

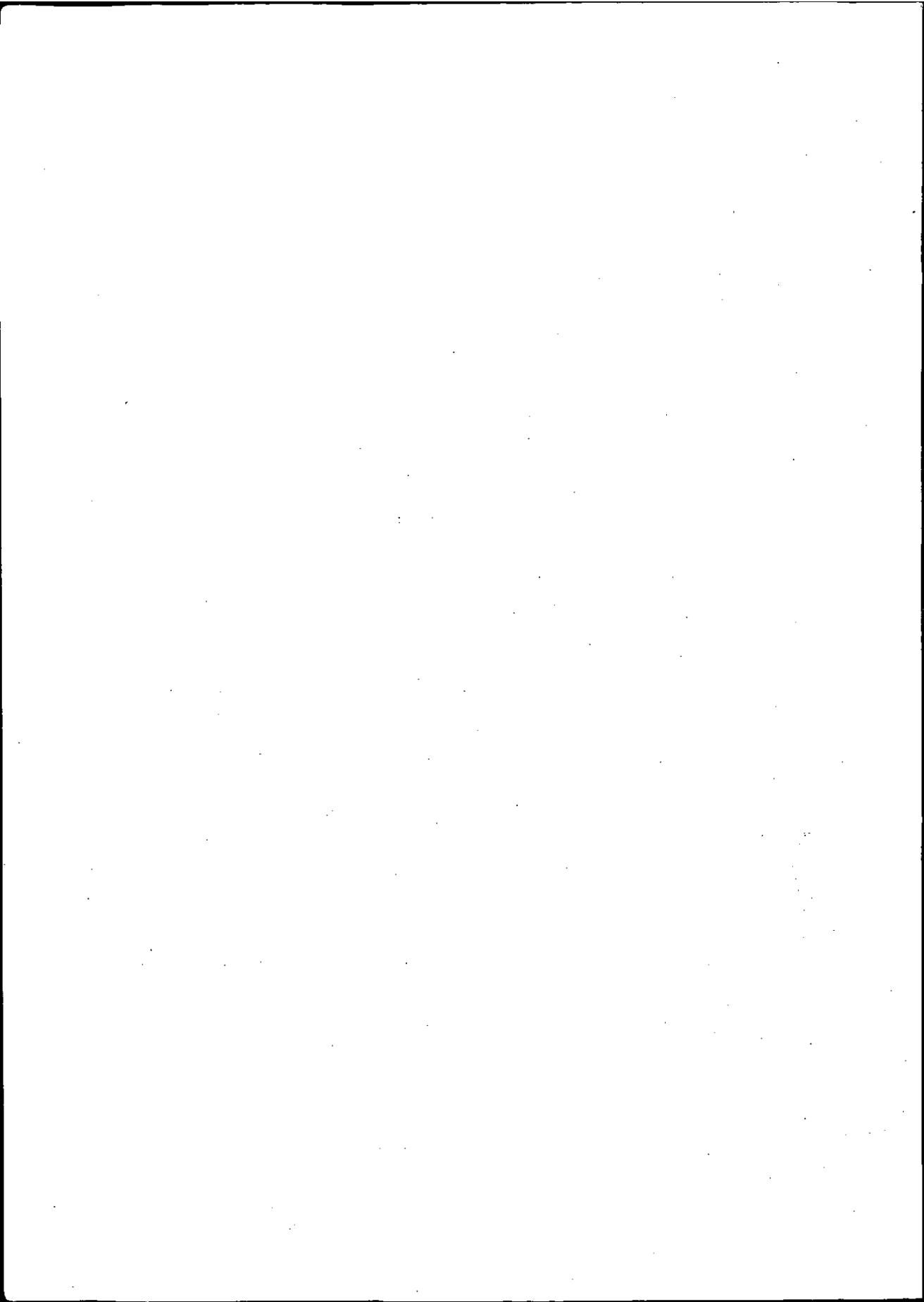
43-S-005

遠隔情報処理に関する調査 研究とシステムの開発

昭和44年6月

財団法人 日本情報処理開発センター





序 に 代 え て

電子計算機による情報処理は、社会・経済の発展にともない、各種情報の蓄積・加工・供給を最も有機的、効果的に進める狙い手として最近とくにその役割の重要性が認識されております。

また、情報処理そのものも、第三代電子計算機の登場以来、その利用分野の拡大とともに経営の意志決定システム、電子計算機の不特定多数による共同利用といった高度化の方向が検討されつつあり、従来の事後処理的な利用からみると、現在の情報処理は大きな発展期を迎えているともいえます。

このような情勢において情報処理および情報処理産業の前途には解決を要する移多の課題があります。

すなわち、電子計算機の有用性が認識されるとともに、電子計算機の利用は、今後の経営合理化のための必要条件となっておりますが、いうまでもなく電子計算機の設置には莫大な資金を要しますので、一般の企業にとっては、電子計算機を共同利用できる、いわゆる遠隔情報処理システムの確立が、その有効かつ経済的な手段として強く望まれております。

遠隔地から利用者が直接電子計算機を操作する、この遠隔情報処理システムの実現には、通信における各種問題の解決とともに、電子計算機のオペレーティング・システム、通信制御装置、端末機器といった面の技術的問題点の解明が必要であります。

当財団は情報処理に関する諸問題解決のため、各種の調査研究事業を実施しておりますが、この研究報告書はその一環として遠隔情報処理に関する調査・研究を行なうとともにソフトウェア技術を研究して、そのあり方と今後の問題点について検討した結果をとりまとめたものであります。

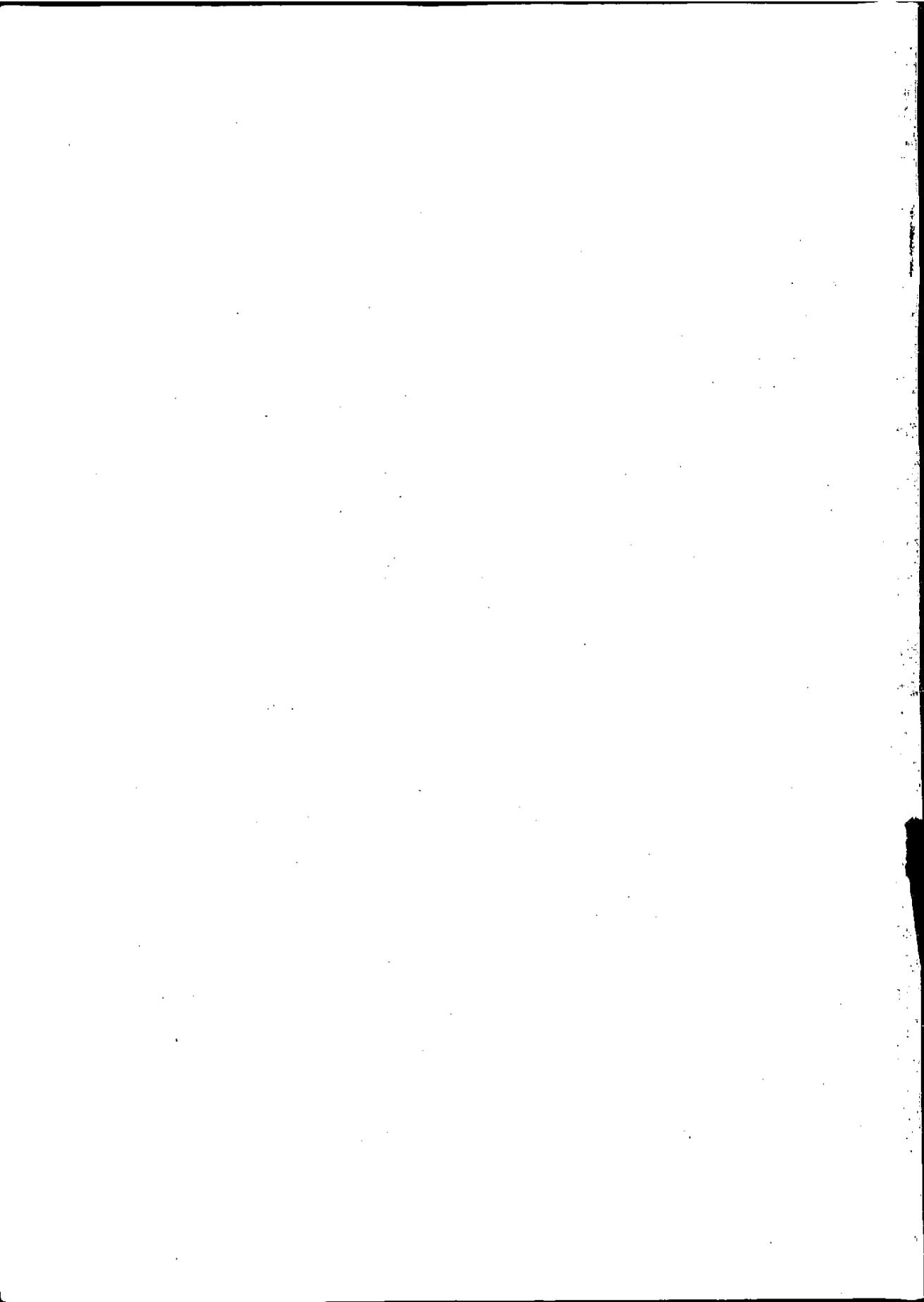
なお、この事業は日本自転車振興会の機械工業振興資金による「昭和43年度情報処理に関する調査研究補助事業」の一部として実施したものであります。

ここに、この研究実施にご尽力ならびにご支援を賜わった林一郎、淵一博、大東栄一、大庭弘太郎の各氏ならびに関係各位に心より感謝の意を表しますとともに、この報告書が各方面に利用され、わが国情報処理産業発展の一助として寄与できますよう念願いたす次第であります。

昭和44年6月

財団法人 日本情報処理開発センター

会 長 難 波 捷 吾



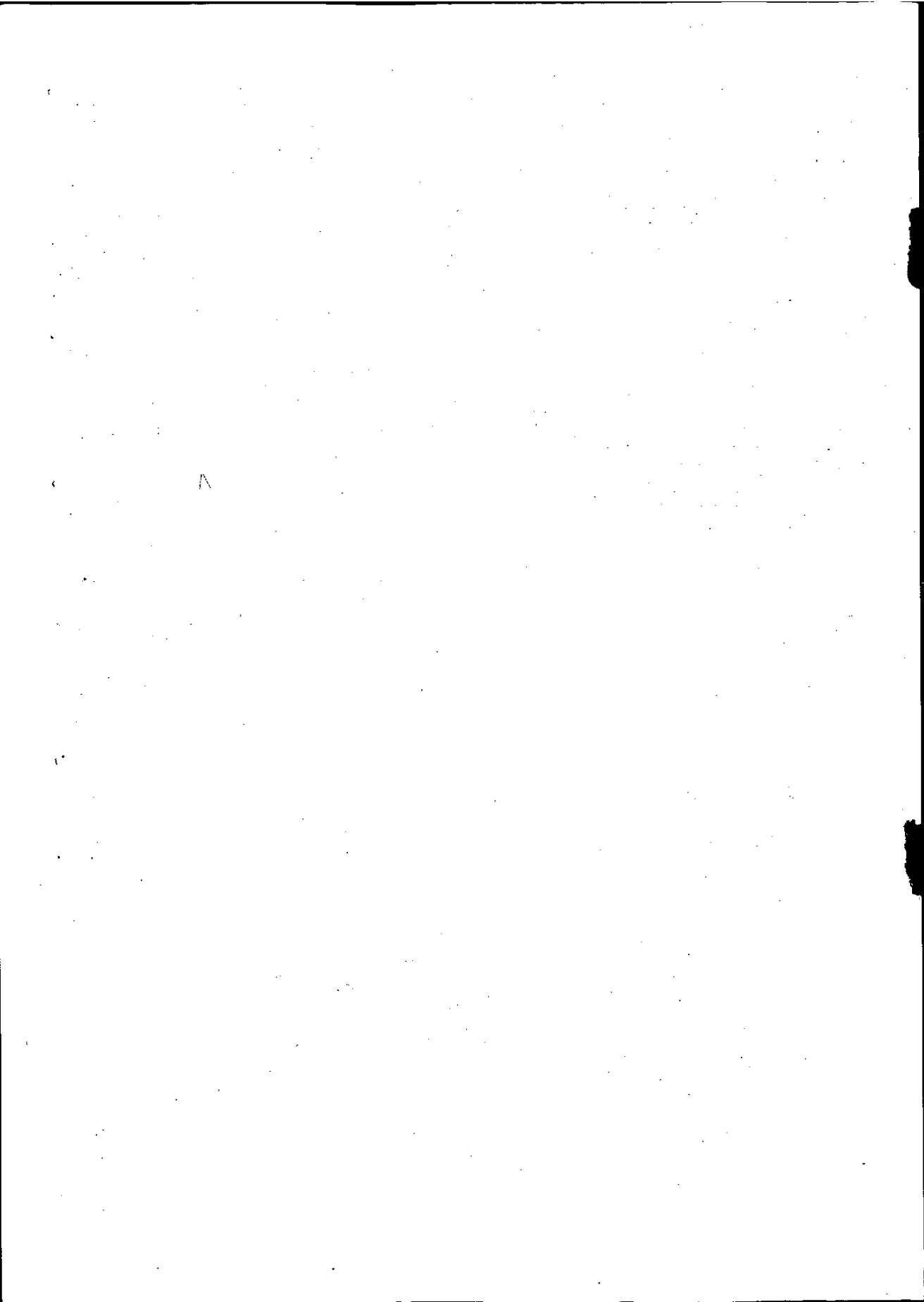
目 次

遠隔情報処理に関する調査研究 とシステムの開発

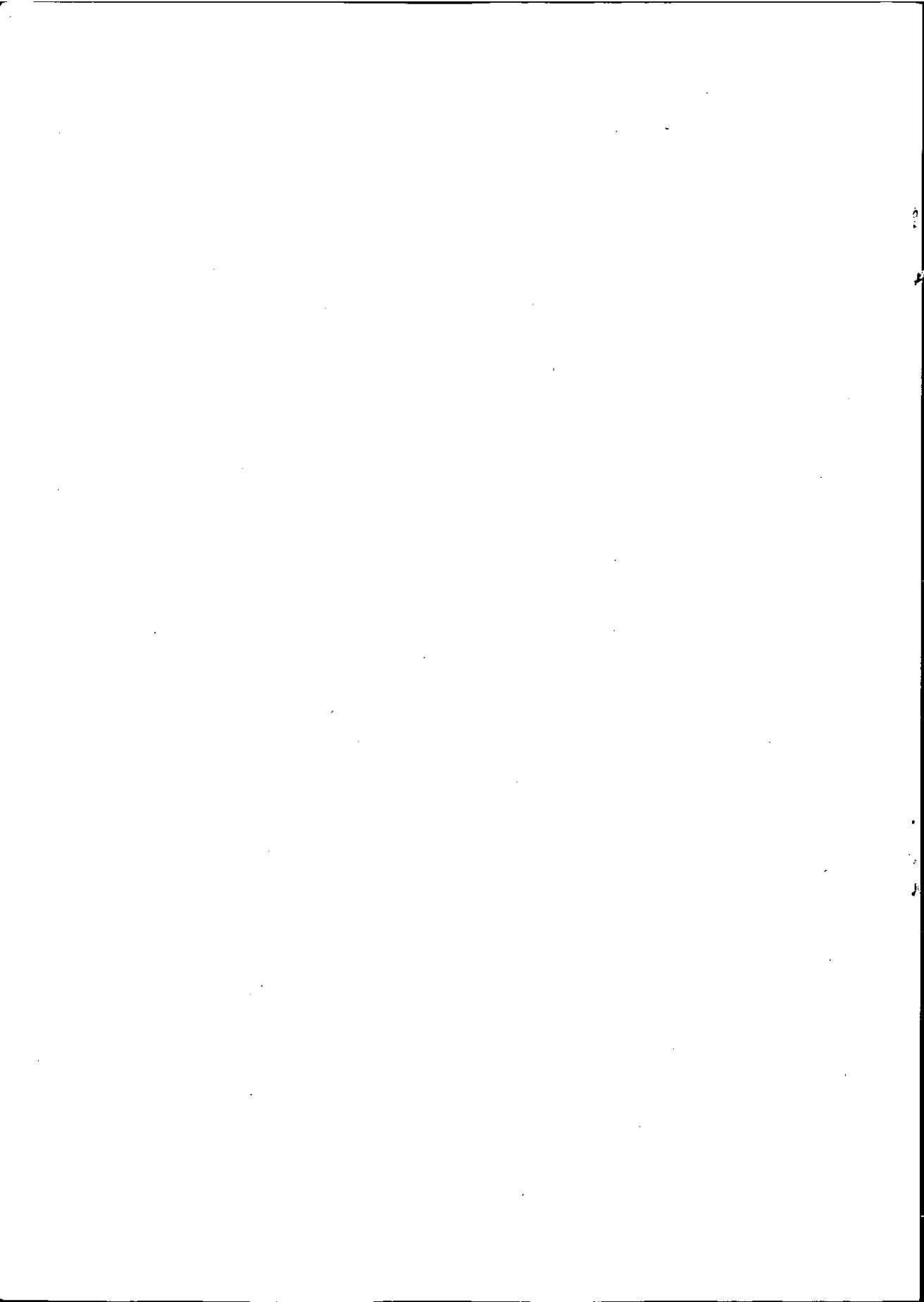
1. 概 説	1
1.1 遠隔情報処理の動向	2
1.2 オンライン文献検索システム JOLDOR	4
1.3 タイムシェアリング・システム COBOL	5
2. 遠隔情報処理技術の構造と展望	7
2.1 まえがき	7
2.2 システムと技術	7
2.3 遠隔情報処理技術の全般的動向	8
2.3.1 システム多様化の要因	9
2.3.2 技術の動向に対するシステム多様化の影響	10
2.4 遠隔情報処理システムと中央処理技術	12
2.4.1 中央処理技術の性格と方向	12
2.4.2 時分割技術と多産処理技術	14
2.4.3 ファイル技術と遠隔情報処理	16
2.4.4 言語処理技術と遠隔情報処理技術	22
2.4.5 システム開発技術の現状と方向	26
2.5 端末と制御	30
2.5.1 通信誤りの対策	31
2.5.2 会話形システム	35
2.5.3 割込み機能	39
2.5.4 計算機による端末の管理	42
2.5.5 オンライン端末のあり方	44
2.5.6 むすび	52
2.6 データ通信網	53
2.6.1 データ通信網の現状	54

2.6.2	これからのデータ通信網	61
3.	遠隔情報処理業務例	75
3.1	計算サービス	75
3.1.1	遠隔情報処理としての計算サービス	75
3.1.2	発展の背景および現状	75
3.1.3	計算サービスの諸方式	77
3.1.4	計算サービスの問題点	82
3.1.5	むすび	86
3.2	自動遠隔検針	88
3.2.1	自動遠隔検針の必要性	88
3.2.2	自動遠隔検針の諸方式	89
3.2.3	伝送系(伝送路と伝送方式)	106
3.2.4	メータ読取帰構	108
3.2.5	NTT方式の概要	109
3.2.6	むすび	111
4.	オンライン文献検索システム; JOLDOR	115
4.1	まえがき	115
4.2	オンライン・システムの特長とJOLDORの設計方針	115
4.3	システムの概要	118
4.3.1	帰器構成	118
4.3.2	基礎データの作成	119
4.3.3	コマンドの種類と帰能	126
4.3.4	ファイルの構成	129
4.3.5	検察の流れ	139
4.3.6	ON LINE UPDATE	145
4.3.7	バッチ用サービスルーチン	148
4.4	JOLDORシステムの検討と反省	155
4.4.1	データの収集と作成について	155
4.4.2	ファイルについて	155
4.4.3	オンライン・プロセッサについて	157
4.4.4	あとがき	158

5. タイムシェアリング・システム COBOL	161
5.1 概 要	162
5.2 システム構成	162
5.2.1 コンパイル時におけるシステム	163
5.2.2 オブジェクトプログラム実行時におけるシステム	164
5.3 スーパーバイザコマンドおよびファイルシステム	167
5.3.1 スーパーバイザとコマンドプログラム	167
5.3.2 ファイルシステム	173
5.4 COBOLコンパイラの構成	175
5.4.1 フェーズフロー	175
5.4.2 モジュールリティ	176
5.5 コンパイラの詳細	178
5.5.1 インタフェースモジュール	184
5.5.2 メインモジュール	190
5.6 会話形式によるコンパイラ	219



1. 概 説



1 遠隔情報処理に関する調査 とシステムの開発

1. 概 説

遠隔情報処理システムとは、情報処理設備（電子計算機）と電気通信設備とを結合して行う情報処理のシステムをいう。従ってこれは電電公社のいうデータ通信と同意語のものとする。

情報処理技術の発展に伴い、遠隔情報処理の需要が急速に増大する傾向が強まっている。このため電子計算機と通信網という二つの分野の結合は、両分野を一体化した新技術の開拓の必要性を生み出した。しかもこの新技術は現在迄の技術だけから生れるものではなく、今後の科学技術の研究開発にまつべきものが多いのである。また従来常識化されていたような概念が新技術の誕生により、これが修正されねばならないことが起るであろう。

たとえばタイムシェアリングの概念などがその一つとしてあげられる。Groschの法則は電子計算機のパフォーマンスは価格の二乗に比例することを証明し、小さな電子計算機を占有するよりも、大きな電子計算機の一部を使う方がコストが安いといっており、タイムシェアリングの有利性の一つの裏付けをしたとみられる。しかるに小形計算機が安価に売られ、ユーザーがこれを占有することが可能となれば、タイムシェアリングの分野の一部が小形計算機に吸収されることとなり、Groschの法則はこのような領域では通用しにくくなる。

将来電気通信分野では電子交換機時代を迎え、従来みられなかったような新しい通信サービスの提供が実現されようとしている。その時代には、電子交換機は単に通話のための回線交換の機能をもつだけでなく、データの交換はもちろん、遠隔情報処理の一部もこれによって行うことが合理的であり、経済的であるかも知れない。元来オンライン・リアルタイム・システムなどの技術は電気通信技術として開発されてきたものであり、電子計算機技術として欠くことのできない待ち合わせ、割り込み、優先処理などの機能は電気通信分野において早くから利用されてきたものである。

また、遠隔情報処理利用の拡大にとともに、高速のデータ伝送網の形成を必要とするものと考えられるが、データ伝送技術はこの必要性に対応して急速な進歩を遂げるものと思われる。この問題は技術的な面よりも、経済的な面に重点がおかれて検討が進められるであろう。いずれにしても将来遠隔情報処理システムについては、情報処理システムの技術と電気通信技術とが融合す

ることにより最適なシステムの構成が考えられて、コンピュータ・ユーティリティ時代に重要な役割を演ずることとなる。

日本情報処理開発センターにより構成された遠隔情報処理調査委員会では、遠隔情報処理システムに関する広汎な分野について調査することは、もとより力のおよぶところではないので、とりあえず、遠隔情報処理技術の動向をみるため、中央処理技術、端末と制御、データ通信網の各事項を検討し、また、遠隔情報処理業務をみるため、計算サービス、自動遠隔検針の例をあげることにした。この業務例としてはこの他に、電力負荷制御、プロセス制御等があるが、これらに関しては除外した。

なお、周知の通り、遠隔情報処理の問題は、諸方面の関心をよび、新聞雑誌業で多くの論説がみられる。本委員会は、これらの文献には可能な限り目を通し、本報告書が単に個人的見解を述べたものとならぬよう配慮した。また、各項目について問題の所在を明かにするよう努めた。

本報告書には遠隔情報処理システムの例としてオンライン文献検索システムJOLDRおよびタイムシェアリング・システムCOBOLを収録した。これらは日本情報処理開発センターで計画中の遠隔情報処理システムの一環として開発したもので、それぞれその道の専門家達によって作られたわが国独特のシステムである。

以下これら各項で検討した結果の概要を述べることにする。

1.1 遠隔情報処理の動向

先づ、遠隔情報処理技術の動向を眺めるため、中央処理技術を検討したが結論として、次の各事項をあげている。

すなわち、第一に、遠隔情報処理システム技術に影響を及ぼす要因を分析し、とくにソフトウェア技術のあり方の位置づけを明確にした。次いで、タイムシェアリングなどに代表されるシステム制御技術の問題点と動向を考察した。

第三に、遠隔情報処理システムでとくに重要になるデータベースすなわちファイルシステムを表現する技術の分析を行なった。この分野は、技術的に、今後もっとも多くの研究努力、開発努力を集中しなければならない分野である。

第四に、情報処理システム開発技術に関連する問題点を分析し、言語処理技術をその一環として、その技術的諸問題の検討を行なった。

以上の4点から、中央処理技術を検討したのである。

次にオンライン・システムに必要な端末と制御の問題につき、特に会話形システムを中心として詳細な検討を試みた。最初に通信回線で起るデータ伝送の誤りを制御する各方式の得失を論じ、

これと信頼性との関連を述べている。また、タイムシェアリングなどの会話形システムについて詳述し、特に伝送上の割込みの実行方法、電子計算機による端末装置の管理の方法とそのあり方についての見解を示した。今後の問題点として、符号、伝送方式、入出力機器の標準化、入出力機器のあり方、機器および伝送回線の高速化等の諸項を挙げている。

続いて、データ通信網の将来のあり方について論述した。データ伝送設備、交換設備、および情報処理設備により構成されるデータ通信網は、その適用業務の性格によりその特徴を異にするものであるが、このような適用業務毎のデータ通信網は、将来相互に連結し、複合化し、共同利用の方向に進むものと考えこの共同利用のデータ通信網の問題点、解決の方策などについて2～3の提案を示している。かくしてデータ通信単一網の特徴を活かしながら、音声通信、データ通信、画像通信などに加え、多彩な情報処理と伝達を経済的にかつ融通性をもって行うことのできる情報通信網の実現に期待をかけている。

遠隔情報処理業務の例としてまず計算サービスをとりあげた。これはタイムシェアリングによるサービスを主目的としたシステムに重点をおいて、計算サービスの現状と実施上の問題点につき述べたものである。

各種計算サービスの一例として、電電公社が行う予定の加入データ通信サービスをあげ、そのいくつかの特徴を示した。

計算サービスはこれから発展しようとしているサービスであり、また技術的にも多くの問題がある。たとえば、タイムシェアリング・システムとバッチ・システムとの経済性の比較の問題、計算サービスは事務計算分野が技術計算分野よりかなり遅れているという問題、システムの信頼性をどのようにして高めるかという問題、機密保持の問題、各種標準化の問題などがある。本文はこれらの問題についての見解を示している。

遠隔情報処理業務の第二の例として自動遠隔検針というものをとりあげた。これは電気、ガス、水道などの家庭用メータの指針値を自動的に読み、これを電気信号として中央の計算センターまで伝送し、これを処理して結果を記録することをいうのである。本項は自動遠隔検針の諸方式を説明し、その伝送システムについて述べるとともにメータ読取機構を例示した。そして電電公社方式の概要を紹介している。自動遠隔検針は現在まだ実用に供された例はなく、技術上、経済上なお幾多の問題をかかえているが、自動遠隔検針の必要性は年々増大するものとみられるので、いずれ実用化されることであろう。それには電話線を利用する電電公社の方式が最も存望されている。

1.2 オンライン文献検索システム JOLDOR

電子計算機の最も有効な適用例の一つとして、情報検索がある。JOLDORと名付けたこのシステムは、オンラインにより、文献検索を行うものであり、一般のバッチ処理による方式に比較して、

- (a) 文献の検索結果が希望した時点ですぐに得られる。
- (b) その結果をもとにして、電子計算機システムとの対話で試行錯誤のうえ、さらに新しい情報を選択附加できる。

といった利点がある。

JOLDORは、当財団設置のFACOM230-60(MONITOR-V:バッチ処理、デマンド処理、リアルタイム処理のできるOperating system)を対象に、キャラクタ・ディスプレイとタイプライタをペアにして端末装置とするもので、現在システム設計とバッチ処理部分(ファイルを作成するルーチン、KWOCサービス、SDIサービス)までが完成し、部内サービスを開始している。オンラインの検索部分については、FACOM230-60のTSSサービスを待つ状態であり、コーディング作業を続けている。

主な特徴——

- (1) 入力データは各文献につき次の項目を含む。
 - 標題
 - 著者名(所属を含む)
 - 文献目録(誌名、巻号、ページ、年代)
 - ディスクリプタ(必要なら20個まで)
 - 文献番号
- (2) 既存のバッチ用KWOCシステムを使って、キイ(key)の分布や長さ、リスト構造にした時のノード(node)数等の分析を行ない、その結果を充分反映させてファイルを設定した。
- (3) ファイル設計は検索時間の短縮を最大目的とし、ターミナル・ユーザとの応答は、5秒以内に行われること。
- (4) ファイルの更新は、オンラインで、端末から随時行い、月に一回程バッチの更新で乱れたファイルのディレクトリを整理する。すなわち、オンライン、バッチの併用で更新を行う。
- (5) 検索タームにはAND, OR, NOTの論理演算をゆるす。
- (6) ユーザの質問に応じて、コマンドの使い方を表示するガイダンス(guidance)の機能を備える。

(7) 収録文献はさしあたり、コンピュータ関係の論文約2万件を対象とする。

(8) バッチ処理で、KWOCサービスと、SDIサービスを行う。

1.3 タイムシェアリング・システム COBOL

NEAC2200 TSS COBOLはオペレーティング・システムMODE/V COBOLコンパイラを、TSS用に改造して作成されたものであり、TSS・COBOLとしては比較的水準の高い言語を使用することができる。更に次のような特徴をもっている。

(1) ソース・プログラム入力時に会話モードで実行できる。

これは入力される行毎にシンタックス・チェックを行い、エラーが検出されると、端末へメッセージを出力し、オペレータの修正を待つ。

(2) 一度入力されたソース・プログラムは、完全に保存される。

コンパイルされるとソース・プログラムはソース・プログラム・ファイルとして保存され、更新はUPDATEコマンドにより行なわれる。

(3) TSSモードでテストランされる。

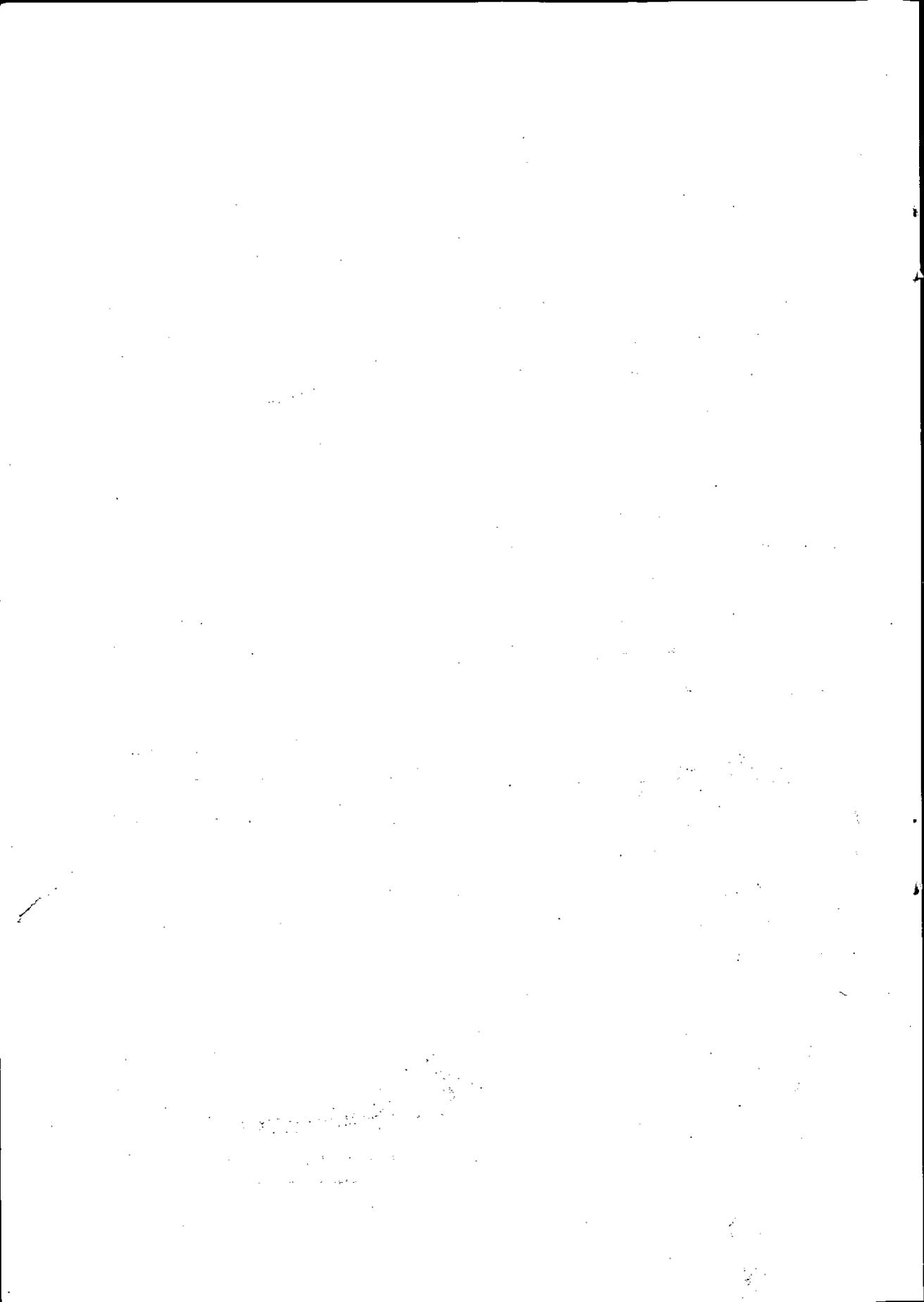
コンパイルされたCOBOLオブジェクト・プログラムでは、全てのファイルがTSS側のディスク・ファイルまたは端末装置を仮想して作成されている。従ってTSS側でテストラン、デバッグが行なえる。

(4) デバッグ済のプログラムは修正せずに、バッチ・モード側へ移行される。

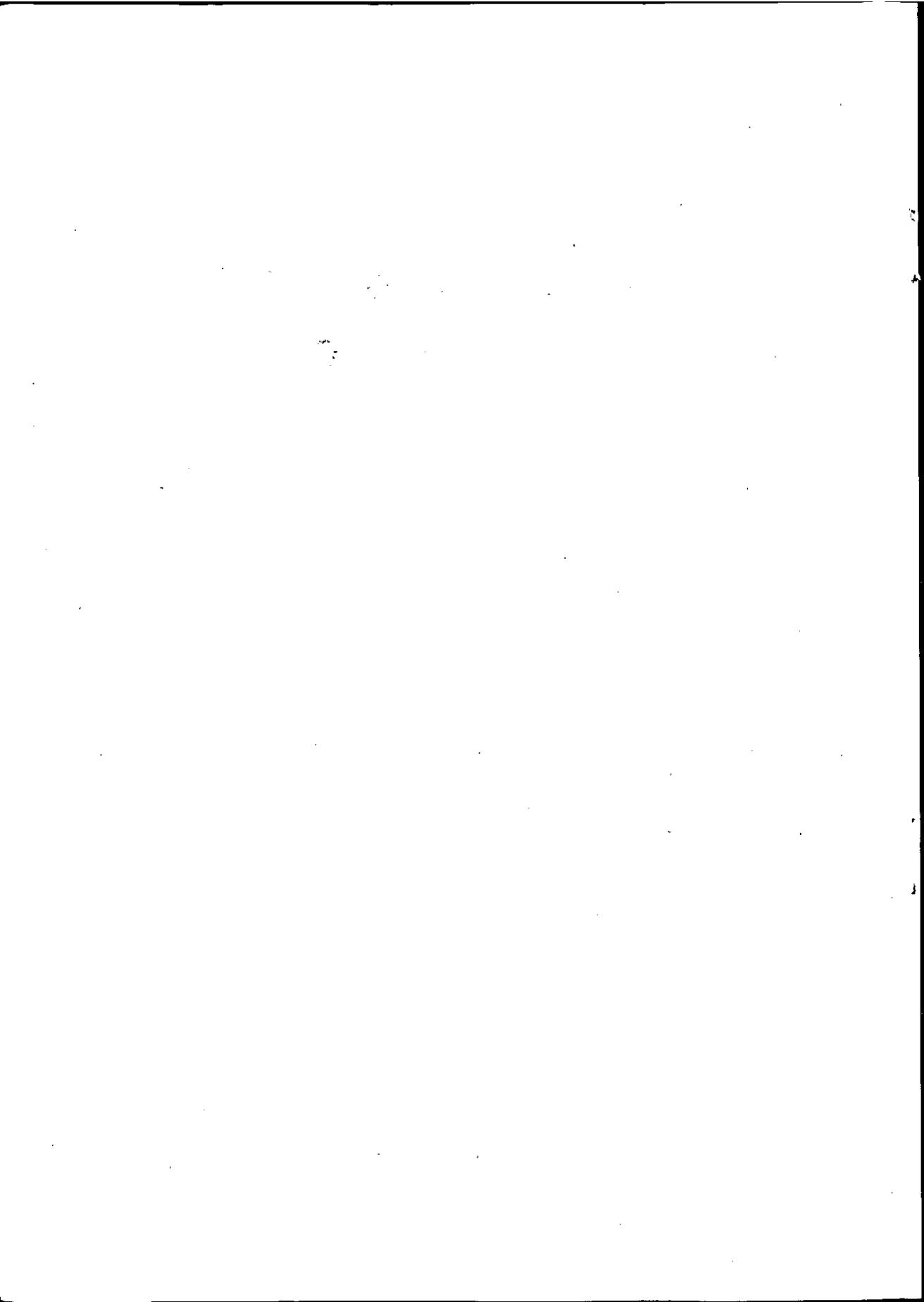
デバッグの済んだプログラムは、修正せずにそのままバッチ・モード側へ再コンパイルすることにより使用される。

すなわち、TSS・COBOL・コンパイラとバッチ・モード側のコンパイラは、ソース・プログラム上完全な互換性がある。

(5) TSSの特徴であるTurn around timeの短縮、マン・マシン・コミュニケーション、システムの高効率な活用等凡てを、TSS・COBOL・コンパイラは充しているといえる。



2. 遠隔情報処理技術の構造と展望



2. 遠隔情報処理技術の構造と展望

2.1 まえがき

この章では、遠隔情報処理技術のとらえ方を明らかにし、その技術の見取図を本編の構成と関連させて述べる。

遠隔情報処理「技術」と遠隔情報処理「システム」との関連を明らかにするためには、広義の遠隔情報処理「システム」の中に、狭義の、すなわち、「機械設備」としての遠隔情報処理「システム」を明確に見分けておかなければならない。

そのことによつて、「利用」の観点からの諸要請を、機械設備の「製作」の中で実現する人間活動としての「技術」のありようが浮び上ってくるのである。

遠隔情報処理システムの利用形態、ならびに諸実例については、本報告書の他の編があげられている。それらをシステムの「外部的」記述とするならば、本章の立場は「内部的」記述ということになる。

2.2 システムと技術

システムを広くとらえれば、現実にオペレーショナルなシステム、たとえば企業体は、内部に、人間集団からなるシステム（組織）と、機械設備システムとしての情報処理システムを、それぞれサブシステムとして含む、トータル・システムであるわけである。このシステム全体を、（機械設備的な）情報処理システムの立場から見れば、それは自分自身を含む、外部的・環境的システムとして見えることになる。

この外部的システムの分析、設計の作業がいわゆるシステム技術者（SE）の第一義的な任務であると、ここではとらえておきたい。とすると、SE作業の結果、情報処理用機械設備への要請が抽出されてくるわけである。

遠隔情報処理システムは、そのようなトータル・システムにおける機械設備の一形態なのであるが、またそれが支配的な形態になるであろうことが予測されている。

外部システムからの要請が需要側であるとすれば、情報処理システムを実現する技術は供給である。供給の水準によつて需要が規定されることは、（外部）システム設計にあつて、計算機技術の水準を適確に把握していることがSEの資格であり、現実にはその知識の上で立つてシステム設計が行なわれている。

トータル・システムの設計にあつて、機械設備としての情報処理システムのありようが決

定的な重要性を持つことから、それを成り立たせる技術の現状と将来予測が、この遠隔情報処理の分野の将来展望においても、中心的テーマにならざるをえないのである。

さて、情報処理機械システム自身も、トータル・システムの中でのサブ・システムとしての一つのシステムであるからには、それ自身にシステム工学的アプローチが必要であり、とくに遠隔情報処理ネットワークの構成にあつてはそのことがあてはまる。これについては2.3で述べる。

遠隔情報処理システムを別の観点でとらえれば、それは一つの龐大な人間、機械系を構成するものと見ることができる。この立場からは人間工学的なアプローチが浮び上がる。

人間と機械の交渉は「会話」としてとらえられる。この会話の実現のための技術が、遠隔情報処理技術の大きな側面になる。会話は人間とシステム間のインターフェイス装置である「端末」装置を介して行なわれるのであるが、そのようなものとしての端末のあり方および会話のための技術的条件について、2.5があげられている。

中央処理技術のとらえ方の一つは、それをシステムの機械設備のうち集中化された諸設備、すなわち中央処理装置、ファイル機器、入出力機器等からなる中央システムにかかわるものとして見ることである。このように形態的に見た場合、少なくとも現存の多くの遠隔情報処理システムについて、その中央設備の形態は、非遠隔システムのそれと類似である。逆にいえば、非遠隔のクローズドショップ的システムに、遠隔用諸設備を結びつけた形において、遠隔システムが構成されている場合が多い、ということでもある。

そこで、中央諸設備に関しては、それについての技術が非遠隔システムについてのそれと、ほぼ同様であるということが、形態的にもいえるわけである。

しかしながら、遠隔情報処理が非遠隔情報処理と並列的に存在するというより、むしろ従来の非遠隔処理を内に吸収し、それを含むものとして発展しつつある傾向をとらえるならば、この観点からの整理が必要であろう。そこで、この中央処理技術について2.4があげられている。

2.3 遠隔情報処理技術の全般的動向

遠隔情報処理技術の動向を、個別の分野の寄せ集めではなく全体的(global)にとらえようとすれば、まず必要なのは、それに対応するシステムのグローバルな動向である。

その観点からシステムの動向を見た場合、システムの多様化の傾向をとりあげるべきだと思

われる。そこでこの節では、その多様化をもたらす要因と、それが技術のグローバルな動向に及ぼす影響について分析する。

2.3.1 システム多様化の要因

遠隔情報処理技術の動向を考える場合、情報処理システムの多様化の傾向を認識する必要がある。この傾向は徐々に現われ始めているが、現状ではその事実はまだ十分に意識されていないとはいえない。ここでは、その多様化をもたらす要因を分析することにしよう。

第一は、遠隔情報処理システムを要求する需要な多彩な点である。需要がますます広がるにつれて、システムの受け持つべき業務の種類も千差万別になる。それぞれの業務は、それに適したシステムを必要とする。

遠隔情報処理を要求する業務は、単に「計算処理」的側面において個別的な特異性を求めるだけではない。それは、例えば端末装置の種類、処理装置の配置、ファイル装置の能力と形態、またそれらを結合したオペレーションの形態についてもそれぞれ独自のものを要求するのである。

第二は、上に述べたことと関連することであるが、端末装置の多様化の現実である。端末装置は、人間と機械のインターフェイス装置として、遠隔情報処理システムの中で重要な役割をもっているが、この端末装置の分野において、多彩な諸装置の製作・販売が、とくに米国において盛んになりつつあることが指摘されている。

その現状については別途詳細な調査を行なう必要が痛感されるが、この分析においてさし当り必要なことは、そのような事実が存在するということの指摘である。

端末装置について、それが「供給」側から多彩化しつつあるという事実は、それが新たな多様な需要を喚起することによって、システムの多様化を促進する。また、単に代替品としての新しい端末装置の出現でさえ、それを取付けられるシステムの側からすれば多様化の要因となるのである。

第三の点は、安価で高性能な小型計算機、俗称ミニコンの出現と普及である。これが遠隔情報処理に及ぼす影響の第一は、それが端末設備に取り入れられて、端末の多様化をもたらすであろう、ということである。

ミニコンの第二の、そしてより根源的な影響は、それが汎用制御装置として、単に端末設備のみならず、システムの各部に配置されるようになるであろうということである。

それは、現在一時的に標準的なシステム構成の図式になつている、中央設備と端末設備群という二部分から成る恒星・惑星群的なシステムの形態に変化を要求することになるだろう。

そのようなものとしてのミニコンの働きは各種各様であつて、ミニコン群を含んだシステムにとつて、単一標準的なシステム形態への集束を考えることはむしろ誤りであつて、ミニコンはシステム多様化の強力な要因と考えるべきである。

遠隔情報処理システムにとつて、システム多様化の認識が、システム内的にも重要な点は、そもそも遠隔情報処理システムが、多くの、かつ多種類のサブ・システム群からなる統合的な複雑体であるということである。一つの遠隔情報処理システムをとつてみても、すでにその内部に多様な(サブ)システムを含むことが要請されているのである。

システム多様化について最後に指摘すべき点は、設備更新の問題である。遠隔情報処理システムを有する活動体は進化・発展する。その要求する機能は、現存するシステムの機能をいずれ上回るであろう。

システム・オブソレッセンス (Obsolescence) は、技術の進歩によつても生じる。ハードウェア的には、まず第一に新しい端末装置の出現によつて、また高性能な内部諸装置の出現によつて、システムの陳腐化が意識される。

ソフトウェアについても、その技術の未熟な現状からすれば今後の技術進歩は著しいであろう。システム稼動開始時におけるソフトウェアの品質は多くの場合、極端に言えば試作品的であり、その後のバージョン・アップによる性能向上が著しいのが現状である。

一つの情報処理システムの寿命は、部分的改良を含めるとしても、まず5年と推測される。この設備更新のサイクルの存在は、ある時間スケールをとつてみれば、その期間内に出現するシステムの多様化を意味している。

2.3.2 技術の動向に対するシステム多様化の影響

遠隔情報処理の分野において、システム多様化のもつ意味はとくに大きい。とすれば、そのような多様なシステムを実現する技術というものが、遠隔情報処理技術の大きな動向でなければならぬ。

そのような観点から、情報処理技術の動向を見るときに、第一に指摘されるのは「標準化」の方向である。

利用面において多様であつても、機械設備としての基本的な情報処理システム(ハードウェア、ソフトウェアを含めて)が確立されていれば、それにこしたことはないわけである。

遠隔情報処理システムにおいて、そのような標準システムがあるであろうか。いわゆるタイム・シェアリング・システム(TSS)が、それに擬せられ、そのような見地からTSSの重要性が論ぜられたこともあつた。またさらに古くは、例えばIBMにおけるOS/360

の当初のねらいのように、全角度的な利用形態をカバーする汎用操作システムのイメージが持たれたこともあつた。

汎用操作システムとTSSとの関連自体が「標準システム」イメージのゆらぎの一つの実例をなしている。趣旨からすれば、TSSも汎用操作システムの上に乗せて実現されてしかるべきであつた。しかしながら、TSSは多くの場合、汎用操作システムと別個に、TSS用操作システムとして作られたのが現実であつた。汎用システムの上に乗せる試みも存在したが、その場合も、汎用システムに数多くの改変を加えなければならないということが実証された。

標準システムとしてのTSSへの期待は、それが一種の汎用共同利用システムであるという性質に基づくものである。しかしながらTSS自体がいまだ実験段階にあること、また、TSSの個別形態自身が多様化しつつある現実、また汎用TSSの困難（とくに効率の面において）の予測から、専用的TSSへの後退現象が盛んなこと、これらによつて、TSSがなまの形で「標準的」システムの役割を果すであろうということば期待できないのである。

TSS研究の意義は、まず第一に、そこに遠隔情報処理の多くは最先端の問題が含まれており、したがつてテクノロジーの基盤を形成する上で大きな役割を果していること、第二に、遠隔情報処理のとくに人間・機械の交渉すなわち会話的分野の研究のための実験システムとして優れた性質をもつこと、に求められなければならない。そのようなものとしてTSS研究がもつ意義は極めて大きいのである。また、TSS向応用分野において当然の価値をもつことはいうまでもない。

標準システムのイメージの早産から、現実においては、オンライン諸システム（のソフトウェア）の個別的製作が一つの傾向となつており、*tailor-made OS* という言葉が使われるようになった。これはしかし、現実作業の動向とはいえ、技術（研究）のあるべき方向とはいえず、需要と供給（技術）のギャップの生み出した現象というべきであろう。

この種の個別的製作ではなく、システム多様化に対処する技術が確立された上での個別的製作こそが将来のあるべき姿なのである。とはいえ、この現在進行しつつある個別的製作の実践が技術進歩への圧力でもあり、また技術進展のための素材を提供する源でもあるという評価を忘れてはならない。

一般的に言つて、標準システムの追求は二つの方向からなされる。一つは *Shell approach* と呼ばれるもので、包括的なシステムを作り上げる方向である。他は、*Core approach* と呼ばれ、すべてのシステムの核となるシステムを作る方向で

ある。

しかしながら、そのような直接的な問題意識は必ずしも実を結ばないことが多く、とくに我々の問題の場合そのことが強く感じられる。

遠隔情報処理技術の進展のためには、システムの形態からする発想ではなく、その根底にあるべき技術そのものの観点を確立しなければならない。システムの多様化の認識をふまえるならば、それらの（現実化された、また可能領域での）現象的な諸システムの形態そのものではなく、それらに共通して流れる本質的なイメージをつかまなければならないのである。

このことは、別の表現では、システム開発の「生産性」を上げるということであり、このことが、遠隔情報処理技術においても、技術の本来的な方向として浮び上ってくるのである。

システム開発の「生産性」は、次節で述べる如く、主としてソフトウェアの生産性にかかわってくる。これは、遠隔情報処理システム開発の技術的努力の過半がソフトウェア建設にかかるということである。ソフトウェアのコストは、システム全体の50%を越しつつあると評価されており、またハードウェアにかかるコストは、ハードウェア自体が、遠隔情報処理システムの部品・原材料的役割を果しているということから、システム「開発」特有のコストがソフトウェアにあると推論されている。システム開発のコスト、すなわち、ソフトウェア建設のコストは、システム多様化の認識から、産業的に見て重大な意味をもつ。

すなわち、遠隔情報処理の「サービス」形態において、我国独自のものが開発されたとしても、その基礎システムである情報処理システムの建造コスト、運営コストにおいて、大巾な不利があれば、システム建設における国際競争に敗れるであろう。したがって、一国の産業全般からすれば、情報「サービス」産業よりも、情報処理「建設」産業、すなわちその主体であるソフトウェア製造業が、より基本的に重要な産業といわなければならないのである。そして、そのような基幹産業を成立させる基盤が技術なのである。

2.4 遠隔情報処理システムと中央処理技術

ここで中央処理技術というのは、項目的には、時分割処理技術、多重処理技術、ファイル技術、言語処理技術などのトピックスを含む技術領域に対する仮称である。この中央処理技術は、とくに遠隔システムとは限定しない、情報処理一般について共通に存在するところの、計算機技術一般と考えてもよいのであるが、ここではそれを遠隔情報処理とのかかわりあいとらえてみようとするのである。

2.4.1 中央処理技術の性格と方向

ここで、個別的テーマに入るまえに、中央処理技術の性格について述べておく。中央処理技術は、システムの中央諸装置について、ある機能を実現する技術である。すなわち、中央諸装置を構成するハードウェアによつて、その上に機能を実現するのであり、その実現はソ

ソフトウェアを介して完全になるのである。そこで中央処理技術は、主として、ソフトウェアを作り上げる技術と見てよいということになる。システム構成上の問題点のほとんどすべては、ソフトウェアの上に投影されて現われるので、そのような見方ができるわけなのである。

ここで、ソフトウェアというのは、プログラムの体系という、言葉の正確な意味でのものである。ソフトウェア＝利用技術という誤った俗説があまりにもはびこっているので、ここでもあえて蛇足を加えたのである。さて、ソフトウェアをハードウェア化するという将来の方向が一般に指摘されているがこれは、遠隔情報処理においてもあてはまるであろう。遠隔情報処理システムとして本質的な部分の実現は、ハードウェアと隣接したいわゆるハードソフトウェアにおいてなされているということによるのである。再び蛇足を加えるならば、アプリケーション島の人の中に、たまた、ハードウェアから離れた、ソフトソフトウェアだけを見、ハードソフトウェアは技術的に（あるいは製品的に見て）解決済みとしてこれを軽視する人が居るが、誤りである。遠隔情報処理において技術的にもつとも主要な課題はこのハードソフトウェアに現在は集中しているのであり、ソフトソフトウェアだけを見ては、遠隔情報処理システムを現実構成することはできない。

ここでいう中央処理技術という言葉の「中央」は、現状のシステム構成をもとにいつている。サブ・コンピュータを含む、システムの多様化との関連では、この中央処理技術は、またそれぞれのサブ・コンピュータ部に適用されるべきものであり、その意味では必ずしも中央ではないのである。以上のような意味で、中央処理技術をとらえた場合、その技術が、まだ「発達途上」にあるということを一に指摘しておく必要がある。

したがって、個別的技術の現状のまとめだけでなく、その技術を全般的にとらえた場合の道筋について、そのとるべき方向についての見解を述べておくことは意味があることであろう。第一は、システムのイメージを記述する言語を開発することである。そのような言語はシステムなるものの本質の把握の上で始めて可能であろう。そして、この言語によつて、システム設計法が確立されることになろう。第二は、システム（プログラム）作製のためのプログラミング言語である。システム・プログラムは、システム・イメージを実現するための諸装置の制御を記述したものである。この言語の確立のためには、装置の制御・管理の本質が把握されなければならない。被制御対象である諸装置すなわちハードウェア・システムの記述法も、上に関連して研究されなければならない。この記述（言語）はハードウェアの設計・製作・検査のためのベースにもなるものである。これらの三種の言語が一つの言語体系に統一されれば理想的である。この場合一つ注意しておくべき点は、システム・イメージが与えられ、またハードウェアの記述が与えられれば、システム・プログラムが自動的にシエネレートされることになるのではないか、という夢である。これは技術の立場からは夢というより妄想としなければならない。

第四は、したがってシステム・プログラムの完全性の検証するためのシステムである。これはデバッグ作業のサポート・システムの完備である。それは論理的不整合の検出だけではなく性能評価を含むものでなければならない。

この分野が、現実にはシステム(プログラム)開発の過半の時間と費用を喰っている事実をよく認識しなければならない。

上に述べた諸点は、システム開発の生産性を上げる方向であり、技術本来の方向なのである。そしてまた、技術の主體的な動向ともいべきであろう。技術の動向に対する受身的な観点は、客観的である如くみえても、それは、「輸入すべき」技術の存在を前提にしてはじめて成り立つのである。

2.4.2. 時分割技術と多重処理技術

ここでは、「時分割技術」というのは、計算機の時分割の利用を実現する「制御プログラム」の構成技術を指すものと考えよう。

同じく、「多重処理技術」は、複数個の処理装置を制御するプログラムの構成技術と考える。

(1) TSS技術の意義

TSS(Time sharing system, 時分割共同利用システム)は、時分割技術、多重処理技術によって実現された、特定の目的(共同利用のある形態)をもったシステムの「類」である。

例えば、時分割技術は、TSSのみならず、工業プロセスの計算機制御システムの中でもまた、いわゆるオンライン・システムの中でも用いられる。

しかし、TSSは、その目指す所が、汎用的なサービスの共同利用であるという点から、技術の最先端部分がそこに特徴的に現われるという事情にある。そこにTSS技術に対する意義がある。

TSSをシステムの形態としてみれば、それがそのまま遠隔情報処理システムに移行するわけではない。後者はより広い「類」であって、前者を包含するものと考えられるべきものである。

遠隔情報システムの「システム形態」は、千差万別で固定化されないであろう。しかし、それを実現する技術としては、形態的類似の割合以上にはるかに多くの共通部分をもつのである。それはとくに、時分割制御技術においてあてはまる。

多重処理方式について、TSS研究において追求された方向は、共通の主記憶装置を中心

にした「対等」な複数個の処理装置からなるシステムである。しかも、各処理装置は機能面においても対等で、個性をもたせぬ方式である（Multicsの例など）。

しかし、この方向の是非については実際的には何等結論が出ていないというべきである。強いて予想すれば、むしろ、処理装置間の「分業」体制の方が現実においてまさるのではないと思われる。

従来行なわれた分業的複数処理装置システムは、処理装置が能力において異なる一場合によって機械語も異なる一もの同志の結合が多い。また処理装置間通信もチャンネルを介すなどの「まわりくどい」方式が多い。

そのような場合には、システム・プログラム開発の重複ならびに、処理装置間通信のあい路が意識されている。また「信頼性・可能性」の点においてシステムの動的な再構成が難しい。

対等な処理装置のもっとも重要な利点は、システムの動的な再構成が可能な点にあるというべきであろう。

分業的システムを意味あらしめる程度の十分多数の処理装置の結合実験はまだ行なわれていないし、それに適するハードウェア・システムもまだ出現していない。

しかし、遠隔情報処理「ネットワーク」は内部に多数の結合された処理装置を含むように進化するのである。とすれば、そこでは分業体制が不可避であろうし、それは実現する技術が開発される必要がある。

(2) 処理装置制御技術

TSSにおいて開発されたこの分野の技術は、1台ないし数台の処理装置の能力を多数の「仕事」に分割する技術である。

それを統一的に把握するために提案された「プロセス」の概念（Multics）は高く評価されるべきである。

それに対応する萌芽的な要素は、もちろん先出の諸システムにおいて見ることができるのであるが、しかし、それを概念化した点、そのことによってシステムへの構成法が透明化された点が重要なのである。

各プロセスには「状態」が附帯されて、それによってプロセスの制御が行なわれる。

現在の主流のハードウェア構成思想によれば、処理装置制御は、「割り込み」装置を介して行なわれる。したがって、プロセス制御の基本部分は、割り込み処理ルーチンを含むプログラムとして実現される。すなわち、割り込み処理は、プロセス制御の中に吸収されてし

まりわけである。

割り込みの概念は、ハードウェア技術の論理から進化出現したものであり、したがってシステム制御の論理と微妙な違いを持っている。そこで（ハードウェア技術が進歩した現在では）プロセス制御の基本部分をハードウェア的に実現する方向が提唱されている。

（Lampson等）。

同様のことが「保護機構」についても言える。保護機能はハードウェアの誤動作、ソフトウェアの不測の誤り、に対してシステムを保護する重要な機能である。それはやはりハードウェア的論理で実現されているのが主なのであるが、Multics等では、セグメントの概念に関連して高度化されている。Dennis等はそれをcapability（権能）の概念で統一しようと試みている。

(3) 主記憶制御技術

ArdenやDennis等によるページ・セグメント方式（2次元アドレス方式）は、計算機システムにおける最近の大きな進化の一つに評価されている。これは、ハードウェア的な主記憶装置を概念化し拡張したものである。

これによって、主記憶装置の補助である二次（大容量）記憶装置の機能が、主記憶装置に統合される道が開かれた。ハードウェア形態的な、記憶装置の「階層的」構成がこれによって統一的に把握されることになったのである。

2.4.3. ファイル技術と遠隔情報処理

遠隔情報処理システムにおいて、処理装置自体は、集中化・中央化の傾向とともに、逆の分散化・地方化の傾向をもっている。

ところが、「データの貯蔵庫」であるデータ・バンクについては、現在分散化しているものを集中化しようとする強い要請が働いている。

すなわち、蓄積され利用されるデータの量が大量になるとともに、そのデータの共同利用の要請が高まる。

一方、データ内における関連が複雑・高度になるために、一個所に集めておくべき相互に関連するデータの量が増大してくる。そもそもデータというのは、単にある種の数値の累積というよりは、むしろその本性は、それらの間の関連そのものなのであり、したがって本質的に集中化の要請をもつものである。

(1) TSSの汎用ファイル・システム

TSSは、計算処理サービスの共同利用とともに、集中化されたファイルの共同利用を

具体化している。したがって、それは多ユーザ間でのデータの蓄積、取出し、交換のシステムの一つのひな型を構成している。もっとも、そこでは(データの種類としての)プログラムの作成、共同利用に重点があるといえるのであるが。

一方、データ処理の高度化により、統一化され、統合化されたデータの集積を扱うための汎用ファイル・システムが実現してきた。これの多くは、オペレーションの形態としてはバッチ処理であるが、TSSと結合した形態も試作されている。(その一例がSDC社のTDMS-Time Data Management systemである)

情報処理システムは二つの焦点をもった楕円にたとえることができよう。一つの焦点は「処理」ないしオペレーションの焦点であり、他は「データ」の焦点である。

TSSはそうすると、処理の焦点を強く浮び上がらせ、他の焦点をやや小さく伴ったシステムと見ることができ、汎用ファイル・システムはデータの集点を主とし、処理ないしオペレーションを従としたシステムと見ることができる。

(2) 汎用ファイル処理システム

ファイル処理の基本的な機能として

ファイルの創成

ファイルの更新

ファイルの検索

レポート作成

が挙げられる。(文献)

汎用ファイル処理システムは、ある固有のファイルの固有の処理をそのまま実現するのではなくて、ファイル処理の基本的な機能を(統合化された)基本的なファイル形態の上で行なり、基礎的なファイル・システムである。したがって、そこに蓄積されるデータ自身ならびにデータ処理の具体的な処理の指示は、あとで、その基礎的システムの上に、用途に応じて乗せられるのである。

これによって、処理の指示は、使われるファイル装置(ハードウェア)の個性や、利用

* 竹下：汎用ファイル処理システムの性格とその発展、情報処理 vol 10, No 2, March 1969, PP 84-93

水野：汎用ファイル処理システムについて、プログラミングシンポジウム報告集(第9回) Jan 1968

者ごとの恣意的なデータ構成のゆらぎに影響されずに与えることができ、たとえその下部構造がかわっても、その指示(プログラム)をいかにすることができる。

汎用ファイル処理システムは、入出力制御システム、データ管理の一つの発展形態と考えられ、一方、COBOL等を中心とするデータ処理用言語などの統合化を見ることができると。

その先行的な一例はGE社のIDS(Integrated Data Store)である。これは大容量記憶装置上のデータ構造をもとにし、それに対する処理をCOBOLの拡張言語で与えることのできるものである。プログラムはブリプロセッサによりCOBOLに変換され、さらにコンパイルされ実行される。そのプログラムが統合化されたデータ・ストアに作用するわけである。

その後IBM社のGIS、他多くのものが作られ始めている。バッチ操作(IDS、GIS)が多いが、端末機能をもったもの、TSSと結合したもの(TDMS)がある。

遠隔情報処理システムの観点からは、TDMS的な方向が将来の方向であるということができよう。

(3) ファイルの管理

統合化されたファイルを、オンライン・ファイルとして持つ場合、ファイルを管理の問題点として次の項目が挙げられる。

- a ファイルの名前の管理と呼び出し
- b ファイルのスペースの割当て
- c ファイルの共用と保護

名前の管理 ファイルの管理は、データ構造の管理の上部構造と考えることができる。したがって、ファイル名の管理は、概念的には「木構造」をとり入れることが自然である。

すなわち、ファイルをグループに分類し名前をつける。グループはさらにサブグループに分けることができ、その場合は組合せ名前(qualified name)となる。

グループ分けの特殊な例として世代データグループ(generation data group)がある。これはファイルの改訂歴を保存するもので、同じ名前で最新のバージョンを指す。古いバージョンはバージョン番号の指定によって取り出される。

このようなシステムを管理するために、カタログ(ディレクトリとも呼ばれる)が用いられる。

スペースの割当て ファイルのスペースの割当ては、概念的には、物理的媒体(の種類や

境界)と無関係になされるのが理想である。しかし、現実には、能率、保管の都合で媒体の取扱いが重要である。しかし、それはシステムにとってであって、ユーザに対しては、装置、媒体からの独立が実現される方向にある。

ファイルの共用 ファイルの共用はTSS的システムすなわち共同利用システムでは重要である。共用の制御は木構造を利用すれば自然に行なえる。むしろ問題なのは共用から発生する保護の問題である。

ファイルの保護 共用のない場合のユーザ間の分離は、木構造で実現されている。共用を導入した時に保護の問題が発生する。

アクセス権制御はその一手法であって、木構造の枝に、その枝を通過して到達したファイルに対する操作の制限であるアクセス権をつける。しかし「病的現象」の出る可能性もあり、種々の試みがなされている。

他にファイル名の暗号化、ラベル付けの手法が使われることがある。

ファイル保護でもっとも重要な問題は、システムの故障に対するものである。その対策の原理は「写し」をとることであるが、その写し(dump)と復帰に関していくつかの手法が試みられている。

(4) データ構造

データ構造の概念は、ファイル・システムの観点からだけでなく、より広く、例えば、自動設計や図形処理あるいは言語処理またはシステム制御の分野において重要な役割をし、ソフトウェア技術の基本的概念になりつつある。

データ構造の問題としては

- a データ構造の構成
- b データ構造の表現法
- c データ構造の操作(その言語)

の側面がある。

データ構造の構成 データ構造処理システムとしては種々のシステムが試作され、システムによって特質がある。構造の構成法はその表現法と関連があり、そこから差が出てくるのであるが、「集合論」的観点が基本になりつつある。データ構造としてより高度の概念は「関係」の概念であるが、「関係」は集合的見地から厳密に把握される。

データの個別項目として例えば

コード

名 前
本 給
年 令
勤続年数
専 攻

を考えよう。このデータの集合は、例えば年令層、勤続年数、専攻分野によって「構造化」される。また

名 前
父親の名前
子供の名前のリスト

のようなデータの集合は、「名前」によって系図的構造がつけられる。

データ構造の表現 データ構造技術の一つの眼目は、記憶装置内でそれをどう表現するかという点にある。

リスト (list), リング (ring), プレックス (plex) 等は、そのための中間概念であり、これらは多くの場合そのまま記憶装置に写像される。それは構造の自然な形の表現であるが、一方能率の観点から、それをハッシュ・コード (Hash code) 等を用いて変形表現することもある。

データ構造の操作 LISP 言語はリスト構造操作に対する言語である。データ構造の構成、表現によって種々の言語がある。

CORAL, APL, ASPなどはその例である。

「集合」的観点に立つものでは、Insert (あるものをある集合に入れる), Remove (集合からとり除く), Find (集合の中で見つける) などが基本操作になる。

「関係」のレベルでの基本操作の例は

Attribute (Object) = Value

属性 (対象) = 値

の3項関係で、2項または1項を与えて他の項を求めるものである。すなわち

1. $A(O) = ?$

2. $A(?) = V$

3. $?(O) = V$

4. $A(?) = ?$

5. $?(O) = ?$

6. $?(?) = V$

が基本操作になる。

例えば

Father (John) = Jim

Son (Jim) = John

のようなデータに対して, Father (John) = ?

