43-S-003

新しい検索技術と検索システムに関する基礎理論の体系化

（機械工業に関する情報の検索システムとプログラムの開発）

昭和 44 年 3 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，効果的に進める担い手として，最近とくにその役割の重要性が認識されてきております。

また，情報処理そのものも，第3世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，コンピュータ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見ると，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幾多の課題があります。

すなわち，現代の企業はますます大規模化の傾向にあり，組織も複雑化しつつあります。このため組織内で扱われる情報も増大する一方で，企業の経営においては情報管理が重要な問題となり，組織のあらゆる人に対して必要なときに必要な情報を必要な形で与えることのできるような情報システムの確立が強く望まれてきております。

この実務上の要求から現在確立されつつある情報システムにおいて特に重要なものがデータを集中管理する大容量ファイルシステムであります。当財団は情報処理に関するこれら諸問題解決のため，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大容量の記憶装置による情報の貯蔵と検索方式の研究を推進するため，㈱富理工学研究所に委託した調査研究の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」のうち，「機械工業に関する情報の検索システムとプログラムの開発」の一部として実施したものであります。

ここに本研究実施にご尽力ならびに御支援を賜わった関係各位に心より感謝の意を表しますとともに，本報告が，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3月

（財）日本情報処理開発センター

会長 難 波 捷 吾

はじめに

情報検索は，これだけで独立しているものではない，人事管理，在庫管理，あるいは資料管理と呼ばれる，いわゆる情報管理システムの中での，情報の収集，蓄積，検索，配布，を中心とした1つの作業である。

情報検索をさらに細分してみると，(1)情報の収集・整理，(2)その情報から新しい情報（2次情報，3次情報）の作成，(3)収集あるいは作成した情報の蓄積，(4)問合せに応じて，必要な情報を探索し，利用者に提供するといった作業が含まれている。

情報検索という仕事に必ずしも電子計算機を使用する必要はない。しかし大量の蓄積情報の中から必要な情報を探しだすという作業につきものの，時間のかかる単純な作業の繰り返し，および情報の出入りのはげしい蓄積情報ファイルの保守などの作業を機械化し，自動化するという目的から，いくつかの専用機械が開発され，また一般の情報処理に使用している汎用電子計算機が利用されるようになった。

現在，電子計算機を使用した情報検索システムでは，収集した情報の整理，それからの新しい情報の作成，蓄積，探索といった作業のほとんどをシステマチックに計算機を使用して行なうように設計されていたが，ここで問題となることは，ハードウェアの問題として，記憶装置の容量と探索時間があり，ソフトウェアの問題として，(i)情報のシステム内での表現，(ii)収集情報から新しい情報を作成する方法，(iii)蓄積情報ファイルの構成方法と探索方法，(iv)検索情報の表現方法，などがある。

情報検索の技術については，昭和40年に社団法人日本電子工業振興協会で発行した報告書があるが，その当時から現在にいたるまでに，電子計算機のソフトウェアの面では，オンラインリアルタイムによる情報処理技術の面では進歩があったが，情報検索の技術では特にいちじるしい発展はなかった。この期間は，これまでに開発された技術の実用化の問題が検討され，いくつかのシステムが開発された実用化の期間であった。

そこでこの報告書では，「新しい検索技術と検索システムに関する基礎理論の体系化」という標題で，第1章では(i)情報検索という作業の概略，(ii)情報の整理（システム内での情報の表現）の理論的な考察，(iii)蓄積情報のファイル構成と探索方法のハードウェア，ソフトウェアの両面からの体系的な解説，第2章，第3章では，大型電子計算機を使用したいくつかの情報検索システムについて，その扱う情報の種類，ファイルの大きさ，システム作成に要した人員，費用などについてそれぞれ解説した。また付録としてランダムアクセスメモリ（磁気ディスク，磁気ドラム）のハード的機能を一覧表にまとめてある。

オンラインリアルタイムで計算機と会話しながら情報のファイルの検索をするという情報検索シス

テムでは，処理時間（ファイル探索に要する時間など）をできるだけ小さくするというような，ある探索時間を限定し，その条件のもとにファイルを構成しなければならない問題がある。またファイルの探索手順もバッチ処理によるファイル探索手順と形式的に多少異なる。すなわち最初の会話である範囲の情報を探索し，何回かの会話のやりとりで，その範囲を縮少し，利用者が満足するような回答が得られるようにファイルを設計する必要があると考えられる。しかし，オンラインリアルタイムでもバッチ処理でも，情報ファイルの構成と探索技術の面で本質的に異なるものでない。そこでこの報告書ではオンラインリアルタイムの情報検索システムのファイル構成および探索技術については特筆することを避けた。

なお，本報告書の作成に関しては，いろいろの点で下記の方々の御協力，御教示をいただいた。深く感謝する。

関根智明　（慶応大学）
小林功武　（ユニバック総合研究所）
坂本徹朗　（東洋レーヨン）
久保末沙　（日本ＩＢＭ）
笹森勝之助　（日本科学技術情報センター）
大駒誠一　（慶応大学）
昆野誠司　（管理工学研究所）
大瀬貴宏　（慶応大学）
秋元宏　（慶応大学）
松下真佐子　（旭化成）

目　　次

第 1 章　情報検索の理論

1. 1　情報検索の概略

現在，一般にいわれている「情報検索」とは，「情報を貯えておき，必要に応じてこれを取り出して使用する」ということである。

これを狭義に解釈すれば，「ある事柄についての情報を集め，これをファイルに貯えておき，問合せに応じて，蓄積してある情報そのものをファイルから探し出すこと」となり，決して，「貯えられているいくつかの情報から新しく情報を作り出し，これを問合せの回答とする」ことは行なわない。現在実用化されている情報検索システムは，ほとんどこの狭義の情報検索である。

たとえば，ある情報のファイルに次のような情報が貯えられているとする。

(i)　虎は中国とインドに棲む。

(ii)　動物園には虎がいる。

(iii)　東京に動物園がある。

これだけの情報が貯えられているとする。これに対して，「東京に虎がいるか？」という問合せに対して，狭義の情報検索では，「YES」という回答を出さない。

情報検索という仕事には次の6つ作業がある。

1）情報の収集

2）情報の整理

3）情報のファイル作成

4）問合せの分析

5）情報のファイルの探索

6）情報の配布

2）の情報の整理では，無形の情報あるいはある種の自然言語で表現されている情報を，特定の表現（自然言語，記号コード，数値）に変え，また有用な情報の選択を行なう。

3）の情報のファイル作成では，これらの蓄積情報をファイルの探索の便宜を考えてファイルにまとめる。

4）では問合せを分析し，どの情報ファイルを，どのような手順で探索し，何を回答として出力するかを指定した「探索指令」を作成する。

5）で探索指令に従って，情報のファイルを探索する。

電子計算機を利用した情報検索システムでは，情報のファイルはランダムアクセスメモリ（磁気ディスク，磁気ドラム），あるいは磁気テープに作られる。ファイルの探索の中心になる作業は，ファイルの中の個々の情報と，問合せから作成した探索指令との照合であるが，ほとんどのシステムでは，

ランダムアクセスメモリから情報を主記憶（磁気コアメモリ）に読み出し，そこで照合を行なっている。

1.2 情報検索における最近の進歩

1.2.1 序　　論

1.2.1.1 はしがき

次に来たるべき社会は「超技術社会」であって，それは物財の生産を中心とする社会から，多様な情報を主体とする社会への転換である，といわれている。農業社会や工業社会では物質とエネルギー経済の主役を演じてきたが，未来の情報社会では無形の知的生産財が主役となる見通しであるというアッピールである。[1)]

この情報化社会を支えるものは，大型電子計算機と通信技術による時分割や即時処理，多重処理のシステムと大容量ファイルの情報検索技術とであろう。

大型電子計算機を時分割で利用する考えは，１９５９年にC．ストライキーとI．マッカーシーによって提案され，この概念がマサチューセッツ工科大学に引きつがれて，１９６１年に具体化された。それ以後，人間と電子計算機との間の応答特性を改善させるものとして，また情報産業の技術的基盤として，急速な勢いで脚光を浴びている。

ハードウエアの面では，このように着々と足固めがなされてきているが，もう一方の柱である大容量ファイルの情報検索システムについては，その開発がまだ緒についたばかりである。その嚆矢として注目されるのは，米国におけるChemical Abstractsの情報検索システムと，National Library of Medjcine のMEDLARSシステム，わが国におけるＪＩＣＳＴ情報検索システムである。

このような大規模の情報検索システムを考えるときに要点となるものは何であろうか。それは第１に大容量ファイルの維持，検索の問題であり，第２に多数の利用者を同時にこなす時分割即時処理――さらにその機能をフルに利用した人間本位の使用法の問題であろう。

これらの問題を論ずるための導入として，情報検索の概観と最近の傾向にふれておく必要がある。

1.2.1.2 情報検索

情報検索は，情報の収集，整理，提供の全工程を対象とする技術である。そのうちの整理の部分を担当するのは索引法であるが，整理がうまく行なえなければ，収集も提供も行なえないということから考えると，索引法は情報検索の中心的部分を占めているといえる。そしてまた技術的に特に問題が多いのも索引法である。

索引は，対象物（文献など）の記述，索引語，それらのファイル中における住所，の３要素からなる。索引法とは与えられた（情報担体としての）文献より上記の３要素を取出してファイリングを行ない，要求に応じてファイル探索を行なう技法をいう。これから索引化の過程は，次の４段階

からなり，それぞれ別名がつけられている。

a　索引語の抽出＝主題分析

b　索引語のコード化＝コード化

c　ファイルの構成＝蓄積（狭義）

d　とりだし＝検索（狭義）

この索引システムの概念図をあげれば図1.2.1のとおりである。すなわち，収集された文献は，まずファイルのための住所記号がつけられる。ついで主題分析が行なわれ索引語が決められ，書誌事項（文献の記述：著者，タイトル，出典など）の記録が行なわれ，それらのコード化が行なわれる。そして，住所記号に従って文献自身も索引の要素もそれぞれファイリングされる。その後，検索の要求が起った時は，文献の分析と同じ過程を通って索引語コードを決め，ファイル上で照合を行なえばよい。

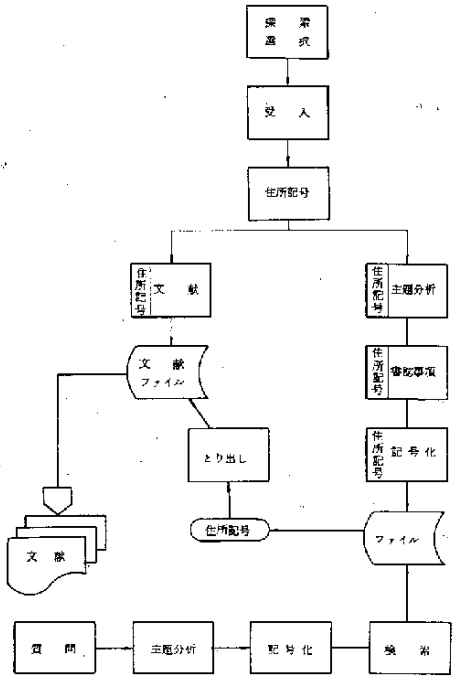


図 1.2.1　索引システムの概念図

現在，索引法はひじょうに精密なものになりつつある。人によっては，このような精密化は技術に溺れるものであって，実際上は無用の長物となすとの主張もある。確に，小規模の索引では，それが一覧可能である故に，どのような方法をとっても問題はでてこないし，自家製の索引（自分で作って自分で使う）では，記憶の助けがあるから，やはり問題は少ないのがふつうである。しかしながら，数万，数十万の項目を持つ大規模な索引では，一覧不可能，記憶の援助がないなどの故に，複雑な問題が生じ，経験法則だけでは律し切れない。また同じファイルを作るにしても経済上の理由から最少限の項目で要求を充足しようとすると，そこに使われる技法も巧緻たらざるを得ないであろう。

現在の索引法は確に精密化してきてはいるが，一方においてそれらの適用は小規模ファイルにとどまり，未だ大規模システムの洗礼を受けていない。ここに技術の脆弱な面が存在するかもしれない。その意味において大規模化に向う現在において，もう一度その基礎を洗直してみる必要があろう。

1.2.1.3　情報の形式

情報は一つの無形物であって，それが伝達されたり利用されるためには担体が必要である。それはたとえば言語とか映像とかである。そしてそれがさらに伝達されるためには，活字記録とか電波

などの媒体にのる必要がある。

このようにして記録された情報は，内容的にはもちろん，形式的にもおよそ考えられるありとあらゆる形態をとって，伝伝達されている。それらは一見種々雑多なように見られるが，よく見ればそこに部分的にある一定の形式的な定着が行なわれている。

たとえば，「物性値データ」は各物質につき図1.2.2のような項目が扱われる。これらの項目範囲は細部では差があるが，中心的なもの密度とか誘電率などはすべてに共通である。

章	物性の種類
1気体の諸性質	$P-V-T$関係・臨界定数・屈折率
2液体の諸性質	密度・熱膨張係数・圧縮率・音速度屈折率・表面張力・誘電率・双極力モーメント
3熱力学的性質	熱容量・エンタルピー・エントロピー・自由エネルギー・熱力学線図・生成熱・燃焼熱
4蒸気圧	蒸気圧・融点・沸点・各転相潜熱・
5気液平衡	$x-y$関係・平衡比・アゼオトロープ・活量係数
6溶解平衡	溶解度・分配係数・溶解熱・活量係数
7粘性係数	気体・液体
8熱伝導熱	気体・液体・固体・プラントル数
9拡散係数	気相・液相・固相・シュミット数
その他	（総説・資料・ノモグラフ）

図1.2.2　物性定数3集の内容項目[2)]

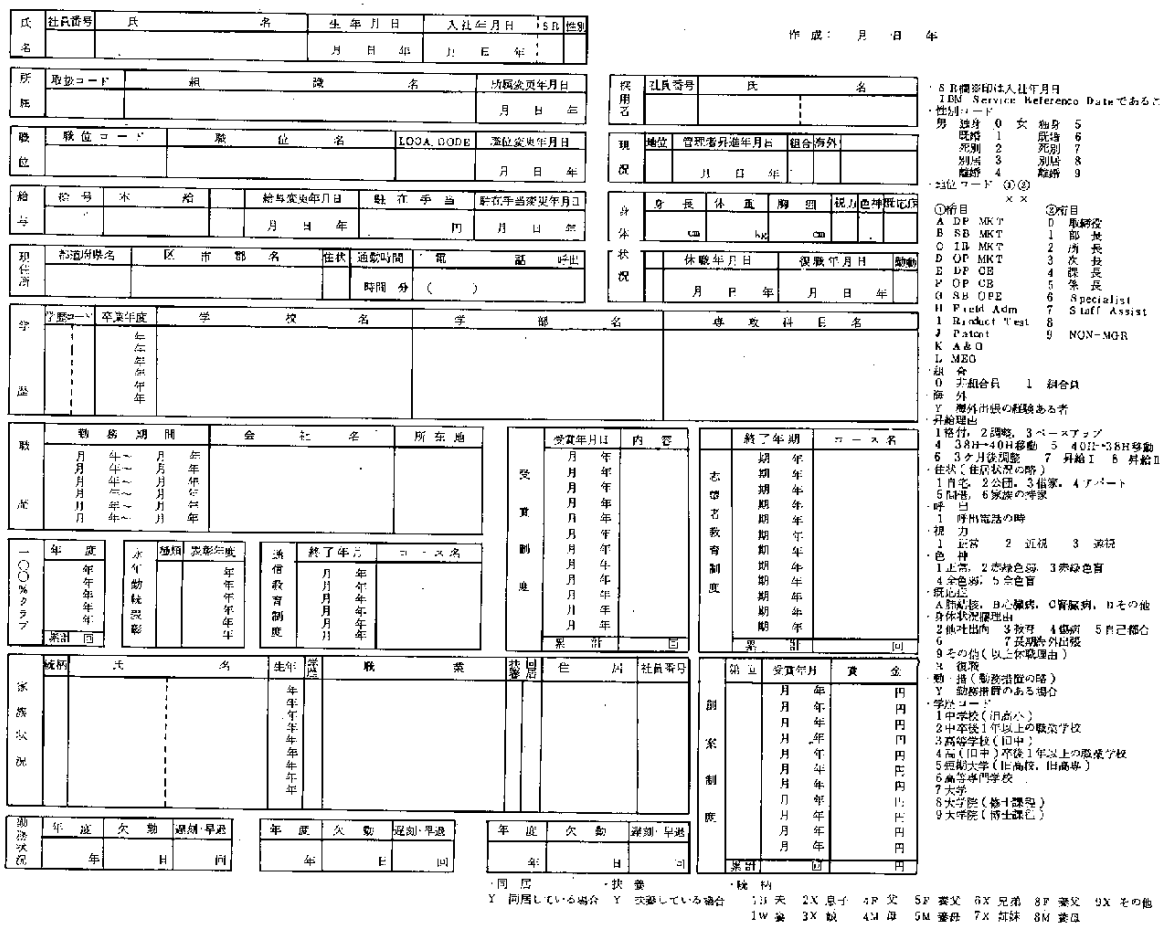
（化学工学協会編　丸善，1965）

また「人事データ」では，図1.2.3のような項目構成が一般的である。すなわち，①社内歴：入社後の給与，所属，職位など，②人事記録：入社時の社員の記録（住所，学歴，家族等）と入社後の記録（教育，勤務状況等）のすべてを含んだもの，③評価記録：個々人についての属性の他，入社試験の点数，面接・評価記録，人事考課の結果，社内講習受講歴とその評価，等々，④特技記録：技能，経験，特有の資格等の記録である。

