmain関数に文字列を返すとき、ポインタ変数を使いwarning に悩む学習者が出る可能性があるからだ。できれば、どうしてwarning が出るかを理解できるような指導を行うとよい。
- 5. 関数の引数の考え方は、FORTRAN等の高水準言語でも悩むところである。特に、C言語では引き数の考え方が難しいので、理解できるように十分時間を取り、かつ、分かりやすい講義を心がける必要がある。
- 6. FORTRANのサブルーチン・コールは、すべてコール・バイ・アドレスであるため、ポインタにじゅうてんをおいたOJTとしては、FORTR-ANから呼ばれるC言語のかんすうの開発・保守が考えられる。

# 用 語

関数, 関数名, 関数の宣言, void, 引数, return文, グローバル変数, ローカル変数, call by value, call by address

## 第11章 標準ライブラリ関数

### 学習目標

学習者がこの章を修了した時点で標準ライブラリ関数(以下標準関数)に関連した次の事柄が出来る様になることを目標とする。

- 1. 標準関数を使い対数の計算を行うプログラムを作成する。
- 2. 標準関数を使い文字列を整数に型変換する。その逆を行うプログラムも作成する。
- 3. 標準ライブラリを使い文字列のコピー、追加、比較を行う。
- 4. #include文の機能を説明する。
- 5. マニュアルを見ることにより、標準関数の引き数の型、渡し方、処理を失 敗した時、返される値を調べ、利用する。

## 内容

# 11. 1 標準ライブラリ関数の呼び出し方(90分)

(1) log の計算

図表11-11-1 の様な簡単なサンプルプログラムを示し、各行について説明する。ここで標準ライブラリ関数とは何かを説明し、『標準ライブラリ関数とはC言語が用意してくれる関数で、使用者はその関数自体を記述することなく使用できる関数』といった程度の説明を行う。#include<math. h>と関数log()の関係については、標準ライブラリ関数の中には、ある決められたファイルを取り込まなければ実行出来ないものがあるとの説明にとどめる。#includeがファイルを取り込むマクロであるとの説明は必要であるが math.h ファイルの中身の説明までは必要ない。

どの関数を使う時、どのファイルをinclude しなければいけないかは、 すべて関数のマニュアルに書かれてあるため関数を使用する時は必ずマニュアルで調べる必要があることにも触れておく。

#### 図表11-11-1 「log の計算」

```
#include<math.h>
main ()
{
    double d=5.0, x;

    x=double(d);
    printf("log(%f)=%f \n", d, x);
}
```

#### 11. 2 標準ライブラリ関数の活用(180分)

#### (1) 変 換

ここでは、図表11-11-2 のサンプル・プログラムを示し、この中で使われている型変換標準関数 atoi(), sprintf(), fgets()の使い方を説明する。この2つの標準関数は『文字型 $\longrightarrow$ 整数型』の変換を行うために良く使われる関数であることを説明する。使い方の説明では、引き数の渡し方が、値渡し(call by value)、アドレス渡し(call by address)の2種類あることを説明した上で、各引き数がどの方法で渡されるかを説明する。標準関数の各引き数の型、渡し方は、すべてマニュアルに記載されているので、マニュアルを確認することを説明し、可能ならば、代表的なマニュアルを示し、説明すると良い。

## 図表11-11-2 「型変換プログラム」

```
#include<stdio.h>
main ()
{
    int i;
    char c[10], buf[80];
    fgets (buf, 80, stdin);
    i = atoi(but)i
    sprintf ( c, "%d", i);
    printf("int=%d, char=%s \n", i, c);
}
```

#### (2) 文字操作

次に、図表11-11-3 のサンプル・プログラムを示し、この中で使われている文字列操作標準関数 strcpy(), strcat(), strcmp()の機能と使い方を説明する。

ここでは他の文字列操作標準関数についても触れておく。プログラムの 説明ではstrcat()のコピー先(このプログラムでは変数buf)の最後(buf [13])にNULLが入っている様な時(このプログラムではあり得ない)、 コピー先変数以外の変数の領域を破壊する危険性があることなど、これら 標準関数を使うに当たっての細かい注意を説明する。

## 図表11-11-3 「文字列操作」

```
#define PASSWD "root/reaf"
#include<stdio.h>
#include<string.h>
main ()
{
    char log[14], passwd[14], buf[28];

    do {
        printf("key in logname : ");
        fgets(log, 14, stdin);
        strcpy(buf, log);
        strcat(buf, "/");
        printf("key in password : ");
        fgets(passwd, 14, stdin);
        strcat(buf, passwd);
    } while(strcmp(PASSWD, buf) !=0 );
}
```

## 11. 3 演習問題(600分)

- (1) atoi(), sprintf()atoi(), sprintf() を使用して、キャラクタとして数字を2つ入力し、2数の和を文字例として出力するプログラムを作らせる。
- (2) strncpy(), strncat(), strncmp() 図表11-11-3 のプログラムをstrncpy(), strncat(), strncmp()を使用し、書き換える。

(3) strtok()

strtok()を使用し、ファイル中の空白で区切られた文字列からある文字列を捜す。

# 育成方法および指導上の留意点

- 1. 標準関数は、用途が多く、機能的なものが多いので、これを知っていると プログラムの開発が楽になることを標準関数を使用しないプログラムと比較 して説明するとよい。
- 2. 型変換の説明では文字、文字列の説明をもう一度行うとよい。
- 3. 文字列操作では、多くの標準関数を使用したプログラムを作らせたり、説明すると良い。
- 4. 各標準関数は返される値(リターン・バリュ)でその関数の処理が失敗したかどうか確認出来ることを説明し、なるべくその値をチェックする様に指導する。

# 用 語

標準ライブラリ関数, #include, <math.h>, <stdio.h>, <string.h>, atoi(), log(), strcpy(), strcat(), strcmp(), strncpy(), strncat(), strncmp(), sprintf(), strtok(), return value, fgets()

## 第12章 記 憶 ク ラ ス

#### 学習目標

学習者がこの章を修了した時点で、変数の記憶クラスに関連した次の事柄ができるようになることを目標とする。

- 1. グローバル変数とローカル変数の有効範囲の違いを理解した上で使い分けする。
- 2. 同じローカル変数でも自動変数と静的変数で性質が異なることを理解した上で適切な用途に使い分けする。
- 3. レジスタ変数の意味、性質を理解した上で有効にレジスタ変数を利用する。

# 内 容

#### 12. 1 変数の有効領域(180分)

(1) グローバル変数 2

ここでは、ローカル変数とグローバル変数の違いを理解させるためのサンプル・プログラムを示し、解説する。ここで使用するサンプル・プログラムは、1つのコンパイル単位(1ファイル)とし、main()以外の関数(今後サブ関数と呼ぶこととする。)を最低一つ含むものを示す。解説時、サブ関数内でグローバル変数を参照/代入可能であること、サブ関数からメイン関数に戻るとグローバル変数がサブ関数で変えられた値をそのまま保持していることを説明する。自動変数の有効領域を理解させるためには、メイン関数とサブ関数で同じ名前の変数を宣言し、サブ関数で値を変えてもメイン関数に戻ると値が保持されていないことを説明する。また、その説明の時、サブ関数内で変数を宣言しない場合と比較するとよい。ここで取り扱うデータ型は、int とする。

# (2) 静的変数

ローカルな静的変数を解説するためのサンプル・プログラムを示す。そのプログラムは1ファイルで、main()と最低1つサブ関数を含み、サブ

関数内で静的変数を宣言しているものとする。サブ関数内の静的変数は2回目にそのサブ関数が呼び出された時に前回の値を保持していることを前回の値を保持していない自動変数と比較し、解説する。たとえば静的変数は初期値を0とし、そのサブ関数が呼ばれる毎に1加算され、値を出力する様にする。これらのサンプル・プログラムは学習者に、実行結果を示すとよい。

静的変数の有効領域はローカルで宣言されていれば他のローカルの自動 変数と等しく、グローバルで宣言されていれば他のグローバル変数と等し いことも解説の中で触れておくこと。

## 12. 2 レジスタの利用(60分)

#### (1) レジスタ変数

レジスタの解説は、メモリ側ではなくCPU側にあり高速にアクセス出来る領域であるといった説明でよい。レジスタ変数をうまく使うと高速に処理を行うことが可能であることをサンプル・プログラムで示す。サンプル・プログラムでは for文のループカウンタの変数をregister宣言した場合としない場合で実行速度が違うことを実演するとよい。レジスタ変数はローカル変数としてしか使用できないことに触れ、グローバルで宣言したり静的に宣言することは出来ないことを説明する。ただし、その理由については解説する必要はない。また、ハードウェアなどの制約から、レジスタ変数をたくさん宣言しても、実際にレジスタ領域に割り振られる(真のレジスタ変数となる)変数は、その内の一部であるため、すべての変数をレジスタ変数にするとよいことを強調する。

# 12. 3 演習問題(420分)

# (1) グローバル変数と引数

関数呼び出し時、引き数なしで、関数内でグローバル変数を使用している関数を引き数ありで内部でグローバル変数を使用しない関数に書き変える。

(2) レジスタ

プログラムを与え、どの変数をレジスタ変数にすれば高速になるかを考えさせる。

#### 育成方法および指導上の留意点

- 1. グローバル変数と外部参照(extern)の関係を必ずどこかで触れておく。
- 2. 静的変数と自動変数でコンパイル時にどの様に領域が取られるのかも、基本ソフト別(MS-DOS, UNIX)に説明する必要がある。
- 3. コンパイル単位が異なるファイルのグローバル変数がどの様になるか、外部参照との関連も含めて説明すると良い。
- 4. この章では、各変数の性質を理解した上での各変数に適した使い方を常に 意識させる必要がある。
- 5. グローバル、ローカルなどで同じ名前を使った場合を説明しながら、同名の場合、どちらが優先されるか、説明すると良い。
- 6. ローカルの自動変数の配列の初期化は、処理系により不可能なものが多い ので使用しているシステムではどうかを実例で示し、必ず説明すること。
- 7. 関数の仮引き数もローカルの自動変数であることにも触れておくこと。

# 用語

ローカル変数, グローバル変数, 自動変数, 静的変数, レジスタ, レジスタ 変数, 記憶クラス

## 第13章 ファイル処理

#### 学習目標

学習者がこの章を修了した時点で、ファイル操作、構造体に関連した次の事柄ができるようになることを目標とする。

- 1. ファイル入出力関数を使用し、ファイルを作成する。
- 2. ファイル入出力関数を使用し、ファイルからデータを取り出す。
- 3. ファイル入出力システム・コールを使用し、ファイルを作成する。
- 4. ファイル入出力システム・コールを使用し、ファイルからデータを取り出す。
- 5. 構造体の意味を配列との違いを説明する。
- 6. 構造体のアドレスからその構造体のメンバを代入/参照する。
- 7. 構造体の配列を使い住所録等を作成するプログラムをプログラミングする。
- 8. ファイルに対してランダム・アクセスする。

# 内 容

## 13. 1 ファイルの開き方(210分)

(1) 文字列の入出力1

fopen, fprintf, fcloseを使用してファイルを作るサンプル・プログラムを示す。この時、各関数の引き数の意味、とくにfprintf の%記述子の意味はprintfと同様であることなどを説明する。また、各関数のリターン・バリュについても説明する。この時、fopen のリターン・バリュ、ファイル・ポインタについては以降の関数で使用するポインタとだけ説明し、内部構造まで説明する必要はない。サンプル・プログラムで作成したファイルの中身も示すと良い。

次にfopen, fscanf, fclose 使用し、ファイルをオープンし、内容を fprintf で画面出力するプログラムを示す。関数の説明は、fscanfについ てのみ行う。fprintf で画面出力する時は、標準出力ポインタstdoutを使

用し、ついでに標準入力ポインタstdin, stderr の説明を行うと良い。

#### (2) 文字列の入出力2

システム・コール、creat, write, close を使用し、ファイルを作成するプログラムとそのプログラムにより作られるファイルを示す。各システム・コールの意味を引き数の説明、リターン・バリュの説明を行う。

システム・コールと前述の関数(fopen, fprintf, fclose)の違いは基本ソフトの機能の呼び出しであるといった説明に止める。ただし、実は関数内部でもシステム・コールしていることにも触れると良い。

システム・コール、open, read, close を使用し、ファイルの内容を読み、write システム・コールで画面出力するプログラムを示し、実行して見せる。ここでは、open, readの説明を行う。この時、openが返すリターン・バリュはcreat が返すものと同じファイル記述子と呼ばれるもので、read, write, closeの第1引き数として使われることを説明すると良い。write を使い画面を出力する時は、ファイル記述子としてfileno(stdout)を使用し、ファイル・ポインタと、ファイル記述子の関係を説明すると良い。

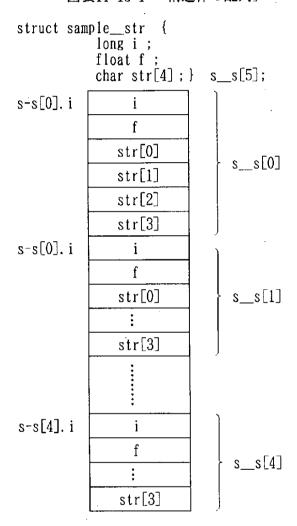
#### 13. 2 順編成ファイルの作成(300分)

#### (1) レコードの出力

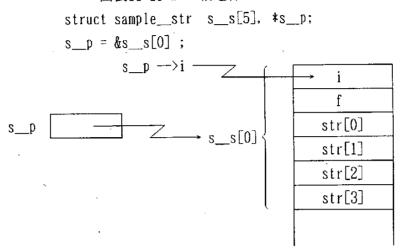
住所、氏名、年令、電話番号の項目からなる住所録を作成するサンプル・プログラムを先ず最初に示す。そのサンプル・プログラムでは、構造体(struct)を用い構造体の要素(メンバ)に住所録の情報を代入し、最後にファイルにwrite する様なものとする。構造体の宣言では、構造体を配列とはせず、1つの構造体を使用する。プログラムの説明をする前に構造体の説明を行い、特に構造体の型の宣言、構造体そのものの宣言の違いに注意して詳しく行う。

次のサンプル・プログラムでは構造体を配列とポインタで宣言し、構造体をポインタでアクセスする様なプログラムを示す。ここでは、構造体の配列の構造を図解(図表11-13-1 参照)などを用いて説明する。構造体へのポインタと、そのポインタを用いた構造体へのアクセスについても図解

# (図表11-13-2 参照) 等を用いて説明するとよい。 図表11-13-1 「構造体の配列」



## 図表11-13-2 「構造体へのポインタ」



## (2) レコードの入力

(1)で作成したファイルを入力し画面に出力するサンプル・プログラムを示す。そのプログラムは図表11-13-3 で示す様にデータの入力部分をサブ関数化したものとする。ここでは構造体をサブ関数の引き数とした場合の説明を必ず行う。

# 図表11-13-3 「構造体と引き数」

```
main ()

struct 住所録 addr__note;

getdata (& addr__note);

}

getdata ( ad__nt )

struct 住所録 *ad__nt;

データをreadし、ad__nt → 

return;
```

#### 13. 3 直編成ファイルの作成(90分)

(1) ランダムにアクセスする方法

システム・コール1 seekと標準ライブラリ関数 fseekを使って、ファイルをランダム・アクセスするプログラムを示し、説明する。ここでは、ファイル入出力時の現在位置(ファイルポインタ/記述子の現在位置)の考え方を1seek, fseekの説明とあわせて行う。厳密な意味でのシーケンシャル・ファイル、ランダム・ファイルの説明を行う必要はないが、シーケンシャルなアクセス、ランダムなアクセスの違いについては説明しておく。

#### 13. 4 演習問題(720分)

- fprintf(), fscanf()
   fprintf(), fscanf()で種々の%記述子を使用し、fprintf(), fscanf()
   の理解を深めるプログラムを作成させる。
- (2) fileno(), fdopen()
  fileno(), fdopen() を使い、ファイルポインタ/記述子の変換を行うプログラムを作成させる。
- (3) 構造体

構造体及び構造体へのポインタ、配列について理解を深める問題を考える。たとえば、13.2で示したサンプル・プログラムを用い住所録ではなく、他の対象を操作するようなプログラムを作成させる。

(4) lseek(), fseek()

lseek(), fseek() を行った住所録をバイナリ・サーチするプログラムなど、ランダムなアクセスを理解させるプログラムを作成させる。

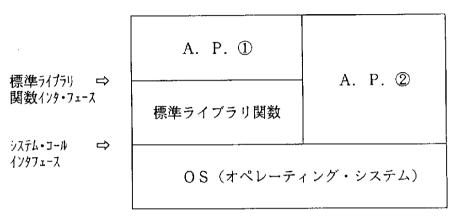
# 育成方法および指導上の留意点

- 1. 構造体の型の宣言と構造体自体の宣言の違いを意識させる。
- 2. 構造体と言うデータ構造は、BASICやFORTRANにはないため、 プログラムを1人1人に作らせ理解させることが必要である。
- 3. 構造体を関数の仮り引き数にするためには、ポインタで受け渡しするしか

ないという前提を説明する。そして構造体に対するアクセス ( xxx. yyy と xxx→yyy ) 方法はプログラムを作らせることによって理解させる。

4. ファイル入出力関数とファイル入出力システム関数の違いを図表11-13-4 により示し、説明すると良い。

図表11-13-4 「関数とシステム・コール」



- ・A. P. ①はfopen、fprintfなどを使用したアプリケーション・フロクラム
- ・A. P. ②はopen, readなどを使用したアフリケーション・フロクラム
- ・A.P.①はシステム・コールを使用してはいないが、標準 ライブラリ関数がシステム・コールを使用している。
- 5. ファイル・ポインタは、内部にファイル記述子を含むことにも触れておく と良い。
- 6. ハードウェアの違いによりデータの表現形態が異なるので、そのCPUののレジスタ幅に合わせた構造体をつくることに気をつけるよう指導する。