はJimであり, ? (Jim) = JohnはSonである。

この操作は, プログラミング言語の基本操作でありうるとともに, (端末から問合わせの場合) コマンド言語でも使われうる。すなわち, 形式を若干変換すれば

Who is Father of John ?

What is Jim for John ?

の形のコマンド言語になる。

データ問合わせシステムの基本コマンドはこのレベルの言語に通常想定されている。

(5) ファイル・システムとハードウェア

遠隔情報処理システムにおけるファイル・システムの重要性をデータの集中化の傾向と合わせ考えるならば, ファイルシステムにおけるハードウェアへの要請も大きくなるといわなければならない。

一つはファイル媒体の問題である。すなわち大量化に対処する媒体と装置である。

現在磁気ディスク, 磁気カード装置がその用途にあてられている。前者の集団ディスクで約2億字, 後者のデータ・セル等では5~10億字の容量が1セットに当たる。

しかし, 需要の予測が100億字, あるいはそれを大巾に上廻るとすれば, 更に他の装置が開発されなければならない。

その一つは, 光学的なファイル装置であろう。この場合, 必要なことは, データの記録取出し, 改訂が一貫して自動化されるべきことである。これは従来の情報処理のハードウェア・システムの枠を越えるものであろう。

第二は、磁気テープの再認識であろう。磁気、機械システムの寿命はかなり永いものを見る必要であろう。このためには、磁気テープの自動着脱、管理のシステムを開発しなければならない。

これらは、容量の要請、データの呼び出しサイクル、コストの点によって慎重に検討される必要がある。

ファイル媒体の問題の他に、ファイル装置の制御装置も再検討を要すると思われる。現行の制御装置の機能は、効率的なファイルシステム実現に必ずしも適しているとは言えないのである。そのために中央処理装置をつぶしている面がなきにしもあらずである。

巨大ファイル・システムを構成するに当ってはハードウェア構成のみならず、制御方法についても、システム工学的アプローチを強力に採用する必要があり、また、そのようなアプローチこそ有効である。

システム工学的アプローチは、処理システム内部の問題に対してよりは、比較的適用が容易な面をもっているが巨大ファイル・システムを構成するという大きな問題の解決には多くの努力を必要としよう。

2.4.4 言語処理技術と遠隔情報処理技術

遠隔情報処理システム開発との関連で言語の問題をとらえれば、まず大きなトピックスが二つある。その一つは、会話形プロセッサの問題である。他はシステム開発のための言語体系の問題である。

次に、遠隔情報処理システム開発において上の二つの問題と関連して、多種多様な言語が出現することから、その言語処理の技術自体が遠隔情報処理システム開発の大きな前提的技術になる、ということである。

(1) 会話形プロセッサ

遠隔情報処理システムにおいて、諸端末におけるユーザに対し、何らかの形で会話的サービスを提供することが必要になる。この会話的サービスを実現するのが会話形プロセッサ(プログラム)であり、そのそれぞれについて会話形言語が対応することとなる。

現在の多くの会話形プロセッサは主としてタイプライタ端末を前提にしている。

会話形言語には、コマンド言語と会話的プログラミング言語がある。

コマンド言語は、システムとの間で指示のやりとりをするための言語である。コマンド言語を広くとらえれば

(i) ファンクション・キー(押ボタン)式

(ii) 線形言語式

(iii) 指示選択式

の形式に分けられる。プロセッサはインタプリタ型である。指示選択式は文字ディスプレイ端末のように出力の速い端末に適している。

会話的プログラミング言語には、JOSS, BASIC, CAL, RUSH, BRUIN, TINTなどがあり、多くは既存のプログラミング言語、FORTRAN, ALGOL, PL/I等をベースにしたものである。これらの会話形言語の詳細については当財団の報告書43-S005に記載されている。

これらに対するプロセッサには、コンパイラ型、インタプリタ型および混合型がある。

BASICはコンパイラ型の典型であり、TINTはインタプリタ型であり、JOSSは混合型である。コンパイラとインタプリタの境界は浮動であり、言語に対してプロセッサの型が一意的に定まるものでもなく、それは具体化の方法に依存している。

(2) 会話形言語と端末

端末における会話の様態は、端末装置の種類、個性によって異なってくる。従来会話的使用についてその技術がよく開発されているのはタイプライタ型端末である。タイプライタ自身がもっともよく普及しているという事情とともに、タイプライタでは、その情報交換形態が線形的であり、それが言語の線形性とよく合致しているからでもある。

会話における内容表示の形態には「言語」と「図形」がある。この二つの形態は人間において発達心理学的にそれぞれ独自に発達する、二つの大きなコミュニケーション機能である。

それに対して機械の側でそれぞれに対応した端末機能(装置)が実現されるわけであるが、その際に、それぞれの個別技術的な諸事情が加わって、端末装置はますます多様化することになる。

端末装置の多様化は、CRT(cathode ray tube)等の図形表示端末の方向、押ボタン電話器のような簡易化の方向においてますます促進されるであろう。これは、会話様態自体の多様化と相乗じて会話用ソフトウェアの多様化を加速することになる。

会話については、その内容表示的側面とともに、その形式的な側面が独自に考察されなければならない。この後者については次章があげられている。

(3) システム開発用言語

システム開発において数多くのプログラムが製作されなければならない。

その一つの類は、言語処理プログラム自身である。多種多様な会話形言語およびその他の

言語に対して、多くのプロセッサ(プログラム)が作られている。その数は言語の種類、目的に応じた処理の具体化方法の種類を相乗したものとなる。これについては、技術の方向として「コンパイラ・コンパイラ」などが目指される。

第二の、そして、システム開発にとって本質的なプログラム類は、システムの管理ならびに制御を司るプログラム類である。

この類は、ハードソフトウェアとも言われるシステムにとって重要な部分であるが、その開発に多くの難点をかかえていることは、伝えられている Multics (大型 TSS のもっとも野心的な試み) の難行の噂からも明らかである。

このプログラムの開発の難点の一つは、そもそも(ハードウェア)システムの制御方法をいかにまとめるか、という問題であり、第二はそのような制御方法をいかに記述し、プログラムとして実現するかという問題である。

これらについては後述するが、ここでの関連は、ある「システム開発用言語」が得られたときに、それに対する良い言語プロセッサが得られるか否か、という問題である。これはシステム・プログラムの性能として、システムの性能に直接かかわってくる問題であるが、その解決のためには、高度精密な研究がなされなければならない。

現在の実践の多くは、人手による「コンパイル」ないし、人手による「書きなおし」で逃げるのであるが、これが技術の究極の姿でないことはいうまでもない。

(4) コンパイラ・コンパイラの技術

遠隔情報処理技術における言語処理技術の重要性は、既述のとおりである。計算機技術の本性から言えば、言語処理技術は、計算機化された言語処理システムに結実すべきものである。言語処理プロセッサ自体が、計算機によって作り出される方向である。これは、標語的に言えば、コンパイラを作るコンパイラを作る方向である。

この方向の努力は、J. Feldman 等によって Translator Writing Systems (TWS) として、総合報告されている。(文献)*

1961年頃、E. T. Irons (Syntax Directed Computer), R. A. Brooker 等 (Computer-Computer) などを先駆的業績として始まったこの分野は当初のセンセーションのあと、しばらく沈滞的ムードにあったということができよう。

* J. Feldman et al: Translator Writing systems, Comm ACM, vol 11/No 2, Feb, 1968, PP 77-113

それは、能率の面に問題があったからで、A Oplerの報告にある。「よいコンパイラは人手によったものであり、コンパイラ・コンパイラへの努力よりも、プログラムの管理の努力の方が大きな成果をあげた。」という評価に事態は集約されていたのである。

しかしながら、この数年この分野は再び注目を集めはじめている。それは、それまでの期間の地道な研究者の努力により、能率改善の方向が見えてきたからなのである。

(i) シンタックス

プログラミング言語などの人工言語の文法(シンタックス)の記述は、Backusなどにより創案され、ALGOL60に採用された記法により明確にされた。これを契機に人工言語のシンタックスの研究が進み、Context Free Languageの理論をはじめ多くの成果が生まれ始めた。

TWSの研究テーマの一つは、このように形式化されたシンタックス(の定義)により言語処理ルーチン(パーザ)の作成する方法である。パーシングの方法として、Top down法、Bottom up法が研究され、また、それと平行して、Operator Precedence法、遺移マトリックス法、Production Language法等が考案された。

現在では、シンタックス定義を自動的に処理しパーザを作製すること、すなわち、定義を、パーザ的言語に自動的に落す技術が詳しく研究されている。

概括的に言えば、シンタックス面での研究はかなり煮つまってきており、精密な研究の対象になってきているといえよう。すなわちContext Freeの立場からは、より能率の高いパーザのための「コーディング」的な精密な検討であり、他方、Context Sensitiveな高度の文法体系の理論的検討である。

(ii) セマンティックス

人工言語の意味論的側面(セマンティックス)は、理論的には今後多く残された分野である。

言語の意味の定義の面では、McCarthyの提唱によるLispの言語による、意味の定義法を、IBMのVien研究所かPL/Iの定義に適用した大きな努力が注目される。これは形式的に定義された仮想機械の上で、言語の意味を設定する一方法である。

しかし、定義としてはまだいくつかの方法が考慮されようし、Vien研究所の方法も、まだ自動処理とは結びついていないようである。

Translator writing systems (TWS)としては、今までのものは、シ

ンタックスとセマンティックスの結びつけ方、すなわちどう結びつけるかという点に関心があった、と見ることができる。すなわち、シンタックス・パーザから、セマンテックスルーチンと呼ぶやり方に種々の工夫が見られる。

すなわち、SDO社のMETA言語シリーズ、FSL言語、COGENT言語など、それぞれ独特の形式が見られる。

しかし、実際のところ、セマンテックス・ルーチンの主体はむしろ、Context Sensitiveな部分の処理にあるのが現状である。これはシンタックスのはみ出た部分と見ることができる部分である。

真の意味でセマンテックな部分は、「コード発生」と呼ばれているものに相当するが、この部分はまだ十分究明されていないと思われる。すなわち、正しいコード発生を保証と「良い」コード発生は、プラクティカルにも重要な意義があり、理論的な取扱いが望まれる。前者は「意味の定義」と関連があり、後者では文脈(context)との関連のつけ方が問題となる。

意味の定義の別の方法として、Galler等の「マクロ」の高度言語への拡張は注目されなければならない。これは言語の拡張性へのアプローチであり、裏の効果として、言語の単純化へのアプローチでもある。

この場合、言語理論的な究明とともに、その「マクロ」の処理方法が、コンパイラの能率の点から研究されるべきであろう。

言語処理の分野は、その重要性に比して、世にあまり知られていない。上の記述も、その観点からは、あまりに舌足らずと言えよう。がとにかく、この分野は、計算機技術の大きな部分であり、まだ多くの研究課題をかかえた分野である。遠隔情報処理システムが、将来の計算機システムの主流であることを考えると、そのような言語処理研究がまた、遠隔情報処理システムの技術的基礎なのである。

2.4.5 システム開発技術の現状と方向

前述した如く、システム開発技術のとるべき方向として、システム開発用言語がある。現状において、しかし、この分野は極めて初期的な段階にある。

実のところ、例えば、システムというものと「システム記述」との関連、またシステム記述とシステム「プログラム」との関連という原理的な問題も、まだほとんど明らかにされていないのである。ここにおいては、現在続けられている苦しい努力を見なければならぬ。

D. L. Parsan: More on simulation languages and design methodology for computer systems, Proc: SJCC69, PP739-743

(1) システム・プログラム用プログラミング言語

システム・プログラムのためのプログラミング言語は、まだそのものとしては出現していない。現在はその前段階にあると言ってよい。

現在の実務的实践において、システム・プログラムの殆んどは、アセンブラ言語によって書かれている。アセンブラ言語は、機械向言語 (Machine Oriented Language) として、システム・プログラミングにとっての究極の避難場所である。

しかし、アセンブラ言語には、欠点があり、そのため、システム・プログラム作製の生産性の低さの一因となっている。アセンブラ言語の欠点というのは一言にして言えば、プログラミング言語として、無構造的、無概念的なことである。それは、一方において、アセンブラ言語の長所であり存在理由なのでもあるが、その無構造化・無概念性によって、プログラム作製者に、構造化、概念化の作業の反復を強制し、またデバック作業を困難ならしめる。

このことは又、プログラミング作業の「評価、管理」に困難をもたらしている。アセンブラ言語によって作られたプログラムの品質が、プログラマの技能にあまりにも依存するのである。

アセンブラ言語の欠点から抜け出すためにいくつかの実験が行なわれている。それは、既存の(高度)プログラミング言語を流用する実験であり、それによって、システム・プログラム用言語の条件が明らかになりつつあるのである。

FORTRANの使用実験はかなり行なわれているが、しかし、その言語の本性上、ほとんど部分的、局部的にしか適用されえないことが明らかにされている。

PL/Iは、MITのMACプロジェクトにおいて、Multics開発に適用され、その記述能力において優れていることが明らかにされたが、能率の点で問題をかかえており、まだ問題自体が十分に解析されていない。

我が国でも、同じ実験が、日立中央研究所のTSS開発でなされている。この場合は、PL/Iは記述用のみに使用され、コンパイル作業は人手で行なわれたが、そのためにかえて、能率の問題がある程度解決されているのは注目に値する。

同様の実験は、電気試験所において、ETSS開発で行なわれている。ここでは、やはりコンパイラがない点を逆用して、PL/Iより高度の、それに連想的機能を加えて拡張したAPL言語を実験し、よりよい記述能力を実証している。

別の実験のアプローチは、いわゆる「シミュレーション言語」の使用実験である。システムをGPSSで記述し、それをシステム・プログラムに移した実験が報告されている。

シミュレーション言語は、システム記述用言語として、よい特性をもっている。それは並列動作の記述の面である。しかし、システム記述と、システム・プログラムとの原理的関連が明らかにされていない点に壁がある。

これらの実験の結果から、シミュレーション言語的要素、手順言語的要素の結合に、システム開発用語の方向があることは言えるであろう。

(2) システム・プログラム・デバッグ用システム

システム(プログラム)開発の難点の一つに、そのデバック作業の困難が挙げられる。

「自動デバック」システムは原理的に不可能であるが“Computer-aided debugging system”は、實際上極めて有用なものである。

それはプログラムのトレーサ(シミュレータ)を核とするものであるが、取扱いの面や能率の面で研究課題を残している。

この分野は、システム開発個所で、数多く試作、試用されているが、成果があまり公表されておらず、経験の交流、成果の一般化が不十分である。これは、デバック・システムに対して、世の中の認識が十分でない点に一因があるようである。したがって、努力、経験が特殊なもの、地味すぎるものとして埋没するよう見受けられる。

この分野を向上させるためには、問題の重要性の認識を広め、経験を公開し公流する必要がある。

(3) システム・プログラムの品質とシステムの性能

システム・プログラムの困難は、それ自身が論理的に複雑なものであることから、論理的な正しさを求めることがその困難の一つであるとともに、そのプログラムの品質がシステムの性能に直接ひびいてくることである。

現在までの経験によれば、最初から理想的なシステム(プログラム)が組めたという事例はなく、最初の曲りなりにも動作する(すなわち一応論理的には正しい)プログラムの完成から、システムの実用に至るまでに、かなりの期間、「チューン・アップ」が行なわれる。

このチューン・アップは、虫(バグ)の除去の他に、性能向上に大きな比重がある。そして、システム・プログラムの改訂によってシステムの性能が一桁以上あがることはまれではないのである。

このことは、システム・プログラムを不変とすれば、10倍以上のハードウェアを入手したことと等価なのである。ソフトウェア・コストの高率化から、ハードウェアの性能向

上に期待する声もあるが、それがすべてではあり得ない。チューンアップの度合いは、効果費用曲線で定められるであろうが、現在のソフトウェア技術の水準を仮定すれば、ハードウェアのコストの比率が数パーセント程度以下にならぬかぎり、チューンアップは不可欠である。

ソフトウェア技術としては、そのようなチューン・アップを必要としない程度の、初めから高品質のプログラムを作製できる技術を開発しなければならない。

その際必要なのは、「コーディング技術」の重要性の再認識である。そうは言っても、それは、コーダ（コーディング技能者）の養成に退歩するものであってはならない。コーディングを「技能」としてではなく、「技術」として取り上げなければならないのである。

コーディングは、高度言語と、「与えられた」機械とのインターフェイスである。コーディング技術の行きつく所の一つは、ハードウェアの改訂でもあろう。

この観点から、現行のシリーズとは別系統の、高度言語向きに設計された計算機の提唱がなされており、それが第四世代の計算機の一形態となる可能性も無視できない。

高性能の遠隔情報処理システムの実現のためには、ソフトウェアからハードウェアにいたる検討がつねに繰り返さなければならぬのである。ソフトウェアの表層の技術、単なるアプリケーションの見地だけでは、遠隔情報処理システムの将来はあり得ないのである。

システム・プログラムとシステムの性能の関連で重要な点は、前に指摘したように、システム・プログラムの質によってシステムの性能が大巾に変わる点にある。これは、システム・プログラムに具現されている制御方法の良否と、プログラム自身の良否である。

この後者の事情が、システム開発において、深刻な問題を提起するのは、実にこのことによって「システム工学」的的手法がうまく適用できなくなる、ということなのである。

すなわち、プログラムの実行速度が零という理想の極限においては、システム制御方式の検討はかなりうまくいく。これは理論的モデル、あるいはシミュレーション法により検討される。ところで、プログラム自体の速度ならびにサイズは、前もってあまり良い精度で予測できないのである。サイズにおいて2倍程度、速度においては10倍程度の狂いはよく起りうる。この「作ってみなければわからない」事情が、制御方法、システムの評価にフィード・バックされる。

理想状態での理想的制御方法も、プログラムにしてみると龐大になったりして、むしろ単純な制御方法の方が現実において効率がよいことなどもある。

システム工学のもっとも良い道具であるはずの計算機システムが、その内部システムに

については、システム工学的アプローチをあまりよくうけつけないという事情は、システム開発の際よく認識する必要があり、このことはまた、別に独自の「システム工学」が建設されなければならないということも示唆している。

2.5 端末と制御

まえがき

遠隔情報処理システムにはいろいろの種類がある。たとえば単なるデータ収集システムもあれば簡単な問合せシステムもあり、同じ問合せシステムでも種々のものがある。さらに、汎用TSSのように、人間と電子計算機のやりとりが会話形式で進むものもある。

これら千差万別のシステムに対して本章では通信という立場から共通して言えることを述べる。その場合一たん問題になるのは会話型システムであるか否かによって、いちじるしく異なる部分があることである。そこで本章では会話型であるか否かで異なる場合はその都度いずれであるかを明記する。

話の中心は2、3節および4節の会話形システムであるが、それらの具体的な方法は、通信における誤りの制御方式に依存する。そこで、まず第1節で誤り制御方式を詳述する。本質的には、これは通信において誤りが発生したときの対策で、信頼性の問題である。このために、本節で述べることは特に会話形に限らない。ただし、種々の誤り制御方式のうち、いずれを選ぶべきかを決めるときは会話形であるか否かによって大変異なるので、この点に重点をおいて述べる。

次に、第2節ではまず会話に適した通信の方法の必要性を示す。つづいてその必要性をみたす諸方式とそれらの利点、欠点を説明する。会話形システムの具体的な方法については、第1節に示す誤り制御方式に依存するが、すべての方式について記すのは煩雑であるので一つの代表的な誤り制御方式を例にとる。しかし、他の誤り制御方式にあてはめることは容易である。

このことは、第3節および第4節においてもまったく同じである。

第3節に示す割込み(QUIT)機能は会話形システムにおいては必須であるといわれているが、会話形でないシステムの場合も有効な事柄であるので、会話形の場合とそうでない場合の両方を示す。

第2節、第3節では会話形通信システムにとって必要とされることを示すが、第4節ではさらに一歩進めて会話形システムにおける一つのあり方を示す。

以上の1~4節はシステム的な要素が強い事柄であるが、人間と計算機の通信においては端末は重要な要素であるので、本章においても、第5節でオンライン端末のあり方を示す。

2.5.1 通信誤り対策

通信において発生する誤りを制御する方法（これを誤り制御方式という）には種々の方式があり、いずれを採用するかによって信頼性のみならず通信システム（端末の方式、計算機の方式およびソフトウェア・システム等）が大きく変わるので、その選択は非常に重要である。そこで、まず誤り制御方式の分類と比較を示す。ただし、ここでは現実に行なわれている方式に限定し、今後の研究が期待されている方式（たとえば、誤りの制御を電話交換局が行なう方法など）については触れない。

誤りの制御方式は大別すると次のようになる。

- (1) 情報の受信側で誤りを検出し、それを送信側に知らせ、通信側が訂正動作を行う方法で判定帰還方式（Decision Feedback systems）または再送要求方式と呼ばれるもの
- (2) 情報の受信側で誤りを訂正する方法
- (3) 情報の送信側で誤りを検出し訂正動作を行う方法

表1.1はこれを示す。

	検 出	訂 正
受 信 側	誤り検出符号	誤り訂正符号
	垂直パリティ	ハミング符号
	水平パリティ	サイクリック符号
	群計数	
	定マーク	
	特別な情報	
	連送方式	連送方式（多数決）
送 信 側	返送方式	

表1.1 誤り制御方式の分類

群計数方式は水平パリティ方式におけるチェック・キャラクタを2文字以上にして精度を上げたものである。

定マーク方式は n ビット中、1の数が r 個のもの（ nCr 個）のみを使用し、残りの $2^n - nCr$ 個は使用しない。このため冗長性が高い。

特別な情報を追加する方法も考えられる。たとえば銀行などで金額を送る場合、その桁数を追加する方法などである。

表1.1, 受信側検出に属する連送方式というのは、ある情報を送信する場合、例えば同じ情報を2回送信し、受信側で1回目と2回目を比較し、同じなら正しいとし、異なっていれば異りとして送信側に知らせるものである。

ハミング符号とかサイクリック符号のように送信する符号に冗長性をもたせれば、受信側で誤りを訂正できることがある。

多数決論理による連送方式というのは、たとえば同じ情報を3回送信し、受信側で3つの内2つが同じなら、それをもって正しいとみなす方法である。

次に、返送方式というのは受信側は受信した情報をそのまま送信側に送り返す。送信側はそれと、はじめに送ったものを比較し異なっていれば同じことを再実行する。なお以上のように情報を全部チェックするかわりに、前もって決めた約束にしたがって、チェック符号のみを送り返す方法も考えられる。(但し、以下特にことわらないかぎり返送方式という場合は情報全部をかえすものをさす)。

次にこれら、種々の誤り制御方式のうち、会話形システムにとって、いずれの方式がよいかを考える。

(1) 連送方式

この方式では、情報を送るのに同じものを何回か送ることになるので、情報送信に時間が長くなり会話形システムには適さない。

(2) 返送方式

この方式は、制御は簡単であるが、伝送の冗長性が高い。ただ連送方式とちがって、会話システム(一般にTSSなど会話形システムでは双方より同時に情報の送信を行う必要はないと言われている)においては全二重通信方式(双方より同時にデータを送れる方式を全二重通信方式とよぶ、これに対して、同時にデータを送れず、受信と受信を切換えて送受信を行う方式を半二重方式という)にすれば受信しながら、同時にそれを送り返せるので、送信時間が長くなることはない。このため、返送方式は広く用いられている。

(3) 判定帰還方式

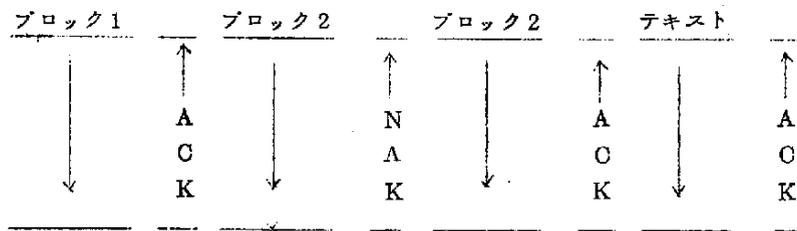
この方式では受信情報に対してそれが正しいか否かの応答のみをかえすので、連送方式や返送方式に比べて伝送効率が高い。

しかし、応答をかえすので制御が複雑になる。

図1.1は判定帰還方式を示す。この方式では情報をテキストの形式にして送信する。すなわち情報の前後に特別な伝送制御符号(Transmission Control Character: TC)のSTX(Start of text)とETX(End of Text)をつける。テキストが長い場合は伝送効率を上げるため(場合によってはバッファの長さを短くするためなど)にテキストをいくつかのブロックに分ける。

誤り率の高いシステムではブロックの大きさを小さくする。普通1ブロックの大きさは90~140字位である。

図1.1 判定帰還方式



ブロックの場合はテキストにおけるETXのかわりにETB(End of text Block)符号をつける。

さて水平パリティまたは群計数などではETXまたはETBのあとにBCO(Block Count Character)をつけて、水平パリティとか群計数の値を受信側に示す。受信側では受信テキストまたはブロックの垂直パリティとか水平パリティまたは群計数などをチェックし、誤りの有無を検出する。誤りがなければそのことを送信側に示すために、ACK(Acknowledge)符号を送る。もし誤りがあればACKのかわりにNAK(Negative Acknowledge)符号を送る。送信側はこのNAKを受けると前のテキストまたはブロックを再送する。

以上が判定帰還方式の基本的なものであるが、種々な理由でその変形が考えられる。たとえば、ACKとかNAK自体も通信中で誤る可能性があるので、高い信頼性が要求されるシステムの場合、情報の送信側はACKまたはNAKが届かないと、受信側に対してENQ(Enquiry)符号を送ってACKまたはNAKの応答を要求する方法も広く用い

られている。図1.2はこれを示す。

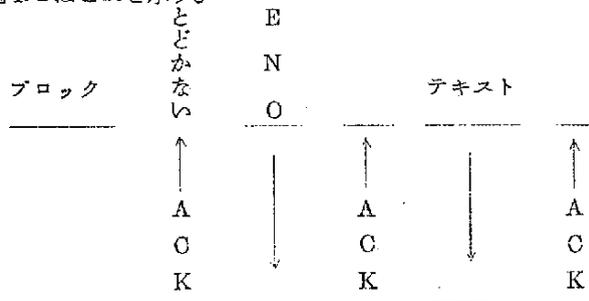


図1.2 判定帰還方式で応答が正しく届かない時の処理例

また別の方法としては情報の送信側はACKまたはNAKが届かない場合でも前記の方法のようなENQは用いずに前のテキストまたはブロックを再送する方法もある。

さらにテキストまたはブロックに番号をつける方法、ACKに番号をつける方法、ACK、NAKを2字以上連続して送って化けを防ぐ方法、交互性ACKによる方法、それらを併用する方法など種々の工夫が行われている。

(4) 誤り訂正符号による方式

前の誤り検出符号を用いる方式に比べて、この方法では誤りを受信側で訂正できるので特に誤り制御手順といったものは必要ではない。しかし、この方式には冗長性が高いことのほか次に示す欠点がある。

すなわち、ブロッキングを行なうシステムで、しかも情報に対して、応答をかえさないシステムではターミナルはあるブロック送信後、すぐ次のブロックの送信を行なうので計算機側には短時間にいくつものブロックが届く可能性があり、一つの受信ブロックのデータをメモリーに入れる動作とか、編集の処理などが終わっていないうちに次のブロックが入ってきて、計算機の処理が追いつかないことがある。

この対策としては、ブロックを受信した場合、何らかの応答をかえす必要がある。ところがこの何らかの応答をかえすということは情報に対して応答をかえさなくてもよいという誤り訂正符号の利点を放棄することになる。

以上述べたように、誤り訂正符号を使用してもブロッキングを行なうシステムにおいては受信側で受信情報に対し応答をかえさなくてもよいという利点は期待できない。期待できないというよりはむしろ、前の再送要求方式の方が優れている。なぜなら、再送要求方式では本来信頼性をあげることを目的としたACKによる次のブロックを送ってもよいこと

を相手側に知らせることができるからである。

以上、誤り訂正方式について述べたブロッキングの問題は、誤り制御を行わず、ただ情報を送るだけのシステムにおいても、まったく同じことが言えることに注意する必要がある。

2.5.2 会話形システム

従来のデータ伝送は主として、オフラインのデータ伝送であった。これに対して、会話形システムの場合は情報の伝送方向が頻繁に反転する。このために従来のデータ伝送方式では好ましくないことが生じる。例としてISO (International Organization for Standardization: 国際標準化機構) のベーシックモード・システムによって、会話形通信を行なう場合の問題点を示す。

ベーシックモード・システムは半二重通信方式で、次の5つのフェイズより成る。

- (i) 回線の接続 (Connection over General Switch Network)
- (ii) データリンクの確立 (Establishment of Data Link)
- (iii) 情報の転送 (Information Transfer)
- (iv) 通信の終結 (Termination)
- (v) 回線の切断 (Clearing of General Switch Network)

(ii)はポーリングとかセレクトィングでリンクの確立を行なうフェイズである。

図2.1はその一例を示す。

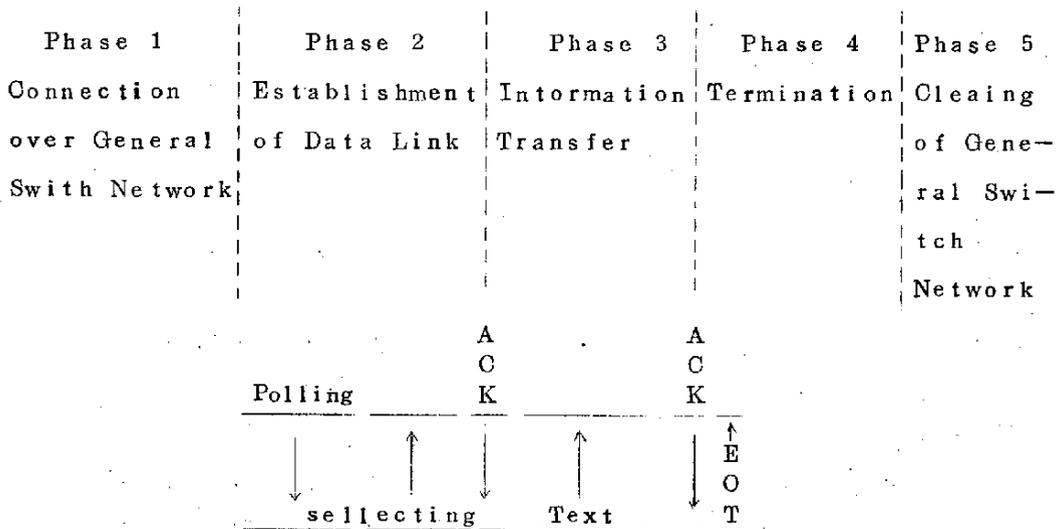


図2.1 ISOのベーシックモードシステム

このように、ISO ベーシックモード・システムでは、フェイズによって、物理的に回線がつながったあと、情報の転送方向が反転する毎に、フェイズ(III)と(IV)を通ることになるので会話モードのようにこの反転が頻繁に行うシステム処理の増加と時間の損失をきたす。

さらに、送受信の切り変わるとにポーリングを行なうのでは情報の伝送が行なわれたあと次の情報を送るとしてもポーリングにかからない可能性がある。一般のデータ伝送ならポーリングにかかるまで待てばすむが会話形通信のばあいのように頻繁に情報の伝送を行なう場合はポーリングにかからず相手側を待たせる可能性があるということは大きな欠点である。

このような理由で、会話に適した方法の確立が強く要望されており、ISOでも検討中である。以下その方法を示す。

まずポーリングの問題を解決する方向としては二つの行き方が考えられる。その一つは、計算機側よりみて会話を行なっている端末に対してはフェイズ(IV)で通信の終結を行なっても、次に必ずその端末をポーリングする。すなわちポーリングのやり方を工夫する方法である。他の方法は次に示すように、処理の増加と時間の損失という問題を同時に解決する方法である。

ベーシックモード・システムにおけるこのような問題の対策としては、フェイズ(II)と(IV)はそれぞれ会話の始めと終りのみで実行することにし、会話の途中では(III)の情報の送受信のみを行なう方法が基本になる。半二重通信方式でこれを行なう場合、主局と従局(情報を送る方を主局と呼び、受ける方を従局と呼ぶ)の切替え方が問題となる。

その方法は次のように分類される。

(1) テキスト毎に主局と従局を切替える方法

(1-1) テキストに対してACKの応答を行ない、それによって主従を切替える。

(1-2) ACKの応答は行なわず、すぐに主従を切替える。

(2) 必ずしもテキスト毎に主従を切替えなくてもよい方法

(2-1) テキスト内に主従を切替えるか否かの情報(以下これを連送情報とよぶことにする)を加える。

(2-2) テキストにヘッティングをつけ、そのヘッティングに連送情報を加える。

(2-3) 主従切替え符号を使用する。

(1-1)の方式

従局はテキスト受信後、ACKをかえすことによって、主局となり、元の主局はテキスト送信後、ACKを受けると従局に変わる。但し、NAKを送ったときは主従の関係

は変わらない。図2.2はこれを示す。このように、この方法ではテキスト受信後ACKを送ることによって、一方的に主局に変わる。このために、伝送上不都合なことが生じる。

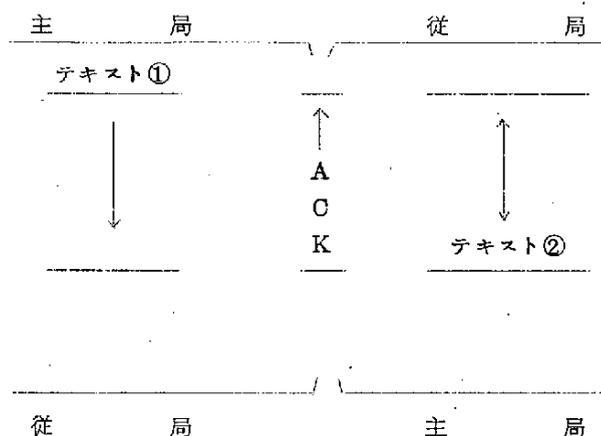


図2.2 主局，従局切換方法（その1）

たとえば，図2.2でACKが端末側に届かなかった場合でも計算機側はテキスト②を送れる。このために，テキスト①に対する応答要求（ENQ）とそれに対する再応答ACKはテキスト②のあとで行なわなければならない。このように処理をあとで行なうことは，それだけ処理手順が面倒になることを意味している。

（1-2）の方式

図2.3はこの方式の一例を示す。図の方式ではブロックに対するACKまたはNAKおよびテキストに対するNAKについてはISOのベーシックモードとまったく同じでテキストに対するACKのみを省略したもので、あまりすっきりした感じがしない。

さらに，テキスト内の文字がETXに誤まった場合，双方が主局となり，通信が不可能になる。

別の方式としては，情報，応答にかかわらず伝送するものはすべてテキスト（またはブロック）の形をとる方法も考えられる。たとえば，NAKの応答も単独で使用せず，STX NAK ETX（BOC）の形にする。

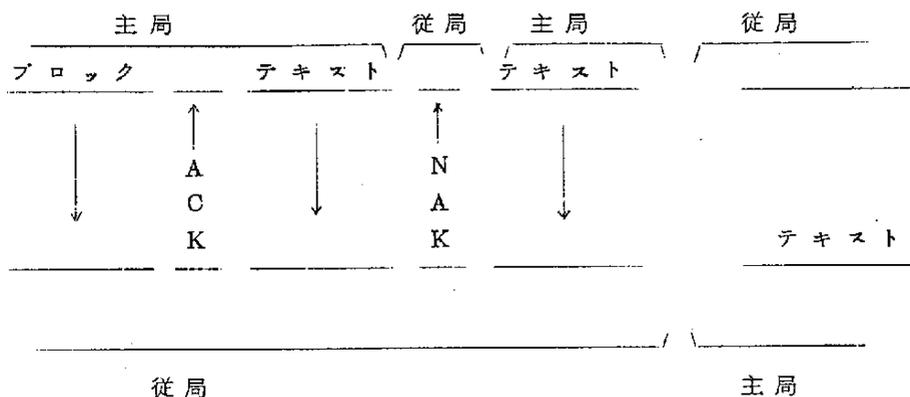


図2.3 主局，従局切替方法（その2）

(1-1)と(1-2)の方法はテキスト送信後必ず主局と従局が切替わるので交互に送信するシステムでは問題ないが，そうでないシステムには使用できない。(これについては4節で詳しく述べる)

(2-1) の方法

この方法はWRITE命令(計算機より情報を送る命令をWRITE命令とよび，逆に端末より情報を送る命令をREAD命令とよぶ)によって，計算機側が情報を送るまえに必ず次の命令がWRITEであるかREADであるかを調べる。それがREADである場合は今送ろうとしている情報をテキストの形で送り，送信権を端末側に渡す(1-1または1-2等の方法をとる)。これに反して，つづく命令がWRITEの場合はテキスト中の所定の位置(たとえば，ETXの直前と約束する)に主局と従局の関係を保持することを示す連送情報(γとする)を追加する。受信側から見ればテキストを受けとって，その中に連送情報があれば従局のままになっている。しかし，この方法は次の難点をもつ。

- (i) テキスト内に連送情報という本来送るべき情報とは別のものをつけることは好ましくない。
- (ii) γ符号は他の用途に使用できない。

(2-2) の方法

この方式のテキストは次の形式をとる。

SOH(ヘッディング)STX テキスト ETX (BCC)

このヘッディングの中に(2-1)で述べた連送情報を入れる。これによって(2-1)

の(イ), (ロ)の問題点は解消する。

以上述べたことのほかに(2-1), (2-2)には次のような重大な欠点がある。すなわち, WR I T E命令で, 情報を送信する場合, 次の命令を調べるまでそれを実行することができない。このために次の命令が決まるまで時間がかかるような場合(たとえば一つの仕事が終わって, 次の仕事が始まるまでの時間がかかる場合がある。)WR I T E命令があっても, 今, 送ろうとしている情報は送れない。

この問題は人間と計算機の通信のあり方に関する非常に重要な問題を提起しているので4節で詳しく述べる。

(2-3) の方法

テキスト送信後, 主局と従局の関係を切換える場合もあれば, 切換えずに続いてテキストを送る場合もあるシステムにとっては好ましい方法である。

2.5.3 割込み機能

ここでいう割込みとは計算機のハードウェア上のもではなく, 端末より計算機へあるいは逆に, 計算機よりターミナルへのQ U I T機能をさす。すなわち, 所定の会話を行なっている最中に, 端末のユーザまたは計算機がその変更を行なう場合に相手側にそのことを知らせる。たとえば, 端末のユーザが受信中, 計算機の送信を途中で中止させたい場合, そのことを計算機に知らせるために割込みを行なう。

伝送上この割込みを可能にするためには, 伝送上の主局従局関係およびI/O命令の状態などにかかわらず, いつでも相手側へ何らかの情報を送れる必要がある。以下その方法を示す。

全二重通信方式の場合は情報の受信中でも送信が可能であるので, いつでも割込みの意志を相手側に伝えることは可能である。しかし, 半二重通信方式の場合は受信中のみならず, 受信中でなくても前節に示した従局は情報を送ることはできないので工夫をこらす必要がある。次に, それを可能にする方法を示す。

受信中でない場合は比較的簡単で, 情報は送れなくても, 特定の割込み符号(特定の伝送制御符号で β とする)だけは送れるようにしておく。たとえば, 端末は従局でキーロックされていても割込み釦はいつでも押せるようにしておくわけである。

次に, 受信中の場合は前に述べたようにテキストが長い場合, それをいくつかのブロックに分けておくいわゆるブロッキングを利用する。たとえば, 判定帰還方式では次のようになる。受信側がそのブロックの受信中, 割込みを行ないたい時は, そのブロック受信後, A C

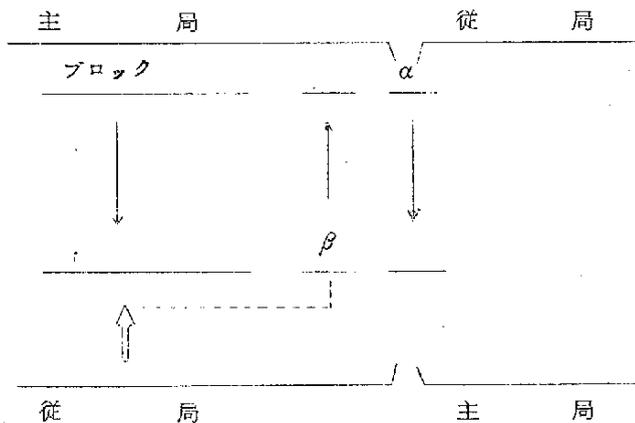
Kを返す時点でACKのかわりに β を送る。主局はこれを受けるとまだ送信すべきデータが残っていても送信権を相手に渡すことによって、割込みに応じる。 β を送った時、すぐに主局に変わる方法も考えられるが、そうすると、第2節で述べたように、テキスト受信後、ACKを送ることによって、一方的に主局となることによって生じた問題が起こる。

このために、 β を送った後、相手側より、“割込み了解”という何らかの情報をもらう必要がある。この何らかの情報としてはテキストの形をとってもよいし、先の主従切替符号（これを α とする）を使うシステムの場合はまさにこの α がうってつけである。

具体的には、端末のユーザは情報受信時、割込みが必要となれば、割込み鈕を押下する。すると端末のハードウェアはそのブロックまたはテキストを受け終わったところで、ACKのかわりに β を送る。計算機はこれを受けるとまだ送信すべき情報が残っていても α を送り、端末側に送信権を与える。

端末が主局になった時点で、割込みの内容を計算機に送る。このように、割込みの内容をテキストの形で相手に知らせることにすれば、割込みの内容がいくら多くても問題はない。

図3.1は受信中の割込の方法を示す。



(注) は端末の割込み鈕を押下したことを示す。

図 3.1 割込みの方法

以上とは別の方法として、先程の β を送る時に、 β のみでなく、 β のほかに割込みの内容を示す符号をつけて送る方法も考えられる。

以上、伝送上の割込み方法を述べたが、最後に、通信制御システム（通信制御プログラム）

で扱ひ割込みの例を示す。これらはすでに述べたように、割込み実行後、テキストとして以下の内容を示す特定の符号を送ることを意味している。

(1) 中止

READまたはWRITE命令の途中でこれを行なうと実行中の命令を中止する。

(2) 再試行

READまたはWRITE命令の途中でこれを行なうと、それまでに実行したものを放棄して、その命令のはじめの状態にもどる。

(3) 再開

READまたはWRITE命令の途中で端末の割込み釘を押下するとすでに述べたように端末のハードウェアは β 符号を送出する。つづいて、計算機側より、 α を受けて主局になっている。この事象を利用すれば実行中の命令を一時中止して、任意の時点で再開できる（再開するときは所定の符号または単なるSTX ETX）を送る。

この機能は次のような場合に利用できる。たとえば、端末で受信中、印字用紙がなくなった場合、一時受信を中止し、用紙をとりかえてから再開する。

本節において今まで述べたことは会話形システムを念頭において割込みについてのことであるが、その方法は会話形でないシステムにおいても使用できる。ただし、会話形でないシステムの場合の割込みは受信側の意向によって、通信を打ち切る場合のみ使うことも考えられる。もちろん、そのように使う場合でも、今迄に記した方法で実行することは可能である。

なぜなら、割込み後におくるテキスト内に通信の打ち切りであることを示す情報を入れればよいからである。しかも通信を打ち切る場合にのみ割込み機能を使用する場合はわざわざテキストにその情報を入れる必要がない。すなわち、受信側が前記の方法によって、 β を送ると（ β を送るところまでは、今まで述べた方法とまったく同じ）

送信側は α のかわりに、EOT (End of Transmission) 符号を送って、第2節に記した通信の終結に入る方が能率的である。これを会話モードの割込みと区別して中断と呼ぶことにする。図3.2はこの方法を示す。

以上誤り制御方式としては判定帰還方式を例にとったが、これらの概念は他の誤り制御方式についても同様であるので省略する。

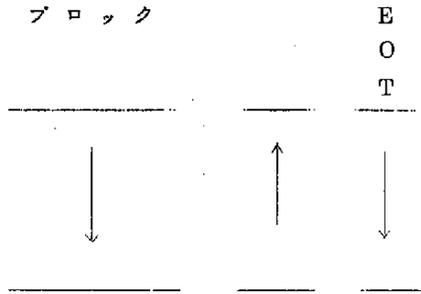


図 3.2 中断の方法

2.5.4 計算機による端末の管理

人間と計算機の会話形システムにおいて、計算機側に要求される条件は種々あるが、基本的な考え方は次の(1)と(2)である。

- (1) システムは端末のユーザに最大限のサービスを提供すること。
- (2) いつでも、ユーザの意向に応じられること。

(2) は先に述べた割込み機能をさしている。(1) については種々のことが考えられる。一つの考え方は次のとおりである。

WRITE 命令と READ 命令の発生する順序および時間がどのようであろうとも、システムは端末のユーザの手をわずらわせず、且つ、わかり易い形でそれを実行できなければならない。

ユーザが送信する必要のない場合は送信できないようにしておき（オフラインで前もって情報を作っておくことは可能であるが、その必要がない場合はキーロックする）、READ 命令で送信すべき場合のみユーザに自然な形で知らせ、送信可能とする（キーロックシステムではキーロックを解除する）

この“自然な形”を実現するためにシステムは上のことを伝送上の事象として、端末を制御するのがよい。すなわち、WRITE 命令で計算機より一つの情報を送ったあととも計算機側が主局で送信権を維持し、READ 命令が出たときにはじめて、端末に送信権を与えることを意味している。逆に READ 命令によって、端末が情報を送り終ったあとは送信権は計算機側に渡る。このように計算機が端末の送受信モードをコントロールするので本方式をモード・コントロール方式とよぶ。

次にこのモード・コントロール方式を第 2 節で述べた会話形システムによって、実現させ

る方法を示す。

(1) 方式 1

READ命令によって、端末より情報を送信する場合は従来通り、テキストとして送信するが、WRITE命令によって、計算機より送信する場合は情報をテキストではなくブロックの形で送信し、送信データがなくなればこのWRITE命令の終りとする。このときETXは送っていないので、伝送上はまだ計算機側が主局のままである。この状態で、次の命令がWRITEのときは上と同様にすればよく、READのときはSTX ETXという空のテキストを送り、従局に変わる。

しかし、この方式では、受信テープをあらためて、同じシステムまたは別のシステムに送信する場合次の欠点がある。命令の順序がWRITE1, WRITE2, READの場合、それを受信したテープのうち、WRITE1で受信した情報のみを送信しようとしても、そのデータはテープ情報にはETBで終わっているので、空のテキスト STX ETXを打鍵しなければならない。

(2) 方式 2

(1)の方式の欠点をさけるため、この方式では1つのWRITE命令によって計算機側が情報を送信する場合も1つのテキストの形をとる。但し、テキストには必ず主局、従局の関係を変えないことを示す連送情報(r とする)を追加する。その後、READ命令において、はじめて計算機側は r のない空のテキストを送ることによって、従局となる。

尚、命令の順序がWRITE1, WRITE2, READの場合、ターミナルの受信テープは

STX M1 r ETX STX M2 r ETX STX ETX

となっている。(M1, M2 はそれぞれWRITE1, WRITE2によって、計算機側より受信した情報を示す) このために、M1のみを送信する場合、人間が受信テープを修正せずに、しかも、先の方法のようにいちいち、STX ETXの空のテキストをつけずに送るためには端末のハードウェアは受信時に自動的に r をパンチしないようにする必要がある。このようにすれば前の方式の欠点はなくなる。但し、この方式には次の欠点がある。

(i) WRITE毎に r をつける必要がある。

(ii) 端末のハードウェアはテキスト内の r をパンチしないように設計しなければならない。

(iii) テープとかカードなどには余分な空のテキスト STX ETXが入りこんでいる。

(3) 方式 3

第2章で述べた主局，従局切換え符号を使用すれば前記の問題点は完全に解消できる。

最後に，このモード・コントロール方式の利点を示す。

- (i) READ命令，WRITE命令の発生する順序がどうであろうとも，その制御は自由自在である。
- (ii) READ命令が出た場合，端末の送信可ランプが点灯するようしておけば，ユーザは安心して，情報の送信を行なえる。
- (iii) 命令の種類と順番がまえもってわかっている場合，端末のテープ装置またはカード装置などに送信データをセットしておけば端末が自動的にシステムと通信を行なうようにすることが可能である。このようにすればユーザは端末のそばにいていちいち鍵盤とリータを切り換える面倒がいらない。このように端末のテープ装置とかカード装置などの機能を十二分に活用すればオンライン・タイムを短縮できる。

これをモードコントロール以外の方式で実行するのに非常に困難である。なぜなら，たとえば（テキストによって主局従局が切換わる方式で），WRITE命令によって，計算機が送信する場合，このテキストを送り終ると，端末側が送信モードとなり，テープ送信を開始する。ところがこの時，計算機側にREAD命令が出ていないことがある。

たとえば命令の順序がWRITE 1，WRITE 2，READの場合，最後のREAD命令用にテープリータに送信情報とセットしておいただけではデータはWRITE 1によるテキストを受け終ったところで自動的に送信されてしまう。このために，WRITEとWRITEの間に常に空のテキストSTX ETXを一々挿入しなければならず面倒である。これに対しモード・コントロール方式によればREAD命令が出たときにはじめて，端末が主局となるので非常にスムーズに自動通信が実行できる。

- (iv) 計算機が端末送受信状態をコントロールするので，READ命令の場合（WRITE命令は問題ない）回線のバッファリングが無理なく容易に実行できる。
- (v) READ命令でユーザが送信中，計算機側より，このREAD命令をキャンセルしたい場合，計算機側より，割込みを行なうことによってユーザの送信を中止させることができる。これも，モード・コントロール方式の重要な機能である。

2.5.5 オンライン端末のあり方

オンライン端末に要求される条件としては次のようなことが考えられる。

- (1) 必要な機能を備えている。

必要な機能はシステムによって異なる場合があるので使用条件に適した機能を備えている必要がある。特定の業務にのみ使用する場合と、汎用の場合で必要とする機能が異なる。2～4節で述べたことも端末についてみると本項に当る。

(2) 操作がやさしい。

一般にユーザは端末の機械とかデータ通信の専門家でないので、端末の操作はできるだけ単純である必要がある。これは操作ミスを防ぐことにも効果がある。

(3) 信頼性が高い。

第1節でも記したように、信頼性を考える場合、回線、計算機、端末などシステム全体としてのバランスに注意する必要がある。

通信上の誤りに対しては、第1節に記したように種々な方法で信頼性を向上させることはそれ程困難ではない。計算機についても多重システムにするとか誤り検出用のロジックを使用するなど種々の工夫が行なわれている。このために端末の信頼性が問題になる。

特に機械部分の信頼性が問題で、信頼性の高い入出力機器が必要である。

(4) 価格が安い

特にターミナルの数が多い場合は切実である。

(5) 融通性がある

これはシステムにもよるが、使用上機能上、システムが変わる可能性がある場合はあらかじめそのことを考慮しておく必要がある。場合によっては同じ端末を2つ以上のシステムに切換えて使用するという場合も考えられる。種々な入出力機器を自由につなぎかえたい場合もある。

(6) その他

以上のほか、騒音が出ない、小型である、保守がしやすいなどの条件も重要である。これらは相反する場合も多い。たとえば、機能を強化すれば価格が高くなるのが普通である。このために、重点のおき方、逆にいえばバランスのとり方が重要になる。

次に、以上の原則にしたがって、実際に端末の選定ないしは設計する場合の問題点を示す。

(a) 端末の種類

使用方よりみると、座席予約装置のような特定の用途にのみ使用される専用型(特殊型)と、タイプライタ型のように、種々の用途に使用される汎用型に分けることができる。

構成法よりみると、タイプライタと紙テープのみなどのように使用する入出力機器が

固定されたもの（I/O機器固定形とよぶことにする）と、1台の端末に、次の項に示すような種々な入出力機器を自由に選択できる形式（たとえば必要に応じてカードリーダーをつけたりはずしたりできるI/O機器可変形）のものに分けることができる。I/O機器可変形の場合は使用条件が変わってもそのときどきに必要なものに変えることができるので便利である。

さらに、入出力機器が複数個の場合、送受信時に、ユーザが入出力機器を選ぶことのほか、計算機が入出力機器を選択できるように伝送制御方式を工夫することも可能である。

たとえばTSSで会話形通信を行なっている場合、TYPE IN COMMANDといった情報はプリンタ（またはディスプレイなど）に表示し、計算結果などの長いデータはラインプリンタとか紙テープ装置などの出力機器に送ると便利である。

さて専用形か汎用形か、I/O機器固定形か、I/O機器可変形かいずれがよいかはシステムの使用法に依存する。特定の予約装置にのみ使用する場合などはとり扱い易さと業務に適した表示方法などに重点をおいて、専用形が多い。汎用形の入出力機器の選定は次に述べるが、I/O機器固定形かI/O機器可変形かについては、汎用機で入出力機器が多い場合は、機器可変形の構成の方が好ましい。

(b) 入出力機器としては何がよいか

本節の始めに示したように使用条件に適した機能は主として使用する入出力機器に依存する。

前項で示したように、専用形の場合はその使用法から入出力機器はほぼ決まるが、汎用形の場合は入出力機器の選定にまよる。汎用形オンライン端末の入出力機器としては表5.1に示すものが使用されている。

紙 入 出 力 機 器	タイプライタ
テープ入出力機器	紙テープ装置, 磁気テープ装置
カード入出力機器	カード入出力装置
紙 出 力 機 器	ラインプリンタ(文字), XYプロッタ(図形)
ディスプレイ出力機器	文字ディスプレイ(文字), グラフィック・ディスプレイ(図形)
音声入出力機器	電話機

表 5.1 端末用入出力機器の例

この内、もっとも多く用いられているのは、タイプライタと紙テープ装置よりなるものである。紙テープ装置の必要性については、会話形式でのみ使用する場合は、絶対必要という程ではない。しかし、受信したデータを後で、同じシステムまたは別のシステムに送信する場合とか、前もって作っておいた多量のデータを送信する場合などにおいては価値がある。

文字ディスプレイ端末も最近、広く用いられつつある。タイプライタに比べて一般に文字ディスプレイは次の利点をもつ。

- (i) 高速である。
- (ii) 編集機能がすぐれている。
- (iii) 騒音を発しない。
- (iv) 信頼性が高い。

しかし次の欠点もある。

- (i) ハードコピーがとりにくい
- (ii) 高価である

高速表示 (10^3 字/秒位) が可能であるので 10^4 ビット/秒程度の回線を使用できれば受信時の文字ディスプレイの高速性は非常に魅力である。

編集機能としては、送信しようとした字を削除したり、変更することはディスプレイにとって極めて簡単である。さらに機器によっては、行毎に修正したり、挿入したりすることも可能である。このために、ディスプレイは会話形システムに適していると言われている。

タイプライタの騒音に比べて、ディスプレイの静かさも大いに価値がある。特に、一般の事務室で端末を使用する場合はいちじるしい。

更に一般的に言えば、タイプライタは機械的であるのに比べて、ディスプレイは電子的であるので信頼性が高い。

一方、ディスプレイはハードコピーがとりにくいことと高価であるという欠点がある。オンライン端末として、予想される第3勢力としては押ボタンダイヤル電話機があげられる。これは電話機であるので、相手側の接続は相手のもので、それに加えて他の端末に比べて極めて安価であるので、簡易形端末としては非常に有望である。簡単な問合わせ用としては、うってつけである。

以上のほか、I/O機器可変形端末の入出力機器としては、磁気テープ装置、カード

装置、ラインプリンタなどの高速入出力機器がある。(次のC項参照)

次に図形をとり扱うシステムではグラフィックディスプレイが多く使用されている。ただ現在のものは高価であるのでXYブロックとか簡単なA-D変換機をもつ機械の使用も考えられる。

(c) 必要とする伝送速度

ユーザが打鍵しながら、情報を送る場合は、人間の打鍵速度はおそいので10字/秒くらいで十分である。

それ以外は、ユーザにとって速度は速い程よい。しかし、表5.2に示すように、入出力機器の速度には制限がある。制限内においても、速くなる程高価になる。さらに速度としては、回線の伝送速度に注意する必要がある。

1. キーボード	: 13字/秒
2. タイプライタプリンタ	: 20字/秒
3. 文字ディスプレイ	: $2 \sim 30 \times 10^3$ 字/秒
4. 紙テープリーダー	: 10^3 字/秒
5. カードパンチ	: 250枚/分 $\approx 3 \times 10^2$ 字/秒
6. カードリーダー	: 1000~1500行/分 $\approx 13 \sim 2 \times 10^3$ 字/秒
7. ラインプリンタ	: 1000~1500行/分 $\approx 2 \sim 3 \times 10^3$ 字/秒
8. 磁気テープおよび 磁気ディスク	: $150 \sim 200 \times 10^3$ 字/秒

表5.2 入出力機器の速度

表5.3は現在の電電公社回線の伝送速度を示す。

公社では 48×10^3 ビット/秒などさらに高速なデータ伝送方式の実用化をすすめている。

専用回線	市内専用回線	： 2400ビット/秒 以下	
		市外専用回線	{ 第1規格：50ビット/秒以下 第2規格：200ビット/秒以下 第3規格第1種：1200ビット/秒以下 第3規格第2種：2400ビット/秒以下
交換網	加入電信	： 50ビット/秒以下	
	準専用	： 200ビット/秒以下	

表 5.3 回線の種類とその伝送速度

シリアルプリンタとしては従来10字/秒以下のものが使用されてきたが、最近20字/秒のものが開発された。

(d) 符号

符号は情報を表現する言語であり、使用者間で合理的に統一されなければならない。

		SHIFT IN 側										SHIFT OUT 側						
		b ₇	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
		b ₆	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
		b ₅	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	R	C	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	0	0	0	0	0		Null	DLE	Space	0	.	P	@	p	空白	間隔	ク	ミ					
0	0	0	0	1	0	0	1		SOH	DC ₁	!	1	A	Q	a	q			。	ア	テ	ム			
0	0	1	0	0	0	0	2		STX	DC ₂	"	2	B	R	b	r			「	イ	ツ	メ			
0	0	1	1	0	0	0	3		ETX	DC ₃	#	3	C	S	c	s			」	ウ	テ	モ			
0	1	0	0	0	0	0	4		EOT	Stop	CS ₁	4	D	T	d	t			，	エ	ト	ヤ			
0	1	0	1	0	0	0	5		Enq	Nack	%	5	E	U	e	u			。	オ	ナ	エ			
0	1	1	0	0	0	0	6		Ack	Sync	&	6	F	V	f	v			ヲ	カ	ニ	ヨ			
0	1	1	1	0	0	0	7		Bell	ETB	'	7	G	W	g	w			ア	キ	ホ	ロ			
1	0	0	0	0	0	0	8		BS	Cncl	(8	H	X	h	x			イ	ク	ネ	リ			
1	0	0	1	0	0	0	9		HT	EM)	9	I	Y	i	y			ウ	ケ	ス	ル			
1	0	1	0	0	0	0	10		LF	SS	*		J	Z	j	z			エ	コ	ハ	レ			
1	0	1	1	0	0	0	11		VT	Esc	+		K	(k	{	③			オ	サ	シ	ロ		
1	1	0	0	0	0	0	12		FF	FS	,	<	L	(CS ₂)	l	←	③			オ	ヤ	シ	フ		
1	1	0	1	0	0	0	13		CR	GS	-	=	M)	m	}	③			ユ	ス	ヘ	ソ		
1	1	1	0	0	0	0	14		SO	RS	.	>	N	^	n		③	SO		ヨ	セ	ホ	ッ		
1	1	1	1	0	0	0	15		SI	US	/	?	O	-	o	Del	SI		ッ	ソ	マ	ッ			抹消

表 5.4 情報処理交換用標準コード

表 5.4 は JIS の情報処理交換用標準符号 (通称 ISO/CCITT 7 ビットカナコード) である。これは機械装置設計使用の立場を代表する ISO (International Organization for Standardization: 国際標準化機構) と、通信伝送の立場を代表する CCITT (Comité Consultatif International Télégraphique et Téléphonique: 国際電信電話諮問委員会) の合作の ISO/CCITT 符号の shift out 側にカナ文字を加えたものである。