これらの定型的記録物の特徴は，項目が比較的定着していて，その項目の意味解釈をめぐって意見が分れないことにある。たとえば"入社時の社員の住所"，"ある物質の融点"の意味が不明確だという人はいない。そしてこれらの項目の内容が，後に索引化を行なった場合に，索引語になるのであるから，主題分析は簡単であるといえる。いかなる観点からその記録を索引すべきか，というその観点を，カテゴリーと呼ぶことにすれば，これらは索引カテゴリー定着型である。

これに対し，いわゆる文献と呼ばれる記録物は未分化であって，索引カテゴリー不定型である。索引カテゴリーは主題分野によって異なり，その上同一主題分野中でもシステムによって一定しない。たとえば，化学において，文献を「物質名と反応」だけで表現しようとする試みもあれば，「原料，反応，生成物，触媒，構造，測定，操作」などの観点から索引語を抽出しようとする方向もある。

そして出てきた索引語のうちでも，たとえば化合物名は特異な存在で，いわゆる分析的と呼ばれる構成をなす。たとえば「スルファニル酸」という索引語は「ベンゼン環── アミノ基── スルフォン基」などと分解できる。



・SR欄※印は入社年月日
IBM Service Reference Date であることを示す。

・性別コード
男　独身　0　女　独身　5
既婚　1　既婚　6
死別　2　死別　7
別居　3　別居　8
離婚　4　離婚　9

・担位コード　①②
×　×

①桁目	②桁目
A DP MKT	0 取締役
B SB MKT	1 部長
C IB MKT	2 次長
D OP MKT	3 次長
E DP OB	4 課長
F OP OB	5 係長
G SB OPE	6 Specialist
H Field Adm	7 Staff Assist
I Product Test	8
J Patent	9 NON-MGR
K A&O	
L MEO	

・組合
0 非組合員　1 組合員

・海外
Y 海外出張の経験ある者

・昇給理由
1 格付，2 調整，3 ベースアップ
4 38H→40H移動　5 40H→38H移動
6 3ケ月後調整　7 昇給Ⅰ　8 昇給Ⅱ

・住状（住居状況の略）
1 自宅　2 公団，3 借家，4 アパート
5 間借　6 家族の寄宿

・呼出
1 呼出電話の時

・視力
1 裸眼　2 近視　3 遠視

・色神
1 正常，2 赤緑色弱，3 赤緑色盲
4 全色弱，5 全色盲

・既応症
A 肺結核，B 心臓病，C 胃腸病，D その他

・身体状況課理由
2 他社出向　3 教育　4 傷病　5 自己都合
6 7 長期海外出張

9 その他（以上休職理由）
3 復旧

・勤…掛（勤務措置の略）
Y 勤務時間のある場合

・学校コード
1 中学校（旧高小）
2 中卒後1年以上の職業学校
3 高等学校（旧中）
4 高（旧中）卒後1年以上の職業学校
5 短期大学（旧高専，旧高専）
6 高等専門学校
7 大学
8 大学院（修士課程）
9 大学院（博士課程）

・同居　・扶養　・続柄
Y 同居している場合　Y 扶養している場合　1F 父　2X 息子　4F 父　5F 義父　6X 兄弟　8F 義父　9X その他
1W 妻　3X 娘　4M 母　5M 義母　7X 姉妹　8M 義母

図2.13　人事ファイル　（日本IBM社）[3)]

古小路四朗，コンピュータによる人事管理，
日本経営出版会　昭43　p．84－85

また一般の索引語にも，この方法は使われている。たとえば「寒暖計」を「空気——温度——測定」などと分解するのがこれである。

すなわち，索引カテゴリー不定型にあっては，与えられた記録物をいかなる観点で分析するかが不定であるのみならず，使われる索引語自身の型も不定である，という特徴がある。

情報検索をシステムとして動かすためにはこれらの不定型の記録物を定型処理に持って行く必要がある。これが主題分析の問題である。

情報検索においては，その典型的処理対象としてすぐに「文献」を選ぶか，その理由は上記のような扱い難くさがあって，この不定型が処理できるようなら，定着型は十分に可能であるという見通しがあるからである。

1.2.1.4　最近の傾向

最近数年の間に情報検索の分野は急速に拡張し，その関係者として言語学者，数学者，電子計算機専門家などが名を連らねるに至った。その当然の結果として，いろいろ新らしい手法が持ち込まれ，道具立てが豊富になってきている。たとえば，索引語抽出のための統計的相関法とか，構文分析法，グラフ理論等々がある。

しかし，この手法の提案のされ方は，ある手法が適用された結果欠陥が発見されその修正案として提示されたものが大部分なので，総合的評価がされていない。また，提案なので定性的にははっきりしているが，定量的データの乏しい感みもあった。

最近の傾向の一つとして，これら手法の定量的評価が行なわれるようになったことが挙げられる。本章では，これら手法間の比較定量の実験結果として，ＳＭＡＲＴシステムの報告を扱う。

もうひとつの傾向として情報検索のオンライン・リアルタイム・システムの適用がある。今までにもこの傾向はあったが，それらは計算機を安く使おうという計算機本位の考え方が強かった。しかし最近では，情報検索システムの弱点が何であるかを考え，その補強策としてオンラインリアルタイムが必要であるという方向に変ってきている。その一例として本章ではGoodyear Aerospace Corporationにおけるテストを紹介する。

基本的な方向では情報検索システムの形式化の傾向があげられる。情報検索には先に述べたように種々の手法があるが，それらに共通的に存在する性質，すなわちそれが情報検索手法の本質であるが，それは何か，を問おうとするのである。また，より総合的，大規模なシステムを設計する際に，どのような手法が問題のどの範囲までを本質的にカバーするものかを完全に押えておく必要があるが，そのための見通しをつける意味もあるであろう。この一例として本章ではUnivcrsity of Pennsylvania の Weaver の報告を扱う。

その他の傾向としては，情報検索システムをシミュレートしてその特性を見ようとする試みがあげられる。情報検索システムは複雑なシステムであって，その振舞いが予め定量的には把え難い。したがって，シミュレーションを行なってみるには好個の対象である，といえる。ただこれまで報

告されたのは，ごく単純なシステムを想定して行なわれているので，その結果が実際上の参考になるとはいえない。むしろ，技法的に可能なことを試している段階であるといえる。その意味で，紙数の関係もあって本章ではとり上げなかった。

以下それぞれについて詳細を紹介する。

1.2.2 種々な検索手法間の効率比較

1.2.2.1 はしがき

情報検索の手法については，今までいろいろな人が各種の方法を考え発表してきた。そこには言語学者あり，論理学者あり，数学家あり，計算機専門家ありで，それぞれの立場から情報問題解決への新方法，新用具の提示が行なわれてきた。たとえば，言語学者は自然語の構造を直接分析してその間の関係を引出し検索に使うという構文分析法を提示したし，また統計学者は頻度および相関分析の手法を応用して文献中の定数を見つける方向を示唆し，図書館員，ドキュメンタリストらは分類法や同意語辞書，Thesaurusを持ち出す，といった状態であった。

そしてそれらの手法間には相当な討論が繰返され，お互いに自己のものを良しとして譲らぬ風潮も見られた。索引語抽出のベースについてもタイトルだけで十分有効とするもの[6]，タイトルより抄録，抄録より全文を主張するもの[7]などがあるし，索引語の構成についても分類派とキーワード派との論争はたえなかった。

このような状況下にある現在において重要なことは，新らしい手法を考え出すことよりも，現存の手法間の比較を行なってどれがもっとも有効な手法であるかを決定することである。さらに一歩進めていかなる環境のもとではいなる手法（あるいは手法の組合せ）が有効であるか，を決めることである。

SMARTシステムによるLesk の実験報告[4]はこれらの点につきひとつの解答を与えるものであろう。以下彼に従いその実験結果を述べる。と共にSMARTシステムの概要を付記する。Leskの実験内容がわかるためには，SMARTシステムの構造がわからなければならないからである。[5]

1.2.2.2 Leskの実験結果（まとめ）

Lesk の実験では次の項目が判明した。

a 文献の長さ：抄録ベースの処理はタイトル・ベースの処理よりも検索率が高い。しかし全文ベースと抄録ベースの比較では差がない。したがって費用のことを考えると全文ベースの処理は損である。

b 語処理：同意語辞書を使用する方法は，使用しない場合に比べて効率がよい。

c 句検索：（句を認知するための）句辞書を使用する方法は，同意語辞書を使用する場合に比べてあまり効果がない。

d 語相関：語相関を使用する方法は，同意語辞書使用に比べて効率が落ちる。

e　概念体系参照：語の親子兄弟関係を使って質問語を展開する方法は使用しない場合に比べて効率上昇は見られない。

f　文献群別検索：文献を予め関連群に分けておき，関係部分だけを検索するという方法をとると，検索率を落すことなく，検索時間を短縮することができる。

g　索引との比較：ＳＭＡＲＴシステムは全文検索，すなわち全文*を無処理でファイルに入れておき，質問がきた時点で全文と質問の照合を行なって検索する。それと人が予め文献を索引化してファイルに入れ，質問時点でその索引をひくというやり方を比較した。その結果，両者に差があるとはいえない，という事実が発見された。

1.2.2.3　試　　料

実験に使用した文献ファイルは，つぎの３つである。

ファイル名	ＩＲＥ	ＡＤＩ	ＣＲＡＮ
ファイル内容	１９５９－１９６１ Trans. ＩＲＥ記載の抄録	Proc. 1963 ADI Conf.	Cranfield実験に使用したもの
主　　題	電子計算機	ドキュメンテーション	航　空　学
文　献　数	７８０	８２	２００
文献長(語)	１００	１４００	１８０
質　問　数	３４	３５	４２
索　引　化	無	無	有

1.2.2.4　評　価　尺　度

種々の手法の検索率を測る尺度として，再現度（ｒｅｃａｌｌ）と精度（ｐｒｅｃｉｓｉｏｎ）の２つを用いる。再現度とは検索文献数と適合文献数の比，精度とは検索文献数とそのうちの質問に適合した文献数との比である。

実務上にこれらの測度にある下限値を設けてその値以下の文献は回答として出さないようにする。しかしこの実験の場合はそうすると，手法間の比較に誤判断を生ずるおそれがあるので，各質問ごとに全文献の相関係数を計算し順位表を作る。そしてその表の位置から使用した検索手法の効率を判定しようというのである。その順位表の首位近くに適合文献が集まるという程度が，再現度であり，不適合文献が首位からできるだけ遠ざかるという程度が精度ということになる。

かくて適合文献の順位ｒｉによってつぎの４つの基本測度が定義できる。

ｉ）　順位再現度＝ $\sum_{i=1}^{n} i \Big/ \sum_{i=1}^{n} r_i$

＊ここにいう全文とは自然文の意味で文献全体の場合もあるし抄録の場合もある。

ii) 対数精度＝ $\sum \log i / \sum \log ri$

iii) 正規化再現度＝ $1-(\sum_i ri-\sum_i i)/n(N-n)$

iv) 正規化精度＝ $1-(\sum_i \log ri-\sum_i \log i)/\log\binom{N}{n}$

記号法：n＝適合文献数

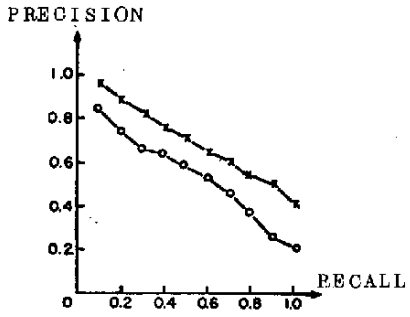
N＝ファイルの総文献数

これらの測度の数値は0～1の範囲で変化する。しかしこれらの数値だけで検索手法の優劣を比較するには多少困難を感ずるので再現度＝精度曲線を計算表示することにした。これは各再現度数に対しその精度をプロットしたものである。

次にこうして計算した各手法間の差は統計的に有意か否かを検証しておく必要がある。このためStudent のt検定とsine 検定を使用した。

1.2.2.5 文 献 長

初期の機械検索は文献タイトルをインプットして用いた。これは精度があまりよくなかった。人手によるシステムの最大の長所は全論文が処理のベースになっていることである。そこで機械検索でもできるだけ全論文に近いものを用いる，という着想が出るのは当然である。Lesk はタイトル，抄録，全論文のそれぞれを使った比較結果を出している。それによると抄録ベースの処理はタイトルのみの処理に比べて格段に効率が上る（図2.1.4参照）。ところが，一歩進めて抄録と全論文の比較ではほとんど差がないことが分った（図2.1.5参照）。抄録処理と全論文処理を費用の点から見れば，全論文処理はここに用いた短いものでも抄録の10倍もの費用がかかる。したがって，抄録ベースの処理が最も良い方法だということができる。

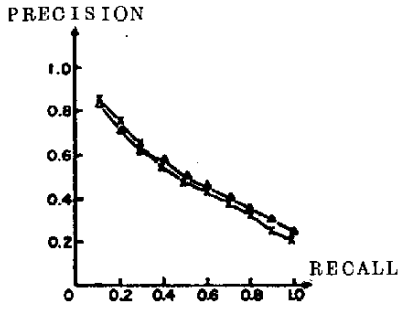


○－○ タイトル（Harris 3辞書使用）

×－× 抄　録（　〃　）

ファイル　IRE

図1.2.4　タイトルベースと抄録ベースの比較



×－× 抄　録（同意語辞書使用）

○－○ 全論文（　〃　）

ファイル　ADI

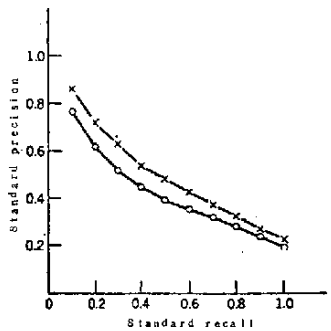
図1.2.5　抄録ベースと全論文ベースの比較

1.2.2.6 語 処 理

ＳＭＡＲＴシステムでは全文を予めファイルしておき，それを質問文と比較して一致するものを取出す。この場合何を一致と見るかで，いくつかの手法に別かれる。ｗｏｒｄとｗｏｒｄｓすなわち単複形のみを同じと見なすのが単複辞書方式，単複形に加えてさらにｓｌｏｗ，ｓｌｏｗｅｒ，ｓｌｏｗｌｙなどの語尾変化群も同じとみなすのが語幹辞書方式である。これらの辞書は計算機によって自動的に作られる。さらに一歩進めてｐｌａｎｅｓとａｉｒｃｒａｆｔなどの同意語関係を同じものと見なし，意味のあいまいな語や重要でない語を区別できるようにしたのが同意語辞書方式である。これは人手で作られる。

ＡＤＩおよびＩＲＥファイルでは単複辞書方式よりも語幹辞書方式のほうがややよいという結論がでた。ところがＣＲＡＮファイルでは逆に単複辞書方式のほうがよいという結果になった。この差の説明として，ＬｅｓｋはそれはＣｒａｎｆｉｅｌｄファイルの特質によるとしている。Ｃｒａｎｆｉｅｌｄファイルはひじょうに特殊な主題を扱っており，安定した術語が多い。したがって語の意味の方に感度が高く，語形変化に鈍くなる傾向がある。

同意語辞書方式では例外なく他のものよりも良いという結果が出た（図1.2.6参照）。



○－○ 語幹辞書方式（抄録ベース）
×－× 同意語辞書方式（抄録ベース）
ファイル ＡＤＩ

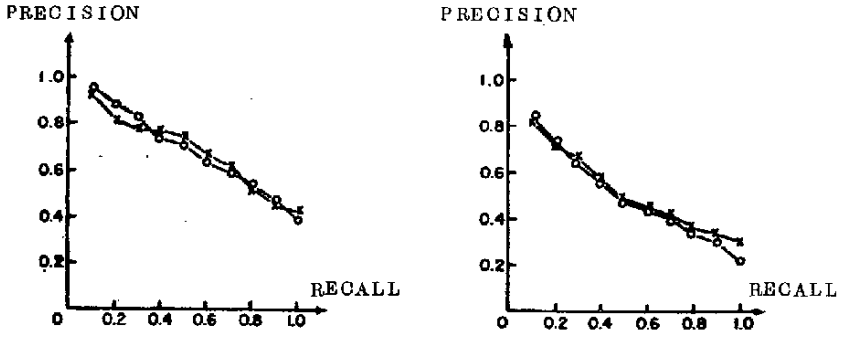
図1.2.6 語幹辞書方式と同意語辞書方式の比較

1.2.2.7 句 検 索

ＳＭＡＲＴシステムでは，２種の句検索手法を持っている。句とは概念あるいは語の組である。ひとつの文章中に共に出現した概念（語）の組は句と見なすことにする。これが第１の手法である統計的句処理と呼ばれる。第２は句の構文的構造も勘案する方法で構文的句処理と呼ばれる。たとえば，統計的句処理では一文中にｅｑｕａｔｉｏｎとｄｉｆｆｅｒｅｎｔｉａｌ が出てくれば，ｄｉｆｆｅｒｅｎｔｉａｌ ｅｑｕａｔｉｏｎと見なすのに対し，構文的句処理ではｅｑｕａｔｉｏｎ が名詞であり，ｄｉｆｆｅｒｅｎｔｉａｌはそれを修飾する形容詞として現われるのでなければｄｉｆｆｅｒｅｎｔｉａｌ ｅｑｕａｔｉｏｎ とは認めない。両者とも予め作られた，同意語辞書に付随する句辞書を使う。

構文的句処理で小規模の実験を行なったところ，悪い構造の句を選別したがさらにそれ以上の数の良い句も拒否してしまった。現在のところ速くて正確な構文処理プログラムがないため，それ以上大規模の実験ができない，ということである。

統計的句処理による実験結果は図1.2.7に示した。句辞書による検索は同意語辞書による検索と比べてほとんど差がない。これは，しかし，統計的句処理が正しい構造の句を見つけられないとい



a）
○—○ 同意語辞書（Harris 3使用）
×—× 統計的句処理（ 〃 ）
ファイル IRE
17質問

b）
○—○ 同意語辞書（全文）
×—× 統計的句処理（全文）
ファイル ADI
35質問

図 1.2.7 統計的句処理と同意語辞書方式の比較

うせいではなく，たぶん句辞書の語数が少ないためである，とLesk はいう。SMARTシステムの句辞書は比較的少数の語にしぼり，"high temperature"とか"Computer controlled"など，句としてでなければ不明確な語になってしまうようなもの，を収録しているためである。