# 用 語

struct, a.b,  $a \rightarrow b$ . a[0].b, fopen(), fclose(), fprintf(), fscanf(), open(), close(), creat(), read(), write(), lseek(), fseek(), ファイル記述子, ファイル・ポインタ, システム・コール, シーケンシャル・アクセス, ランダム・アクセス,

#### 第14章 特徴的なプログラム

#### 学習目標

学習者がこの章を修了した時点でC言語に関連した次の事柄ができるようになることを目標とする。

- 1. システム関数を使い時刻を求める。
- 2. 割込みを利用し、シグナル処理を行うプログラムを作成する。
- 3. 共用体を使ったプログラムを作成する。
- 4. リカーシブ・コールを利用してn!を求めるプログラムを作成する。
- 実行時に渡された引き数を表示するプログラムを作成する。
- 6. #define を使ってマクロ関数を定義する。
- 7. #ifdefをコンパイル時のオプション(-Dなど)と供に用いデバッグ情報 を出力する。
- 8. グローバル変数でデータを受け渡すプログラムを分割コンパイルする。

# 内 容

## 14. 1 システム・コール (240分)

(1) 時間を呼び出す

図表11-14-1 のサンプル・プログラムを示し、この中で使用しているti-me()システム関数の機能、およびシステム関数について説明する。システム関数の説明は第13章でも行っているが、ここではファイル入出力だけでなく色々な機能があることにも触れる。

図表11-14-1 「時間を呼び出す」

```
main ()
{
    long t ;
    time (&t) ;
    printf("time from '70/1/1 is %d sec\n", t) ;
}
```

#### (2) 割込み

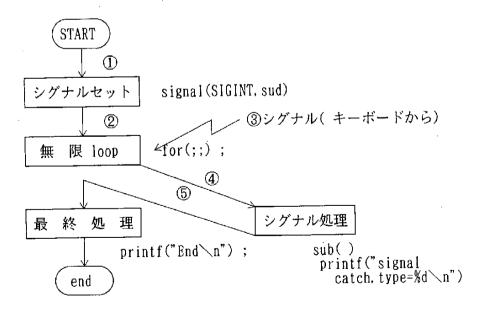
図表11-14-2 で示すサンプル・プログラムを説明し、実行してみせると良い。ここでの説明ではシグナルを受け取った時どの様に動くかなどに触れる必要がある。シグナルの種類はシステムにより異なるので具体例で説明する。図表11-14-2 のプログラムは、このままでは終了しないのでキーボードから割込みシグナル( $\land$  Cなど)で終了させる。この時のこのプログラムの動きについては図表11-14-3 を使い図解する。

図表11-14-2 「割 込 み」.

```
#include <signal.h>
main()
{
    void sub();
    signal(SIGINT, sub);
    for(;;);
    printf("End n");
}

void sub(sigtype)
{
    printf("signal catch, type=%d n", sigtype);
}
```

図表11-14-3 「割込みの動き」



## 14. 2 共用体(120分)

#### (1) 共用体

ここでは共用体の機能、使い方、利点について説明する。共用体の機能としては、同一の領域を intや charfloatで使用出来ることを説明し、使い方では宣言の方法などを説明する。共用体の利点としては、色々な型の変数のバッファを共用体化することで記憶領域の節約が可能な点や、浮動小数点型式のデータからbit 情報を取り出すことが可能、レジスタ情報の上位下位 bitの取り出しが可能なことをサンプル・プログラムで示す。共用体の配列や共用体のメンバとして構造体を使った例も説明すると良い。

#### 14. 3 リカーシブ・コール(120分)

#### (1) n!の計算

nの階乗を求めるプログラム図表11-14-3を示し、この中で使われているリカーシブ・コールのテクニックとその意味を説明する。リカーシブ・コールと漸化式の関係についても触れ、漸化式化されたものは、リカーシブ・コールにより答を出すことができることを $\Sigma$ nの例などで示すと良い。リカーシブ・コールが何回まで可能であるかは、システムにより異なることをスタックとの関係に触れながら説明すると良い。

# 図表11-14-3 「階乗を求める」

# 14. 4 テキスト・ファイルの出力(120分)

#### (1) コマンド・ライン引数

図表11-14-4 を示しこのプログラムの説明を行う。ここでは、とくに実行時に渡す引き数(コマンドライン引数)がプログラム内で利用可能であることを実行例も示して説明する。入力した引数はコマンド名自身もargv [0] で受け取れることも説明すると良い。このプログラム自体は実行時に引数として渡されたファイル名をプログラム内でオープンし、中身を出力するプログラムである。argcの説明では実行時に渡されたコマンド名を含めた引数の数であることを述べる。

図表11-14-4 「コマンド・ライン引数」

```
#include <stdio.h>
main(argc. argv)
int argc;
char *argv[];
{
    FILE *fp;
    char buf[80];
    int i;

    for ( i=1; i < argc; ++i ) {
        fp=fopen(argv[i], "r");
        while(fgets(but, 80, fp)!=NULL) {
            fprintf(stdout, "%80s", but);
        }
        fclose(fp);
    }
}</pre>
```

# 14. 5 プリプロセッサ(240分)

# (1) プリプロセッサの使い方1

図表11-14-5 のサンプル・プログラムを示し、#define の使い方を説明する。ここでは、-1が ENDとして使われている例と、大文字、小文字変換を行うマクロ定義の例があるので、この2種類がコンパイル時にどのように展開され使われるのかなども含めて説明する。そしてプリプロセッサが

コンパイルの直前、またはコンパイルの第一段階で定義された文字や式に 置き換わること(サンプルプログラムでは BNDが-1に、LOWE(c) がc+0x20 にUPPER(c)がc-0x20に置き換わること)を説明する。

図表11-14-5 「プリプロセッサ1」

(2) プリプロセッサの使い方2

図表11-14-6 のプログラムを示し、コンパイルの方法を

cc file名

cc -DDBG file名

の2種類行い、どのように実行結果が違うかを示す。

#ifdef説明では、条件が合った部分のみがコンパイルの対象となることを説明する。この時、#ifndef, #ifについても触れておくと良い。

#### 図表11-14-6 「プリプロセッサ2」

```
#define
         RND
         UPPER(x) (x - 0 x20)
#define
         LOWER(x) (x + 0 x20)
#define
main()
       char c;
       while((c=getchar()) !=END) {
            outchar(c);
  #else
                       <=c && c <= 'Z')
putchar(LOWER(c))
            if ('A'
            } else {
                         putchar(UPPER(c)) ;
  #endif
```

## 14. 6 自作関数のリンク(120分)

#### (1) 分割コンパイル

ここでは1ファイル、1関数でコンパイル、リンクする方法(分割コンパイル)とその利点、注意などを説明する。

リンクの方法はシステムにより異なるので現在使用のシステムについて 説明する。利点の説明では、手直しした関数のみをコンパイルするだけで 良い事や、必要な関数を他のシステムから持って来る時は、その関数のファイルのみを持って来れば良い事などを説明する。注意することとしては、グローバル変数を宣言する時、1つのファイルで宣言を行い、他のファイルではextern(外部参照)とすることが必要であることなどを説明し、その中で変数の有効領域(scope)について関連付けて説明することが重要である。

## 14. 7 演習問題 (900分)

#### (1) シグナル

サンプル・プログラムとは異なるシグナルを受け取るプログラムを作成 させる。

- (3) コマンド・ライン1与えた引数をすべてエコー・バックするプログラムを作成させる。
- (4) コマンド・ライン 2 echoという名前で実行された時は引数をエコー・バックし、cat という 名前で実行された時は引数で指定されたファイルの中身を表示するプログラムを作成させる。
- (5) プリプロセッサ
  #define を使ってMAX(x, y), MIN(x, y) の名前で引数の大きい方/小さい
  方を返す関数を定義させる。
- (6) 分割コンパイラ 1ファイル複数関数のプログラムを分割コンパイルさせる。

## 育成方法および指導上の留意点

- システム・コールにはどの様なものがあるかについてマニュアルで調べて 利用できる様にする。
- 2. システム・コールと標準関数の違いはプログラムを作る側からはあまり意識できないため、コンパイラが自動的にリンクするライブラリ・モジュールの存在を示すなどの工夫が必要である。
- 3. シグナルを受け取ることにより関数にジャンプすることと、通常の呼び出しによりジャンプすることの違いは、プログラムの動きを追わせる様な指導をすると良い。
- 4. リカーシブ・コールの概念を、はじめて学ぶ学習者に対しては、"自分で自分を呼び出す"という説明や、動きを図解したり、サンプル・プログラムでリカーシブ・コールの動きを表示させるような指導も考えるとよい。
- 5. リカーシブ・コールに重点を置いたOJTとしては、コンパイラ、インタ プリタなどの構文解析の部分のシステム開発・保守が考えられる。

- 6. #define を使ったマクロ関数と、通常の関数との違いを理解しているかを確認するため、両者の長所、短所などを説明させると良い。
- 7. #ifdefがデバック時に有用なツールであることと、同じプログラムを異なるシステムでも生かそうとする時、図表11-14-7 の様なコーディングを行うと同一ファイルを使用できることについても触れると良い。

図表11-14-7 「システムの差異の吸収」

main()
{
#ifdef UNIX

UNIX 固有のコーディング

#endif
#ifdef MSDOS

MS-DOS 固有のコーディング

#endif

語

用

システム関数, 割込み, シグナル, time(), <signal.h>, signal(), SIGINT, 共用体, リカーシブ・コール, 実行時の引数, argc, argv, プリプロセッサ, #define, #ifdef, #ifndef, #if, 分割コンパイル, extern, 有効領域(scope)

#### 推薦教材

## インストラクタ必読書

文献1. プログラミング言語C第2版

B. W. カーニハン/D. M. リッチ- 著・石田晴久訳 共立出版 1989年

文献 2. PROGRAMMING with C, BYRON S. GOTTFRIED, McGRAW-HILL, 1990

文献3. 実習C言語 三田典玄著 アスキー出版 1986年

文献 4. 実用UNIXシステムのクロクラミンク 塩谷修著 日刊工業新聞社 1986年

# 参考文献

文献1. プログラミング言語C第2版

B. W. カーニハン/D. M. リッチ- 著・石田晴久訳 共立出版 1989年

文献 2. PROGRAMMING with C, BYRON S. GOTTFRIED, McGRAW-HILL, 1990

文献3. 実習 C 言語 三田典玄著 アスキー出版 1986年

文献 4. Turboリファレンスガイド Borland International 著

(株)マイクロソフトウェア アソシエイツ訳 1988年

文献 5. UNIX system V ブログラマ リファレンス・マニュアル第2版 リリース 3.0,

AT&T UNIX ソフトウェアオペレーション・バシフィック発行 共立出版 1985年

# 第12部 システム開発技法

#### 教育目標

当部では、アプリケーション・システムの開発からプログラム開発、プログラム・テストからシステム運用に至るまでのすべての局面について説明する。 さらに、それらの各局面を成功裡に遂行していくための開発管理、組織体制のあり方、および情報処理技術者の位置づけと役割についても理解させる。

当部で取り上げるテーマのいくつかは、初級情報処理技術者の守備範囲を超えるものもあるが、これらは初級情報処理技術者の担当業務を全体の中に位置づけて考え、かつ視野を拡大してもらう狙いがある。また中上級情報処理技術者を目指すためのキュリアの一環としての知識を修得させる目的もある。プログラム設計やテストの知識・技術は初級情報処理技術者にとっても不可欠なものであり、詳細設計からプログラム設計、さらには統合テストまでは、一貫したグループ演習・実習を通して、知識・技術の定着を図る。

# 内容の構成

	内容	時 間 数	
	内 容 	講義	演習
1. 1 1. 2 1. 3 1. 4	システム開発とシステム設計局面 システム開発過程の概観 システム分析と要求定義局面 システムの外部設計(概要設計)局面 システムの内部設計(詳細設計)局面 システム設計演習	6	12.
2. 1 2. 2 2. 3 2. 4	プログラム設計技法 プログラム開発手順 プログラムの構成 プログラムの結合方法 プログラムの構造化設計 プログラム設計演習	. 6	10
第3章 3.1 3.2 3.3	プログラミング技法 モジュールの論理設計 論理設計の代表的な手法 プログラミング技法演習	· 6	6
4.5		6	16
5. 1 5. 2	システムの導入と運用 システムの導入 システムの保守 システムの評価	3	
6. 1	システム開発の進め方 システム開発と組織体制 システム開発の管理	2	
<u> </u>	合 計	2 9	4 4

## 第1章 システム開発とシステム設計局面

## 学習目標

この章を修了した時点で、学習者がシステムの開発手順とシステム設計に関 して、次の事柄ができるようになることを目標とする。

- 1. システム開発の全過程を展望し、それぞれの段階(局面)の主要な目的と作業内容を説明する。
- 2. システム分析と要求定義および外部設計段階で用いられるデータ・フローダイヤグラムやトップダウン展開の技法を説明し、その成果物である要件業務フローやデータ・フローを解読する。さらに典型的なデータ・フローを参考に類似のデータ・フローを作成する。
- 3. サブシステム展開、画面設計と報告書設計、入力原票設計、コード設計、 論理データ設計およびシステムの安全性と信頼性の設計の手順と要点を説明する。
- 4. 処理の詳細設計、入出力の詳細設計および物理データ設計の手順と要点を説明する。さらに例をもとに簡単な処理の詳細設計を行う。

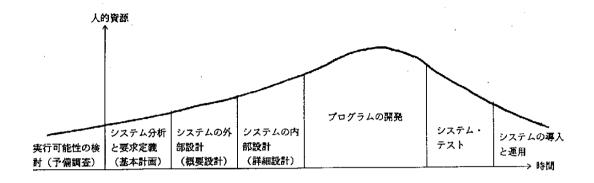
# 内 容

# 1.1 システム開発過程の概観(60分)

システム開発は、通常図12-1-1で示しているようなステップを踏んで行われる。この開発過程に、最近は局面化開発技法を適用するやり方が目立ってきている。この技法は、各局面(過程)が完了したことをしっかり確認したうえで、次の局面に着手するやり方である。これによって、開発管理を容易にし、かつ利用部門からの参画をしやすくし、ひいては開発生産性の向上に結びつけることができる。

システム開発過程を概観するにあたっては、それぞれの局面の主要な目的と 作業内容および成果物についてふれる程度にとどめる。

図12-1-1 システム開発過程



- (1) 実行可能性の検討(予備調査)
- (2) システムの分析と要求定義(基本計画)
- (3) システムの外部設計(概要設計)
- (4) システムの内部設計(詳細設計)
- (5) プログラム開発
- (6) システム・テスト
- (7) システムの運用

# 1.2 システム分析と要求定義局面(60分)

この局面は、システムに含めるべきユーザの要件を明確にする段階であり、 新システムの必要機能、全般的な情報要件など明らかにする。以下の内容展開 で説明する。

- (1) システム分析と要求定義の過程
  - ① システム課題の明確化各部門、各層から集めたシステム化のニーズを源として、システム化の課題として統一的な方法にまとめてあげることである。
  - ② システムの構想立案 新システムとして確定するために、現行システムとの差異を明らかに し、その原因と対策を考案することで新システムの内容を具体化できる。

#### ③ 基本計画書のまとめ

これまで検討してきたシステム構想を要件業務フローにまとめる。さらに日程計画、要員計画、資金・設備計画などの実行計画を付加し基本計画書とする。

- (2) システム分析と要求定義の技法
  - ① データ・フロー・ダイヤグラム (Data Flow Diagram : DFD)特徴、記号、階層表現、作成上の注意点を説明する。
  - ② トップ・ダウン展開
- (3) 成果物の種類
  - ① 要件業務フロー
  - ② 処理の記述
  - ③ データ・フロー記述
  - ④ データ・ストア記述

#### 1.3 システムの外部設計(概要設計)局面(120分)

この局面は、利用者(ユーザ)の立場からみたシステムの定義、つまりユーザとのインタフェースを中心としたシステム設計の段階である。要求定義局面での成果物を確認し、新システムの機能を展開し、実現性を検討していく。さらに入出力仕様やファイルの論理仕様を決定していくのが主要な作業内容である。ここで決定するシステム概要が、最終的なシステムの機能・性能や信頼性に大きく影響することを十分認識する。

- (1) サブシステムの定義と展開
  - ① サブシステムの定義システムの下位機能であるいくつかの情報サブシステムを定義する。
  - ② サブシステムの展開 サブシステムの機能をさらに細分化し、最終的にユーザ部門へ提供情報を作成する単位までブレーク・ダウンする。
- (2) 画面設計、報告書設計、入力原票設計
  - ① 画面設計
    - (a) 画面設計の手順

- (b) 画面全体像の作成……使用する画面全体の構成と関連を一覧図に表わす
- (c) 画面設計の標準化……画面レイアウトと画面フローの標準化を図る。 設計面と運用面での効果が期待できる
- (d) 画面フローの設計……画面をどのような順序で実行させるかを設計 する
- (e) 画面の表現方法の検討……画面の使用頻度、利用者の熟練度などを 考慮して入力方法を決定する
- (f) 画面レイアウト設計……画面イメージをレイアウトし、同時に画面 仕様を設計する
- (g) 画面設計上の注意点
- (h) 画面設計の例
- ② 報告書設計

まずコンピュータからの出力情報が利用者の業務に役立つための条件 を考える。すなわち、正確性、タイムリー性、内容の妥当性、簡潔な表現である。

- (a) 報告書設計の手順
- (b) 出力概要検討……情報名、使用目的、周期、時期、配布先、発生量、 内容項目を検討する
- (c) 出力方式と媒体の決定……出力情報に応じた出力方式と媒体を選択する
- (d) レイアウト作成……出力イメージならびに出力仕様を作成する
- (e) 報告書レイアウト例
- ③ 入力原票設計
  - (a) 入力原票設計の手順
  - (b) 入力概要検討……入力データ名、発生源、周期、時期、データ量、 内容項目を検討する
  - (c) 入力方式と媒体の決定……発生した入力データを伝票からフレキシブル
    - ・ディスクなどに変換する間接入力方式と、キーボードやバーコード