ISO/CCITTの7ビット符号は情報は7ビットで、文字の総数は $2^7 = 128$ まで可能である。それは、制御用文字、英文字、数字および特殊記号よりなり、それぞれグループをなしており、グループ内の文字の配別は順序をつけている。0と1列は伝送制御コードである。

(c) 鍵盤配列

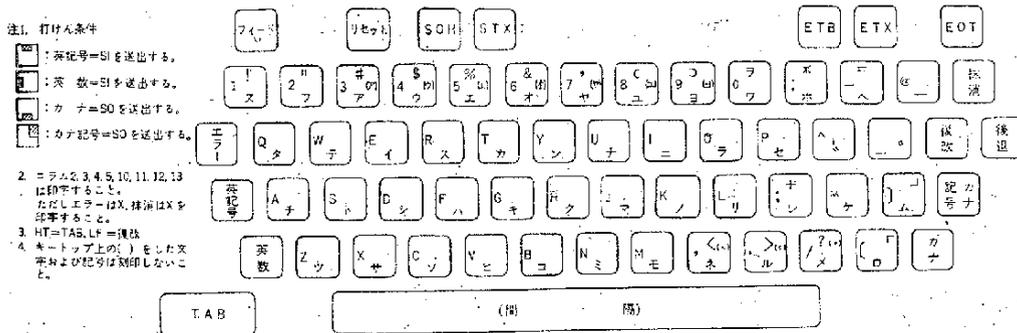


図 5.1 鍵盤配列の例

図 5.1は前項で示した情報処理交換用標準符号の代表的な鍵盤配列を示す。この特色は英小文字が入っていないことである。英小文字まで入れるためにはキーの数を増すか、または「英記号」「英数」「カナ」および「カナ記号」のほかに「英小文字」の切換えを設ける必要がある。しかし、使いやすい鍵盤は文字数に制限があり、文字数を増せば機構が複雑になる。そこで、システムによってはカナ文字より英小文字のほうがより有用な場合もあり、その場合は「英数」と「英小文字」中心の鍵盤を使用することになる。このようなカナを使用しなければ先に述べたSIとSOを使用しなくてもよいので扱い方と制御が簡単になる。(通信制御装置によってはSIとSOの制御ができないものがあり、その場合はソフトウェアで制御することになる。)

このソフトの切換えによって、同一キーを複数個の文字に使用する場合にも問題がある。たとえば、図5.1の鍵盤配列ではSHIFT IN側の2列と3列で対応する行の文字は同一キーになっている。たとえば3と井は符号の5ビット目が異なるだけであり、英数と英記号における変換機構は容易である。ところが、SHIFT OUT側の10列のカナ記号とカナ文字の間にはこのような統一性がない。すなわち前もって「カナ」キーと「カナ記号」キーを押すことによって、カナ記号とカナ文字を同一のキーによって切換えるには機構もしくは回路が多少複雑になる。

(f) 通信方式

半二重通信方式では送信と受信を切り替えて伝送を行なうが、全二重通信方式では送信と受信を同時に行なうことができる。全二重通信方式が必要になるのは次の場合である。

- (i) 双方より同時に情報を送る必要があるシステム
- (ii) 誤り制御を返送方式で行なう場合は受信しながらそれを送信できる方が好ましい。特に会話モードでは必須である。

一方、全二重通信方式は次の欠点をもつ。

- (i) 既製の通信制御装置によっては、半二重通信方式の場合に比べて、この機構が2倍必要となり、価格が高くなる。
- (ii) 誤り制御方式とか通信制御装置によっては、制御が複雑となり、端末およびソフトウェア・システムが複雑になる。

まとめ

以上オンラインデータ通信について述べたが最後に、端末に関する問題点のいくつかを記して本章のまとめとする。

一般的に言えば、安価で機能の優れているものが要求されるわけであるが、これは矛盾している場合が多い。さらにオンライン・端末の場合種々なシステムがあり、要求される条件も種々多様である。このためにオンライン・端末のあり方を一層困難なものにしている。この意味で、これまで記したことはすべて重要な問題を提起しているわけであるが、特に「標準化」、「入出力機器のあり方」、および「高速化」が重要と思われる。

標準化には符号の標準化、伝送方式の標準化、入出力機器の標準化などがある。符号の標準化についてはすでに述べたように、ISO/CCITT 7ビットコードがJIS化された。

伝送制御方式の標準化も非常に重要である。第2節で述べたように、普通のデータ伝送についてはISOのページック・モードが確立されているので、今後は第3節に記したように

会話モードのあり方が重要である。ISOでも現在そのあり方の検討を行っている。このような氏による利益については種々考えられるが価格を下げる一つの方法であることも見がせない。

価格の問題に関しては入出力機器が大きな問題を占めている。さらに入出力機器のあり方については価格のみならず使用上の機能、使い易さなど多くの重要な問題を含んでいる。前にも記したように、特別なシステムのみ使用される場合は比較的問題は少ないが汎用システムの場合は多くの要素を考慮しなければならない。結局、使用条件によって少なくとも当分は何種類かの機種が存続するものと考えられる。

一つの推測では、次のような機種に分けられる。

- (1) タイプライタ・ベースのもの
- (2) ディスプレイ・ベースのもの
- (3) 押ボタンダイヤル電話機
- (4) 高速I/O機器可変形
- (5) 計算機形

(1)のタイプライタ・ベースのものは今後とも広く使用されるものと思われるが、タイプライタの受信速度が問題である。

(2)のディスプレイは高速で静かであるので会話形システムとか特別な用途に広く用いられるであろう。ただし、前にも記したが、ハードコピーと、価格が高いことに問題がある。

(3)の押ボタンダイヤル電話機は価格が安くて、簡便であるので、簡単な問合わせシステムなどに使われる可能性が高い。

(4)はラインプリンタ・カードなどの高速I/O機器を自由に選定接続できるもので高級形である。このタイプが普及するためにはこれらの高速性に見合った回線が使えるようになる必要である。

(5)の計算機形というのは前項の高級形をさらに高級にしたものに当たるわけであるが、それとは別に、ユーザの手持ちの計算機を中央の計算機に対する端末兼衛星計算機として使用することに多くの意義をもつ。この場合も(4)の場合と同様、高速回線が望まれる。

2.6 データ通信網

前節までに遠隔情報処理技術のうち、電子計算機と端末装置について述べたが、ここではこれらの設備や回線、交換設備などにより構成されるデータ通信網について網構成を主に解説する。個々の設備については概念的に扱うだけに留めたい。

まずデータ通信網の現状を代表的な適用業務を例にとりて説明し、適用業務の性格に応じて

データ通信網の構成に現われる特徴を概観する。

このような適用業務毎のデータ通信網は、企業内や企業間における情報交流の迅速化、緊密化の要請をうけて、相互に連結し、複合化し、共用化の方向に進むものとみられる。ここでは次に、これからのデータ通信網として共同利用されるデータ通信網に焦点を絞り、その技術的問題点、解決の方策などについて2、3の提案を紹介し、解説を行なうことにする。

2.6.1 データ通信網の現状

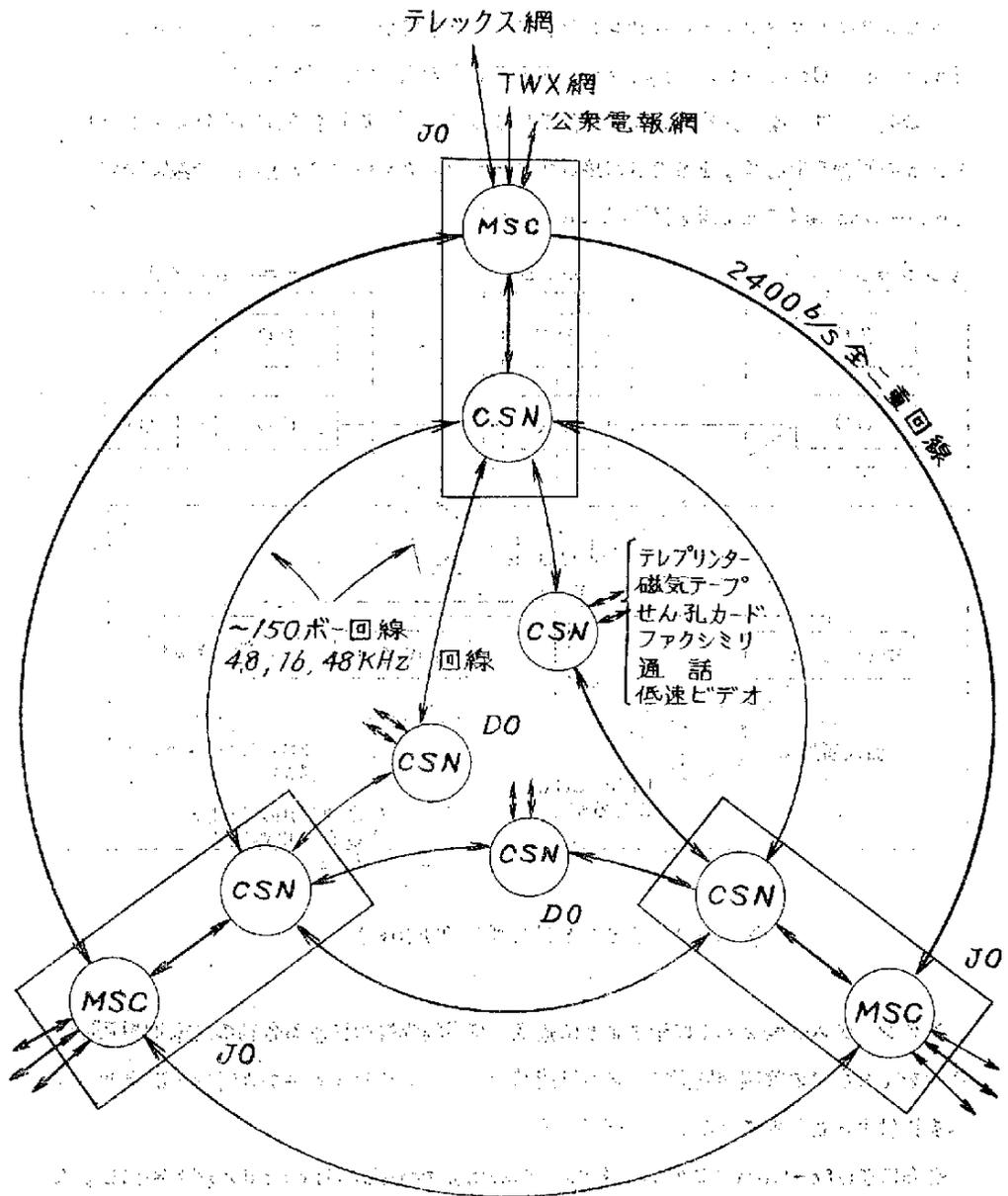
本節では、データ通信網の現状について、いくつかの適用業務を例にとりて概観する。適用業務にはデータの流れ、処理のしかたに着目し、その代表的な組み合わせとしてメッセージ交換、運用管理(座席予約、銀行預金等)、計算サービス、情報提供サービスの4種をとりあげる。これらは、データの伝達を主とするもの、処理内容が相互に関連する複数の端末のデータ処理を主とするもの、関連のない複数端末のデータの処理を主とするもの、ファイルからの情報の取り出しを主とするものに、それぞれほぼ対応している。ここでは、人間-計算機間で通信を行なうシステムを取り上げ、遠隔計測とか自動制御など、対機械のシステムは対象としていないが、網構成としては前者に類似する場合が多い。

(1) メッセージ交換

企業間にまたがるメッセージ交換システムに、全国地方銀行協会加盟の62行相互を結ぶ地銀協為替交換システムがある。⁽¹⁾ 本システムは加盟銀行内の一店舗に紙テープ入出力の端末装置等を置き、これと中央に設置された蓄積交換機能[※]をもつ汎用電子計算機を200 b/s の回線で直結したメッセージ交換システムである。このシステムには各銀行毎の本支店をつなぐ回線網(これを自行網という)が紙テープ中継により接続されこのようにして自行網を含め、実質的に全国4000店余りを結ぶ為替交換システムを構成している。

図2.6.1のシステム系統図に示すように、自行網には専用電信回線と電信交換装置を組み合わせたもののほか、交換機に電子計算機を使うもの、さらに電話網を利用するもの等、多種にわたり、端末装置の種類も多い。したがってデータ・コードの単位数や符号の組み合わせなどはまちまちで、このため地銀協システムは端末装置にコード変換機能をもた

※ 蓄積交換はふつりメッセージ交換と同義に解釈されているが、ここでは蓄積交換を「受信メッセージを一旦蓄積してから送出する」という機能に着目した表現として捕え、メッセージ交換を字義通りに解釈してこの2つの表現を使いわけてみた。



- MSC 蓄積交換形交換機
- CSN 回線交換形交換機
- JO 接続局3局
- D0 従局24局

図 2.6.2 ARS の中継方式概念図

図 2.6.3 はウエスタン・ユニオンの情報サービス計算機システム (Information Service Computer System, ISCS) の網構成図である。(4)

この通信網は、蓄積交換機能をもつ計算機 ISCS と、これら相互間を結ぶ 2400 b/s の回線を中心に、ISCS に接続されるテレックス網や TWX 網、公衆電報網、Info-Com 端末等の端末網からなる。

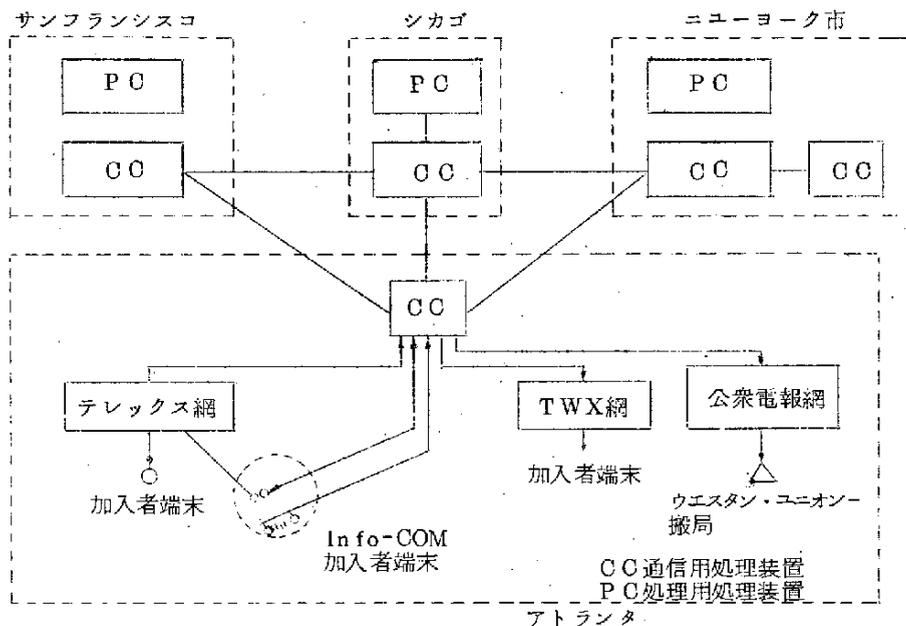


図 2.6.3 ISCS 構成図 (初期計画)

このシステムは表 2.4.1 に示すように速度、符号構成等の異なる各種の端末網相互間の通信を ISCS の蓄積交換機能により可能にするほか、テレックス加入者等に計算サービスを提供することもできる。

さらに Info-Com はこの総合通信システムのなかで顧客毎の Closed な網を実現し、機密の保護やおぼえやすい局名の付与等、専用網としての特色と、経済性やシステム構成の融通性という公衆網の長所を共にいかしている。

せシステム内の使用コードを統一している。

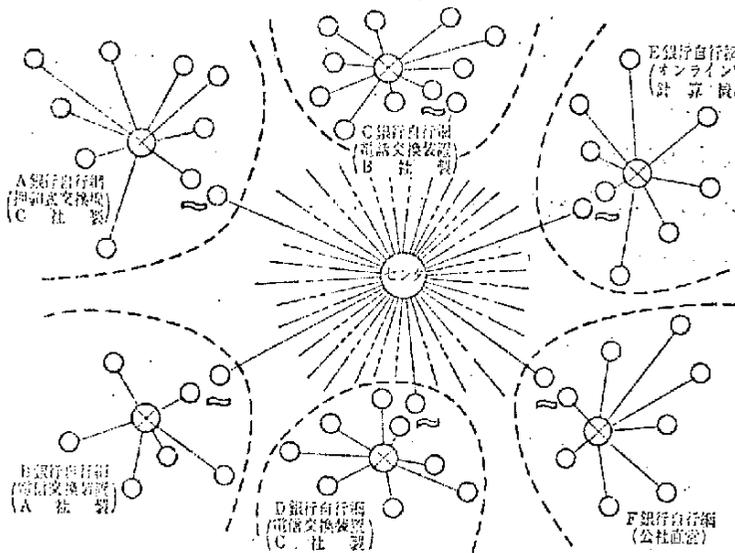


図 2.6.1 地銀協システム系統図

※

上述の自営網では加入電話網を利用するシステム — 厳密にはダイヤル通話のできる接続区域相互間において、特定の加入者相互が電話のほか、符号通信などを切替により行なり制度、いわゆる準専用制度を利用するシステム — が使われているが、これは専用線を使うほど通信量がない所で経済的にデータ伝送を行なり手段の一つである。上述のシステムには使われていないが、加入電信をデータ伝送用に利用して同様な効果をあげている事例もある。⁽²⁾

※ 加入契約に基づき電気通信サービスに専ら使われる通信網はふつう公衆網ともいわれているが、公衆という言葉が公衆電気通信法にいう公衆電気通信とか、公衆電話、公衆電報の公衆とまぎらわしいので、あえて加入網、加入電話網、加入電信網などと呼ぶことにした。

米国では政府機関に大規模なメッセージ交換システムが導入されている。そのうちの
一つに米国連邦調達庁のARS (Advanced Record System) がある。⁽³⁾

このシステムは図 2.6.2 に示すように、回線交換網と蓄積交換網を結合したシステムである。

このような網構成をとることによって、端末相互間で任意な形式で即時に電文やデータの伝送ができるほか(回線交換)、コードやフォーマット、端末設備の形式等が違い端末相互間の接続も自由にでき(蓄積交換)、回線交換、蓄積交換の両者の特色を十分に生かしている。また、接続局相互を結ぶとともに、従局から接続局2局に回線を設け、迂回中継機能をもたせて網の信頼性を高めている。

表 2.6.1 I S C S に接続される回線の諸元

回線名	速度 (b / s)	単位数	符 号	同期方式
テレックス	5 0	5	ボ ド	非同期
公衆電報	5 6.7	5	ボ ド	非同期
Info - COM	1 1 0	8	A S C I I	非同期
T W X	1 1 0	8	A S C I I	非同期
割込ポジション	1 1 0	8	A S C I I	非同期
通信ログ印刷装置	3 0 0	8	A S C I I	非同期
CC-CC	2,4 0 0	8	A S C I I	同期

(2) 座席予約と銀行預金

列車や航空機等の座席予約システムは、その性格から広域網を構成する必然性をもち、このため、通信設備の経済化が主要な課題となる。国鉄のシステムでは、通信量の少ない端末に対して1～2段の集信装置を置いている。⁽⁵⁾ 座席予約のシステムではないが、多分岐構成としたり、時分割多重方式を採用する場合もある。⁽⁶⁾⁽⁷⁾

電話網を利用する場合、接続時間が応答時間に較べかなり長いのが、運用面などにおいて問題になるといわれる。これは計算機に問い合わせ、即座に応答を求める形の業務に全般的に言えることで、ダイヤル市外通話で十数秒、電子交換機を使った場合でも5～15秒程度の接続時間は、航空機の座席予約等、ある種のシステムの応答時間の要求条件、数秒以下に比べるとかなり長い。⁽⁸⁾

一方、預金システムは、一つの銀行に着目すると、地域網を連結した網を構成する傾きが強い。これは顧客がほとんど一店舗に固定し、そのような性格をもつ顧客を中心にした一店舗の事務の機械化という性格が強く、計算機の処理能力にみあう処理量を得るために数店舗がまとめられて地域網を構成するからであり、為替交換や、普通預金の他支店での預け入れ、払戻し等の機能をもたせるために、このような性格の地域網が相互に連結されるからであろう。

上述の座席予約、預金システムは共に処理形態が定型化し、単純化しており、符号の種類も一般に少なく、データの流れ方も端末から問い合わせで計算機が応答するという比較的単純な形が主であるため、端末装置を単能化し、経済化と操作性の向上をはかり、データ伝送方式も制御手順や符号構成を簡略にするほか、問い合わせ応答形式の特長を生かした集

信形式を使って経済化をはかっている事例がある。(6)

(3) 計算サービス

科学技術計算や事務計算などのオンライン計算サービスでは、システムの規模は、距離の増大に伴う回線費用の増加と計算機の大型化による計算費用の減少との兼ね合いにより決り、地域網の構成をとることが多い。米国での資料によると、小形の計算機を専有する場合との比較では、回線距離の限界は、専用電話回線を使った場合75マイルないし100マイルのところにあるといわれているが⁽⁹⁾、現実には端末の設置場所は計算センタと同一市内域の場合が大部分のようである。

システム相互間の連結はサービスの性格から本質的ではないが、頻度の少ない高度の計算を、処理能力の大きい別のシステムで集中的に処理するために連結をはかる事例は、今後増えていくだろう。

米国の現状を見ると回線網として加入電話網や専用電話回線とか専用電信回線が使われているが、一般的傾向として加入電話網が使われていくものとみられる。端末装置としては、キーボード・プリンタが主に使用されているが、人間—計算機間の相互作用を緊密化するために、映像表示装置(CRTディスプレイ等)が使われる方向にある。

一方、電電公社で計画中の簡易計算サービスのようになり、電話機—押ボタン・ダイヤル電話機を端末に使い、音声応答と組み合わせることで経済化と普遍化をはかる方法が注目される。

このような計算業務、特に科学技術計算では、従来の一括処理の運用のしかたから、計算結果をみて判断しながら、次の計算に進むという、会話的な使い方が増えていくのは明らかであるが、この場合、判断をしている時間、いわゆる思考時間の通信料がこの種の使い方には大きな影響を及ぼすだろうといわれている。(10)

Project MACの作業分析によると、通信回線を通してデータが伝送された時間は通信時間の3.1%と報告されている。⁽¹¹⁾ この問題のデータ通信網の面からの検討は次節(2)項で触れることにしよう。

いずれにせよ、計算サービスのようになり、サービスの効用がセンター—端末間の距離や通信時間に必ずしも一義的に依存しない場合に、サービス料金をどのように設定するのが妥当かは単なる費用、効用論だけでかたづけ問題ではないのかも知れない。

(4) 情報提供サービス

情報提供サービスとして現在実用に供されているものは、単なる情報案内とでもいうべきものがほとんどで、したがって株価案内のように迅速性を要求するものなどに限られて

いるが、情報の爆発的増大と情報検索技術の進歩に伴い、情報検索と会話的使用を武器とした情報提供サービスの需要が今後飛躍的に増大するのは明らかである。

このような情報提供サービスには、株価案内、信用調査等、特定多数を対象にする傾向のもの、旅行案内、買物案内等、不特定多数を対象とする傾きの強いものの両者があり、したがってその業務の性格に応じて専用網や加入網が使い分けられ、あるいは併用されることになる。

網の地域的特性については、特定の地域社会に密着した情報を対象とする場合を除いて広域網となるが、情報の更新の頻度が少ない場合などには、地域網に分割されることもあるであろう。

端末には、株価案内サービスのよりの特殊端末から、計算サービスと同様キーボード・プリンタさらに押ボタン・ダイヤル電話機と音声応答装置の組合わせなどが広く使われよう。将来は映像表示装置やファクシミリ等を使って、所要の情報の索引だけでなく、情報そのものをソフト・コピー、必要によりハード・コピーで得られるようになる予想されている。このような場合、端末から計算機側への情報はわずかだが、逆方向には大量の情報が流れ、したがって、方向で帯域巾の異なる非平衡が使われ方になるだろう。

2.6.2 これからのデータ通信網

前節では、システムの特異性に着目してシステム内で最適化をはかっている事例に触れた。しかし、座席予約システムであれば、旅行計画全般にわたる予約サービスを提供していくとか、預金システムであれば、貸付業務や為替業務の機械化からさらに進んで、資金の移動や取引の中核として関連業務までも対象に加えていくなど、企業内はもちろんのこと、関連企業も含めてシステムの総合化の方向をたどっていくとともに、また専用のシステムを使用する余裕のない企業や機関では、システムを共用化するようになるだろう。このようにして、データ通信網は必然的に共同利用の方向に進み、このなかで端末の入出力形式、伝送速度、運用方法等に対する広範な利用者の要望を取り入れながら、総合的な効率化、経済化をはかっていくだろう。

ここでは、これからのデータ通信網としてデータ通信網の共同利用に焦点を絞り、このような通信網を実現していくための技術的問題点をいくつか取り上げ、解決のための提案を紹介し、解説を加えることにしたい。

(1) 計算機の集中と分散

複数の計算機が連結されたデータ通信網は、単独のデータ通信網が業務上の要請により

相互に結ばれた結果として生まれることもあるが、また集中と分散の比較において分散方式が採用され、あるいは単一のシステムが処理量の増加やサービス地域の拡大に伴い地域的に分割され、その結果このような網が構成されることもある。

後者の場合に分散方式が選ばれる主要な要因は次の2点である。⁽¹⁰⁾ ⁽¹²⁾

イ) 回線費用の節減

処理要求の大部分をその地域網で処理でき、計算費用の増大を回線費用の節減で補うことができる場合である。

ロ) 危険分散による信頼性の向上

災害等によるシステムの潰滅やサービスの全面停止を避けるためにセンターを分割する場合である。この観点からすると、処理形態は一つの業務を地域的に分けて複数のセンターが分担し、その結果余程多量の業務でなければそれぞれのセンターが複数の業務に共用される形になるだろう。

このようにして分散が行なわれると、さらに、あるセンターの繁忙時に過剰な仕事を別のセンターに処理させると言った負荷の平均化に利用され、また、特殊業務を集中化して処理費用の節減をはかる傾向も生じるであろう。Dennis はこのような傾向を過少評価すべきではないが、網構成の重要な要因とはならないだろうと述べている。⁽¹⁰⁾ 上述のような分散の実現は、データの更新が多いファイル中心のシステムでは、分散したファイルの管理が繁雑になるという難点がある。例えば一つのアイテムを更新するために、いくつかのセンターのファイルのアイテムを更新する必要がある場合に、あるセンターで更新中に別のセンターで問い合わせが起ったり、相反する更新要求が別々のセンターで同時に発生したときである。このような場合、全てのファイル更新の間、そのアイテムへの照会や更新を禁止するような方法をとることが必要であろう。⁽¹⁰⁾

このように繁雑な措置を要するため、共通ファイルへの照会が本質的な業務で、しかもファイルの更新が非常に多いものについては、ファイルを分散すべきではないと言われている。

ファイル中心のこれからのデータ通信網をPhister は次のとおり予想している。⁽¹²⁾ 複数の計算機センター相互間およびセンターと端末間を必要に応じデータ交換機を介して、通信回線で結び、このようにして構成されたデータ通信網において、それぞれのセンターは、その近傍の利用者から要求のあった手軽なサービスや全国を相手とした独自のサービスを提供する。あるセンターに属する利用者からの処理要求は、そのセンターで処理でき

るものはそこで処理し、できないものは直接あるいは他のセンターやデータ交換機を介して適当なセンターへ廻される。

それではこのようなデータ通信網のイメージを満たすには、通信設備の面でどのような構成にしたらよいのであろうか。以下に述べることにしたい。

(2) 加入網の利用—回線交換と蓄積交換

前節に述べたように疎通対地に限られ、しかも通信量の多い場合や軍用システムのように信頼性が高度に要求される場合は、専用線によるのが普通であるが、このような条件がなく、データ伝送にそれほどの高速性を必要としない場合は、一般に加入網によるのが設備を共用でき、経済的である。

表 2.6.2 加入網と専用網の比較

項 目	加 入 網	専 用 網
高速伝送に対する適応性	電話網の場合、音声帯域まで 〔これをこえる高速伝送を要する場 合、別の網を設置しなければなら ない。〕	容 易
伝送品質	右に較べやや劣る	高
接続時間	長い（加入電話網では数～十数秒）	高速化により短縮が 可能
機密の保持	右に較べやや劣る	容 易
特殊機能の付与に対する 融通性（表 2.6.4 参照）	余りない	大
量的変更に対する融通性 （個々のユーザに着目し て）	大	左に較べ劣る
経済性	通信量が少なく通信対地が多い場合 に有利	通信量が特に多い場 合に有利

加入網と専用網の比較を表2.6.2に、データ伝送に使われる加入網の現状を表2.6.3に示す。

2.6.3 加入網を使うデータ伝送サービス

国名	サービス名称	データ伝送方式		使用される加入網 (交換方式)	記 事
		通信速度	通信方式		
日本	加入電信	50b/s	半二重	加入電話網 (回線交換方式)	
	準専用	200b/s以下		電話網(")	
米国 (ATT)	データ・ホーン				
	低速	300b/s以下	全二重	電話網(")	
	中速	1200b/s以下	半二重	"	
	高速	2000b/s以下	半二重	"	
	並列	20字/秒	単向	"	
	アナログ	900Hz以下	-	"	FAXおよび低速 テレビ 心電図伝送
	超高速	100Hz以下	-	"	専用交換網 試行サービス中
	TWX 100	110b/s	半二重	HWADS (")	8単位符号 (ASCII)
	60	45b/s	"	電話網(")	5単位符号
米国 (エニ スオン ウェ ン)	テレックス	50b/s	"	テレックス網(")	
	広帯域交換	2,448KHz		専用の交換網(")	
英国	デーテル 100	100b/s以下	半二重	テレックス網(")	
	200	200b/s以下	全二重	電話網(")	
	300	20字/秒	単向	"	データ収集システム用
	600	600, 1200 b/s以下	半二重	"	
	テレックス	50b/s	"	テレックス用(")	
西 ド イ ッ	テレックス	50b/s	"	テレックス網(")	
	ダテックス	200b/s	全二重	ダテックス網(")	
	電話利用のデ ータ伝送	200b/s以下 600, 1200 b/s以下	" 半二重	電話網(") "	

注 加入電信、テレックス、ダテックスおよびデーテル100は、直流電送方式を使用

表に示すように、既存の加入網は殆んどが回線交換方式を採用しており、その得失は回線交換の可否に大きく依存する。

表2.6.4は回線交換と蓄積交換の比較である。(13)

表2.6.4 回線交換と蓄積交換の比較

項 目		回 線 交 換	蓄 積 交 換
回 線 利 用 度		低能率 送信方向の利用度が低いほか、受信方向は殆んど使われない。また、接続動作の間回線が遊ぶ。*	高能率
特殊通信機能	同報通信	一般に不可	可
	優先度通信	一般に不可	可
	速度変換	不可	可
	符号変換	不可	可
データの信頼性他	データの記録	不適	適
	機密の保護	問題が少ない	問題が多い。(メッセージを一旦交換機が預かる形になるため)
	データの形式チェック	不可	可
融通性 データ内容の	データ形式の変換	不可	可
	編集 データ内容の任意性	大 (ただし両端末が同一機能をもつ必要がある。)	小
会話し式通信	回線による応答遅れ	適	不適(ただし、伝送速度が高速な場合適)
	転送量に比例した課金	やや不適	適
接 続 時 間		大	小(情報は接続と同時に伝送される)

* 既存の回線交換方式では、上下の通信路の伝送容量が同一であり、また接続動作が階梯毎に順次行なわれるため

本表に示すように、蓄積交換方式は回線の使用効率や、特殊通信機能、接続時間等の面で回線交換方式に比べすぐれている。また、前項(3)に述べた思考時間の問題に対して、会話的通信という面で検討の余地は残るにしても、一つの解^{*}になるものと考えられる。というのは、蓄積交換方式では、メッセージが送られるときだけ回線（加入者回線を除く）と交換機が占有されるからである。

このように蓄積交換方式は、情報を一旦蓄積し、整理して再送出する機能によって通信網の効用を高めることができ、新しいデータ通信網の提案のなかに色々と生かされる。

この点については次項以下で述べることにしよう。

一方、回線交換方式は端末相互間のデータ伝送方式やデータの形式にあまり制約を与えないという点で優れ、大量のデータを一括して伝送する場合にも適している。また蓄積交換方式の利点も回線交換網のなかに蓄積交換機能を一部はさむことによって、相当程度まで生かすことができる。さらに既存の通信網を利用できる点経済性の面で特に優れており、利用度の低い加入者線階梯についてはその利点を無視できない。回線交換方式は、将来のデータ通信網の一部として十分な意義をもつものとみられる。

既存の加入網、特に電話網の場合、通話用として設計された通信網をデータ伝送用に使うことによって、様々な問題が派生する。(11)(15)

信号レベルが過大であったり、不適正な周波数成分があると、漏話や通話歪などの妨害を網に与える。

また、網の疎通に影響を及ぼすことがある。通話の平均保留時間は数分、呼数は1時間あたり数呼程度であるのに対し、データの呼は著しくばらついており、きわめて長いのある反面、データ・バンクへの問い合わせの場合など、保留時間が数秒以下のこともあり、呼数も計算機からの自動ダイヤルを認めると非常に高頻度になると予想される。このため電話網が共通制御方式の交換機で構成されている場合、このような性格のデータがふえてくると、交換機の共通制御装置に過負荷を生じ、さらにこれが波及して電話網の異常ふくそりをきたすおそれも十分考えられる。加入網の利用にあたっては、このような点についても慎重に検討する必要がある。

* 回線費用の節減をはかる方法では、多くの端末からのメッセージに対し、回線を共用する方法としてTADI (Time Assignment Data Interpolation) があるがこれより、集線装置とマルチプレクサの組合わせによる方法の方がさらに一層経済的だといわれる⁽¹⁴⁾

(3) 端末機能の一部集中化

オンライン・システムの総費用のうち端末の占める割合は、座席予約業務などの場合、30～40%といわれている。⁽¹⁶⁾これが50%を超えることも珍らしくない。このために端末装置のコスト・ダウンはデータ通信システムの重要な課題の一つである。

端末装置のコスト・ダウンは、装置を標準化し、需要増をはかるなどの正統的方法の他に既に大量生産の効果があがっている卓上計算機や、会計機、家庭用テレビ・セットに遠隔端末としての機能をもたせるなどのアプローチも考えられる。ここでは、端末装置の制御機能の集中による経済化について述べることにしよう。

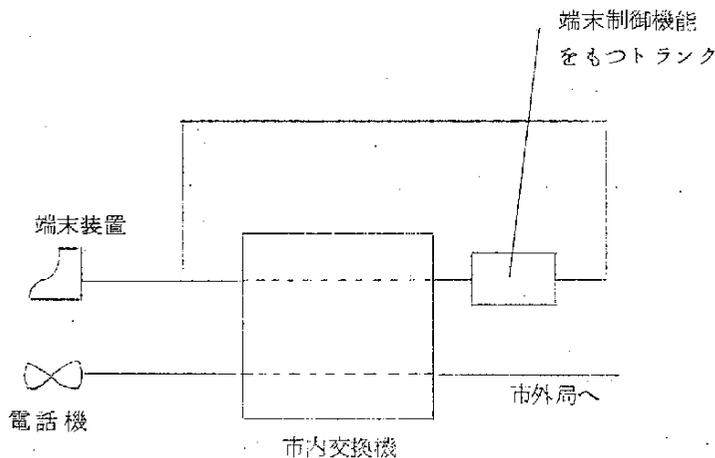


図 2.6.4 端末制御機能の局集中の中継方式の一例

図 2.6.4 は市内交換機等に端末装置の制御機能の一部をもたせる場合の中継方式の一例である。集中可能な機能としては、キーボード・プリンタ端末などの誤り制御機能、変復調機能とか映像表示装置の表示制御機能等が考えられる。

誤り制御機能の集中化については、伝送誤りのほとんどが市外伝送路で発生するという前提にたつわけであるから、この点を確かめておく必要がある。

表示制御機能の集中化については、文字パターンの発生、マーカ制御、さらに画面再生用の記憶等の機能や、表示情報の追加、削除等の編集機能の集中化が考えられる。⁽¹⁷⁾

この場合、交換機→端末方向の情報伝送は、端末の経済化をはかるうえでは、アナログ形

式が良いが、既存の加入者線や交換機の伝送可能帯域がそれぞれ100KHz, 1MHz (クロスバー交換機)程度までなので⁽¹⁸⁾, 画像品質などの点から加入者線等の広帯域化が前提と考えられ、このような集中方式の導入はテレビ電話など画像通信サービスの一環という形になるのかもしれない。

英国国立物理研究所(NPL)では、超高速メッセージ通信網という構想を打ち出している。⁽¹⁹⁾⁽²⁰⁾⁽²¹⁾ その内容は次のとおりである。すなわち、ある範囲内の端末回線を集約し、ここにバッファ機能をもつインターフェイス計算機を置く。端末とインターフェイス計算機の間では通常の端末に対してはバイト単位で、計算機端末に対してはブロック単位で情報の転送を行なう。インターフェイス計算機から外側に対してはブロック単位で送受する。このような方法をとることによって、端末の制御機能のかなりな部分を集中化できるだけでなく、端末-計算機間の転送制御を単純化し、融通性をもたせることができ、その結果、速度や機能のちがう各種の端末に対して容易に適應できるようになる。この方式はインターフェイス計算機相互間の転送方式にも特色をもつので、これについて次に述べることにしよう。

(4) データ通信統一網

NPLの提案によると、インターフェイス計算機相互間は、蓄積交換形の中継交換機を含む複合網を構成しており、この間では、パケットという一定のフォーマットのメッセージが1.5 Mb/sまたはそれ以上の、ある一定の速度で高速転送される。このようにしてインターフェイス計算機相互間の通信網は、端末相互でやりとりされるデータの速度、形式に無関係に統一化された通信網(unified network)を構成している。パケットという形でメッセージをブロック化することによって、誤り制御や中継交換機、インターフェイス計算機のバッファ処理を簡単にしている。パケットにはデータまたは状態情報のほか、発着信のアドレスや一連番号等、パケットの身元をあらわす情報が置かれる。一連番号は複合網のなかを中継する経路がパケットによって異なる場合に、相手側のインターフェイス計算機へ到達する順序がまちまちになるので、この順序を識別するためなどのものである。

このようなメッセージ交換方式では、中継点での伝送遅れが避けられないが、伝送速度を上述のように高速化することによって伝送遅れは十分小さく、例えば0.1秒程度にでき会話的使い方にも支障を与えない。

伝送速度や伝送形式の異なる様々の端末からのデータをセンターの計算機へ転送するの

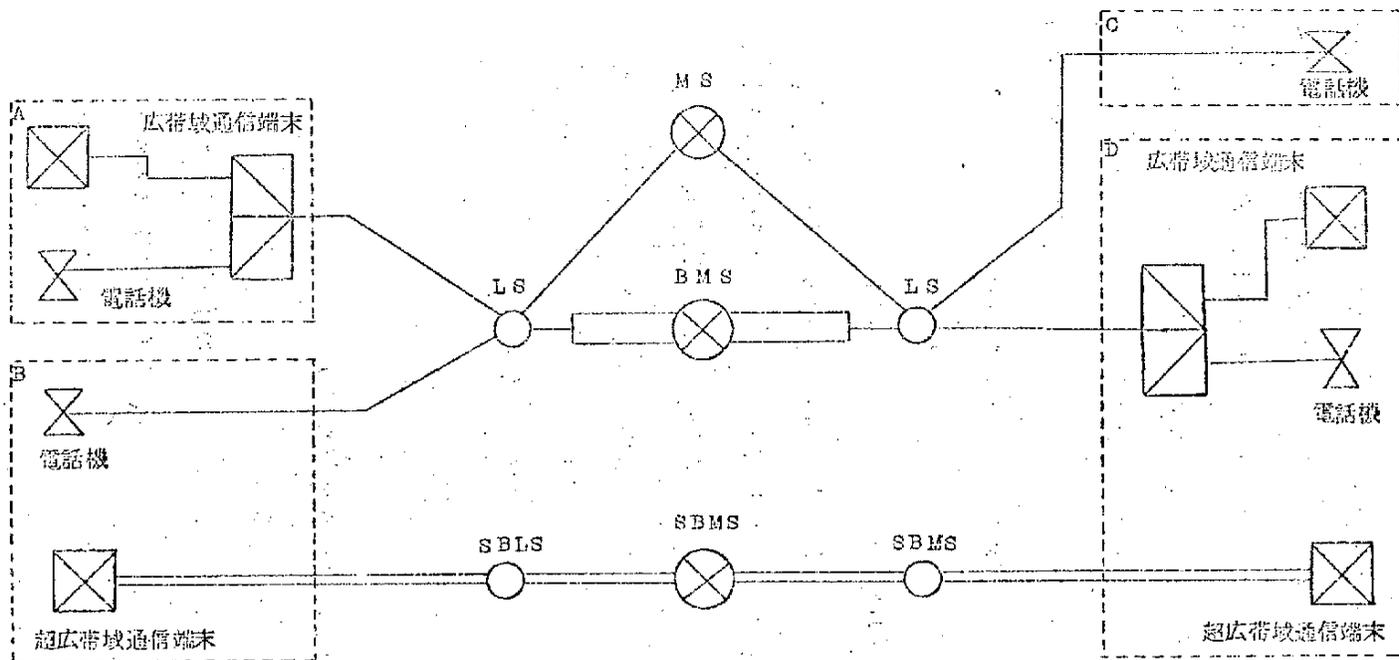
に現在普通に使われている方法は、前節に述べたように、端末において伝送速度や伝送形式を整えるか、中間に集信装置をおいてそろえるか、計算機にいたるまで整理を行わないかなどであるが、第1の方法は端末に制約を与えるし、第3の方法は伝送路と中継交換機がしわよせを受ける。第2の方法も回線を削減する狙いになる程度に過ぎない。

NPLの方式は市内局相当の階梯に計算機を置き、計算機相互間は少数の同一規格の広帯域伝送路を介して超高速で非同期転送を行なうことにより、前述の各種の方式に較べて伝送路や計算機の通信制御、それに端末装置の経済化も図っている。この方式はシステムとして理想的な方式の一つではあるが、インターフェイス計算機に膨大な投資を要し、近い将来の実現は期待しえない。しかし、電子交換機が大巾に導入された時点では、市内局階梯の電子交換機にインターフェイス計算機の役割をもたせることによって実現が可能となるであろう。

(5) PCM統合通信網と通信サービス総合網

PCM交換機とPCM伝送路を組み合わせ、変復調をくり返さずに済むようにして伝送路と交換機の経済化や通話品質の改善を狙ったPCM統合通信網の考え方がある。(図2.6.5)

この方式では、通信網内の同期方式がきわめて重要な問題で、種々の解決策が提案されているが⁽²²⁾、データ通信への適用については必ずしもあきらかではない。電話と同様端末の小束の回線を集めてそのままPCMハイウェーにのせ、所定の計算機に接続するというやり方であれば、センターの計算機の通信制御の部分で時分割符号を一旦デコードし、計算機側のマルチプレクサで再度時分割しなおす必要が生じ、計算機の通信制御にしわよせを受ける。(1)項に述べたPhisterのイメージからすれば、PCM統合通信網は計算機相互間すなわちNPLのインターフェイス計算機間に適用されることになるのかもしれない。



- A : 広帯域通信加入者
 B : 超広帯域通信加入者
 C : 電話加入者
 D : 広帯域および超広帯域通信加入者

- MS : タンデム交換機
 BMS : 広帯域タンデム交換機
 SBMS : 超広帯域タンデム交換機

図 2.6:6 通信サービス総合網

PCM統合網については、このシステムが既存のシステムにくらべ機能の付与に融通性(1)があると考えられるから、データの通信量が支配的になった場合には、データを送出していない受け方向の通話路とか送り方向でもデータを送出していない時間に対してタイム・スロットの割当てを止める方法とか、データの実時間性、重要度に応じて優先処理や誤り制御を行う方法により、データ通信網に真に適したシステムとする必要も出てこよう。

このPCM統合通信網は、電信、電話、データなどの各種の情報を包含したデジタル通信網を狙いとしたものであるが、さらにテレビ電話などの画像通信サービスへの適用や既存の通信設備の利活用に適した通信網として通信サービス総合網という提案が発表されている。(18) この提案では通信サービスを周波数帯域別に分け、一般電話網(4 KHz)現用電話網を改良した広帯域通信網(50~100 KHz)、超広帯域通信網(500 KHz)の3つの網を考えている。超広帯域通信網では、加入者線を音声信号2線、ビデオ信号4線の6線とし、超広帯域網の交換制御を電話網を通じて実施するシステムを提案している。この網構成図を図2.6.6に示す。さらにこの総合網に適する市外網として、FDMとPCMの両方式の伝送路に広帯域の伝送特性をもつ空間分割形機機接点の電子交換機を組み合わせ、帯域またはパルス速度の異なる各種のアナログおよびデジタル情報の交換機能をもたせて、非常に融通性に富んだ通信網を構成しようとしている。

提案では、データ通信サービスについてくわしく触れていないが、データ通信はその用途と帯域市に応じて、これら3つの網のそれぞれに位置を占めるようになる。この場合データ通信特有の問題としては、PCM統合通信網において述べたと同様な問題のほか、さらに節前(2)項や前節(4)項で述べた問題を初め、多くの問題について配慮する必要がある。

このようなデータ通信網としての課題は、前項のデータ通信統一網の思想などをとりこみながら解決がはかられ、このようにして音声通信、データ通信、画像通信を初めとする多彩な情報の処理、伝達を経済的かつ融通性をもって表現する総合的な情報通信網へと発展していくにちがいない。

2.6.4 むすび

前節に重点をおいて、これまでに述べた事項を要約してみると

- 1) 特定の業務の処理を機械化したシステム、たとえば座席予約や銀行預金などのシステムでは、そのシステムの扱い適用業務の特殊性を生かして経済化や運用の利便を追求している。

ロ)しかし、システムの総合化や共用化の方向は、データの形式や処理方法の多様化をうながし、これらの要求を満たすデータ通信網を必要としている。

ハ)このようなデータ通信網は、適用業務の性格あるいは信頼性や経済性などシステムとしての要求条件から、一つの計算機センターを中心とした網から複数の計算機センターを含むデータ通信網を構成するようになるだろう。

ニ)このようなデータ通信網は、従来のデータ通信網のもつ各種通信機能等に加えて次の条件を具備することが望ましい。

i) 端末装置の入出力形状、伝送速度等の多様化に対応できるものであること。

ii) 計算機相互を結ぶ通信回線等に対し、高速伝送(広帯域伝送)が可能であること。

iii) 迅速な(交換)接続を要する適用業務に適応できること。

iv) データ通信網のコストのかなりな部分を占める端末装置を、端末機能の一部集中化等により経済化できること。

v) 送り方向が広帯域で受け方向が狭帯域の上下非対応の網や端末回線において端末受が広帯域で、送り方向が狭帯域の網など融通性に富んだ網構成が可能であること。

vi) 機能や設備の共用化によって、電信電話や画像通信サービスをも含めた総合的な経済化が達成できること。

ホ) データ通信網の網構成としてデータ通信統一網、PCM統合通信網、通信サービス総合網を取り上げ、上述の観点を含めてデータ通信サイドの問題点を検討した。

ヘ) 既存の加入網は上述の条件をあまり満たしていないが、既存設備の活用による経済性の点で優れており、将来の網への過度段階として、あるいは加入者線階梯など網の一部として十分な存在意義をもっている。なお、加入網は本来データ通信用として設計されたものでないから、その適用にあたっては、網に及ぼす影響等について慎重な配慮が必要である。

参考資料

(1) 米沢：“地銀協データ通信システムの設計概要”，施設Vol.19, No.7(昭和42年7月)

(2) ビジネス・コミュニケーション編集部編：“テレックス・アプリケーション”企画センター，昭和44年

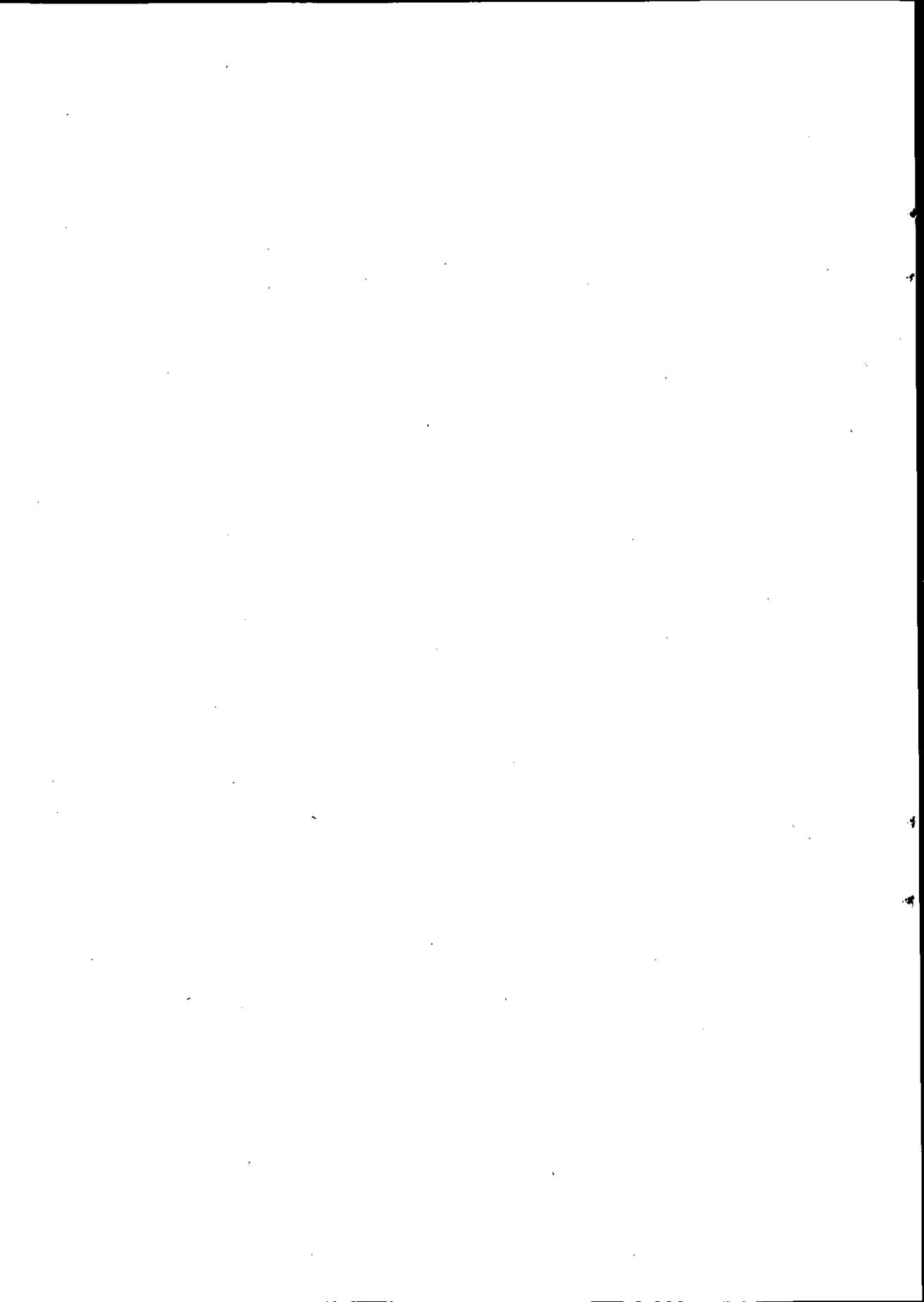
(3) D. E. Carruth：“Advanced Record System”，Western Union

Technical Review, Jan, 1966

- (4) J. A. Hunt: "Information Service Computer System Expands Western Union Services", Western Union Technical Review, Apr 1967
- (5) 伊藤他: "国鉄の座席予約システム", 信学誌, Vol 52, No 1, (昭44年1月)
- (6) 緒方: "オンラインシステム入門", 企画センター, 昭43年
- (7) 川端他: "データ伝送技術" 信学誌 Vol 52, No 4 (昭44年4月)
- (8) FCCの諮問に対するIBM社の回答
"Regulatory and Policy Problems presented by the Interdependence of Computer and Communication Services and Facilities," Docket 16979, P1-54
- (9) F. Gruenberger: "Computers and Communications - Toward a Computer Utility," P. 203, Prentice-Hall, Inc, 1968
- (10) J. B. Dennis: A Position Paper on Computing and Communications Com. of the ACM, Vol 11, No 5 (May, 1968)
- (11) "コンピュータと通信に関するBBMAの回答," P 182 日本情報処理開発センター 昭. 43年9月
- (12) (9)に同じ P 135
- (13) M. J. Elton: "Message Switching Techniques in digital Telecommunication Networks," Data systems, nov, 1968
- (14) (9)に同じ P 79
- (15) P. M. Walker: "Data Transmission and the foreign Attachments Rule", Datamation Feb, 1969
- (16) A. D. P News Letter Vol XIII, No 17, may 6, 1968
- (17) 釜江他: "テレビ電話機のデータ・ディスプレイへの応用" 通研実報, Vol 18, No 3, P 800, 昭. 44年3月
- (18) 熊谷他: "通信サービス総合網の提案(その2)", 通研実報 Vol 16, No 3, 昭. 42年3月
- (19) CCITT Com, SPA-No, 189-E

- "A Digital Communication Network for Computers
giving Rapid Response Remote Terminals" Jan. 1968
- (20) D.W.Davies: "Communication networks to serve Rapid
Response Computers" Proc. of IFIP Congress 1968
- (21) D.W.Davies: "A Versatile Data Communication Network
Teleinformatique mar. 24 1969
- (22) 尾佐竹他: "POM交换方式", 信学誌 Vol 149 No 11 昭41年11月

3. 遠隔情報処理業務例



5. 遠隔情報処理業務例

3.1 遠隔情報処理としての計算サービス

3.1.1 遠隔情報処理としての計算サービス

電子計算機が、情報処理システムにおいてかなめの役割りを果していることは、いまさら言うまでもないことである。電子計算機という言葉自身が示すとおり、計算機発展の歴史における中心命題は、計算速度の向上と計算量の増大とであった。

最近にいたり、通信回線と電子計算機の結合による遠隔情報処理すなわちデータ通信が、その質、量両面において急速に広がりを見せている。計算サービスが、遠隔情報処理の分野で、サービスの重要な形態と考えられていることは、上述の点からいえば、ごく自然なことである。

しかしながら、遠隔処理としての計算サービスが実用化されたのは、他のデータ通信サービス、たとえば座席予約、為替交換等にくらべると比較的新しい。米国において、商用計算サービスが開始されたのは、僅々数年前のことにとすぎない。計算サービスの実現がこのように遅れていた理由のうち最大のもの、計算サービスが本質的に汎用サービスとしての性格を有することであろう。すなわち、先に示した座席予約等のサービスでは、システムはあらかじめ定められた処理のみを取扱えばよい。これに対して、計算サービスでは、電子計算機の機能をできるだけ小さい制約で操作できることが要求される。このように、計算サービスは、電子計算機システムを制約なしに共同利用するシステムへと発展して行く性格を持ったものである。

もつとも、計算といってもたくさんのレベルのものに分けられる。そのすべてが、電子計算機システムの汎用的使用を要求することはない。しかしながら、現時点におけるもつとも代表的な計算サービス用システムは、タイムシェアリング・システム (Time sharing system: TSS) である。そして、このシステムは、将来のコンピュータ・ユティリティにおいてその汎用的性格のゆえに重要な役割りを果すと考えられるものである。この意味で、本節においても、TSSによる、科学技術計算を主目的としたシステムに重点を置いて、現状と実施上の問題点につき以下のべることとしたい。

3.1.2 発展の背景および現状

(1) 発展の背景

周知のとおり、米国におけるここ数年の商用計算サービスの発展はいちじるしう。このように短時間で、計算サービスがかくも急速な発展をとげるに至つた理由として、いろいろな説明すけがなされている。

たとえば、(1)大型電子計算機にしかない機能が多数共同利用のため比較的 low price で利用

できること、通信網と電子計算機の結合技術の進歩、大容量ランダム・アクセス・ファイルの発達、信頼性の向上、需要層の広まり、などがあげられる。

しかし、直接の契機となったものは、TSS技術の進歩である。さらに、人間と計算機との会話、ないしは相互作用の有効性が、実験により経験的にも裏付けられたことである。

TSSの概念が、はじめて公式の席上で提案されたのは、1959年である。

実験は、1960年からMITによってはじめられた。これはCTSSとしてまとめ上げられ、TSS技術の基礎に大きな貢献をなしている。このあたりの事情については、すでに多くの文献で紹介されているので、本節では省略することとしたい。

人間と電子計算機との相互作用という点については、初期に考えられたほど大きくはない、というのが現時点における一般的な見解のようである。(2)しかし、ターンアラウンド・タイムの短縮、使いやすさ、など、会話モードのTSSによるサービスが、問題解決の道具としての電子計算機の有効性を、大巾に拡大している。たとえば、問題を解く際に、パラメータ等を変更修正して最良の解に達するという使い方は、従来ならかなり長い日時を必要とするのに、TSSは時間的に大巾な短縮が可能となる。

こうした有利さが、技術的には第3世代の電子計算機技術によって支えられ、発展のための環境が急速にととのったものと言えよう。

(2) 現 状

1965年に、米国で商用の計算サービスを提供していたのは、BBN, CEIR, GE, IBM, Key Dataの5社にすぎなかった。それが、1968年始めには30の企業により70個所のセンターが運営され、研究開発用についてはそれを上回っている。(3)商用システムは、年率75%という成長をとげているので、1969年には180システム、3万台の端末で総収入は1億4千万ドルに達するものと推定されている。(4)

商用の計算サービスを提供している企業体は4種にわけられる。すなわち、計算機メーカーまたはその子会社であるサービス会社、通信会社、独立専門会社、および大手ユーザーが自家用システムの余力を売る場合、とがある。現在大きいシェアを有しているのは、GEと、IBMの子会社SBCとであるが、独立系会社のうちではCall-A-Computer, Com-Share, Tymshare, Allen-Babcockの4社がそれに次ぐ地位をしめている。今後、まだまだ多くの企業が新規に開業するので、競争は激化するであろう。

技術的な方向は、いまのところ全く定まっていない。(2)すなわち、どのようなシステム

によりどのようなサービスを提供すべきか、などについては見通しはつけがたい。

サービスの内容についてみると、現用の商用システムは、大部分が実際上一つだけの言語しか使用できないものである。たとえば、GEのBASIC, Allen-BabcockのPL/Iサブセット, BBNのTELCOMP (JOSS)などがこの例である。科学技術計算を目的とするのが大部分であるため、FORTRAN, BASICが殆んどである。事務用はまだ例がすくないが、Key Dataの伝票作成サービス, IBMのDATA TEXTサービスなどがある。いずれも、機能的にはかなり単純化されたものである。

汎用をねらった大規模システムに対しては、MITのMULTICS, IBMの360/67がいずれも難行しているため懐疑的である。しかし、中規模のシステムでSDS-940によるものが、現在Com-Share, Tymshare両社を中心としてサービスに供されているが、これらは、将来において汎用多目的システムが商業ベースで成り立つという見通しを与えてくれるもの、と評価されている。(2)

ここで、わが国における状況を見ると、電気試験所、電気通信研究所をはじめ、計算機メーカー、大学などにより、いくつかのシステムが開発されている。また、株式会社電通はGE 625を導入して社内向けのTSSによるサービスを開始している。これらは、今のところ実験的色彩の強いものである。商用サービスとしては、電々公社が加入データ通信サービスを準備中である。昭和45年度開始予定のものは、簡易計算、販売在庫管理、科学技術計算の3種類である。日本においても、本格的なTSSの時代が、まさにむかえられようとしているものと言うことができよう。

3.1.3 計算サービスの諸方式

(1) 各種サービスの位置づけ

計算サービスとひとくちに言っても、その内容は多岐にわたる。サービス面からは、たとえば次のような点について分類が可能である。

- (a) サービスの特殊性—特殊目的専用、一言語、多言語ないしは汎用
- (b) 会話的性格—レスポンス・タイムの大きさ、または言語が会話的か、コンパイル・ゴ—形式かりモート・バッチか
- (c) ファイル利用—ファイル無し、限定された利用、汎用

このような区分は、計算サービスに限らず他のサービス分野をも包含できる。その広い体系の中での計算サービスの位置づけを考えると、さきにも述べたとおり、計算サ

サービスとは特殊性が小さく、レスポンス・タイムは極端に短い必要はなく、ファイル利用はまず必須、と考えられるようなものである。

こうした条件を満たすために、どのような処理方式がとられているかをつぎに検討する。

オンライン・リアルタイムと一般に言われている方式は、工業制御システムから、座席予約、バンキング・システム等にいたるまで、広汎な目的に用いられている。この方式では、一列に並んで待合せている処理要求を、順次とかまたは一定の選択順位を設けて処理して行く。一つの処理のためのプログラムの実行は、つぎのプログラムが開始される前に終了する。

計算サービスで、このオンライン・リアルタイム方式を適用しようとする、用途はかなり限定されたものとなる。すなわち、TSSの本来の機能は、処理時間の短いユーザーが、処理時間の長いユーザーとプロセッサを共用するのを許すことにある。したがってこの機能を持たないオンライン・リアルタイムで処理可能なのは、明らかに、処理業務が定型化していて、かつ処理時間が比較的短いものに限られる。前述したサービスの特殊性という観点で考えると、特殊目的用、または単一言語で強い制限のおかれているような場合に限られる。

TSSは、これに比べて自由度が大きい。主記憶と二次記憶間でプログラムをスワップすることにより、処理を人間の思考に合わせて進めることが可能となる。反面、このスワッピングはシステムのオーバヘッドの大きな部分を占めるので、能率は高く保てない。

この対策として、ページング方式、ヴァーチャルメモリ方式、また大容量磁心記憶方式など各種の方式が試みられている。いまのところ決定的といえるものはないが、IBM 360/85のローカルメモリ方式など今後の考え方にかなりの影響を及ぼすように思われる。

ここで米国における商用計算システムが、どのように位置づけられるかを考えてみると、大勢として次のようなものである。

- (a) 中型計算機を用いたTSS
- (b) 言語は一種類。オンラインでFORTRAN Vを使用できるシステムは少ない。
- (c) ファイルは利用可能
- (d) 会話形処理だけ。リモートバッチ併用はまだ少ない。
- (e) レスポンス・タイム2～10秒程度で40～50回線が接続可能。

(2) 電々公社の加入データ通信サービス

前項では、計算サービスに関する一般論をのべた。本項では、具体例として、加入データ通信サービスについて、その特徴をのべることにしたい。

加入データ通信の対象としては、前述した簡易計算、販売在庫管理、科学技術計算の3種の他にも、将来は情報案内、教育、キャッシュレスなど広汎な分野のものが考えられる。当面は、需要見込、技術的難易度などを考慮して、上記3種が選定され、現在準備がすすめられている。