1.2.2.8 語 相 関

文献中においてある語が他の語といかなる関係で，何回くらい現われたという統計をとるのが索引語相関である。その結果はたとえば図1.2.8のようになる。[5] これは強い関係のある語の組の表であるといえる。

この統計をとった結果として，ひじょうに頻度の高い語の組と，頻度の低い語の組はあまり有用ではない。文体に影響される度合とか偶然的な出現による度合が大きいためである。この実験では3～50あるいは6～100の頻度値をもつものを対象とした。COS相関値でいえば，切捨値は0.45から0.75の間にあるものである。

その結果上記の頻度値範囲にあるものの約4分の3が関係のある語であるということが分った。それらの語の真の意味を調べずに表面的な意味だけを勘案した場合は，全体の20%くらいしか有用でないように見える。たとえば，"scheme"と"machine"の語の組では一見意味上の関係はないようにみえる。ところが文献内容を調べてみると"scheme"は実は"algorithm"を意味し，"machine"は"digital computer" を意味していることがわかった。そうなれば単なる統計的関連だけでなく意味上も関係のあることが明瞭になる。小規模のコレクションで

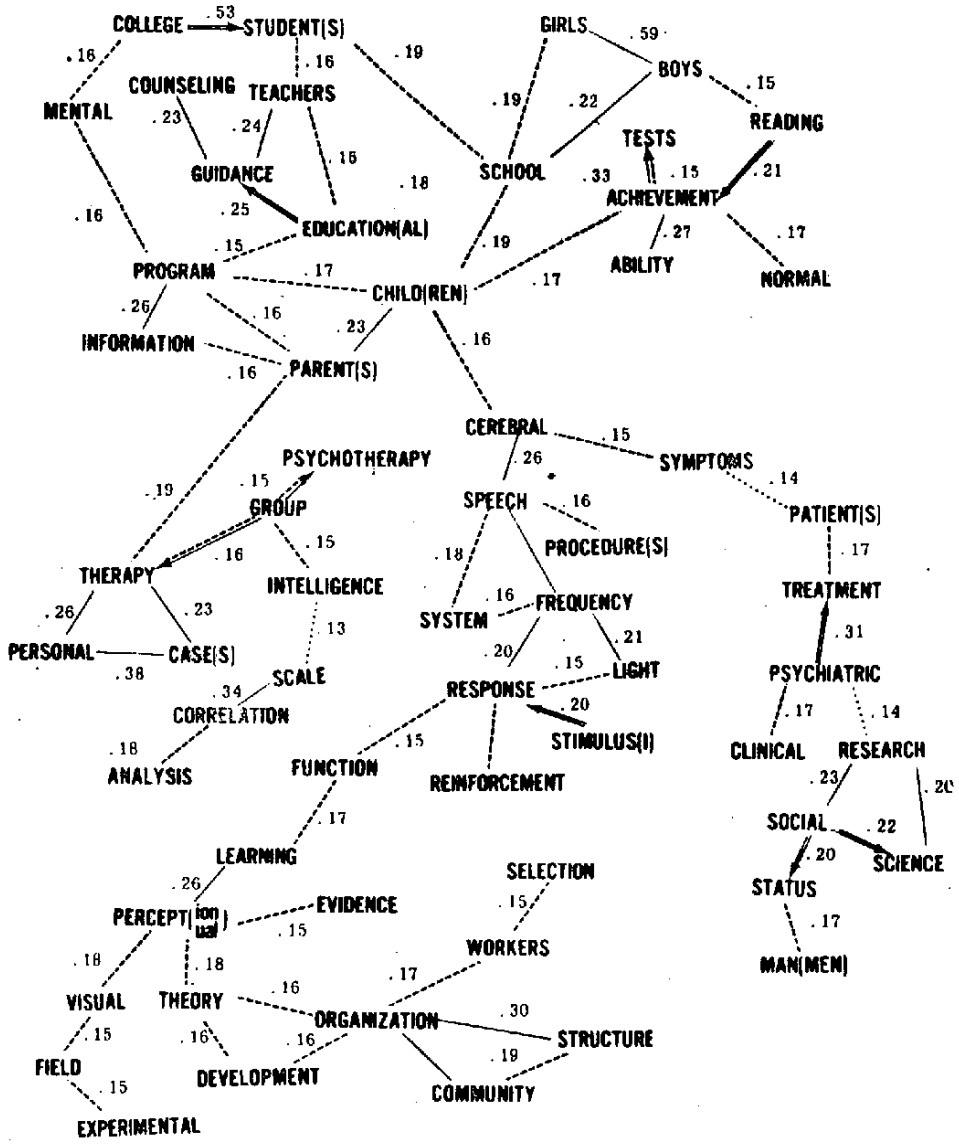


図 1.2.8　索引語の相関表

は，全体の語の組の約６０％がこの種のものであるので，語相関は語の意味上の関連を見るにはあまり役に立たない。

語相関法と語幹辞書方式とで検索を行ない比較した結果を図 1.2.9 に示した。ここで指摘できることは，Ｃｒａｎｆｉｅｌｄファイルの場合のみ語相関法がやや優位に立っているということと，再現率のある特定の範囲で語相関法が優位に立つことがある，ということだけである。

詳細な吟味によれば，語相関法はそうでなければ落ちてしまうようなものを救い上げる能力はなく，再現範囲に入っているものの順位をさらに上位に押し上げるくらいの役目しか果さない。

語相関法を同意語辞書方式と比較した結果では，図1.2.10に示すように明らかに同意語辞書方式のほうが優位である。

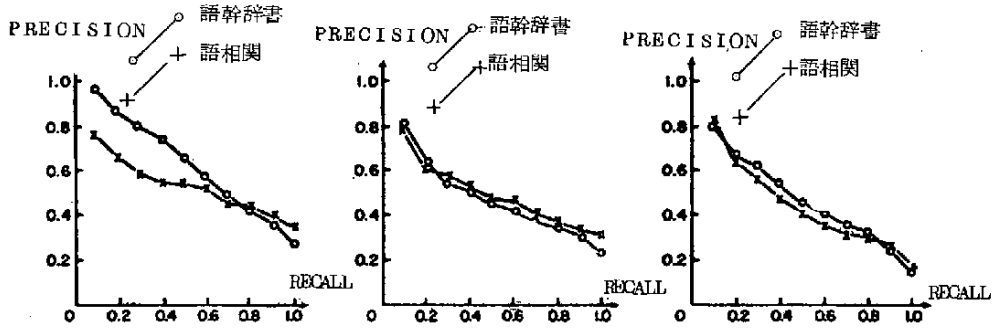


図1.2.9　語相関法と語関辞書方式の比較

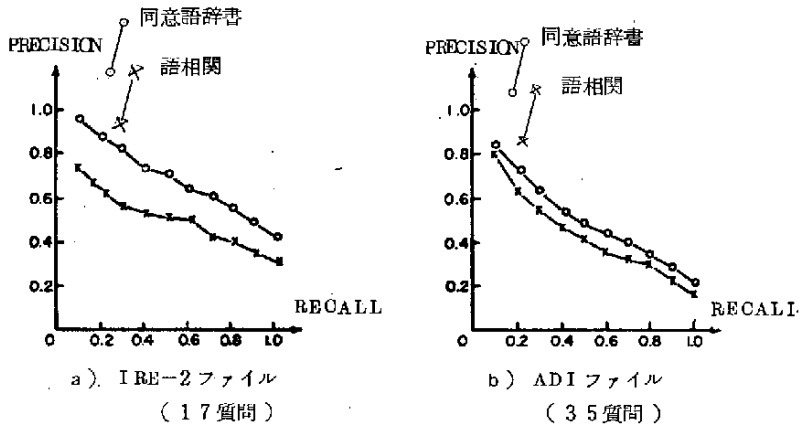


図1.2.10　語相関法と同意語辞書方式の比較

1.2.2.9　概念体系参照方式

体系的辞書（たとえば分類表）を使って質問の用語を修飾したり拡張したりする方法がある。た

とえば，ペニシリンの論文を探すのに，「ペニシリン」という語のみでなく，「抗生物質」（親概念）とか「ストレプトマイシン」（兄弟概念），「肺炎」（関連概念）なども使って探そうというのである。これらの拡張を計算機内で自動的に行ない，その上で検索しようというのが概念体系参照方式である。

この方法は，語の変更拡張がひじょうに多くなるため，効率には期待を反してずっと落ちる。ＩＲＥファイルについての実験結果では，親方向への参照がやや改善を見せたが，それも統計的に意味のある差ではなかった。

この方法は，むしろ悪い質問を質問者自身が改善する時の助けにするべきであって，検索プロセスの標準装備として組込むものではない。

1.2.2.10　判別式の比較

検索においては，質問と文献の類似度を計算し，それがある一定値以上のものを回答して取出す。その類似度の計算式にコサイン相関式と重複相関式の２種を使い，それぞれの効果を実験によって確めた。

コサイン相関式とは，質問の概念ベクトルを文献の概念ベクトルとのなす角度を測定するものである。その式は次のとおりである。

$$r_{cos} = \left(\sum_{K} W_k V_k\right) \Big/ \sqrt{\sum_{k} W_k^2 \sum_{k} V_k^2}$$

記号法：W_k ：質問ベクトルにおける第ｋ番目の概念の重み

V_k ：文献ベクトルにおける第ｋ番目の概念の重み

第ｋ番目の概念が現われないときは重みは０である。

重複相関式は，小いベクトルが大きいベクトルに含まれる程度を現わすものである。その式は次のとおりである。

$$r_{overlap} = \left(\sum_{k} \min(Wk, Vk)\right) \Big/ \min\left(\sum_{k} Wk, \sum_{k} Vk\right)$$

記号法はコサイン相関式と同じ。

コサイン相関式は全文献と全質問を勘案してその中で最も質問に適合した文献を選出する。重複相関式の行なうところは一般のキーワード検索で行なわれているプロセスと同じであって，質問の概念のセット全部を持っている文献のみを検索するだけで，それ以外の情報を持つ文献の区別はしない。

実験の結果，重みづけをした概念ベクトルをコサイン相関式で判別して検索する方法のほうが，論理ベクトル（その概念が存在するか否か）を重複相関式で判別して検索する方法よりよいということが分った。

1.2.2.11　文献群別検索

ＳＭＡＲＴシステムではふつう各質問ごとに全文献を比較する。これは文献数が多くなれば検索の費用がそれに比例して高くなるということである。そこで文献を予め何らかの基準たとえば主題別に従って群別しておき，質問に関係のある群だけを探すようにすれば，費用は格段に安くなるであろう。しかしながら，群別検索はいわばサンプル検索であるから母集団自身を探す全検索に比べれば検索率はどうしても落ちるであろう。それがどの程度に落ちるものかが問題である。これを実験で確めた結果が図1.2.11である。

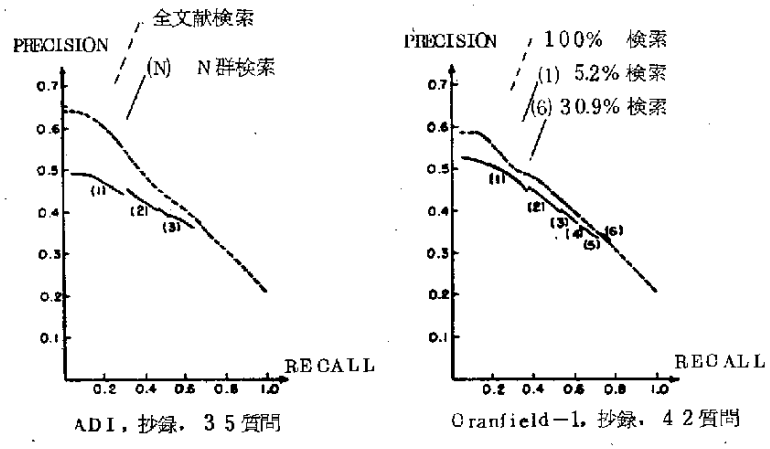


図1.2.11　群別検索と全文献検索の比較

点線は全文献検索の場合の検索率，実線のうち(1)とあるものはひとつの群だけを検索した場合，(2)とあるのは2群の検索，(3)は3群検索，等々を示す。1群検索では精度は大幅に落ちる。3群検索になると精度はほとんど全検索に近くなる。そして注目すべきは3群検索でも全文献のわずか20％を探しただけという事実である。しかしながら再現率はさほど上らず，3群検索では全文献の場合の６０％が上限となっている。

文献を群別したと同じ方法で質問も群別し，それぞれ対応する群で探すことにすれば，さらに検索スピードを上げることができるであろう。この方向でさらに実験を続行すると報告されている。

1.2.2.12　索引との比較

航空学文献のコレクション（ＣＲＡＮファイル）は，人手で作った索引がついていて，電子計算機に読めるような型式になっている。索引は1文献あたり３０語以上で重みずけが行なわれている。伝統的な考え方に従えば，全文に対して詳細な分析を行なうＳＭＡＲＴシステムのほうが，人手で作った索引よりも検索率が良いと考えられる。

しかし実験の結果は，図1.2.12に示すようにほとんど差がない。

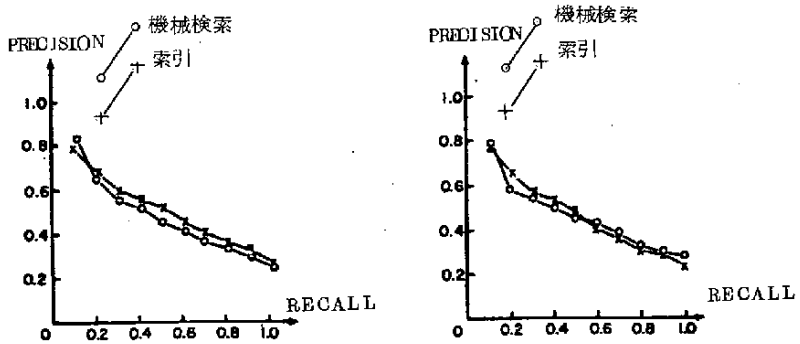


図 1.2.1 2　人手による索引と機械検索との比較

Lesk の見解によれば，索引の統一性が比較的よく保たれる小規模のファイルとの比較では差が出ない，ということである。すなわち機械のメリットはどんなに文献数が多くなっても最初に決めた処理原則が統一的に保たれるという点にあるが，人手による索引では文献数が多くなればなるほど統一をとるのが困難になるからである。

1.2.2.1 3　システムの相互通信による検索

ただ一回限りの質問＝検索でてくる回答は，質問者にとってはおそらく不満であろう。その直接原因は，質問構成が悪いからであるが，そういう不良質問になる原因は種々あって，情報検索システムにとっては本質的な問題が少なくない。

第1に，質問者は自己のアイディアを完全には索引語で表現できない，第2に，システムではどのような索引語および構成論理が使われているか不明で，質問との間にずれが出る，第3に，質問中には必ずなければならない索引語とあってもなくてもよい索引語とがあって，それらの取捨は回答の出方によって決めたほうがよいものであるが，一回の質問では皆同じ比重で扱われてしまう等々である。

これらの難点を是正するために，回答結果を質問者にフィードバックし，質問を修正させるという方法を考える。まず第1回の質問の結果出てきた回答文献を質問者が適合するものとしないものに分ける。計算機は適合文献と不適合文献の概念ベクトルを比較する。すなわち，適合文献のベクトルで質問にないものを加え，不適合文献のベクトルで質問にあるものは除くのである。そうして修正した質問を使って再検索を行なう。これによって適合文献数は増加する。この方法は適合文献がひとつも検索されない場合でも使うことができる。

場合によっては，最初の回答によって完全に考えて変えて，全く新らしい質問を作ることがある。しかし，これでは安定に時間がかかるから次のような方法が，むしろ望ましい。計算機が質問に従って検索する前に，システム内のアルファベット辞書とか体系順辞書から，質問語に関係する語を打出す。そしてその中からより適切な語を選んで質問を修正させればよい。それに加えて，ファイル内におけるその語の頻度とか，代表的文献などを併せ表示するようにすれば，より有効であろう。

1.2.2.14　システムの概要[9)]

ＳＭＡＲＴシステムはＣＨＩＥＦと呼ばれるモニターによって統轄される。このモニターは，行なうべき処理の型をインプット指令として受取り，対応するサブルーチンを呼び出すように作られている。

その処理方式は，基本的には次の８種類ある。

a　一般処理方式

b　アルファベット辞書方式

c　概念体系方式

d　統計的相関方式

（ひとつの文中で共出現した語を基準とする。）

e　構文分析方式

f　統計的相関方式*

（一文献中の語の相関を基準とする。）

g　文献相関照合方式

h　辞書更新処理**

上記に加えて次の４つの辞書を使用する。

a　アルファベット順語幹辞書

（構文コードおよび意味コードもつけられている。）

b　アルファベット順語尾辞書

c　意味上の関連を示す概念体系辞書

d　基準句辞書

（構文処理の際参考にする。）

ＳＭＡＲＴシステムは，全文処理方式をとっている。すなわち，文献あるいは質問の全文章とか抄録を読み込んで処理し照合して，類似文献を見つけるというやり方をとる。逆にいえば，予め処理をして索引語をつけ，それらの索引語間で照合を行なって適合文献を見つける方法はとらないということである。

*　これはｄと原理的には同じなのでここでは扱わない。

**　手法のａ比較を理解するためには不要なのでここでは説明を省略する。

1.2.2.15　アルファベット辞書方式

文章が読みこまれると，各単語に文献コードと文章番号がふられ，切離される。その各単語はアルファベット辞書と照合され，構文コードと意味コードがつけられる。ここにいうアルファベット辞書は，実際には2つの辞書——語幹辞書と語尾辞書からなる。それらの辞書は図1.2.13のように木構造でファイルされており，その内容は図1.2.14のとおりである。