- ・リーダなどによる発生データを直ちに入力する直接入力方式とがあ る
- (d) レイアウト作成……入力イメージ図ならびに入力仕様を作成する
- (e) 入力原票レイアウト例
- (3) コード設計
  - ① コードの必要性と機能
  - ② コード設計上の留意点 利用範囲と使用期間を決定し、拡張性があり体系的で扱いやすいコー ドを設計する
  - ③ コード設計を以下の手順で示す
  - · (a) コード化対象の選定
    - (b) コード化目的の明確化
    - (c) データ数の規定
    - (d) 利用範囲の決定と使用期間の推定
    - (e) コードの考案……コードの種類と特徴を知り、組合せや応用を考える
    - (f) コード化作業とコード表作成、コード改廃基準の作成
    - (g) コード・ファイルの作成(コンピュータ用)
  - ④ コードのチェック方法 コード入力の誤りの種類をあげ、チェック・ディジットの算出手順と エラー検出のしくみを理解させる
  - ⑤ コード設計の例
- (4) 論理データ設計

サブシステムの定義と展開で明確にされたインタフェース・データと入 出力設計で検討された入出力データから、システム内に蓄積すべきデータ 構造とデータ項目を決定する。

- ① 論理データ設計の手順
  - (a) データ項目の洗い出し
  - (b) データの関連性の分析
  - (c) ファイル候補の決定

- (d) ファイル候補内のデータ項目とジョブとの対応を検討し、ファイル 項目の抜けとジョブの抜けの双方をチェックする
- (e) ファイル仕様の作成
- ② 論理データ設計の結果(例)
- (5) システムの安全性・信頼性の設計 入出力設計に当たるとき、特に人と機械の関連性について最初に考慮しておくことが必要である。
  - ① マンマシン・インタフェース
    - (a) 操作性
    - (b) 安全性……誤動作を起こさせない
  - ② システムの信頼性
    - (a) 障害の種類
    - (b) 隨害対策
- (6) 概要設計書のまとめ

概要設計書はトップやユーザ部門の承認を得た後、詳細設計局面に渡す ことになる。概要設計書の内訳は次のようなものである。

- ① はじめに……全体構成と主旨
- ② 総説……システム開発の背景、目的と期待効果、主要機能など
- ③ システム概要……トップやユーザ部門に対して説明するもので、運用 方法を中心に絵や表で平易に表現する
- ④ 情報システム構造図、プロセス図……情報システムの構造、機能、インタフェースを階層構造図、IPO図などを用いてシステムの内部仕様を表現する
- ⑤ 入出力仕様、レイアウト……画面、伝票、帳票に関係する入出力仕様 とイメージとをつけ、入出力一覧にまとめたもの
- ⑥ ファイル仕様
- ⑦ コードブック
- ⑧ コンピュータの機器構成図、性能表
- ⑨ 詳細設計以降の日程計画表

#### 1.4 システムの内部設計(詳細設計)局面(120分)

この局面は、情報システム部門または内部からみたシステムの定義である。 したがって、外部設計局面で作成されたシステムの概要設計を、装置との対応 やシステムの最適化の観点から、より詳細な設計へと発展させる。この段階で の主な作業内容としては、処理の詳細設計、入出力詳細設計および物理データ 設計がある。

- (1) 処理の詳細設計
- ① 処理設計の目的
  - ② HIPO (Hierarchy plus Input Process Output)による記述
    - (a) 図式目次(階層構造図)
    - (b) TPOダイヤグラム(入力・処理・出力)
    - (c) 作成上の考慮点
    - (d) HIPOの例
  - ③ プログラム構造図の作成
    - (a) 階層構造図の指針……階層の深さと並列ブロックの数の制限
    - (b) 作成上の考慮点……処理のタイミングを考慮し、使用するファイル が共通のものをまとめ、構造図を見ただけで内容が理解できるよう表 現方法を工夫する
    - (c) プログラム構造図の例
  - ④ プログラム機能仕様の作成
    - (a) 総括ダイヤグラム……いくつかのプログラムで実行される機能と入 出力をIPO形式で記述したもの
    - (b) 詳細ダイヤグラム……プログラム単位の I P O 記述
    - (c) プログラム機能仕様の作成……プログラム機能仕様を次の構成でま とめる
      - ア 表紙、目次
      - イ システム流れ図
      - ウ プログラム構造図
      - エ 総括ダイヤグラム
      - オ 詳細ダイヤグラム

カ 入出力詳細仕様 キ テーブル仕様他

#### (2) 入出力詳細設計

概要設計では、画面を含む入出力設計はユーザ部門にわかりやすいよう にイメージを中心に表現した。詳細設計ではハードウェアの制約などを考 慮しつつ、入出力フォーマットをスペーシング・チャートなどに記述する。 さらに入力のチェック方式もこの段階で設計する。

### ① 原票設計

概要設計に引き続き、原寸大のレイアウトを作成し、仕様も完成させる。

#### (2) 帳票設計

出力イメージに基づき、コンピュータで出力できる様式に整えてスペーシング・チャート上に設計する。このとき内容項目の配置、編集はもちろん、用紙の形式についても決定する。

#### ③ 画面設計

画面イメージを検討し、フィールド属性(入力可不可、反転、2重輝度など)、色などを決定する。入力エラー、入力の習熟に関する検討も加える。

④ 入力チェック方式

入力データを正しく保つためにエラーの種類を想定し、修正および救 済方法を考案する。

# (3) 物理データ設計

論理データ設計で定められたファイルのデータ項目や使われ方からファイルの特性(使用率、増加率など)を検討し、編成とレコード・レイアウトを決定する。

- ① 物理データ設計の手順
  - (a) ファイル特性の理解
  - (b) ファイル編成方式の決定
  - (c) ファイル媒体の決定
  - (d) レコード・レイアウトの決定

- (e) アクセス時間と容量の見積り
- ② レコード・レイアウトの設計
  - (a) ファイル/レコード/フィールドの関係
  - (b) レコード・タイプ (固定長レコード、可変長レコード)
  - (c) レコード・レイアウト設計上の考慮点
- (4) 詳細設計書のまとめ

概要設計書に基づいて、コンピュータ・システムを中心に設計してきた 詳細設計書を次のようにまとめる。

- ① はじめに……全体構成と主旨
- ② 設計方針……設計方法、文書化方法、設計手順、変更処理など
- ③ プログラム構造図……サブシステム概要、サブシステム別プログラム 構造図
- ④ 共通プログラム一覧
- ⑤ プログラム機能仕様
- ⑥ システム性能/容量分析書
- ⑦ 共通テーブル仕様書
- ⑧ データ・ファイル仕様書
- ⑨ モジュール設計指針

#### 1.5 システム設計演習(720分)

個人単位の小演習により知識・技術を一部修得させた後、研究テーマを与えて グループ演習により、その定着化を図る。

- (1) 個人単位の小演習
  - ① データ・フロー記述

簡単な販売管理のDFDを示し、処理機能の変更/追加を与え、DFDを書き直す。DFDはシステム分析と要求定義局面で使用される技法であるが、機能分割の考え方はプログラムの構造化設計と同一であり、決して難解なことではない。

② 画面設計 出力項目を示し、レイアウトを作る。できれば、端末にレイアウトを 入力し、画面をみながら評価し合う。

- ③ コード設計条件を与え、簡単なコードを設計する。
- ④ 論理データ設計

ファイル項目が決定してしまうと、処理機能、出力情報も大きく制約 されてしまう。このような重要な論理データ設計を理解するうえでも小 演習を行ってみる。演習内容としては、出力辞意を与えてみて、必要な 蓄積データは何かを洗い出してみるような問題が適当である。

#### (2) グループ演習

簡単な販売管理システムの概要設計書を準備し、詳細設計をグループで 行う。設計終了後、他グループのレビューを受け、課題の発表を行ってみ る。

- 研究テーマ
   身近なシステムを取り上げる
- ② グループ編成4~5名を1グループとする
- ③ 用意するもの
  - (a) 概要設計書
  - (b) 演習の手引き
  - (c) 詳細設計書の見本

# 育成方法および指導上の留意点

当章は、学習者がシステムの分析や設計を行うことができるようにすることを狙っているのではない。システム設計が完成するまでにどのような視点でいかなるプロセスが遂行され、それぞれのプロセスでの成果物がどのような内容のものであるかを把握させるのが目的である。これによって、広い視野をもたせ、自分達が担当するプログラム開発の位置づけと意義とを正しく認識させる。また、将来システム・エンジニア(SE)を目指す人達にとってはキャリア教育の一環とみなすべきである。ただし、講義だけでは理解も表面的であり、シ

ステム設計は何をするのかも認識できない。また急増するプロダクション・エンジニア(PE)育成の要請に答えるためにも、詳細設計以降のシステム設計 演習は必要である。

なお、システム設計の重要な要素の1つとしてデータベースの設計や通信ネットワークの設計があるが、この章の内容としては高度すぎるので特にふれていない。初級情報処理技術者に対しては、この種の設計面まで立ち入らなくてもよい。

# グループ演習の指導上の留意点

個人演習のみであると設計できなくても個人の責任範囲で済むが、グループ 演習であると、グループ内の他のメンバーに迷惑がかかることから思わぬ力を 発揮することが多い。又、グループ内討議を通じて、色々な考えを持つ人間が 居ることを認識することは、グループ演習の大きな効果である。

グループ運営での注意点を次にあげる。

- 1. グループ内で安易な分担主義に陥らぬように注意すること。
- 2. 徹底的なレビューを行うことで、チームとしてまとまりのある成果を生む努力をさせること。
- 3. 集団になじめないメンバーが多いグループには指導者が介入し、グループ・リーダへの支援が必要である。必ずしもスキルのある人間が集まったグループが良い成果を生むとは限らない。チーム・ワークの良いグループが良い結果を生む。

# 用 語

局面化開発,要件業務フロー,データ・フロー・ダイヤグラム,データ・フロー記述,データ・ストア記述,トップ・ダウン展開,サブシステム展開,画面フロー,出力イメージ図,入力イメージ図,コードの機能,データ構造,マン・マシンインタフェース,障害対策,HFPO,入力チェック方式

# 第2章 プログラム設計技法

### 学習目標

この章を修了した時点で、学習者がプログラム設計技法に関連して、次の事柄ができるようになることを目標とする。

- 1. プログラム開発における作業ステップを列挙し、それぞれのステップでの主要な目的と作業内容を説明する。
- 2. プログラムを構成する要素を指摘し、それらの相互の関連を説明する。
- 3. 分割して作成したプログラムを結合する4つのやり方の概要を説明する。
- 4. 最適なプログラム分割技法を適用してプログラムの構成化設計を行い、 かつ設計した結果のモジュールの分割と独立性の妥当性を評価し必要に応 じて手直しをする。

# 内容

# 2.1 プログラムの開発手順(60分)

前章では、システム開発の全過程を概観し、そのうちの「システム分析と要求定義」から「システムの内部設計」までの過程を詳述した。本章の主題であるプログラム開発は、内部設計の成果物であるプログラム構造図とプログラム機能仕様、およびファイルと入出力詳細仕様などの情報を入力源に、その作業を進める。つまり、プログラムの構造化設計、モジュールの論理設計とコーディング、テストといった作業を進めプログラムを完成するのが、プログラム開発である。

本節では、プログラム開発手順とそれぞれの段階での主要な作業内容を概観 する。なお、テストに関しては第4章で取り上げる。

- (1) プログラムの構造化設計
  - ① 複雑さの減少(分割、独立性、段階構造化)
  - ② 分割(モジュールの大きさ)
  - ③ 独立性(機能強度、データ結合)

- ④ 階層構造化(段階的精練)
- (2) モジュールの論理設計とコーディング
  - ① 3つの基本制御構造(順次、選択、繰返し)
  - ② GO TO LESSプログラム
  - ③ ストラクチャード・コーディング
- (3) 品質管理とテスト
  - ① 人間による検証(ウォークスルー)
  - ② テスト方法の検討(トップダウン・テスト、テスト計画の作成、テストの実施、結果の検証)

#### 2.2 プログラムの構成(60分)

プログラム開発のそれぞれの局面での具体的な作業内容や技法を解説するに 先立って、本節と次節とでは予備知識として必要になる事柄についてふれる。

1個のソフトウェア・システムやアプリケーション・システムは、数種類から数十種類のプログラムで構成されるのが普通である。そのおのおののプログラムの構成を以下の順序で説明する。

- (1) ソフトウェア・システムとプログラム
  - ① ソフトウェア・システム
  - ② プログラム
- (2) プログラム・モジュール
  - ① 上位モジュール
  - ② 従属モジュール
- (3) セグメント
  - ① 上位セグメント
  - ② 従属セグメント
- (4) ステートメント
  - ① ステートメントと複合ステートメント
  - ② 実行ステートメントと非実行ステートメント
- (5) 要素

#### 2.3 プログラムの結合方法(60分)

プログラムの処理内容が複雑で高度なものになってくるにしたがって、2つのプログラムを機能単位にいくつかの部分に分割し、複数のプログラマが同時に作業を進めていくやり方が一般化してきた。いわゆるプログラミングの分業と協同作業の体制である。このための作業分担の単位に用いられるのは、通常プログラム・モジュールである。

プログラムを分割して作業を進めるとなると、当然のことながらその結果を 1つのプログラムに結合する必要が生じる。結合のしかたは、その結合時点に よって次の4つに大別できる。それぞれの方法について、そのやり方の大要を 説明する。

- (1) 原始プログラム・モジュールの作成段階
- (2) 原始プログラム・モジュールの翻訳段階
- (3) 目的プログラム・モジュールの連係編集段階
- (4) ロード・モジュールの実行中

### 2.4 プログラムの構造化設計(180分)

本節では、プログラムの構造化設計の基本的な考え方を整理したうえで、構造化設計の手順と具体的な構造化設計の技法を紹介する。さらに、構造化した設計結果の評価のやり方についても解説する。

- (1) 構造化設計の基本概念
  - ① プログラム設計の問題点
  - ② よい設計の要点
    - (a) 複雑さの減少……適切な大きさに分割し、独立性を高め、適切な階 層構造にする
    - (b) 分割……開発言語によって異なるが、COBOLでは/モジュール が300~500ステートメント程度が適当
    - (c) 独立性……モジュールが機能的にまとまっていて、モジュールとモジュールの関連が単純になるように分割する
    - (d) 階層構造化……1つのモジュールから呼び出す従属モジュールの数を制限し、階層もあまり深くならないようにする

- 3 プログラムの構造化設計
  - (a) 階層モジュール構造図の作成
  - (b) モジュール間インタフェースの定義
  - (2) 構造化設計の手順
    - ① 最上位モジュールの定義
  - ② モジュールの機能分析
    - ③ 分割技法の選択
  - (4) モジュールの分割
    - ⑤ インタフェースの定義
    - ⑥ 分割すべき他のモジュールの検討
  - (3) 代表的な分割技法
    - ① データの流れに着目した分割技法
      - (a) 源泉/変換/吸収分割(STS分割)
      - (b) トランザクション分割
      - (c) 共通機能分割
    - ② データ構造に着目した分割技法・ジャクソン法
  - (4) 設計結果の評価
    - ① 評価のポイント(分割、独立性、階層構造化)
    - ② モジュール強度 (暗号的、論理的、時間的、手順的、連絡的、情報的、 機能的)
    - ③ モジュール間結合度(内容結合、共通結合、外部結合、制御結合、スタンプ結合、データ結合)
    - ④ 制御範囲と影響範囲

# 2.5 プログラム設計演習(600分)

第1章に引続き、本章にてもプログラム設計に関する小演習とグループ演習 を行う。

- (1) 個人単位の小演習
  - ① データの流れに着目したモジュール分割

受注あるいは売上などのトランザクション処理を別に示し、ほぼ同様のトランザクション処理をモジュール分割し、HIPOを作成する。設計後、何人かでグループを作り、評価する。

### ② データの構造に着目したモジュール分割

売上報告書作成あるいは請求書作成などのバッチ処理を例に示し説明 した後、入力データ構造を変更し、モジュール分割を行う。HIPOあ るいはデータ・フロー記述で表現し、評価し合う。なお、例とする処理 はソートを含んだ3モジュール程度の簡単なバッチ・ジョブが適当であ る。

### (2) グループ演習

第1章で作成した詳細設計書をもとに、プログラムの構造化設計を行う。

### 育成方法および指導上の留意点

本章の1節「プログラム開発の手順」は、導入部であり全体を概観することが大きなねらいである。したがって、ここではあまり深入りしない方がよい。また、2節と3節は4節以降の説明の前提になる知識を与えようとしている。これらの節も簡潔な説明ですませる。

本章の中心は、4節と5節である。ここには十分時間をかけ、実践的能力が 修得できるよう演習を通じてじっくり指導してほしい。

# 用 語

プログラムの構造化設計上位モジュール、従属モジュール、上位セグメント、 従属セグメント、モジュール分割、独立性、階層構造化、階層モジュール構造 図、モジュール間インタフェース、源泉/変換/吸収分割(STS分割)、ト ランザクション分割、共通機能分割、モジュールの強度、モジュール間結合度、 ジャクソン法

### 第3章 プログラミング技法

#### 学習目標

この章を修了した時点で、学習者がプログラミング技法に関して、次の事柄 ができるようになることを目標とする。

- 1. 最適なモジュール論理設計技法を適用してモジュールの論理設計を行う。
- 2. モジュール論理設計の結果を構造図、構造化フローチャートなどでわかりやすく表現する。

#### 3.1 モジュールの論理設計(120分)

モジュールの論理設計の最大の主眼は、論理を構造化することである。そのためには、3つの基本制御構造(順次、選択、繰返し)で論理を組み立て、論理の中でGO TO命令が飛びかうことがないようにする(GO TO LESS)ことである。また、コーディングにさいしてもストラクチャード定理を遵守することが必要である。