表 1 サービス内容の概要

サービス種別		簡易計算	科学技術計算	販売・在庫管理
サービス内容		簡易計算	科学技術計算	販売・在庫管理
処理形式		即時処理	即時処理 一括処理	即時処理 一括処理
処理内容	端末で受けるサービス	直接計算 定義計算 ライブラリ計算	ファイルの作成、 加入者プログラム 作成・実行、ライ ブラリ計算	伝票作成ファイル更 新 ファイル問合せ
	センタで受けるサービス	なし	大量データの印刷	加入者プログラム作 成、ファイル初期作 成、日報・月報の印 刷
加入者ファイル		定義式5個および 定数または中間結 果10個を一時的 にストアできる	共用の一時ファイ ルにストアできる ほか専用の加入者 ファイルを貸与す る。	専用の加入者ファイ ルを貸与する
加入者が端末から使用できる言語		指示(押しボタン 操作による)およ び特定様式の言語	ターミナル・コマ ンドとFORTRAN	ターミナルコマンド
ライブラリ		高次方程式、 連立方程式、 数値積分、平均分 散、分散分析等	左記のほかLP、 PERT、シミュレ ーション等 多数	業種別の標準パッケ ージ・プログラム
加入者宅内装置		押しボタンダイヤル 電話機、簡易プリ ンタ	キーボード プリンタ (注)	キーボード プリンタ (注)

(注) 紙テープリーダー・パンチつき

これら3種のサービス内容の概要は、表1(1)に示すとおりである。前項でのべた計算サービスとしての特殊性という点からながめると、つぎのようになる。

- (a) 販売在庫管理—特殊目的システム
- (b) 簡易計算—特殊な単一言語システム
- (c) 科学技術計算—汎用システム(当面はFORTRANのみに限定される)

販売在庫管理サービスは、計算サービスというよりは、ファイル更新、維持を目的とする。各種伝票の発行、ファイル内容の問い合わせ等は、オンラインで可能である。日々のトランザクション情報は、ジャーナルとして磁気テープに記録されて行くので、これを分類、集計して週報、月報等が作成される。

伝票作成、問合せは、一件あたりの処理時間が短かくかつ比較的定形化されているのでTSSによる必要はない。すなわち、一件の処理ごとに該当するプログラムを読み込んでくる。各トランザクションは相互に独立である。このような方式をとることにより、処理可能回線数はかなり大きくできる。

簡易計算は、宅内装置として押しボタン・ダイヤル電話機を用いる、きわめてユニークなシステムである。回答は、音声応答装置からの音声によって与えられる。サービス内容は、入出力の点で人間工学上ある制約内にとどまらざるを得ない。しかし宅内装置がごく安価なこと、かつ非常に多数用いられるであろうから、きわめて大衆性の高いサービスである。

操作上は、最近普及しつつある電子式卓上計算機にくらべて劣ることはやむを得ない。しかし、簡易な印字装置などを付加すれば、かなりの程度までこの欠点を克服できる。一方、質的な面ではライブラリとしてかなり高級なものまで組みこめるから、卓上計算機とは違った独自の分野を確保することになる。

この簡易計算も、処理方式上はスワッピングを必要としない。

システムの構成は、つぎに説明する科学技術計算サービスを含めて、いずれも待機予備方式(デュープレックス)をとっている。すなわち、2台の中央処理装置のうち、1台をオンライン処理(即時処理)に用い、他の1台はオフライン処理(一括処理)およびオンライン側ダウン時の予備装置として用いられる。

科学技術計算システムでは、一括処理が完全にオフラインの処理ではない点が、すこし異なっている。すなわち、図1のように、2台の処理装置はファイルを介してデータの受け渡しを行なっている。(5)その他、このシステムの主な特徴をあげると次のとおりである。

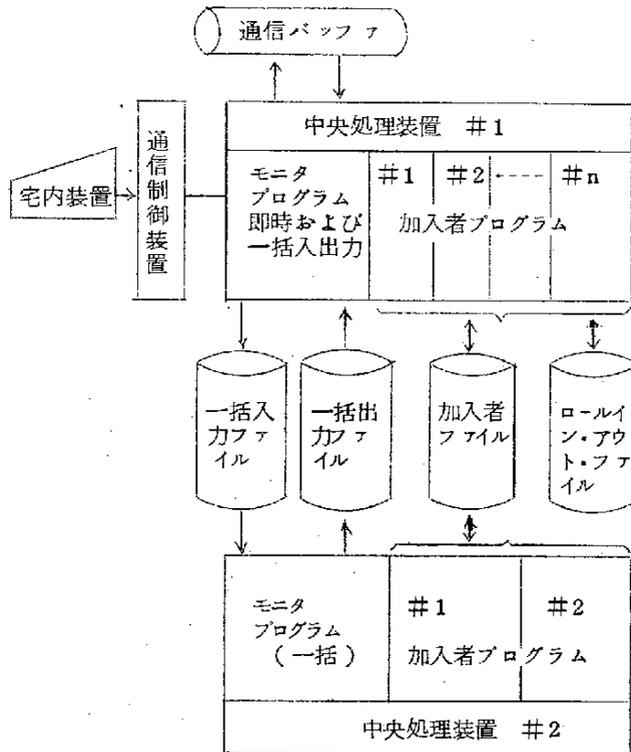


図 1 科学技術計算システムの概念図

- (a) TSSによる処理と、リモート・バッチ処理が、2台の処理装置で別々に実行される。ユーザーは、会話的処理の便利さとリモート・バッチ処理の能率の良さを、適宜使い分けることができる。
- (b)上記2つの処理間では、メモリサイズ、ファイル数などの違いは当然ある。しかし、ソース・モジュール、オブジェクト・モジュール、ロード・モジュールの各レベルにおいて互換性がある。
- (c) FORTRAN JIS7000レベルの言語が使用できる。

- (d) 会話形処理の利点を生かすため、FORTRAN 構文チェック、デバッグ用コマンドが用意されている。
- (e) コマンドは、投入の容易さを考慮して英語であるが、システムからのメッセージはカナを用いる。
- (f) 加入者用ファイルには、作成したファイルを登録、保存しておくためのものと、作業用の一時ファイルとがある。
- (g) リモートバッチ処理が終了すると、システムは自動的に加入者宅内装置を呼び出して、結果の送出行なう。

3.1.4 計算サービスの問題点

計算サービスは、目下非常な拡張期にあり、技術的にも多くの問題がある。ここでは、前項までのべてきた動向の補足として、いくつかの問題点につきふれることにしたい。

- (1) TSSとバッチ処理の経済性の比較については、多くの議論が展開されてきた。

この点におけるTSSの基本的なメリットは、いわゆるGroschの法則にしたがり大規模化のメリットである。これに対して、費用面でマイナスと考えられるのは、端末装置および回線のコストである。さらに、間接的には、通信回線を経由するサービスであることにもなる障害対策、ファイル・システム等の複雑化がコストに加わる。ここで計算機本体のコストは急速に低下しているのに、通信回線のコストは加入者ごとに電話局から線をひくことがどうしても必要になるため、それ程下るとは期待できないことに注意を払っておかなければならない。

経済性の問題に関しては、いろいろ異論もあるが、つぎのような考え方(6)が一応の結論であるとおもわれる。すなわち、TSSを用いると人間の側の能率は向上するが、多くの計算機時間が必要になる。利用者は多くの場合、TSSに好意的である、というものである。

量的にみた計算能力についてのみであれば経済比較により判断を下すことはできる。機械利用の能率という点でみれば、バッチ処理の方がすぐれていることは当然である。たとえば、7094を使用したバッチ処理で1時間かかるプログラムを、QUIKTRANにより40個のコンソールの1つを通して処理すれば、1ヶ月程度かかるという見積りすらある。(7)

しかし、TSSはこうした比較のみで評価してはならない。TSSは、計算機の可能性の範囲を拡張するものであって、新しい需要を創造するものである。したがって、

経済性は本質的な問題ではなく、新しい応用分野の開拓こそがTSSSについての今後の責務である。

(2) 計算サービスの対象範囲の拡大

ここでは、2つの面について考察する。

第一はリモートバッチについて、第二は汎用性への一ステップとしての事務処理への適用についてである。

前述したとおり、TSSSは機械の能率という点ではバッチに劣る。したがって、これを利用するのは、新しいプログラムの作成とか、人間と機械の相互作用が有効に利用できるような場合が望ましい。ユーザーの介入を必要としないすて作成済みのプログラムの処理を行なう場合は、会話的機能はあまり役に立たない。したがって、リモートバッチは、TSSSと補完的な機能であると言える。一つのシステム内でこの2つがコンパティブルになっていることはきわめて望ましいが、さき述べてとおり電々公社のシステムではこれを実現することとしている。

つぎに事務処理への適用については、科学技術計算にくらべてかなりおくられている。純事務用ということで注目されたKey-Data社のシステムも、その後の実績はあまりかんばしくないようである。IBMによる文書編集のサービスが最近かなり拡張されているが、事務処理の分野での本命であるとは考えられない。

事務計算分野への適用がおくれた理由として、次のようなものがあげられている。(3)

- (a) 事務計算を必要とする層は、計算機に関する経験がすくなかった。
- (b) 膨大な事務用ファイルを自由に取扱うためには、現状では記憶装置に関する制約が大きすぎる。

その他に、広範な適用範囲を持つ有効なプログラム・パッケージの開発という困難な問題がある。この点は、米国においても目下の所解決策は見い出されていない。

また、事務処理システムについては、障害に対して特にファイルは厳重な保護策を講じておかねばならない。

事務処理の問題は、リモート・バッチよりは難かしいが、潜在的にはきわめて大きなマーケットであるだけに、ファイルの取扱いを中心として今後多くの努力が払われて行くことになるう。

(3) システムの信頼性

米国における商用システムは、大部分が予備系なしのシングル・システムである。い

ままでのところ、信頼性に関する苦情はそれ程大きくはないようである。しかし、今後適用領域が拡大され、また機能も次第に複雑化することを考えれば、いつまでも今の水準で良いとは思われない。

ハードウェアの素子レベルでは、信頼度は非常に改善されている。しかし、システム規模の拡大もいちぢるしいので、系全体としてみるとあまり良くなってはいない。したがって、ソフトウェア的な手段により、信頼度を更に高めることが望ましい。

かなり古くから注意が払われてきたのは、加入者ファイル内の情報の保護についてである。一般に行なわれるのは、適当なタイミングでファイルのコピーを磁気テープへとっておくことである。この周期をあまり長くしては保護の効果が低くなる。さりとて、頻度を高くすると処理能力に影響する。また、復旧に要する時間の要素も考えておかねばならない。したがって、一日ないし一週間間隔で全部の内容のコピーをとり、数時間間隔でその間隔内に内容の変更が行なわれた分についてのみコピーをとるといった、組合せが考えられている。

それ以外の障害についても、最近ある程度の検討が行なわれるようになってきたが、(8)商用計算サービスの分野ではコストとのかね合いでほとんど考慮されていないのは前述したとおりである。将来、マルチコンピュータ・システムが用いられる場合には、ハードウェアの一部分の障害が全体におよぶことのないような対策が必要となる。しかし再開点の設定、そのための情報の確保、ファイル障害時の対処など、比較的小さなシステムですらすべての場合を洗い出すことは、インプリメンテーション上もきわめて難かしい。今後の大きな課題といえるが、ここでは、一例として電々公社の科学技術計算システムにおける信頼性確保のための対策の主なものについてのべることにする。

- (a) 中央処理装置をデュープレックス構成とすることにより、サービス再開までの時間を短かくする。
- (b) 本体装置を、周辺機器に対して対称に構成することにより、障害回復時にもとの配列に切り戻す必要はない。
- (c) 主な周辺装置は2重化するか、または予備装置を設ける。
- (d) 加入者ファイルの内容は、夜間コピーをとるほか、オンライン処理中もファイル更新、登録などの際に磁気テープへコピーする。
- (e) 処理中の各タイミングでチェック・ポイント情報をとり、障害回復時の再開を円滑に行なわせる。

- (f) システムの過負荷に対して、ストール状態とならないように、検出および措置のための機能を用意する。
- (g) 電話網を通した大汎な加入者に対する障害対策のため、センターに試験受付装置を設置する。加入者に対する案内用窓口の役割も兼ねさせる。
- (h) システム内各装置の動作状態を集中的に監視する装置を設ける。表示ランプ、警報ブザー、押しボタン・スイッチなどを持ち、装置の切りかえもここで行なう。

(4) 機密保持

不特定多数のユーザーが共同利用するシステムでは、ユーザーの情報の機密を保持することはきわめて重要である。これは、通信の秘密を保持すると同等のレベルで考えるべき性質のものである。

機密保持のための一般的な方法は、パスワードによる方法である。(9)パスワードは、他人に知られてしまうと保護機能がなくなってしまうので、乱数を用いて暗号化する方法も試みられている。(10)

電々公社のシステムでは、これよりもさらに機密保護を厳重に行なうための手段を設けている。すなわち、宅内装置に組み込まれているIDコード(加入者識別符号)、SAコード(相手確認甲局指定符号)を利用する方法である。

通話開始時には、まず端末側からIDコードを送って自己を表示する。センタでは、すぐ登録者リストのIDと照合し、OKであるとその端末のSAコードを探し出し返送する。これが、その端末が持っているSAコードと一致しなければ通信は開始されない。加入者ファイルへのアクセスのためには、このようにして2重に確認されたIDコードが必要とされるので、機密保持はパスワードだけによる場合より、はるかに確実である。

このシステムでは、普通のパスワードも使用できるが、これは加入者内部での利用者制限という補完的な役割を果たすものと考えられる。

機密保持を一層確実にするためには、加入者端末が収容されている電話交換機で行なう方法がある。現在用いられている交換機はそのままではこの機能はない。しかし、将来において電子交換機が利用されるようになれば、データ通信システムが、電気通信網のこうした機能を利用することは、当然検討されてよいとおもわれる。

(5) 標準化

計算サービスを利用する人口が増大するにつれて、標準化に対する要望は大きくなってくる。標準化の対象と考えられるものは、加入者コマンド、およびプログラム言語で

あろう。ファイル・システムについても、これらの標準化を可能にする程度には、当然のことながら統一がはかられていなければならない。

このうち、プログラム言語については、すでにISO、JISで作業がすすめられており、FORTRANとALGOLについては、すでに規格ができています。電々公社のシステムでも、当然のことながらJISに準拠したFORTRANを使用することとしている。

コマンドについては、見通しはくらい。一般に、オペレーティング・システムを標準化しようとするとき、システム制御カード、メッセージの書式、コーディング・シーケンス、レイベル記法などが対象となる。(1)今のところ統一への動きもほとんどないし、また思想的にもまちまちであるため、あまり見通しはない。しかし、コマンドについては、オペレーティング・システム全部を統一するよりは可能性が大きい。ただしTSSに関して標準的方式、手法といえる程のものはまだないので、米国においても今のところ動きはないようである。

しかし、今後システムが大規模化するにつれ、その開発期間、費用も大きくならざるを得ないので、計算サービスを提供する側からみても、その寿命はできるだけ長いことが必要である。この必要性からも、標準化への動きは今後も努力が続けられるであろう。一つの理想としては、電子計算機方式が素子の技術から独立の生命を有することが望ましいように、ソフトウェアが特定の電子計算機から独立した存在となることである。これをねらいとして、システム・プログラムをPL/Iなどの高レベル言語で記述するために多くの努力がなされているので、(2)いずれはこうした理想を達成されることになる。

3. 2. 5 むすび

以上、遠隔計算サービスの現状、およびその問題点の若干についてのべた。本文において記した通り、この分野は変化がきわめて激しい。わが国においても、商用の遠隔計算サービスが、45年度開始を目途として電々公社の手で準備がすすめられている。

このような状況を背景として、遠隔計算サービスの技術は、ここ数年で急速な変貌をとげることが予想され、動向については注目が怠れない。

参 照 文 献

- (1) 美間敬之：加入データ通信方式
施設 Vol. 20 No. 10 (1968-10)
- (2) J. I. SCHWARTZ: Interactive systems - Promises, present
and future FJCC, 1968
- (3) D. C. WHITNEY, C. H. WHITE Jr. Time-sharing Services
Modern Data Systems Vol. 1, No. 1 (Feb., 1968)
- (4) 高柳 晃：アンケート
ビジネス・コミュニケーション Vol. 6 No. 3 (1969- 3) P. 123
- (5) 山田雅朗, 亀島昭徳ほか：加入データ通信方式センタ設備 (その2)
施設 Vol. 20 No. 12 (1968-12)
- (6) H. SACKMAN: Time-sharing versus batch processing: the
experimental evidence
SJCC, 1968
- (7) スチーブンド・ボベル著, 石原善太郎訳：タイム・シェアリング
日本経営出版会 1967, P. 73
- (8) G. OPPENHEIMER, K. P. CLANCY: Considerations for
software protection and recovery from hardware failu-
res in a multiaccess, multiprogramming, single proce-
ssor system FJCC, 1968
- (9) 水野幸男：TSSのファイル管理
信学誌 Vol. 50 No. 4 (1967-4)
- (10) 益田隆司ほか：HITAC5020TSSのファイルシステム 情報処理 Vol. 9
No. 6 (1968-11)
- (11) JOHN A. GOSDEN: Software compatibility: What was
promised, what we have, what we need FJCC, 1968
- (12) たとえば J. H. SALTZER: MULTICS SYSTEM
日本電子工業振興協会 昭42年2月

3.2 自動遠隔検針

データ通信の発達から、電子計算機と通信回線を結んで、遠隔の地にある計測器類の指示値を自動的に読み取って、これを磁気テープ、磁気ディスクなどのファイルに書き込むということが可能になってきた。すなわち、メータ類の自動遠隔検針が可能になったのである。

従来から、電気、ガス、水道などの家庭用のメータ類は、検針員が一軒づつ廻って歩いて、検針し、これを記録して前月の指示値との差し引きをして、当月分の使用量を算出決定して、集金員がこれを集金して歩いているのである。加入者の数が少いうちは、一応これで用が足りているが、一都市内の加入者数が数十万、数百万という膨大な数に達し、検針員、集金員の人員費が昂騰するようになり、また、労働力不足によってこれらの人手が得られなくなれば、何らかの改善手段が考えられなければならない。また、人手による検針では、計量値のよみ違いや、計算間違いなどが問題となるため、電気、ガス、水道などの事業会社では、検針方法の改善に意を用いている。

これらの努力の一環として、遠隔検針、自動検針などが検討され、過去50年間に、メータを遠隔地から自動読みとりする方法が多くの人達によって計画されて、数々の方式が提案されてきた。一般家庭用のメータ類に対する遠隔自動検針は、未だ実用の域に達してはいないが、これを応用した工業用メータの読み取りには実用されているものもある。

一般家庭用のメータ類の読み取り方法の改善策として、(1)写真による計量値の撮影、(2)メータのテープ・パンチ方式、(3)携帯用テープレコーダによる方法、(4)自動パンチ・カード方式、(5)OCR (Optical character reader) 方式、などによって、検針員の負荷の軽減、計算集計事務の能率化が検討されている。

これらは、いずれも自動遠隔検針への一段階と考えられているが、直接自動遠隔検針につながるものではないので、ここではこれらの説明は省略し、現在まで考えられている自動遠隔検針の主な方式についてその概要を解説し、現在電電公社で開発中の方式についても多少ふれることにする。

3.2.1 自動遠隔検針の必要性

人手による検針では戸別訪問が不可欠となるので、相手の状況によって検針が非常に困難になる場合が生じ、外国の例によれば、最近これによるトラブルが検針能率の低下をまねくに至ったとさえ言われている。その主な原因は、(1)多くの家庭の主婦の就労による不在、(2)勤労者のレジャーブームによる遠出のための不在、(3)昼間無人世帯の増加傾向、などであると言われ、わが国でも大都市においては、除々にこのような傾向が現われつつあることは否

定できない。

この点、自動検針では、計量値を計算機入力として簡単に磁気テープ、磁気ディスクなどに記憶させることができ、また必要な時はいつでも検針ができ、負荷の状態が常に把握できるので負荷制御に利用することもできる。時間による計量も自由に行なうことができる、等の幾多の利点がある。

ここで、自動遠隔検針としての技術上の問題点をあげれば、

- (1) ガス、電気、水道などのメータ類の指針値を電氣的に取り出すこと。
 - (2) 取り出した指針値に相当する電気信号を伝送できる符号に変換する装置を作製すること。
 - (3) 中央の計算センターまで、この符号を伝送すること。
 - (4) 中央の計算センターにおいては、これを受信して、処理し、結果を記録すること。
- などがあり、各項目とも非常に大きな問題をかかえている。

伝送路を新たに設置すれば莫大な資金が必要であるため、電話線、配電線などを利用することによって解決をはかり、また端末装置の値段を下げるためには種々の方式が考案されている。以下これらの問題点と各種方式について説明する。

3.2.2 自動遠隔検針の諸方式を大きく分類してみると、

- (1) 走査方式
- (2) 直読方式
- (3) 中継方式
- (4) その他

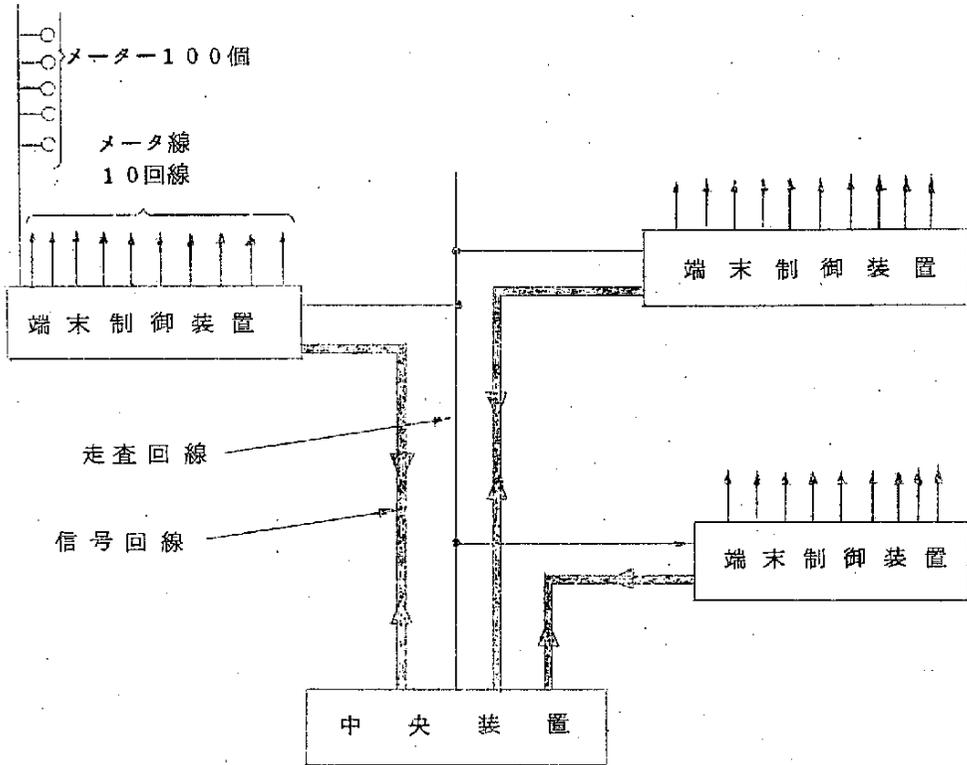
に分けることができる。これらの各種方式につき代表例をあげて説明し、その他多くの実験例の概要をつけ加える。

- (1) 走査方式とは、計量の1単位毎にメータから情報を取り出すようにし、検針センターは伝送路を通してこれら多数のメータを周期的に常時走査監視し、メータからの1単位毎の情報を読み取り、各メータ毎に積算して集計する方式である。

この方式では読み取りのための走査周期は単位計量の消費時間以内にする必要があり、従って相当短い周期で、しかも常時走査しなければならないから、伝送路としては専用線を用いなければならない。この方式の特徴はメータの構造が極めて簡単でよいことであり、また一つの伝送路に多くのメータを並列に接続し、それぞれ異なる情報を送出するようにしておけば、伝送路としての専用線の数は非常に少なくてすむ利点がある。この方式においては各メータの積算値は常に検針センターにあり、従ってこのデータは料金計算以外の

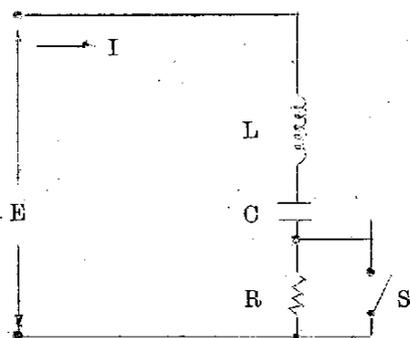
目的にこれを使用することが容易である。

この方式の代表的な例として、GB社の検針方式がある。第3.3.1図はそのブロック図である。

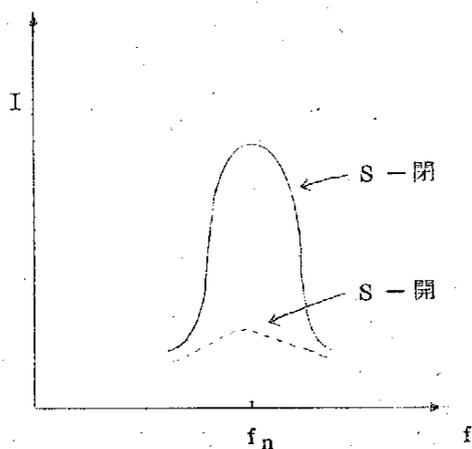


第 3. 3. 1 図

この方式では各メータは第3.3.2図のようなそれぞれ異なる固有の周波数の共振回路をもち、接点Sがメータの単位計量値毎に閉じるようになっており、その電流特性は第3.3.3図のようになる。



第 3.3.2 変換回路

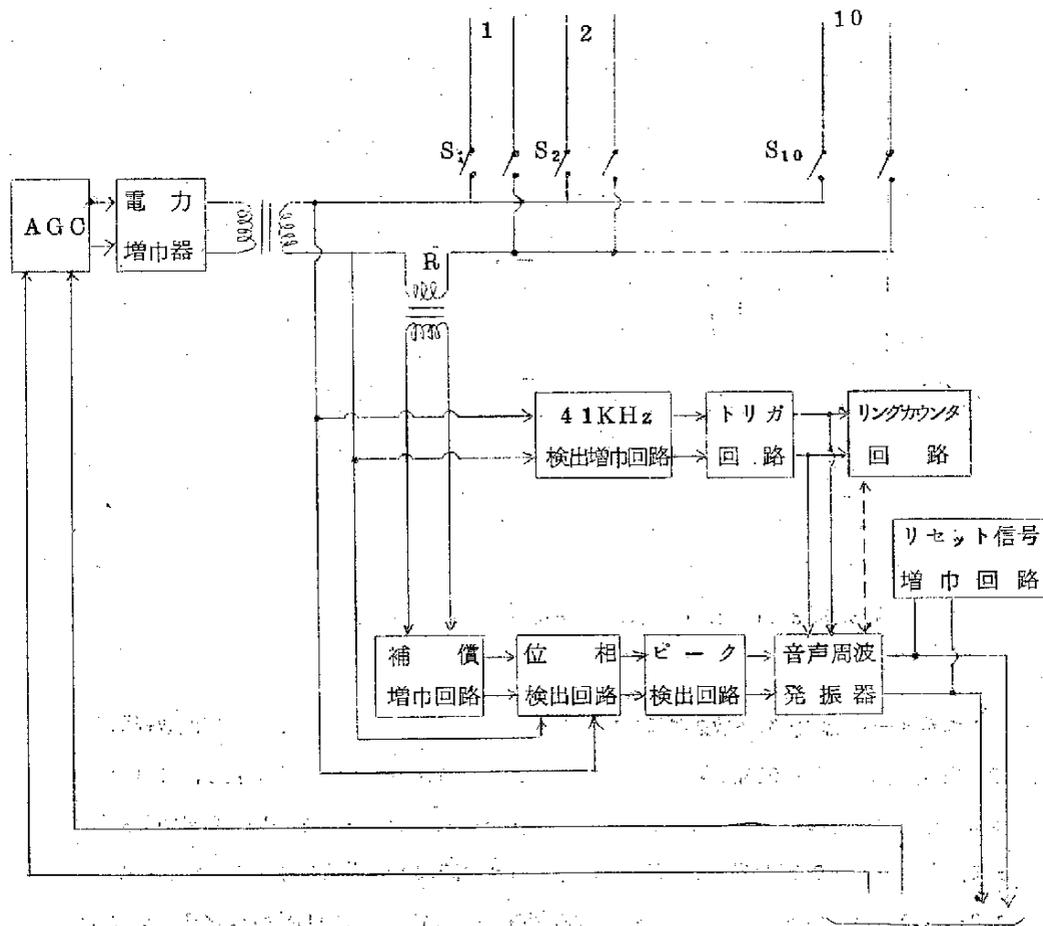


第 3.3.3 図 変換回路の共振特性

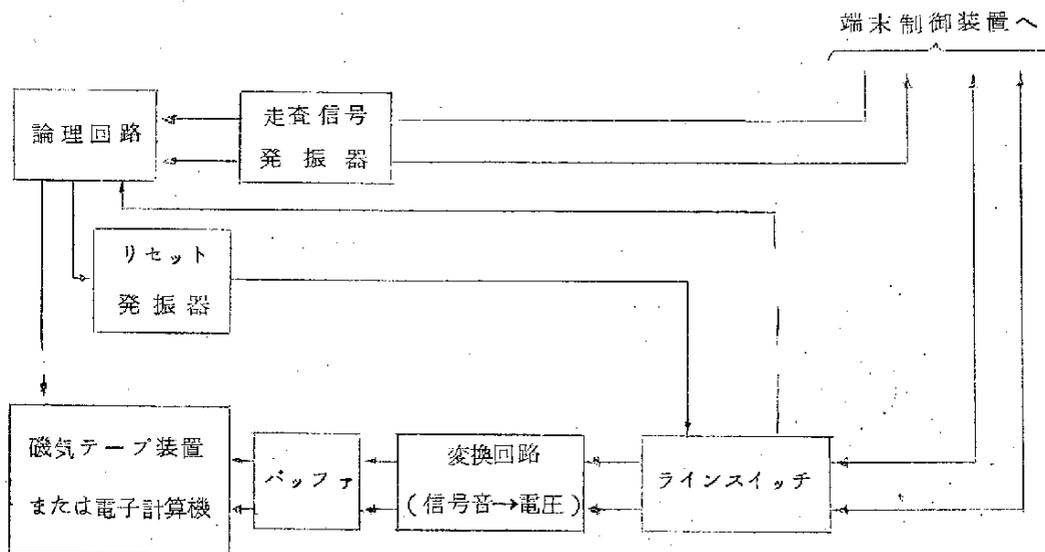
これがメータ読取線に多数並列に接続されており、検針センターはこれを周波数掃引信号で走査し検出する。走査周期は75秒で1回当たり100個のメータを走査し、これを10回の走査による10組計1,000個のメータの走査をもって1周期としている。従って、各メータは75秒間隔で走査され、1個当りの読取り時間は75mSとなる。この10組1,000個のメータは1個の端末制御装置を通して、ここから検針センターまで2対4線で導き計測情報を送受するよう設計されている。なお端末制御装置から検針センタ

一までは音声周波のAM方式 (amplitude modulation) により情報を伝送する。また、メータ読取走査用の周波数掃引信号として 3.5 KHz ~ 4.1 KHz を使用している。

これによって音声電話信号との漏洩による相互干渉を避けており、また場合によっては一般加入電話回線を共用出来る可能性もあると言われている。ただし、GE社での試験によれば、音声信号と掃引信号との分離フィルタがかなり高価になり、結局伝送路を共用しない方が経済的であるとしている。第 3.3.4 図に端末制御装置、第 3.3.5 図に検針セン装置のブロック図を示す。



第 3.3.4 図 GE 社方式の端末制御装置ブロック図



第 3.3.5 GE 社方式のセンタ装置ブロック図

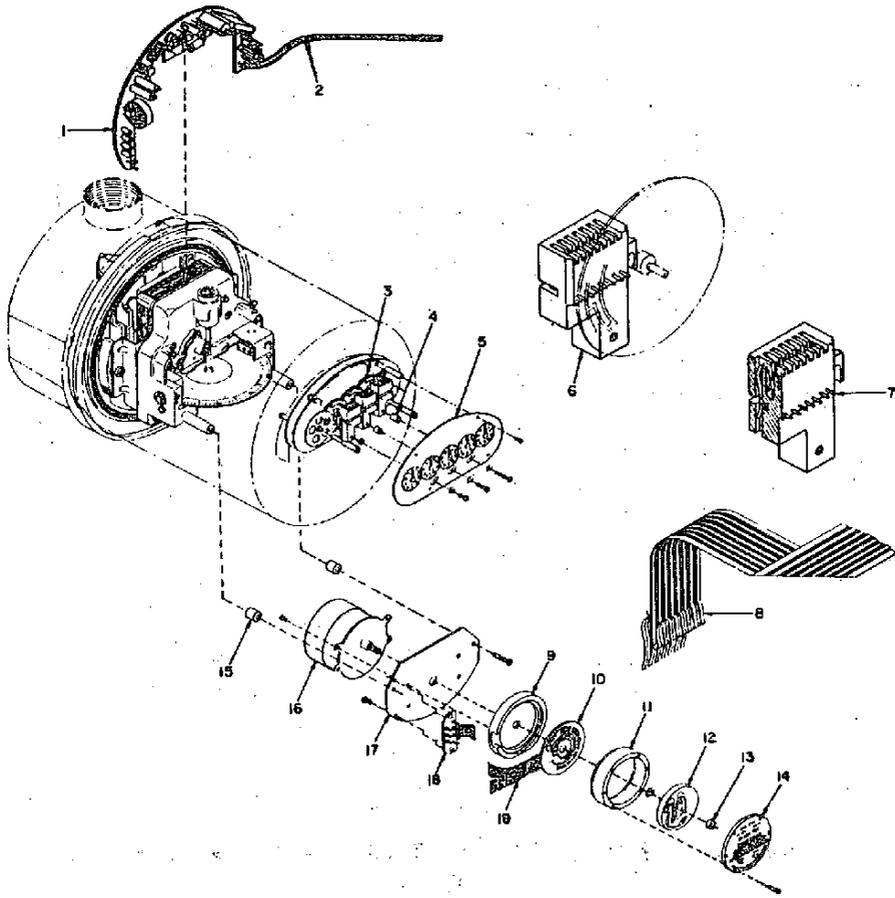
(2) 直読方式

この方式は計量値を示すメータの指針値をそのまま電気的符号に対応する形式でとり出し、さらにこれを伝送し易い信号系に変換して検針センターに送る方式である。

この方式では、1個のメータの検針に要する時間は高々数秒以下であり、しかも月1回程度の検針でよければ、特に専用線を設ける必要は無く、当然他の線（例えば、電話線）の利用が考えられる。

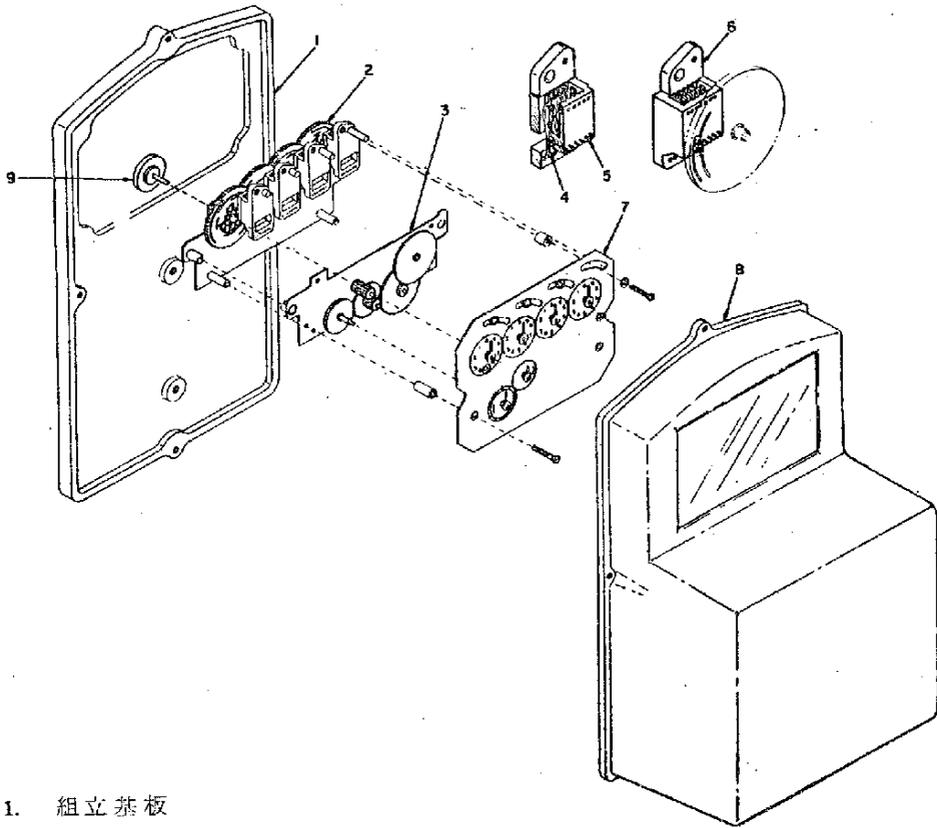
この場合、一般加入電話線を利用すれば、交換機による選択が可能となり、高能率の自動遠隔検針システムを構成することができる。しかし、メータ毎に積算値の読取機構とその読取値の送信装置を設備する必要があるため、この端末装置は比較的高価なものとなる。

この方式の代表的な例として、トランジテル社の方式がある。この方式ではメータの積算には第 3.3.6 図、第 3.3.7 図のように 4 桁の連動ギヤを使用し、これに直結したコード板によって積算値を 2 進の電気符号に変換し、検針時にセンター側から送出される起動信号によって、4 桁の積算値を順次 2 進符号で返送するものである。



- | | | |
|--------------|---------------|------------------|
| 1. 電子回路 | 8. リボンケーブル端子 | 15. スペーサ |
| 2. 電話線 | 9. プログラマ基板 | 16. モータ |
| 3. 符号変換板歯車 | 10. プログラマ接点板 | 17. 組立基板 |
| 4. スペーサ | 11. プログラマケース | 18. プログラマ端子板 |
| 5. 日盛板 | 12. プログラマブラシ板 | 19. プログラマリボンケーブル |
| 6. ピンホルダ | 13. ナット | |
| 7. 符号変換板接触ピン | 14. ネームプレート | |

第 3.3.6 Transitel 電気メータ

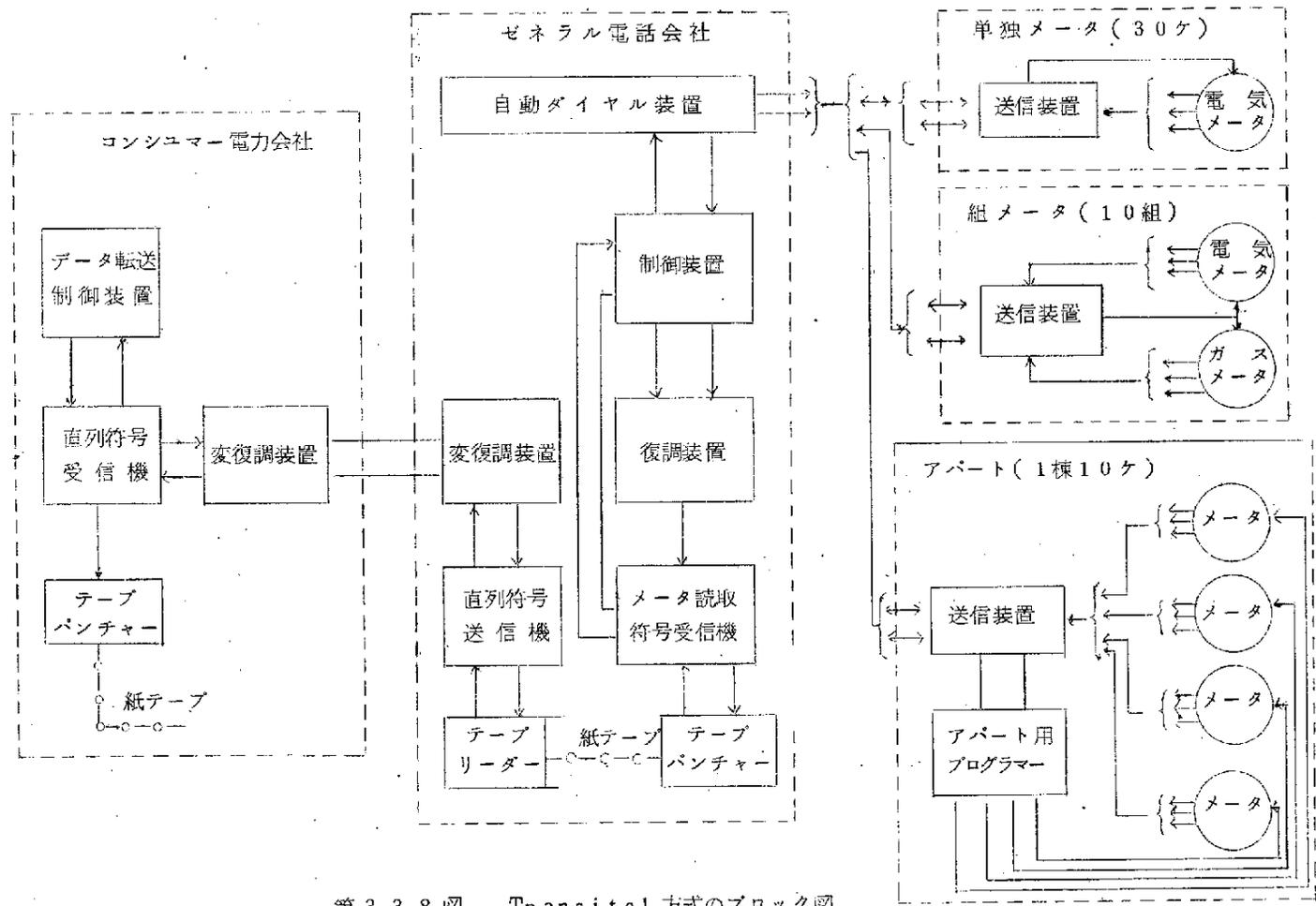


1. 組立基板
2. 符号変換板歯車
3. レジスタ
4. リボンケーブル接続ソケットバネ
5. 符号変換板接触ピン
6. ピンホルダ
7. 目盛板
8. カバー
9. 磁気カップリング用磯石

第 3.3.7 図 Transitel ガスメータ

この方式では電力量だけでなく、ガスなどの計量値もひとまとめにして、順次返送できるよう設計されている。また、アパート等のメータ密度の高い所では、1回の起動信号によって多数戸の検針が順次出来るようにも考えられている。1つのメータの検針時間は約1.5秒であり、各桁の符号は時間割で送られるが、この手段として「プログラマ」と称する多接点円板を機械的にブラシで回転走査する単純な方式をとっているのが特徴である。

このTransitel社の試験はALIT（自動回線絶縁試験装置）と称する装置、自動ダイヤル装置および電子計算機を組合せて、第3.3.8図の構成により一般電話回線を使用して実験が行なわれた。この試験の報告によれば、技術的には殆んど問題はなく、経済的見地からの検討のみがあとに残されたと言われている。



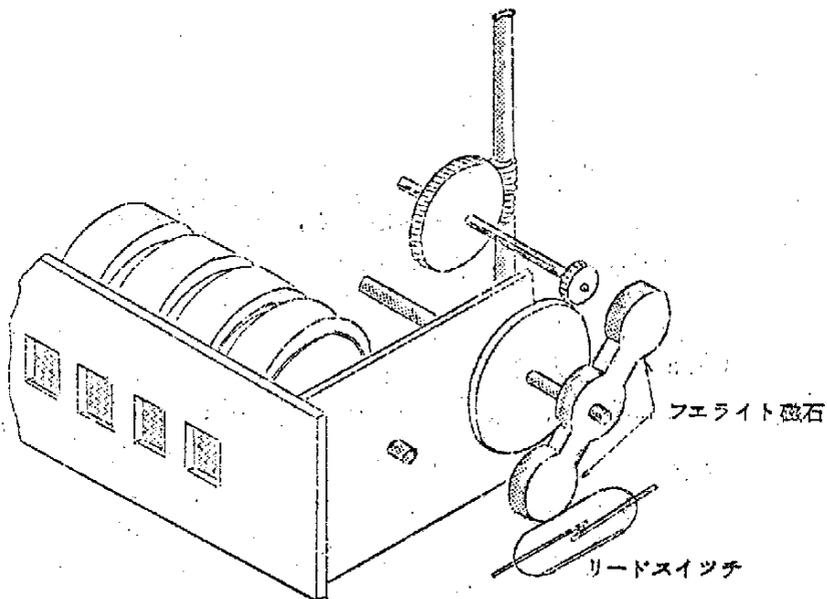
第 3.3.8 図 Transitel 方式のブロック図

(3) 中継方式

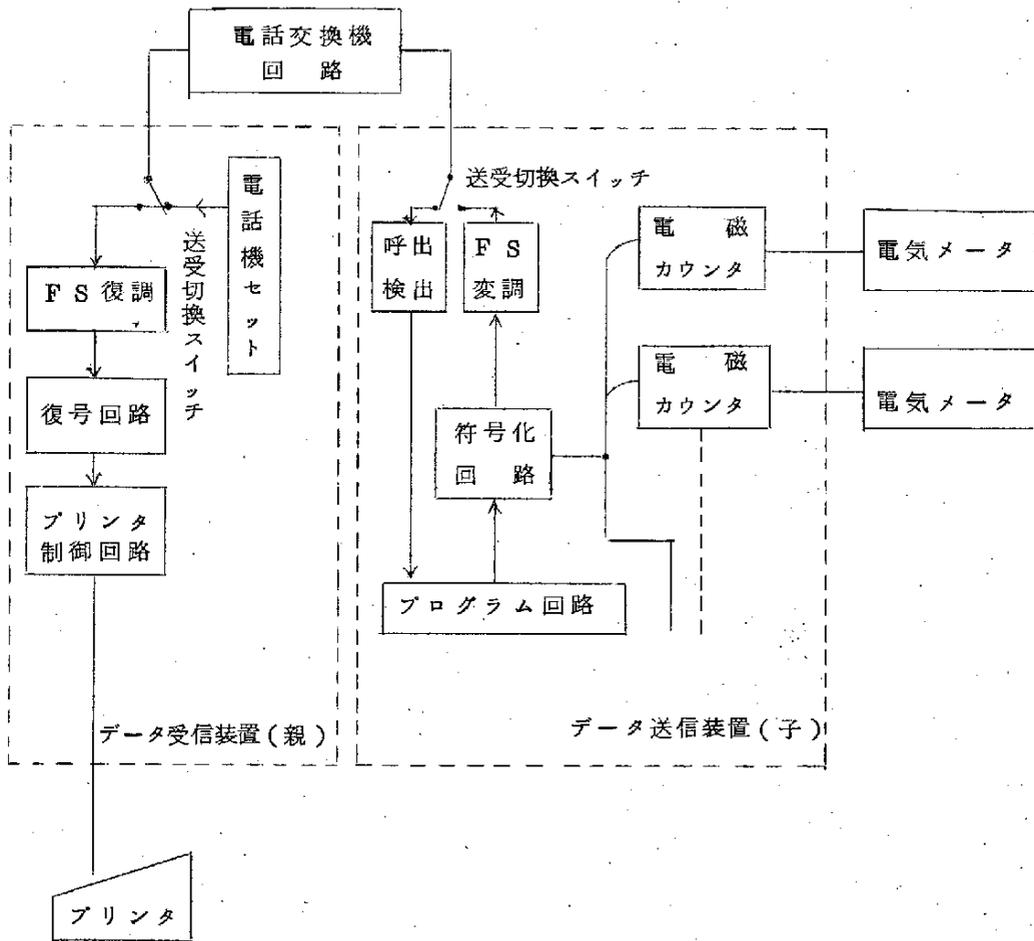
中継方式とは、メータからの単位読取の情報を一度中継点に送り、ここで積算した計量値を検針時にセンタからの要求によって返送し読取るもので、前記二者の混合された方式である。

この方式では走査方式と同様に簡単なメータが使用できるが、中継点に多数の積算用レジスタを集中して設置すれば、各戸のメータから中継点までそれぞれケーブルを引く必要があり、各戸が散在しているときは線材費が高価につく。また逆に、積算用レジスタを各戸に分散すれば、直読方式とほぼ同じ形態となり、この方式の利点はなくなる。

この方式の代表的なものとして、わが国の電気計器検定所の方式がある。第3.3.9図にそのメータ構造、第3.3.10図に方式のブロック図を示す。この方式では中継点は配電用変圧器付近となっており、同一配電系統にある10～20戸が一群とをって中継され計測されるようになっている。伝送路としては各戸のメータから変圧器までは計測線を電力引込線と並行に引き、中継点からは一般加入電話線を利用するようにしており、伝送方法はFS50ボーである。



第3.3.9図 電気計器検定所方式メータ



第 3. 3. 1 0 図 電気計器検定所方式

(4) その他

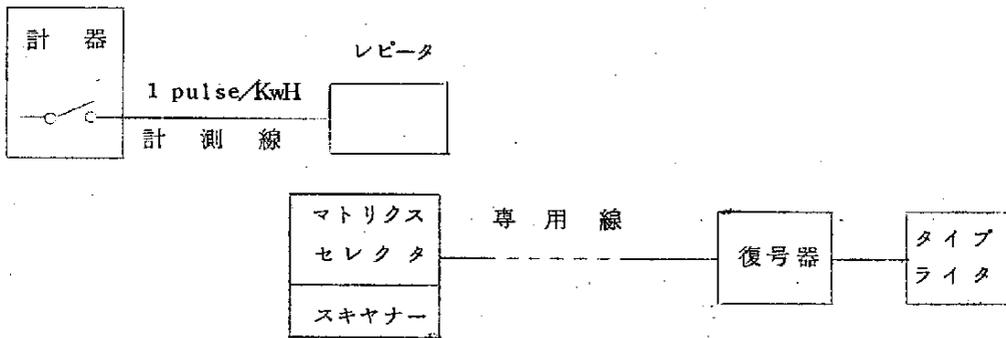
以上の他、配電線と電話線の混用等非常に特殊なものも考えられているが、いずれも前3者に類似のものか、混合形態のものとなるようである。

以上に例示した、検針方式の他、各国で検討されている種々の実施例について、次に簡単にふれておく。

(A) ロンドン電気局の各種方式

(a) レビータ方式

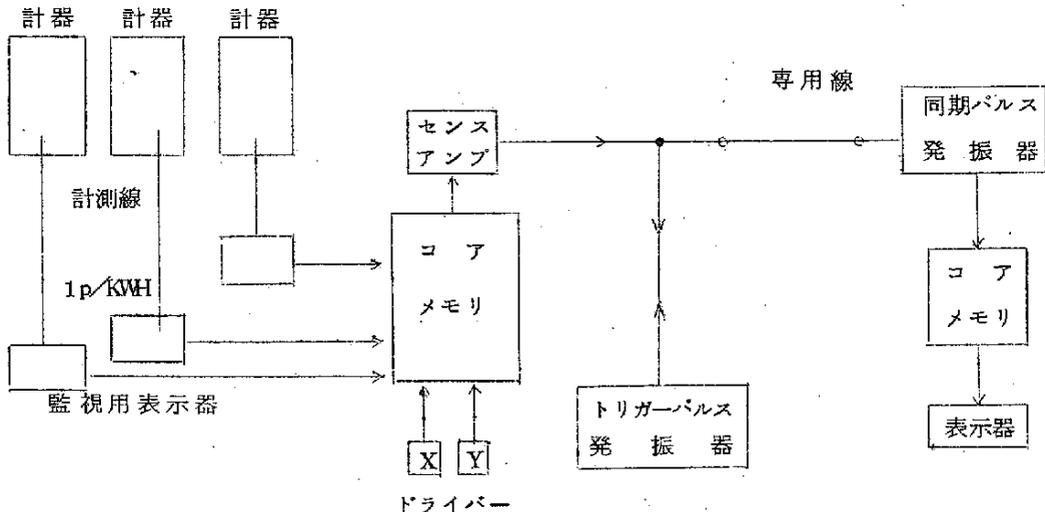
第 3.3.1.1 図に示すように、計器に機械的の接点を取りつけ、1 KWHにつき 1 パルスを送るようになっており、これをレビータ (ステップモータにより駆動される) に送る。レビータの表示機構はプリント回路のロータリー・スイッチ 4 桁を持ち、端子間に抵抗を接続して、表示数に比例した電圧をうる。これを専用線で送り、需要家番号とともに、復号器を通してタイプライタに印字する方式である。



第 3.3.1.1 図 レビータ方式

(b) 磁気記憶方式

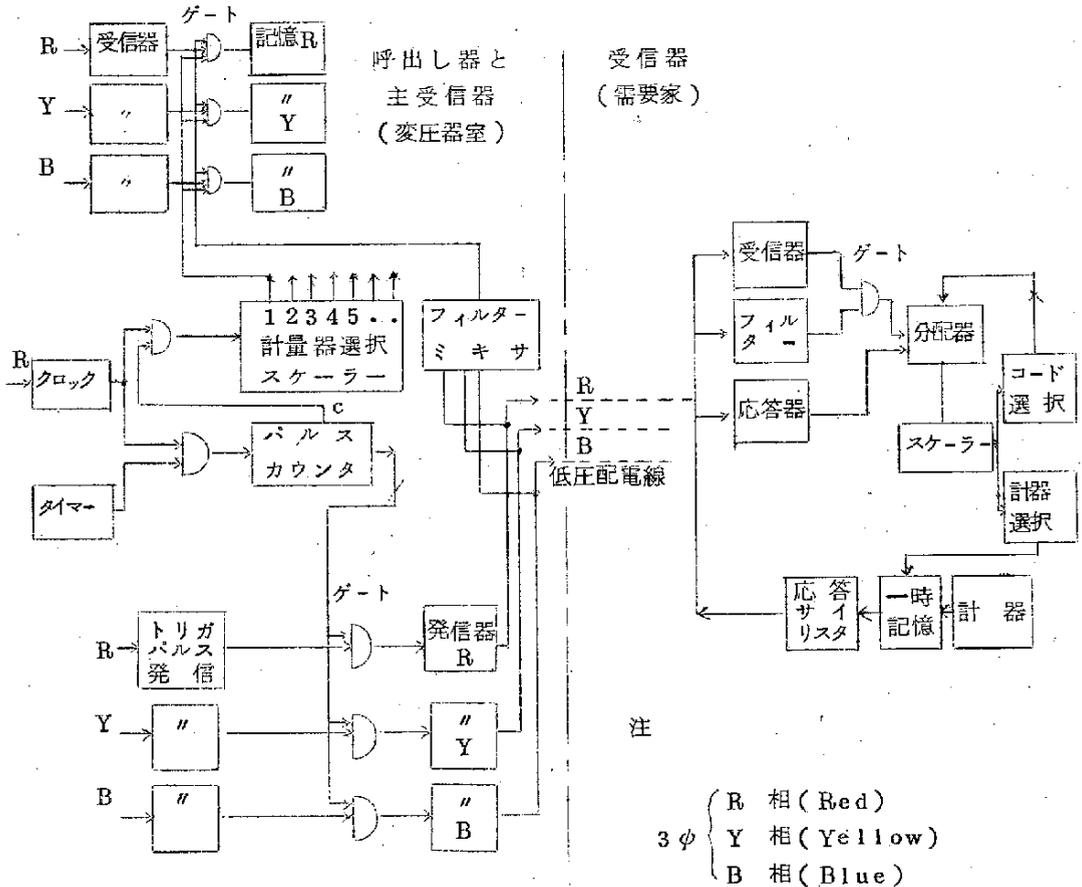
前のレビータ方式のレビータを個々に多数設備するかわりに、各計器からのパルスを磁気コアに記憶させ、これを走査して中央の磁気記憶装置に移し、さらに内部処理して使用量を取り出す方式である。この方式の概要を第 3.3.1.2 図に示す。



第 3.3.1.2 図 磁気記憶方式

(c) 配電線方式

電燈用の配電線を用いて家庭用のメータの検針をする方法も検討されていて、その方式としては、第3.3.1.3図のようなものがあり、技術的に可能であると言われている。



第3.3.1.3図 配電線方式

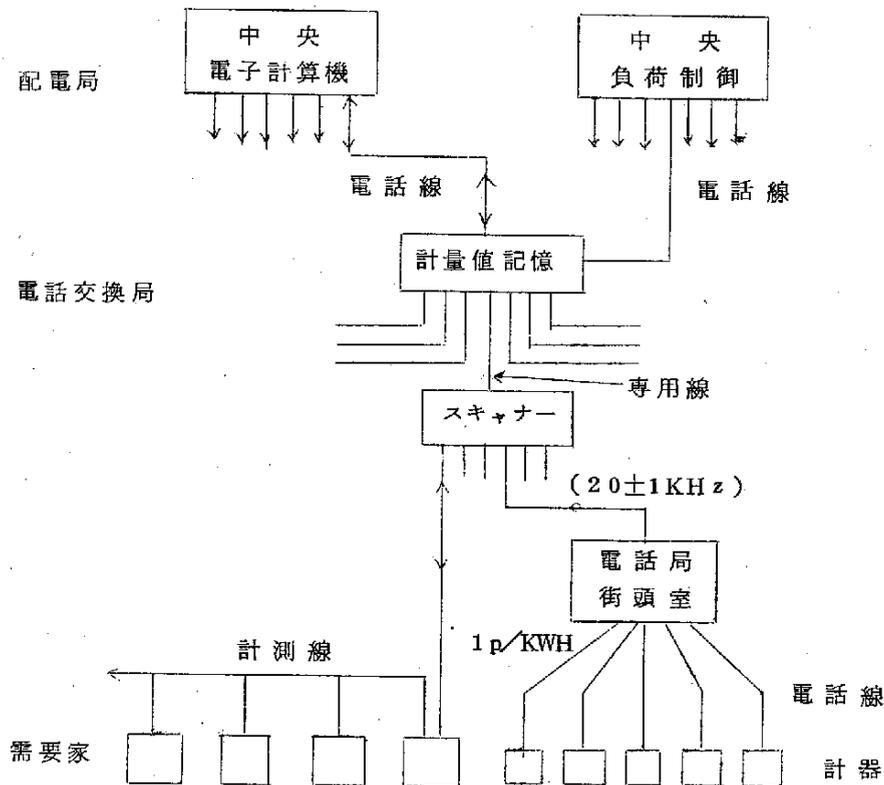
第3.3.1.3図では、変圧器室の計量器選択スケアラが起動すると、これに同期して、需要家のスケアラが動作し計器を選択する。計器は1 KWH単位の情報を送信し、それぞれの記憶回路に受信される。命令の同期はすべて3相の50 Hzを使用している。

(d) 負荷調査・制御兼用方式

需要家のメータの検針とその負荷の調査およびその制御をも兼ねた方式が検討されている。このために、1 KWHごとのパルス発信接点は30秒ごとと20 ± 1 KHzの周波数帯を使って走査され、一種の小形コンピュータに計量値を記憶させ、それを中央の電子計

算機で制御し、また中央の負荷制御装置でコントロールする方式である。

計量値の記憶は、料金計算のため電子計算機の指令に応じてデータを送出するほか、そのデータにもとづき中央負荷制御の指令を30秒ごとに計器受信器に逆相して、負荷を制御することができる。また、さらに走査中に試験用信号を送出して、計器の接点を強制的に開閉させ、装置を検査し事故計器を検出することができると言われている。この方式の概要を第3.3.14図に示す。

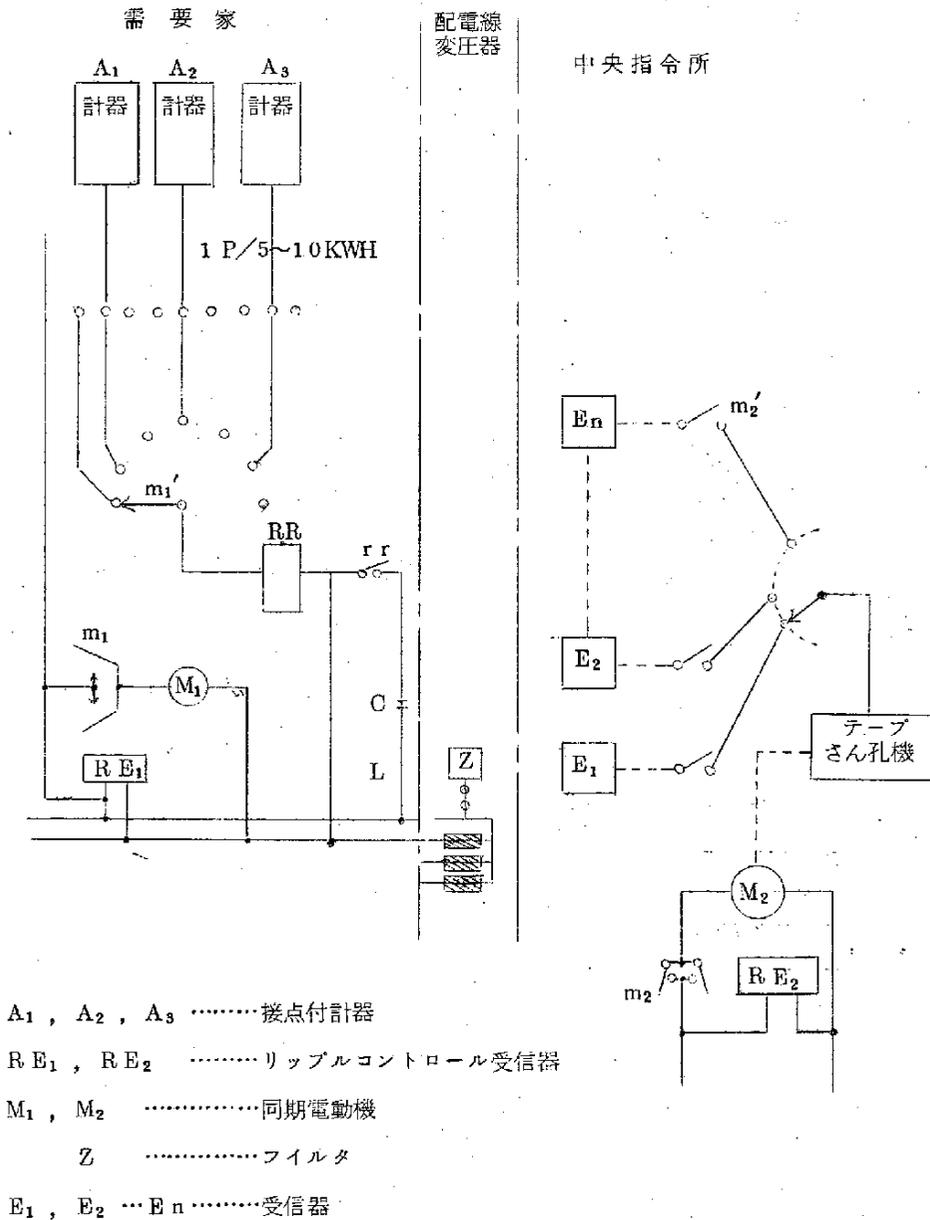


第3.3.14図 負荷調査・制御兼用方式

(B) スイスのMetragyer方式

需要家のメータには5~10KWHで動作する接点がつりつけてあり、中央指令所の信号でRE₁、RE₂(リップル・コントロール受信器)が動作すると、M₁、M₂(同期電動機)が回転し、ワイパーm₁、m₂を動作させ、計器の接点を読み出す。接点がOnの場合はRR(25Hzで機械的に共振する)を励磁し、その接点rrを閉じLC共振による575Hzの信号をフィルターZを通じて中央指令所の受信器に送り、m₂'による同期で紙