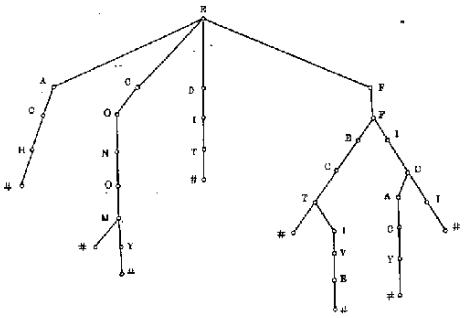


図1.2.13　アルファベット辞書の木構造

照合は左から右への字順で行なわれ，可能な限りの最長照合が行なわれる。たとえば，effectivenessはeffectで照合完了ではなくeffectiveまで辿られる。照合が確認されたら，その語は辞書中の意味コードと構文コードで置換えられ，以後種々の検索にはそのコードが使われる。したがってこれは必須の過程である。

語幹辞書だけでは，その語の全字が照合されない時は，残りの部分が語尾辞書と照合される。語尾辞書の形式は語幹辞書と同じであるが，意味コードは記入されない。たとえば図1.2.14の語幹辞書の語に対する可能な語尾は図1.2.15のようになる。

語　　幹	意味コード	構文コード
FACH	1096	081，008，012
ECONOMY	0064，0213	070
ECONOM	0064，0213	000*
EDIT	0063	043
EFFECTIVE	0064	001
EFFECT	0142	070，043
EFFICI	0064	000*
EFFICACY	0064	070
⋮	⋮	⋮

*　語尾変化をするものは構文コードが0で適当な変化にしたがって語尾辞書の構文コードが埋められる。

図1.2.14　語幹辞書の表現内容

英語の語尾変化にはいくつかの例外がある。すなわち，語幹語尾のYはIに変える，Eは省略，M，N，Pは2字重ねる，などが決まっている。

```
ECONOM:  ist, ists, ical, ically, ize, izes, ized, izing, ies;
   EDIT:  s, ed, ing, ion, ions, or, ors;
EFFECTIVE:  ly, ness;
   EFFICI:  ent, ency, ently, encies;
 EFFICACY:  ious;
  ECONOMY:  ies.
```

図1.2.15　種々の語幹に対する可能な語尾

その例をあげれば，次のとおりである。

HANDY	→	HANDILY
HOPE	→	HOPING
HOP	→	HOPPING
PROGRAM	→	PROGRAMMER

これらはすべて語尾解析プログラムによって適切に認知，処理される。

1.2.2.16　自動作成辞書

有用な辞書を作成することは，ひじょうに困難なことである。アルファベット辞書方式での最低のねらいは，インプット文の語と質問の語の照合一致をとることにあるから，語幹辞書に該当語がない場合でも，その最低線は保障されているのが望ましい。このため自動的に辞書を作成するという方法をとる。SMARTではこれをSimulated vacuous dictionaryとかnull dictionary と呼ぶ。ここでは自動作成辞書と呼ぶことにする。

この辞書は最初はぜんぶ空欄である。インプット文を処理する過程において，語幹辞書にない新語が出てくると，架空の意味コードをつけながらここに記入していく。この場合機械は意味の確認はできないから，意味コードは語ごとに異なったものになる。

そこで，この自動作成辞書を使って検索を行なうときは，質問と文献全文の（生の語の）一対一対応になる。

1.2.2.17　概念体系方式

語幹辞書をひいた結果として意味コードと構文コードがつくが，このうち意味コードは概念体系を表わしている。したがってこの体系処理を行なうのであれば，その準備はできている。

概念体系処理とは，上記の意味コード数字に従って図1.2.16のよう

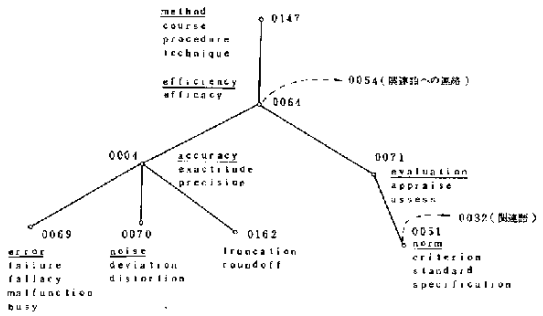


図1.2.16　概念体系の木構造

な木構造に作りあげる処理をいう。その木構造は，計算機内では図1.2.17のような1語36ビット3語一括のリスト構成で表現されている。

木構造に表現されると，それを辿って自由に上位概念あるいは下位概念，関連概念などに移ることができる。

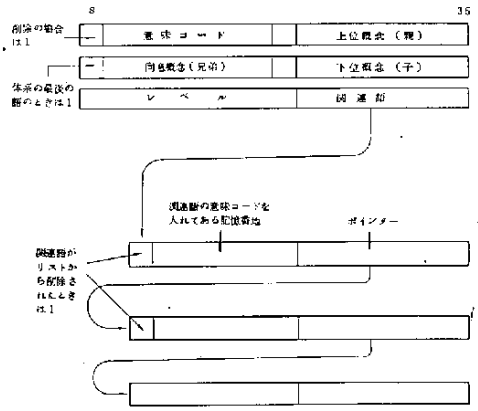


図1.2.17　概念体系のリスト構造

1.2.2.18　一文中の共出現語の相関

文章ごとにそれに出てくる単語をチェックして，語＝文一致行列を作る。たとえば図1.2.18のようになる。文S_1中に語T_1が1回出てくればM_{11}に1，T_2が2回出てくればM_{21}に2，等々とプロットする。これをひとつの文献中の語が（あるいは文が）終るまで繰返えす。終了したらT_i行をひとつのベクトルと考え，T_i行とT_j行の相関係数を計算して，それをその文献中における語T_iとT_jの相関と呼ぶことにする。

相関係数の計算には次のコサイン係数を使用する。これは視覚的には2つの語ベクトル間のなす角を計算するものである。

$$r_{cos} = \left(\sum_k W_k V_k\right) \Big/ \sqrt{\sum_k W_k^2 \sum_k V_k^2}$$

W_k ： T_i の第k番目（即ち文k）の数値

V_k ： T_j の第k番目（　〃　）の数値

	S_1	S_2	S_3	S_4
T_1	1	2	2	0
T_2	2	0	3	2
T_3	0	3	2	3

S_i ：文

T_i ：語

$$r_{cos}(T_1, T_2) = (2+6) \Big/ \sqrt{(1+4+4)\times(4+9+4)} \fallingdotseq 0.89$$

$$r_{cos}(T_2, T_3) = (6+6) \Big/ \sqrt{(4+9+4)\times(9+4+9)} \fallingdotseq 0.63$$

＊故に$r_{cos}(T_2, T_3)$のほうが夾角が小さく，関係が大きいということになる。

図1.2.18　語＝文一致行列とCOS相関値の計算

これから，ひとつの文中で現われる2語以上の概念で，文献中で繰返されているものは新らしい概念コードで置換えるとか，重要な文のみを抜出して文献を圧縮しよう，などの考えがでる。

検索に際しては，予め計算した相関関数ごとの照合（一定値以上は回答とする）か，この語＝文一致行列のままとし，質問の語＝一致行列との相関を計算し，一定値のものを回答するかである。その際もコサイン関数が使われる。

1.2.2.19 構 文 分 析

統計的相関法によって得られた重要な意味を持つ（と思われる）文は，ここで文法的に解析され，構文的関係が同じ語がひとかたまりにされる。また句にいつも同じであって，質問が文献中の同等と思われる句と照合され，構文的に等しい場合に検索されることになる。

構文分析法としてはKuno—Oettinger のMultiple—path Syntactic Analyzer を用している。[10] この構文分析の結果は構文依存型の木構成となるから，木の照合を通じて文章と句とを比較することができる。この照合にはSussenguth のグラフ照合法[11]が使用されている。

実際には，この処理のため"基準句辞典"が使われる。辞典には典型的な句，たとえば，"information retrieval" とか"Syntactic analysis of phrases" などが入っており（実際にはその対応する意味コード），さらに各語の構文コードと各語間の構文構造を示すコードが記入されている。

構文分析をされた文は，この基準句辞典のすべての項目と照合され，一致するものが見つけられる。たとえば，基準句が"information retrieval" であれば，"the retrieval of information" は一致するとされるが，"because the text contains secret information retrieval is vital" などは一致するとは見なされない。

分析の結果として，各基準句と文献中の文との間で一致した回数が記録される。そして一致した句の意味コードが，対応する文献の意味コード表に追加記録される。それはすぐに意味ベクトルに追加拡張され，検索に使用される。

1.2.2.20 文献相関照合方式

図1.2.18で述べた語＝文一致行列と同じ考え方で語＝文献一致行列も作ることができる。そして文献（列）単位でベクトルと考え，各文献ベクトル間の相関係数を計算する。計算式はやはりコサイン関数である。

これを検索に使うためには，質問も同じ考え方で処理し，質問と文献の相関係数を計算して一定値以上のものをとることにすればよい。

他の応用としては，相関値の類似のものを集めて文献群とし，検索の際に最も関連の深い群を探す，という方法も考えられる。

1.2.3　検索における機械＝人間の相互交信

1.2.3.1　は　し　が　き

機械検索においては，質問者と機械の間を結ぶものは，質問者の作った質問のみである。しかしながら，質問者は多くの場合自分の作った質問がうまく機械に通じるかどうかについて自信がない。自分の欲するものを正確に表現する技術に欠けるせいもあるであろうし，またかりに正確に表現しえたとしてもそれは質問者の知識的背景にたったことばであって，検索システムのコトバとくい違っているかもしれない，という心配もある。これは検索システムをいかに精密化してもコトバのもつあいまいさ，キーワードの守備範囲の不明確さなどに起因する本質的な問題なのでちょっとやそっとでは解決できない。

質問者と情報（文献）ファイルとの中継者が，それでも，いろいろと"気をきかせて"くれるので，この検索システムの困難もさほど障害にならない。しかし中継者が機械の場合は，統一がきびしいために，その困難が生の形で出てくる。

ふつうの質問者が質問キーワードを並べても，その中にはぜったいなくてはならないものから，どちらでもよいがどちらかといえばあったほうが良い，という程度のものまで種々雑多なレベルで含まれている。そのどちらでもよい程度のキーワードが，機械の中では，ぜったいになくてはならないキーワードと同じ比重で演算されるから，出てくる答に大きな影響を与える。たとえば，A，B，C，Dの4つのキーワードで，（A∨B）∧（C∨D）という質問を作ったとする。この場合ファイル中には概当するものがなく，A～ BとかC～ Dはあったとする。質問者にとっては，"no information" であるよりも，多少所期の回答と異なってもよいから，関連回答が出てくれる方が望ましい。しかし機械の"馬鹿正直さ"かげんから，そのような回答は得られない。これも機械検索の問題点のひとつである。この故にブール代数による一致演算は不適当で質問の全キーワードに対する関連度演算を行なって回答を出せ，というような提案が出る。[12] しかし，これはブール演算の欠陥によるものではなく，むしろ質問者と情報（文献）ファイルとの間に適切な交信がないことによるものである。

そしてこの質問者＝情報ファイルの間のくい違いは，情報ファイルが大きくなればなるほど加速度的に拡がっていくものなのである。

また心理的側面からいうと，質問者は出てきた回答以外に質問に対する答は情報ファイル中にない，ということを自分の目で確めないと気がすまないという傾向を持つ。関連文献とか周辺地域にまたがるような質問においては特にそうである。人に（あるいは機械に）文献調査を頼むと"思わぬ拾い物"がない，というその理由だけで，自身ではん雑な時間のかかる調査を行なう人が多い。このことは質問者の心理的不安感を如実に物語るものといえよう。

このような問題点を解決するものは何か。それは機械と人間の相互通信すなわちOn－line

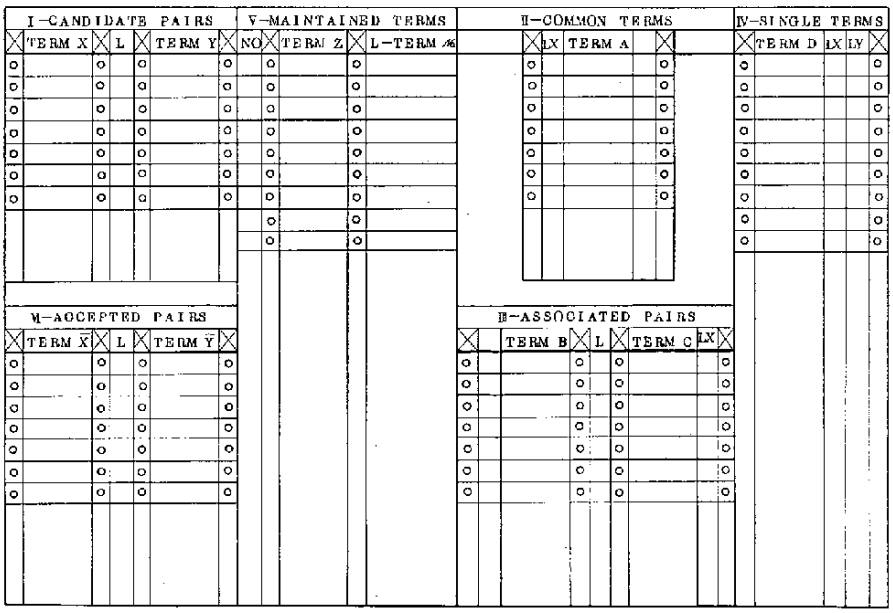
realtime system より他にない。On-line time-sharing system は今まで計算機の効率的利用という計算機側の理由によってのみ主張されてきた。情報検索の場合もそうであって，質問が生じた時にすぐ問合わせることができ，しかも割安である，ということぐらいしかその利点に挙げられなかった。したがって，オンライン情報検索システムといっても，オフラインの情報検索システムに多少の修正を施した程度のもので，その利点を生かすための焦点がはっきりしていないうらみがあった。

最近は上記のような問題点解決というポイントをはっきり打出したオンライン検索システムが，少しずつではあるが現われはじめている。たとえば，Treu[13] のデザインによるNASAファイルの検索実験などがそれである。[12] 以下そのアイディアを見てみることにする。

1.2.3.2 探索者の検索ゲーム

Treu は彼のオンライン情報検索システムの前半分を"探索者の検索ゲーム"と呼んでいる。すなわち，システムは2つの部分から成り前半分は質問者が自分のアイディアにぴったりした索引語を選ぶ段階，後半分はその索引語により該当文献を引出す段階である。そして後半の部分は普通の情報検索であるから，Treu のシステムの特徴は前半分にあるといえる。

検索ゲームは，質問者が第1次的に思いついたキーワードを計算機のコンソール・タイプライターに打込むところから始まる。そのキーワードは図1.2.19のような表示枠を持ったディスプレイ・



L　：関連度数（1～9）
LX：キーワードXの関連度数
O　：押しボタン（あるいはラインペンの指示場所）（左側は移動用，右側は消去用）

図1.2.19　キーワードの表示枠

コンソールに表示される。

次に計算機とのやりとりが開始されるが，それは3種の様式がある。

様式1：

<表示1> 質問者の打込んだキーワードの組と関連度数とが表示される。たとえば次のような形になる。これを候補組と呼ぶ。

ATOMOSPHERIC ENTRY —(0)— SPACECRAFT SHIELDING

関連度数は1から9まであり，1はもっとも関連の弱いもの，9は最も強いものを示す。0は便宜上4以上のものを示す。

次に計算機がそれらのキーワードと関連の深いキーワードを選び，関連度数と共にその下に表示する。関連キーワードは，あらかじめ計算機内に作ってあるキーワード相関行列*から指示された関連度数以上のものが選ばれ表示される。これを関連語と呼ぶ。

* キーワード相関行列の作成についてはTreu は何も述べていないが，常識的には次のようなものであろう。ある任意の2語がひとつの単位文（タイトル，自然文の文など）中に共出現した場合を1とする。このままではインプット・データ数に比例して共出現数が多くなるから，何らかの数式を使って正規化し，それぞれの数値範囲を決めて1～9の度数に割り振るのである。その数式（相関関数）としてTreuはGoodyear Aerospace Corporation[13]で開発した次のものを使っている。

$$F_{A,B} = \frac{f - \frac{ab}{n}}{1 + \frac{1}{2} \cdot \frac{ab}{n}}$$

fはキーワードA，Bの共出現頻度 $\frac{ab}{n}$ はfの期待値である。

それらは，すぐに取出せる場所に記憶させる。

<表示2> キーワード組の任意の2組の間で同じキーワードがあれば表示し，それぞれ対応する関連度数を表示する。これを共通語と呼ぶ。

質問者は，この共通語，関連語，候補組の中から任意の組合せを選び，新らたな候補組を作ることができる。またそれらの語の中から不要なものを消去し，必要なものだけ単純に記憶させておくこともできる。これを保存リストという。これらは後に<表示5>のところで一覧にして見ることができる。

様式2：

<表示3> ここでは最初の候補組の各語にそれぞれ対応する関連語を入れ替えた新らしい組合せを，関連度数つきで表示する。この時さきに消去された語は考慮に入れられない。そして，すでに候補組に入れられたものは表示されない。

質問者は，このようにして表示された組合せの中から良いと思うものを選んで候補組に入れる。候補組に入れるほどではないが，とっておきたいと思う語は保存リストに入れることができる。そ

の他のものは消去してもよい。候補組の語は自動的に保存されるから，保存リストに入れる必要はない。

様式3：

<表示4> これまで語として表示されたのに，候補組にも入れられず，消去もされなかったものが，関連度数と共に表示される。この時その語が関連して出てきた新キーワードの番号（XかY）も表示される。

質問者は，この親キーワード＝単語の組で有望なものを選んで，候補組に入れることができる。その他の単語で興味のあるものは，保存リストに入れることもできる。その他の語は，この段階で自動的に消去される。

質問者はこれまでの全過程を必要な回数だけ繰返えすことができる。また必要とあらば<表示5>として，今まで保存リストに入れた語を表示させ，新たな候補組を作るかどうか考えることもできる。

<表示6> 以上のようにして十分吟味して候補組を作ったならば，最終的にその中からいくつかを指定して（その他を消去して）決定組を作る。検索はこの決定組について行なわれる。

1.2.3.3 検索ゲームの例

検索質問：

What protective shielding against aerodynamic heating is provided for spacecraft during reentry?