- (1) モジュールの論理設計の考え方
  - ① 論理の構造化の目的
  - ② 構造化定理(ストラクチャード定理)
    - · 順次形(Sequence形)
    - ・選択形 (IF THEN ELSE形)
    - ・繰返し形 (DO WHILE形)
    - ・追加構造 (DO UNTIL形、CASE形)
- (2) モジュール論理設計の手順と要点
  - ① 論理設計の手順
    - (a) データ構造の分析……選択、繰返し
    - (b) データ構造の階層化……入力、出力データの構造化
    - (c) データ構造の対応づけ……入出力データの対応関係を明確化する
    - (d) 論理の記述……擬似コードなどを使って記述する
  - ② 論理設計の要点

- (a) セグメント化……制御セグメントとデータ処理・加工セグメントを 分ける
- (b) わかりやすい論理……構造化定理に基づいて論理展開する
- (c) 条件の明確化
- (d) 明解な表現……文章による曖昧な表現はさけ、デシジョン・テーブルあるいはフローチャートなどを用い表現する
- (3) コーディング技法-ストラクチャード・コーディング
  - ① PL/Iの場合
  - ② COBOLの場合
  - ③ コーディングの考慮事項

### 3.2 論理設計の代表的な手法(240分)

プログラムの構造化設計と論理設計の手法としては、伝統的なものとしてプログラム流れ図がある。しかし、構造化プログラミングが普及するにつれて、プログラム流れ図の限界や問題点が認識されるようになり、これに代る手法として様々なものが出現し利用されるようになっている。これらは、いずれもプログラムの構造やモジュールの論理を図式化して表現する手法であり、2章の4節と当章の1節で述べた構造化のアプローチを支える技法である(ただし、ほとんどの手法は論理の構造化が中心)

プログラム構造や論理を図式化する手法を大別すると、流れ図を中心にした表記法(プログラム流れ図、判断表、ワーニエ法など)、流れ図から脱却した新しい表記法(ジャクソン法、NSチャート、擬似言語、HIPOなど)、新表記法の特別なものとしての木構造チャート(PAD、HCPチャート、YACII、SPDなど)の3つになる。これら手法のうちの代表的なものとして以下のものを取り上げる。

- (1) 構造化プログラム流れ図
  - ① 構造化流れ図の書き方
  - ② 構造化流れ図の具体例
  - ③ 流れ図表記による問題点
- (2) 判断表(決定表、デシジョン・テーブル)

- ① 判断表とは
- ② 判断表の構成図
- ③ 構造化プログラミングと判断表
- ④ 判断表による具体例
- (3) ワーニエ法
  - ① ワーニエ法とは
  - ② ワーニエ法によるプログラムの作成手順
  - ③ ワーニエ法によるブレークダウン
  - ④ ワーニエ法によるプログラム流れ図
  - ⑤ ワーニエ法による具体例
- (4) ジャクソン法
  - ① ジャクソン法とは
  - ② ジャクソン法の手順
  - ③ 構造図の基本形
  - ④ データ構造図の書き方
  - ⑤ プログラム構造図の作成と具体例
- (5) NSチャート
  - ① NSチャートとは
  - ② NSチャートの記述のしかた
  - ③ NSチャートによる具体例
- (6) 擬似言語 (プログラム記述言語)
  - ① 擬似言語とは
  - ② 擬似言語による表現の仕方
  - ③ 擬似言語による具体例
- (7) HIPO
  - ① HIPOとは
  - ② HIPOの構成(図式目次、IPOダイヤグラム)
  - ③ 図式目次の書き方
  - ④ IPOダイヤグラムの書き方
  - ⑤ HIPOによる具体例

- (8) PAD
  - ① 大構造チャートとは
  - ② PADの特徴
  - ③ PADの基本図式
  - ④ PADの標準図式
  - ⑤ PADによる図式化の手順と具体例

#### 3.3 プログラミング技法演習(360分)

(1) 個人単位の小演習

2章で分割したモジュールの論理設計を行う。セグメント分割し、セグメント構造図を記述する。このとき制御セグメントを多階層にした構造図は論理が分りにくくなるため、さける。

構造化した論理を次の技法で表現する。

- ① 構造化定理に従ってフローチャートで記述する。
- ② 同様に擬似コーディングする。
- (2) グループ演習

2章で定義したモジュールの論理設計からコーディングまでを行う。

# 育成方法および指導上の留意点

1節の「モジュールの論理設計」では、論理の構造化(図式化)の手法は余り具体的にしていない。したがって、2節の「論理設計の代表的手法」の説明を終了してから、本格的な演習に取り組ませる。なお、2節では、すべての代表的手法を取り上げる必要はない。それぞれの環境諸条件を勘案して、重要とみなされる手法だけを選択して、指導してほしい。

# 用 語

モジュールの論理設計、ストラクチャード定理、基本制御構造(順次、選択、 繰返し)、DO UNTIL形、CASE形、データ構造の分析、セグメント化、構造化プ ログラム流れ図、判断表、ワーニエ法、NSチャート、擬似言語、木構造チャート、GO TO LESS

# 第4章 プログラムのテストと品質管理

# 学習目標

この章を修了した時点で、学習者がプログラムのテストと品質管理に関して、 次の事柄ができるようになることを目標とする。

- 1. 品質管理の重要性とその技法の種類や内容を説明する。
- 2. ウォークスルーとインスペクションの目的と適用の仕方を説明し、それ ぞれの適用段階でその技法を活用する。
- 3. プログラム・テストの種類を列挙し、それぞれのテストの特徴を説明する。
- 4. プログラム・テストの計画を作成し、テスト・ケースの設計をする。
- 5. 統合テストの種類と進め方、およびそれらの長所と短所を説明し、テスト環境にふさわしい手法を選択してテストを行う。

# 内 容

# 4.1 品質管理とその技法(60分)

本節では、品質管理ないしは品質評価とそれに用いられる技法を概観する。 品質管理については、その必要性と評価の着眼ポイントを説明する。管理技法 としては、ウォークスルーとインスペクションおよびプログラム・テストを取 り上げる。

- (1) 品質管理の必要性
- (2) 品質管理の方向
  - ① エラーの防止
  - ② エラーの検出
- (3) 品質管理の技法
  - ① ウォークスルー/インスペクション
    - ・目的と特色
    - ウォークスルーとインスペクションの相違

#### ② プログラム・テスト

# 4.2 ウォークスルー/インスペクション技法(60分)

ここでは、品質管理技法のうちのウォークスルーとインスペクションの技法 の活用の仕方を指導する。ウォークスルーとインスペクションとは本質的には 同じ技法といってよいので、本節では両者を1つの技法として一緒に取扱う。

まず、ウォークスルー/インスペクションの技法を適用するにふさわしい開発段階と適用上の考慮点について説明する。その後で、具体的な進め方とそれぞれのステップでの考慮点を明らかにする。

### (1) 適用段階

- ① プログラム構造化設計段階(その目的、出席対象者、事前配布資料など)
- ② モジュールの論理設計段階(その目的、出席対象者、事前配布資料など)
- ③ モジュールのコーディング段階(その目的、出席対象者、事前配布資料など)
- (2) 進め方と留意点
  - ① 事前準備の内容と留意点
  - ② 打合せ (ミーティング) の内容と留意点
  - ③ 事後処理の内容と留意点
- (3) 演習問題 (グループ演習)

# 4.3 プログラム・テストの種類と特徴(60分)

本節では、次節以降の前提になるプログラム・テストの種類とそれぞれの目 的や特徴を説明する。

- (1) 単体テスト (モジュール・テスト)
- (2) 統合テスト(つなぎテスト)
- (3) システム・テスト (総合テスト)
- (4) 承認テスト(引渡しテスト)
- (5) 導入テスト

#### 4.4 テスト計画とテスト・ケースの設計(120分)

ここでは、テスト計画の内容と作業手順、およびテスト計画において主要な 作業となるテスト・ケース設計の技法を指導する。

- (1) テスト計画の作成手順
  - ① テスト・ケースの設計
  - ② モジュールのテスト順序の決定
  - ③ テスト・ケースのテスト順序の決定
  - ④ テスト・データの生成方法の決定
- (2) テスト・ケースの設計技法
  - ① テスト・ケース設計の考え方と手順
  - ② テスト・ケース設計の技法-因果グラフ (cause and effect graph)
    - ・手 順
    - ・論理関係の表現
    - ・具体例
- (3) テスト・ケース設計の演習問題

# 4.5 プログラム・テストの手法(60分)

プログラム・テストのうち最も困難を伴うのが、統合テストである。本筋では、統合テストを中心にテストの種類とそれぞれの進め方を説明する。また、デバッグの手法についてもふれる。

- (1) 統合テストの種類と特徴
  - ① ボトムアップ・テスト
  - ② トップダウン・テスト
  - ③ 折衷テスト
- (2) ボトムアップ・テストの進め方
  - ① テストの進め方
  - ② 長所と短所
- (3) トップダウン・テストの進め方
  - ① テストの進め方
  - ② 長所と短所

- (4) 折衷テスト
  - ① テストの進め方
  - ② 長所と短所
- (5) プログラム・デバッグの手法
  - ① 机上デバッグ
  - ② プログラム・デバッグとデバッグ・ツール
  - ③ 対話式デバッグ機能

#### 4.6 プログラムのテスト技法演習(960分)

(1) 個人単位の小演習

因果グラフを用い、テスト・ケースを作成する。作成されたテスト・ケースを持ちより、何人かで評価し合い、余分なケースあるいは抜けを見つけ出す。

(2) グループ演習・実習

1章の詳細設計から3章のコーディングの各段階でウォークスルー/インスペクションを実施し、各段階で発見したエラーを記録していく。さらにテスト・ケースを作成し、モジュール・テスト、統合テストまでを一貫して行う。

# 育成方法および指導上の留意点

4.1 節は導入部であり、全体を概観することが狙いであるので、余り深入りしない。4.2 節以降が各論であり、4.2 節はウォークスルー/インスペクションの技法を指導する。4.3 節以降は、プログラム・テストを取扱う。

4.2節の演習はグループ演習を主体にして、ウォークスルー/インスペクションを体験させる。事前配布資料をあらかじめ全学習者に配布し、各人に資料の中の欠かんを発見させてから、グループに分かれてレビュー・ミーティングをやらせ検討させる。

4.4 節のテスト・ケースの設計演習は、各人でまずテスト・ケースを作成し、 グループで集合してウォークスルー方式でそれぞれが作成したテスト・ケース をレビューし、検証する指導方法が効果的である。

用語

ウォークスルー, インスペクション, 単体テスト, 統合テスト, システム・ テスト, 承認テスト, 導入テスト, 因果グラフ, ボトムアップ・テスト, トッ プダウン・テスト, 折衷テスト

# 第5章 システムの導入と運用

#### 学習目標

この章を修了した時点で、学習者がシステムの導入と保守に関して、次の事柄ができるようになることを目標とする。

- 1. システムの導入の手順と作業内容を説明する。
- 2. システムの保守がシステムのライフサイクルを伸長させる重要な作業であることを認識し、最適な保守体制、作業を説明する。
- 3. システム全体の達成度の評価と開発工程の評価の2種類の評価の内容を 説明する。

### 内 容

### 5.1 システムの導入(移行)(60分)

システムの構築が終了すると、開発されたシステムがユーザの要件を満たしているかどうかを確認する。この検収が済むと、運用を開始するためにシステムの導入が行われる。このシステム導入は新システムに慣れることと、既存システムとの整合性のチェックとを主に行う。

- (1) 運用マニュアルの作成
  - ① 運用手順マニュアル(ユーザ部門用) ユーザに対して、システム運用上に必要な情報を与えるためのもので、 システムの利用方法やシステムと対話する方法を示す。
  - ② 作業手順マニュアル(情報システム部門用) システムの運用管理を行う部門用に作成するもので、支障なく働くた めに必要な作業手順や方法を記載する。
- (2) ユーザ教育

教育は設計段階から準備を進め、システムの導入期間中に教育が終了するようにスケジュールを立てる。おもな教育内容は次のようなものである。

① システムの概要、目的

- ② 変更された機能、新しく作られた機能に重点を置いた事務処理方法
- ③ すべての取引に対する端末操作法
- ④ 専任オペレータ教育
- (3) マスタ・ファイル作成

既存システムのファイルを新システム用のファイルに変換あるいは新規 作成する。ファイルの移行は次の2つの方法がある。

- ① 短期間にマスタ・ファイルを整備する……システム稼働とともに全機 能が正常稼働できる
- ② 運用しながらマスタ・ファイルを整備していく……システム稼働直後 は一部の機能が働かないが、整備されるに従い全機能が稼働していく
- (4) 並行ランの実施

新旧システムを一定期間同時に並行的にランする。既システムからの移 行時にトラブルが生じる可能性を少なくするために必要なことである。

#### 5.2 システムの保守(60分)

システムへのあらゆる要求を、常に運用可能な状態に維持していくことが保守の役割である。保守が不要なシステムは理想的であるが、現実にはそのようなシステムは存在しない。年々、保守作業の量は増加し費用も増大している。保守作業は絶えず行われるものではないが、無計画な場あたり的な作業では保守どころかシステムを破壊しシステム・ライフを縮めてしまう。

#### (1) 保守体制

保守体制は業種や企業形態によっていろいろな形をもっている。次に示す標準的な二つの体制が存在する。

- ① 業務別の保守体制
  - 業務単位に独立した保守のチームを作り、業務に関連する部分の保守 作業を行う方式である。責任者には業務に精通した者を配する。
- ② システム単位の保守体制

システム(あるいはサブシステム)ごとに保守グループを作り、その システムが続くかぎり保守管理する方式である。リーダにはシステムの 内容をよく知っている者が設計段階で選ばれ、構築作業を通して保守の 準備をしておき、運用に際しては保守作業量を見積って要員の決定も行う。

#### (2) 保守作業

保守作業の多くは移行に際しての不整合によって引き起される。作業量は運用開始直後が最も多く、システムが安定するに従って少なくなる。

- ① 修正作業……バグという名の不整合の修正作業。優先度は最も高く早 急に対処すること
- ② 変更作業……法律の改正、金利の変更などの環境条件の変化への対応作業である。保守計画においては最優先作業である。
- ③ 改良作業……ユーザ部門からの処理追加や機能追加の要求や、不十分 な評価を下された処理機能の改良に対応するもの。保守本来の仕事であ る。
- (3) 保守作業へのアプローチ

障害復旧のための修正作業と、システムの変更や改良作業とではアプローチ方法が異る。

- ① 修正作業のとき……原因がわかったときすぐに修正を行わず、原因分析を行うことである。安易な修正は新たなバグを生む。
- ② 変更・改良作業のとき……他人の作ったプログラムを改良することであり、要求をどのように受入れるかその方法を考案することが大切である。

以上のアプローチ方法の共通性は、仕様やプログラムの変更を加えると きの変更管理の重要性である。

#### 5.3 システムの評価(60分)

システムを開発するには、効果が費用を上回わり、ユーザの満足したものでなくてはならない。当然の事であるが、システム開発の一部分にまみれていると忘れがちになる。初級情報処理技術者といえど、上級を目指すために、原価意識を持ち、作業終了後の結果を自問自答することは必要である。

(1) システム全体の達成度の評価 計画段階で設定した目標に対して新システムがどの程度達成できている かを評価すること。そのためには目標は明確にされていなければならず、 目標を項目ごとに細分化し定量化しておくことも必要である。これは評価 の客観化と精度の向上につながる。

### ① 経済的目標に対する評価

システム開発に投入されたコストと、新システムの運用によって得られる経済的効果目標とを対比して行うものである。コストの算出は容易でもシステム効果中の間接的効果は金額として算出しにくい。

#### ② 性能的目標に対する評価

システムの性能・機能面における目標に対し、定量的・定性的に評価を行う。

#### 図12-5-1 システムの評価

#### 評価方法

<sub>(</sub>定量的評価……金額・数量・回数・率など,数値に換算して数量的な評価を行う ・定性的評価……数値に換算することが困難な項目に対し,質的な評価を行う

#### 評価項目

, システム費用 { 開発費(直接費+間接費)……調査・設計・開発の人件費, ほか 運用費(直接費+間接費)……機器使用料, 人件費, 運用経費,

#### 経済的評価

直接的効果(金額計上が可能なもの)……費用の削減、利益の増 大、ほか 間接的効果(金額計上が不可能なもの)……促進効果、イメージ 向上、ほか

<sub>(</sub>効率性······応答時間,アクセス・タイム,ターンアラウンド・タイム,処理件数, ・ 能率、ほか

適時性……出力タイミング、提供情報の満足度、ほか

#### 性能的評価

正確性……誤処理発生件数,誤インプット率,誤差率,有効桁数,ほか 信頼性……処理の信頼度・故障率,平均故障間隔(MTBF)による信頼度、ほか

柔軟性……システムの変更・拡張の容易さ、例外処理への対応の容易さ、ほか

安全性……事故に対する安全配慮の範囲と度合い、ほか

機密性……機密保護手段の実施範囲と度合い、ほか

#### (2) システムの開発工程の評価

目標達成のための作業工程の評価である。工程の評価は情報システム部門の内部資料として残されるものである。また、次の開発への指針ともなる。

- ① 開発日程・工程の評価
  - (a) 計画における日程と工程の見積りの妥当性を評価する
  - (b) 予定の日程・工程と実績に大差があるときは原因を明示する
  - (c) 予定より早いところと、遅れているところを共に明示する
- ② 開発コストの評価
  - (a) 開発コストは正しく見積られているか
  - (b) 費用項目ごとの開発コスト(集積)のチェック
  - (c) 見積りと開発コストとに差があるときは理由を明示する
- ③ コンピュータ使用時間の評価
  - (a) 全般的なコンピュータ使用時間は適切か
  - (b) 各工程(単体テスト、結合テスト、総合テスト)別のコンピュータ 使用時間の中で、異常な値を示しているものの原因をつかむこと
- ④ 障害の分析評価
  - (a) 原因別の障害発生件数により、どの原因に障害が集中するかを知る
  - (b) サブシステムまたはそれ以下の作業単位で発生件数をまとめ、障害 集中のサブシステムがあるかを知る
  - (c) サブシステム、作業単位ごとにエラー修正工数をまとめ、エラー修正する作業単位をみつけ、作業のしかたの問題を修正する
- ⑤ 残留問題点の評価