テープにさん孔するようになっている。この方式の概要を第 3.3.15 図に示す。



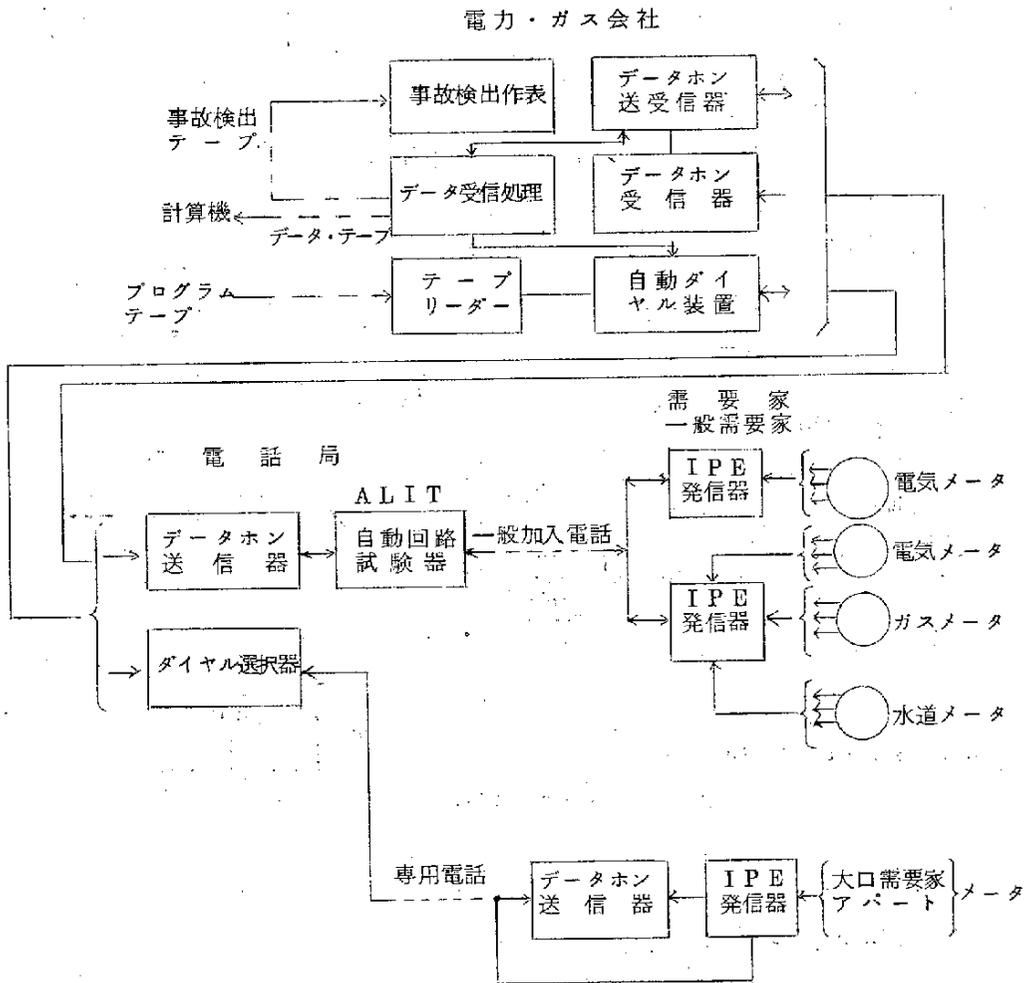
第 3.3.15 図 metragyr の方式

米国における自動検針方式

TransitelとGEの方式はすでに説明したので、ここでは他の一、二の例を示す。

(a) IPE-ATT方式

IPE (Industrial Process Engineers Inc.) およびATT (American Telephone & Telegraph Co.) によって開発された方式で、その中継方式を第3.3.16図に示す。



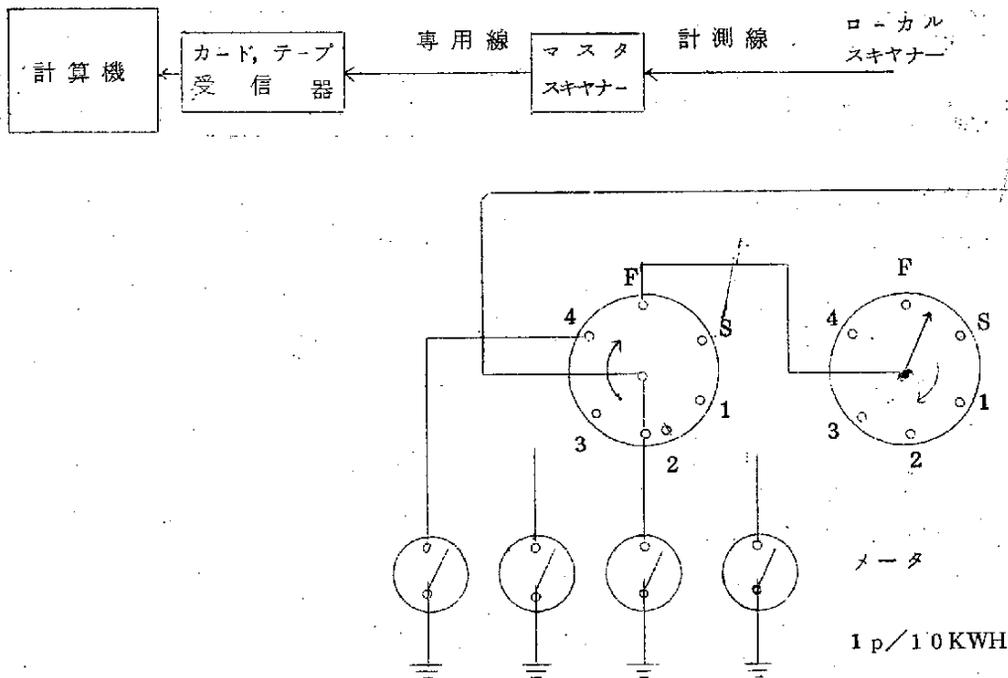
第 3.3.16 図 IPE-ATT方式

電気、ガス、水道メータの指針軸に、円筒形A-D変換器をとりつけ、6ビットの信号を計量装置から直接に取り出す。これをIPE発信器で変換し、加入電話線を利用して伝送し、データフォン送信器を通して、電力、ガス会社の計算センターへ送るようになっている。

また、大口の電力、ガス需要家およびアパートの場合は専用電話線を用い、一般需要家には既設の電話線を用いて、電話交換局からダイヤルで呼び出し検針することになっている。

(b) Westing house の方式

各メータからのパルスを時分割的に多重読み取りを行なう方式で、計器の符号器は10KWHごとに開閉する接点である。この方式は第3.3.17図に示すように、各メータ、ローカル・スキャナ、マスタ・スキャナなどで構成されている。各計器の状態は専用線により、中央の受信器に送られ、カード、紙テープ、磁気テープなどに入れられ、電子計算機にかけて処理することになっている。



第3.3.17図 Westing house 方式

各種方式の比較

以上、種々の方式について述べたが、これらの代表的な走査方式、直読方式、中継方式の3方式について、その特徴を比較すれば、次の表のようになる。

各方式比較表

	走査方式	直読方式	中継方式
メータ構造	簡 単	複 雑	簡 単
端 末 装 置 (含中継装置)	不 要	低 価・多 数	高 価・小 数 (中継装置)
端 末 線 材	小	中	大
信 頼 度	や や 劣	良	良
検 針 時 間	常 時	短 時 間	短 時 間
電 話 線 利 用	不 可	容 易	や や 困 難
セ ン タ 装 置	複 雑	簡 単	簡 単

3.2.3 伝送系（伝送路と伝送方式）

遠隔検針に使用する伝送路としては、電話線によるものと配電線によるものと大きく分けられる。配電線によるものとしては、50Hzを用いたロンドン電気局の例が知られているが、未だ検討の域を出ていないようである。

電話線によるものには、専用線と、加入電話回線とがあり、各種の方式に採用されているが、これは使用法によってさらに

- (1) 専用線のみによるもの
- (2) 加入電話線のみによるもの
- (3) 混合して用いるもの

に分類される。

また、特に配電線と電話線の混用ということも一部では考察されているが、これは未だ検討の域を出ていない。

データ伝送に用いられる一般の方式として、周波数分割による並列伝送方式と時分割による直列伝送方式とが考えられている。これら両方式の優劣についてはすでに多くの検討がなされているので、ここでは一般的な比較検討は省略し、これを遠隔検針に利用する場合、特に端末装置の回路技術の面、コストの面、信頼度・安定度の面等への影響について述べる。

(1) 並列伝送方式

これはいわゆる多周波信号(MF信号)によるデータ伝送方式であり、信頼度を上げるため普通2周波以上の組合わせが使用される。

従って、端末装置としては少くとも数個の周波数の中の任意の2周波を同時に発生し得る相当安定な発振器が必要であり、またこの周波数が検針データにより簡単に制御できるものであることが要求される。この条件に適したものとしては、最近開発された押しボタン・ダイヤル電話機のダイヤル符号発振回路がある。

これは低群4周波と高群4周波の中のそれぞれ1周波ずつから、2周波を組合せたダイヤル数字相当の送出回路が、わずかに1個のトランジスタにより構成されており、小電力で高安定度の発振器としてデータ伝送用にも十分使用に耐えるもので、これの遠隔検針への応用が考えられている。

この方式による端末装置は直列伝送方式に比して回路が非常に簡単であり、また一応の誤データ検出機能もあるので、経済的で信頼度の高い端末装置を開発できる可能性をもった伝送方式である。

(2) 直列伝送方式

この方式では音声帯域内のAMまたはFM信号による符号が考えられるが高信頼度を目的とする場合はFM信号が使用される。

この場合端末装置としては相当複雑な回路構成となり、従ってコストも高くなる。非常に大雑把にみてもMF信号方式の2~3倍はかかるものと思われる。しかしそれだけに信頼度は高く、誤り率、安定度などはMF信号方式より良好であると考えられ、またデータ伝送速度もやや速いものが期待できる。

すなわち、直列伝送では符号が直接パルスの形で伝送でき、中央処理装置が特に符号変換の必要がなく、また誤信号もパリティ・ビットの付加によって検出することができる。

しかし、端末装置において、直読方式を採用するかぎり、指針値の符号変換装置、パリティ・ビット付加回路等が必要になるので、並列伝送方式の端末装置より簡単なものは得られないであろう。

結局、端末装置として、直列伝送の場合は符号変換のデジタル回路が必要であり、並列伝送としては多周波発振器が必要となる。

これらを中心として、これの付属回路を考慮して選定しなければならない。

3.2.4 メータ読取機構

積算メータの計量値をどのように蓄積するか、また、これをどのように電気信号に変換するかには種々の提案があり、これが端末装置の方式にも大きく関係してくることになる。

現在考えられている案を大きく分類すれば次のようになる。

従来のメータに直接電氣的読取り機構を設けたものを直読型と言い、直読型には各桁が連続回転するアナログ型と、断続回転するデジタル型とがある。またこの場合、読取り符号として10進コードのままを取り出すものと、2進コードに変換して取り出すものと2種類がある。

また、これらとは別に、計量部と積算部を分離し、この間を電氣的に接続したものを分離型という。分離型では、一定の使用量ごとにパルスを取り出し、これを別のレジスタ部で積算するようになっている。

以下、これらについて説明する。

(1) 直読形式

(a) アナログ型

メータの積算値を読取るために、ロータリ・スイッチ型のもを直接各桁毎に取付け読取る方式のものであって、各桁の軸の回転は直接10:1のギヤで連結され、各読取接点は連続的に回転移動するブラシにより読み取られる。

この場合、各桁毎の軸回転は上位桁ほど回転がおそく、従って回転に要する力も総合的にみれば最下位桁を動かすための力が大部分であり、これに少し余分の力があれば全桁を動かすことができる。従って、読取り接点のための回転力により生ずる計量誤差も少なくすむ利点がある。

一方この方式では、ロータリ・スイッチの接点上のブラシの移動が連続的であるため、特に桁上りのとき大きな読取り誤差を生ずる危険がある。これを防ぐにはいろいろな手段がとられており、たとえば前述のTransitel社の場合は、各数字の接点間に中間地帯を設け、ブラシがこの地帯に入ったら一桁下位の数字を参照して数値を決定する方式をとっている。このような複雑な方法をとらない限り、そのままでは使用することが出来ず、これが一つの欠点となっている。

(b) デジタル型

アナログ型の桁上りのときの読取りのあいまいさを防ぐため、桁上りを瞬時に行なうデジタル型の積算読取機構がある。この形式は読取りはうまくいくが、各桁が同時に

桁上りする場合があります、その時に必要とするトルクは相当大きくなり、無視出来ない計量誤差となることが考えられる。特に、積算電力では取出し得る回転力は非常に小さく、各桁の文字車以外にロータリ・スイッチまで回転させるとなるとかなりの回転力が必要であり、許容値以上の計量誤差が生ずる。この点の解決が出来なければこの型は実用とはなり難い。

(2) 分離形式

直読形式の2つの方法は何れも計量値の積算を行なうのに機械的積算機構を使用する方法であって、計量装置そのものからのエネルギーにより積算するため、性能に一定の限度がある。分離型はこれと異なり、計量単位毎の情報を計量機構から電氣的にとり出し、積算は全く別の電氣的または電磁的カウンタにより行なうものである。従って、これの計量に対する影響は殆んどなく、また積算値を電気符号に変換するにもかなり自由な方法が採用できる。また、アパートなどでは多くのメータの積算を1カ所に集中して行なうことが出来るという便利さがある。

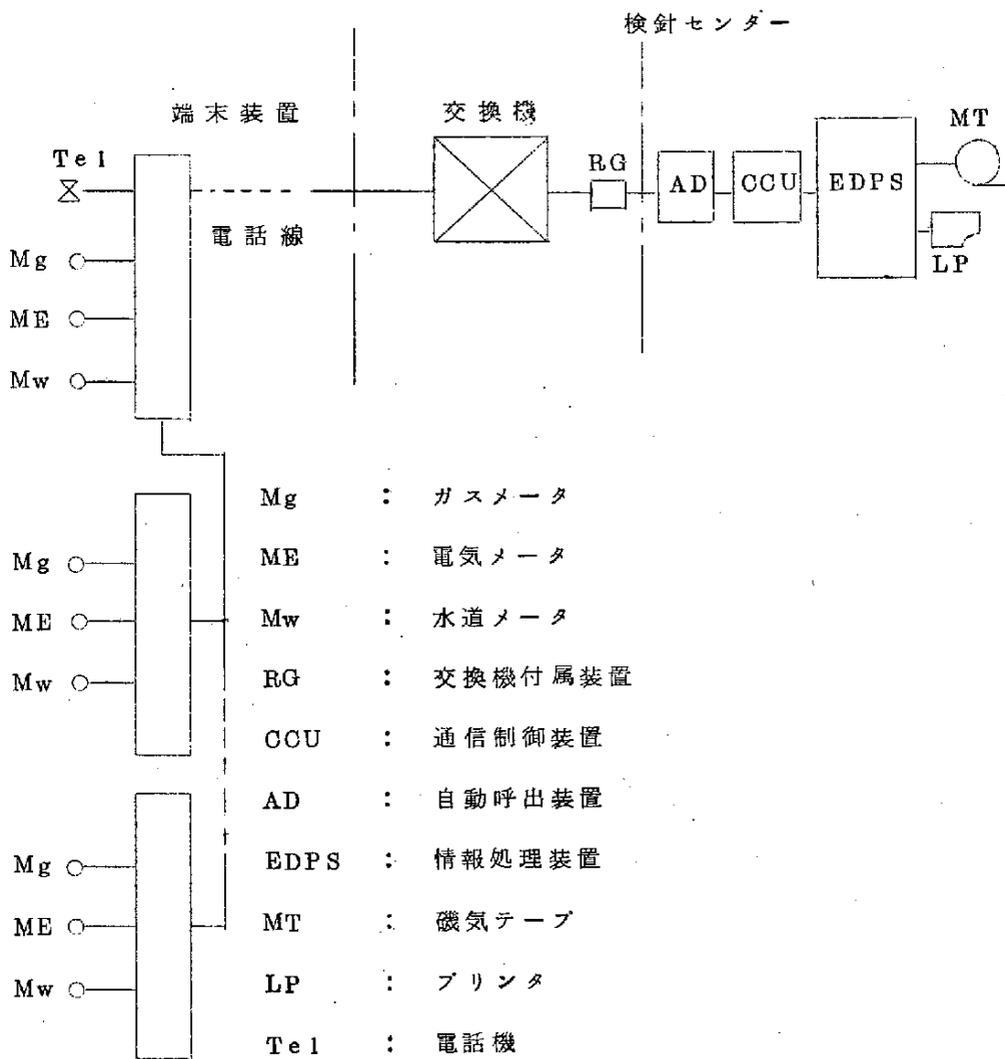
前述の送査方式および中継方式に使用するメータはこの分離型のものである。

この形式は、電気メータには比較的取り付け易いが、ガス、水道メータには固有の電源がないため、特殊の工夫が必要である。ガスメータでは回転トルクを利用した発電形式のものもあるが、現在では未だ高価である。

すなわち、この方法は積算装置がやゝ高価になるため経済性に欠ける難点があり、また、大部分が電氣的に積算するため、商用の100V電源を利用しようとするれば、停電時にガス、水道などの積算ができないことになり、積算値そのものも停電によりリリースされない方式でなければならない。

3.2.5 NTT方式の概要

日本電信電話公社(NTT)では、加入電話回線を全面的に活用した自動遠隔検針方式の検討を進めている。この方式の概要を第3.3.18図に示す。



第 3.3.18 図 N T T 方式

この図において、検針センターには電子計算機、通信制御装置、あるいはこれと同じ機能をもつ情報処理装置を設置し、自動呼出装置を電子計算機側から制御する。自動呼出装置は電子計算機のプログラムにより設定された加入者電話番号を受信し、これを一時蓄積して、交換機側（A形、H形またはクロスバの既設設備を活用）へインパルスを送出して、目的とする加入者の電話回線に接続する。接続後、電話機のベルを鈴動させたいためENQ信号（2周波の組合せにより構成）を送出して、電話機の通話回路をガス、電気、水道のメータ側へ切替える。このために端末装置が取り付けられている。各メータには各指針に相当する接点を取り付けられていて、この指針値をデジタル信号（多周波により構成）に変換して中央の検針センターに送り込む。中央の検針センターでは、この符号を順次受信して、誤信号のチェック、前回指針値との比較、差し引きなどの簡単な演算処理をした後、磁気テープ等のファイルに書き込む動作を行なうことになっている。ガス・電気・水道の1軒分の処理時間は、交換機の接続時間を除けば、1.5秒～2秒程度であることが確認されているが、目下実験を続行中である。

3.2.6 む す び

以上、自動遠隔検針の各種方式の概要について述べてきたが、この他にも、ここでふれなかったSystemが存在する。しかし、いずれも宅内装置、伝送路、読み取り形式について類似のものが多く、実験的には、一応技術的な可能性が確認された段階であり、未だ実用されているものは皆無である。

電話線以外の方法では、いずれも伝送路のコストに問題があり、電話線を用いた場合でも、加入電話機との切替回路、端末機のコストにはいぜんとして問題が残されている。

各種のレポートによれば、各国の方式において、実験が成功していることが報告されているが、人手による検針の所要経費と経済比較をしてみると、現在の段階では多くの困難が横たわっていると言われている。

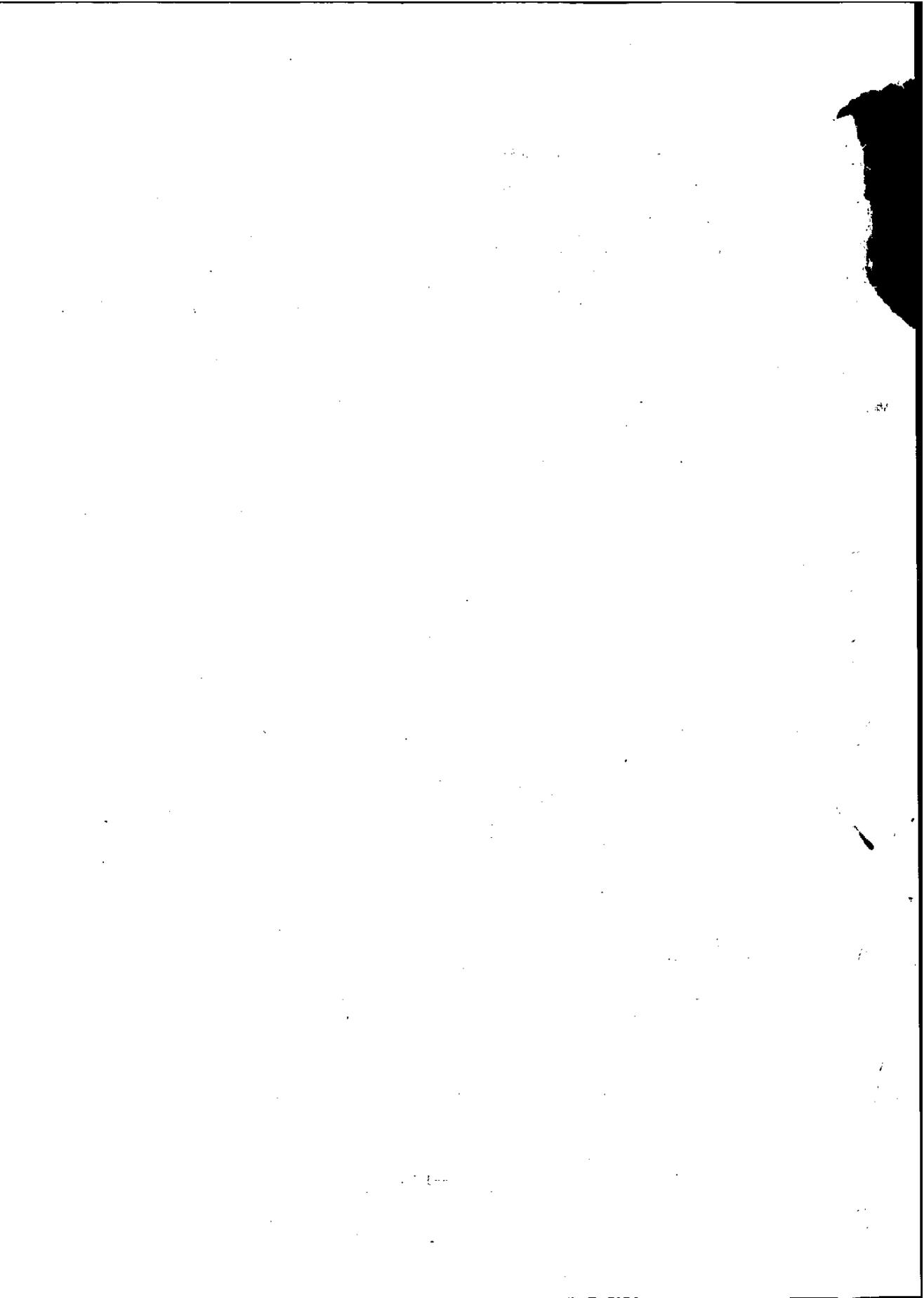
NTT方式について言えば、現在実験が進められている段階であるが、Systemとしての信頼性もあり、また、伝送路のコストにも問題はなく、ただ端末装置のコストが今後の問題として残るものと思われる。

これは、回路的な技術の改良はもちろんであるが、電子技術全般の発達に待つところが非常に多い。この点さえ解決するならば、NTT方式は最も有望な方式の一つであると言えることができる。

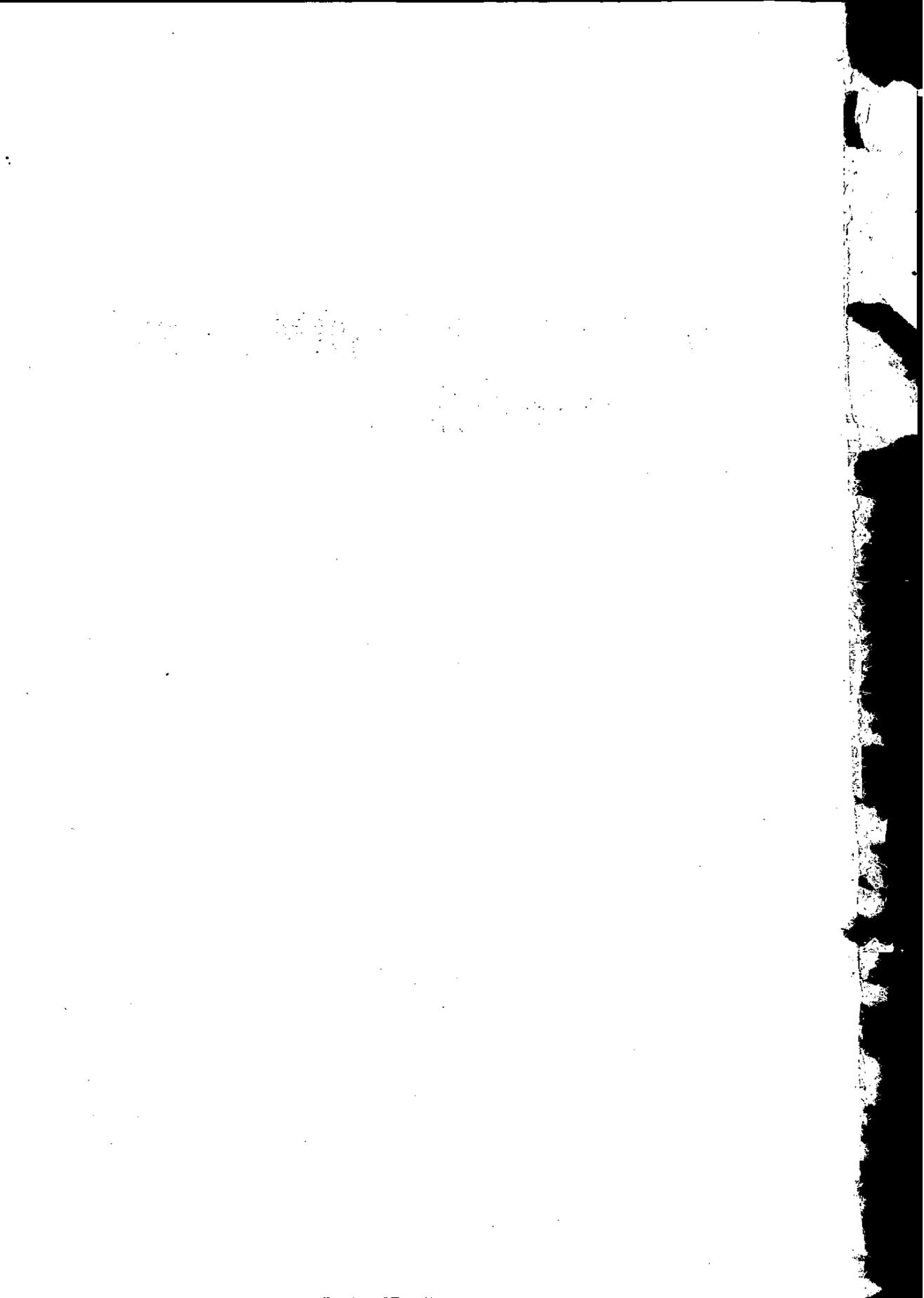
参 考 文 献

- (1) A. Brothman : IEEE Trans. Commun. Tech.
(1965 ~ June)
Automatic Remote Residential Meters
- (2) F. Stuart Rittenhouse and Roy A. Wells, JR.
: A. G. A. Operating Section Distribution Conference.
(1965)
- (3) G. E. Bell : Telephone Engineer and Management,
(May 15, 1966)
Remote Reading of Residential Utility Meters
- (4) 杉野, 柿沼, 古川, 関川 : NEC日本電気技報
(No. 85 1967)
積算電力計の自動遠隔検針装置
- (5) 古川, 関川, 梨子本 : NEC日本電気技報
(No. 83 1967)
集中デジタル テレメータ・テレコントロール・システム
- (6) Lufkin Research Laboratories Inc. :
自動メータ読取り装置に関する報告書
(昭和42年3月)
- (7) Martin W. Hellar, Jr : Electrical World
(1961 Dec)
Automatic Meter Reading
- (8) Jack Davey, C.A. Campbell, G.O. Mietz : Electrical World
(1963 May)
Meters Remotely Read On 80 - House Loop
- (9) A. Brothman, R.D. Reiser, N.L. Kahn, F.S. Rittenhouse,
R.A. Wells : Conference Paper (1965, January 31 -
February 5)
Automatic Remote Reading of Residential meters

- (10) 渡辺 , 杉野 : 計測と制御
第6巻 第5号 (昭和42年5月)
積算電力計の自動検針の動向
- (11) 渡辺 : OHM (昭和42年7月)
電話回線を利用した自動遠隔検針装置を開発



4. オンライン文献検索システム
; JOLDOR



4. オンライン文献検索システム；JOLDOR

4.1 まえがき

インフォメーション・リトリバル (information retrieval; IR) に対する要求の一つは、何時でも必要な時に必要な情報が、すぐ手に入るということであろう。

文献検索に関しては、時間的緊急度は一般にそれ程必要ではないかもしれないが、社会活動のスピード化にともなつて最近は即時性がより尊ばれる傾向がある。更に文献検索の常として、一度で目的の文献を探し出すことはかなり困難であつて、検索結果を手がかりとして繰返し要求を出し、最終的に目的にかなうものを取り出すという、いわゆる試行錯誤の行動をとることが多い。また、膨大な検索ファイルを各所で保持して行くことは経済的にも困難であるから、どこかに集中管理したものを、多人数が共同利用するのが望ましい。

それらの点を総合してみると、文献検索を含めて、一般に IR システムは本質的にオンライン向きの仕事ではないかと思われる。

その上、IR の仕事は、膨大なファイルを対象とした検索作業、あるいは入出力と CPU タイムのバランスなど、計算機内部の処理の効率化という面からすると、IR 作業自身の内部でも、あるいは他のジョブとの間でも、いづれにせよマルチ処理が不可欠となる。

とくに、いくつかの端末からの同時アクセスという事を考えるとタイムシェアリング・システムの下で動かすということが必然的な要求となってくる。

JOLDOR (JIPDEC On-Line Document Retrieval) はキャラクター・ディスプレイを端末としたオンライン文献検索システムであり、FACOM 230-60 の MONITOR-V の下で働かせる。これは当センターで計画中の遠隔情報処理システムの一環として開発したものである。

4.2 オンライン・システムの特長と JOLDOR の設計方針

オンライン文献検索システムに要求される項目はいくつかある。

- (1) 検索時間が早いこと。
- (2) 検索結果をみて、検索範囲の拡張、縮小、変更等の検索要求の繰返しがり易いこと。
- (3) 会話形的要素を持っていること。
- (4) コマンドの指定が簡単で、素人でも容易に使えらること。

- (5) 端末の使い方が人間工学的に考えられていること。
- (6) ガイダンス(guidance)の機能があること。
- (7) Comfortable なこと、わかりやすいメッセージ、待たされない、など使い心地がよいこと。
- (8) 端末から新たな文献の追加や、ファイル内容の変更ができること。
- (9) アウトプット内容の指定ができること。
- (10) 必要に応じて検索結果のハード・コピーがとれること。
- (11) アウトプットの多い時は、セントラルのライン・プリンタや磁気テープが利用できればなおよい。
- (12) ユーザ・ファイルの形で中間または最終検索結果がテープやディスクにセーブ(save)され、後日利用できること。
- (13) 他のジョブとマルチ処理ができること。

これらの項目は、ある程度、お互いに矛盾もあり、理想に近いものもあるが、一般にIRシステムは

「標準的な機能はなるべく簡単に指定できて素人でも容易に使用できること。そして熟練した利用者のためには、かなり複雑な機能も用意されていること」というのが条件なのだそうである。

初心者にも容易に使えらるということは、他の会話形プロセッサもほぼ同じような目的を持っているが、文献検索では更に加えてガイダンスの機能があることが望ましい。

たとえばコマンドの種類や機能を、オンラインでシステムが教えてくれる。また、シソーラス(thesaurus)にもとづいて、類似または同義のターム(term)や、上位、下位のタームを示してくれたり、検索質問に該当する文献の数が、予め表示されたり、というきめこまかいサービスがのぞまれる。

一方、オンラインであるからは、大なり小なりリアルタイム的要素を持っているので、検索時間の短縮ということが、どうしても必要となってくるが、多くのバッチ処理のシステムは、そのために数個から数十の検索質問をまとめて処理し、少くとも一つのファイルから採り出すべきものは、まとめて一度にとり出してしまおうという方法をどっている。どむるがオンラインでは、それは不可能で、やはり一つ一つ処理せざるを得ない。そこでファイル構成をどうするかが問題になる。ファイルのアップデートという面から考えると、サブファイル(subfile)や

ンバーテッド・ファイル (inverted file) をいろいろと作りたくなる。このへんは、そのシステムの利用条件から両者の妥協点をみつけねばならない。そしてそれにはメインメモリー、ドラム、ディスク、テープ等の使いわけというものがからんでくるが、実は何といっても一番大きな影響は、オペレーティング・システム (operating system) との関係である。オンライン IR 専用のシステムならば、それなりの OS の設計方針があるだろう。しかし一般の TSS の中に含めた文献検索システムでは、その OS が IR のファイルの特殊性をどこまで考えて設計されているかによって、システムの作り易さが全然ちがうし、逆に言えば、IR システムが検索時間を減らそうと、あれやこれやと細工をしても、大もとの OS のオーバーヘッド (overhead) がやたらと大きかったりすると、細かいことを考えるだけ損だということになる。

さて、以上のように、いろいろの問題点があるわけだが、この JOLDOR では、使用する機械の機能あるいは、さしあたりの利用目的からして、次のような点を設計目標および特長とした。

- (1) ユーザ との応答は 5 秒以内に行われること。検索に手間どる場合でも、現在処理中であることを旨、何等かのメッセージ (message) を出す。
- (2) ファイル設計は検索時間の短縮を最大目的とした。
- (3) ファイルの更新は、オンラインで、端末から随時行い、月に一回程バッチのアップデイトで乱れたファイルのディレクトリを整理する。すなわち、オンライン、バッチの併用で更新を行う。
- (4) 端末として、タイプライタとキャラクタ・ディスプレイ (character display) をペアにして使うこととした。ディスプレイで表示が一度にパッと出るという利点を生かし、必要なもののみを指定してタイプライタにハード・コピーをとらせる。またユーザが使用したコマンド・エコー (command echo) もタイプライタに全部プリントさせる。
- (5) 入力データは各文献につき次の項目を入れる。

標 題

著者名 (所属をふくむ)

文献目録 (誌名、巻、号、ページ、年代)

ディスクリプタ・ワード (必要ならば 20 個まで)

文献番号

- (6) 検索タームは自然語を対象とし、各文献の標題と附加されたディスクリプタ・ワードによる。
- (7) 検索タームにはAND, OR, NOTの論理演算をゆるす。
- (8) 現在のシステムにソーラスは入っていないが、将来適当なソーラスができれば、入れられるように設計してある。
- (9) バッチ用のKWOCシステムを使って、キイ(Key)の分布や長さ、リスト構造にした時のノード(node)数等の分析を行ない、その結果を充分反映させてファイルを設計した。
- (10) 端末での応答と、ファイルの探索をそれぞれ別のタスク(task)とし、マルチ・タスク処理を行い、検索時間を早めている。
- (11) 他のジョブとマルチ処理が可能である。
- (12) 2つ以上のユーザから同時の要求に応じられるように、このシステムはリエントラント(reentrant)に作られている。
- (13) 年号、書誌名等により検索範囲をしぼることができる。
- (14) ヒストリー・ファイル(history file)を別に作り、検索統計をとり、システムの改良、ファイル内容のアップデート等の資料とする。
- (15) 収録文献はさしあたり、電子計算機関係の論文約2万件を対象とする。
- (16) ユーザの質問に応じて、コマンドの使い方をディスプレイする。
- (17) バッチ処理として、SDIサービスとKWOCサービスを行う。

4.3 システムの概要

4.3.1 機器構成

対象とする計算機はFACOM230-60で、そのMONITOR Vのもとで働かせる。MONITOR Vは、バッチ処理、デマンド処理、リアルタイム処理の3種類が取り扱えるオペレーティング・システムである。

JOLDORで使用する現在の機器構成は

主記憶装置	128KW(36bit/W)
磁気ドラム	2.56MByte 1台(KWTF及びワーク・ファイル)
ディスクバック	7.25MByte 3台(DIF, DSDF, ヒストリー・ファイル)
磁気テープ	60KByte/sec数台(システム, データの保存)

等であるが、端末装置として

F6221A ディスプレイ(190×140mm, 50字×20行, 128字種)と,

F791A タイプライタ(900字/分, 120字/行, 128字種)

を組み合わせて使う。

いづれにしろ, この機器構成ではTSSとしての容量が不足しているので, 主記憶容量, ディスク台数等の増強を考慮中である。特にライトペン付のディスプレイによる使用法も計画している。

4.3.2 基礎データの測定(FACOM230-50 KWOC システムによる)

各ファイルやテーブル類の大きさがある程度予測しておく必要があり, 手もとに在った7000~8000の文献資料をもとにして種々の測定を行なった。出来ることなら2~3万程度の文献データを測定したかったが現在の機器構成から言って高々1~1.5万文献しか収容出来ないので現システムにとって十分なものと思われる。

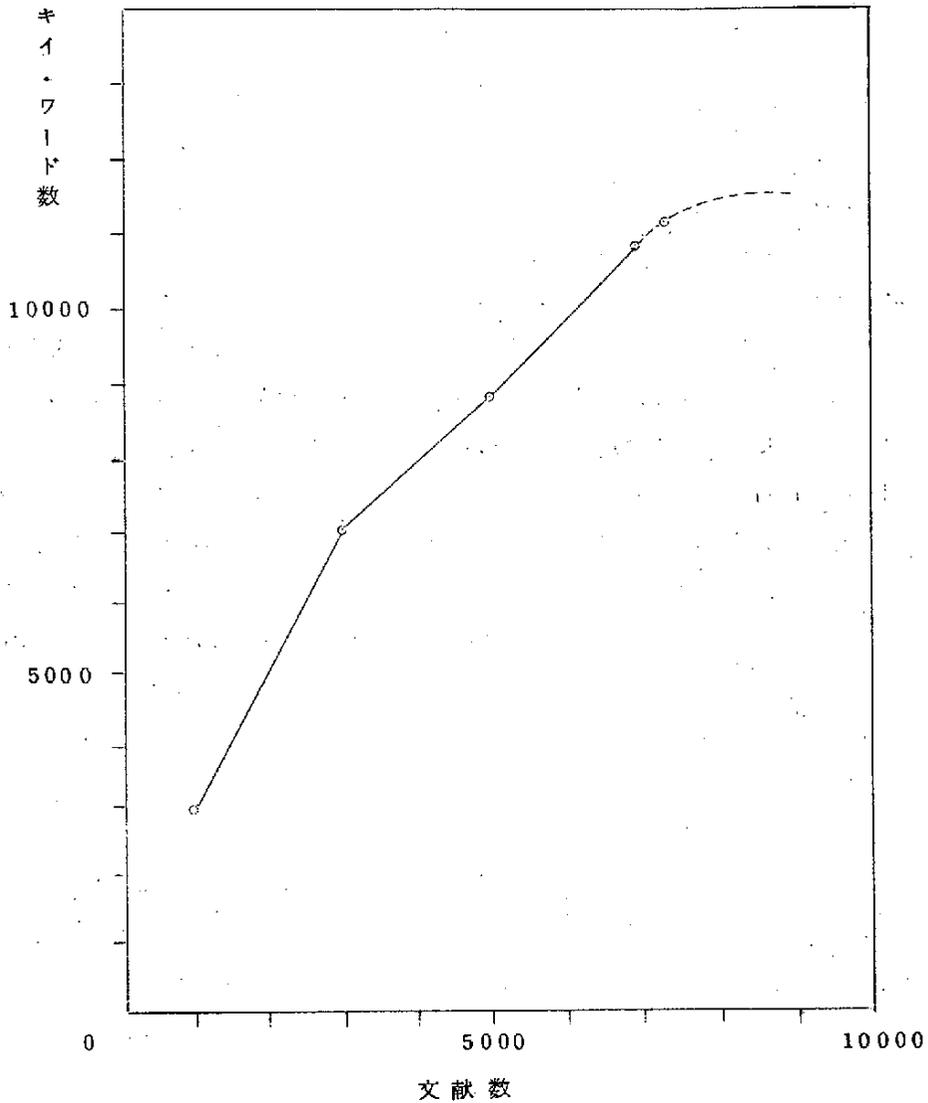
尚, 文献はほとんどが電子計算機関係のもので英文のものと一部独文のものが入っている。キー・ワードを抜き出す作業には既存のKWOCシステムの一部を用い, ノン・キー・ワード(non key word)として240ワード程指定した。またキー・ワードに関しては"ON-LINE"やPL/I"の様に"-"や"/"でつながっているものは1つの単語として扱った。

(1) 文献数とキー・ワード数

手もとに在る全文献7400からランダムに1000, 3000, 5000, 7000の文献を抜き出しKWOCシステムを使って, 文献数と異なるキー・ワード数との関係を求めたものである。

これによると7000文献のあたりで丁度キー・ワード数の増加率が落ちているのでさらに何千文献かあると良いカーブが得られたと思う。尚破線の部分は補間法により予測した。

文献数とキイ・ワード数



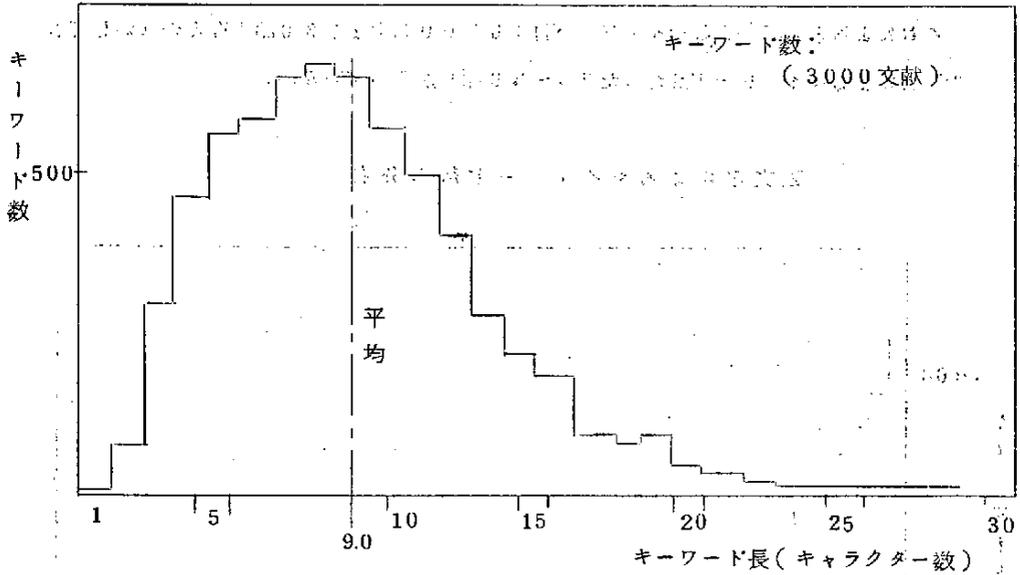
(2) キイ・ワードの長さ

このIRシステムに於てはキイワードとして最高30文字迄許しているが、実際にキイワードの長さがどの程度であるか知っておく必要があり、KWOCシステムを通して得られたキイワードについて長さの分布を求めてみた。

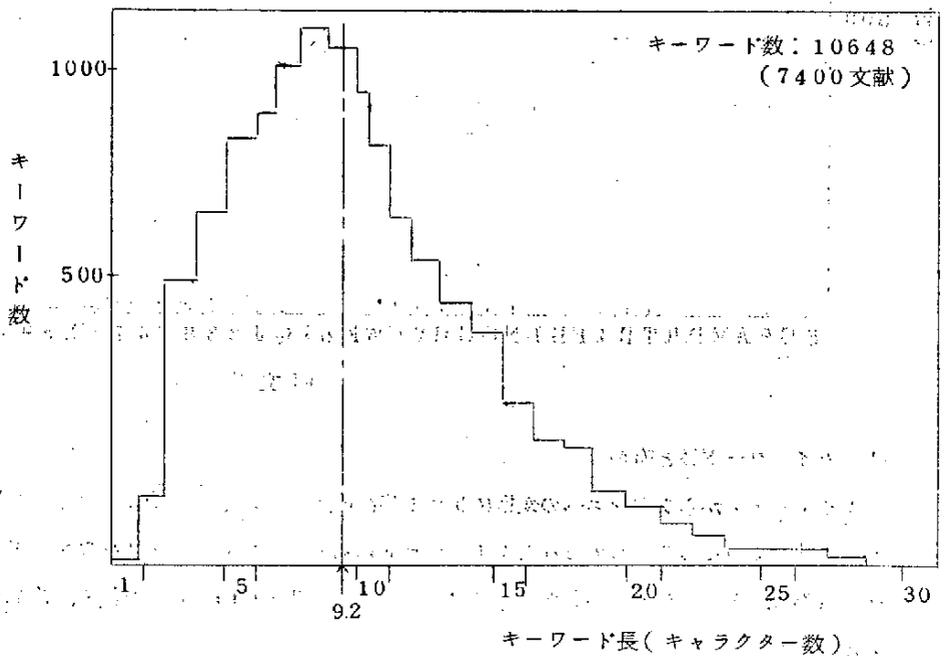
1つは3,000文献(キイワード6,100)、もう1つは7,400文献(キイ・ワ

ード10648)で測定したが、ほとんど同じ様な分布をしている。キーワードが4000~5000を越えると長さはほぼ一定の分布に従うと考えて良い様である。

キイ・ワードの長さの分布



キーワードの長さの分布

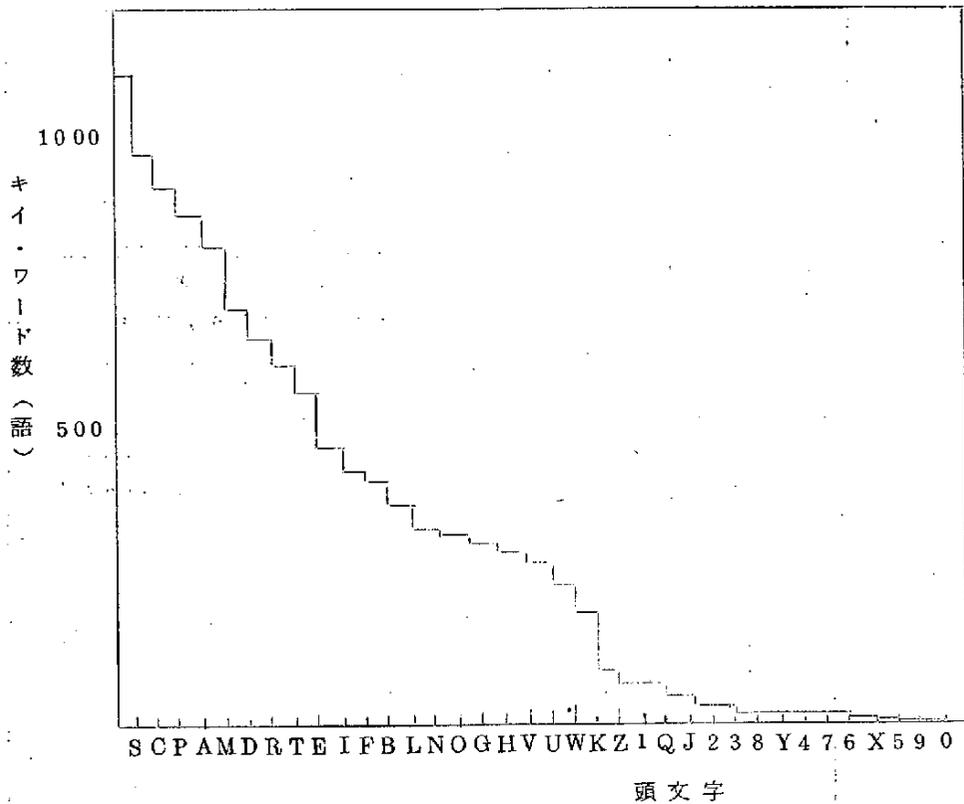


(3) 頭文字とキー・ワード

これはシステム設計に直接関係するデータではないが、頭文字ごとのキー・ワード数を数えた結果を次に示しておく。

これによるとSで始まるキー・ワードは10700語中1130語も在るのに対してXやYで始まるキー・ワードはたった20~30語しか入っていない。

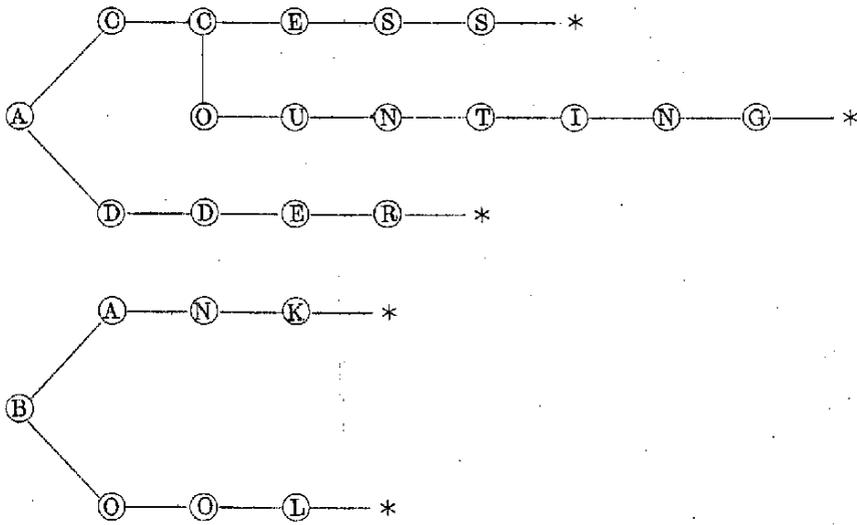
頭文字によるキー・ワード数の分布



(4) キー・ワード数と節数

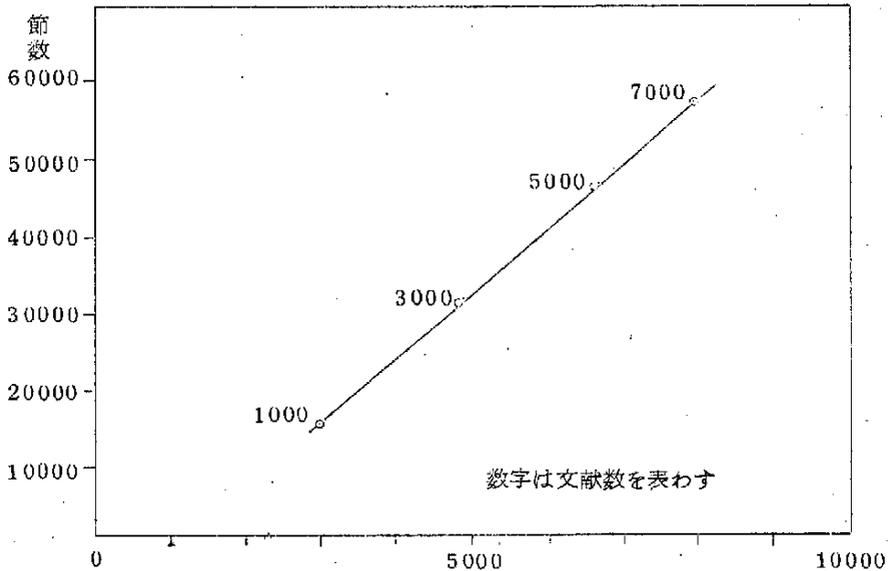
キー・ワードからアドレスへの変換にリスト構造のテーブル(次の節で説明するK W T F)を使っているため、与えられたキー・ワードに対するリスト・テーブルのサイズを予想する為に、キー・ワードの各キャラクタを節としてツリー(tree)を作り、節数を数えた。

例えば“ACCESS”, “ACCOUNTING”, “ADDER”, “BANK”, “BOOL”, というキー・ワードで作ったツリーは



となるが、節の数は*も含めて、この場合29と数えてある。

他の測定の時と同じ様に7400文献の中からランダムに1000, 3000, 5000, 7000文献を抜き出し、KWOOシステムによって得たキー・ワードをもとにして、それぞれツリー(tree)を作り節数を数えた。



(5) キー・ワードが文献に現われる頻度

1つ1つのキー・ワードがどのくらいの文献に含まれているかを調べてみた。

次の表は7000文献から抽出したキー・ワード11000のキー・ワードそのものと、そのキー・ワードを標題に含む文献数を表わしている。

全部のキー・ワードを示すわけにはいかないので、そのうちに特に多くの文献に含まれるもののみを示してある。

COMPUTER	921
SYSTEM	579
DIGITAL	352
DATA	344
CONTROL	332
MEMORY	312
DESIGN	300
COMPUTERS	273
SYSTEMS	264
LOGIC	219
ANALYSIS	200
SIMULATION	196
MAGNETIC	191
USING	180
STORAGE	165
CIRCUITS	159
PROCESSING	156
INFORMATION	153
PROGRAM	145
CIRCUIT	141
METHOD	141
MIT	127
ANALOG	124

AUTOMATIC	1 2 0
PROGRAMMING	1 1 5
APPLICATION	1 1 1
RECOGNITION	1 1 0
PROBLEMS	1 0 2
NEW	1 0 1
FUNCTIONS	1 0 0
DEVICE	9 7
TECHNIQUES	9 5
TIME	9 0
ELECTRONIC	8 9
ON-LINE	8 7
RECORDING	8 6
MACHINE	8 5
LINEAR	8 4
INTEGRATED	8 3
PROCESS	7 8
COMPUTING	7 8
FILM	7 7
TAPE	7 6
TECHNIQUE	7 5
SEQUENTIAL	7 3
ANALOGUE	7 3
APPROACH	7 3
CHARACTER	7 2
APPLICATIONS	7 1
LANGUAGE	7 1
MEMORIES	7 0
HIGH	6 8
PROGRAMS	6 7

DISPLAY
 SOLUTION
 SWITCHING

66
 65
 63

しかしこの表にある様な多くの文献に含まれるキイ・ワードは全体から見るとほんのわずかで、10以上の文献に含まれるキイ・ワードは11000語中602でキイ・ワード全体の5.5%にしかすぎない。一方1つの文献にしか含まれないキイ・ワードが7337語もあり全体の62.5%もしめている。

次の表はこの様な関係をまとめたもので、表の見方は、例えば文献数の欄が20~30 ($20 \leq n < 30$ の意)文献に現われるキイ・ワードが128種類有ると言う意味である。

文 献 数	キイ・ワード数
200~	11
100~200	19
90~100	3
80~90	6
70~80	12
60~70	10
50~60	15
40~50	35
30~40	49
20~30	128
10~20	314
1~10	10411
合 計	11013

4.3.3 コマンドの種類と機能

このシステムで用いるコマンドは次の4種類に分けられる。

A. FACOM230-60 TSSとの会話用

1. ¥JOLDOR

2. BYE

B. キー・ワードにより文献を検索する為のもの

3. SELECT

4. COMBINE (現在は2項のみを許す)

C. Bで検索した文献集合の出力に関するもの

5. DISPLAY

6. NEXT

7. TYPE

8. PRINT

9. LIMIT (未完)

D. 一連の検索作業の終了を示すもの

10. ENDSEARCH

これら10個のコマンドの他に、利用者へのサービス機能として現在、次の様なガイダンス・ルーチンを備えている。

E. コマンドの種類と、その機能を教えてくれるもの

TEACH (利用できるコマンドと機能の一覧表が表示される)

F. ある一つのコマンドの機能を教えてくれるもの

TEACH any-command

以下に前記10個のコマンドの機能を説明する。

(1) ¥JOLDOR

(; はエンド・マークでディスプレイ・ターミナルのSEND ボタンを表わす。以下同様)

TSS に "これからJOLDOR システムを使いたい。" と意思表示するマクロ・コマンド呼出し指令であり、システム側から質問してくる利用者情報の各事項にうまく答えることができれば、そのユーザーは本システムを利用することができる。利用者情報は利用者の登録番号などである。この時システムからは、システムの正式名称や検索開始日時などをタイプライタに出力する。

(2) BYE ;

すべての仕事が終了したことを知らせるコマンドで、全所要時間、CPUタイムなどを打出して交信を終える。

TSSとのつながりを断つLOGOUTの機能である

(注1)

(3) SELECT キー・ワード1 [, キー・ワード2, … , キー・ワードn] ; (下線は省

略形)

指定したキイ・ワードが引用されている文献の集合を選びだす。システム側からは文献集合の set-no, その set 中の文献数などをディスプレイ端末とタイプライタに出力してくる。

(4) COMBINE set-no の論理演算式;

今迄 SELECT された複数個の文献集合を論理演算で結合し、新しくできた文献集合に対して新しい set-no をつける。

1 + 2 * 3 の様に論理和, 論理積, 論理否定を許し, 夫々 + * - を用いて表わす。COMBINE でできた set-no も以後の COMBINE で利用できるようになっており, ユーザーは次々と興味ある内容を含む文献集合を規定していくことができる。

システム側からは set-no, set 中の文献数, 演算式が端末に出力される。

(5) DISPLAY set-no ;

今迄の SELECT, COMBINE でできあがった文献集合の中から, 指定された set-no の文献集合の最初の文献内容をディスプレイ端末に表示する。ユーザーディスプレイされた文献内容を眺めて, さらに新しい検索用のキイ・ワードをみつけ出すことができ, 再び 3 ~ 5 の操作を繰り返すことにより効果的な検索が可能である。

(6) NEXT ;

今ディスプレイされている文献の次の文献をディスプレイする様に要求するコマンド

(7) TYPE ;

今ディスプレイされている文献の内容をそのままタイプライタにハードコピーをとらせる指令

(8) PRINT set-no ;

指定した set-no の文献集合をセンターのライン・プリンタにハード・コピーをとる様に指令するコマンド

(9) LIMIT set-no ;

文献集合をディスプレイする際, 著者名, 書誌名, 発行年度などの制限を加える。この制限をパスした文献のみ出力される。

(10) ENDSEARCH ;

この command がキイ・インされると, それ迄の検索に要した時間を出力して一連の検索を終了する。そのユーザーが使用していたコアのワーク・エリアはクリアされ, 今迄

の検索結果も廃棄する。

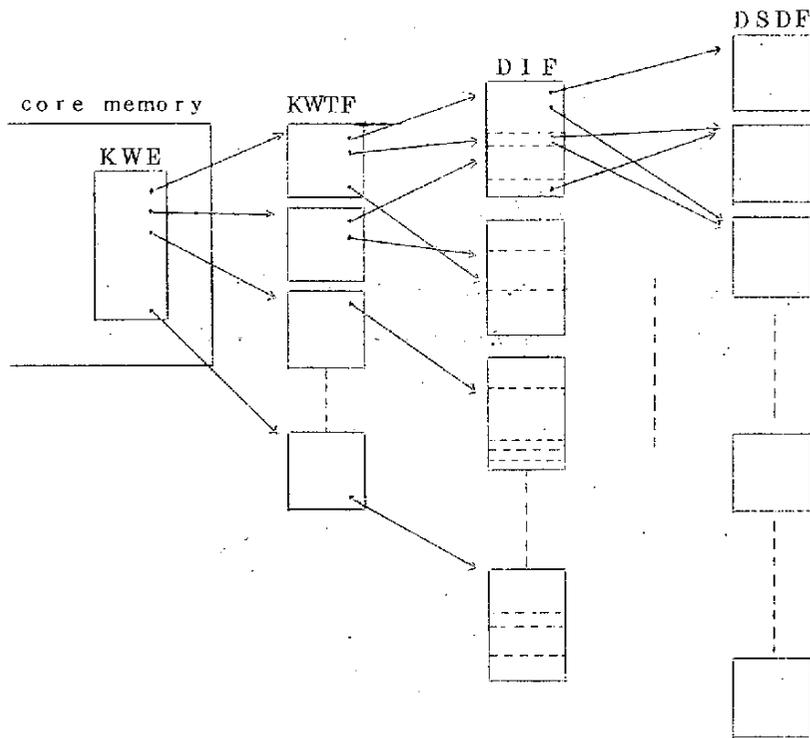
必要ならば又新しく SELECT から次の検索を続けることができる。

(注1) 以後キイ・ワードとは、文献の標題中に表われているキイ・ワードとディスク
リブタ・ワードの両方をあわせてキイ・ワードと呼ぶ

4.3.4 ファイルの構成

この IR システムで検索はあくまでキイ・ワード中心に行なり事が根本思想となっているので、ファイルの構成も指定したキイ・ワードに対して関連する文献をなるべく早く、簡単に取り出せる様に考えている。

その為にアップデートが少々めんどうになっても、何段階にもチェーンを付ける事をなるべくさけて、ごく単純な構成にしてある。ファイルの構成は次の図の様になっている。



ファイルの構成図

この図にあるファイルは次の様なものである。

D S D F : Document Source Data File

いわゆる文献のソース・データで標題、著者名、著者の所属、文献番号、書誌事項、ディスクリプタ・ワードが入っている。

1つ1つの文献データの大きさは可変長なのでセグメント化を行なっている。

1つの文献に対するデータには50~100Wぐらい使っている。

D I F : Document Index File

D S D F の見出しになるファイルで各キイ・ワードに対応する文献の集合を表わしており、固定長にブロック化されている。

K W T F : Key word Table File

キイ・ワードの各キャラクタを節とするリスト構造のテーブルで各キイ・ワードの終りにはD I F の対応するブロックのアドレスとブロック内での相対アドレスを示すポインタが付いている。これも固定長にブロック化されている。

K W E : key Word Entry table

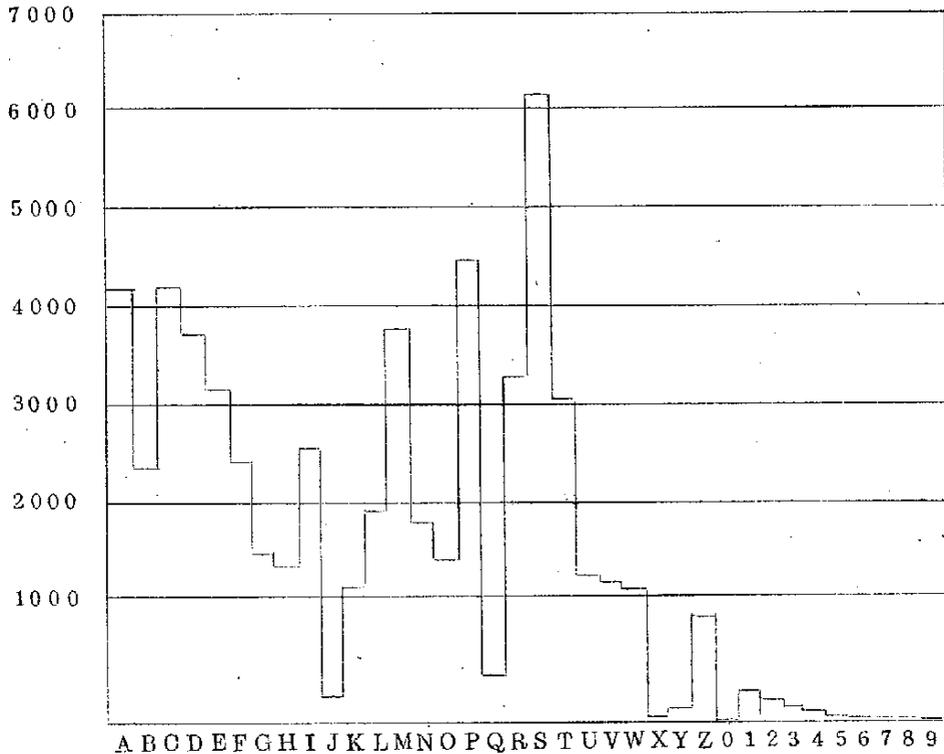
指定されたキイ・ワードがK W T F のどのブロックに入っているか調べる為のテーブルでコア・メモリーに常駐している。