（宇宙船が大気圏再突入をする際の空気摩擦熱を保護遮断する方法）

初期索引語：

Initial Query:

ATMOSPHERIC ENTRY — SPACECRAFT SHIELDING

AERODYNAMIC HEATING — SPACECRAFT REENTRY

検索はタイトルによって行なわれる。初期索引語による検索は次のとおりである。

Due to the pair(With occurrence frequencies of 32 and 13 respectively):No responses.

Due to the second pair(with occurrence frequencies of 50 and 58 respectively):

1. "Theory of Stagnation Point Heat Transfer In a Partially Ionized Diatomic Gas"
2. "Future Problems in Reentry Physics"
3. "Effects of Nonequilibrium Flows on Aerodynamic Heating During Entry Into the Earth's Atmosphere from

Parabolic Orbits"

この結果は不満である。したがって次に検索ゲームを行なうことにする。

＊様式1開始の指令を出す。

<表示1> 候補組

```
DISPLAY I(Candidate Pairs):

          Term X                          Term Y
  ATMOSPHERIC ENTRY    —(0)—  SPACECAFT SHIELDING
  AERODYNAMIC HEATING —(0)—  SPACECRAFT REENTRY
```

In computer storage, operation on the first candidate pair:

```
          Associated with Term X
PLANETARY REENTRY          (6)
REENTRY EFFECTS            (5)
SPACECRAFT REENTRY         (9)
TRAJECTORY                 (6)
          Associated with Term Y
RADIATION SHIELDING        (6)
```

<表示2> 共通語

```
DISPLAY II(Common Terms):
    No entries.
```

＊様式2開始の指令を出す。

<表示3> 組合せ

```
DISPLAY III(Associated Pairs):
    No entries.
```

＊様式3開始の指令を出す。

<表示4> 単 語

```
DISPLAY IV(Single Terms):
      Associated with    Term X or Term Y
PLANETARY REENTRY          (6)
REENTRY EFFECT             (5)
SPACECRAFT REENTRY         (9)
TRAJECTORY                 (6)
RADIATION SHIELDING                (6)
```

質問者は最初の2つの語を保存リストに入れることに決定。SPACECRAFT REENTRYは候補組に入っているので，自動的に保存される。残りの2つの語と候補組の最初の組は，有望とは思われないので消去する。ここで，質問者は，今までの3様式を第2番目の候補組について繰返すことにする。

*様式1開始の指令を出す。

<表示1> 候補組

```
DISPLAY I(Candidate Pairs):

          Term X                          Term Y

   AERODYNAMIC HEATING —(0)— SPACECRAFT REENTRY

In computer storage:

          Associated with Term X

          HEAT TRANSFER          (5)

          THERMAL                (5)

          THERMAL PROTECTION     (5)

          Associated with Term Y

         *ATMOSPHERIC ENTRY      (9)

          REENTRY                (5)

          REENTRY CONDITION      (5)

          REENTRY EFFECT         (8)

          REENTRY TRAJECTORY     (9)

          REENTRY VEHICLE        (9)

         *TRAJECTORY             (9)
```

*印は消去語

<表示3> 共通語

```
DISPLAY II(Common Terms):

   No entries.
```

*様式2開始の指令を出す。

<表示3> 組合せ

```
DISPLAY III(Associated Pairs):

          Associated with                 Associated with
              Term X                          Term Y

   (5) THERMAL PROTECTION —(5)— REENTRY VEHICLE (9)

   (5) THERMAL PROTECTION —(7)— VEHICLE         (9)
```

質問者は，このうち最初の組を候補組に入れることにし，2番目のものを消去することに決定。ただしＶＥＨＩＣＬＥだけは保存することにした。

＊様式3開始の指令を出す。

DISPLAY Ⅳ(Single Terms):

Associated with	Term X	or Term Y
HEAT TRANSFER	(5)	
THERMAL	(5)	
REENTRY		(5)
REENTRY CONDITION		(5)
REENTRY TRAJECTORY		(9)

質問者はこれらの語をすべて消去することにした。これらの状況から判断して質問者は2番目の候補組をとったのは失敗であったと感じる。しかしＡＥＲＯＤＹＮＡＭＩＣ ＨＥＡＴＩＮＧはまだ保存しておく価値ありと判断しそうする。この回の表示3の所で入れた候補組がまだ残っているので，次にそれについてゲームを繰返えすことにする。

＊再び様式1開始の指令を出す。

<表示1> 候補組

DISPLAY I(Candidate Pairs):

Term X		Term Y
THERMAL PROTECTION	—(5)—	REENTRY VEHICLE

In computer storage:

Associated with Term X	
AERODYNAMIC HEATING	(5)
REENTRY VEHICLE	(5)
*THERMAL	(9)
VEHICLE	(7)
Associated with Term Y	
LIFTING REENTRY	(9)
PARAGLIDER	(5)
*REENTRY	(6)
*SPACECRAFT REENTRY	(9)
THERMAL PROTECTION	(5)
VEHICLE	(9)

<表示2> 共通語

DISPLAY Ⅱ(Common Terms):

Associated with	Term X and Term Y
VEHICLE	(7) (9)

この語は消去に決定。

＊様式2開始の指令を出す。

<表示3> 組合せ

DISPLAY Ⅲ(Associated Pairs):

Associated with Term X		Associated with Term Y
(5) AERODYNAMIC HEATING	—(5)—	THERMAL PROTECTION

この語の組合せを候補組に入れることに決定。

＊様式3開始の指令を出す。

<表示4> 単 語

DISPLAY Ⅳ(Single Terms):

Associated with	Term X or Term Y
LIFTING REENTRY	(9)
PARAGLIDER	(5)

これらの両語も消去決定。次に質問者は保存リストを表示させ<表示5>吟味した結果，これ以上保存しても用がないと判断する。これまでのゲームの結果残った2組の候補組を検索決定組とすることにする。

<表示6> 決定組

DISPLAY Ⅳ(Accepted Pairs):

THERMAL PROTECTION —(5)— REENTRY VEHICLE

AERODYNAMIC HEATING —(5)— THERMAL PROTECTION

すなわちこれが最終の索引語となる。

これらの索引語を使い，文献タイトルを対象に検索した結果を次に示す。

Due to the first pair (with occurrence frequencies of 37 and 67 respectively):

1. "Thermally Integrated Structure—A Review of Low-Intensity Cooling Systems For Thermally Protected Airframes"
2. "Thermal Protection of Lifting Vehicles"
3. "Pyrolytic Materials For Thermal Protection Systems"

4. "Glide-Vehiele Thermal Protection Performance"
5. "Design Considerations For A Reentry Vehicle Thermal Protection System"
6. "The Charring Ablater Concept, Application to Lifting Orbital or Suborbital Entry"

Due to the second pair (with occurrence frequencies of 50 and 37 respectively):

1. "Aerodynamically Heated Structures"
2. "The Outlook For Ablation Heat Protection Systems"
3. "Thermal Protection With A Temperature Capability to 2,500 Degrees F, For Cool Structures
4. "Structures For Manned Entry Vehicles"
5. "Charring Ablaters in Lifting Reentry"

これらの初期検索および最終検索に使用した検索論理は（索引語A，索引語B）V（索引語C 索引語D）である。

1.2.4 句形式索引語

1.2.4.1 はしがき

句形式索引語とは，文献主題の記述に際しキーワードの単純な羅列の代りに"てにおは"の入った句を用いる形式をいう。

たとえば，

Formaldehyde, polymerization of, by ultraviolet light in dispersions,

Polymerization of Formaldehyde, by ultraviolet light in dispersions,

などとする形式であって，その典型はChemical Abstracts誌の主題索引に見られる。

"てにおは"のついた形式が，キーワードの単純な羅列に優さるのは自明である。第1にキーワード間の関係が明示され意味がはっきりする。たとえば，次のものを比較してみればよい。

Caesium, adsorption, Hg, electrodes, methylformamide

Adsorption of Caesium by Hg electrodes in the presence of methylformamide

第2に説明的であって，余分な記号上の約束なしで理解できる。

このような句形式索引語は，人が文献を読んでつける。できた句形式索引語をどのように配列し

て索引を作るかという時に，いくつかの変化型が考えられる。まず最初に考えられるのがＫＷＩＣ形式である。たとえば図1.2.20のようになる。つぎにChemical Abstracts 誌が採用しているような転置型がある。前例と同じものを転置型とすれば図1.2.21のようになる。

fertilizer effect on cesium absorption by plants
soil colloids and cesium absorption by plants
ca and cesium absorption by roots
sphatase response to cesium: adenosine tripho
ystem in relation to cesium adsorbed on pt electrodes: +s
methy+ adsorption of cesium by hg electrodes in presence of
clay: adsorption of cesium from radioactive waste water by
+t on adsorption of cesium from na solution by clinoptiolite

図1.2.20　句形式索引語のＫＷＩＣ配列

Cesium.
absorption by plants, fertilizer effect on, 60:13833f
by plants, soil colloids and, 60:11321h
by roots, Ca and, 60:12620b
adenosine triphosphatase response to, 60:4400b
adsorbed on Pt electrodes, hydroquinone—quinone system in relation to, 60:3733g
adsorption of, by Hg electrodes, in presence of methylformamide, 60:8668c
from radioactive water by clay, 60:3865e
from Na soln. by clinoptilolite, heat—treatment effect on, 60:15482h
from water by brown coal, clay and clinoptilolite, 60:13009c
agaroid gel properties in presence of, 60:6246e
argon elec. plasma contg., spectroscopic temp. detn. of, validity of, 60:10077a
atomic scattering factor of, 60:7528d, 8655h
from barium—133 decay, r—r angular correlation in, 60:60:12835h
base exchange of, in alcs. or aq. alcs., 60:2359b
with NH_3 on faujasite—type zeolites, 60:7490h
on $(NH_4)_3PMo_{12}O_{40}$, 60:11405h
on Bio—Rex 70 and Dowex—50W, hydration in relation to, 60:42e
with Ca and Li, solvents in relation to, 60:7493c

図1.2.21　句形式索引語の転置型配列

どちらの形式が人にとって読み易いかというと，それは転置型の方であろう。ＫＷＩＣ型の欠点は，同じ単語（たとえばCesium）を繰返し印刷するので見た目に冗長であることと，スペースの制限から物理的に句の逆転が起りひじょうにつながりが理解し難いことである。転置型の長所はその裏返えしであって，重複される語が省略され，句の区切りは意味にもとづいていて逆転もその単位で行なわれる。

編集上におけるＫＷＩＣ型と転置型との差は，前者が完全に機械的に作られるのに反し，後者は人手によって作られるという点にある。

そこでこのいくつかの長所を持つ転置型の配列を機械的に考えられないかということが問題になる。これが可能になれば人はちょうど文献にキーワードをふるようにキーワード句をふるだけでよい。後は機械が句を適当に区切って逆転を行ない，同じ語を省略して図1.2.2 1のような形に印刷してくれる。このような機械編集の試みはJ.E.ArmitageとM.F.Lynchによって報告されている。[16)17)] 以下彼らの所論に従ってその概要を説明する。

1.2.4.2　句形式索引の生成

まず例で示そう。つぎのような句があったとする。

((Duraboline protection against(metabolism of (calcium) by bones) after administration of cortisone or its analogues) in bone disorders)

＊カッコは後の説明の便宜上つけたもので，ここの例として本質的なものではない。

これを目標としては，たとえばCalclum を最初に持ってきて次のように変形することにある。

Calcium
metabolism of, by bones, after administration of cortisone and its analogs, durabolin protection against, in bone disorders, 60：16173b

この区切の最小単位である要素句は，名詞（句）とその前あるいは後に前置詞，接続詞など機能詞をつけた形で構成されている。たとえばby bonesとかmetabolism of などである。この場合by boncs after などという区切り方は許されていない。

そこで変形の方法もそれにもとづいて考えればよい。最初の例を参照されたい。Calcium を最初に持ってきて変形する場合は，まずCalcium をかっこでくくる。ついでその前後の句をさきの要素句の法則にもとづいて区切っていく。それを概念図で表わせばつぎのようになる。

（（D（B（A）C）E）F）

これを配列するには，

A：B，C，D，E，F　　あるいは

A：B，C，E，D，F

などとすればよい。

この場合Aに先行する要素句B，Dは名詞＋機能句の形をなしており，後続の要素句C，E，Fは機能句＋名詞の形をなしているから，先行群あるいは後続群それぞれの中で相対位置を狂わせさえしなかったら，置く順序はどのように置いてもよい。したがって，

A：C，E，F，B，D

A：B，D，C，E，F

A：C，E，B，D，F

などが生じる。

これはかなり自由度がある方法であって，人の場合はよいが機械に対してはかえって障害になる。そこで可能は選択の中からひとつを決定する方法として，つぎのような考え方をとる。先の例を使うと，Calcium という見出しのもとには，その語を含むいろいろな成句が集まる。それらのうちでもっとも共通している要素句を最初におく。第2番目には，2番目に頻度の高いものをおく等々である。その際，先行群，後続群の間で順序の逆転が起ってはならないという先の制約は守らなければならない。

これは索引配列の際に先行のものと重複する語は省略するという事実にもとづき，できるだけ省略が多いようにする方法である。すなわち単独の成句の中では一意的に順序が決まらないから，配列の相対位置を決定要因中に含めようと考え方といえる。

12.4.3 句配列型の数

先に述べたように句の配列にはいくつかの変型が考えられる。それはどのくらいあるかがArmitage らによって考察されている。その2，3の場合をあげればつぎのようになる。

要素句の数	成　句	可　能　な　句　型　式	数
1	A	A	1
2	AB	A：B　B：A	2
3	ABC	A：BC　B：AC　C：AB B：CA　C：BA	5
4	ABCD	A：BCD　B：ACD　C：ABD　D：ABC CAD　BAD　BCA CDA　BDA　CAB DAB　CBA DBA	13

実際に数えて数列を作ることはつぎのようになる。

要素句の数	可能な句型式の数
1	1
2	2
3	5
4	13
5	34
6	89
⋮	⋮

これはFibonacci級数であって次の式によって定義される。

$$a_n = \frac{1}{\sqrt{5}}\left\{\left(\frac{\sqrt{5}+1}{2}\right)^{2n-1} + \left(\frac{\sqrt{5}-1}{2}\right)^{2n-1}\right\}$$

1.2.4.4　アルゴリズム適用上の問題点

上記の考え方に従って処理アルゴリズムを作った場合，いろいろの困難な点が出てくる。それはまたアルゴリズムを拡張して解決を計らなければならない点であるといえる。以下それらの問題点を列挙する。

a　不要語

句の中には索引の対象となるキーワードばかりでなく，ノン・キーワードも含まれている。たとえば，"effect" というような語であるとか"in relation to" などという句がある。配列の際にこれらが見出しとして選択されることがないようにしなければならない。

b　複合名詞

いくつかの名詞，形容詞が複合してひとつのまとまった意味を表わしているようなものがある。たとえば"nuclear reactor fuel element"とか"fat-high, gluten low diet" などである。これらは語が分散しているので，どちらかといえば見出しとして適当ではない。これは"by cracking hydrocarbon oil"とか"haemoglobin reconstituted from" など，動名詞，分詞を伴った句についても同様である。見出しとならないような処理をしなければならない。

c　形容詞－名詞

形容詞－名詞とつながったものでどちらか一方の語を見出しとする場合は，先のアルゴリズムが適用できない。たとえば次例がそうである。

```
Cesium
  absorption by plants,
Hydrocarbons
  aromatic,
```

すなわち前置詞，接続詞などの機能詞がないのに区切る場合がある。

d　andで結ばれた句

"dying and bleaching of linen" という句があるとする。人手で句形式索引を作る場合はこれを，

Bleaching

of linen　　と

Dying

of linen

の2つの見出しに分けて記入する。計算機の場合これをどのような形に処理するか，一工夫を要するところである。

e　不定詞

不定詞のtoが前置詞toと誤認されると，to後で区切られて意味の通じないものができ上がってしまう。

f　構文の変更

標準的な例では，できあがった句形式は順序のみを変えるだけで，後は変更のないのがふつうである。ところが特殊な例では，利用者の理解を助けるため，機能句の変更を行なう場合がある。たとえばつぎのようなものがそうである。

Aluminosilicates

catalysts, for hydrocarbon conversion,

Catalysts

for hydrocarbon conversion, aluminosilicates as,

Hydrocarbons

conversion of, aluminosilicate catalysts, for

Dielectric saturation

ion electrostatic forces and interactions in relation to,

Ions

electrostatic forces between, in solvents, dielec. satn.

in relation to,

Solvents

ion electrostatic interactions in, dielec. satn, in

relation to,

このような例は構文分析法で処理するとしてもかなり複雑なものにならざるを得ないであろう。

g 略 語

句中では簡便のために略語を使用することがあるが，それが見出しとなった場合は完全綴に戻す必要がある。

h 語の置換

句の意味表現のつごうおよび見出語との関連で句中の単語を置換える場合がある。それは多かれ少なかれ同意語的なものであるが，機械で処理する場合はまず不可能な部類に属するものである。たとえばつぎのようなものがある。