新システムで吸収できていない問題点を明示しておく。これは改良作業などで吸収できるかを評価しておく。

# 育成方法および指導上の留意点

本章では、講義中心で進めざるを得ないが、発問したり、例を示すなどしな がら進めてほしい。尚、評価の一側面である開発日程の評価、障害の分析など については、グループ演習の結果をまとめ、成果発表の1テーマとする。 用語

運用マニュアル,作業手順マニュアル,ファイルの移行,並行ラン,保守体制,達成度の評価,経済目標に対する評価,性能目標に対する評価,開発工程の評価,障害分析の評価

### 第6章 システム開発の進め方

### 学習目標

システム開発を進めていくうえで、技術面のみならず、開発組織の確立、開発管理の重要性を認識する。この章を修了した時点で、学習者がシステム開発の進め方に関して、次の事柄ができるようになることを目標とする。

- 1. システム開発に必要な推進組織と担当組織のあり方、開発担当組織を構成する要員の職種を説明する。
- 2. システム開発管理の中心となる日程計画の立案、工程管理と工程遅延の対策を説明する。

# 内 容

#### 6.1 システム開発と組織体制(60分)

システム開発に必要な組織として、ユーザ部門による開発推進組織と、実際 に開発を行う開発担当組織とがある。

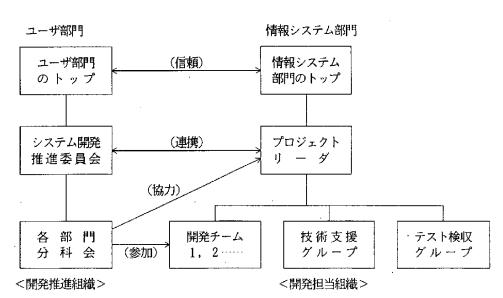


図12-6-1 システム開発組織

#### (1) システム開発推進組織の機能

システムの成功、不成功を判断するのはユーザ部門である。ユーザ部門 はシステム開発に深くかかわることが必要で、最大の効果をあげたシステ ムは、ユーザ部門が大いに関与したシステムであるといわれる。

① ユーザ部門のトップ 開発推進組織の最高責任者であり、システムを成功させる責任と開発 活動を拒否する権限を持っている。

#### ② システム開発推進委員会

各ユーザ部門の代表者からなる委員会の主な機能はシステム分析と要求定義(基本計画)の支援であり、各段階で作られたドキュメントのチェックと承認である。

#### ③ 各部門の業務熟達者による分科会

システム開発の対象が大きいとき、委員会の分科会で具体的な手段や 方法について協議を行う。メンバーとしては、主任・係長・課長などの 職名を持つその部門の主流の人物であることが望ましい。この分科会か ら、開発担当組織の開発チームに参加する者が選ばれる。

#### (2) システム開発担当組織の機能

開発の担当はユーザ部門がみずから行うべきであるという考え方があるが、ユーザ部門にシステム開発の経験者が少ない現在では情報システム部門の要員を中心とした開発チームが行うことが妥当である。しかし、情報システム部門はコンピュータ処理には精通しているがユーザ部門の業務に対する知識・経験・認識に欠けているため、開発されたシステムがコンピュータ寄りとなりユーザ部門の現場で扱いにくいものになるおそれがある。これを解消するためには、ユーザ部門から(分科会のメンバー)業務精通者が参加することである。

### ① 情報システム部門のトップ

大規模かつ複数のユーザ部門に係わるシステム開発の場合、システム 開発担当組織の責任者としてユーザ部門のトップと信頼関係を持って開 発プロジェクトを進める。

# ② 開発プロジェクト・リーダ

開発の牽引役で、開発担当組織全体の責任と管理が任務である。リー ダとして行うべきことを列挙すると次のようになる。

- ・メンバーの教育訓練、指導、権限の委譲および動機づけを行う
- ・開発担当組織内の開発チーム、技術支援グループ、テスト検収グルー プなどの融合を図る
- ・システム開発の各段階で、チーム要員の再配置を行い組織の強化をは かる。

#### ③ 開発チーム

計画から運用・評価までを行う中心チームである。メンバーは下記の通り。

- 業務経験者のアプリケーション・エンジニア(AE)
- ・システム開発経験者のアプリケーション・エンジニア(AE)
- ・詳細設計を主に行うプロダクション・エンジニア(PE)
- プログラムを作成するプログラマ(PG)

#### ④ 技術支援グループ

概要設計、詳細設計、構築時のプログラミングの標準化、データベースやデータ通信の技術などの開発チームを支援する技術を持つグループである。

- ・社内外の経験に基く開発技術を持つテクニカル・エンジニア(TE)
- ・パッケージやアプリケーション・プログラムに精通しているディベロップメント・エンジニア(DE)

# ⑤ テスト検収グループ

開発チームから独立して詳細設計の時期からテスト・ツールを用意 し、テスト項目からテスト・データを作成する。このグループは高度 な技術を持つPE、PGによって構成される。

# 6.2 システム開発の管理(60分)

・ システム開発の失敗の多くは、プロジェクト管理の不備、なかでも計画の不 備に起因することが多く、特にその大半は過小見積りによるものである。

- (1) 日程計画の立案
  - ① システム開発の各作業を細分化し、さらに構造化する(作業構造図、 作業一覧表)
  - ② 作業工数の見積り(類似作業からの見積り、3点法見積り)
  - ③ 作業担当者と責任者の明確化
  - ④ 日程表作成(PERT、ガントチャート)
- (2) 工程管理の要件
  - ① 正確な工数見積り
  - ② テスト検収グループの存在
  - ③ レビュー工程の設定
  - ④ 定期的で簡単な報告制度の設定
- (3) 工程管理の実施
  - (1) 管理項目の設定(作業、作成ドキュメント、プログラム)
  - ② 進捗状況のデータ収集
  - ③ 進捗状況の表現方法(消化率管理、ディジタル管理)
- (4) 工程遅延の対策
  - ① 遅れ原因の調査
  - ② 工程配分の見直し、遅れ要因の除去
  - ③ 要員の補充
  - ④ 作業の効率化

# 育成方法及び指導上の留意点

1. ユーザ部門の開発推進組織と、情報システム部門の開発担当組織の、各々の役割分担を認識させる。又、担当組織の職種であるSE(AE、PE、TE、DE)やプログラマの役割についても認識させる。

組織の作り方や、メンバーの条件などは形だけ示されても理解できないものである。現実に存在している例を詳しく論証することが望ましい。できれば異なる複数の組織例を比較するとよい。組織について机上で学ことには限界があり、実際に組織の中で働くことによって正しい認識が可能となる。

2. 日程計画を立案するには作業の細分化や作業の見積りなどの技術を要し、 初級情報処理技術者には難しい。したがってグループ演習時、指導者がプロ グラム開発の日程を提示し、実績を記入させる程度にする。予実績の分析か ら、個々の作業の工数配分、遅れ原因の討議などを通じて開発管理の一端を 経験させる必要がある。

### 用語

システム開発推進組織、システム開発担当組織、アプリケーション・エンジニア(AE)、プロダクション・エンジニア(PE)、テクニカル・エンジニア(TE)、ディベロップメント・エンジニア(DE)、プログラマ(PG)、PERT、ガントチャート

# 参考文献

- 「ソフトウェアの生産技法」
   菅野文友 編, 日科技連, 1987年
- 2. 「効果的プログラム開発技法」 図友義久 著,近代科学社,1987年
- 3. 「高信頼性ソフトウェア複合設計」 Myers 著,久保未沙・國友義久訳,近代科学社,1985年
- 4. 「コンピュータ情報処理論」 前川良博 著,白桃書房,1989年
- 5. 「情報システムの開発と設計」 (株)東和コンピュータマネジメント 編, 啓学出版, 1987年
- 7. 「システム開発と運用一開発編一」 (財)日本情報処理開発協会中央情報教育研究所 編, 1987年

# 第13部 表現技法

#### 教育目標

日本人なら誰でも日本語で文章を書き、日本語で話しをすることができる。 しかし、誰でもわかりやすい文章が書けるわけではない。むしろ、書けない人 の方が圧倒的に多い。話し言葉も同様である。わかりやすい正しい日本語で話 すことはけっこうむずかしい。そのため、相手のレベルや関心に合わせたわか りやすい話し方をする人は多くない。

なぜわかりやすい文章表現ができないのだろうか。なぜわかりやすく正しい 口頭表現ができないのだろうか。その理由はいろいろ考えられる。最大の理由 は、基本をふまえたちゃんとした訓練がなされていないことだ。本来なら、学 校教育の中で文章表現や口頭表現の基本をきちんと指導すべきである。実際に はそれが十分なされていない。そこで、表現技法の必要性を最も痛感している 企業で指導せざるを得ないのが現状である。

ところで、情報処理技術者にとってみると、一般のビジネスマン以上に表現技術が重要である。日常の文書作成はもちろんのこと、顧客やユーザ部門あるいはトップ層に対する各種の提案書を作成する必要がある。要求定義書やシステム設計書等の各種システム仕様書をまとめる必要がある。さらには、マニュアルを作ったり、技術論文や報告書を作成する必要もある。口頭表現としては、提案書にもとづいて提案説明を行い、相手を説得する必要がある。また、仕様書やマニュアルをもとに関係者に説明し、正しく理解させる必要がある。それだけに、表現技術は情報処理技術にとって欠くことのできない基本技術である。

初級情報処理技術者が提案書やシステム仕様書を作ることはまずないだろう。 また、それらを用いて説明することもまれだろう。したがって、当部では提案 書や仕様書の作成技法といった具体的な技法までは取り扱わない。表現技法の 基本を体得してもらうのが最大のねらいである。具体的には、次の3つの事柄 を学習し、それらをちゃんと応用できるようになってもらうことが、当部の教 育目標である。

① 文章の書き方:正しい日本語でしかも適切な表現の仕方で、わかりやす

い文章を書くことができる。

- ② ビジュアル表現の仕方:単に文章表現だけでなく、図表(グラフ)や図 解、イラスト的な表現をすることができる。
- ③ 話し方の技術:聞き手の立場を考慮に入れたわかりやすい効果的な話し 方をすることができるようになる。

上記の目標を達成するためには、ふんだんに演習をする必要がある。ごく基本になることだけを講義で教える。演習では、その基本がちゃんと応用できるまで訓練する。

# 内容の構成

内容	時 間 数	
	講義	演習
第1章 文章の書き方 1.1 わかりやすい文章を書くためには 1.2 わかりやすい表現をするための工夫 1.3 用字・用語の使い方 1.4 文章の書き方 1.5 文章を書く前の準備 1.6 文章構成のポイント	3. 0	9. 0
第 2 章 ビジュアル表現のしかた 2.1 ビジュアル表現の効用 2.2 ビジュアル表現の種類 2.3 図表(グラフ)の書き方 2.4 図解の仕方	2. 5	1 1. 0
第3章 話し方の技術 3.1 話し方の重要性 3.2 話し方の基本 3.3 効果的な話し方 3.4 基本的配慮点	4. 0	1 4. 5
合 計	9. 5	3 4. 5

## 第1章 文章の書き方

## 学習目標

当部を受講された方々が、この章を修了した時点で、文章の書き方に関して、 次の事柄ができるようになることを目標とする。

- 1. 簡潔で明快なわかりやすい文章を書く。
- 2. 正しい用字・用語を用いた文章にする。
- 3. 起承転結のある筋の通った文章を作成する。
- 4. 十分構想を練ったよい文章構成を作り上げる。

## 内 容

### 1.1 わかりやすい文書を書くためには(120分)

(1) わかりやすい文章とは

わかりやすい文章について、昭和初期のジャーナリスト千葉亀雄氏(東京日々新聞学芸部顧問)は、次の3点を指摘しているとのことである。

- ① ハッキリ書け。すれば人は理解する。
- ② 短かく書け。すれば人は好んで読む。
- ③ 絵で書いたように書け。すれば人は記憶する。 これを肝に銘じて指導することが大切である。わかりやすい文章とそう でない文章を対比して示し、上記3つの観点から「わかりやすい文章」を 具体的に印象づけること。
- (2) 話すことと書くことの違い

書くことと話すこととの間には本質的な差がある。このことをわきまえて、文章を書く必要があることを指導する。

話す場合は、相手が目の前にいる。したがって、相手の反応や表情をみながら話を進めることができる。それに、相手の関心やニーズ、性格や地位などを知っている場合が多い。それだけに、おたがい話の内容を理解しやすく、不十分な説明でも内容を正しく伝えることができる。

一方、文章では、違った考えをもった人や違った立場の人に文字だけで情報を伝達しなければならない。それに、会話の時のように、相手の反応や表情をみながら情報を伝えることはできない。それだけに、伝達すべき情報を順序よく整理し、誤解をまねかないような言葉を使う必要がある。

## (3) やさしい言葉、使いなれた表現を使う

文章を書くさいには、とかくかしこまった言葉、高尚な言葉を用いようとし勝ちである。これが、かえってむずかしいわかりにくい文章にしてしまう。やさしい言葉、日頃使いなれた表現を使うと、ずっとわかりやすい文章になる。

## (4) 読む人の立場に立って書く

読む人の関心やニーズ、悩みや問題点および知識レベル等を十分配慮して、文章を書く必要がある。文章は、読者に<読む気>を起こさせなければならない。

### (5) 漢字を多用しない

漢字と仮名の割合が2対3位の割合になるようにする。漢字がこの割合いを越えると読みずらくなる。

漢字が5割近くの文章と35%程度の文章を対比して説明すると、説得力がでる。

## (6) 演習

- 「わかりやすい文章の書き方」というテーマを与えて、800 字で文章を 書かせる。
- ・「自宅(下宿先)から乗車駅迄の道順をわかりやすく文章で書け」といったテーマを与えて、文章を書かせる。

上述のようなテーマを 2 、 3 題与えて演習をさせる。完成した文章は添削して各人にフィードバックする。できたら、数人の文章を O H P で提示して、どのように改善すべきか受講者に検討させるとよい。

## 1.2 わかりやすい表現をするための工夫(240分)

(1) 簡潔な文章を書くための工夫

文章は簡潔明瞭を旨とする。簡潔にするための工夫のポイントは、次の

5点に絞られる。

- ① 短かめのセンテンスにする(原則として50文字以内)。
- ② 形容詞や修飾語を多用しない。程度は、具体的な数字で示す。
- ③ 意味の曖昧な表現は避ける。
- ④ 抽象的な表現は避ける(どのようにでも解釈できる)。
- ⑤ 美辞麗句は用いない。

上記それぞれに関して、良い例と悪い例を対比して説明する。また、悪 い例を提示して、受講者に訂正させる。

(2) 明快な文章を書くための工夫

上記5項目に加えて、明快な文章にするためには、次の工夫が必要である。

- ① 文章に区切り(段落)を作り、まとまりのある要旨を目立たせて書く。 まず主題をはっきりさせ、その主題を述べるための要点を小見出しとし て立てる。その小見出しのもとで、それに関する内容だけをまとめる。
- ② 主張を首尾一貫させる。主張は論理的に正しく、筋の通ったものでなければならない。「なぜ」と聞かれて困るような文章にしない。単なる印象や風間でものを言っては駄目である。裏付けのあるデータを生かすことである。

この項目の指導にさいしても、具体例(良い例、悪い例)を提示することが大切である。

(3) 「事実」に語らせる工夫

述べようとする内容を理解しやすくし、印象づけるためには、具体的な 事実や出来事を上手に利用する。また、主張を裏付けるためにも、「事実」 を生かすことが大切になる。

- (4) 主部と述部の関係をはっきりさせる 主部と述部の関係をはっきりさせるために、特に留意すべき点は、以下 の3点である。
  - ① ねじれた文章を書かない。
  - ② あと戻りする文章を書かない。
  - ③ 舌たらずな文章を書かない。

まず、いくつかの例を示して、主部と述部をはっきり理解させる。その あとで、上記3点のそれぞれに関して悪い例を示して、改善結果と対比し て説明する。さらに、悪い例を提示して、受講者に改善させる。

- (5) 句読点の打ち方、符号・記号の使い方 以下の諸点について、それぞれ具体例を示して指導する。
  - ① 句点一つの文が終ったら、句点(。)をつける。
  - ② 読点
    - ・ 誤読・難読の恐れがあるときに使う
      - 一 主題となる語につく助詞のつぎに
      - ― 修飾語が誤解される恐れのあるとき
      - 副詞の帰属を明らかにする必要があるとき
      - ― 修飾語の対象を明らかにさせるとき
      - 仮名や漢字がくっつきやすいとき
    - ・ 語句を並べるときや、対立節に使う
    - ・ 形容詞節と直接つぎに続く被修飾語の名詞との間には使わない
    - ・ 倒置文のあとに用いる
  - ③ 中点(中黒)
  - 4) 斜線
  - ⑤ カッコ
  - ⑥ かぎカッコ
  - ⑦ 連続符号
  - 图 点線
  - ⑨ 疑問符、感嘆符
  - ③~⑨の使い方については、「朝日新聞の用語の手びき」や「毎日新聞用語集」等を参考にしてほしい。

それぞれの項目について、間違った使い方を示して、受講生に訂正させる。これらの事柄は身をもって体験させないと、とても身につかない。

(6) 助詞、接続詞の使い方 つぎの3点を中心に、具体例を示して指導すること。

- ① 「は」の使い方
- ② 「が」の使い方
- ③ 同じ助詞を続けない

#### (7) 修飾する側とされる側

わかりにくい文章で、最も目につくのは、修飾する言葉と修飾される言葉とのつながりがはっきりしていない場合である。このつながりを明白に 意識しないまま、文章を書いている人が圧倒的に多いようである。