(1) K W T F のブロックング

1万語程のキイ・ワードで1つのリスト構造のテーブルを作ると1つの節を1ワードで済ませたとし60~80KW必要となる。本来ならこれをコア・メモリーに常駐させたいところだが、なにせ現在使えるコア・メモリーは30KW程しかない。これではとても無理なのでテーブルを適当なブロックに分割しておき必要なものだけコア・メモリーに読み込む様にした。

分割方法について始めは単純に考えてキイ・ワードの第1文字がAで始まるものを1つのブロックに、Bで始まるものを又1つのブロックに、……とキイ・ワードの頭文字ごとに1ブロックを作れば良いと思っていた。しかし実際には7000ほどの文献から抜き出したキイ・ワードについてツリー(tree)を作り節を数えてみると次の様な結果を得た。

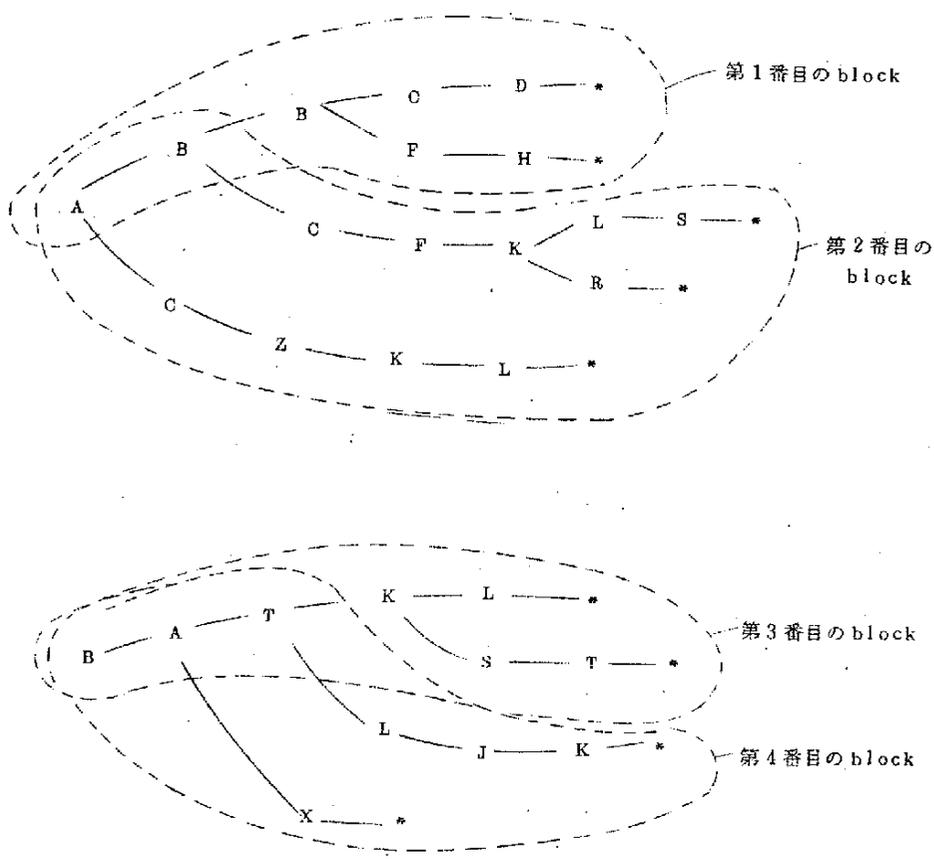
←ノード数)



これを見ると明らかにブロック・サイズのパラッキが大きい。出来れば各ブロックは固定長にしたいのでこの方法は良くない事が解った。

そこでキー・ワードを、まずアルファベット順にソートしておき初めから順にリスト構造のテーブルを作っていく。テーブル・サイズが一定の大きさ $L - \epsilon$ を越えたらここまでを1つのブロックにする。キー・ワードは最高30文字までしか許していないから、 ϵ として30を取って置けば1つのブロックが L を越える事はない。

次のキー・ワードから又新たにリスト構造のテーブルを作り始めて前と同じルールに従って新しいブロックを作っていく。これを繰り返せば全部のキー・ワードを固定長のブロックに納める事が出来る。



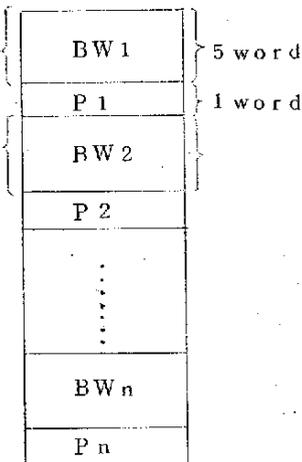
この様にリスト構造のテーブルを分割すると一見ブロック間にダブって入る文字が出来て得策と思えない様だが、このダブリは高々ブロックの切れ目に在るキイ・ワードの文字数以下であるから、これもキイ・ワードの平均表が9~10文字である事を思うと、1ブロック1KWとして平均、約 $(10/1024) \times 100 = 1$ (%)の増加にしかすぎない。仮に全体が60~80ブロックから成るとしても600W~800Wぐらいいしかテーブル・サイズが増えない。

(2) KWEとKWTFの関係

KWEは指定されたキイ・ワードがKWTFのどのブロックに在るかを探す為に使うテーブルでコア・メモリに常駐している。

このテーブルにはKWTFのブロックの区切りとなるキイ・ワード(以下境界語と言ひ)をアルファベット順に並べて各境界語の後にKWTFの各ブロックへのポインタを持って

第1番目のブロックの区切りを表わす
第2番目の……



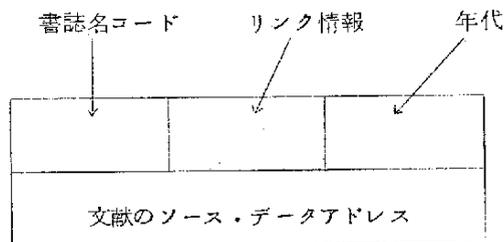
いる。ここにBW₁, BW₂, …… BW_nは境界語を表わし, P₁, P₂, …… P_nはKWTFの対応するブロックのアドレスを表わしている。

あるキイ・ワードが指定されるとKWEの境界語と順に比較していき,始めて順位(アルファベット順)が逆転する所を見つけてKWTFの対応するブロックを見つけることが出来る。

(3) DIFの構造

コマンドの項で説明した様にこのシステムでは"SELECT"というコマンドにより指定されたキイ・ワードに対して関係ある文献を取り出し,文献集合にset-noを与えたり,"COMBINE"を使ってset-noがすでに与えられ文献集合に論理演算をほどこして新しくset-noの付いた文献集合を作り出せる様になっている。この様な操作を文献のソースデータで直接やり取りするのでは不便なので,それぞれの文献に対して「見だし」を作り,「見だし」で扱える様にしてある。

計算機内部で一つの「見だし」は2Wで表わされ次の様な形式をしている。

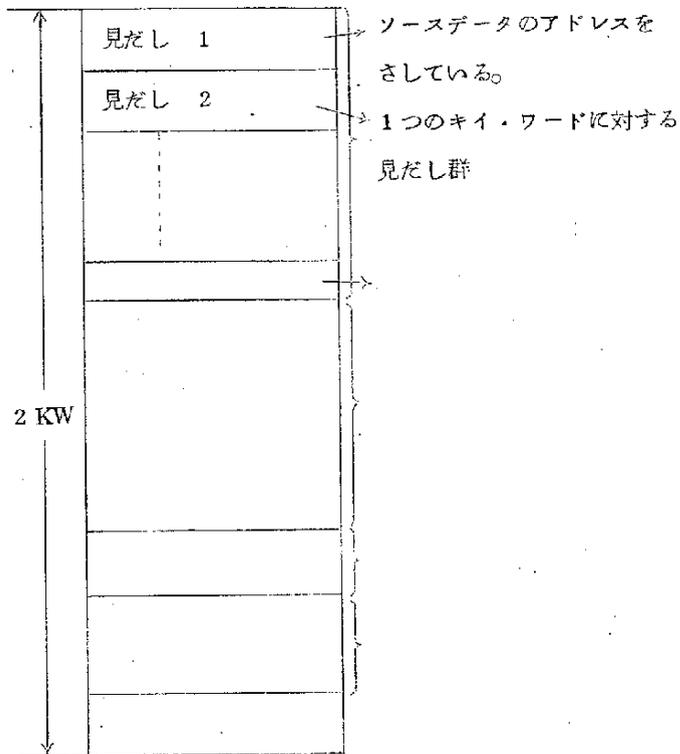


発行年代や書誌名に

D I F の 1 ブ ロ ッ ク

関する情報を持たせてあるのは「LIMIT」と言うコマンドで、キーワードを指定して得られた文献の集合に対して発行年代や書誌名で絞られる様にしたかと思っているからである。

又、リンク情報はオンライン・アップデートの為に用いる。この様な「見だし」を使ってキーワードに関係ある文献のソース・データがD S D Fの何処に在るか解る様にしてある。1つ1つのキ



ーワードに関係ある文献数は、大きいものは1000文献小さいものは1文献しかないものまで非常にバラツキている。D I Fのサイズは実測の結果7000文献でおよそ80KWぐらいになるので適当なブロックに分割せねばならず、現在ブロック・サイズは2KWとしてあるが文献数が増えた時の事を考えるとどの様に決めるかは一考を要する所と思われる。

以上のファイルの具体的な例を次に示す。

①, ②, 2つの文献があり、各キーワードは下線で示してある。

KWTF, D I F, D S D Fの関連を下図に示す。

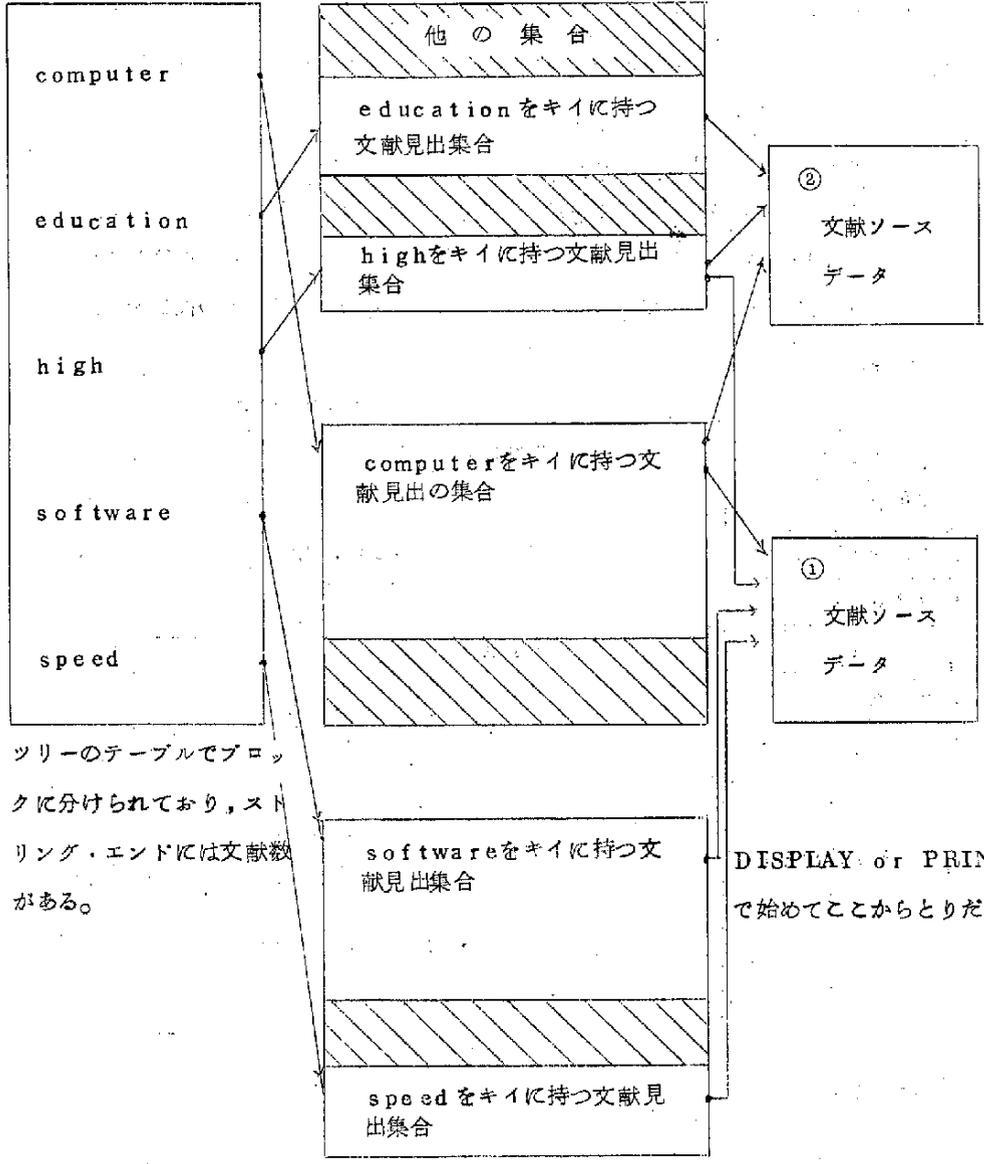
① Software for High speed computer

② Computer in High education

KWTF

DIF

DSDF



ツリーのテーブルでブロッ
クに分けられており、スト
リング・エンドには文献数
がある。

DISPLAY or PRINT
で始めてここからとりだす。

SELECTでこの中から set-no に
対応するものがワーク・ファイルに入る。

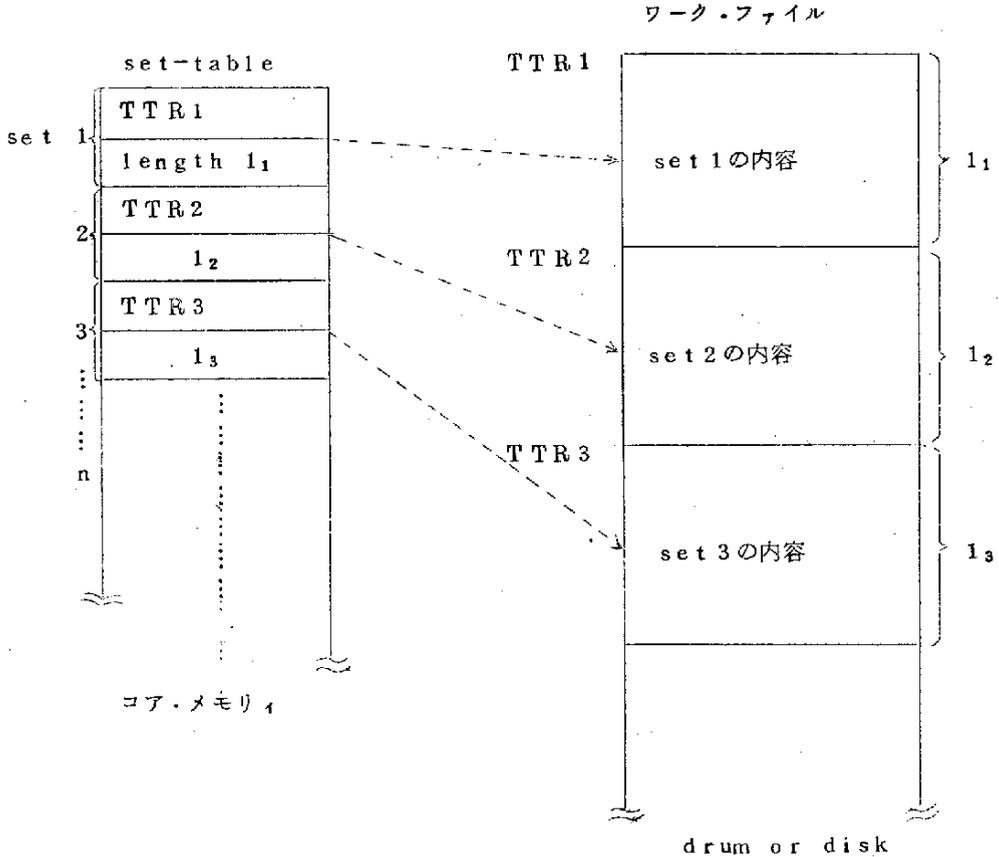
(4) ワーク・ファイルの役割

JOLDORではSELECT, COMBINEコマンドによって, 次々と文献の見出しの集合をつくり出して行く。すでにつくられている集合群は検索の途中でCOMBINE又はDISPLAYコマンド等によって, 再度参照される。従って一度つくった集合は検索終了時迄保存しておかなければならない。これらの文献見出しの集合群の保存に使用されるのがJOLDORワーク・ファイルである。

a) ワーク・ファイルの内容

ワーク・ファイルの内容そのものは, DIFの内容と全く同じである。

set-table (コマンド) でつくられた文献集合が, ワーク・ファイルのどこに書いてあるかを示すテーブルで, アクティブな端末毎に持っている) との関係を下に図示する。

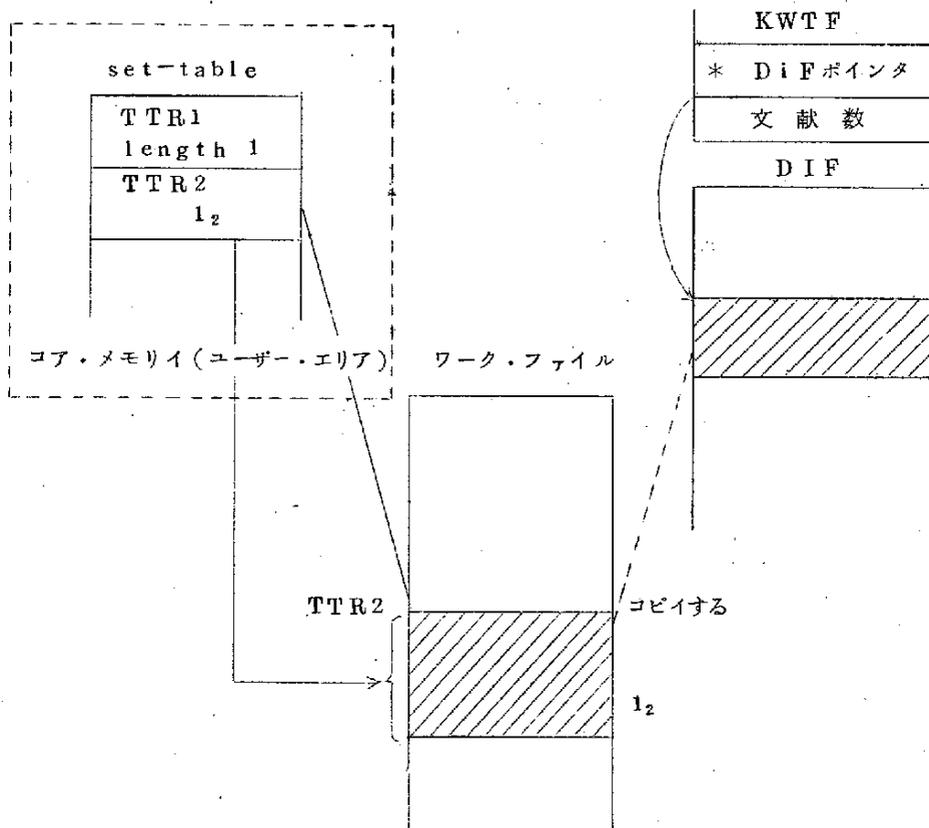


年代	リンク	書誌名
文献ソース・データのアドレス		

ワーク・ファイルの各アイテムの内容

- ワークファイル及びset-tableはアクティブな端末の数だけ用意され、そのユーザーの為だけに用いられる。
 - ワーク・ファイルへの記録形式は可変長あるいは不定長として、ブロックングは行なわない。
 - ファイルの編成法は、順編成あるいは、順インデックス編成とする。
- b) 各コマンド実行時のワーク・ファイルの使われ方。

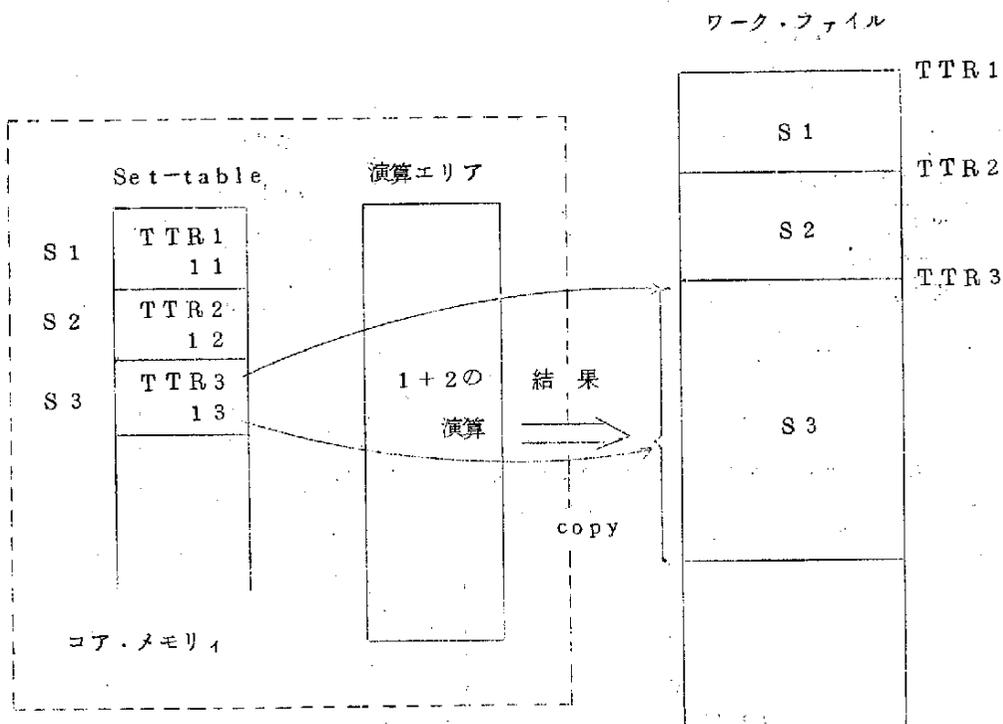
イ SELECTコマンド



SELECTコマンド・ルーチンはキー・ワードを抜き出すとKWEテーブル(コア・メモリ中の全ユーザー共通のテーブル)により、そのキー・ワードが含まれているKWTをユーザー・エリアにロードし、文献数をディスプレイする。又、そのディスプレイとマルチタスクで、上の図に示した様を手順でset-tableへの登録と、ワーク・ファイルへの書き込みを行なう。

□ COMBINEコマンド

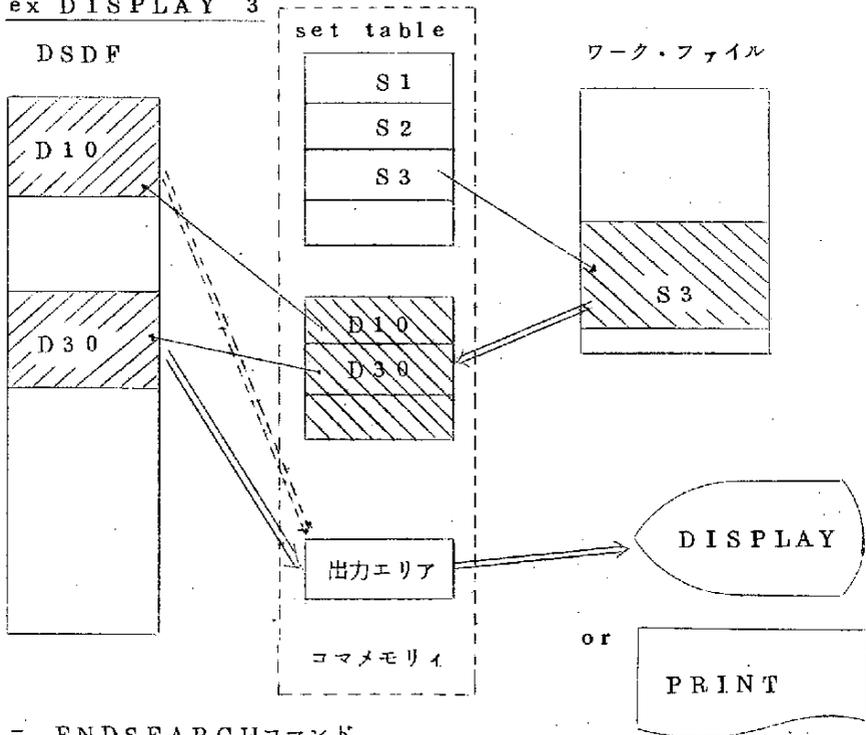
ex COMBINE 1+2



set-tableを参照してset1とset2を演算エリアに読み込んで、演算し、その結果をset-tableに登録し、ワーク・ファイルに書き込む。演算エリアの大きさ、とり方はユーザー毎にダイナミックに決定するように設計してある。

ハ DISPLAY, TYPE, PRINTコマンド

ex DISPLAY 3



ニ ENDSEARCHコマンド

ENDSEARCHがくると、set-table をクリアして、又、最初から SEARCHの仕事ができる様にしてやる。

4.3.5 検索の流れ

前節のファイルの構成の説明で、これらのファイルをいかにたぐっていけば望む文献を得ることができるかが大體理解できると思われる。本節では主にコマンドの実際的な動きの面からなるべく具体的に検索の流れを示したい。

(1) SELECTコマンドの動き

例えば "AUTOMATA" というキー・ワードを持つ文献を選ぶ時は

- イ) SELECT AUTOMATA: とキー・インすると、先ずコマンド・ディストリビュータ (コマンドを識別するルーチン) により、コマンドの識別が行われ、SELECT処理ルーチンが呼ばれる。
- ロ) SELECTルーチンはキー・ワード "AUTOMATA" を抜き出す。
- ハ) キー・ワード "AUTOMATA" がKWTFのどのブロックに含まれているか

をKWEテーブルにより調べる。

- ニ) KWEでわかったポインタでKWTFの“AUTOMATA”を含むKWTの該当ブロックをコア・メモリィにロードする。
- ホ) KWTFはリスト構造のテーブルであるから、このリストをたぐっていくことにより、“AUTOMATA”というキー・ワードが使われている文献の集合に関する情報を示すDIFのブロック・ポインタがわかる。
- ヘ) KWTFの各ストリングの終端には4.3.4節で述べた様にDIFへのブロック・ポインタの他にそのキー・ワードが使われている文献数も表わされている。これで“AUTOMATA”に関する文献の数がわかるのでJOLDORシステムはこの文献集合のset-noと文献数をディスプレイ端末に表示する。

```
*ENTER COMMAND*
SELECT AUTOMATA
SET NO      1
NO IN SET  13
```

画面1

ユーザが画面1をみて、次に打込むコマンドを考えている間は、コマンド待ちの状態であるから、CPUは遊ぶかもしれない。

そこでシステムはDIFへのブロック・ポインタをたよりに次のレベルのファイルをローディングする別のタスクを発生させる。すなわち画面1のような情報をディスプレイするタスクとDIFに関する処理を行うタスクとの2つのタスクをマルチ・タスクとして処理するわけである。

- ト) DIFをサーチするタスクについて

まずDIFのブロック・ポインタというのは4.3.4のKWTFの説明にあるようにDIFのブロック・ナンバーを示すポインタと、そのブロック中で、あるキー・ワードに関する文献集合が入っている相対先頭アドレスを示すポインタの2つから成っている。これらのポインタにより、例えばキー・ワード“AUTOMATA”に関する文献を示す見出しの集合をワーク・ファイルに書き移しておき、その場所をset-noと共に記録しておく。これでこのタスクは終了する。

- (2) COMBINEコマンドの動き

これは文献集合にset-noで論理演算を施すコマンドであることは4.3.3節で述べた。

イ) コマンド・ディストリビュータにより識別され, COMBINEのルーチンが呼ばれる。

ロ) COMBINEルーチンはオペランドの関係式を評価する。

ハ) 評価にしたがって文献集合どうしの論理演算を行なう。例えばCOMBINE 3 * 4 : とキイ・インされたとするとワーク・ファイルからset-no 3 と set-no 4 の文献集合に新しくset-no をつける。

ディスプレイ端末には, COMBINEで新しくできた文献集合の set-no とその集合中の文献数を表示する。

```
*ENTER COMMAND *
      COMBINE 3 * 4
      SET NO   1 5
      NO IN SET 1 5
```

画面 2

こうしてCOMBINEでできた文献集合をさらにCOMBINEで論理演算を施すことも可能である。ユーザはSELECTとCOMBINEをうまく使うことにより次々と興味ある内容の文献を検索することができる。

(3) DISPLAYコマンドの動き

このコマンドはset-no を指示してその文献集合の最初の文献内容をディスプレイ端末に表示するものである。

イ) コマンド・ディストリビュータにより識別し, DISPLAYルーチンを呼び出す。

ロ) 指示されたset-no によりワーク・ファイルから文献の見出しの集合をコア・メモリーにロードしてくる。

ハ) 文献の見出しは, 4.3.4節のDIFの構造に述べてあるように文献のソース・データの場所を示すDSDFへのポインタを持っている。このポインタによってDSDFから文献の内容をコア・メモリーにロードしてディスプレイ端末に表示してやる。

ニ) ハ) でディスプレイ端末に一つの文献を表示している間はCPUが遊ぶかもしれないので, システムは次の文献表示の準備として, 同じ文献集合中から次の文献内容をマルチ・タスクでDSDFからコア・メモリーにロードしておく。

(4) NEXT コマンドの動き

(3)でディスプレイした文献の次の文献内容をディスプレイ端末に表示する指令のコマンドでありこの時も次の文献内容表示の為先取りのマルチ・タスクを行なう。

(5) TYPE及びPRINTコマンドの動き

これらは端末のタイプライタ及びセンターのSYSOUT(システム・アウトプット・プリンタ)にハード・コピーをとる指令である。

ディスプレイ端末に文献内容を表示してみて、タイトルあるいはディスクリプタ・ワード中にもっと適当なキイ・ワードを発見したなら、そのキイ・ワードでもう一度SELECTコマンドを指令して同様に検索を繰り返していく。

(6) ENDSEARCHコマンドの役割

こうして検索の仕事をつづけていくうちに、今迄いくつもつくられた文献の見出しの集合がワーク・ファイルのどこにあるかを示すテーブル、あるいはワーク・ファイル自身が一杯になってしまいます。こうなってしまうともうこれ以上検索の仕事が続けられなくなるのでこのコマンドでワーク・ファイル中の文献見出し集合のありかたを示すテーブルをクリアしてやる。(set-no tableのクリア)これでユーザは又新しく検索の仕事を開始することができる。

しかしせっかくつくりあげた文献集合をENDSEARCHでクリアして以後使えなくするのは非常に惜しい。

この為将来SAVE, LOADなどのコマンドを用意して後日の検索にもそれまでの検索結果を利用できるようにしたいと思っている。

(7) コマンド・エコーのプリント

ディスプレイ端末には、画面1または2のように、コマンド毎の情報が表示されるが、ディスプレイの画面の大きさには限りがあるので、比較的長い検索作業を行なうと前の方の情報は消えてしまう。しかし、set-noを次々とCOMBINEしたり、あるsetの文献数を振り返ってみたいというような時に、いままでの検索過程の記録が残っていることが望ましい。そこでターミナルのタイプライターにその過程をプリントする。例えば次の様なコマンド・エコー(command echo)がプリントされる。

USER NO	1005	69/1/15
COMMAND	SET NO	NO IN SET
S AUTOMATA	1	13
S THEORY	2	10
C 1*2	3	6
S AUTOMATION	4	9
C 3+4	5	15
D 5		
N		
N		
P 5		
E		
TIME	XXSEC	

SはSELECT

CはCOMBINE

DはDISPLAY

NはNEXT

PはPRINT

EはENDSEARCH

4.3.6 オンライン・アップデート

MONITOR-Vではファイルのブロックごとにロックをかける機能を持っている。

(排他的占有機能)

ファイルをアップデート・モードで使い、或るブロックを読むとそのブロックにロックがかかり、そのブロックに書き出すか、ロックをとくマクロ命令を使えばロックがとける様になっている。

この機能を利用すると同時に何人ものユーザーがアップデートをし、又検索をする事が可能になる。

(1) プログラムの構造

UPDATEルーチンはリエントラント (Reentrant) をプログラムで検索ルーチンと同じプロセッサに組み込まれる。この為“UPDATE”と言うコマンドをもうけ、検索モードからいつでもアップデート・モードに切り換えることが出来る様になっている。

UPDATEルーチンと検索ルーチンが同じプロセッサに組み込まれると言うのは、かならずしも同じ領域にあるとはかぎらず検索ルーチン内にある“KWE”がアップデート・ルーチンからも参照可能であれば良い。

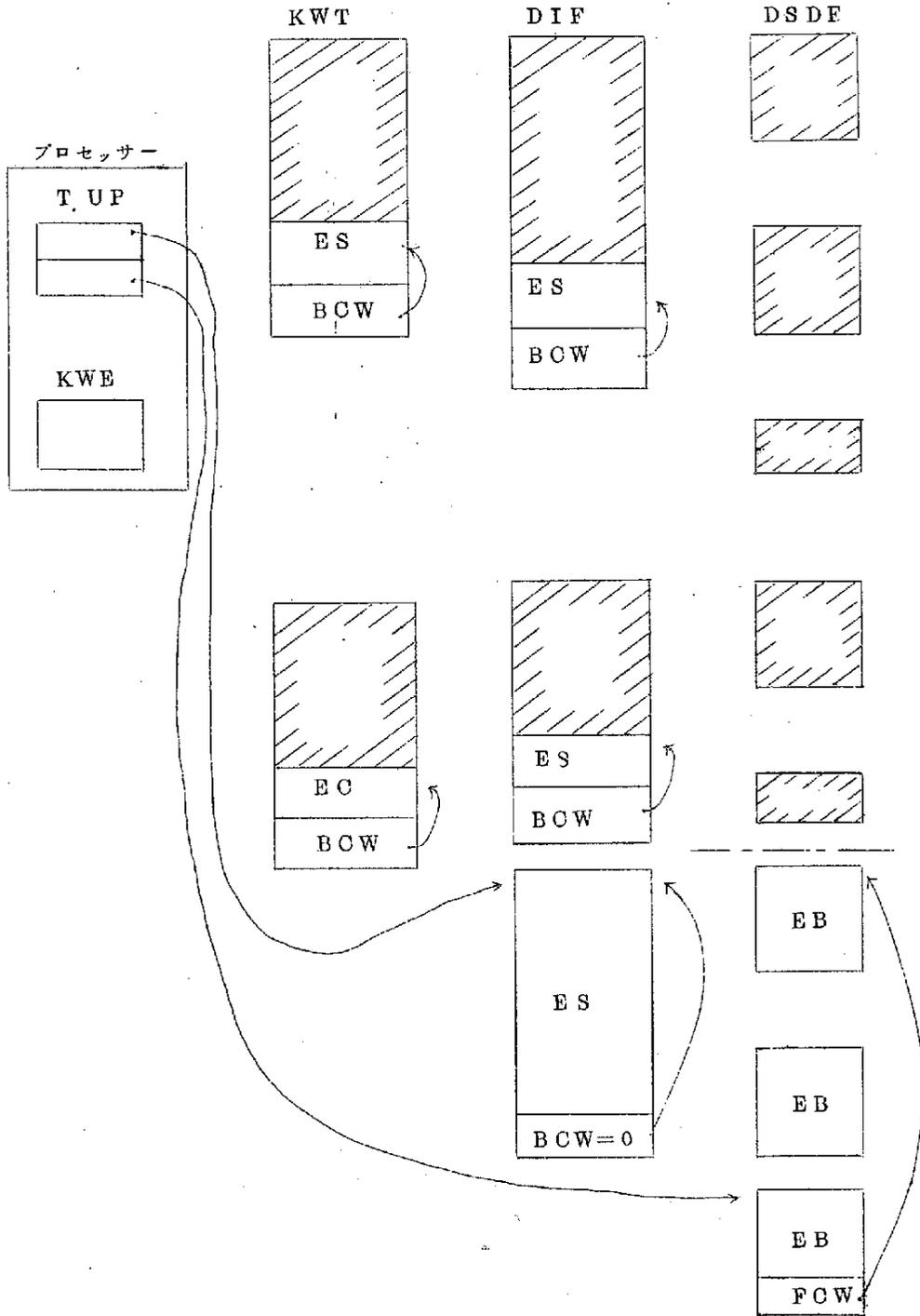
JOLDORに於けるオンライン・アップデートはただ単にオンラインでアップデートが出来るだけでなくアップデートされた情報はただちに検索の対象となる事に特徴がある。その為各ファイルのアップデート手順は勝手に出来ず、順番には十分気を付けなくてはならない。

検索の対象となる情報は、それが単に物理的に付け加えられるだけでなく各ファイルのリンケージがとれた時始めて生きてくる。

例えばD S D Fにソース・データが加えられてもこの時点ですぐに検索の対象とはならない。この事は勝手にD S D Fにデータを加えても検索のじゃまをしない事を意味している。同じ事がD I Fにドキュメント・ディスクリプタ (文献の見出し) を加える時にも言える。しかしK W Tに新しいキイ・ワードが登録されると話は違ってくる。

K W Tに登録されたキイ・ワードはただちに検索の対象となるので、この時点ですでにD I F, D S D Fにはアップデートが完了し、しかもK W T, D I F, D S D Fのリンケージがとられていなくてはならない。そうしないとキイ・ワードでたまたま検索しようとするこのキイ・ワードに対応する文献が実際あるはずなのに見つからない事になって都合がわるい。そこでアップデートの手順はまずD S D Fに文献のソース・データをアップデートし、次にD I FにそのディスクリプタをアップデートしてD S D Fのデータとリンケージをとる。その上でキイ・ワードをK W Tに登録してD I Fとのリンケージをとればこの時点でもうこの文献が検索の対象となり得る。しかもアップデートを行なっている間検索ルーチンからはアップデートされた情報は見えないから最後の、キイ・ワードアップデートされるまで検索のじゃまをしていない。

各ファイルはファイルの創成或はバッチによるアップデート時に次の様なフォーマットにする。



- BCW: 各ブロックの空き領域の先頭番地(ブロックでの相対番地)が入っている。
 FCW: D S D Fにある先頭の空きブロックの相対番地(T T R)が入っている。
 T U P: D S D Fの最終ブロック(FCWを含むブロック)のアドレスとD I Fの最終ブロック(ESのみのブロック)のアドレスをもつテーブル
- ES : 空き領域
 EB : からのブロック

a) KWTのアップデート

アップデートしようとするキー・ワードを含むはずのブロックを読み、実際にそのキー・ワードを含むかどうか確かめる。実際にあれば文献数を1つ増す。ない時はキー・ワードを登録してBCWを増す。

b) D I Fのアップデート

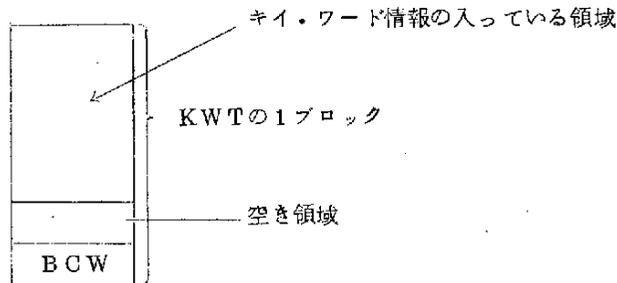
1つのキー・ワードに対応するドキュメント・ディスクリプタの集合はお互にポインタでリンクをつけておき最後のディスクリプタには最後であると言う情報をもたしておく。

アップデートしようとしたキー・ワードがすでにKWTにあった場合はD I Fにディスクリプタがあるはずだからこのブロックを読み、空き領域の先頭に新しくディスクリプタを付け加え、BCWを増しさらに、すでにあるディスクリプタとのリンクをつける。

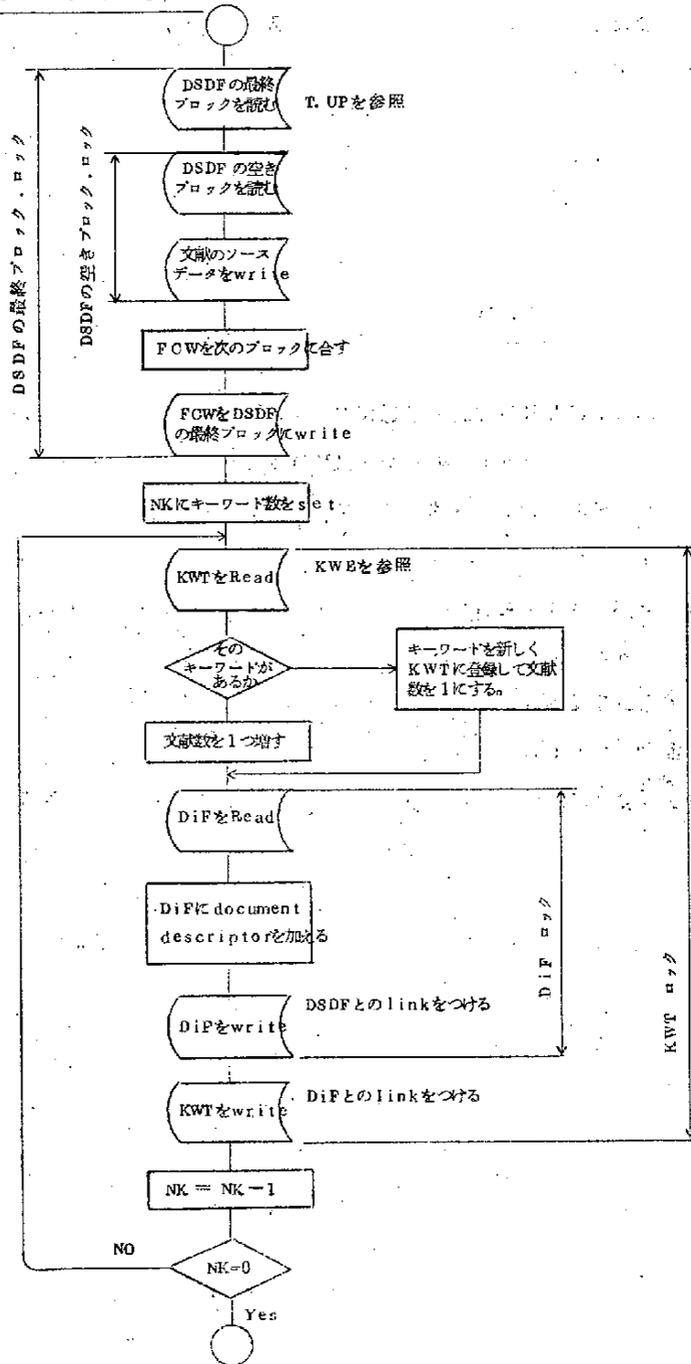
もしKWTに新しくキー・ワードが登録された場合には、そのキー・ワードに対応するドキュメント・ディスクリプタはD I Fの最終ブロック(T . U Pを参照して位置がわかる。)に登録される。

(2) オンライン・アップデートの為に各ファイルには次の様なコントロール・ワードを持つ事とする。以下BCWと略記する(ブロック・コントロール・ワード)

a) KWT

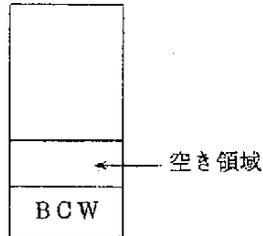


ON-LINE UPDATE BLOCK CHART



KWTの各ブロックの最終語をBCWとして使いBCWにはいつも空き領域の先頭番地(ブロック内での相対番地)が入っている。

b) DIF



BCWにはKWTと同じ様にDIFの各ブロックの空き領域の先頭番地が入っている。アップデート・ルーチンはKWT, DIFのBCWを参照してキイ・ワード或はDSDDFのディスクリプタのアップデートを行なう。

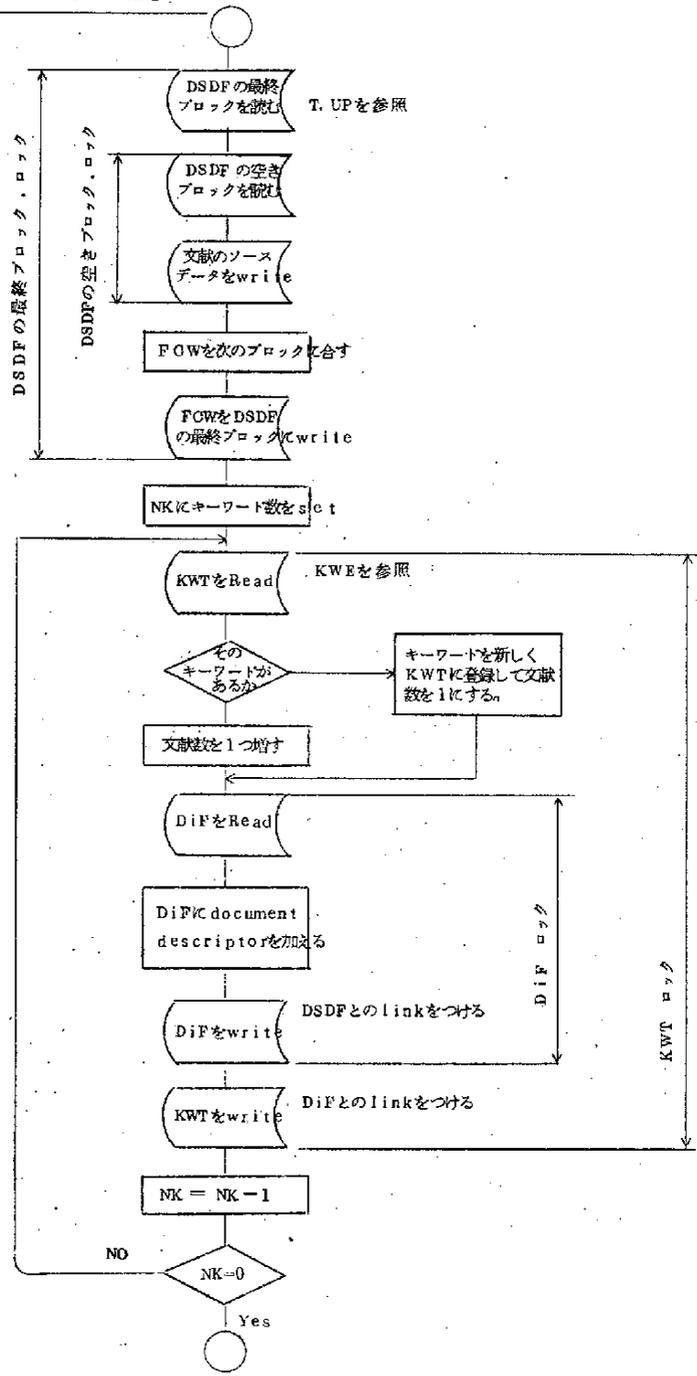
4.3.7 バッチ用サービス・ルーチン

このシステムで考えられているバッチ用サービス・ルーチンは次の3種類である。

- (1) ファイル・ジェネレータ
- (2) STATISTICSルーチン
- (3) SDIサービス

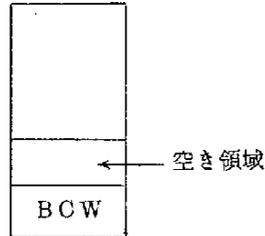
システム全体に対するこれらのルーチンの位置づけは次図に示す通りである。

ON-LINE UPDATE BLOCK CHART



KWTの各ブロックの最終語をBCWとして使いBCWにはいつも空き領域の先頭番地(ブロック内での相対番地)が入っている。

b) DIF



BCWにはKWTと同じ様にDIFの各ブロックの空き領域の先頭番地が入っている。アップデート・ルーチンはKWT, DIFのBCWを参照してキー・ワード或はDSDFのディスクリプタのアップデートを行なう。

4.3.7 バッチ用サービス・ルーチン

このシステムで考えられているバッチ用サービス・ルーチンは次の3種類である。

- (1) ファイル・ジェネレータ
- (2) STATISTICSルーチン
- (3) SDIサービス

システム全体に対するこれらのルーチンの位置づけは次図に示す通りである。

イ) キー・ワードインテックス・リスト

KWOCのキー・ワードインテックス・リストと同じ様式のリストで定期的にアップデートで追加する文献のぶんどけリ스팅する。このリストは新刊紹介の形で利用者に配布される。

ロ) トランザクション・ファイル

これは、定期的にアップデートで追加する文献のトランザクション・ファイルである。このファイルの様式は当然マスター・ファイルと同じになっている。

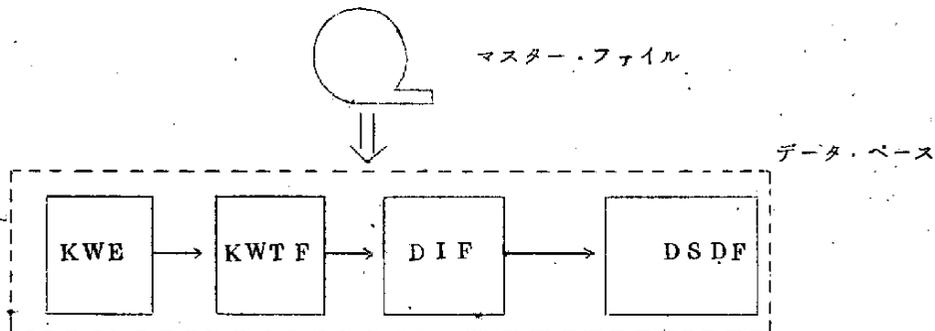
タイトル中に表われているキー・ワード及びディスクリプタ・セグメントに記載されているディスクリプタ・ワードをキー・ワードとして抜き出しそれを見出し語として文献情報を整理してテープ上に書き出して置く。

○ マスター・ファイルの更新

すでに今迄の文献のぶんについてつくられているマスター・ファイルと前記のトランザクションとを併合して新しいマスター・ファイルとする。したがって文献情報の更新は常にaルーチンから行なり。

b) データ・ベースジェネレータ

a)でできたマスター・ファイルからデータ・ベース(KWE, KWTF, DIF, DSDF)をジェネレートするルーチンである。4.3.3節で述べたファイルの構造をみてわかる様にマスター・ファイルからKWE, KWTF, DIF, DSDFを各々関連をつけながら作りあげていく。



FACOM 230-60 MONITOR-V のデータ管理システムの下で働かなければ
ならないので、色々細かい制約が加わっている。

(2) STATISTICSルーチン

◎ 本システムは、JOLDORにおける文献検索から、各種の統計データを取り、JOLDORシステムのデータ・ファイルのアップデート、及び、システムのアップデート時の基礎資料とする。

◎ 統計データは、次のものとする。

- ユーザー毎の使用回数と、使用時間
- キー・ワード毎の参照頻度
- 文献毎の参照頻度

上記の統計データをとるため、JOLDORのオン・ライン検索プログラムにヒスト
リィ・ファイル(history file)を設け、下記のレコードを記録する。

- 処理年月日
- 使用ユーザー名、使用ユーザー・コード
- ターミナルNO
- JOB開始時刻
- JOB終了時刻
- JOB使用実時間
- 検索件数 (ENDSEARCHの回数+1 or 0)
- 参照キー・ワード
- 文献参照コード(リファレンス・コード)

◎ オンライン文献検索プログラムでは、参照キー・ワード、及び文献参照コードを、一
担 work-file に書き出し、¥ (yen mark) BYEコマンドの後、JOB TE
RMINATEルーチンが、ヒストリィ・ファイルに書き出す。

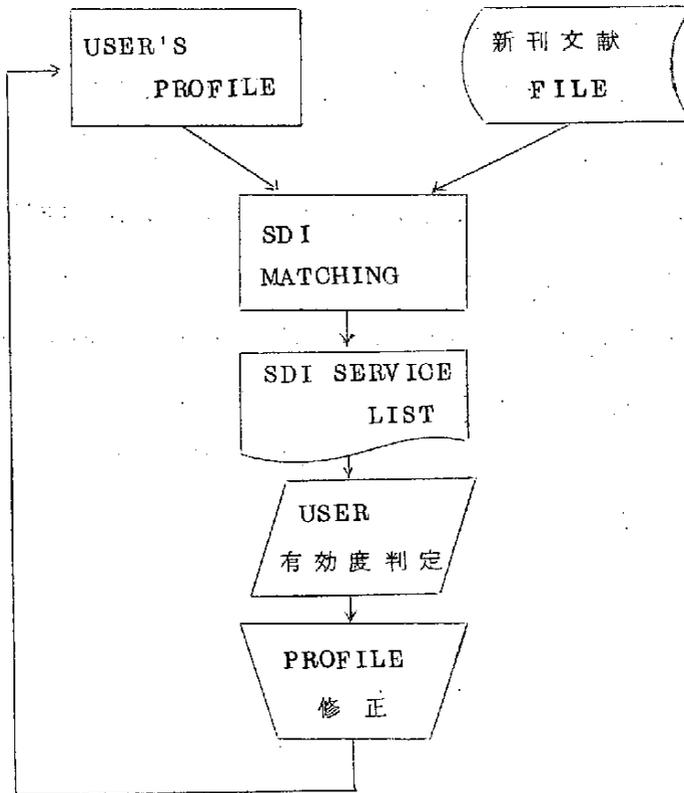
◎ ヒストリィ・ファイルは、週間業務で、テープ・セーブ、及び統計処理を行ない、フ
ァイルをイニシャライズしなおす。

(3) SDIサービス

◎ SDI (Selective Dissemination of Information) は、JOLDOR システムの一部として、JOLDOR を利用するユーザーが、あらかじめ登録し
たキー・ワードをファイル (USER'S PROFILE) に登録しておき、JOLDOR システムに新た

な文献集合が加えられた時点で（ファイル・ジェネレータのアップデート）その文献集合とマッチングし、キー・ワードの適合した文献を、その各々のユーザーにリスト・サービスするシステムである。

- ◎ このシステムによって、ユーザーは自分に必要な文献についての、Current Awareness を情報を得ることができる。



SDIサービス

- ◎ このシステムでは、ユーザーは複数のキー・ワードを登録し（20までの予定）、その内の、いくつを含む文献をリストすべきかを指定できるものとする（カード・コード=00カードのPRIORITY=nnで指定）。また、必ず含んでいなければいけないキー・ワードを指定することができる（カードコード=01カードのPRIORITY="MS"を指定）

[例]

ユーザーの登録語

1. INFORMATION
2. RETRIEVAL "MS"
3. SYSTEM

計3語で、PRIORITY=1~3とすると、

標 題

DIRECT ORGANIZATION DATA SET

×

(ディスクリプタ)

COMPUTER, RANDOM ACCESS MEMORY

標 題

INFORMATION STORAGE AND RETRIEVAL

○

(ディスクリプタ)

COMPUTER, IR SYSTEM

○の文献が適合する。

4.4 JOLDORシステムの検討と反省

1968年6月に、オンライン文献検索システム“JOLDOR”の開発を開始し、現在迄ほぼ一年を経過している。本格的な運転は1969年秋になる見込みなので、システム全体としての評価はまだできない。

本章では主にシステムの設計上の反省点をまとめてみる。

4.4.1 データの収集と作成について

IRシステムはデータが一番の基本となる。JOLDORではClearing Houseで発行している「U. S. Government Research & Development Reports」中から電子計算機に関連する分野を抜き出したもの、OACMの「Computing Reviews」の一部を収録文献の対象とした。

昭和43年6月現在 2,500文献のデータ・コーディングを完了しているが、転記の作業は非常に単調な仕事なので飽きやすい。かなりスペリングのミスもでている。

こういう大量の専門的なデータの収集には専門の情報サービス団体を利用するように考えないとデータ作成だけでかなりの大仕事になる。さいわい、日本でもJICSTあたりで海外論文のMT販売の計画があるようなので今後は積極的にそういうシステムを利用する様にした。

JOLDORのデータ・フォーマットでは、著者の所属と書誌名は記入のスペースが少し不足きみであった。

4.4.2 ファイルについて

IRのシステム設計上で一番の難点はファイルの設計であろう。

FACOM230-60 MONITOR-Vのデータ管理機能をフルに活用して、限られたコア・スペースで検索時間を最小にすることが、このファイル設計の第一目標だったので、かなり無理な小細工を加えた形となっている。

以下にファイル設計上の問題点を列挙する。

- (1) 当初の段階でユーザーが使用可能なコア容量は40KW足らずであり、TSSに拡張されるとなると、20KW位しかない。この少いスペースを多数のユーザー・ジョブが分割共用して用いることになるのでJOLDORは1ユーザーあたり10KWあれば検索ができる様にした。

従ってKWTやDIFが分割されたり、かなりファイルを切りつめて設計してある。

- (ロ) 検索時間を早くするのがJOLDORのファイル設計の第一目標なので、同じ情報が各ファイルに二重、三重に入ったりしてかなり無駄と思われる部分が生じている。

しかしこの問題は検索時間の短縮とアップデートの容易さという相反する条件の中からどちらかを選択しなければならなかったので止むをえない処置であった。

- (ク) 当初、ファイルの更新はバッチで行う計画にしており、オンライン・アップデートは途中からつけ足した部分なので、あまりすっきりしない方法となっている。これは(ロ)とも関連がある。

- (ニ) ファイルの logical space と physical space の対応がうまくなく、例えばDIFでは各シリンダー当たり15%が無駄なスペースとなっている。

これはMONITOR-Vのデータ管理、コア上に獲得できる作業用領域の単位、ファイルの1ブロックの大きさとの関連でデバイスのスペースをフルに使用できる様に改善可能である。

- (ホ) MONITOR-Vでは、複数のユーザーがコア上の同一領域(読書可)を共有できないためオンライン・アップデートがやっかいになってしまった。KWEやオンライン・アップデートのためのDIF、DSDFの空ブロック・ポインタをオンライン・プロセッサに組み込んでおかなければならぬので、バッチ・アップデートする度にプロセッサ自身もアップデートしなければならなくなってしまった。

- (ヘ) ファイル・スペースを縮小させるためにポインタやディレクトリがワードにまたがったりしているので、検索部分での操作がめんどろになってしまった。

かなり小細工が過ぎた設計になっている。

- (ト) 現在迄に設計したファイルだけではLIMITのコマンドはうまく生かせない。

年代と、書誌名のディレクトリ・ファイルをつくるか、各々のグループのチェーンをつけられよう。

- (チ) MONITOR-Vのデータ管理は一般的につくられているため、JOLDORのような使い方をするとかなり無理をした使い方になる。

これらの問題点を考えてみると、やはり本格的なIRをやるには、IR専用のモニターが欲しい。

4.4.3 オンライン・プロセッサについて

- (イ) 検索の過程で独立に並行してできる処理はマルチ・タスクとして動くように設計してある。しかし、処理プログラムで色々と小細工をしても、OSのオーバーヘッドがどの程度になるか予想がつかないので、マルチ・タスクにした効果の程も疑問である。この事は実際の運転が開始されてから色々と実験してみたいと思う。
- (ロ) 検索プログラムの処理を簡単にするためKWT, DIFのサイズを、コア上のページ単位に分割したので、ファイル設計の方で、デバイスとのかねあいがうまくいかなくなり、ファイル・スペースの無駄を生じた。
この事はMONITOR-Vデータ管理の機能とも関連している。
- (ハ) COMBINEのルーチンは現在は2項演算しか許していないが、スタッキングを行う事により将来は拡張可能である。
- (ニ) ヒストリ・ファイルに書き込む情報及びそれをワーク・ファイルに書いておく時点、処理方法はもっと詳細に検討する必要がある。
- (ホ) さき程ものべた通りオンライン・アップデートは追加設計をした機能なので詳細についてはなお検討が必要である。

以上各処理部分毎に反省点を列挙してみたが、これはお互いからみ合っている要素である。それだけに、これらの矛盾点の解決は難しいことであるが、今後実際の運転を通して少しでも解決への糸口をみつきたい。

あ と が き

このシステムはLockheedのPalo Alto Research Laboratoryの On-line retrieval system DIALOGを参考にしている。

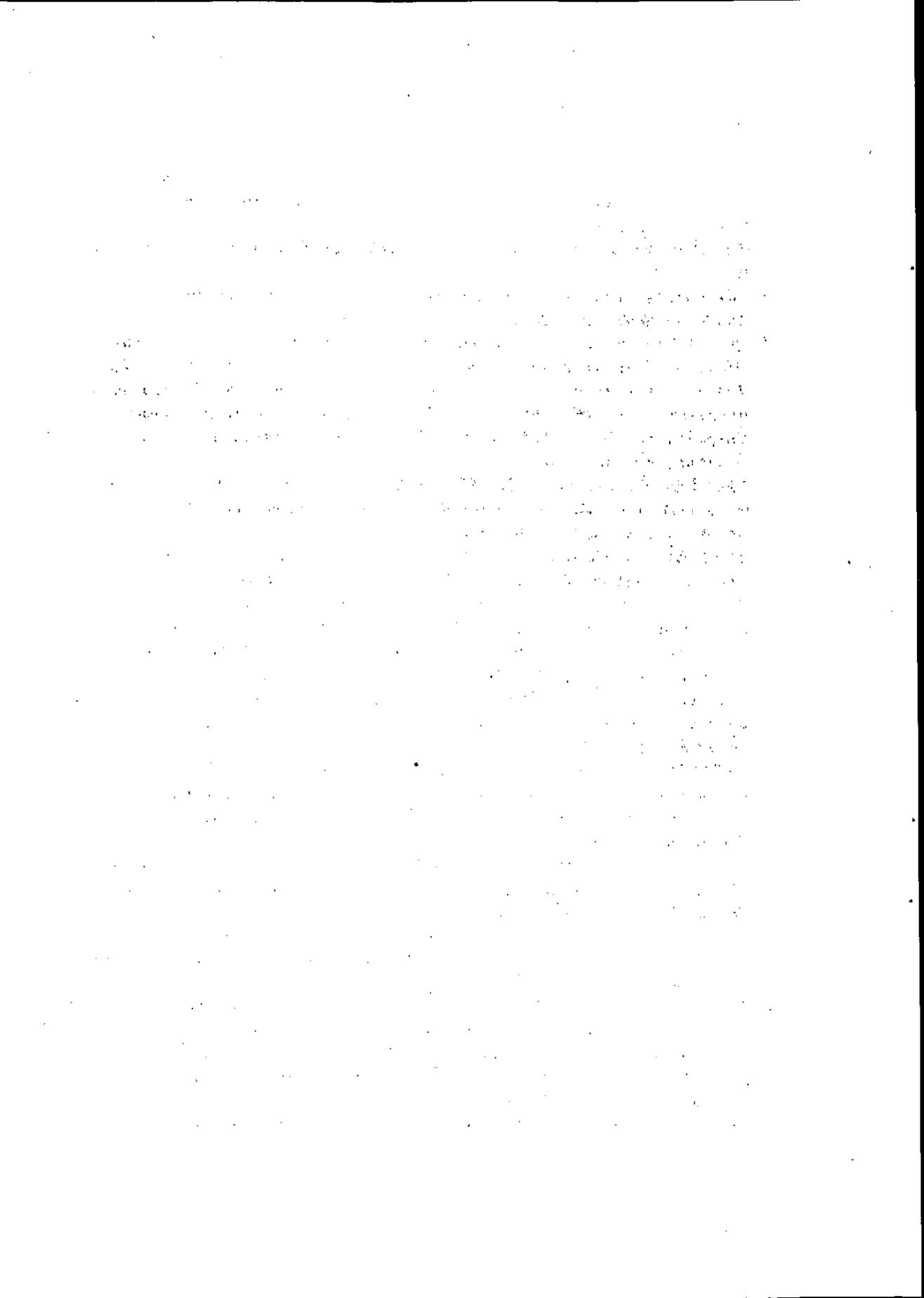
最近, MITのTIP, SDCのBOLD等をはじめ, かなりオンライン文献検索システムが実用に使われているようである。