Blood sugar
in diabetes,

Diabetes
glucose metabolism in,

Diabetes
kidney infection from Escherichia coli in alloxan

Kidney
diseases or disorders of, Escherichia coli-induced, in diabetes,

Escherichia coli
Kidney disorder from, in diabetes,

1.2.4.5 プログラム言語

句形式索引を生成するプログラムはＳＬＩＰ言語で書かれている。ＳＬＩＰとはSymmetric List Processorの略であって，J.Weizenbaum[18]によって作成されたものである。そのコンパイラーはAtlasと呼ばれD.B.Russell[19]によって作られたものである。

ＳＬＩＰはリスト処理言語で，後方にリンクがつくのみならず，前方にもつく特殊な形式を持っている。したがって，ある語から始めて前の語あるいは後の語を（法則にしたがって）辿る必要のあるこのような場合には，最適のプログラミング言語であるといえる。

1.2.4.6 応 用

句形式索引は直接にはChemical Abstractsなど雑誌論文の索引に使われることも意図したものであるが，このように機械的処理の可能性が出てきた現在ではその他にもいろいろな応用が考えられるであろう。

たとえば，Kimber[20]はこれを図書館の蔵書索引に使用できると報告しているし，また，Freeman[21]はこれを分類表たとえばUniversal Decimal Classificationの索引に使うべきだと提言している。

1.2.5. 情報検索システムの形式化

1.2.5.1 はしがき

最近の動きとして，情報検索システムを形式化し，その基礎を評価し直そうとする試みが見られる。一般総合的なものとしてはD.Soergelの論文[22] 化合物検索についてはUniv. of PennsylvaniaのG.Weaver[23]の報告などが見られる。その他W.Uhlmann[24]の報告も，その本筋はIRシステムの定量的側面を述べたものであるが，形式化にふれている。

形式的な体系とは，あらかじめ決められた公理と推論を使って，それから直接誘導される定理群の総称である。実際上あらゆる問題についてそれが真か否か判断できるような定理を導出しておくことはできないので，その各場所に対応して推論を行ない，証明できればよいとされる。形式的体系は体系内で閉じていて，推論の過程でそれまで認められていない新らしい公理，定理などを導入することはない。このような形式的体系を作ることを形式化という。

情報検索システムは，これまで幾多の技法を生み出した。すなわち図書館目録，ユニターム方式，分類法，計算機処理方式，等々枚挙にいとまがない。しかしこれらはいわば問題本位に案出された技法であって，情報検索システムの本質とどういう関係をするのか，未だ解明されていない。これら一見異なる現存の技法あるいはシステムからその共通的性格を抽出できれば，それが情報検索システムの基本的構造ということになる。そしてそれは現存のシステムを比較評価する上の基準になるであろうし，またより大規模，包括的なIRシステムを設計する際の基盤ともなる。

情報検索システムの形式化というのは，上述のような意義なりねらいがあるということができる。

形式主義の立場からいえば，文献あるいは質問からその特徴を抜き出し手がかり語を構成する操作と，検索を行なう操作とは，基本的に異なった操作であると考えられる。手がかり語の構成は，ソースとなる文献の言語集合から，手がかり語の集合への変換であるが，検索は手がかり語の集合間でお互いに連絡をつけ関係を確認する操作である。

そこで形式化のスケジュールをたてれば，まず手がかり語変換の形式化，ついで検索操作の形式化となるであろう。しかし現実には必ずしもその全部が完成されているとはいえない。むしろまだ緒についたばかりであるといえよう。

1.2.5.2 記号系の数学的モデル

情報検索では，対象文献の情報を能率良く蓄積・検索するため，二次資料化を行なうことが多い。これは言語記述を手がかりにして，処理目的のために必要かつ十分に詳しくその内容を把握し，その結果を適当な言語で表現する操作である。たとえば：

抄録は $A : N \xrightarrow{N} N$　抄録操作Aによる同一の自然言語間における圧縮変換

索引は $I : N \xrightarrow{I} C$　索引操作Iによる自然言語Nから検索言語Cへの変換

と考えられる。[25]

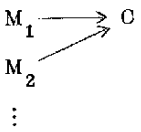
索引において一度検索語Cへ変換されると，その後はファイリングも検索もすべてCのみを頼りにして行なわれる。もし，この変換においてあいまいさが入りこんだりすると，それは後々まで影響し，ついには検索精度の劣化となって現われる。したがってこのようなあいまいさのない言語系とはどのようなものか，いかなる変換を行なえばあいまいさが入りこまないか，などを吟味しておくことは重要なことである。

記号系はそれ自身独立であって，組立方いかんによっては，記号系不備の故に検索不可能になることもある。記号系としてそれが完備であるためには最低どの程度の性質を備えていなければならないかを吟味しておくことは，形式化の一部分としても，特に記号系が複雑に変化する計算機による情報検索システムの設計上もおろそかにできないことである。以下Weaver の所論[23]を参考にして進める。

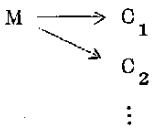
1.2.5.3 一語性および一義性

ある事物とそれを表現する記号の間には，一語性（uniqueness）と一義性（unambiguity）という2つの重要な概念が成立つ。

一語性とはある事物があればそれに対してひとつの記号が決まる場合をいう。これを図示すればつぎのとおりである。ある事物に対し一記号であるから，異なった事物が同じ記号に対応してもよい。



一語的

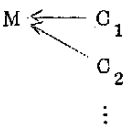


多語的

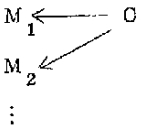
M：事　物

C：記　号

一義性とはある記号があれば，それに対してひとつの事物が決まる場合をいう。ある記号に対し一事物であるから，異なった記号が同じ事物に対応してもよい。これを図示すれば次のようになる。



一義的



多義的

一語性と一義性は，それぞれ記号化（coding）と解記号（decoding）に対応する。

この2つの概念を組合わせると図1.2.2.2の4つの場合ができる。一語一義的な関係は自明である。一語多義的なものは自然語に例をとれば同義語であり，多語一義的なものは同語異義語である。

一語一義的な場合が事物＝記号の関係ではもっとも望ましいものである。その他の場合はそれぞ

れ難点があり，多語多義的な場合では記号として使用することは不可能である。これらの関係を厳密に定義すればどういうことになるかを，以下に示す。

	一語的	多語的
一義的	$M \longleftrightarrow C$	$M \rightleftarrows \begin{matrix} C_1 \\ C_2 \end{matrix}$ ⋮
多義的	$\begin{matrix} M_1 \\ M_2 \end{matrix} \rightleftarrows C$ ⋮	$\begin{matrix} M_1 \rightleftarrows C_1 \\ M_2 \rightleftarrows C_2 \end{matrix}$ ⋮ ⋮

図 1.2.2 2　一語性と一義性の組合せ表現

1.2.5.4　記　号　法

記号系の要素は次の 4つである。

ⅰ）　記号化される対象物の集合：G

これはたとえば，文献内容の表現体(注)とか，化合物構造式が相当する。

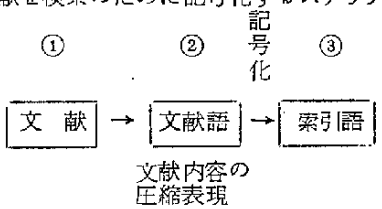
ⅱ）　対象物を記号化する有限個の規則の組：E

ⅲ）　記号化された表現の集合：S

たとえば，分類記号とか結合行列など。

ⅳ）　S の要素を解読してもとの文献内容の表現体とか化合物構造式などに直すための有限個の規則の組：D

(注)　文献を検索のために記号化するステップとして，次の順序を考える。



①と②の間は意味的内容，あるいは背景的知識が制御する変換であり，②と③の間はここでいう意味の記号系が制御する変換である。

ⅱ）の記号化規則は，さらに次の 5つの細則に分れる。

ⅱ－ⅰ）　部分構造を識別する規則の組

たとえば，意味担体である語とそれらの間の関係を表わす関係詞との区別とか，化合物部分構造の識別などがある。

ⅱ－ⅱ）　その部分構造を記号化する規則の組

ii－iii）　記号化された種々の部分を組立てる規則の組

ii－iv）　ある構造を処理するのにどの規則を適用するかの判断基準を示す規則の組

ii－v）　i～vの規則の適用順序を決める規則の組

以上の規則群中にない規則があってもよい。

記号法にはその重要な要素として，事物を記号に直す一種の関数が必要である。これを写像という。この写像は前述の記号を使っていい直せば，定義域をG，値域（すなわち規則ii）をSとする写像であって，それが機械処理可能なるためには帰納的でなければならない。

記号法を形式的に定義すれば，次のとおりである。

定義　1：情報検索における記号法Kは7個組で表現される。

$$K=\langle G,\ E,\ D,\ C,\ S,\ \{Ge\}_{e\in E},\ \{Sd\}_{d\in D}\rangle$$

i）　$G\subseteq \underline{G}$　　$G\neq\phi$

ii）　E：写像eの空でない有限の集合族

$$e:Ge\rightarrow S \qquad Ge\subseteq \underline{G},\ Ge\neq\phi$$

iii）　S：記号化された表現の集合

iv）　$d:Sd\rightarrow \underline{G}$　　$Sd\subseteq S,\ Sd\neq\phi$

v）　C：eによって記号化された表現の集合

$$C=\bigcup_{e\in E} e\,(Ge)\subseteq S$$

定義　2：$K=\langle G,\ E,\ D,\ C,\ S,\ \{Ge\}_{e\in E},\ \{Sd\}_{d\in D}\rangle$ は次の各項が成立つときに限って強い記号法という。

i）　Kはひとつの記号法である。

ii）　E中の各eが帰納的で，かつGeがGが帰納的に相関（recursive relative）。

iii）　D中の各dが帰納的で，かつSdがSに帰納的に相関。

v）　SがS⊆Tであるとき，そのある集合Tに帰納的に相関。

定義補1：fをf：A→Bなる関数とする。$\forall\alpha\in A$について終結に導く算法が存在する時，fを計算可能あるいは帰納的であるという。

定義補2：A⊆Tとする。Tのある与えられた元がAに属するか否か，有限回の手続によって決定できるような終結算法が存在する時，AはTに対し帰納的に相関であるという。

1.2.5.5　完備（Completeness）

記号法において，各事物がそれぞれ対応する記号あるいは記号群を持つときに限りその記号法は完備であるという。また，各事物が高々一つの記号をもつとき一語的，各記号が高々一つの事物を表わすとき一義的であるという。

定義　3：記号法$K=\langle G,\ E,\ D,\ C,\ S,\ \{Ge\}_{e\in E},\ \{Sd\}_{d\in D}\rangle$は，$G=\bigcup_{e\in E} Ge$でGが$\underline{G}$に帰納的に相関であるときに限って，Gに相対的に完備であるという。

定義　4：記号法Kは，KがGに相対的に完備であり，かつG＝$\underline{G}$であるときに限って絶対的に完備であるという。

KがGに相対的に完備である場合は，$\bigcup_{e \in E} Ge$は帰能的である。$\bigcup_{e \in E} Ge$の帰納性は記号法が完備になる必要十分条件である。完備でない記号法は使用不能である。

1.2.5.6　一語性（Uniqueness）

定義　5：記号法Kは，各e∈E の関数のときに限って弱く一語的であるという。

定義　6：記号法Kは，次の場合が成立つときに限って一語的であるという。

ⅰ）　Kは弱く一語的

ⅱ）　ある事物gが2つの異なった関数e，e′の定義域にある場合はe(g)＝e′(g)

定義　7：　記号法Kは次の場合が成立つときに限って強く一語的であるという。

ⅰ）　Kは一語的

ⅱ）　各記号関数eが一対一対応

定義　8：　記号法Kは次の場合が成立つときに限って中間的に一語的であるという。

ⅰ）　Kは弱く一語的

ⅱ）　各記号化関数eが一対一対応

記号化規則が関数であれば，ある事物に対応して記号が（高々一つとはいえないが）決まるから，弱い意味で一語的であるといえる。つぎに一語的といわれる場合は（もし使えるなら），いかなる記号化規則を使っても対応する記号がひとつに定まる記号である。強くあるいは中間的に一語的な場合は，使う記号化規則によって対応する記号がそれぞれ異なる。

上記のことから次の定理が導き出される。

定理1：Kをひとつの記号法とする。写像e，c′の定義域Ge，Ge′が記号化写像のすべての組について互に素ならば，次のことが成立つ。

ⅰ）　Kが一語的であれば弱く一語的。

ⅱ）　Kが強く一語的であれば中間的に一語的。

すなわち，Ge∩Ge′＝φでe(g)≠e′(g)となるから，すべての場合に一レベルずつ落ちることになる。

1.2.5.7　固　有

一義性は解記号規則に関連したものである。したがって一義性は解記号写像の状態と記号化写像＝解記号写像の状態の双方に関係する。そこでまず解記号写像の状態に関して"固有"という概念を定義する。

定義　9：記号法Kは，次の場合に限り弱く狭義固有であるという。

ⅰ）　解記号写像が存在し

ⅱ）　それぞれの解記号写像dをc上に縮小した場合，dはやはりひとつの関数である。

すなわち，$Sd' = Sd \cap C$ で，かつ $\alpha \in Sd \cap C$ であるようなすべての α につき $d'(\alpha) = d(\alpha)$ なる新写像 d' が存在して，それが関数である，ということである。

定義１０：記号法Ｋは，次の場合に限り狭義固有であるという。

ⅰ） Ｋは弱く狭義固有

ⅱ） ある記号が縮小された２つの解記号写像の定義域に属するとき，これらの写像はその記号について同じことを行なう。

定義１１： 記号法Ｋは，次の場合に限り強く狭義固有であるという。

ⅰ） Ｋは狭義固有

ⅱ） 各解記号写像がｃ上に縮小されるときは一対一対応。

定義１２：記号法Ｋは，次の場合に限り，中間的に狭義固有であるという。

ⅰ） Ｋが狭義固有

ⅱ） 各解記号写像がｃ上に縮小されるときは一対一対応。

定義１３：記号法Ｋは，次の場合に限り弱く固有であるという。

ⅰ） 解記号写像が存在し，

ⅱ） 各解記号写像はそれぞれひとつの関数。

定義１４：記号法Ｋは，次の場合に限り固有であるという。

ⅰ） Ｋは弱く固有

ⅱ） すべての解記号写像ｄ，ｄ′においてＳのひとつのもとたとえばＳｏが，ｄおよびｄ′の定義域に属するとき。

$$d'(So) = d(s)$$

定義１５：記号法Ｋは，次の場合に限り強く固有であるという。

ⅰ） Ｋは固有。

ⅱ） 各解記号写像は一対一対応。

定義１６：記号法Ｋは，次の場合に限り中間的に固有であるという。

ⅰ） Ｋは弱く固有。

ⅱ） 各解記号写像は一対一対応。

定理１と同様に次の定理が成立つ。

定理２：記号法Ｋがすべてのｄ，ｄ′において解記号写像 $Sd \cap Sd' = \phi$ になったとする。その場合次のことが成立つ。

ⅰ） Ｋが（狭義）固有ならば，弱く（狭義）固有。

ⅱ） Ｋが強く（狭義）固有ならば，中間的に（狭義）固有。

狭義固有は解記号写像が常にＣの上に制限されているのに比べ，固有ではＳ全域に拡張されそのような制限のない違いがある。

一般にCとSdとの関係は次の3つの場合がある。

ⅰ) $C = \bigcup_{d \in D} Sd$

ⅱ) $C \subseteq \bigcup_{d \in D} Sd$

ⅲ) $C \supseteq \bigcup_{d \in D} Sd$

狭義固有と固有が区別されるのはⅱ)の場合のみであってⅰ)とⅲ)の場合では差が生じない。

定義17：記号法Kは $\bigcup_{d \in E} Sd \subseteq C$ であるときに限り正規であるという。

すなわち，記号化された記号の集合とSの部分集合で解記号される記号の集合が等しい場合を正規というのである。

定義18：記号法Kは $\bigcup_{d \in D} Sd \subseteq C$ の場合に限り準正規であるという。

さきの説明とこの定義18から次の定理が成立つ。

定理3：Kが準正規になれば，次のことが成立つ。

ⅰ) Kが弱く固有ならば，Kは弱く狭義固有。

ⅱ) Kが固有ならば，Kは強く狭義固有。

ⅲ) Kが強く固有ならば，Kは強く狭義固有。

1.2.5.8 構　成

次に一義性定義の2つの前提のうち2番目の記号化写像＝解記号写像の関係を定義する。

定義19：記号法Kは次の場合に限り良い構成（fine）であるという。

ⅰ) 各eおよび $g \in Ge$ なるgにつき

$$d(e(g)) = g$$

なるdが存在する。

ⅱ) 各dおよび $\alpha \in C \cap Sd$ なる α につき，もしd（α）＝gならばそのときはe(g)＝αなるeが存在する。

これからすぐに次のことがいえる。

定理4：Kが良い構成の記号法であるならば，$C \subseteq \bigcup_{e \in E} Sd$ かつ

$$\bigcup_{e \in E} Ge = G = \bigcup_{d \in D} d(Sd \cap C)$$

である。

定義20：記号法Kは次の場合に限り特に良い構成（ultra fine）であるという。

ⅰ) Kは良い構成

ⅱ) すべてのdにつきe(e)∈Sdならば

$$d(e(g)) = g$$

1.2.5.9 一 義 性

以上一義性を定義するための道具立てができた。

定義21：記号法Kは次の場合に限り，弱く一義的である。

ⅰ) Kは弱く固有

ⅱ） Kは良い構成

定義２２：記号法Kは次の場合に限り一義的であるという。

ⅰ） Kは固有

ⅱ） Kは良い構成

定義２３：記号法Kは次の場合に限り中間的に一義的であるという。

ⅰ） Kは中間的に固有

ⅱ） Kは良い構成

定義２４：記号法Kは次の場合に限り強く一義的であるという。

ⅰ） Kは強く固有

ⅱ） Kは良い構成

定理　５：Kが一対一対応記号化特性を持ちかつ良い構成であるときは，Kは $\bigcup_{d\in D} Sd \subseteq O$ の場合において（すなわちKが正規の場合）一対一対応解記号特性を持つ。