ここでは、修飾する側とされる側の関係を、具体例をもとに、十分認識 させる必要がある。また、文章をわかりにくくしている大きな原因が、修 飾する語とされる語が離れすぎていることにあることを理解させる。なお、 修飾語の並べかたで文章の意味が変わることを把握させることも大切であ る。

## (8) 修飾の順序

次の4点を原則にすることを具体例に基づいて説明する。

- ① 句を先に、形容詞をあとにする。
- ② 修飾語は、修飾する語句に密着させる。
- ③ 長い修飾語ほど先に、短かいほど後に。
- ④ 重要なものから重要でないものへ並べる。 この項目に関しても、それぞれ悪い例を示して、受講者に改善させる問題をやらせて訓練することが肝要である。

## (9) 段落の生かし方

段落は、かなりのまとまった思想表現(または、独自の要旨を持つまでにまとまっている文章)の単位である。そのまとまりの頭では、行を変えて一字下げて書き始める。これが段落である。行を改め一字下げて書けば段落になるのではない。実際には、段落がいい加減な文章が実に多い。文章が長くなったから、適当に段落を設けている。これは、欠陥文章である。1行だけでも、それが独自の要旨を持つ内容であれば、段落にすべきである。20行以上でやっと1つの要旨のまとまりになることだってある。それを長すぎるからといって複数の段落にすると、欠陥文章になってしまう。以上のことを念頭に、いろいろな具体例を示して指導すること。また、

誤った段落の文章を提示して、受講者に訂正させる練習問題をやらせる。

(10) 知っておくと便利なその他の文章技術

上記(1)から(9)以外で、知っておくと便利な文章技術を紹介する。例えば 人名・地名の書き方、敬語の使い方、数字の書き方など。

(11) 演習

次のような演習をできるだけ多くやらせる

- ① 「わかりやすい表現の文章」にするための基本原則に違反した文章を 提示して、違反個所を訂正させる。
- ② 「簡潔な文章を書くための工夫」など、わかりやすい表現をするための注意事項をテーマにして、800 字程度の文章を作成させる。
- ③ 「情報処理技術者をめざして」といったテーマを与えて、わかりやすい表現の文章を書かせる。

## 1.3 用字・用語の使い方(150分)

(1) 外来語(カタカナ)の表記の仕方

外来語 (カタカナ) 表記は、かなり厄介なものである。どの表記が正しく、どの表記が間違っている、ということが一概には言い切れない。「朝日新聞の用語の手びき」や「毎日新聞用語集」などの<外来語の書き方>の項を参考にして指導してほしい。なお、情報処理用語でJISに規定されているものは、これに従うように指導してほしい。

(2) 間違えやすい漢字

特に、次の4点に気をつける必要がある。

① 同音異字の誤り

例:・ご多聞にもれず → 多分

・顧客と応待する → 応対

② 形の似た字

例:・遺跡の発堀 遣→遺 堀→掘

·組識編成 組識 → 組織

③ 同じ字だが形が違うもの

- 例:・台と臺

#### ・体と體

ただし、この場合は、常用漢字表に採用されている字体で書くのが原 則である。

- ④ 人名や会社名には略字や平易な字は使わない。正式の名前で書く。 それぞれに関して代表的な誤りの問題を出題して、訂正させる。なお、 「間違いやすい漢字」に関しては、「朝日新聞の用語の手びき」の<誤 りやすい慣用句>の項を参考にして、指導してほしい。
- (3) 間違えやすい送り仮名

間違えやすい送り仮名には、「送り過ぎ」と「送り不足」の場合がある。

① 送り過ぎ

例:・失なう→失う

- 短かい→短い
- ・必らず→必ず
- ・身近か→身近
- ② 送り不足

例:・省る→省みる

- ・恐しい→恐ろしい
- 鮮か→鮮やか
- ・木枯し→木枯らし

「間違えやすい送り仮名」に関しても、「朝日新聞の用語の手びき」の <送り仮名表> および<誤りやすい送り仮名>の項などを参考にして指 導してほしい。ここでも、誤りの問題を出題して、練習させる必要がある。

(4) 間違えやすい仮名づかい

「現代仮名づかいの要領」について説明したうえで、誤りやすい仮名づかいとして、「ぢ・づ」と「じ・ず」の用例を示す。「現代仮名づかいの要領」については、「朝日新聞の用語の手びき」や「毎日新聞用語集」の <現代仮名遣い要領>の項などを参考にしてほしい。

ここでも、「ぢ・づ」と「じ・ず」の具体的な用法を例示して、説明する。また、現代仮名づかいとして正しいかどうかの問題を出題して指導する。

(5) 仮名書きが望ましい語句

以下のような場合は、仮名書きが望ましい(「毎日新聞用語集」より)。

① 活用語の語尾部分や接続詞、感動詞、助動詞、助詞など補助的に使われる語は、原則として仮名で書く。

例:・接続詞 一 あるいは、かつ、なお

- ・感動詞 ― ああ、まあ、はい
- ・助動詞 ごとき、べき、ようだ
- ・助 詞 一 くらい、ながら、など
- ② 常用漢字表外の字、甚だしい当て字に類する語は仮名書きにする。

例:挨拶→あいさつ 天晴れ→あっぱれ 日出度い→めでたい

③ 本来の意味の薄れた語は仮名書きにする。

例:・やってみ(見)る

- ・子供が遊んで(居)る
- ・聞いたこと(事)がない
- ・要らないとき (時) は返せ
- ④ 擬態語は原則として平仮名で書く。

例: のろのろ ぶらり ぬるぬる

⑤ 外来語のうち国語化しているものは原則として平仮名で書く。

例:かっぱ かるた きせる たばこ ここでの指導にさいしても、「朝日新聞の用語の手びき」の<漢字と 仮名の使い分け>や「毎日新聞用語集」の<漢字と仮名の使い方>の項 などを参考にしてほしい。

(6) 演習

次のような演習をできるだけ数多くやらせる。

- ① それぞれの項目に関して、誤った表記の文章ないしは望ましくない表 記の文章を与えて訂正させる。
- ② それぞれの項目に関する誤った表記や望ましくない表記のすべてを含んだ総合的な演習問題を与えて、訂正させる。

### 1.4 文章の書き方(150分)

(1) 体系的な整理

主題をはっきりさせ、体系的に整理することにより、章、節、項がきちんとする。具体的な例をあげて、文章の骨組みを作ってみせる必要がある。

## (2) 簡潔明瞭

これは、1.2の(1)でもふれたことであるが、ここで再度強調する。 「簡潔」といっても、短かければいいというものではない。必要な要素 はもれなく書かなければならない。このことも併せて強調してほしい。

### (3) 起承転結

文章の基本は、起承転結である。その趣で文章全体を構成する。

<起>は書き出しの部分で、読者を話に軽く引き込むようにする。<承>は展開の部分であり、読者の関心を高め、話を盛り上げ発展させる。<転>は、視点を変えて、自分の主張を展開する。起承はそのための準備段階である。<結>はいよいよ結論の部分である。ここで話をまとめ、読者を納得させる。

起承転結は、文章を簡潔明瞭にするための手段でもある。ここでは、いくつかの例示をしながら、起承転結のある文章が書けるように指導する。 たとえば、次のような例を用いるとよい。

### ① 例1

どんぐりころころドンブリコ(起) お池にはまってさあ大変(承) どじょうが出てきて今日は(転) 坊ちゃん一緒に遊びましょう(結)

### ② 例2

三条木屋町 糸屋の娘(起) 姉は十八 妹は十五(承) 諸国大名は 刃で殺す(転) 糸屋の娘は 眼で殺す(結)

## (4) 筋の通った文章(論理性)

これも、1.2の(2)でふれたことであるが、「文章の書き方」という視点

で、再度強調する。

(5) 裏付けがしっかりした文章 読者を納得させるためには、裏付けが必要である。裏付けとしては次の ようなものを使用するよう指導する。

- ① 統計的事実
- ② 権威者や有名人の言葉
- ③ 身近な事例
- ④ 自分の体験
- ⑤ 歴史的観察(歴史は繰り返す)
- (6) 感動を与える文章

読者に感動を与えることのできる文章が書ければ最高である。どうすれば読者の感性に訴えることができるかを指導する必要がある。

(7) 良い文章の要件

良い文章とは人を引き付ける文章である。その要件を整理すると、カンカラコモデケアになるという(これは新聞記者の教育で使用されている)。

- ・カン 感動 (ハットさせる、気づかせる)
- ・カラ カラフル (鮮明な印象を与える)
- ・コ 今日性(トレンディ、時代にマッチしている)
- ・モ 物語性 (ストーリがある、筋道がある)
- ・デ データ(裏付け、根拠がしっかりしている)
- ・ケ 決意(主張がある、論理が明解)
- ・ア 明るさ(ビジョンがある、明るい未来を感じさせる文章)

## (8) 演習

以下のような文章を書かせる演習問題を数題やらせる。この演習は、ここでまとめてやるよりも、説明項目と対応させた分散演習にした方がよい。ここでは総合演習問題をやらせる。

- ① 起承転結のある簡潔明瞭な文章を書かせる。
- ② 首尾一貫した論理性のある文章を書かせる。
- ③ 裏付けがしっかりした文章を書かせる。
- ④ 感動を与える文章を書かせる。

⑤ 良い文章の要件を備えた文章を書かせる(総合演習問題)。

### 1.5 文章を書く前の準備(180分)

### (1) 主題を決める

文章を書くためには、まず書く目的と内容が明確になっていなければならない。何をしたいのか、何を訴えたいのか、要するに何のためにこの文章を書くのかを明らかにする。つまり、まず文章の主題を決める。

ここでは、主題の確定の重要性を認識させる。具体的な主題を例示し、 (2)以降ではこの主題にもとづいて、具体例を展開していくようにする。

### (2) 要旨(粗筋)を考える

主題が決ったら、要旨を考え、筋書きを検討する。この段階では、大見 出しレベルで全体の粗筋を考える。

(1)で例示した主題をもとに、粗筋の具体例を示して、粗筋を検討するさいの留意点を指摘し、指導する。

### (3) 材料を選ぶ

粗筋を決めたら、それぞれの項目(大見出しに相当)の中にどんな話を 盛り込むかを考える。

まず、言いたいことを思いつくままに箇条書きにしてみる。これは、下書きなので、数や順番にこだわる必要はない。もれがないように洗出してみる。

次は、図に表現してみる。言いたいことを図に書いてみる。図解すると、言いたいことのまとまりや相互関係が見えてくる。自分の頭を整理し、自分でしっかり理解するためにも、図で表現してみることは欠かせない。これも下書きなので体裁にこだわらないで自由に表現してみる。

このようにして、書くべき要旨の材料を揃える。いくらきちんとした粗筋を考えても、中身がなければ文章は書けない。ここでは、材料を選定することの重要性を認識させると同時に、材料の選定の仕方を指導する。(2)で例示した要旨にもとづいて、どのような材料を選ぶべきかを具体例で示してやる。

### (4) 書く前に頭の中で「構想」を練る

(3)で選んだ材料をどのような順序に並べるか、どのようなストーリにするか、起承転結にそれぞれ何を持ってくるかなどの「構想」を練る。次に読者に共感や親近感を覚えさせるような、あるいは、裏付けになるようなエピソードや事実を考える。また、理解を助けたり、印象を深めるための「図解」や「具体例」を工夫する。

以上のことを、一貫した例示にもとづいて指導する。

### (5) 文章化の手順

これまで説明してきたことを、文章化の手順として、以下のように整理してやる。

- ① 主題を決定する。
- ② 要旨(粗筋)を決める。
- ③ 材料を選ぶ。
  - ・ 箇条書きにしてみる。
  - ・ 図に表現してみる。
- ④ 頭の中で「構想」を練る。
  - どんな順序に並べるかストーリを考える。
  - エピソード(ちょっとした話題)や事実を選択する。
  - ・ 「図解」や「具体例」を工夫する。
- ⑤ 文章を組み立てる。

いよいよ文章化である。④までで文章全体のイメージはできているので、あとは、わかりやすく簡潔明瞭に表現する。

#### (6) 演習

次のような演習を数題やらせる。

- ① 主題だけを与えて、受講者員体が同一主題で、たとえば1200字程度の 文章を作成させる。この演習では、でき上った文章を相互に比較して、 検討すると効果的である。
- ② 主題を含めて、受講者自身に選ばせ、上記の文章化の手順にしたがって、文章を作成させる。

## 育成方法および指導上の留意点

- ① この章においては、できるだけ良い例と悪い例を対比して説明するようにする。また、説明後には、悪い文章を示して受講者に訂正させる問題をやらせる。
- ② 講義では、文章の書き方の基本原則や基本的な配慮点に絞って説明する。余り欲ばった説明や細細とした内容には立ち入らないようにする。 要は、基本的な事柄がちゃんとやれることが重要である。
- ③ 演習結果は、面倒でも必ず添削してフィードバックしてやる。また、 演習結果の文章をいくつかコピーして配布するか、〇HPで映して、受 講者をまじえて添削するやり方も効果的である。演習時間が十分とれな い場合は、宿題にして翌日提出させるようにする。演習の手抜きをする と、教育効果は半減してしまう。演習は必ず実施するようにしてほしい。
- ④ 「文章の書き方」は、いろんな場面で指導することができる。受講期間中は、毎日受講日誌を書かせる。あるコースが終了すると、受講感想文を書かせる。職場に帰ったら、受講報告書を提出させる、等々。できるだけ、それらの添削をしてフィードバックしてやるとよい。

## 用語

主題,要旨,修飾語,修飾される語,主部,述語,句点,読点,被修飾語, 倒置文,中点(中黒),連続符号,助詞,接続詞,段落,用字,同音異字,慣 用句,常用漢字,常用漢字表,擬態語,起承転結,カンカラコモデケア,粗筋

## 第2章 ビジュアル表現の仕方

## 学習目標

当部を受講された方々が、この章を修了した時点で、ビジュアル表現の仕方に関して、次の事柄ができるようになることを目標とする。

- 1. ビジュアル表現の4つの主要な効用を列挙する。
- 2. ビジュアル表現の代表的な種類とそれぞれの使用目的を指摘する。
- 3. 使用目的にふさわしい効果的なグラフを作成する。
- 4. 使用目的にふさわしい効果的な図解(チャート化)を行う。

## 内 容

## 2.1 ビジュアル表現の効用(60分)

ビジュアル表現の主要な効用について、それぞれ具体的な例を示しながら説明する。

- (1) 印象の強化
  - グラフや図解は、数字の羅列や文章よりもはるかに印象が強い。したがって、上手に視覚に訴えると、一段と効果をあげることができる。
- (2) 要点の明確化 グラフや図解は、数表や文章より一覧性があるので、要点が明確になる。
- (3) 体系や関係の明示 グラフや図解は、全体の中の位置付けや相互の関係を示すのに適してい る。
- (4) 流れや変化の明示

グラフや図解は、傾向や推移、作用や影響など、流れの方向や変化の順 序を示すのに適している。

上記の4つの効用の他に、視覚に訴えることによって時間の節約ができる、 文章や数字だけでは表現できない内容のイメージ化ができる、などがあること も指摘しておくとよい。

### (5) 演習

4つの効用にそれぞれ対応したテーマ(文章で表現したもの、ないしは 数表)を与えて、それぞれにふさわしい図解ないしはグラフ化を行わせる。

### ? 2 ビジュアル表現の種類 (90分)

代表的なビジュアル表現の種類をグラフと図解(チャート)に大別して、目的別に説明する。それぞれのビジュアル表現の例を示しながら、その用途を把握させるように指導することが大切である。

- (1) グラフ '
  - ① 内訳を表現するグラフとその用途
    - 円グラフ
    - ・ 半円(相対半円)グラフ
    - 二重円グラフ
    - ・ 構成比率帯グラフ
    - 内訳構成比率棒グラフ
    - パレート図
    - 構成比層グラフ
    - 正方形グラフ
  - ② 大小比較、順位などを表現するグラフとその用途
    - 棒グラフ
    - 積み上げ棒グラフ
    - 重ね合わせ棒グラフ
    - 複合棒グラフ
    - 水平対称棒グラフ
    - ・ ピラミッド・グラフ
    - 偏差棒グラフ
    - 面積グラフ
  - ③ 推移を表現するグラフとその用途
    - 折れ線グラフ
    - 層グラフ

- 階段グラフ
- Zグラフ
- 動線グラフ
- ・ 浮動棒グラフ
- ・ ファンチャート
- ④ その他の目的を表現するグラフとその用途
  - ・ ヒストグラム
  - レーダーチャート
  - 損益分岐点グラフ
  - 点グラフ
  - 散布図
  - 三角グラフ

沢山の種類のグラフが出てくるので、ここで受講者の頭の中を整理させる必要がある。そのためには、各種のグラフを示してその名前と用途について発問する方法をとるとよい。

- (2) 図解(チャート化)
  - ① 相関関係を表現する図解とその用途
    - · 円交差図
    - 網目図
    - 二軸法(表型)図解
    - 二軸法(図型)図解
    - ダイヤモンド図
  - ② 構造を表現する図解とその用途
    - 焦点図
    - 拡散図
    - · 円多重図
    - 三角図
    - フローチャート
  - ③ 評価・判定を表現する図解とその用途
    - Jチャート

- SDチャート
- 順位チャート
- ④ 計画を表現する図解とその用途
  - ガント・チャート
  - パート図(アローダイヤグラム)
- ⑤ 整理・発見を表現する図解とその用途
  - ・ デシジョン・ツリー
  - · PDPC
  - 親和図
  - 系統図
  - 連関図
  - 特性要因図

ここでも多種類の図解(チャート)が出てくるので、受講者の頭の中を 整理させる必要がある。グラフの場合と同様に、各種の図解例を示してそ の名前と用途について発問するとよい。