文献検索の要点は, 検索内容の質の向上, 検索時間の減少それと経済性であるという。オンライン・システムとバッチ・システムと比較して, どちらが有利であるかということは一概には言えない。むしろシステムの利用目的からいづれかが選ばれるべきであるし, 経済性という面からいえば, 一般にリアルタイム性の強いもの程高価であるといえる。しかし, バッチはまたバッチゆえの無駄も多いだろう。

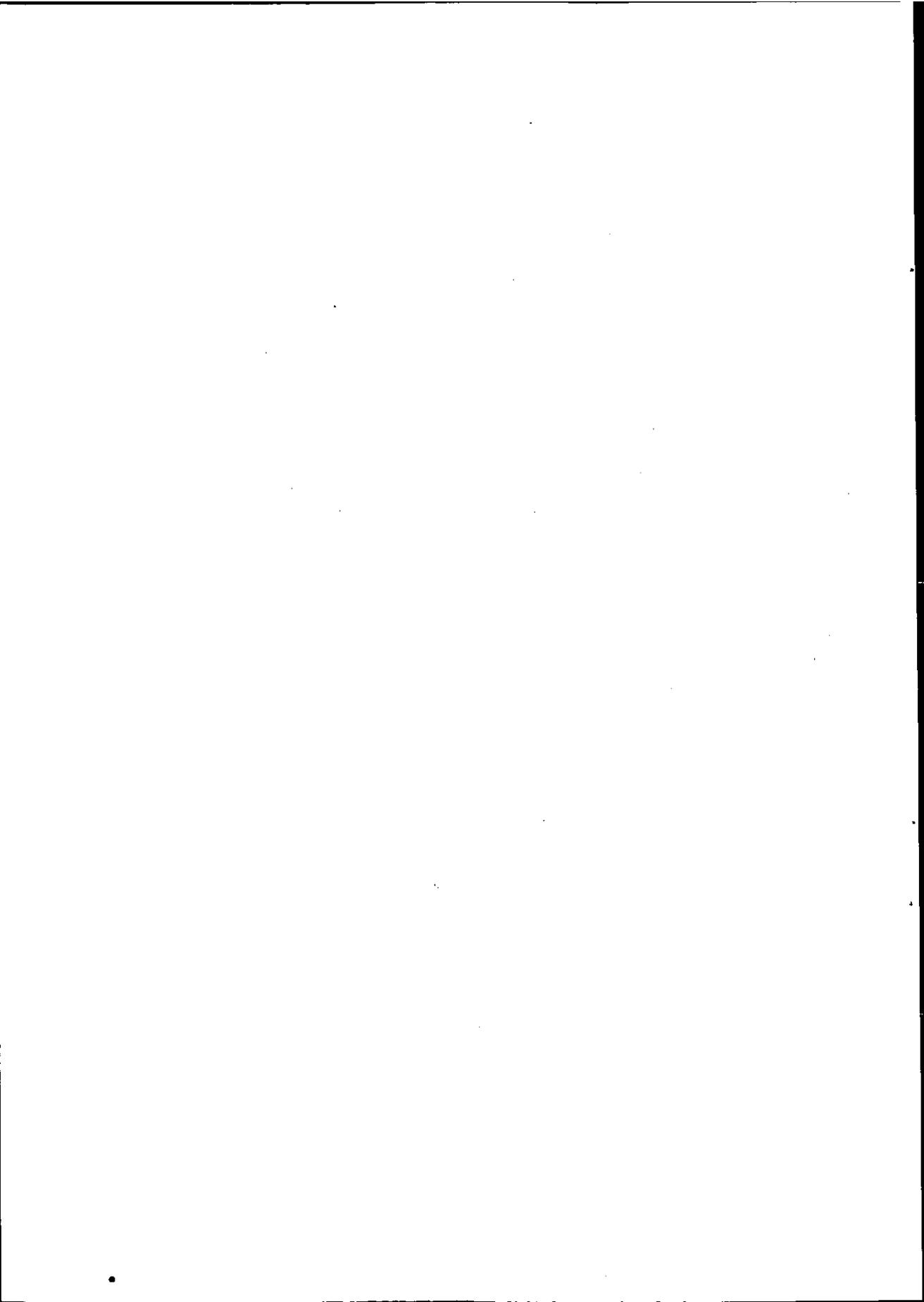
今後, 大規模のデータ・センター, あるいはデータ・バンクが実現する可能性からみて, その基礎または初期段階として, オンラインIRの研究, 開発は大いに意味のあることと思われる。

参 考 文 献

1. DIALOG: An operational on line reference retrieval system. ROGER K. SUMMIT, Lockheed Palo Alto Research Laboratory, Proceedings - A. C. M. National Meeting, 1967
2. The MIT Technical Information Project. Kessler, M.M., Pysics Today, March, 1965, p. 28
3. Information retrieval. A Critical View, Edited by George Schecter, Thompson Book Company, Washington D. C., 1967
4. Use of Tree Structure for Processing Files. Edward H. Sussenguth Jr., Harvard University, Cambridge, Mass., C.A.C.M., Vol. 6, No. 5/May, 1963
5. Treating hierarchical data structure in the SDC time-shared data management system (TDMS). Robert E. Breier, System Development Corporation, Santa Monica, California, in Proceedings - A. C. M. National Meeting, 1967
6. The SMART Automatic Document Retrieval System - An Illustration. G. Salton and M.E. Lesk, Harvard University. Cambridge, Mass., in CAOM, vol. 8, No. 6/June, 1965
7. BARONET: a simplified CRT-based information storage and retrieval system. John W. Young, Jr., The National Cash Register Co., Hawthorne, California 90250, in Proceedings - 1968 ACM National Conference
8. An algorithm for retrieving indexed documents and its application. Allan J. Humphrey and Shelby L. Brunelle, Chevron Research Co., Richmond, Calif., in Proceedings - A. C. M. National Meeting, 1966
9. Data Manipulation and Programming Problems in Automatic Information Retrieval. G. Salton, Harvard Univ., Cambridge, Mass., in CAOM, Vol. 9, No. 3/March, 1966
10. DATAPLUS: A language for real time information retrieval from hierarchical data base. Norman R. Sinowitz, Bell Telephone Laboratories, Holmdel, New Jersey, in Spring Joint Computer Conference, 1968
11. GIPSY: A generalized information processing system. Giampaolo del Bigio, International Atomic Energy Agency, Vienna, Austria, in Spring Joint Computer Conference, 1968
12. DATA FILE TWO: A data storage and retrieval system. Reuben S. Jones, General Electric Company Phoenix, Arizona, in Spring Joint Computer Conference, 1968
13. Experimental Evaluation of Information Retrieval Through a Teletypewriter. Morris Rubinoff and others, University of Pennsylvania, Philadelphia, Pennsylvania, in CAOM, Vol. 11, No. 9/Sept. 1968
14. The IBM Information retrieval center - (ITIRO) system techniques and applications. Samuel Kaufman, IBM Technical Information Retrieval Center, Yorktown Heights, New York, in Proceedings - ACM National Meeting, 1966



5. タイムシェアリング・システム
COBOL



5. タイムシェアリング・システム COBOL

謝 辞

以下の謝辞はCODASYL (Conference on Data System Language)の要求により米国防省COBOL-1965から引用したものです。

“COBOL-1965”はCODASYL COBOL委員会とECMA (European Computer Manufacturers Association)の努力で作られたものです。初めに、説述や内容はCODASYL COBOL出版小委員会のメンバーによって開発されたものです。

“COBOL”は産業言語であり、いかなる団体や団体のグループ、いかなる組織や組織グループにも迎合するものではありません。

プログラミング・システムや言語の正確さや機能についていかなる寄与者COBOL委員会にもいかなる形でも保証するものではありません。さらにこれらに関していかなる寄与者や委員会は責任を負うものではありません。

COBOLの保全に対する手続は確立されています。変更に関する手続の照会はCODASYLの理事会に指示されることが願われます。

本書に使用されている資料FLOW-MATIC (Sperry Rand Corporationの登録商標)、Programming for the Univac(R) I and II, Data Automation Systems (1958, 1959年版, Sperry Rand Corporation); IBM Commercial Translator Form № F28-8013 (1959年版IBM); FACP DSI 27 A 5260-2760 (1960年版 Minneapolis-Honeywell)の各団体および著作権所有者はCOBOL仕様中にその一部又は全体を使用することを認可しています。

このことはプログラミング・マニュアルや刊行書を複製したり使用する場合にも適用されます。

いかなる組織体もCOBOL報告書と第一回仕様書を複製すること、その報告書からアイデアを利用すること、又その報告書をインストラクション説明書に利用すること、その他いかなる目的にも利用することができます。

しかし、その時には、その刊行物の序の一部にこの章を記載しなければなりません。書評として短い文章に引用する場合には全文を載せることは不必要ですが、資料についての謝辞においてはCOBOLに言及することが必要です。

5.1 概 要

NEACシリーズ2200 TSS COBOLはオペレーティングシステムMOD III/IV COBOLコンパイラをTSS用に改造して作成されたものであり、TSS COBOLとしては比較的水準の高い言語を使用することができる。更に次のような特徴をもっている。

- (1) ソースプログラム入力時に会話モードで実行できる。

これは入力される行毎にシンタックス チェックを行いエラーが検出されると端末へメッセージを出力しオペレータの修正を待つ。

- (2) 一度入力されたソースプログラムは完全に保存される。

コンパイルされるとソースプログラムはソースプログラムファイルとして保存され、更新はUPDATE コマンドにより行なわれる。

- (3) TSSモードでテストランされる。

コンパイルされたCOBOL オブジェクトプログラムでは全てのファイルがTSS側のディスクファイルまたは端末装置を仮想して作成されている。従ってTSS側でテストラン、デバックが行なえる。

- (4) デバック済みのプログラムは修正せずにバッチモード側へ移行される。

デバックの済んだプログラムは修正せずにそのままバッチモード側へ再コンパイルすることにより使用される。

すなわちTSS COBOL コンパイラとバッチモード側のコンパイラはリースプログラム上完全な互換性がある。

- (5) タイムシェアリングシステムの特徴である Turnaround time の短縮 マシン コミュニケーション、システムの高効率な活用全てTSS COBOL コンパイラは充しているといえる。

5.2 システム構成

図5-1のようなシステム構成になっている。このシステムはNEACシリーズ2200 モデル500とモデル200の2つの中央処理装置から構成されている。

モデル500が主体で、モデル200は端末装置とモデル500のコミュニケーションサービスを処理する。

モデル500 CPUは393 k字のコアメモリがあり記憶保護機構によりスーパーバイザが多重

プログラミング処理を行う。

従って複数の独立したユーザプログラムおよびコマンドプログラムが他のプログラムを乱すことなく実行されている。

モデル 500 とモデル 200 は On-Line-Adapter (OLA) を通して結合されており、コントロール情報はすべてこのOLAを通して行なわれる。

端末装置 (リモート・ターミナル) はモデル 200 に接続されているMCCU (多重通信制御部) と通信回線で結ばれており、コントロールはモデル 200 内のコミュニケーション I/O スーパーバイザが行なっている。

5.2.1 コンパイル時に於けるシステム構成

図5-2がCOBOL ソースプログラムをコンパイルする際に使用されるシステム構成である。

ソースプログラム入力装置としてはデータステーションに接続されている。カードリーダー、紙テープリーダー、またはキーボードからか、すでにディスク上に登録されている。

ソースプログラムファイルいずれの装置からでも入力される。ディスク上にソースプログラムが登録できるのはINPUT コマンドを用いる場合とコンパイラが作成する場合がある。コンパイラが作成するのは端末 (リモートターミナル) からソースプログラムを入力した場合である。

コンパイラはコンパイル結果をオペレータの指示による (コマンドのパラメータで指定) 装置に出力する。(但しリロケータブルファイルは違う)

端末に出力する場合はキーボードプリンタに出力され、他の場合はディスク上のユーザエリアに出力される。オペレータはこれをコンパイル終了後どこに打出すかは別のコマンドで指定する事により結果が見られる。

リロケータブルファイル (ソースプログラムのオブジェクト) は必ずディスク上のユーザエリアに出力される。

もしソースプログラムに致命的エラーがあった場合は自動的に消される。

GOBOL コンパイラの必要コアメモリは 64k 字である。

5.2.2 オブジェクトプログラム実行時に於けるシステム構成

GOBOL オブジェクトプログラムで取扱える装置はディスクおよび端末装置である。

図5-3がそのシステムである。

オペレーティングシステムMOD3/4のGOBOL コンパイラLとTSS COBOL コンパイラは言語上互換性があり、GOBOL コンパイラLで取扱える装置はカードリーダを除いてすべてディスク上にシミュレートされる。

カードリーダを使う端末(リモートターミナル)のカードリーダが使われる。

しかし、実行時にファイルの装置を変更する事は可能になっており(ASSIGN コマンドによる)融通性のあるシステムになっている。

図5-1 システム構成

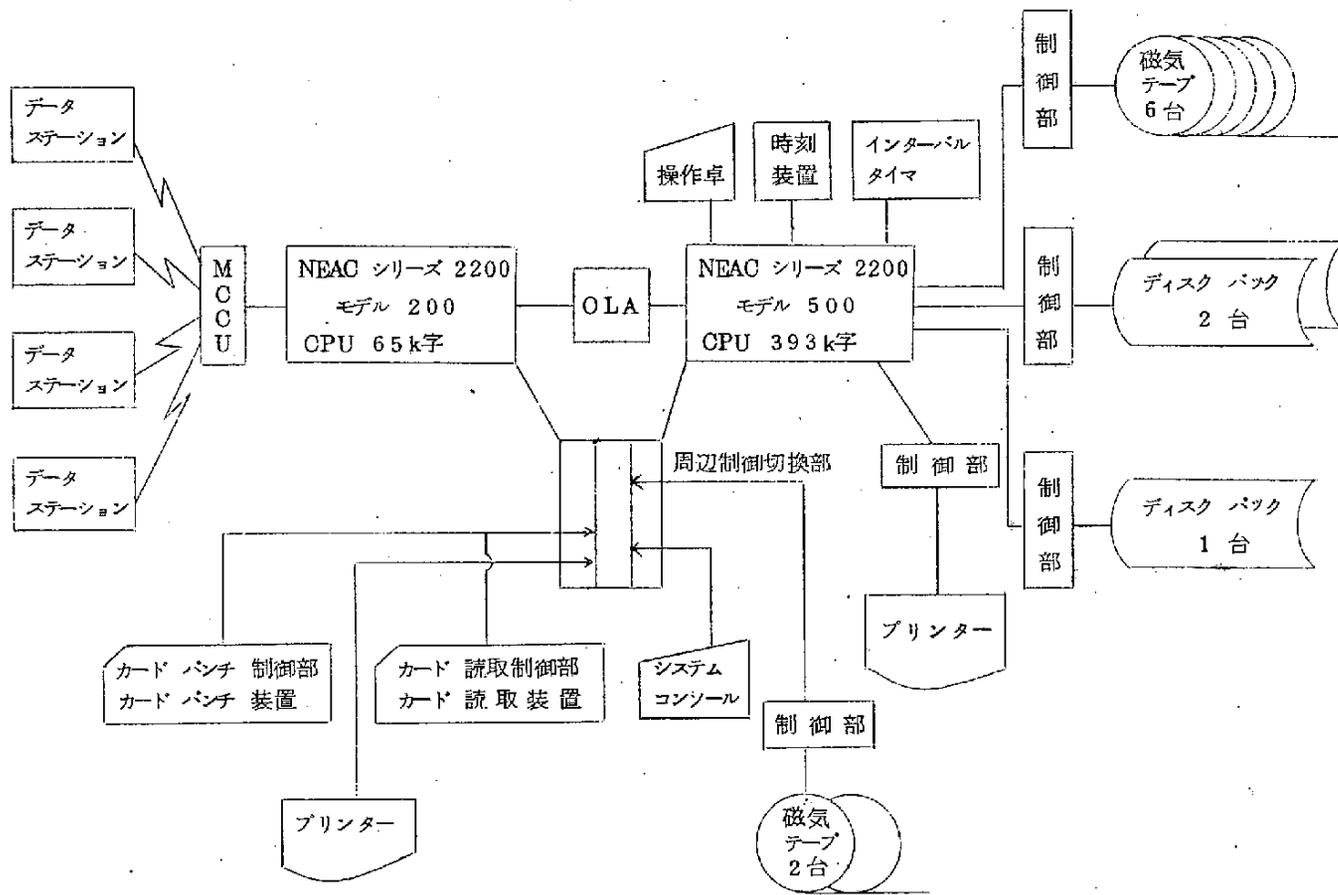


図5-2 コンパイラ システム構成

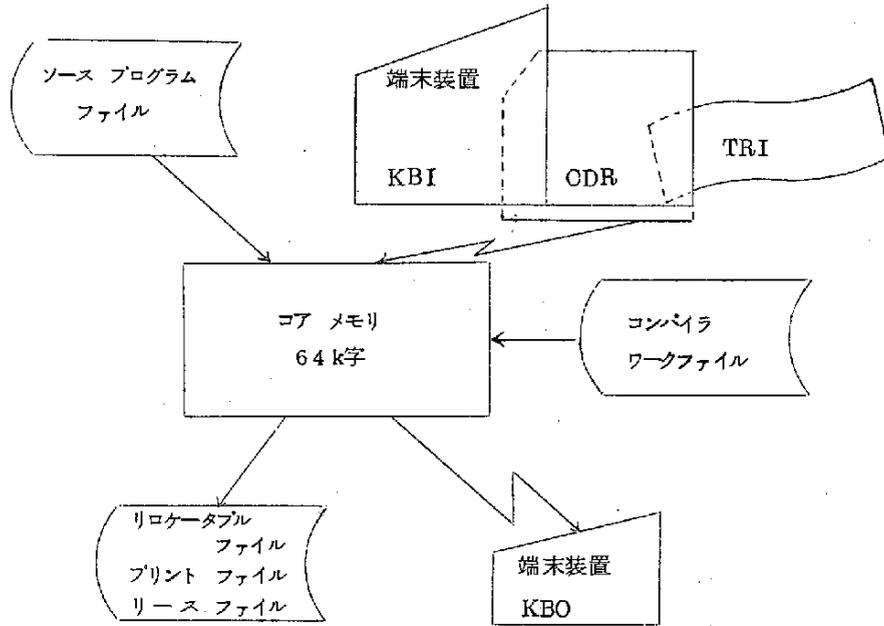
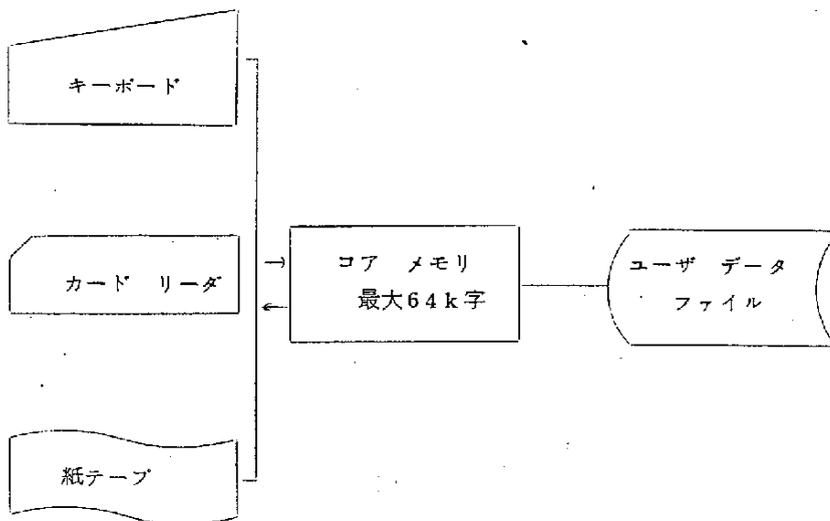


図5-3 オブジェクト プログラム集行時のシステム構成



5.3 スーパーバイザ コマンドおよびファイルシステム

5.3.1 スーパーバイザとコマンドプログラム

スーパーバイザとは一般にモニタともいわれるものでシステム全体の管理を行うコントロールプログラムである。TSS スーパーバイザは

- (1) スケジューラ
- (2) IOFCS (Input Output File Control System)
- (3) コミュニケーション I-O コントロール

により構成されている。

コマンドプログラムは端末のユーザ(オペレータ)とスーパーバイザのコミュニケーションのために用意されているもので、次のように分類される。

- (1) システムとの会話の開始、終了のコマンド
- (2) ファイルに関するコマンド
- (3) プログラム実行を制御するコマンド
- (4) ファイルやシステムの間合せコマンド

スーパーバイザとコマンドプログラムの関係を図示すると図3-1のようになる。

1. スーパーバイザ

スーパーバイザは NEAC シリーズ 2200 モデル 500 モデル 200 のコアに常駐するコントロールプログラムで、次のような機能がある。

- (1) 端末(リモートターミナル)とのコミュニケーションの制御
- (2) 端末へのサービスのスケジューリング
- (3) システムリソースの割当てと制御
- (4) 多重プログラミングの制御
- (5) 入出力装置に対する入出力要求のスケジュールと入出力動作の制御
- (6) ファイルの統一的な管理

現在可動中の NEAC TSS における多重プログラミングの多重度は

Foreground Program は最大 N ヶ (任意)

Background Program は最大 2 ヶ

Foreground側とはTSSモニターで管理されている方で、Background側はセンターのバッチ処理を行うためNEACオペレーティングシステムMODモニターにより管理されているプログラムである。モデル500内のコア状態は図3-2で示す。

2. スケジューラ

スケジューラはスーパーバイザの中核であり、同時にアクセスする多数のユーザに対しどのような順序でシステムの機能を割当てるか、またどれだけサービスする時間を考えるかなどのスケジュールを管理する。

スケジューラの機能を項目別に上げると

- (1) ユーザとシステムの会話の開始、終了の処理
- (2) コマンドの受付、時間割当てのスケジュール
- (3) システムリソースの割当てとその管理
- (4) プログラムの実行開始、終了の処理
- (5) ユーザへのサービスの一旦停止、一日のサービスの終了あるいはサービスの強制中断などの処理

以上のような機能である。

3. IOFCS

IOFCSは入出力装置に関する動作のコントロール及び多重プログラミングのコントロールを行っている。

IOFCSにおける一番重要な機能はファイルの統一的な管理であり、多数のユーザが同時にシステムを使用していても各ファイルはプライバシーが確保され、またあるファイルは共同利用ができるようになっている。ユーザがファイルにアクセスする場合はファイル名とファイルクラスによって簡単に行なえるようになっている。

IOFCSの機能を項目別に上げる

- (1) ファイルの保護、共用等の管理
- (2) 入出力装置に関する全ての管理
- (3) 中央処理装置と入出力動作をオーバーラップさせCPUタイムを有効に利用し、システム効率を上げる。
- (4) プログラムのスイッチングを行い、多重(マルチ)プログラミングの管理を行う。
- (5) モデル200とのコミュニケーションを行う管理をする。

4. コミュニケーション I-O コントロール

このスーパーバイザはモデル 200 に常駐するコントロールプログラムで端末とモデル 500 とのコミュニケーションを管理している。

機能は次のようなものである。

- (1) 端末(リモートターミナル)のユーザからの会話要求をスケジューラに知らせる
- (2) 端末に対する入出力動作のコントロールを行う
- (3) モデル 500 とのインターフェースをとる

5. コマンドプログラム

コマンドはスーパーバイザとのコミュニケーションのために用意されているターミナル言語であり、コマンドによって簡単にユーザはシステムと会話する事ができる。

コマンドはシステムに登録されておりスケジューラが全て管理している。

現在、27種類のコマンドが用意されているが、ユーザがコマンドプログラムを作成しコマンドライブラリに登録する事により追加していく事が可能となっている。

各コマンドにはパラメータを指定するようになっている。したがって融通性のあるコマンドプログラムとなっている。

コマンドを使用目的によって分類すると

- (1) システムとの会話の開始、終了のコマンド
- (2) ファイルに関するコマンド
- (3) プログラム実行を制御するコマンド
- (4) ファイルやシステムの間合せコマンド

以上のように分類され"COBOL"コマンドは(3)のプログラム実行を制御するコマンドの中に入りCOBOL ソースプログラムをコンパイルする。

NEAC TSS で現在までに登録されているコマンドを表3-1に示す。

6. COBOL コマンド

COBOL コマンドはCOBOL ソースプログラムをコンパイルしオブジェクトリロケータブルファイルを作成する。

コマンドの形式は

COBOL FN1, -MODE-, -IDV-, -ODV1-, -ODV2-, -ODV3-, -ODV4-, -TRC-

となっている。各パラメータにはそれぞれ次の意味がある。

- FN1 : ソースプログラムのプログラム名を指定する。ソースプログラムファイル(SPF)から入力する場合はSPFのファイル名を指定する。
このパラメータは必ず指定しなければならない。
- MODE- : Compile modeを指定する。MOD1, MOD3, NEAC 省略時の解釈はMOD3である。
- IDV- : ソースプログラムを入力する装置を指定する。
KBI 端末のキーボードから入力する。
CDR 端末のカードリーダーから入力する。
SPF ディスク上にあるソースプログラムを入力する。
TR1 端末の紙テープから入力する。
指定がない場合は“KBI”と見なされる。
- ODV1- : コンパイル中に出力される診断メッセージを出力する装置を指定する。
- ODV2- : ソースプログラムリストを出力する装置を指定する。
- ODV3- : データのメモリーマップを出力する装置を指定する。
- ODV4- : PROCEDURE DIVISIONのオブジェクトコードリストを出力する装置を指定する。
- ODV1からODV4まではいずれの場合も指定がなければそのオプションは不用と見なされ、リストは出力されない。
- 装置を指定する場合は次のいずれかである。
- KBO : 端末のキーボードに出力される。
LNP : 端末のプリンターに出力される。
端末にプリンターのない機器構成の場合は磁気ディスク上にプリントイメージを出力する。
- TRC- : オブジェクトプログラムのトレース機能を使う時に指定する。

図3-1 スーパーバイザとコマンドプログラム

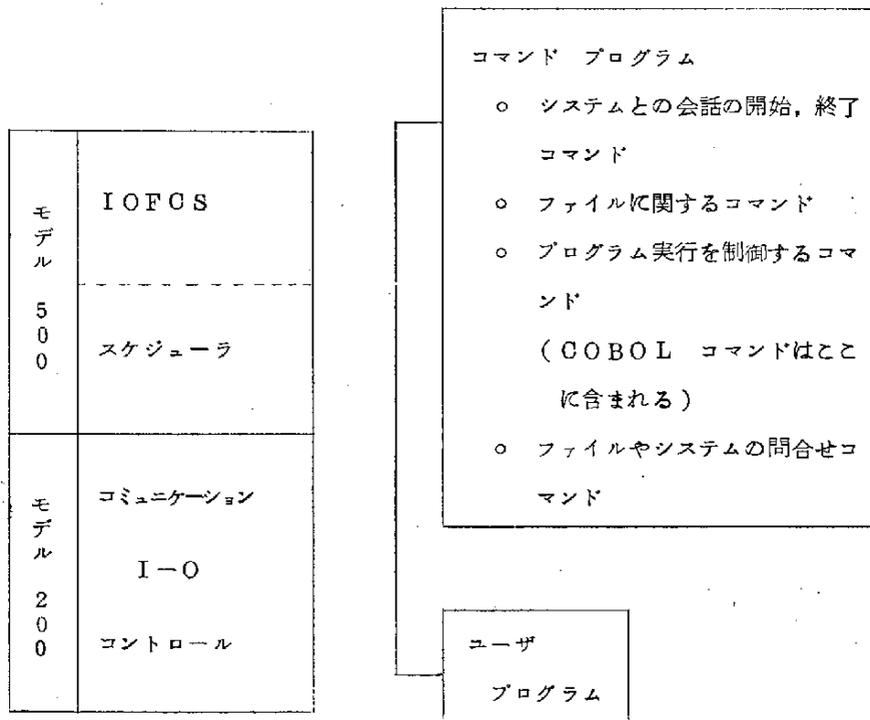


図3-2 モデル500コア状態

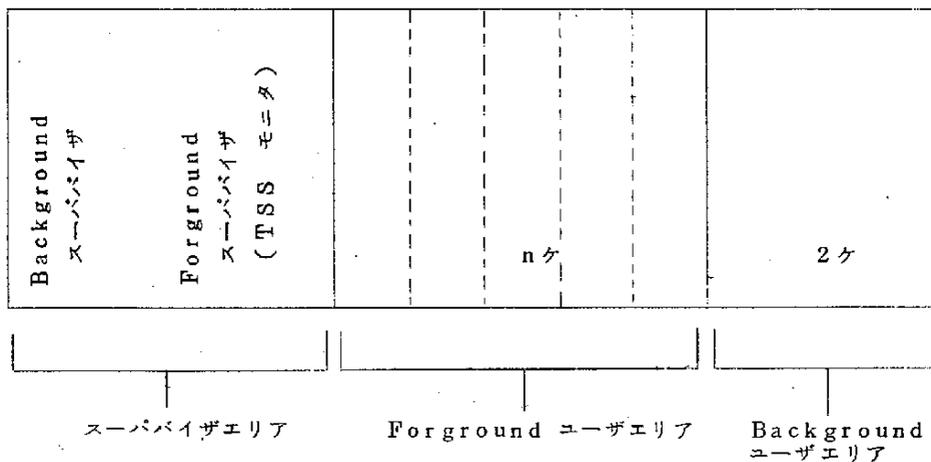


表3-1 コマンド プログラム

	コマンド	機能
会話の開始、終了に関するコマンド	HELLO BYE INPUT UPDATE OUTPUT COUTPUT RENUMBFR	システムとの会話を開始する。 システムとの会話を終了する。 ファイルを作成する。 ファイルを修正する。 ファイルを端末に出力する。 ファイルをセンターのプリンタに出力する。 ファイルのライン番号をつけ直す。
ファイルに関するコマンド	COPY SAVE LOAD RESERVE RENAME MODECHANGE PURGE PULL	同一の内容のファイルを新たに登録する。 ファイルを磁気テープに保存する。 磁気テープに保存されたファイルをディスクに格納する。 ファイルを登録する前にエリアを確保する。 ファイル名を変更する。 ファイルのモードを変更する。 ファイルを消去する。 別のグループのファイルを利用可能にする。
プログラム実行を制御するコマンド	COBOL FORTRAN LINK LNKS START RESTART END ASSIGN	COBOL ソースプログラムをコンパイルする。 FORTRAN ソースプログラムをコンパイルする。 プログラム間のリンケージをとる。 COBOL リロケータブル プログラムおよび他のプログラム間のリンケージをとる。 プログラムを実行させる。 プログラムを再実行させる。 プログラムを中止する。 プログラムで指定した入出力装置を変更する。
ファイルやシステムの問合せコマンド	ASK LIST INQUIRY TEACH	ファイルに関する情報を問合せる。 登録されているファイルの一覧表を要求する。 会話の状況を問合せる。 登録されているコマンドに関する情報を問合せる。

5.3.2 ファイル システム

タイムシェアリングシステム (TSS) では同時に多数のユーザがシステムを共用している。したがって各ユーザファイルの保護およびユーザ間でのファイルの共用等ファイルを統一的に管理しなければならない。

現在のNEAC TSSではユーザがアクセスできるI/O装置は端末とマスメモリーファイル (MMF 現在はディスク) である。

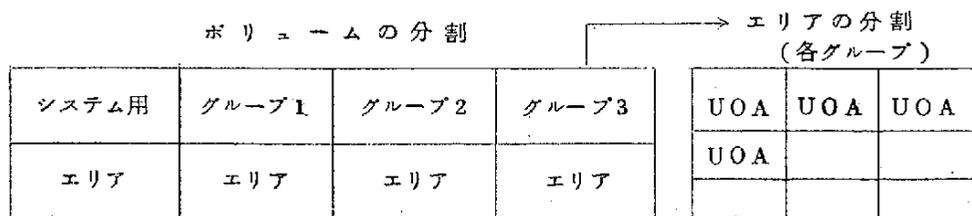
ここではMMF上におけるファイルシステムの概略を示す。

MMF ボリューム (ディスクの場合1パック) は

- (1) システム用エリア
- (2) 各グループ用エリア

に分割される。

各エリアは、さらに UOA (Unit Of Allocation) と呼ばれるエリアに分割される。



各ファイルはこのUOAの集まりとして構成され、1つのUOAはただ1つのファイルにのみ属し、他のファイルと共用される事はない。

グループ内でファイルとして使用されていない。UOAはタミー UOAとしてグループ毎にプールされている。

ファイルの登録消去およびUOAの割当はファイルシステム (IOFGS) で統一的に管理しており、ファイルシステムは各ボリュームにあるディレクトリーを管理することにより行なわれる。

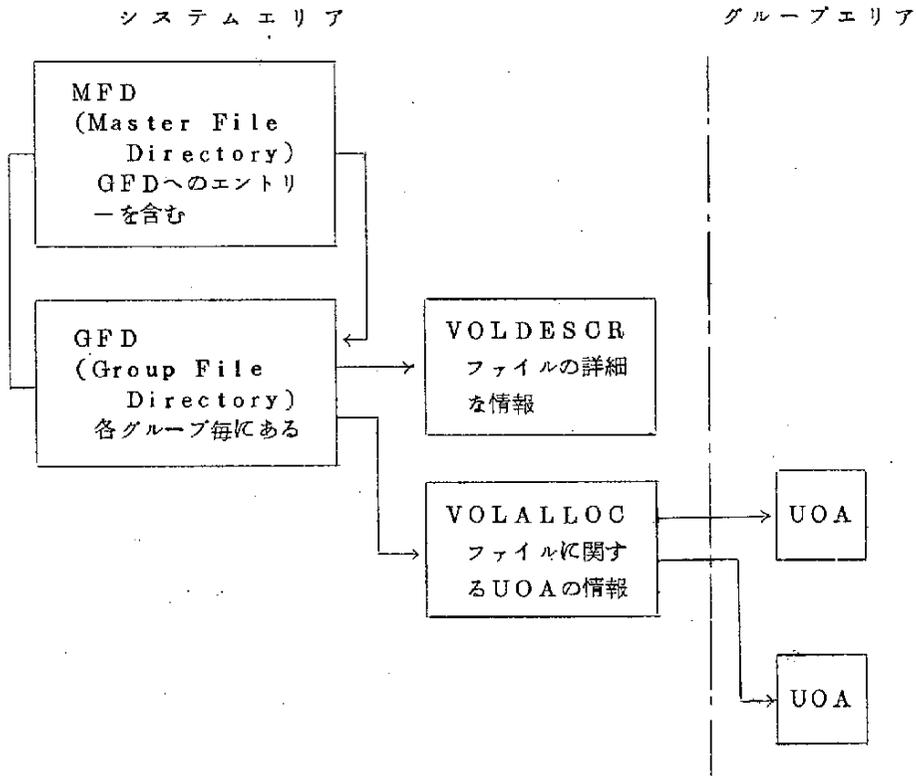
ディレクトリーはシステムエリアにあり

- (1) VOLNAMES (MFD GFDの2ケのディレクトリーがある)

(2) VOLDESCR

(3) VOLALLOC

の3つのファイルで構成され下の図のようになっている。



NEAC TSSで取扱うファイル構造には2種類がある。

(1) シーケンシャル ファイル

(2) パーテションド シーケンシャル ファイル

シーケンシャル ファイルとはテープ上のファイルのように各レコードがシーケンスになっている形式である。

パーテションド シーケンシャル ファイルとはメンバーと呼ばれるグループがありそのメンバー内はシーケンシャル ファイル形式でアクセスする場合メンバーの第1レコードからアクセスできるようになっている。

COBOLのオブジェクトで取扱えるのはシーケンシャルファイル構造だけである。

ユーザが端末からファイルをアクセスする場合やユーザプログラムでアクセスする場合は、グループコード(HELLO コマンドで指定)とファイルクラスとはそのファイルにどのようなデータが含まれているかによって分けられるもので、

- (1) Symbolic (S) COBOLやFORTRANのソースイメージを含むファイル
- (2) Update (U) ソースイメージファイルを修正する時に使うファイル
- (3) Relocatable (R) コンパイルされたプログラムでリロケータブルファイル
- (4) Object (O) LINK, LNKSの出力するファイル
- (5) Data (D) データファイルやワークファイル

以上のクラスがある。

またファイルのプライバシーを確保するためにファイルにパスワードを付ける事ができる。このパスワードは各ユーザ毎に持ち、システムで管理され、パスワードチェックによりファイル作成者のみアクセスできるようにされている。

5.4 COBOL コンパイラの構成

コンパイラはインタフェース部分と大きく分けて5つのフェーズで構成されている。

コンパイラはソースプログラムを変換しLNKS コマンドで処理できるようなりロケータブルファイルを作成する。

5.4.1 フェーズ フロー

コンパイラはソースプログラムを変換する場合、ソースプログラムのシンタックス解析を行い中間言語に分解し、次のフェーズに渡し順次処理を続け、第4フェーズで完全な形式のりロケータブルファイルが作成される。

コンパイラの各フェーズにおける処理の概略は次のようになる。

フェーズ 1

COBOL ソースプログラムのIDENTIFICATION, ENVIRONMENT, DATA DIVISIONを読み

- (1) シンタックス解析を行い
- (2) ネームテーブルを作成する(これはフェーズ2で使われる)
- (3) DATA DIVISIONまでのコアの割当てを行いリロケータブル形式で出力する。

フェーズ 2

COBOL ソースプログラムのPROCEDURE DIVISIONを読み

- (1) シンタックス解析を行う。(COMPUTE IF UNTIL)
- (2) ソースプログラム上の複雑な表現を簡単なステートメントの形式に変換する。
- (3) フェーズ1で出力したネームテーブルを用いデータ名の構成(構造)を示す中間言語を作る。
- (4) 最適化(optimization)の処理を行う。

フェーズ 3

フェーズ2で出力された中間言語を読み

- (1) PROCEDURE DIVISIONのシンタックス解析を行う。
- (2) リロケータブル形式のオブジェクトコードを出力する。

フェーズ 4

セグメンテーションの処理を行う。

フェーズ 5

コンパイラの終了処理とエラー処理を行う。

フェーズフローの図を図5.4.1に示す。

5.4.2 モジュールリティ

COBOL コンパイラはモジュール化されておりモジュールの修正、入替によって機能の変更が容易に行なえるようになっている。

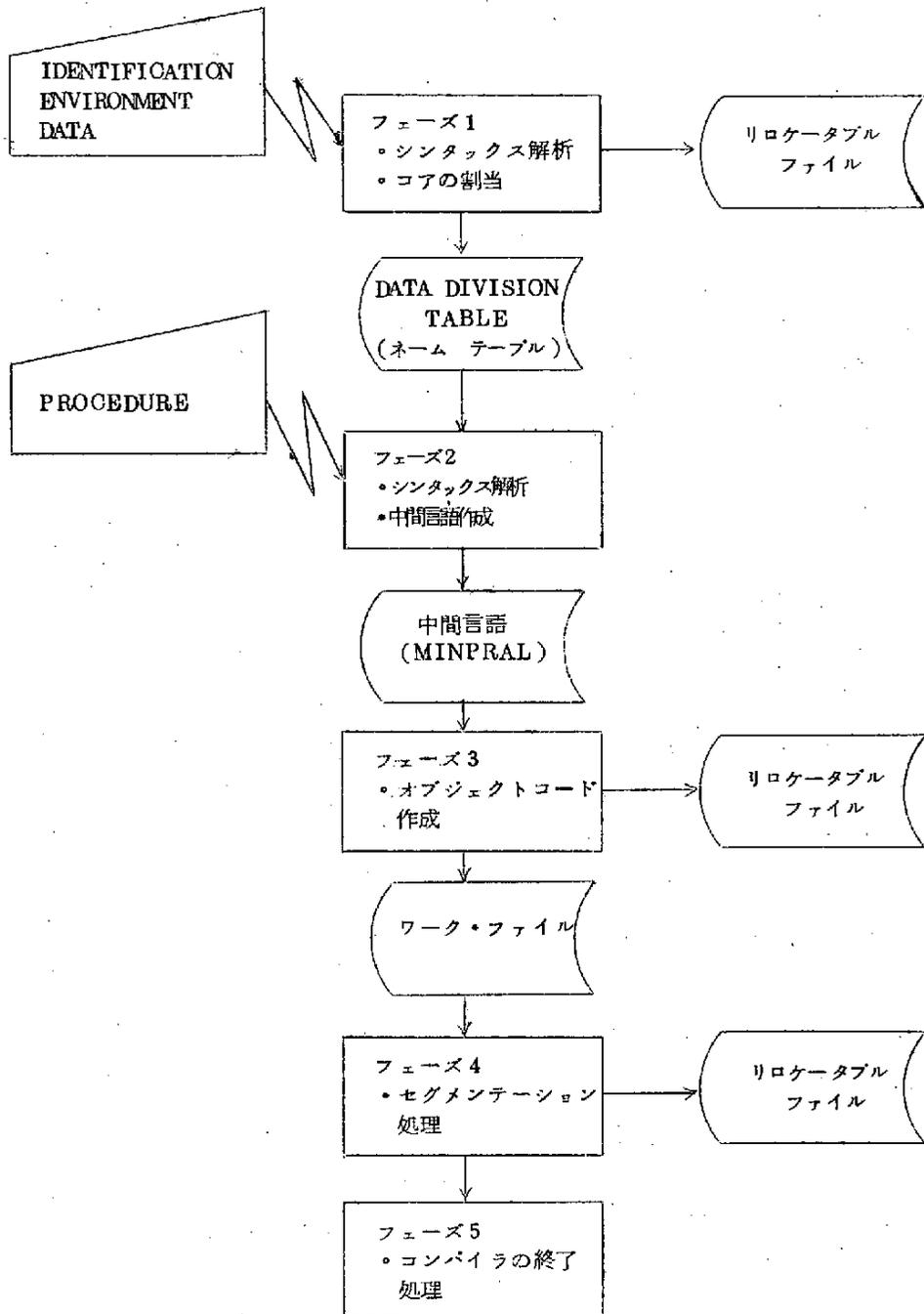
各モジュールは

- (1) インタフェース モジュール
- (2) メインモジュール

の2種類に分類される。

インタフェースモジュールはコンパイラのフェーズ間及びコンパイラとスーパーバイザ(モニ

図 5.4.1 フェーズ フロー



タ)特にIOFCSとのインタフェースとして使われ (OPEN, CLOSE, GET PUT) 必要なI-Oエリアがとられている。

メインモジュールはコンパイラのメインルーチンであり、インタフェースモジュールと結合 (リンク) されソースプログラムのシンタックス解析やオブジェクトコードを作成したりする。

5.5. コンパイラの詳細

TSS COBOL コンパイラは、TSS モニタのもとで働くコマンドプログラムの一種であり、COBOL ソースプログラムをコンパイルレオブジェクトプログラムを作る。このオブジェクトプログラムは LNKS コマンドでリンクロードされ、ジョブファイルに作りかえられ START コマンドによって TSS モードで実行される。

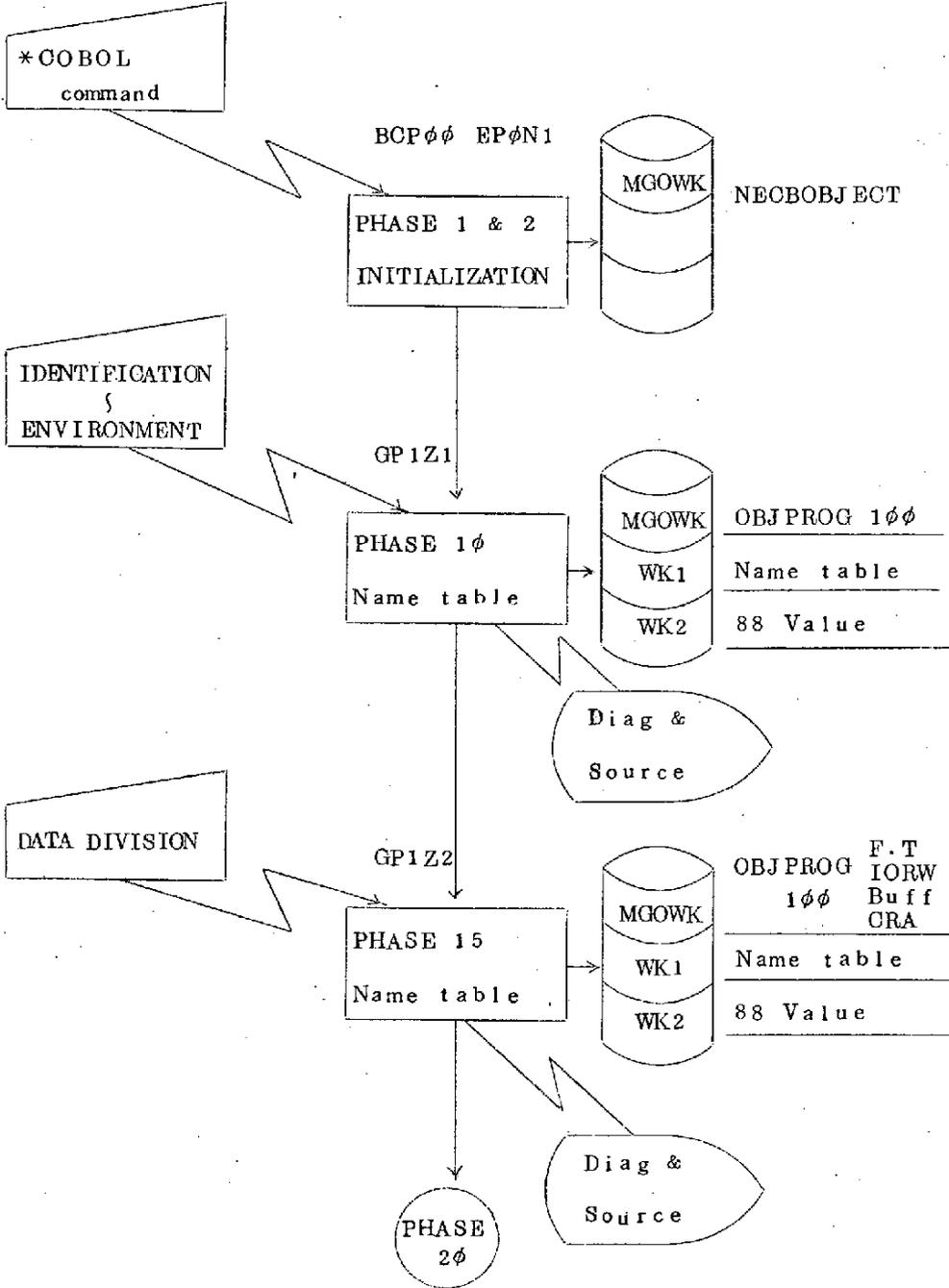
コンパイラにはソースプログラムをコンパイルする方法として次の2通りがある。

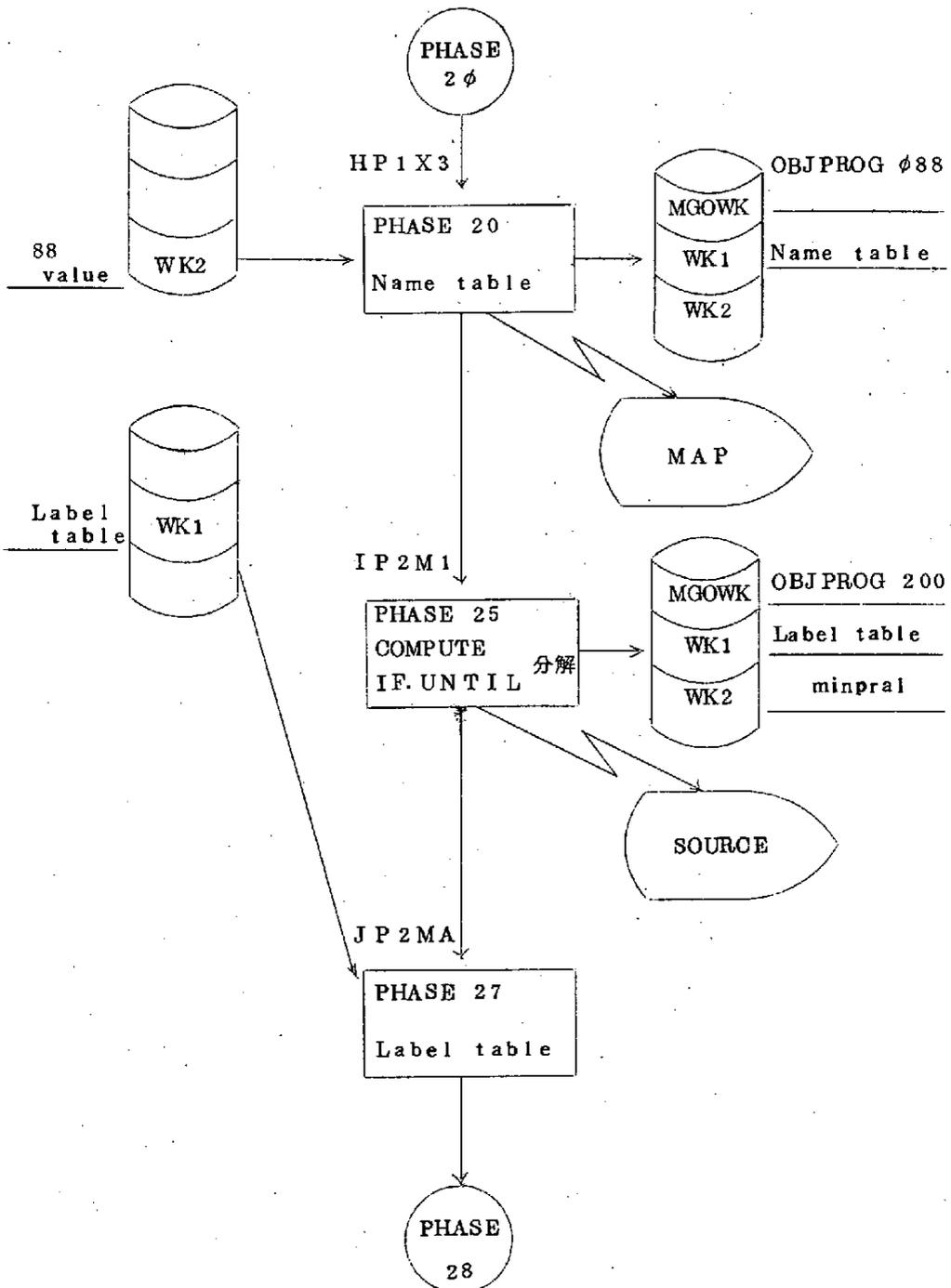
- (1) ソースプログラムを一度アセンブラ言語 (NEAC オペレーティングシステム ASSEMBLER L) におとす方法
- (2) ソースプログラムを直接マシン言語 (リロケータブルファイル) におとす方法

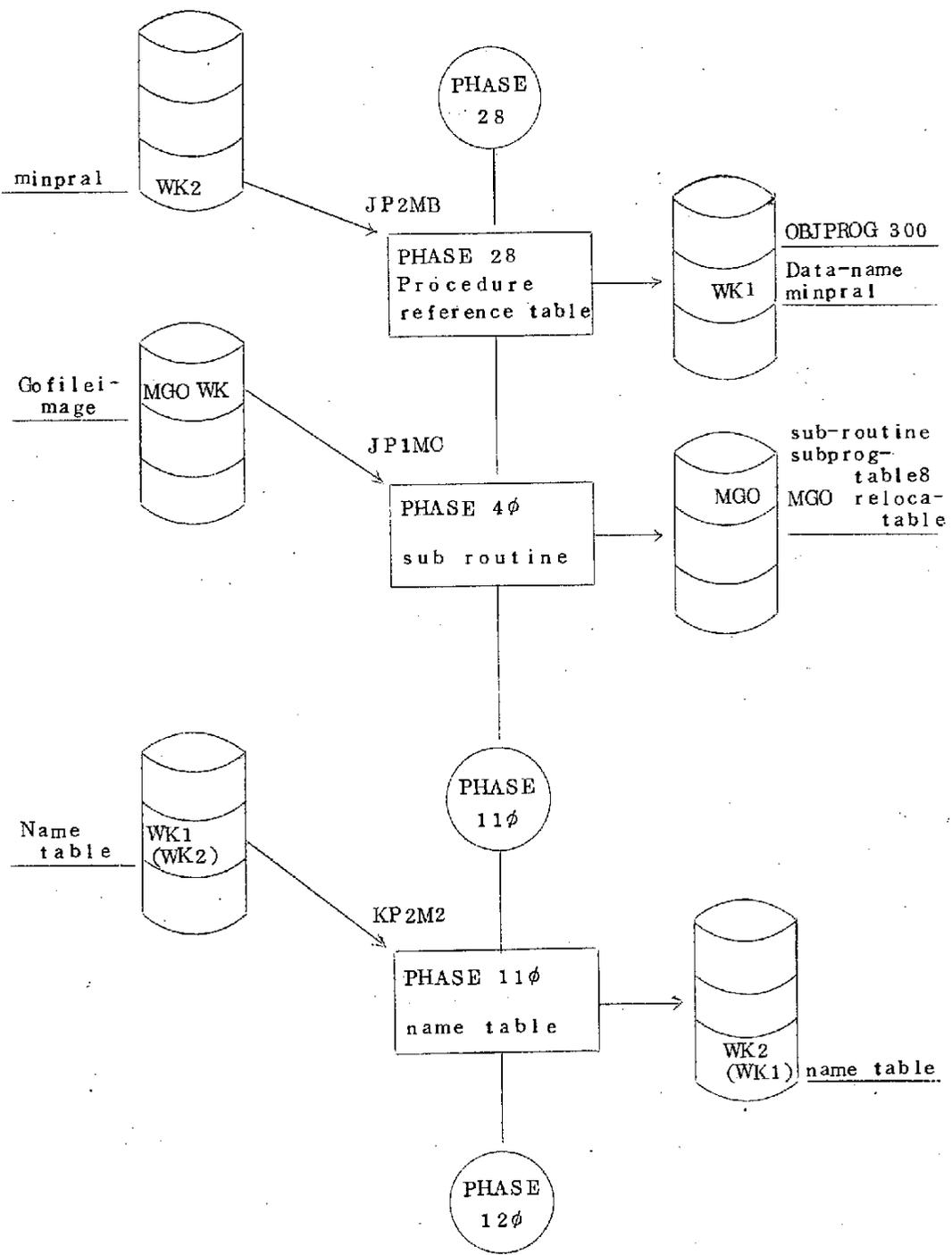
TSS COBOL コンパイラは(2)の方法で行っており、(1)の方法に比べコンパイル時間が短くTSSにむいた方法といえる。

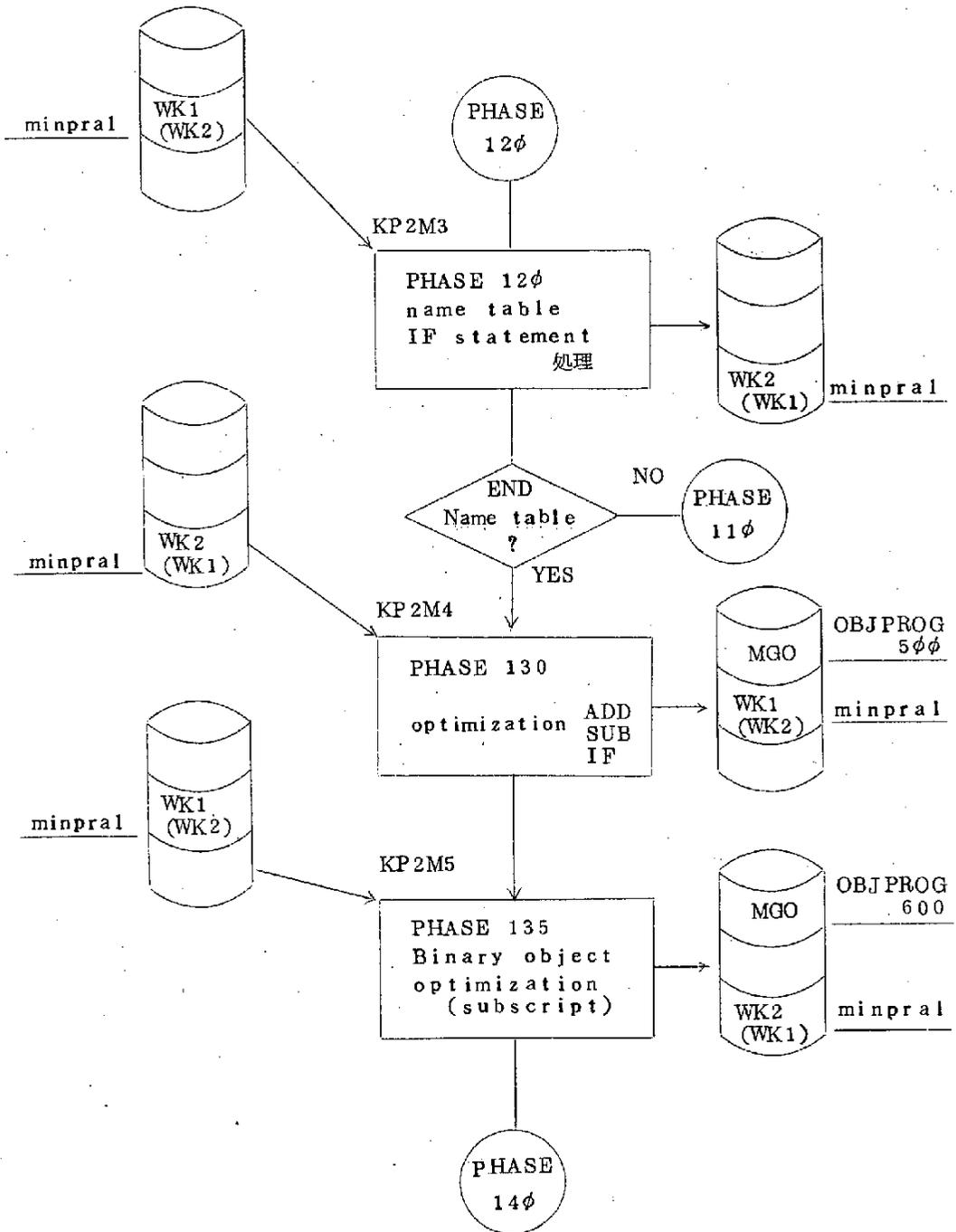
このTSS COBOL コンパイラはNEAC オペレーティングシステム MOD III/IV COBOL コンパイラLをTSS用に修正し作成されたものである。概説でも述べたようにコンパイラはモジュール化されており、モジュールの変更追加により簡単にシステムの機能拡張が可能となっている。

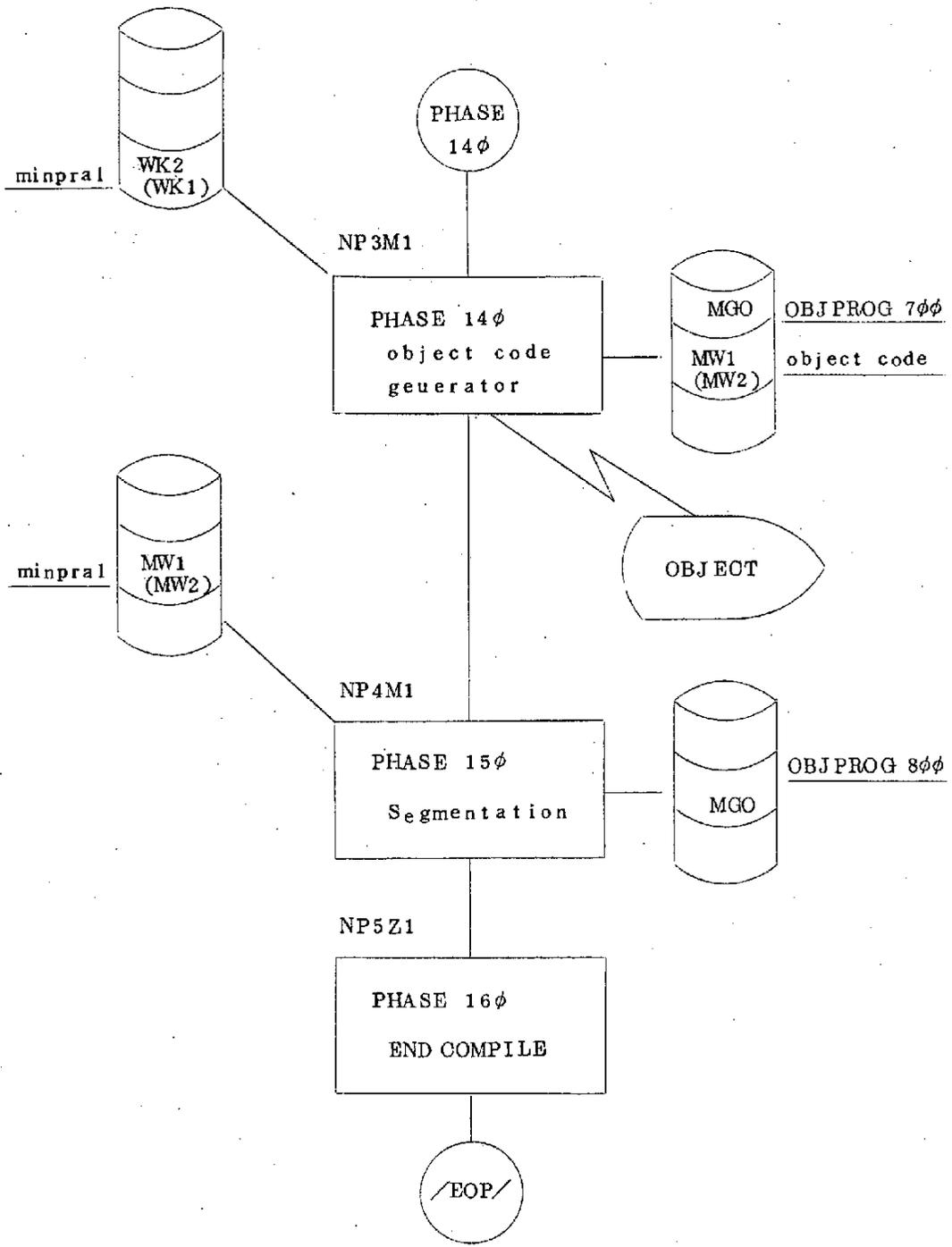
この章ではコンパイラを構成する各モジュールの機能とコンパイラの処理方法を詳しく説明する。この章でのフェーズ番号はすべて論理的なフェーズ番号を用いて説明している。











5.5.1 インタフェースモジュール

インタフェースモジュールではTSS モニタ(スーパーバイザ)とのコミュニケーションおよびコンパイラのフェーズ間のコミュニケーションを行う。インタフェースモジュールは約20コあり、各モジュール間のリンケージはリンケージシンボル、システムシンボルで行なわれている。