1.2.5.10　本質的多義性

本質的に多義的というのは，ある記号法がそれと同じ記号化規則をもち，かつ弱く一義的であるような他の記号法に拡張できないときをいう。

定義２５：記号法Kは次の場合に限り本質的に多義的であるという。

すべてのK′において。

ⅰ） Kの記号化写像がK′に含まれているか，あるいは，Kの記号化写像のある縮小がK′に含まれている。

ⅱ） K′は $\bigcup_{e\in E} Ge$ に対し相対的に完備。

ⅲ） K′は良い構成でも固有でもない。

定理　６：Kはひとつの記号法であって，無限に多くの事物が記号αの上に写像されるようなものである場合は，Kは本質的に多義的であるという。

定理　７：Kはひとつの記号法であって，記号化写像e，e′に対するe(Ge)，e′(Ge′)が互に素でなく，g≠g′であるg，g′に対してもe(g)＝e′(g′)である場合は，

ⅰ） Kが良い構成であるならばKは（狭義）一義的ではない。

ⅱ） Kが良い構成であるならばKは強く（狭義）一義的ではない。

1.2.5.11　結　論

ここで上述の定義の結果を総合すると，いかなることがいえるか，そしてそれが実際上の問題解決にいかに役立つか，を吟味することにする。

まず，弱く一語的な記号系と一語的な記号系の関係および中間的に一語的な記号系と強く一語的な記号系との関係をみてみる。以下において記号法はすべて強いものとする。

系１：$K=\langle G, E, D, C, S, \{Ge\}_{e\in E}, \{Sd\}_{d\in D}\rangle$ は弱く一語的であるとする。この場合次のようなK′が存在する。

ⅰ）$K'=\langle G, E', D, C', S, \{Ge\}_{d\in D}, \{Sd\}_{d\in E}\rangle$

ii） K′ は一語的

iii） $\bigcup_{e\in E} Ge = \bigcup_{e\in E'} Ge = G$

iv） Kが中間的に（狭義）一義的ならばK′ も同じ。

v） Kが強く（狭義）一義的ならばK′ も同じ。

vi） Kが弱く（狭義）一義的ならばK′ も同じ。

vii） $C' \subseteq C$

（証明略）

系2：$K = \langle G, E, D, C, S, \{Ge\}_{e\in E'}, \{Sd\}_{e\in D}\rangle$ は中間的に一語的で $G = \bigcup_{e\in E} Ge$ であるとする。この場合次のようなK′ が存在する。

i）$K' = \langle G, E', D, C, S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D}\rangle$。

ii）Kは強く一語的。

iii） $\bigcup_{e\in E'} Ge = \bigcup_{e\in E} Ge$。

iv）Kが中間的に（狭義）一義的ならば，K′ も同様。

v）Kが強く（狭義）一義的ならば，K′ も同様。

vi）Kが弱く（狭義）一義的ならば，K′ も同様。

vii）Kは（狭義）一義的ならば，K′ も同様。

系3：$K = \langle G, E, D, C, S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D}\rangle$ で，

i）KがGに対し相対的に完備

ii）Kが特に良い構成。

であるならば，

$K' = \langle G, E, D', C, S, \{Ge\}_{e\in E}, \{Sd\}_{d\in D}\rangle$ で次のようなK′ が存在する。

i）K′ はGに対し相対的に完備。

ii）K′ は特に構成。

iii）Kが弱く（狭義）一義的ならば，K′ は（狭義）一義的。

iv）Kが中間的に（狭義）一義的ならば，K′ は強く（狭義）一義的。

v）Kが一語的ならば，K′ も一語的。

vi）Kが中間的に一語的ならば，K′ も中間的に一語的。

vii）Kが弱く一語的ならば，K′ も弱く一語的。

viii）Kが強く一語的ならば，K′ も強く一語的。

系4：$K = \langle G, E, D, C, S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D}\rangle$ で

i）KはGに対し相対的に完備。

ii）Kは弱く（狭義）一義的。

iii）Kは一対一記号化特性を持つ。

であるならば，

次のような$K'=<G, E, D', D, S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D}>$が存在する。

ⅰ）K'はGに対し相関的に完備。

ⅱ）K'は一対一記号化特性を持つ。

ⅲ）K'は一対一解記号特性を持つ。

系5：$K=<G, E, D, C, S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D}>$で

ⅰ）KはGに対し相関的に完備。

ⅱ）Kは中間的に（狭義）一義的かあるいは弱く一義的。

ⅲ）Kは中間的に一語的。

であるならば，

次のような$K'=<G', E', D, C', S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D'}>$が存在する。

ⅰ）K'はGに対し相関的に完備。

ⅱ）K'は一対一解記号特性を持つ。

ⅲ）K'は一対一記号化特性を持つ。

系6：$K=<G, E, D, C, S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D}>$において

ⅰ）KはGに対し相対的に完備。

ⅱ）Kは中間的にあるいは強く一義的ならば，

次のようなK'が存在する。

ⅰ）$K'=<G, E', D, C, S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D}>$

ⅱ）K'は中間的あるいは強く一義的。

ⅲ）K'はGに対し相対的に完備。

ⅳ）K'は強く一語的

次に中間的および強く一語的な系について述べる。これはそのうちのどちらが中間的あるいは強く一義的な系に拡張しうるかを決めるものである。

系7：$K=<G, E, D, C, S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D}>$において

ⅰ）KはGに対し相対的に完備。

ⅱ）Kは（中間的に）強く一語的。

であるならば，

次のような$K'=<G, E, D', C, S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D}>$が存在する。

ⅰ）K'はGに対し相対的に完備。

ⅱ）K'は（中間的に）強く一語的。

ⅲ）K'は中間的に一義的。

系8：$K=<G, E, D, C, S, \{Ge\}_{e\in E'}, \{Sd\}_{d\in D}>$において，

ⅰ）KはGに対し完備相対。

ⅱ）Kは一対一記号化特性を持つ。

ならば

次のようなK′＝＜G，E，D′，C，S，$\{Ge\}_{e\in E'}$ $\{Sd\}_{d\in D'}$＞ が存在する。

ⅰ）K′はG′に対し相対的に完備。

ⅱ）K′は一対一記号化特性を持つ。

ⅲ）K′は一対一解記号特性を持つ。

最後の問題として一語的あるいは弱く一語的な記号系は，果して本質的に多義的かどうか吟味しておく必要がある。系4から一語的な系だけを考えればよい。

系9：K＝＜G，E，D，C，S $\{Ge\}_{e\in E'}$ $\{Sd\}_{d\in D}$＞ において

ⅰ）KはGに対し相対的に完備。

ⅱ）Kは一語的。

ⅲ）それぞれの記号化関数eにつき集合Geは次の条件を備える。

Geは有限な数の部分集合を持ち，次の形をしている。そしてその各項は有限である。

$Ge(1)=\{g_1, \cdots\cdots, gn : e(gi)=e(gj)\quad 1\leq i, j\geq n\}$ であるならば，

次のようなK′が存在する。

ⅰ）K′＝＜G，E，D′，C，S，$\{Ge\}_{e\in E'}$，$\{Sd\}_{d\in D'}$＞

ⅱ）K′＝Gに対し相対的に完備

ⅲ）K′は一語的

ⅳ）K′は中間的に一義的

1.2.5.12　評　価

ここでは上記の結論から，記号法の評価を行なう。現在の記号法を分類すると次の3つになる。

a　分類記号

b　トポロジー法[26]

c　キーワード法

これら記号法の主要用途は，a，ファイルの特性，b．類概念検索，c．種概念，関連概念の検索，の3つになる。

キーワード法においては，記号化規則が一対一記号化特性を備えていれば，その系は強く一語的であると共に強く一義的である。

トポロジー法および分類記号では，記号法に一語的あるいは強く一語的な性格を求めるのは酷であって，少なくとも弱く一語的な線で置くべきではないかと思われる。トポロジー記号は，一対一解記号写像があり，その写像が一対一解記号特性を備えかつ $C\subseteq \bigcup_{d\in D} Sd$ であれば，一語一義的な記号法とすることができる。

1.3　ファイルの構成と探索

1.1で述べたように，狭義の情報検索とは「ファイルに貯えてある情報を探し出すこと」であり，決して情報を作り出すことを意味していない。すなわち貯えてあるいくつかの情報から新たに情報を作り出すことは行なわない。

一般に情報検索では，種々の情報を整理し，ある基準に従っていくつかの情報をまとめてレコードを作り，このレコードを集積して情報のファイルを作成する。情報検索では，質問（問合せ）から，このファイルの探索指令を作り，これに従ってファイルのレコードを1つ1つ調べ，探索指令で指定したアイテムとマッチしたアイテムを持つレコードの他のアイテムを回答として取りだすことを行なう。

電子計算機を利用した情報検索で，いかに大量の情報をランダムアクセスメモリに蓄積していても，探索指令とレコードのアイテムの照合は主記憶装置上で行なわれる。

そこでまず主記憶装置上のデータの蓄積，探索について述べ，次に情報のファイル化について述べ，次にファイルの探索について述べることにする。

1.3.1　電子計算機の主記憶装置へのデータの蓄積と探索

まず，いくつかのデータを電子計算機の主記憶装置（コアメモリ）に格納しておいて，必要に応じてあるデータを探しだす場合について考えてみる。格納すべきデータにはそれぞれ異った大きさの記憶場所が必要であるが，データを整理し格納する場合に次の2通りが考えられる。

① 格納するデータがすべて同じ大きさの記憶場所を必要とするとき，またはそれぞれのデータが異った大きさの記憶場所を必要とするが，その最大のものに合わせてすべて同じ大きさの記憶場所にそれぞれのデータを格納する場合。

② 格納するデータがそれぞれ異った大きさの記憶場所を必要とし，かつそのような大きさの場所を確保してデータを格納する場合。

①の場合は格納するすべてのデータが同じ大きさの記憶場所をとり，②の場合は記憶場所の大きさは可変であるが，このような記憶の仕方に対してデータの探し方が異ってくる。

主記憶装置に貯えてあるデータを1つ1つ調べながら必要なデータを探し出すときに，①の形式で格納されている場合には各データの記憶場所の大きさは一定であるから，サブスクリプト（添字）を用いるという，いわゆるテーブルサーチの方法で探索することができる。この①の形式をテーブル形式とも呼ぶ。

これに対して②の形式でデータが格納されているときは，単なるテーブルサーチの方法ではデータ

の探索はできない。次に示すような3種のいずれかのデータをそれぞれの格納データに付加しておかなければならない。

(i)　1つのデータと他のデータの間にデータの区切りを示すもの。

(ii)　そのデータの専有する記憶場所の大きさ（桁数，語数といったもの）を示すもの。

(iii)　そのデータの次に調べるデータが格納されている場所（格納されている先頭番地）を示すもの。

特に(iii)のデータをアドレスポインタ（単にポインタともいう）と呼んでいる。

一般に②の形式でデータを格納する場合には，データの探索の方から見て，(i)，(ii)，(iii)のいずれかまたはこれらを併用したデータを追加しなければならない。

この②の形式でのデータの処理は，①のテーブル形式のデータの処理に比してプログラムが面倒になる。というのは，ほとんどすべての計算機（ディジタル型）はハードの機能として，メモリを逐次一定間隔で探索するのに便利な機能を持っている。このインデックスを扱う命令を利用することによってテーブル形式のサブスクリプトの処理が容易にできる。

上記の(iii)のデータ（アドレスポインタ）は他の目的にも利用することができる。(i)，(ii)はあくまで一連のデータを逐次調べる時にしか有効でないが，(iii)のアドレスポインタは，次に調べるべきデータのアドレス（番地）を示しているものであるから，格納してある全体のデータを調べる必要がなく，ある種のデータのみを調べればよい場合には，それらのデータをこのアドレスポインタで結んでおけばよい。

以上の蓄積の方法と探索の方法の関係を図にまとめたものが図1.3.1である。

図1.3.1から判るように，データの蓄積の方法と探索の方法は密接な関係にあり，それぞれを切りはなして考えることは無意味である。そこで次にいくつかの事例を考えてみることにする。

〔事例1〕

蓄積すべきデータはすべて同じ大きさの記憶場所を必要とし，かつ格納してあるデータをすべて調べなければならない時。

蓄積方法　：　①の形式

探索方法　：　テーブルサーチ

〔事例2〕

各蓄積データが必要とする記憶場所は本質的には可変であるが，ハード的に，テーブルサーチの方法で探索することが望ましい場合。

蓄積方法　：　最大のデータの大きさに統一して固定にとる①の形式

探索方法　：　テーブルサーチ

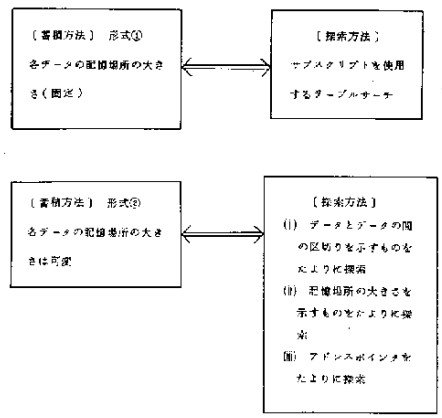


図1.3.1　データの蓄積方法と探索方法の関係

〔事例3〕

各蓄積データが必要とする記憶場所は可変であり，なるべくメモリを有効に使用し，特にハード的には制約を受けない。またすべてのデータを調べなければならない。

蓄積方法 ： ②の形式（①の形式にするときもある）

探索方法 ： ②の形式の探し方

このような場合，記憶場所を各データについて固定にしたとき起るメモリの無駄と，探索時間の縮少を考慮することによって，形式①にするか，形式②にするかを決定する。

〔事例4〕

数種類のデータが混在し，探索の際必ずしも全データを調べる必要がない時。

蓄積方法 ： 同種のデータについてアドレスポインタで結ぶ。または種類別にソートしその先頭番地を示すポインタを別に用意する。

探索方法 ： ポインタを繰りながら探索する。

これは探索するとき調べるデータをなるべく必要最少限にとどめることが主目的であるため，個々のデータが必要とする記憶場所の大きさが固定であっても，ポインタを利用する方が有利となる。

以上述べたこの4つの事例はあくまで簡単な例について検討したにすぎない。実際問題としては，次の点を考え合せなければならない。

(1) 個々の蓄積データの記憶場所の大きさが固定長であるか可変長であるか。

可変長であるとしたら，最大に合わせて固定にしたときどのくらいメモリの無駄が生じるか。

(2) これらのデータ群をサーチするとき，全データについて探索しなければならないか。

もし探索の時と場合によって全データを調べる必要がなければ，物理的にグループ分けして格納するか，ポインタで同種のデータを結ぶか。

などについて検討し，最終的に

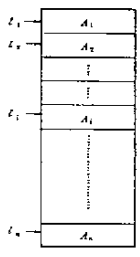
(i) 無駄なメモリの最小化

(ii) 探索時間の最小化

を考えなければならない。

次に蓄積形式のメモリマップと探索方法のフローチャートについてまとめてみる。

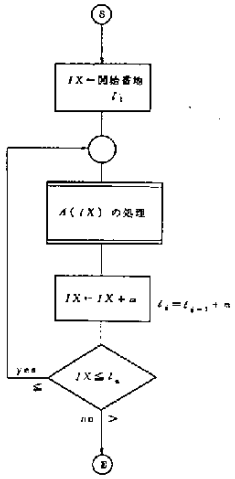
1） 形式①の場合



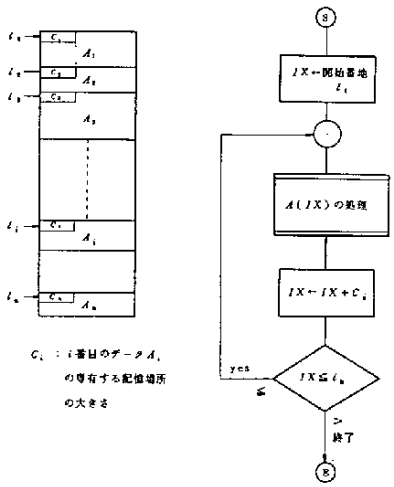
各データの記憶場所の大きさ m

ℓ_i ：i番目のデータ A_i の先頭番地

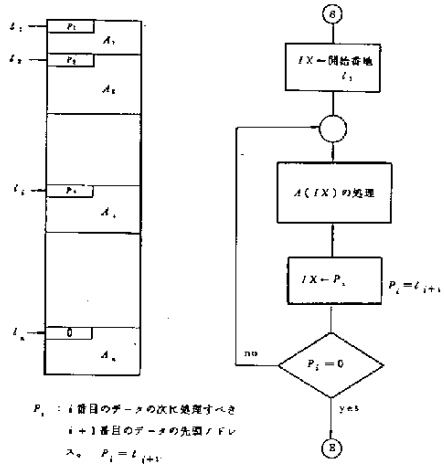
ℓ_1 ：開始番地



2） 形式 ②—(ii)



3） 形式 ②—(iii)



1.3.2 個体と属性，属性の値

ファイルの構成，探索の説明に，「個体」，「属性」，「属性の値」といった用語を使用するので，ここで定義しておく。

「何々に関する情報」と言ったときの「何々」に相当するものを個体と呼ぶことにする。

たとえば，

「1つの文献に関する情報」とか

「1人の研究員に関する情報」とか，または

「1つの化合物に関する情報」といった

文献，研究員，化合物を個体と呼ぶ。

個体に関する情報にはいくつかの種類がある。この種類に相当するものを属性と呼ぶことにする。

たとえば，

文献に関する情報として，著者名，主題，発行所といったものがあるが，これらの著者名，主題，発行所といったものを属性と呼ぶ。

研究員に関する情報で言えば，氏名，年令，性別，研究テーマとかいったものを属性と呼ぶ。

各個体についてのそれぞれの属性の内容を属性値と呼ぶことにする。すなわちある個体の無形である情報を，自然言語や記号で表現したものが属性値ということになる。

1.3.3 蓄積情報

ある個体の集合を考えそれに関する情報をすべて集め，行を個体に，列を属性に対応させた2次元の表に整理したとする。

属性 / 個体	a (1)	b (2)		i (i)	.....
A (1)	I_{11}	I_{12}	.....	I_{1i}	.....
B (2)	I_{21}	I_{22}	.....	I_{2i}	.....
C (3)	I_{31}	I_{32}	.....	I_{3i}	.....
⋮					

図 1.3.2

I_{ij} ： i 番目の個体の j 番目の属性の値

たとえば個体の集合として，ある研究所の研究員の集合をとれば上記の表は次のようになる。

属性 / 個体	氏　名	研究員№	年令	性別	研究分野	専門分野
研究員 1	佐　藤	223	25	男	計算機ソフトウエア	物　理
研究員 2	斉　藤	010	40	女	ドキュメンテーション	図書館学
研究員 3	山　田	115	32	男	数値解析	数　学
⋮						

上記の例では，どの個体のどの属性についてもその属性値は唯一つであるが，属性の選び方によっては1つの属性が多値になることがある。たとえば，個体に文献をとり，その属性に「その文献の著者名」とるとき，多人数で文献を著わしたときは，この属性値は多値になる。

たとえば，

	.....	著者名	主　題	
文献 A	.....	佐藤，斉藤，高橋	計算機のソフト，ドキュメンテーション	.....
文献 B	.....	山本，鈴木	オペレーティングシステム	.....
文献 C	.....	.....	.....	.....