(3) 演習

代表的なグラフと図解の具体例を示して、その名前と用途を答えさせる (発問によるのではなく、個人演習として実施する)。

#### 2.3 グラフの書き方(300分)

- (1) まず、書いてみよう 受講者のなじみの深い下記のようなグラフを、それぞれ数表を与えて書 かせる。
  - ① 折れ線グラフ
  - ② 棒グラフ
  - ③ 円グラフ
  - ④ 帯グラフ

受講者が完成したグラフのいくつかをOHPで投映し、その出来映えを 評価する。特に、改良すべき点や工夫すべき点を上手に指摘してやること。

(2) グラフ化の手順

グラフ化するためのステップとそれぞれのステップでの留意点を明確に する。

- ① 数値データを収集する。
- ② グラフ化の目的と用途を具体化する。
- ③ 目的・用途を念頭に数値データを整理・加工する。
- ④ グラフの形態を決定する。
- ⑤ レイアウトを決める。
- ⑥ 作図する。
- ⑦ 最終確認をする(数値の正確性、表現目的と表現形態、表現内容等)。 ステップの内容と留意点を示すだけでなく、特定の数値データを用いて 具体的な例示を行いながら、講義を進める必要がある。
- (3) グラフ化の基礎技術
  - ① 文字の使い方
  - ② 記号・矢印の選び方
  - ③ 線の選び方
  - ④ ハッチングのパターンの選び方
  - ⑤ アイソタイプ (絵文字) の使い方とピクトグラム (絵図表)
  - ⑥ 地図の使い方と地図グラフ ここで簡単な演習を行う(数値データを与えて、絵図表と地図グラフを 作成させる演習)。
- (4) グラフ化する際の主要な考慮点
  - ① 座標とその使用法
  - ② 強調の表現技法
  - ③ 気をつけたい表現(錯覚しやすい表現、誤解されやすい表現、わかりにくい表現)

それぞれ具体例を示しながら指導する必要がある。

(5) 目的・用途に応じた表現の工夫

2.2の「(1)グラフ」で紹介したグラフの具体例を再度提示し、表現上の 工夫と作成上の留意点を説明する。さらに、代表的グラフについては、それに工夫を加えた発展型のグラフを示して説明する(たとえば、円グラ フの発展型グラフとしてのドーナツ・グラフや円グラフのある要素の内訳 を見せる別の円グラフ)。

#### (6) 演習

目的別にそれぞれ別個の数値データを与えて、下記の目的に最もふさわ しいグラフを作成させる。

- ① 内訳を表現するためのグラフ
- ② 大小比較をさせるためのグラフ
- . ③ 推移を表現させるためのグラフ
  - ④ 全体の傾向(バラツキ)や特異性を評価するためのグラフ 作成したグラフのいくつかをOHPで提示し、受講者を交えて評価し合 う。

あるいは、6人程度のグループに分かれて、それぞれの成果物を提示して、 評価し合う。

### 2.4 図解 (チャート化) のしかた (360分)

(1) まず、書いてみよう

箇条書きの文章を与えて、受講者に比較的なじみの深い下記のような図解をさせる。

- ① 網目図
- ② 焦点図
- ③ パート図(アローダイヤグラム)
- ④ 系統図

受講者が完成したグラフのいくつかをOHPで投映し、評価する。特に、 改良すべき点や工夫すべき点を上手に指摘してやることが大切である。

- (2) 図解 (チャート化) の手順
  - 2.2の(2)で示したような図解の形態(パターン)の中からふさわしいものを選んで適用する場合は、下記の手順に従うように指導する。
  - ① 資料収集(言語データを収集する)。
  - ② 図解の目的・用途を具体化する。
  - ③ 目的・用途に基づいて、言語データを整理・加工する(要約化、短文

化)。

- ④ 図解の形態を選択する。
- ⑤ レイアウトをする。
- ⑥ 図解する。
- ⑦ 最終確認をする(表現の正確性、表現目的と表現形態の妥当性、用語の妥当性と文字数等)。

ある種の言語データを用いて具体的な例示を行いながら、各ステップで の作業内容と留意点を明らかにするように指導することが大切である。

(3) 図解(チャート化)の基礎技術 ここでは、下記のような図解にさいしての基本ルールや留意点を指導す る。

- ① 文字の使い方
- ② 文字の字詰めと行数
- ③ 文字の行間隔
- ④ 文章の要約化・短文化のしかた
- ⑤ 線の選び方
- ⑥ パターンの選び方
- ⑦ 記号の選び方
- ⑧ 矢印の選び方
- ⑨ 地図の使い方
- ⑩ アイソタイプ(絵文字)の使い方とピククトグラム(絵図解)
- ① 強調などの表現技法 ここでの指導においても、できるだけ具体例を示しながら説明していく ことが大切である。
- (4) 目的・用途に応じた表現の工夫

2.2の「(2)図解(チャート化)」で取り上げた図解の具体例を再度提示し、表現上の工夫と作成上の留意点を説明する。

(5) パターンにとらわれない図解の仕方

(4)迄は、図解の種類の中から形態(パターン)を選んで、図解をする方法を指導した。ここでは、パターンにこだわらず自らの創意工夫を取り入

れて図解を行うさいの手順を説明する。その手順は以下の通りである。

- ① 図解の目的や主題を明確にする。
- ② 図解を構成する要素を列挙する(キーワードを選んで、列挙してみる)。
- ③ 各要素をいくつかのグループに分ける。
- ④ グループの中を、階層化できる部分は階層化する。
- ⑤ 各要素もしくは要素グループ間の関係を整理する。
- ⑥ 各要素間の作用(移動、変化、対立、従属、支配など)を定義する。
- ⑦ 枠で囲ったり、線で結んだりして、相互の関係を整理する。
- ⑧ レイアウトを考える。
  - ・ 線や記号を使い分ける。
  - スペース(空間)を活かす。
  - ・ 立体化を考える。
  - 必要があれば、凡例をつける。
- ⑨ 言葉を厳選して、簡単な解説をつける(必要があれば)。 単に手順を説明するのではなく、具体的なテーマに基づいて、図解のしかたと留意点を明らかにしていくことが大切である。

## (6) 演習

目的別にそれぞれ別個の言語データを与えて、下記の目的にふさわしい 図解を行わせる。①~⑤に関しては、2.2の「(2)図解(チャート化)」で 示したパターンの中から選んで作成させる。⑥は、パターンにとらわれず 自分の創意工夫に基づいて作成させる。

- ① 相関関係を表現するための図解
- ② 構造を表現するための図解
- ③ 評価・判定をするための図解
- ④ 計画を表現するための図解
- ⑤ 整理・発見を表現するための図解
- ⑥ 構造を表現するための図解

作成した図解のいくつかをOHPで提示し、受講者を交えて評価し合う。 または、6人程度のグループに分かれて、それぞれが作成した図解を提示 し合って、相互に評価させる。

## 育成方法および指導上の留意点

- ① この章の講義は、例示中心で進めることが大切である。つまり、グラフや図解の具体的な例を示しながら、わかりやすい説明をする。
- ② 2.2 節の「ビジュアル表現の種類」では、沢山のグラフと図解のパターンを示すことになるので、受講者が混乱しがちである。したがって、それぞれの説明の終りで頭の中を整理してやる必要がある。パターンの名前や用途を回答させるような発問をするとか、練習問題をやらせるとかの工夫をして欲しい。
- ③ 所定の演習時間では完成しきれない受講者が続出する恐れがある。その場合には、翌朝ないしは次の時間迄に完成して提出するように指導する必要がある。演習を中途半端に終らせないようにすることである。
- ④ 作成したグラフおよび図解に関しては評価とフィードバックを必ず行って欲しい。可能なかぎり添削して、各受講者にフィードバックして欲しい。また、作品のいくつか(出来映えの良いものとそうでないものの双方)を選んで、OHPで提示し受講者を交えて評価し、要約してやる。あるいは、6人程度のグループに分けてお互いの作品を評価し合う方法もある。ただし、この場合は各グループに適切な助言が行える指導者を配置する必要がある。いずれの場合も、最後にはインストラクタが全体の講評を行い要約することを忘れないでほしい。
- ⑤ 日常のレポートや報告書作成、感想文や論文作成などにおいて、ビジュアル表現を工夫するように指導してほしい。教育期間中だけでビジュアル表現を習慣づけることはとてもできない。あらゆる機会を使って、ビジュアル表現を定着させるように気を配る必要がある。

## 用語

円グラフ,半円(相対半円)グラフ,二重円グラフ,構成比率帯グラフ,内 訳構成比率棒グラフ,パレート図,構成比層グラフ,正方形グラフ,棒グラフ, 積み上げ棒グラフ,重ね合わせ棒グラフ,複合棒グラフ,水平対称棒グラフ,ピラミッド・グラフ,偏差棒グラフ,面積グラフ,折れ線グラフ,層グラフ,階段グラフ,乙グラフ,動線グラフ,浮動棒グラフ,ファンチャート,ヒストグラム,レーダーチャート,損益分岐点グラフ,点グラフ,散布図,三角グラフ,円交差図,網目図,二軸法(表型)図解,二軸法(図型)図解,ダイヤモンド図,焦点図,拡散図,円多重図,三角図,フローチャート,Jチャート、SDチャート,順位チャート,ガント・チャート,パート図(アローダイヤグラム),デシジョン・ツリー,PDPC,親和図,系統図、連関図,特性要因図,ハッチング,アイソタイプ(絵文字),ピクトグラム(絵図表),地図グラフ,ドーナツ・グラフ

## 第3章 話し方の技術

### 学習目標

当部を受講された方々が、この章を修了した時点で、話し方の技術に関して、 次の事柄ができるようになることを目標とする。

- 1. 最近になって話し方の重要性が一段と増してきている主要な要因のうち5つを指摘する。また、情報処理技術者、とりわけシステム・エンジニアにとっての重要性の理由を5つ列挙する。
- 2. 話し方の基本ともいうべき十原則を指摘し、そのうち自分の弱点であるも のに気づいて適切に改善する。
- 3. メリハリがあり、意図した間を入れた効果的な話し方をする。また、一方 的に話すだけでなく、適切な発問をしたり、質問や意見にうまく対処し、相 手を上手に巻き込んだ話し方をする。
- 4. 効果的な話しの組立てをし、ノンバーバルなコミュニケーション(動作や 表情など)を生かした話し方をする。また、視線や例示などを生かした話し 方をする。

## 内 容

## 3.1 話し方の重要性(180分)

この節は、「話し方の技術」の導入部である。話し方の重要性を十分認識させ、受講者各自が、「聴きたい、知りたい、効果的な話し方ができるようになりたい」といった意欲や問題意識を持つように動機づけることが大切である。そのためには、冒頭で「話し方の技術」が今後ますます重要になっていることを印象づける必要がある。

「話し方の技術」の重要性が増している主要な要因としては、次のようなものが考えられる。

① 「人間らしさをとり戻したい」とか「他とかかわりたい」という欲求は、 確実に高まりつつある。逆に、世の中は他者とかかわりづらくなっている。 そこで、他の人とかかわるすべが求められるようになっている。

- ② 「言葉・表現・情報」が強い力を持つ高度情報社会では、自己表現をしていかないと、自分の存在すらつかめない。ところが、ヘタにものをいうと落ちこぼれかねない。このような社会では、自己の存在を証明し、「豊かな人間関係」を作っていく必要がある。そこで、うまくものをいうすべが求められている。
- ③ 高度情報社会では、人も企業も、自らを情報化していかないと生き残れない。そこで、ビジネスのあらゆる局面を乗り切って生き残るすべが求められている。自己表現をしないと生き残れない有言実行の<提案型社会>の様相を呈してきている。
- ④ 高度に成長した産業社会は、黙っていると、どんどん人間性が失われていく社会である。したがって、このような社会の中では人間本来の姿を復活させるすべが求められている。
- ⑤ 現代は、人間関係が非常に複雑な時代である。そこで、円滑な人間関係 をつくり、日常生活のあらゆる局面を有利に展開するすべが求められてい る。
- ⑥ 現代は、スピードの時代である。「書き言葉」では、変化の激しいビジネス社会に即応できなくなってきている。そこで、「話し言葉」が一層活用されるようになってきている。また、スピードの時代であるだけに、短かいつき合いのなかで効率よく自己を売り込み、他者から認められるすべとして重要性が増してきている。

以上の重要性をふまえたうえで、以下の4項目については、バズ・セッション方式の討議法でグループに分かれて検討させる。検討した結果は、グループ毎に発表させて板書し、インストラクタが要約する。

(1) 日常生活での重要性

「日常生活での話し方の重要性を5つ挙げよ」というテーマでグループ に分れて、7分間討議させる。

(2) ビジネスマンとしての重要性

「ビジネスマンとしての話し方の重要性を5つ挙げよ」というテーマの もとでグループに分れて7分間討議させる。

## (3) 情報処理技術者としての重要性

「情報処理技術者、とりわけシステムズ・エンジニアにとっての話し方の重要性の理由を5つ挙げよ」というテーマで、グループに分れて10分間 計議させる。

討議結果を発表させて、要約する。要約するにさいしては、次のような 重要性の理由を指摘する必要がある。

- ① 問題発見や解決の情報を収集するインタビューが欠かせない。
- ② 顧客やユーザに提案の説明をし、説得するために欠かせない。
- ③ 円滑で効果的なシステム作りやシステム利用を促進するために欠かせない。
- ④ SEの活動は、基本的にチーム活動である。チーム活動を効果的に進めて行くためには、メンバ相互間の遠慮のない話し合いが不可欠である。
- ⑤ SE活動の中でのコミュニケーションの時間的比率は非常に高い。中でも、各種のレビュー・ミーティングや打合せ会議等の比率が高い。しかも、そのような会議で説明を求められる機会が多い。
- ⑥ SEが取り組むシステム化業務の多くは、旧来のシステムややり方から新しいシステムややり方への変革である。そのためには、顧客やユーザをはじめとした関係者の意識革新や変革をする必要がある。相手の態度、信念、価値観などを変えるための説得力として欠かせない。
- ⑦ チーム・メンバや関係者をシステム化の目標に向かって動かすための 原動力である。
- (4) 情報処理技術者として「話」が必要となるとき

「情報処理技術者、とりわけSEとして<話>が必要になるときを5つ 指摘せよ」というテーマで、グループに分かれて7分間討議させる。

グループ発表の後、要約するさいには、次のような例を示すとよい。

- ① 顧客やユーザから情報や考え方を聴き出すとき
- ② 顧客やユーザに対して各種の提案を説明し、説得するとき
- ③ 各種の仕様書を関係者に説明し、理解・納得させるとき
- ④ プロジェクト・リーダに報告をするとき
- ⑤ レビュー・ミーティングでレビューを受けるとき

- ⑥ プログラマやオペレータからトラブル等の状況や問題点を聴き出すと き
- ⑦ チーム・メンバから悩みや問題点を聴き出すとき

### (5) 演習

3分間で<他人紹介>を行わせる。そのためには、受講者を2人のペアにする。まず、それぞれ10分間で相手の紹介に必要な情報を上手に聴き出すインタビュを行わせる。

お互いが収集した情報を整理させたうえで<他人紹介>のための3分間 スピーチを行わせる。

3分間スピーチが終るつど、コメントをフィードバックしてやる。なお、 受講者の人数が多い場合には、10人以下のグループに分けて、<他人紹介> を実施する。

## 3.2 話し方の基本(210分)

ここでは、話し方の基本である十個の原則についてふれる。一般論として説明するのではなく、前節の演習で行った<他人紹介>での良い点、悪い点を引用しながら、指導してほしい。

- (1) 正しい日本語を使う 変な造語や一般化していない用語などを使わないようにする。
- (2) 話し声が聞きとれる 聞き手が無理なく聞きとれる声で話す。聞き手の人数によって、声の大 きさの調整をするように気をつける。
- (3) 話している言葉が聞き手にわかる 明瞭に話す。1語1語をはっきりと区切って話す。
- (4) 話している内容が聞き手にわかる

1つ1つの言葉がわかっても、話している内容の意味がわからないとどうしようもない。わかりやすさこそが重要である。情報処理技術者の場合、以下のような事柄がわかりにくくしている主要な要因である。十分気をつけるよう指導してほしい。

① 専門用語、略語の続出

- ② 抽象的で難解な表現
- ③ 細細とした数字の多用
- ④ 外国語の多用
- ⑤ 飛躍や回りくどさ
- (5) 話し方が早過ぎない

聞き手が内容を理解するのにふさわしいペースにする必要がある。早口でなく、やや遅目がよい。早口である人は、早口を直すことに心がける前に、まず十分な間を入れることに気を配るよう指導してほしい。十分な間を入れることによって、聞き手が理解のペースを合わせることができる。それに、間を入れた後のでだしはゆっくり話すことができる。

(6) <立て板に水>の話し方はいただけない

<立て板に水>の話し方だと、聞いている時は何となくわかったような気持になるが、後に何も残らない。聞き手が情報処理(思考活動)をする余地がないからだ。意図した間を随所に入れる必要がある。詳しくは、次節の「効果的話し方」の中でふれる。

- (7) 話しに抑揚をつける
  - 一本調子の話し方は、聞き手を退屈にし、眠気をもよおす。話の内容に合せて、話す速度や声の大きさなどを変える必要がある。詳しくは、次節の「効果的な話し方」の中でふれる。
- (8) 活気をもって楽しそうに話す 活気のある動作と陽気な表情で話す。こわばった表情、無表情、冷淡な 表情、苦虫をかみつぶしたような表情などはさける。
- (9) 姿勢を正しくして話す

前かがみになり、原稿や机あるいは足元などに目が落ちるのはよくない。 また、直立不動の姿勢もいただけない。ポケットに手を入れたまま話すな ども見苦しい。

- (10) 自分のペースを失わず、感情的にならない 話す内容に自信をもち、冷静なペースを保持することが大切である。
- (11) 演習

「話し方の基本原則と自分の改善ポイント」というテーマで7分間のプ

レゼンテーションを実施する。

「話し方の基本原則」の講義内容を整理・要約すると同時に、<他人紹介>でコメントされたことを念頭に基本原則の中で改善すべき点をまとめさせる(このための準備時間を40分程度とる)。