NECBSBCPφφ

このモジュールにはコンパイラのフェーズ間のコミュニケーションを行うエリアがある。コンパイラのコモンエリアであり、コンパイル終了までコアに常駐する。

NECBSBCPφ1

このモジュールもコンパイラのフェーズ間コミュニケーション用である。コマンドパラメーターで指定されたオプションを示すインディケータがある。

リンケージシンボル	
INDV/	コマンドで指定された入力装置を示す1桁のフィールド
MODE/	各種リストオプションを示すインディケータ
KBOS/	リストを出力する装置を示すインディケータ

他に二進十進変換ルーチン、十進二進変換ルーチン及びI-Oエラーが生じた際のエンتریポイントERRX/がありエラーメッセージをキーボードに出力するためフェーズ5にコントロールを渡すルーチンなどがある。

NECBSACPφ2

このモジュールはソースプログラムの入力をコントロールする。

実際の入力をIOFGSに要求するのはNECBSAKBINかNECBSASPIのモジュールである。このACPφ2では入力装置によりどのモジュールを使うか判定LGFTを要求するようコントロールしている。

メインモジュールからのエントリポイントはリンケージシンボルCSIU/で示される。CSIU/にコントロールが入るとソースプログラム1桁がこのモジュールにあるソースバッファ(リンケージシンボルRBUF/)にセットされメインモジュールに渡している。

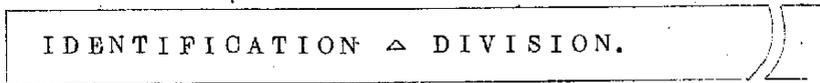
ソースプログラムの終了はカラム8からの“END Δ COBOL”で判定している。
 “END COBOL”が来たらフェーズ2のメインモジュールにコントロールを帰す。
 このモジュールのブロックチャートは図2-1で示す。

NECBSAKBIN

このモジュールは端末に対する実際の入出力動作をIOFCSに要求し、端末から読まれたソースプログラム1行又は1枚の内容をBCP領域にあるソースプログラムエリアに移送する。端末から入力バッファに入る形式は装置によって下図のようにになっている。

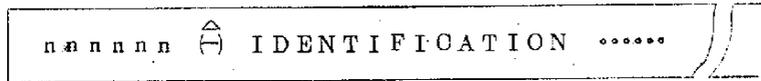
KBI(キーボード)の場合

バッファ

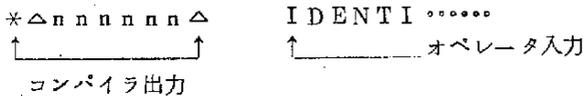


カラム8の位置から

GDR(カード)の場合



すなわち入力装置にKBIを指定した場合コンパイラの処理が会話形式(conversational mode compile)になり1行毎にシンタックス解析が行なわれるが、この時コンパイラは端末のオペレータへソースプログラムのシーケンス番号を打ち出してやるようになってくる。



ハイフンが使えずソースプログラムが2行にまたがる事はできない。

ソースプログラムエリアに移送する時はカードから入力したと同じ形式で移送される。

ブロックチャートは図2-2で示す。

NECBSASPMI

ディスク上にあるソースプログラムファイルから入力するルーチンである。ソースプログラムファイルはファイルクラス“S”で144 桁/レコード 5レコード/ブロック シー

クエンシャルファイル構造である。(ブロックチャートは図2-3)

ソースプログラムがディスクに作られるのはINPUT コマンドによる場合とCOBOL コンパイラが端末からソースプログラムを読んだ場合である。いずれの場合もディスク上の形式は一致しており次のような形式になっている

(1) CDR (カードリーダー)

カラム1~80	シーケンス番号	C	144桁
---------	---------	---	------

(2) CDR以外

カラム8~80	シーケンス番号	△	144桁
---------	---------	---	------

このモジュールのソースプログラムエリアへの移送方法は、端末のカードリーダーから読まれた場合と同じ形式で移送される。

すなわちコンパイル中のソースプログラムの形式はただ1つで端末のカードリーダーから入れた時の形式である。

NECBSAFTIN

このモジュールにはASPFIL, AKBINに共通なファイルテーブル, IORW (Input Output Request Word) バッファエリアがある。

コマンド パラメータの入力装置を調べるモジュール (NECBSEPØN2) で入力装置にあったようにファイルテーブルが変更されている。

初期状態は端末のキーボード用になっている。

NECBSACPØ3

COBOL コンパイラが出力するリストに関する全てのコントロールとソースプログラムファイルをディスクに出力するコントロールを行っている。

コンパイラが出力するリストは

- (1) ソースプログラムリスト
- (2) 診断メッセージ
- (3) データのコア・マップ
- (4) オブジェクトコードリスト

このモジュールでは実際のI-Oは要求せずNECBSAKBOU, NECBSACPRF, NECBSASPF0の3モジュールにやらせる。

このモジュールへのエントリーポイントはリンクージンボルOPRT/により行なわれる。同時にOPRA/, PADR/に必要な情報がセットされていなければならない。

OPRA/には出力したいエリアの左端アドレスを, PADR/にはソースプログラムおよびオブジェクトコードリスト用エリアの左端アドレスをそれぞれセットしておく。

このモジュールのブロックチャートは図2-4で示す。

NECBSAKBOU

端末のキーボードにリストを出力するモジュールである。

キーボードには

- (1) コマンド パラメータで指定されたリスト
- (2) コンパイル中に生じたI-Oエラーメッセージや致命的エラーによりリロケータブルファイルが作成されない時に出る赤字のメッセージ。

上の2種類があるが, (1)の場合はACPダ3でコントロールされており, 直接他のモジュールからは使えない。(2)のメッセージ出力ルーチンは直接どのモジュールからでもコントロールが渡せる。この場合メッセージエリア左端アドレスがコーリングシーケンスに必要である。

このモジュールには共通メッセージ出力ルーチン(赤字用)があるので, コンパイル中はコアに常駐する。

(ブロックチャートは図2-5)

NECBSASPF0

ソースプログラムをディスク上に出力するモジュールである。

ファイル名はコマンドで指定したソースプログラム名を用いファイルクラスは“S”で出力する。

ディスク上に出力する時の形式はINPUT コマンドと一致するふうに考慮されており, ファイルの更新はUPDATE コマンドで行なわれる。ブロックチャートは図2-6で示す。

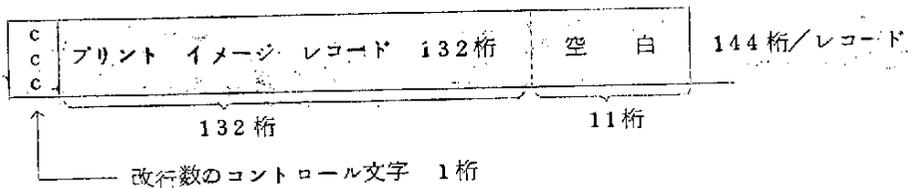
NECBSACPRF

このモジュールはディスク上にプリントイメージファイルを作成する。即ちコマンドのパラメータでLISTの指定があったものは全てディスク上にイメージファイルとして1コのファイルにまとめられる。

そのファイル名はソースプログラム名でクラスは“D”で出力される。(ブロックチャート 図2-7)

コンパイル終了後GOUTPUTかOUTPUT コマンドによりファイルの内容が見られる。

レコード形式は下図のようになっている。



NECBSGCDØ4

コンパイル中使用されるワークファイル2コのファイルテーブルIORW, バッファエリアがある。

ワークファイルは1レコード/ブロックで720桁/レコードとなっているシーケンシャルファイルである。

NECBSGCDØ5

GCDØ4のモジュールとリンクして使われるモジュールでワークファイルへのI-Oのコントロールを行う。

GCDØ4, GCDØ5はフェーズ1でのみ使われる。

ワークファイルに出力されるのはネームテーブルと conditional value (レベル番号88で指定された値)である。

NECBSHCDØ6

データのマップリスト指定があった時, 1度ディスク上に出力されたネームテーブルを読み込むモジュールである。従ってマップの指定がなければ使われることはない。

NECBSICDØ7

フェーズ2で使われるワークファイルのファイルテーブル, IORW, バッファエリアがある。またMINPRAL及びPROCEDURE DIVISIONの処理名テーブルをワークファイルに出力するルーチンがある。MINPRALは可変長アイテムであるので出力する時2桁のMINPRAL アイテムサイズを前追加して出力している。

NECBSJCDØ8

MINPRALを読みデブロッキングするモジュールである。この時ICDØ7で追加されたMINPRAL アイテムサイズを用いている。

NECBSKCDØ8

フェーズ2でネームテーブルを読み込んだ時にネームテーブル全てがコアに入らない場合残りのネームテーブルブロックをワークファイルにコピーするモジュール。

NECBSJCD55

フェーズ3で使うMINPRAL GETのモジュールJCDØ8と同じ機能であるが、このモジュールはファイルテーブル, IORW, バッファエリアがある。

NECBSJCD66

ソースプログラムが Segmentation されている時に使うモジュール, Segmentation されている時は1度オブジェクトファイルをワークファイル上に出力しておき次のフェーズでリロケータブルファイルに出力する。
このワークファイルに出力するリロケータブル形式は80桁/レコード, 9レコード/ブロックで出力している。

NECBSJCD77

フェーズ4で使うモジュールで Segmentation されていた時のみ実行される。前のフェーズでワークファイルに出力したりロケータブル形式のファイルを読み実際のリロケータブル(MGO)に出力するためデブロッキングしてメインモジュールに渡す。

Source Program File 作成ルーチン (NEOBSASPFO)

このルーチンは、初期手続をする入口 SPOP/ と、実際にデータを put する入口 SPPT/ と、終了手続をする入口 SPCL/ の3つの入口がある。SPPT/ の入口で受けわたされたデータは5 record ずつ blocking されて、1 block 720 character として disk 上に書き込まれる。

5.5.2 メインモジュール

メインモジュールは15コのモジュールで構成され論理的にはフェーズ0からフェーズ5までの6フェーズを、物理的には15フェーズを構成する。

フェーズ0

NECBSEP0N2

このモジュールはコマンドのパラメータチェックを行いパラメータ指定の情報をコモンエリアにセットする。

パラメータにエラーがあった場合は

“ PARAMETER ERROR ”

を赤字でキーボードに出力しUEP (Unusual End of Program) となる。

このモジュールではオブジェクトプログラムで使うコモンエリアサブプログラム

(NECBOBJECT) をリロケータブルファイルに出力するNECBSBCP14のインタフェースモジュールにコントロールを渡して終了する。

フェーズ 1

フェーズ1の構成モジュールはNECBSGP1Z1, NEOBGP1Z2, NECBSHP1×3の3コである。

フェーズ1における機能は

- (1) データ名やスペシャル名 (Special-names), 条件名 (Condition-name) などの性格を示すテーブル (ネームテーブルと呼ばれる) を作る。
- (2) オブジェクトファイル用のファイルテーブル, IORW それにバッファエリアやデータエリアのコアを割当てリロケータブルファイル上にサブプログラムを出力する。

(3) IDENTIFICATION, ENVIRONMENT, DATA DIVISION

のシンタックス解析を行う。

(4) 診断メッセージを出力する。

NECBSGP1Z1

ソースプログラムを読み IDENTIFICATION と ENVIRONMENT DIVISION の処理を行う。

ソースプログラムは記述形式および文法上の誤りがないか調べられる。その診断メッセージはコマンドパラメータに従ってユーザの指定した装置に出力される。

(実際のコントロールはインタフェースモジュールの ACP Ø 3 で行なわれる。)

このモジュールの出力は

- (1) スペシャル名 (Special-names) をネームテーブルに登録する。
- (2) SELECT で指定されたファイルの情報をコア内に作り次の NECBSGP1Z2 に渡す。この形式は図 2-8 で示す。

SELECT を処理する時 TSS 用のために特殊な処理がある。それは ASSIGN で指定できる入出力装置とオブジェクトプログラムで使える入出力装置の関係である。

現在は次のように仮定している。

ASSIGN	オブジェクト時	備考
CARD-READER	CDR (端末のカードリーダー)	端末にプリンタのないシステムではディスクになる
CARD-PUNCH	CDP (端末のカードパンチ)	
PRINTER	LNP (端末のプリンタ)	
TAPE-UNIT	ディスク	ダイレクトアクセスはできない
MASS-STORAGE	ディスク	

Compel time file table (3)

Bit-Structure

<u>α</u>		
B	OPTIONAL-file	
A	assigned by /IPI/	
8	Rerun end-of-reel	
4	Rerun begin-of-reel	
2	Rerun integer records	
1	not used	
<u>β</u>		
B	Sequential access	
A	Partitioned-seq. access	
8	Indexed-seq. access	
4	Direct access	
2	ACTUAL-KEYの指定あり	
1	not used	
<u>γ</u>		
B	OPEN-WITHOUT-REWIND	
A	PRINTER-CONTROL	
8	CARRIAGE-CONTROL	
4	HOLLEWITH-MODE	
2	TRANSCRIPTION-MODE	
1	not used	
<u>δ</u>		
B	VERIFICATION	
A	INPUT-RECOVERY	
8	OUTPUT-RECOVERY	
4		
2	not used	
1		
<u>ε</u>		
	ON	OFF
B	not used	
A	Odd parity	Even parity
8	8φ chs. label	12φ chs. label
4	non-std. label	std. label
2	Omitted label	
1	not used	

NECBSGP1Z2

ソースプログラムを読みDATA DIVISIONの処理を行う。ソースプログラムは記述形式および文法上の誤りがないか調べられる。その診断メッセージはコマンドのパラメータに従ってユーザの指定した装置に出力される。

このモジュールの出力は

(1) オブジェクトプログラムで使われるファイルのファイルテーブル、I ORW、およびバッファエリア、レコードエリアにコアを割当てリロケータブルファイル上にサブプログラムとして出力する。

(2) DATA DIVISIONで定義されている全てのデータ名のテーブル(ネームテーブル)を作る。

このネームテーブル内のエレメントの形式は図2-9で示す。

ファイルにバッファエリアを割当てる場合TSS用に特殊な処理を行なっている。すなわちバッファエリアの桁数は必ず720桁割当てられる。これは各ユーザのUOA内のブロックサイズが720桁であるためである。

端末の装置にSELECTで指定されたファイルも実行時にディスクに変更される場合があるので(ASSIGNコマンドによる)バッファエリアの桁数は720桁割当てられる。

NECBSHP1X3

このモジュールはGP1Z1, GP1Z2でコア内に作られたネームテーブルを次のフェーズ2に渡すため一度ディスク上に出力する。

コマンドのパラメータにデータのマップリスト指定があればマップリストを出力する。

このマップはデータ名にレベル番号、データに割当てられたリロケータブルアドレス、クラス、サイズなど編集され出力される。

フェーズ1 特にNECBSGP1Z1, NECBSGP1Z2の2モジュールで共通したネームテーブルのリンクリスト手法と、シンタックス解析で使われる特殊な言語 Pseudo code languageについて述べる。

リンクリスト手法

ネームテーブルにデータ名やスペシャル名が登録されるのはソースプログラム上で定義さ

れた順にテーブル上も登録される。各ネームテーブル上の構成は基本的には、次の4つの部分である。

- (1) データ名
- (2) レベル番号
- (3) リンクフィールド
- (4) そのデータの特徴を示すフィールド(パラメータ)

リンクリスト手法とはこのリンクフィールドを用い各データ名のチェーン構造を作り次のフェーズにおけるネームテーブルサーチ時間を短縮するために有効な手法である。

その構成は、リンクリストテーブル(NECBSSBCPØ9)とネームテーブルである。

リンクリストテーブルは4桁のフィールド、128コのエレメントで構成される。

リンクリストテーブルの各エレメントには、データ名のリンクフィールドの相対番地(ネームテーブルオリジンからの)が入りリンクフィールドには同一チェーンを構成している次のデータ名のリンクフィールドの相対番地が入り最後のエントリーは“ØØØ”になる。

図2-10を参照

リンクリストエレメント128コのどれを使うか決めるのはデータ名をスクランブル(scramble)して行い。

スクランブルした結果が同じ数になったデータ名が1コのチェーンを構成する。普通128通り全て使われる事は非常に少ない。

スクランブルの方法

リンクリストのエレメント決定に使うスクランブルは次のように行い。最初、データ名の右端2文字の数字ビットを使い基数を作る。

1桁のデータ名は、Øが前にあると見えず、すなわちAならばØAとする。

その数字ビットで作られた2桁の数値の左側1桁にデータ名の左端1文字を加える。

最後にその2桁の数値にデータ名の長さを加えてそれを4倍し、リンクリストテーブルのベースアドレスを加え使用する。リンクリストエレメントの右端を決める。

例 NAMEというデータ名をスクランブルすると次のようになる。

○ NAMEの内部コードは 45 21 44 25 である

① 初期値を作る	04	05	} 2進演算で行なっている
② データ名左端1文字加える	45		
③ データ名の長さを加える	00	04	
	51	11	
	×	4	
	44	44	

4倍した値は今の例では4444になるが下3桁444のみが使われる。

今の例ではリンクリストテーブルのベースアドレスから444番地後エレメント番号で111番目のエレメントを用いてチェーンを作る。

このようにスクランブルした結果、同じ数になったデータ名が1コのチェーンを構成している訳であるがPROCEDURE DIVISIONで使われたデータ名を捜す場合今と同様な方法でスクランブルし、そのリンクリスト上のデータ名のみ捜す事により目的のデータ名が見つかる。

もし見つからなければ、そのデータ名は未定義のデータ名となる。

Pseudo code language

NECBSGP1Z1, NECBSP1Z2で使われる言語で、このPseudo codeは命令コード(Operator)とオペランド(Operand)で構成されている。

Pseudo codeはPseudo code interpreterというルーチンで命令コードが解釈されそれぞれのPseudo code サブルーチンにコントロールが渡される。

Pseudo codeの形式は表2-1で示す。

Pseudo code interpreterのブロックチャートは図2-11で示す。

Pseudo code サブルーチンからは必ずinterpreterルーチンにコントロールがもどり、次のPseudo codeの解釈を開始する。

Pseudo codeは、機能別に分類すると8コになる。

1. 入力
2. シンタックス解析
3. エラーメッセージ処理
4. スイッチ操作
5. バケット操作

6. サブルーチン リンケージ
7. マシンコード リンケージ
8. ロジカル ブランチ

次に Pseudo code ルーチンのそれぞれの機能を述べる。

1. 入 力

Get Next Word (GNW)

ソースプログラム上の1語を処理できるようにワードバッファに作る。

このルーチンでは話やリテラルの桁数チェックも行っている。

ソースプログラム1行がなくなれば次の行をGETする。

2. シンタックス解析

Integer test and Load Bucket (ILB)

現在ワードバッファにある内容が整数か調べる。

整数であれば指定されたバケットにその内容を入れ、これに続く Pseudo code を実行する。

整数でなければオペランドで指定された所にブランチする。

Data Name Test (DNT)

現在のワードバッファの内容がデータ名かどうかチェックする。

データ名であれば次の Pseudo code を実行、データ名でなければオペランドで指定された所にブランチする。データ名であるためには次の条件が必要である。

- (1) COBOL リザーブド ワードでない事
- (2) 特殊文字がない事
- (3) ワードの最後が“ ”でない事
- (4) 最低1文字はA~Zである事

Optional word Test (OWT)

現在のワードが指定したリザーブド ワード(キーワード)と一致しているかどうかチェックする。一致している場合は次の Pseudo code を実行する。一致しなければ指定された所にブランチ。

Mandatory Word Test (MWT)

現在のワードと指定されたキーワードを比較し一致しなければ次の Pseudo code を実行する。一致した場合は“エラースキャンモードスイッチ”の状態により次のようになる。

- (1) スイッチがONの時は診断メッセージ

“SYNTAX CHECK RESUMED WITH ××××”

(××××はワードバッファの内容)を出力し、スイッチをOFFにする。

- (2) OFFの場合はそのまま次の処理を続ける。

“エラースキャンモードスイッチ”のチェック後 Operand で指定されているスイッチ状態を調べる。

そのスイッチがすでにONであると

“DUPLICATE KEYWORD CLAUSE”の診断メッセージを出力する。

指定されたスイッチがOFFであればそのスイッチをONにする。

この事によりスイッチの状態でどのキーワードがGETされたかの判定が行なえる。

以上の処理をやり一致していた時は指定された所にブランチする。

3. エラーメッセージ処理

Start Error Scan (SES)

“エラースキャンモードスイッチ”を調べOFFならば診断メッセージ

“SYNTAX CHECK DISCONTINUED WITH ××××”

(××××はワードバッファの内容)を出力しスイッチをONにして指定された所にブランチする。(このスイッチがONであるときのMWTを参照)

エラースキャンモードスイッチがONであった場合は何もせず指定された所にブランチする。

Output Error Message (OEM)

Operand で指定されているエラーメッセージコードにより診断メッセージを作成しパラメータで指定された装置へ出力する。

Exit (EXIT)

エラーキャンモードスイッチをOFFにする。

4. スイッチ操作

スイッチは全部で62個ある。これらはマシンコード1文字で表示されており Pseudo code では特定スイッチを指定する時この1文字を用いて行う。

スイッチは次のような処理で使われる。

(1) ソースプログラム上、各エントリーが正しい位置にあるか。

(2) 二重に指定されている事がないか。

スイッチ操作の Pseudo code には次のものがある。

Set switch Off (SOFF)

operand で指定されたスイッチを off の状態にセットする。

Set switch On (SON)

operand で指定されたスイッチを on の状態にセットする。

Set Off Range of switches (SOFFR)

operand で指定されたスイッチ2コとその間にある全てのスイッチが off にセットされる。

Test switch Off (TOFF)

operand で指定されたスイッチの状態がテストされOFFであれば指定された所にブランチする。ONであれば次の Pseudo code を実行。

Test switch ON (TON)

operand で指定されたスイッチの状態がテストされONであれば指定された所にブランチする。OFFであれば次の pseudo code を実行。

Test off Range of Swifches (TOFFR)

operand で指定されたスイッチ2コとその間に含まれる全てのスイッチ状態がテストされ全てがOFFの時、指定された所にブランチする。

1でもONのものがあれば次のpseude code を実行。

5. バケット操作

バケットは全部で39コあり、データ名の桁数計算やエリア割当て用のバケットして使われる。各バケットはバケットテーブルの開始番地からの相対番地 (relative address) で指定される。

pseudo code には次のものがある。

Integer Test and Load Bucket (ILB)

シンタックス解析のILB参照

Move Bucket (MB)

operand で指定されるバケット1の内容をバケット2に移送する。

Add Bucket (AB)

バケット1の内容をバケット2に加える。

Subtract Bucket (SB)

バケット1の内容をバケット2から引く。

Multiply Bucket (MB)

バケット1の内容をバケット2の内容に乗ずる。

Move Immediate (MI)

operand で指定されている2桁の内容を指定されたバケットに移送する。

Compare Bucket (CB)

バケット1の内容とバケット2の内容を比較し、その結果を表示するフィールド

“SETLATCH”に0, 1又は2をsetする。

0.....バケット1 < バケット2

1.....バケット1 = バケット2

2.....バケット1>バケット2

Compare Immediate (OI)

operand で指定された2桁の内容とバケットを比較し、Compare Bucket と同じ情報をセットする。

6. サブルーチン リンケージ

pseudo code サブルーチンを使う時に用いられる。

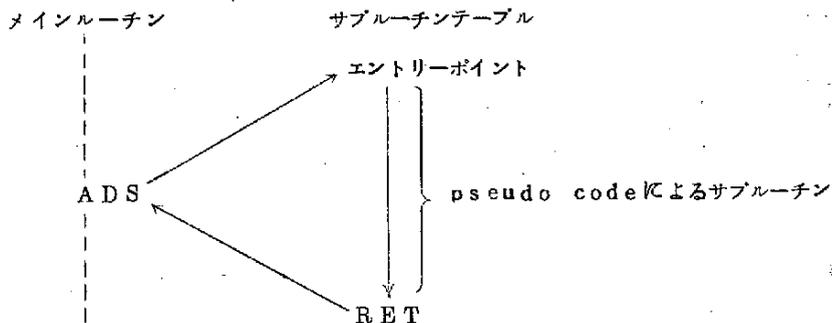
Advance Subroutine (ADS)

サブルーチンのエントリーポイントに pseudo code interpreter のテーブルポインタ (PSPTR) を set しサブルーチンの実行を開始する。

この時ADSの次の pseudo code テーブルの位置をバケット3にセットしておく。

Return (RET)

サブルーチンからメインルーチンに戻る。これはADSでバケット3に帰り番地が set されているので、PSPTRにバケット3の内容を set する事により行なわれる。



7. マシンコード リンケージ

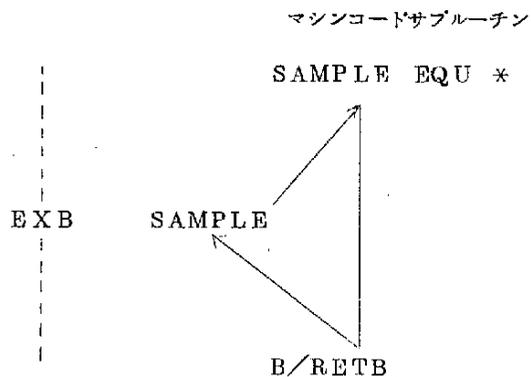
Enter Machine Code (EXB)

マシンコードサブルーチンにコントロールを渡す。この時EXBの次の pseudo code

位置をPEENTRYというフィールドに入れておく。

マシンコードサブルーチンから pseudo code にもどる場合はRESBというルーチンにコントロールを渡す。

このルーチンではREENTRYのフィールドの内容をPSPTRに set し pseudo code interpreter routineにコントロールを返す。これによりふたたび pseudo code が実行される。



8. ロジカル ブランチ

Advance Equal (ADE)

Advance Not Equal (ANE)

Advance Greater Than (ADG)

Advance Less Than (ADL)

以上4コの pseudo code はOI, OB pseudo code でセットされた SET LATCH を調べ、条件が真であれば指定された所にブランチする。偽であれば次の pseudo code を実行する。

Advance (ADV)

指定された所にブランチする。

Range Limit (RL)

現在のワードの長さが operand で指定された長さと比較される。

比較された結果、ワードの長さが大きい場合指定された所にブランチする。小さいが等しい場合は次の pseudo code が実行される。

フェーズ 2

フェーズ2ではPROCEDURE DIVISIONを分解し、フェーズ3で処理できるような形式(MINPRALと呼ばれる)でワークファイル上に出力する。

フェーズ2での処理は次のような事である。

- 1) シンタックス解析
- 2) ソース・プログラム上の複雑な表現(IF UNTIL COMPUTEなど)を簡単なステートメントの形式に変換する。
- 3) フェーズ1で作られたネーム・テーブルを用い、データの構造を示す中間言語(MINPRAL)を作る。
- 4) 最適化の処理を行う。
- 5) オブジェクト・プログラムで使われるサブルーチンを出力する。

フェーズ2は、8コのメイン・モジュール

```
NECBSIP2M1,  NECBSJP2MA,  NECBSJP2MB  
NECBSJP2MO,  NECBSKP2M2,  NECBSKP2M3  
NECBSKP2M4,  NECBSKP2M5
```

で構成されている。

NECBSIP2M1

このモジュールはソース・プログラムのPROCEDURE DIVISIONを読みステートメントを分解しMINPRAL(machine independent pre-assembly language)として出力する。

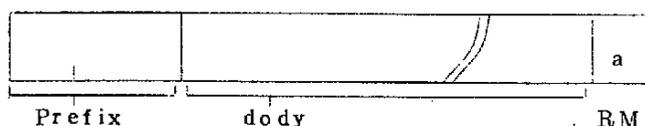
MINPRALの形式は図2-12のようにになっている。

IP2M1の機能を項目別に上げると

1. ソース・プログラムの記述形式(reference format)のチェックを行う。
2. IF UNTIL COMPUTEなどのシンタックス解析を行い簡単なステートメントに分解する。
3. PROCEDURE DIVISIONで使われるリテラルやCOMPUTEの演算用中間エリアにコアを割当てサブプログラム(リロケータブル形式)に出力する。

このモジュールでの内部処理方法は、エレメントと呼ばれるMINPRALの基本になる形式を用いて行なわれる。

エレメントの形式は下図のようになっている。



これらはさらに7コの型に分類される。

1. キー・エレメント
2. SECTION/PARAGRAPH名エレメント
3. 診断メッセージ・エレメント
4. ガード・ナンバ・エレメント
5. reference procedure name (処理名参照)エレメント
6. リテラル, 中間エリア用エレメント
7. パラメータ・エレメント(ネーム・エレメント)

TERMINATOR(ターミネータ)

ターミネータとは、ステートメントの論理的な終り(logical end)を示すもので、キーワードや処理名(procedure name)が相当する。このターミネータ決定はIP2M1モジュールのGet Next Word(GNW)ルーチンが行っている。ターミネータには3種類がある。

- 1) ターミネータにより特殊な処理ルーチンが必要となる“Tφ”の形式。Tφはエレメントの prefix 2桁で示されている。

Tφに含まれるキーワードは

ALTER, ADD, ACCEPT, COMPUTE, CALL,
CLOSE, CORRESPONDING, DIVIDE, DISPLAY
DECLARATIVE, EXIT, ENTER, EXAMINE,
GO, IF, MULTIPLY, MOVE, NOTE, OPEN
PERFORM, READ, STOP, SUBTRACT
USE, UNTIL, WRITE

以上である。

- 2) 特殊な処理ルーチンは必要でないが、ターミネータである。“T1”の形式。T1はエレメントの prefix 2桁で示される。

T1のキーワードは次の3コのみである。

ELSE, NEXT, OTHERWISE

3) カラム8から書かれる処理名もターミネータであり“T2”の形式となる。

以上3種類のターミネータの他に次のような prefix がある。

- 1) △φ データ名又は処理名参照エレメント
- 2) ▽1 保有語 (reserved word)
- 3) △1 数字アイテム (リテラル又は名前)
- 4) △4 英数字アイテム

prefix が使われるのはこのモジュールだけであり、MINPRALは別の形式で出力されている (図2-12参照)

このモジュールのメイン・ルーチンのブロックチャートは図2-14で示す。

次にこのモジュールにおけるCOMPUTEの分解について述べる。COMPUTEルーチンは数式化されたステートメントを分解し、普通の演算ステートメント (ADD SUBTRACTなど) にする。数式、分解の鍵になるのは数式の記述形式と演算の優先度 (precedence) である。

各演算の優先度には次のような値をつけておく。

+, -	1
*, /	2
**	3

即ち 乗、乗算と除算、加算と減算という順になっている。それに加え、カッコはどの演算記号よりも優先度が高くしてあり、特にφという値をつけておく。

各演算記号はGNWルーチンから渡される時キーワードの形式になり3桁のキーに変換されている、この2桁目に優先度の値 (precedence value) がある。

ADD	A1
SUBTRACT	S1
MULTIPLY	M2
DIVIDE	D2
EXPONENTIATE	E3

COMPUTE ステートメントで、受領エリア (receiving area) の名前 ANSNAMBと呼ばれるバケットに入れておく。

ROUNDEDの指定があったら、ROUNDSWを set しておく。左カッコが見つかる
とCOMPUTE分解エリアに“φ”“φ” をセットしPAREN COUNTに1加えて
おく。

この“φ”をセットするのは、カッコが使われていることを示すと同時にカッコの前の演算
を出力させるためである。

データ名やリテラルが数式の中にあるとシンボリックが与えられ、それらに加えられてリ
ストエリアに登録される。シンボリック名は、データ名やリテラルの代りに数式分解エリア
に登録される。

リテラルがリストエリアに登録される時には、その前にパラメータの形式に変換され登録
される。

演算記号が見つかる時、そのキーは分解エリアに登録されキーの後にワードマークを付けた
優先度の値がセットされる。

ここで例を示すと、

COMPUTE A = (B + C)

分解エリア φ φ A φ φ φ 2 @ " 1 A 1 - @ 1 A φ φ φ 3 @

φ 初期値 (テーブルの最初に必ずセットする。)

φ 左カッコで作る

A φ φ φ 2 B のデータ名に与えられたシンボリック名

1 A 1 - ADD キー

1 ADD の優先度

A φ φ φ 3 C のデータ名に与えられたシンボリック名

右カッコが見つかる時 PAREN COUNT から 1 引き、カッコ内の数式を出力する。た
だし、不要のカッコが使われたような場合

COMPUTE A = (B) +

COMPUTE ルーチンは不要なカッコは無視する。

数式を簡単なステートメントで出力するのは次の 3 つの場合のいずれかの時である。

- 1) 演算記号が見つかりその優先度が先にセットされている優先度より小さい場合。

COMPUTE A = B * C +

- 2) 右カッコがあった場合

```
COMPUTE    A = ( B + C ) * .....
```

3) 数式が終った時

```
COMPUTE    A = B + C
```

1) の場合の処理の仕方は、現在の優先度と先に登録されている優先度（ワードマークにより簡単に挿せる）が比較され、現在の優先度が小さいか等しい時に先に登録されているキーとその前後のオペランドでステートメントを作り出力される。現在の優先度が大きければ何も出力せずキーは分解エリアに登録され処理を続ける。

例 COMPUTE A = B * C +

出力はMULTIPLY B BY C GIVING TEMP

```
COMPUTE    A = B + C *
```

何も出力しない。

2) の処理の場合は、カッコ内の数式が右から左にの優先度（左カッコにする）があるまで出力されるカッコが幾重にも使われている時には1番内側のカッコから処理される。

例 COMPUTE A = ((B + C) * D) *

出力はADD B TO C GIVING TEMP

```
MULTIPLY TEMP BY D GIVING TEMP
```

3) の処理の場合、優先度の初期値としてセットしたのがあるまで出力され、受領エリアはANSNAMEのオペランドとなる。

以上のような方法でCOMPUTE ステートメントは分解されていく。

次にIFとUNTILの分解について述べる。

この分解ルーチンは、大小関係テスト（関係テスト）、条件名テストクヌや符号テスト、スイッチの状態テストなどを処理しIF ステートメント内に書かれた数式をIFの前に出したり、またTHENキーを必ず出すようにしている。

例 IF A = (B + C) OR D

出力は ADD B TO C GIVING TEMP1

```
IF    A = TEMP1 OR flag D = A    flag THEN
```

条件ステートメントを分解するのはこのモジュールのほかNECBS KP2M3でも行なっている。

このモジュールでの条件ステートメントの分解は略記法を正す事、NOTを取りさる事、条件内の数式をIFの前に出すことである。

このための処理で3コのバケットが使われる。

バケット1は比較条件の左辺(主語)用

バケット2は比較演算子のキイ用

バケット3は比較条件の右辺(目的語)用

これは略記法を書く場合でも必ず最初に完全な形式で書いてあるからで、1度は全てのバケットに情報がセットされる。これらのバケットにはデータ名やリテラルが登録されたエリアのアドレスがセットされ出力する時に正しいオペランドを出すようになってい

最初の3コのオペランドでバケットがいっぱいにならない場合、すなわち

```
IF A+B .....
```

このような時はまず

```
ADD A TO B GIVING TEMP
```

```
IF TEMP
```

に変換され処理を続ける、ソースプログラムIで、論理接続詞(AND,OR)が見つかった場合、最初のバケットの内容は出力される。

```
IF Subject - 1, KEY-1, Object - 1
```

次に論理接続詞の次のアイテムにより対応するバケットにセットされる。例えば、新しい比較演算子であればバケット2にセットされ、次のオペランドがバケット3にセットされる。

この時のバケット状態は

```
Subject - 1, KEY-2, Object - 2
```

となっている。

新しいオペランドであるとこれはバケット3に入れられる。

すなわち

```
Subject - 1, KEY-1, Object - 2
```

となる。ところがこの Object - 2 がほとんどの略記法によるオペランドかそれとも条件名やスイッチ状態の special-name かこのモジュールでは不明である。従ってこのバケットの内容を出力する前に特殊キイ(flag と呼ぶ)を付けバケット3から出力する。

```
iflag Object - 2, KEY-1, Subject - 1 flag
```

この処理を正しく行なうのはNECBSKP2M3であり Object - 2 のデータの性質が判明するので行える。

NOTを取りさる方法は次のように関係演算子を変更することにより行なっている。

GREATER EQUAL OR LESS THAN
LESS EQUAL OR GREATER THAN
EQUAL UNEQUAL

カッコの前にNOTがあった場合はカッコの中の全ての関係演算子に変更される。

IFの中の数式は前に述べたCOMPUTERルーチンで処理し、条件文の前に出力され最後の演算結果の入ったエリアがバケット1かバケット3にセットされる事によりIFの処理を続ける。

以上がNECBSJP2M1におけるおもな処理である。

NECBSJP2MA

このモジュールは次のNECBSJP2MBのためにprocedure name table (処理名テーブル)を作る。

NECBSIPM1でワークファイルに出力するテーブルの内容はセクション名、パラグラフ名として定義されたものとALTERやPERFORMステートメントで参照された処理名である。このモジュールは2コのテーブル

procedure define table (処理名定義テーブル)

procedure reference table (処理名参照テーブル)

を作る。

define table はPROCEDURE DIVISIONで定義された処理名順に構成される。

テーブルを作り終わるとreference table に登録された処理名をdefine table 内で探し見つかればALTERやPERFORMで参照された事を示すインディケータをセットする。

このモジュールの終りではreference table は消されdefine table のみ次のモジュールに渡される。

NECBSJP2MB

このモジュールではJPMAで作ったprocedure define table を用い

MINPRALの変更を行なう。

前のモジュールでセットされたALTERやPERFORMインディケータの付いたMINPRALが出力される事になる。

またこのモジュールではALTERやPERFORMの out of line coding をリロケータブルファイルに出力する。

このオブジェクトコードが出力されるのはソースプログラムがセグメント化されている時で、もしも優先度がSEGMENT-LIMITと49の間のALTER可能なスイッチ及びPERFORMの end of PERFORM (PERFORMのリターン機構)のコーディングのみである。

出力する理由はSEGMENT-LIMITから49までの範囲にあるALTER可能なスイッチは前の状態を保持していなければならないためである。

NECBSJP2MC

COBOLのオブジェクトプログラムで使われるサブルーチンをリロケータブルファイルに出力する。

これを行なう事によりLINKコマンドを使う時にサブルーチンをCALLするCALLカードは不要となる。

サブルーチンは表2-2で示すようなものがある。

表2-2

オブジェクト サブルーチン

サブルーチン名	機能
NECBOSACTP	ACCEPT サービスルーチン(端末KBIより)
NECBOSADVR	OVER FLOW ルーチン
NECBOSALTS	ALTER サブルーチン
NECBOSBINY	2進演算サブルーチン
NECBOSCDAD	not use
NECBOSCLER	バッファパンクチュエーション エリア
NECBOSCMPR	比較(Compare)
NECBOSDCBN	10進, 2進, 変換ルーチン
NECBOSDSPY	DISPLAY ルーチン(端末キーボード)
NECBOSDVZR	DIVIDE BY ZERO エラー
NECBOSEXP	Exponentiation (乗)
NECBOSFLDM	Floatingの編集
NECBOSMSCS	Mass storage Control ルーチン
NECBOSSUBS	Subscript 演算ルーチン
NECBOSUPR	Supervisor Control ルーチン
NECBOSZERO	Zero insertion の編集ルーチン

NECBSKP2M2

このモジュールはフェーズ1でワークファイル上に出力されているネームテーブルを読み込みコアに再びネームテーブルを作る。

このネームテーブルを作る時、コアにテーブル全体が入る場合とコアがなく途中までしか入らない場合がある。途中までの場合はテーブルが終るまで処理が繰返される。(iteration passと呼ぶ)

全体がコアに入った場合、ITERSW(ITER/)をoffにする。

すなわち iteration pass 不要である。

途中までの場合はITERSWをonにし iteration pass 必要を次のモジュールに知らせる。

ネームテーブルを読み込んだら次の処理を行う。ITERSWがoffならネームテーブルの最後は論理的にも物理的にも一致しており、そのアドレスをLOGE/(logical end)フィールドにセットする。もし、ITERSWがonなら論理的なテーブルの最後としてはFDか、WORKING-STORAGE、CONSTANT SECTIONのφ1 レベルの前のアイテムとしそのアドレスをLOGE/にセットする。

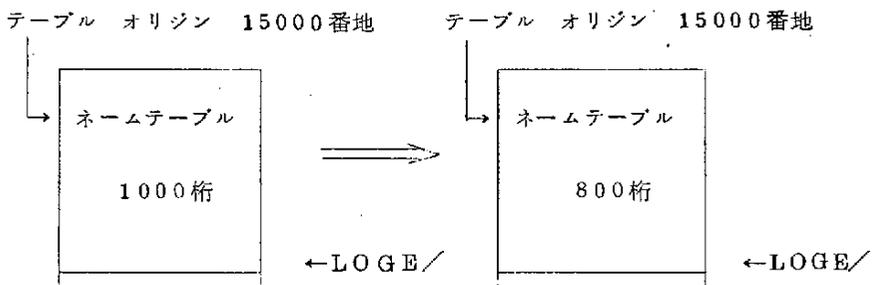
次にテーブルのサイズを更新するこの処理はSIZE/(前に読み込まれたテーブルの桁数)をCUMU/に加えることにより行なわれる。

CUMU/には合計のテーブル桁数が入ることになる。

最後に adjustment factor(ADJU/)を計算しておく。この値は次のモジュールでリングアドレスから実際のテーブル位置を計算する時用いられる。

$$ADJU/ = \text{ネームテーブル オリジン} - CUMU/$$

これを図で示すと次のようになる。



1回目

CUMU/ = $\phi\phi\phi\phi$

ADJU/ = 15000

SIZE/ = 1000

2回目

CUMU/ = 1 $\phi\phi\phi$

ADJU/ = 15000-1000=14000

SIZE/ = 800

NECBSKP2M3

このモジュールでは次のような処理を行う。

1. NECBSIP2M1で処理されたIF, UNTIL ステートメントを完全な形式にする (IP2M1 参照)
2. MINPRALのデータ名を完全なパラメータ形式に変更する。
3. CORRESPONDINGの第1処理を行う。

このモジュールの一番重要な処理はデータ名があったらネームテーブルを捜し、実際のネーム形式からパラメータ形式に変換する事である。

このネームテーブルを挿す時にフェーズ1で行なったりリンクリスト手法を用いている。

フェーズ1と同じようにスクランブルし、リンクリストエレメントを決める。このリンクリストエレメントの内容にADJU/を加えるとデータ名が登録されている実際のアドレスが決まる。

その結果がLOGE/で示される値より小さければデータ名は現在コアにあるテーブル内にあり、LOGE/より大きければ残りのテーブルにあるか又は未定義の名前である。

テーブル内にある時はチェーン構造を構成するアイテムのみ等しいものがあるか捜される。見つければそのアイテムの構造を示すパラメータを出力する。チェーン構造の最後(リンクフィールドが $\phi\phi\phi$)まで捜して見つからなければ

“UNDEFINED NAME ××××”

を出力する。

修飾の処理は次のように行う。

例えば A OF B OF C

と書いてある場合、最初にCを捜す。このCを捜すのはリンクリスト手法による。

CがあったらA, Bともにシリアルサーチ(左から右)で捜されAが見つけれれる。

Aが見つかったら修飾されていないデータ名と同じ処理となる。前のモジュールで論理的なテーブルの最後を決める時 F 1) か ϕ 1 レベル (W-STORAGE or CONSTANT) の前にしたのはこの修飾の処理のためである。

このモジュールでは IF, UNTIL を完全な形式に変更する。すなわち、略記法 3 の形式 (比較演算子, 主語の省略) の場合データ名の性格が不明のため IP 2 M 1 のモジュールで処理不可能であったが、このモジュールでは行なえる。

この IF ルーチンの処理方法は flag を見つけたらその次のオペランドが何かによって行う。本当のオペランドなら IP 2 M 1 で仮定したのが正しいのでそのまま処理される。

条件名か Special-name = condition-variable
special-name ならばキーの形式にして出力する。

本当のオペランドなら次のような処理となる。

条件文 IF A > B OR flag C > A flag

とって入力される。

従って flag を取りさり比較演算子を逆にして

IF A > B OR C < A

出力される。

CORRESPONDING の処理ルーチンはこのモジュールと次の NECBSKP 2 M 4 にある。

このモジュールでは CORRESPONDING オペランドとそれに従属する全てのデータ名のテーブルを MINPRAL として出力する。例えば

MOVE CORRESPONDING A TO B

ネームテーブルは

AA ϕ 2 LINK P-LIST @

A1 ϕ 3 LINK P-LIST @

A2 ϕ 4 LINK P-LIST @

A3 ϕ 4 LINK P-LIST @

C ϕ 2 LINK P-LIST @

となっていると

AからA3までのネームテーブルの内容を出力する。

Bも同じように処理される。

以上がNECBSKP 2M3の処理のやり方である。

NECBSKP 2M3

このモジュールでは、CORRESPONDINGの処理及びADD SUBTRACT, IF の最適化 (optimization) を行う。

CORRESPONDINGの処理方法は前のモジュールNECBSKP 2M3で出力したオペランドのネームテーブルからCORRESPONDINGテーブルを作り、同じデータ名があるか調べてあればMOVE ADD SUBTRACTの普通のステートメントと同じ形式のMINPRALを出力する。もし、何もない場合は診断メッセージ

NO MATCH FOR CORRESPONDING

が出力される。

このモジュールの重要な機能は、最適化処理である最適化を行う事によりオブジェクトプログラムの実行時間を短縮するようにしている。

最適化処理で作り出されたオブジェクト・コードはエミット ストリーム (Emit Stream) と呼ばれるコードに変換され、全てのオブジェクト・コードを作成する。フェーズ3のモジュールNECBSMP 3M1でオブジェクト・ファイルに出力される。

(Emit に関してはフェーズ3参照)

全てのADD SUBTRACT IFが最適化されるのでなく最適化できる条件に合ったステートメントのみ行なわれる。

IFの場合

- 1) 5番目のワードはTHENかANDキーである事。
- 2) 比較するオペランドは基本項目がREDEFINEされている項目である事。
- 3) 基本項目でも編集されていない事。
- 4) 表意定数 (figurative constant) はSPACEがZEROである事。
- 5) 英数字項目ならば比較演算子がEQUAL又はUNEQUALである事。

以上の条件が全て満足されていた時に最適化コードを出力する。

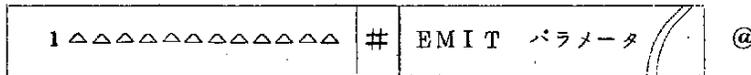
ADD, SUBTRACTの場合は

- 1) GIVING, ROUNDED, SIZE ERRORを含まない事。
- 2) 小数点以下の桁数が等しい事。
- 3) 両オペランドとも添字されていない事。

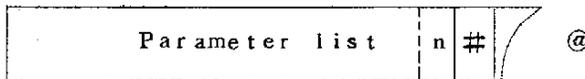
以上の条件が満足されていた時最適化コードを出力する。

ここで出力する Emit Stream の形式は次のようになる。

- 1) 命令コード



- 2) オペランド



- n 1 6(8) ITEMTEMP バケット使用
3 6(8) ITEMTEMP 2 バケット使用
7 6(8) ITEMTEMP 3 バケット使用

NECBSKP2M5

フェーズ2の最後のモジュールである。

このモジュールでは、添字の最適化を行なっている。

最適化を行う事により添字による計算を可能な限り少なくするようにしている。

このモジュールにおける添字の最適化処理は次のように行う。

各添字のセットは1~3組(1次元テーブルなら1組, 3次元テーブルなら3組)からなり、各組は添字のエリア アドレスとその添字により増加(increment)されるテーブルの桁数(ここでは increment number と呼ぶ)により構成される。今、2コの添字のセットがあり、それが同一である場合意味することは、添字が等しく、しかも各添字で increment される increment number も等しい事になる。(最適化できる)そこで、すべて一意なセットをテーブル(INDEXTAB)にプールしておき、次のような

条件になるまで維持しておくことにより最適化を行う。

その条件とは

- 1) パラグラフやセクションの定義
- 2) CALLステートメント
- 3) PERFORMステートメント

以上の条件どれかになると添字、すべてのセットは最適化不能となる。この場合 INDEXTAB を初期状態(ブールのない状態)にもどす。

また、リース プログラム中に次の条件に合うオペランドがあると、

1. 演算, ACCEPT, READINTO, MOVEステートメント@受領域
2. READステートメントで読むファイルのレコード・エリア
3. この2条件のどちらかに合うオペランドに従属するオペランド
4. 上の3条件, どれかに合うオペランドと同じコアを共有するオペランド
(REDEFINESか, マルチ・レコード・エリア)

そのオペランドを含む添字のセットは使えない, 即ち最適化できないのでプールからはずされる。

又 次のような場合, 必ず計算されるように最適化のためのプールには加えない。

- 1) READのAT ENDの通り道
- 2) 条件文の“TRUE”とELSEの通り道
- 3) 複合条件内の最初の2オペランド以外のオペランド

INDEXTABは次のようになっている。

```
X1-φ.2  dddd 11111 rrrrrr dddd 11111 rrrrrr dddd 11111 rrrrrr
          |-----|         |-----|         |-----|
          1 次元             2 次元             3 次元
          :
          :
          :
X1φ
```

各エントリーは45桁で構成されており、

インデックス レジスタのアドレス2桁, そのエントリーの今の状態を示すインディケータ,
42桁で, increment number (dddd)と添字オペランドの左端アドレス(11111)
と右端アドレス(rrrrrr)1~3組が含まれる。

添字つきオペランドが get されるとそのセットを作り INDEXTAB を探し同じものがあるか調べる。同じものがあれば、その添字計算のオブジェクトコードは出さずに INDEXTAB にあるインデックスレジスタのアドレスを用いる。(これで最適化されたことになる) もなければ新しく計算するオブジェクトコードを Emit stream (KP2M4, MP3M1 参照) で出力し、今のセットを INDEXTAB に登録する。

フェーズ 3

フェーズ3は1つのモジュール NECBSMP3M1 のみで構成される。このフェーズの機能は

1. シンタックス解析を行いエラーがあれば診断メッセージをコマンドで指定された装置に出力する。
2. PROCEDURE DIVISION のオブジェクトコードを出力する。

NECBSMP3M1

フェーズ2で出力された MINPRAL を読みオブジェクトコードを作成する。このモジュールには読み込んだ MINPRAL アイテムを一時セーブしておくために80桁のバケット、ITEM, ITEMTEMP, ITEMTEMP2~6の8コがある。各パラメータはそのバケットの左端からの相対番地で知る事ができる。例えば

アドレス部	(ITEM+6)
英数字桁数フィールド	(ITEM+11)

このモジュールでは全てのオブジェクトコードが出力できるが一部分最適化の処理により Emit Stream で入ってくるものもある。(フェーズ2 NECBSKP2M4, KP2M5 参照)

そこでこのオブジェクトコード作成方法を述べる。

EMITルーチンがこのオブジェクトコード作成をコントロールしており、出力するオブジェクトコードは LNKSコマンドで処理可能なリロケータブルサブプログラムとして出力する。このEMITルーチンはブランチ命令に続くコーリングシーケンスを構成するストリーム (Emit Stream) を解釈してオブジェクトを作成している。

ストリームはヘッダ文字とトレーラ文字に分けられる。ヘッダ文字は一種の仮の命令コードでトレーラ文字がそのオペランドと見なせる。

ここで例をあげて説明すると

```
MOVE A TO B
```

これがMINPRALとして入力されると、AというオペランドはITEMTEMPにBというオペランドはITEMに入る。

今、MOVE ルーチンがオブジェクトコード

```
MCLB A, B
```

を出力しようとしたらEMIT ルーチンには次のようなコーリングシーケンスを渡す。

```
B      EMIT
```

```
DCW   @=ABAAPLφ@ ..... Emit Stream
```

これは

= 実際の命令コードで13(s)である。

AB これはITEMTEMPに入っているパラメータのアドレス部を用いて命令のB-アドレス部を作る。

PL この命令のバリエーションを作成、今の場合43(s)となる。

φ 命令コードが終了リロケータブルファイルにこの命令を出力する。

このように簡単な Emit Stream で全てのオブジェクトコードが出力できるようになっている。

最適化されることにより出力された Emit Stream はMINPRALを get するルーチンで判定して、そのルーチンで直接 Emit Stream を渡してオブジェクトコードを作っている。

ソースプログラムがセグメント化されている場合としない場合ではオブジェクトコードを出力するファイルが違っている。

セグメント化されている場合、オブジェクトコードは1度ワークファイルに出力してしまい、リロケータブルファイルには処理名の番号から作られたリンケージシンボルのDEFINEのみ行うサブプログラムを出力しておく。

セグメント化されていない場合は全てこのモジュールでリロケータブルファイルに出力してしまう。

フェーズ 4

NECBSNP4M1

このモジュールが使われるのはソースプログラムがセグメント化されていた時であり、MP3M1で出力したオブジェクトコードファイル(ワークファイル)を読んでリロケータブルファイルにCOPYする。

フェーズ 5

NECBSNP5Z1

COBOL コンパイラの最後のフェーズである。

このモジュールはファイルのCLOSE後、スーパーバイザにコントロールを返すため /EOP/ (end of Program) ルーチンに行く。

他にコンパイル中に生じたI-Oエラーなどのメッセージを端末に出力するためのコントロールも行っている。

以上で COBOL コンパイラの詳細を終了する。

5.6 会話形式によるコンパイラ

会話形式 (conversational mode) のコンパイラはソースプログラムをコンパイルする時1行毎にシンタックス解析を行い、文法上のエラーがあったら端末のオペレータへ診断メッセージを出力してやる。

この診断メッセージをオペレータは判断し、エラーを正してその行を入れ換える事によりコンパイルを続ける事ができる。

この会話形式によるコンパイル方法がとられるのはコマンドのパラメータで入力装置として KBI (キーボード) が指定された時である。

会話形式でコンパイルしている場合、エラーが見つかりと赤色でメッセージが出力され、続いて次の行にそのシーケンス番号が黒色に打たれる。

```
例 *001φφφ ADD A FROM B
    * F INVALID OPERATOR
    : *φφ1φφφ
```

オペレータはメッセージからエラー状態を判断し、次のいずれかを行わなければならない。

第1にはエラーの行を正しく入れ直す場合

*φφ1φφφ ADD A TO B

打ち直す場合はカラム8からすなわちその行全体を打ち直す。

第2にメッセージを無視してその行の処理を読けた時

*φφ1φφφ Δ IGNORE

“IGNORE”がキーボードから入力するとコンパイラはそのまま処理を続ける。

1行毎のシンタックス解析を行うためフェーズ1のNECBSGP1Z1, NECBSGP1Z2, フェーズ2のNECBSIP2M1が会話用の特別処理を行なっている。

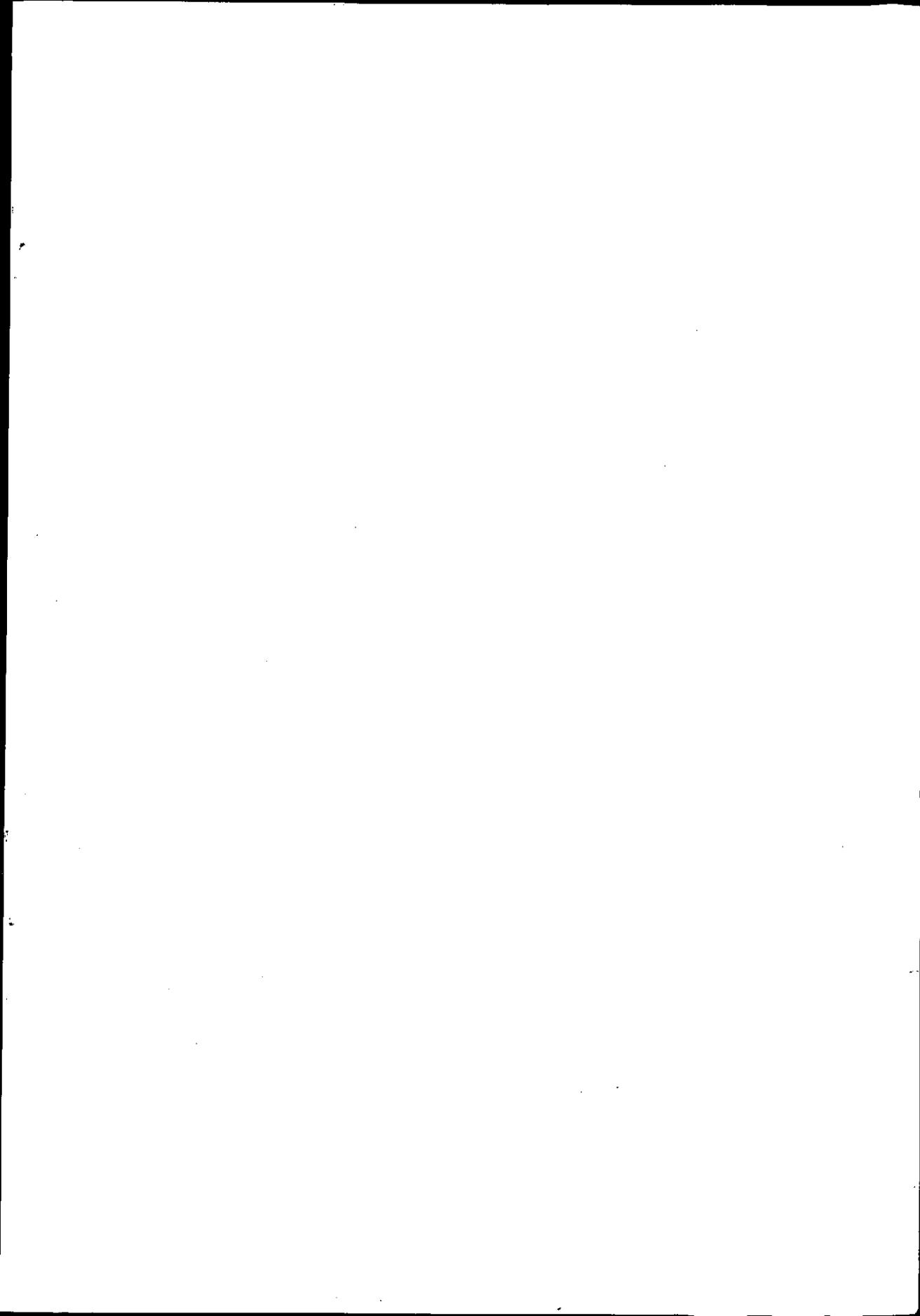
フェーズ1における処理は次のようにやっている。最初にシンタックス解析を行い記述形式、文法上の誤りがないかチェックされる。

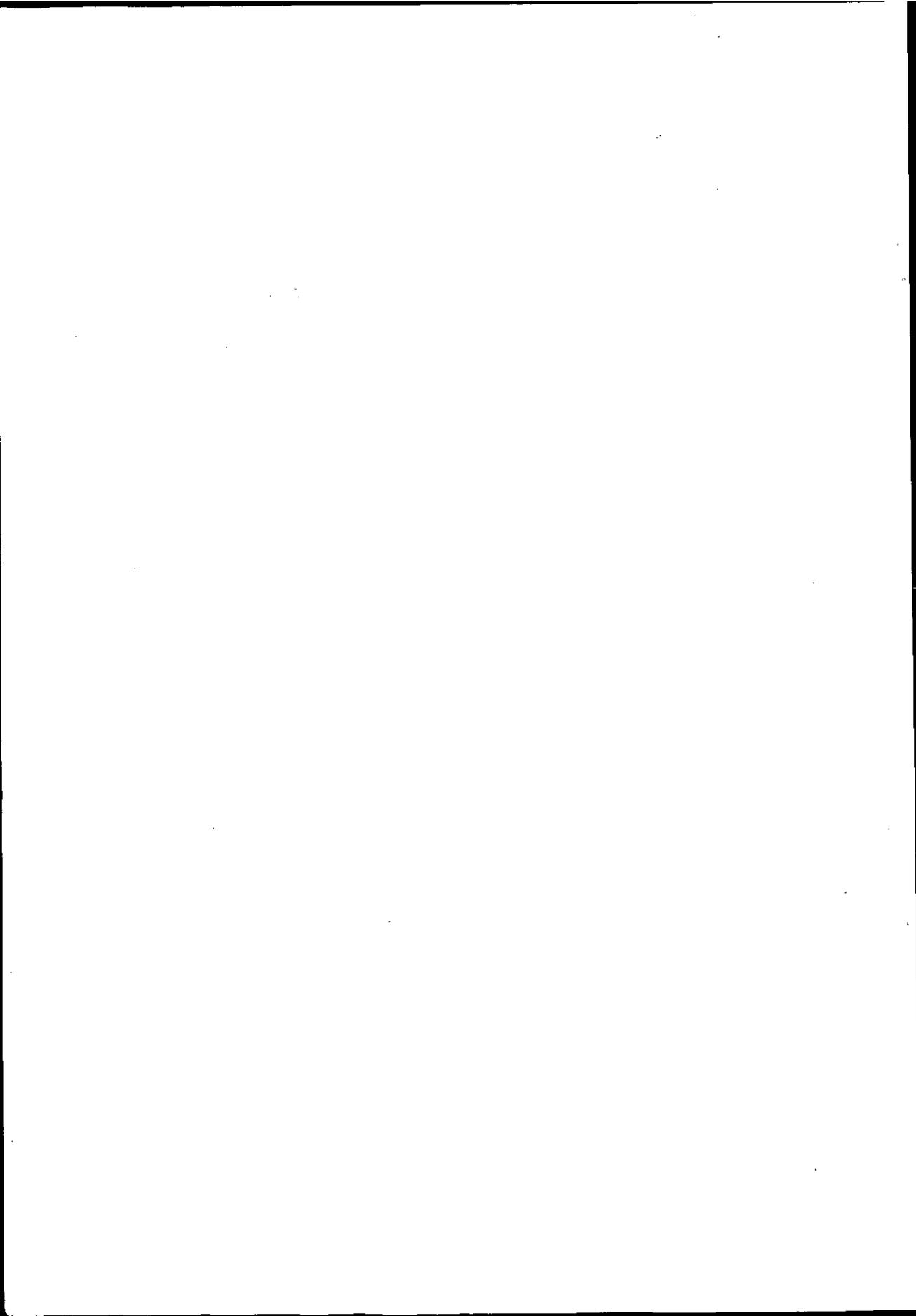
このシンタックス解析中はPseudo code tableでのみ行なわれネームテーブルやコンパイル中のファイルテーブルが変更されることはない。

1行のシンタックス解析が終りエラーがなければ再度その行の処理を行う。この時はネームテーブルやコンパイル中のファイルテーブルは更新されていく。

フェーズ2における方法もフェーズ1と同じような形式で行なわれるが、2度ソースプログラムを調べる事はない。

1行の処理を開始したらMINPRALや処理名テーブルのアイテムをスタックしておく。1行のシンタックス解析が終りエラーがなければスタックされたMINPRAL, 処理名テーブルを出力する。途中でエラーがあったらオペレータに診断メッセージを出力しオペレータの指示によって処理が違ってくる。新しく1行が入れ直されたら最初から処理しスタックしておく。メッセージが無視されたら(“IGNORE”が指定されたら)そのまま続けてスタックしていく。





禁無断転載

昭和 44 年 3 月 発行

発行所 財団法人 日本情報処理開発センター

東京都港区芝公園21号地1番5

機械振興会館内

TEL (434) 8211 (代表)

印刷所 東京サンライズ印刷

東京都新宿区市ヶ谷本村町2番地

TEL (267) 2060