全員にプレゼンテーションを行わせる。7分間を有効に使うようにさせる(早く終り過ぎても、時間をオーバし過ぎても問題である)。

プレゼンテーションが終るつど、5分程度のコメントをする。なお、受 講者が多い場合には、10人以下のグループに分けて実施するとよい。

### 3.3 効果的な話し方(360分)

- (1) 話の速さに変化をもたせる 以下のような諸点を強調する。
  - ① 速過ぎるのはよくない。聞き手が内容を理解するに十分なだけゆっく り話す。
  - ② 定速だと、単調さと眠気をもよおす。
  - ③ 重要な内容、強調すべき内容のときは、速度を遅くする。
  - ④ 聞き手の興味・関心を引きつけるためには、速度を速くする。
- (2) 声の大きさ以下のような諸点を強調する。
  - ① ボソボソと蚊の鳴くような声ではなく、張りのある大きい声で話す。
  - ② 重要な内容、強調すべき内容のときは、声を大きくする。
  - ③ 聞き手の興味・関心を引きつけるためには、声を小さくする。
- (3) 声の高低で話に活気を与える 次の諸点を指導する。
  - ① 声の高さを変えることによって、話に活気を与えることができる。
  - ② 同じ調子、読経口調にならない。
  - ③ 興味を引きつけるためには、声は低くする。
  - ④ 強調するときには、声は高くする。
  - ⑤ かん高い声、耳ざわりな声を出さない。
- (4) はっきり発音し、声の質を高める

次のような諸点に留意させる。

- ① はっきりと発音し、明瞭に話す。特に、単語の終りと文章の終りの言葉は、注意して明瞭に話す。
- ② 姿勢を正し、腹からしゃべる(横隔膜を使う)。
- ③ 大きく口を開いてしゃべる。
- ④ 感じのよい話し方に心がける。
- (5) 意識して<間>を入れる <ピリッと効いたわさびのような>間を意識して入れる。次のようなと きに、間を入れる。
  - ① 話の区切りには間を入れる。
  - ② 聞き手に考えてもらうためには、十分な間を入れる。
  - ③ 重要点・強調点を印象づけるためには、間を入れる。
  - ④ 聞き手を注目させるために、間を入れる。
- (6) 話から注意をそらす癖はなくする 話し手や話し手の話題から聞き手の注意をそらせてしまうような癖は、 できるだけ直すように指導する。
- (7) 効果的な発問を工夫する

- ① 全員対象発問
- ② 指名発問
- ③ リレー発問
- ④ 自問自答型発問
- ⑤ 投げ返し発問 なお、次のような点に留意して発問をするように指導することも大切である。
- ① 発問は、意地悪クイズではない。
- ② 発問内容は、<やさしからず、むずかしからず>がよい。

- ③ 回答が出ない場合は、ヒントを与えて答えやすくする。
- ④ かみくだいたわかりやすい発問をする。
- ⑤ 一度には一項目だけしか発問しない。
- ⑥ 中途半端な発問をしない。
- ⑦ 一人だけに集中して発問しない。
- ⑧ 発問で相手を不快にさせない。
- ⑨ 回答は大切に扱う。
- (8) 質問や意見にうまく対処する 質問や意見に対する応答の仕方には、次のような5種類がある。上手に 使い分けることができるように指導してほしい。
  - ① 即答型応答
  - ② ブーメラン型応答
  - ③ 代返型応答

ほしい。

- ④ 後回し型応答
- ⑤ 無回答型応答 応答にさいしては、次のような点に留意する必要があることを徹底して
- ① 発言者をできるだけほめる。
- ② 質問・意見を大切にする。
- ③ 質問や意見の内容は、全員に徹底する。
- ④ 回答は全員にフィードバックする(発言者だけに返さない)。
- ⑤ 1対1で終始しない。
- ⑥ 回答は手際よく行う。
- ⑦ 質問・意見は歓迎する。
- (9) 聞き手や環境に気を配る

聞き手に対する気配りはもちろんのこと、環境に関して以下のような諸 点に配慮するように指導してほしい。

- ① 座席の数と配置
- ② 部屋の広さ
- ③ 出入口

- ④ 喚気と温度
- ⑤ 部屋の明るさ
- ⑥ 設備や視聴覚メディア
- ⑦ 備品や消耗品
- ⑧ 騒音
- (10) して良いこと、悪いこと

「効果的な話し方」のまとめとして、以下に示すような<よい話し手がすること>、<してはならないこと>を徹底してほしい。

<よい話し手がすること>

- ① 公正に、自然にふるまう。
- ② 思いやりがある。
- ③ 過失を認める。
- ④ 聞き手に関心を払い、上手にリードする。
- ⑤ 聞き手をくつろがせる。
- ⑥ 熱意を示す。
- ⑦ 臨機応変、融通性がある。
- ⑧ 聞き手に敏感である。
- ⑨ ユーモア、機知がある。

<してはならないこと>

- ① 皮肉を言う。
- ② 排他的、挑戦的である。
- ③ 威張る、威圧する。
- ④ 言い訳をする、誤魔化す。
- ⑤ 感情的である。
- ⑥ 早口、まくし立てる。
- (7) 悪趣味なことを言う、当惑させる。
- ⑧ 見下げた話し方をする。
- ⑨ 手抜きをする。
- (11) 演習

「効果的な話し方と自分の改善ポイント」というテーマで10分間のプレ

ゼンテーションを行う。実施の要領は、前節の演習と同じであるが、準備時間を3時間程度とり、ビジュアル表現のOHPを3~5枚作成させる。つまり、OHPを使ったプレゼンテーションにする。

### 3.4 基本的配慮点(360分)

(1) 聞き手を分析する

プレゼンテーションのようなあらたまった話をする場合には、事前に聞き手を慎重に分析する必要がある。聞き手を分析することによって、聞き手の関心、ニーズ、悩みや問題点を把握する。また、相手の知識レベルを知る。そのためには、次のような事柄を分析する必要があることを徹底する。

- ① 地位や職位
- ② 所属している部門
- ③ 経験・職歴
- ④ 学歴・専攻分野
- ⑤ 年齢
- ⑥ 性別
- ⑦ 趣味・性格
- (2) 的を絞った狙いを定める

聞き手の分析結果、彼らの関心やニーズ、悩みなどが判断できたら、それらに応えるにふさわしい話の狙いを定める必要がある。しかも、欲ばった狙いではなく、相手の最大の関心やニーズなどに応えるように的を絞った具体的な狙いにする。

狙いの具体例をあげながら、その重要性を徹底してほしい。

## (3) 効果的な話の組立てを考える

具体的で的を絞った狙いが定まったら、その狙いにふさわしい話の組立 てを考える。つまり、話のストリー作りである。話の組立ての基本は、序 論(導入)、本論(展開)、結論(整理)の三段構えである。これをふま えて、起承転結のあるストリーを工夫する。

三段構えの組立てに関しては、具体例を示して説明する必要がある。ま

た、時間が許せば、具体的な狙いを与えて話の組立てを行う小演習を実施 するとよい。なお、話の組立てが完成したら、説明用の資料作りの前に以 下の事柄を行っておくことの必要性を強調してほしい。

- ① 時間配分を慎重に行う(重要な内容に十分時間をかける、質疑応答の ための時間を見込むなど)。
- ② 狙いからみて枝葉末節なことは割愛し、簡潔で、明解単純な内容にする。
- ③ 狙いからみて、強調すべき点や重要な内容を具体化する。
- ④ 聞き手の知識レベルを念頭に、定義すべき専門用語や略語を洗い出す
- (4) 上達の決め手 ― 評価とフィードバック

話し方は、場数をふめば上手になるといったものではない。1回1回の プレゼンテーションの結果を謙虚に評価・反省し、次の機会に改善のフィードバックを意識的に行う。これなくして、上達は望めない。

プレゼンテーションが不成功に終ったとしたら(つまり、狙いを達成できなかったとしたら)、どこかに改善すべき点があることを意味している。 話の組立てや内容に問題があったり、話し手自身に問題があったりする。 それらの真の原因をつかんで、改善する必要がある。

とはいうものの、話し手自身が自らの欠点に気づくのは容易なことではない。同席した同僚や仲間などから遠慮のない指摘をしてもらうように努める。最もよいのは、プレゼンテーションの様子をVTRに収録してもらい、それを再生してじっくり観察することである。なお、改善効果を高め、一段と上達するためには、以下の事柄が重要であることを強調してほしい。

- ① 他の人の指摘を謙虚に聴き、自己反省をする。言い訳けをする人は上達しない。
- ② 欠点を指摘してくれた人に誠意をもって応える積りで、改善努力する。
- ③ 大きな欠点から改善につとめる。大きな欠点を改善するだけで、見違えるように良くなる。
- ④ 熱意をもって、改善努力する。熱意をもてば、思いもよらない改善エネルギーがでる。
- ⑤ 創意工夫をする。基本ができたら、あとは創意工夫しだいで上達する。

## (5) ジェスチャを生かす

聞き手は、話し手の話を聞いているだけではない。話し手の身ぶりや表現などもくみとっている。熱意や自信をはじめいろいろなニュアンス(信頼感やさわやかさなど)は、ジェスチャや態度を通じてはじめて相手に伝わる。ジェスチャには次の三要素があることを説明し、それぞれの効果的ジェスチャを実演してみせることが大切である。

- ① 強調動作
- ② 視覚化動作
  - ③ ポイント動作
- (6) 話の展開にふさわしい動きする

動きもジェスチャと同様に重要である。動きに関しては、以下のような 留意点を強調する必要がある。

- ① 話の展開に合わせて動く。
- ② 強調のために動く。
- ③ 自信、冷静さ、リラックス等を表現するために動く。
- ④ 意味のない動き、せかせかとした動きはさける。
- ⑤ 話の内容や視聴覚メディアから注意をそらしてしまうような動きはさける。
- (7) 視線を生かす

視線は、次のような目的のために、効果的に生かす必要がある。

- ① 聞き手の反応を把握する。
- ② 親近感、一体感を築く。
- ③ 聞き手を注意・注目させる。
- ④ 自信、熱意を伝える。

視線を生かすためには、次のような配慮をする必要があることを徹底する。

- ① 全員を見る(特定の人だけを見ない)。
- ② 顔を中心に見る。
- ③ ランダムに見る。
- ④ 良い表情で、顔を相手に向けて見る。

- ⑤ 相手にプレッシャを与えない見方をする。
- (8) 表情を生かす

表情を生かすためには、次のような配慮をする。

- ① 何よりも笑顔が大切である。
- ② 自信に満ちた表情をする。
- ③ 冷静さを表わす表情をする。
- ④ 表情は、意識して練習すれば改善できる。
- (9) 事実や例示を生かす

<事実>の引用は、裏付け情報として役立つし、説得力がある。それに、聞き手の興味を引きつけることができる。一方、<例示>は抽象的内容を具体化し、分りやすくすることができる。また、具体例と結びつけることによって、記憶に留まりやすくする。それだけに、<事実>や<例示>を生かすことは、効果的話し方にとって不可欠である。ただし、下記のような諸点に留意して使わないと、逆効果になりかねない。

- ① 裏付けとしてふさわしい<事実>、こじつけでない<事実>を引用する。
- ② 聞き手にとって関心のある分りやすい <事実 > を使用する。
- ③ イメージがわきやすい、興味をおぼえるような具体例を生かす。
- ④ 聞き手にとって身近かな < 例示 > 、一人善がりでない < 例示 > を使用する。

## (10) 演習

自由テーマで10分間のプレゼンテーションを実施する。

各自にどのような対象者に対して、どのようなテーマでプレゼンテーションを行うかの条件設定を行わせる。そのあとで、プレゼンテーションの狙いを具体的にして、効果的な話の組立てをさせる。また、前節の演習と同様に、OHPを使用してプレゼンテーションを行わせるためにビジュアル表現のOHPを3~5枚程度製作させる。全体の準備時間は3時間程度を見込む。

プレゼンテーションが終るつど、7分程度のコメントをする。なお、受 講者が多いときは、10人以下のグループに分けて実施するようにしてほし

### 育成方法および指導上の留意点

- ① この章は、話し方教室で指導しているような話術を教えるのが目的ではない。話し方の基本をおさえたプレゼンテーション型の話し方を中心に指導することが主目的である。したがって、余り、細細とした技法や留意点まで立ち入らないようにすべきである。基本的技法や配慮点が実践できるようになることこそが重要である。
- ② 3.1節は、バズ・セッション方式を中心に進めるようにしている。バズ・セッション方式の典型的な進め方は、次の通りである。
  - 6人を標準にしたグループに分ける。
  - リーダ(進行係)を決める。必要なら、記録係も決める。
  - ・ リーダの司会のもとに、テーマについて自由に討議する。討議時間は、 1つのテーマについて6分から10分間にする。
  - グループとしての見解をまとめ、全員の了解を得る。
  - グループの結論をクラス全体に発表し、インストラクタは発表結果を 要約する。
- ③ 他人紹介および3回にわたるプレゼンテーションでは、1人1人の話が終るつどレビュー・コメントをフィードバックする。コメントは、受講者にも言わせる。ただし、全員に発言させる時間的余裕がないので、あらかじめコメント・シートを配布し、良い点と改善すべき点を記入することを義務づける。コメント・シートは、後で発表者それぞれに手渡す。なお、コメントにさいしては、まず良い点をほめてから改善すべき点を指摘するように指導する。
- ④ 他人紹介およびプレゼンテーションは、すべてVTRに録画する。録画したVTRは、レビュー・コメントを聞いたあとで、各自が再生視聴する。レビュー・コメントあるいはコメント・シートで指摘された点が果してその通りであるかどうかを自己評価し、次回の改善ポイントを定めさせる。 VTRの再生視聴は、完全に独りで行わせる必要がある。他の人が同席

すると、自分のありのままの様子を冷静に観察し、反省するさまたげになるからである。なお、VTRの再生視聴は、次の演習の時間のときに準備と併行して行なうとよい。

- ⑤ プレゼンテーションにおいては、前章で学んだビジュアル表現の技法を できるだけ活用するように仕向けることが大切である。〇HPを使って、 ビジュアル表現を生かしたプレゼンテーションを行わせる。
- ⑥ プレゼンテーションの準備時間は、人によって相当の差がでる。そこで、 プレゼンテーションは、準備の翌日に実施するようにする。そうすること によって、時間外や自宅に帰って準備を完了させることができる。
- ⑦ 日常業務における各種説明や報告、学校における課題発表などプレゼン テーションの機会は少なくない。そのような場合、できるだけ改善すべき 点を指摘し、本人に気づかせるようにする必要がある。気づかない以上、 改善の手がかりが得られないからである。

## 用語

バズ・セッション, インタビュー, プレゼンテーション, 他人紹介, 意図した間(ま), 抑揚, 発問, 全員対象発問, 指名発問, リレー発問, 自問自答型発問, 投げ返し発問, 即答型応答, ブーメラン型応答, 代返型応答, 後回し型応答, 無回答型応答, 序論(導入), 本論(展開), 結論(整備), 起承転結, 強調動作, 視覚化動作, ポイント動作, 視線, 表情, 例示

### 推薦教材

- ・第1章用として:「短い文章のコツ」、石川真澄、KKベストセラーズ、 1982年
- ・第2章用として:「プレゼンテーションの説得技法」、富士ゼロックス・ ドキュメントマネージメント推進室編、日本経済新聞社、1989年
- ・第3章用として:「説得力ある発表技術のすすめ」、江村潤朗、ビジネス・オーム社、1982年

## インストラクタ必読書

- (1) 「日本語の作文技術」,本多勝一.朝日新聞社,1976年
- (2) 「理科系の作文技術」, 木下是雄, 中央公論社, 1982年
- (3) 「文章構成法」、樺島中夫、講談社現代新書、1980年
- (4) 「ビジネス文章論」、扇谷正造、講談社現代新書、1980年
- (5) 「コミュニケーションのための図解の技術」, 久恒啓一, 日本実業出版 社, 1980年
- (6) 「技術者のためのやさしい図表の作り方」,小山勝彦,日本能率協会, 1987年
- (7) 「話し方入門」, D. カーネギー, 創元社, 1984年
- (8) 「上手な話し方」, 江木武彦, 東西社, 1986年

## 参考文献

- 文献1. 「朝日新聞の用語の手びき」,朝日新聞社,朝日新聞社,1990年
- 文献 2. 「毎日新聞用語集」,毎日新聞社,毎日新聞社,1990年
- 文献 3. 「文章・用字用語スタイルブック」, テクニカルコミュニケーション 研究会編, 日経 B P 社, 1988年
- 文献 4. 「科学論文をどう書くか」、未武国弘、講談社、1981年
- 文献 5. 「聞き上手・話し上手」,扇谷正造,講談社現代新書,1979年

- 文献 6. 「話し方と聞き方の構成」,堀川直義,至文堂,1977年
- 文献 7. 「情報処理技術者のためのプレゼンテーション技法」,三浦大亮,通 産資料調査会,1987年
- 文献 8. 「プレゼンテーションの進め方」、山口弘明、日経文庫、1986年
- 文献 9. 「プレゼンテーションの上手なやり方」,水島習平. 日本実業出版社, 1986年
- 文献10. 「技術者のための発表の仕方」, 河野徳吉, 日本能率協会, 1988年
- 文献11. 「プレゼンテーション新戦略」(「ビジネス戦略戦術講座」第四巻), 坂川山輝夫編者, 講談社, 1988年

# 禁 無 断 転 載

平成2年3月 発行

財団法人 日本情報処理開発協会中央情報教育研究所 〒105 東京都港区浜松町2丁目4番1号 TL 03(435)6511(代表)