このように属性の選び方，すなわち，情報の整理の仕方によっては１つの属性の値（内容）が複数個になる。

上記の属性の１つの値をファイルの構成上，アイテムと呼ぶ。アイテムをある基準に従って集めたものをレコードと呼ぶ。このレコードを集めたものがファイルということになる。

電子計算機の記憶装置の内部では，情報を次のいずれかで表現する。

(i) 文字列 { 自然言語の語
　　文字，数字，特殊記号を用いたコード

(ii) 数　値

これは主記憶装置（磁気コア）でも，補助記憶装置（磁気テープ，磁気ドラム，磁気ディスク，磁気カード）上でも，１つの電子計算機システム内では同じ表現をとるのが普通である。

情報を自然言語の語（欧字，漢字，カナ文字）で表現したときは，一般にこの文字列の長さは可変になる。この文字列を記憶装置に格納するわけであるから，自然言語の語で表現したアイテムの記憶場所の大きさは可変になる。

1.3.4　ファイルの探索

狭義の意味の情報検索のファイルの探索という作業は図 1.3.3 に示すごとく，質問（問合せ）から作成した探索指令に従って，まずファイルを選び，次にそのファイルの全部のレコードを１つ１つ，または質問に関連するレコードのみを１つ１つ，指定のアイテムについてその内容を調べ，該当するレコードの指定された他のアイテムを取り出すことである。

簡単に述べると，ファイルの探索の中心となる作業はアイテムの照合である。ファイルの中のレコードのある特定のアイテムをどのような順番で照合するかは，探索指令の作り方とファイルの構成とに直接関係してくる問題である。

探索指令の内容を一般的に述べると，

(i) どのファイルを探索するかを示すもの。

(ii) 各レコードのどのアイテムを照合するかを示すもの。

(iii) そのアイテムの内容。

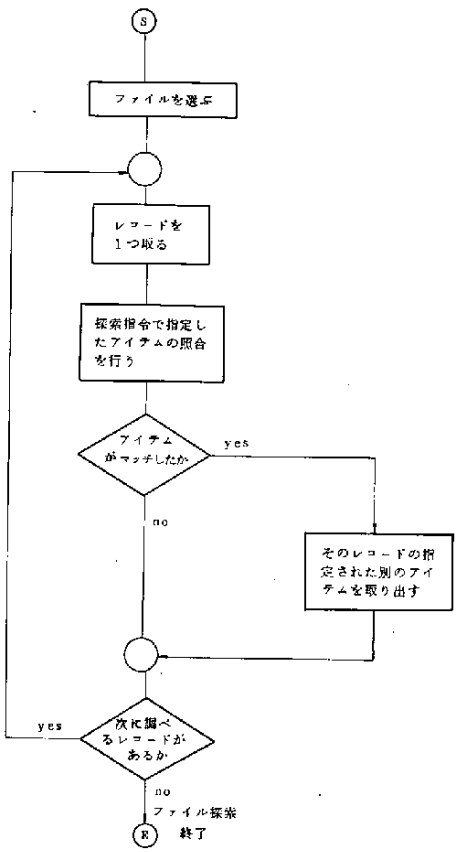


図 1.3.3　ファイル探索のフロー

(iv) 照合の対象となったアイテムがマッチした時，回答として取り出す他のアイテムを指定するもの。

などがある。

情報検索システムを設計する時に考慮しなければならない点は，

(i) 1つの質問についての検索時間をなるべく少なくすること。

(ii) 検索の誤差を出来るだけ少なくすること。

であるが，(ii)の問題は情報の収集，表現に関するものでファイルの探索には直接関係はない。ファイルの探索法を考えるときに一番重要なことは，探索時間をできるだけ少なくすることである。ファイルの探索の中心となる作業は前述の如く，アイテムの照合という作業であるが，探索時間を少なくするために次の点を考える必要がある。

問題点1：

ファイルのアイテムの照合回数をできるだけ少なくすること。どの質問についても蓄積情報のすべてを照合の対象にするのでなく，その質問と全く無関係であることが初めから判っているレコードは照合の対象にしないこと。

その対策：

(1) 質問の内容によっては照合の対象となるアイテムが片寄るはずであるから，照合の対象となるアイテムと回答として取り出すだけのアイテムのみで1レコードを構成するような，ある種の質問専用のファイルを作成しておく。

(2) アイテムの内容についてレコードをソートして記憶装置内に並べておく。そして各グループの先頭のレコードの格納場所を示すチェイン（ポインタで結んだもの）を示すデータを付加しておく。

(3) あるアイテムについて同じ系統の内容を持つレコードをチェインで結ぶ。

問題点2：

レコード内の指定されたアイテムの位置が簡単に求められること。

その対策：

レコード内のアイテムの位置を直接示すデータ（相対番地，アイテムの大きさを示すもの）を用意する。

問題点3：

アイテムの照合の手間を出来るだけ簡単にし，1回の照合時間を少なくする。

その対策：

情報のファイルは一般に，磁気テープ，磁気ディスク，磁気カードなどに作られているが，アイテムの照合は主記憶装置（磁気コア）内で行なわれるのが普通である。

主記憶を語単位で区切るワードマシンでは，ハードで1語単位で照合を行う。アイテムの照合はこ

のハードの照合を何回か繰り返えして行なうわけであるので，もしワードマシンを利用し，アイテムの内容が情報を文字列で表現しているものであれば，1語に格納される文字数の整数倍に合わせてアイテムの大きさを定めておいた方が照合の手間が簡単になる。

1.3.5 ファイルの構成

1.3.5.1 マスタファイルとトランザクションファイル

まず，ある個体の集合に関する情報を集め，これを整理し情報のファイルを作るとする。

1つの方法は，1.3.3で述べたように，第1.3.2図でいえば1行に並ぶ情報，すなわち1つの個体に関する情報をまとめて1レコードとしてファイルを構成する。この場合，情報を自然言語で表現したアイテムは大きさが可変となり，また個体の属性によっては値が多値になることがありレコードの長さが個体によってまちまちになる。

たとえば，文献の情報に関してファイルを構成する場合，文献の著者名，標題といったものは普通自然言語で表わされる。また著者名とか主題という属性は多値になる。このような時1個体に関する情報をすべて1レコードにまとめると，レコードの長さは個体によって可変になり，このようなファイルの処理（ファイルの探索，レコードのソーティング）は複雑になり，処理時間も増大する。

そこでいくつかの属性をまとめて1つのレコードを構成し，1つの個体の情報を複数個のレコードに分けてファイルを構成する方法が考えられる。1.3.5.2ではこの実例を示す。

1.3.4で述べたように，探索の対象となるファイルは，探索の能率を上げるために，ある種の質問専用のファイルをいくつか用意することが必要である。

ここで便宜上，前者のファイルをマスタファイルと呼び，後者のファイルをトランザクションファイルと呼ぶことにする。

マスタファイルもトランザクションファイルも共に，1.3.4で述べたように，各レコードは

(i) 蓄積情報

(ii) ファイル探索に必要なデータ

の2種から構成されている必要がある。

1.3.5.2 ファイルの構成の実例

ここで日本科学技術情報センター（JICST）の文献検索システムのファイルの構成を実例としてあげる。

マスタファイルには2種類ある。

(i) IR漢字ファイル

これは文献の情報を漢字コードで表現したもので，「文献速報」誌作成のために作られたマスタファイルに若干の修正を加えたものである。レコードの構成は図の通りであるが，1つの文献

IR漢字ファイル

〔注〕本レイアウトに使用する用語は，科学技術文献速報自動作成システムによる。

セグメントB（定期刊行物）

1w=36bit　　BCDモード　8w=48桁　　1字=18bit　　漢字モード　1字＝18bit，20w＝40字

セグメントNO	外注先コード	部門コード	情報員NO	原稿NO	エラーフラグ	補助原稿NO	分類コード1	分類コード2	年月日	抄録名	記事番号	記事区分	料金判定
B:01:0													
1 2	1	1	2	3	8	1 1 1 1	6	9	9	8字	5字	9字	2字	2字

書架NO	使用言語1	使用言語2	使用言語3	年
5字	3字	3字	3字	2字

BCDモード　　1字＝18bit　　漢字モード　1字＝18bit　20W＝40字

セグメントNO	外注先コード	部門コード	情報員NO	原稿NO	エラーフラグ	補助原稿NO	分類コード1	分類コード2	発行国	雑誌名（Variable）
B:02										
1 2	1	1	2	3	8	1 1 1 1	6	9	9	3字	Variable

BCDモード　　1字＝18bit　　漢字モード　1字＝18bit　20W＝40字

セグメントNO	外注先コード	部門コード	情報員NO	原稿NO	エラーフラグ	補助原稿NO	分類コード1	分類コード2	挿入図コード	写図	巻	号	巻，号（Variable Variable）
B													
1 2	1	1	2	3	8	1 1 1 1	6	9	9	4字	3字	3字	3字	Variable

セグメントB
（原子力レポート）

1w=36bit　　BCDモード　8w=48桁　　1字=18bit　　漢字モード　1字=18bit，20w=40字

セグメントNO				入出力区分	入出力コード	データ種別コード	部門コード	情報源NO	原稿NO	エラーコード	訂正区分	修正区分	補助区分	補助原稿NO	分類コード1	分類コード2	年月日	抄録者	記事番号	記事区分	料金判定
B	01	0																			
1	2	1		1	1	1	2	3	8	1	1	1	1	6	9	9	6字	5字	9字	2字	2字

書架NO	使用言語1	使用言語2	使用言語3	年
5字	3字	3字	3字	2字

1字=18bit

BCDモード　　漢字モード　1字=18bit　20W=40字

セグメントNO				入出力区分	入出力コード	データ種別コード	部門コード	情報源NO	原稿NO	エラーコード	訂正区分	修正区分	補助区分	補助原稿NO	分類コード1	分類コード2	挿入図コード	写図	表	参	発行国	整理NO
B	01	0																				RM
1	2	1		1	1	1	2	3	8	1	1	1	1	6	9	9	4字	3字	3字	3字	2字	variable

RM：レコードマーク。

整理NOはVariableであるが24字以上の場合はレコードが増える。

セグメントB
（会議資料）

1w=36 bit　　1字=18bit

BCDモード　8w=48桁　　漢字モード　1字=18bit，20w=40字

セグメントNO	入力区分	入力データコード	部門コード	情報員NO	原稿NO	エディションNO	原稿区分	補助原稿NO	分類コード1	分類コード2	年　月　日	著　者	記　事　番　号	記事区分	社会制定
B 01 0															
1 2 1	1	1	2	3	8	1	1	6	9	9	6字	5字	9字	2字	2字

番　組　NO	使用言語 1	使用言語 2	使用言語 3	年
5字	3字	3字	3字	2字

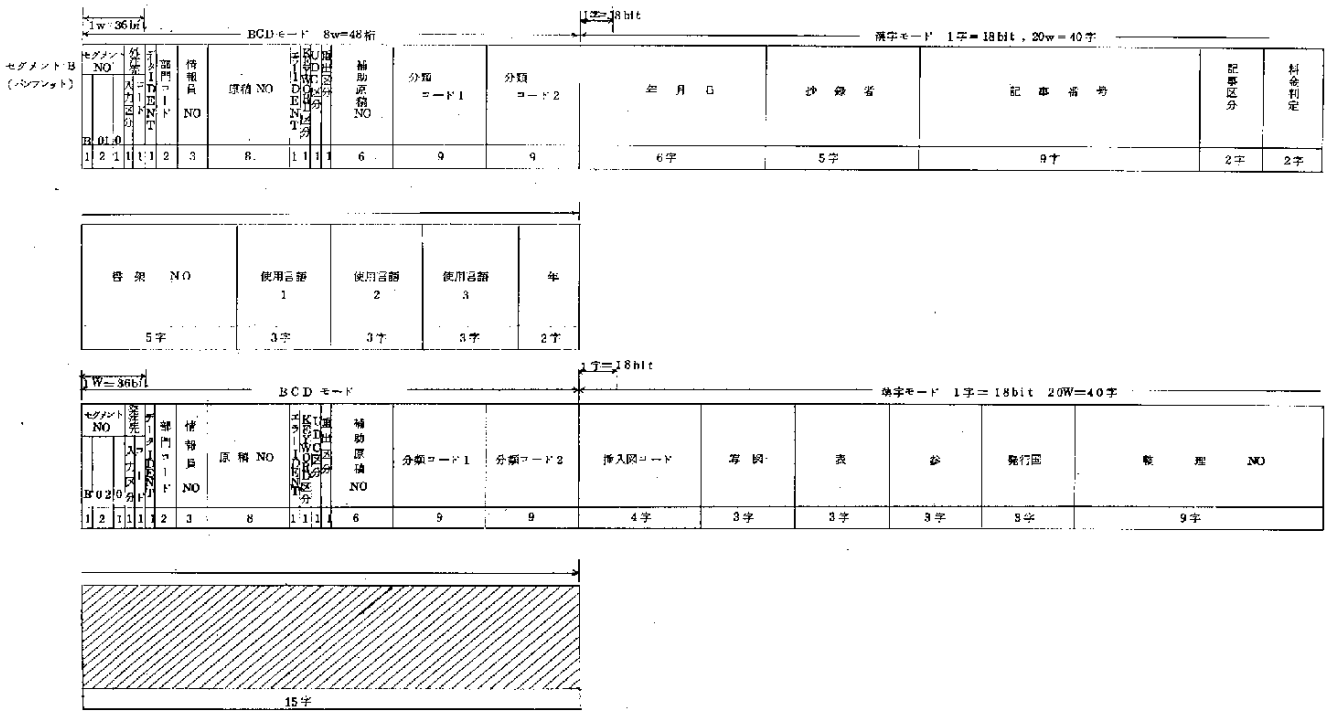
1字=18bit

BCD モード　　漢字モード　1字=18bit　20w=40字

セグメントNO	入力区分	入力データコード	部門コード	情報員NO	原稿NO	エディションNO	原稿区分	補助原稿NO	分類コード1	分類コード2	挿入図コード	写　図	表	参	発行国	整　理　NO
B 02 0																
1 2 1	1	1	2	3	8	1	1	6	9	9	4字	3字	3字	3字	3字	9字

難　読　名
Variable
Variable

※　難読名が15字以上の場合はレコードが増える。


1w=36bit
BCDモード 8w=48桁
1字=18bit
漢字モード 1字=18bit，20w=40字
セグメントB
(パンフォト)
セグメントNO
B'01.0
1 2 1 1 1 1 2
3
8
1 1 1 1 6
9
9
6字
5字
9字
2字
2字
巻 架 NO
5字
使用言語 1
3字
使用言語 2
3字
使用言語 3
3字
年
2字
1W=36bit
BCDモード
1字=18bit
漢字モード 1字=18bit 20W=40字
セグメントNO
B'02.0
1 2 1 1 1 1 2
3
8
1 1 1 1 6
9
9
4字
3字
3字
3字
8字
9字
15字

セグメントB
（特許）

1w=36bit　　BCDモード　8w=48桁　　1字=18bit　　漢字モード　1字＝18bit，20w＝40字

セグメントNO	分類コード	部門コード	情報員NO	原稿NO	ファイルDC区分	訂正DC区分	訂正区分	参照原稿NO	分類コード1	分類コード2	年月日	抄録者	記事番号	記事区分	料金判定
B 01 0															
1 2 1	1 1	2	3	8	1	1	1	6	9	9	6字	5字	9字	2字	2字

番組NO	使用箇所1	使用箇所2	使用箇所3	年
5字	3字	3字	3字	2字

1W=36bit　　BCDモード　　1字=18bit　　漢字モード　1字＝18bit　20W＝40字

セグメントNO	原稿入力区分	ファイルDC区分	部門コード	情報員NO	原稿NO	ファイルDC区分	訂正DC区分	訂正区分	補助原稿NO	分類コード1	分類コード2	挿入図コード	写図	表	参	発行国	特許NO
B 02 0																	
1 2	1	1	2	3	8	1	1	1	6	9	9	4字	3字	3字	3字	3字	7字

整理NO	
6字	11字

セグメント B
（PB, AD
レポート）

1w=36bit　　BCDモード　8w=48桁　　1字=18bit　　漢字モード　1字=18bit，20w=40字

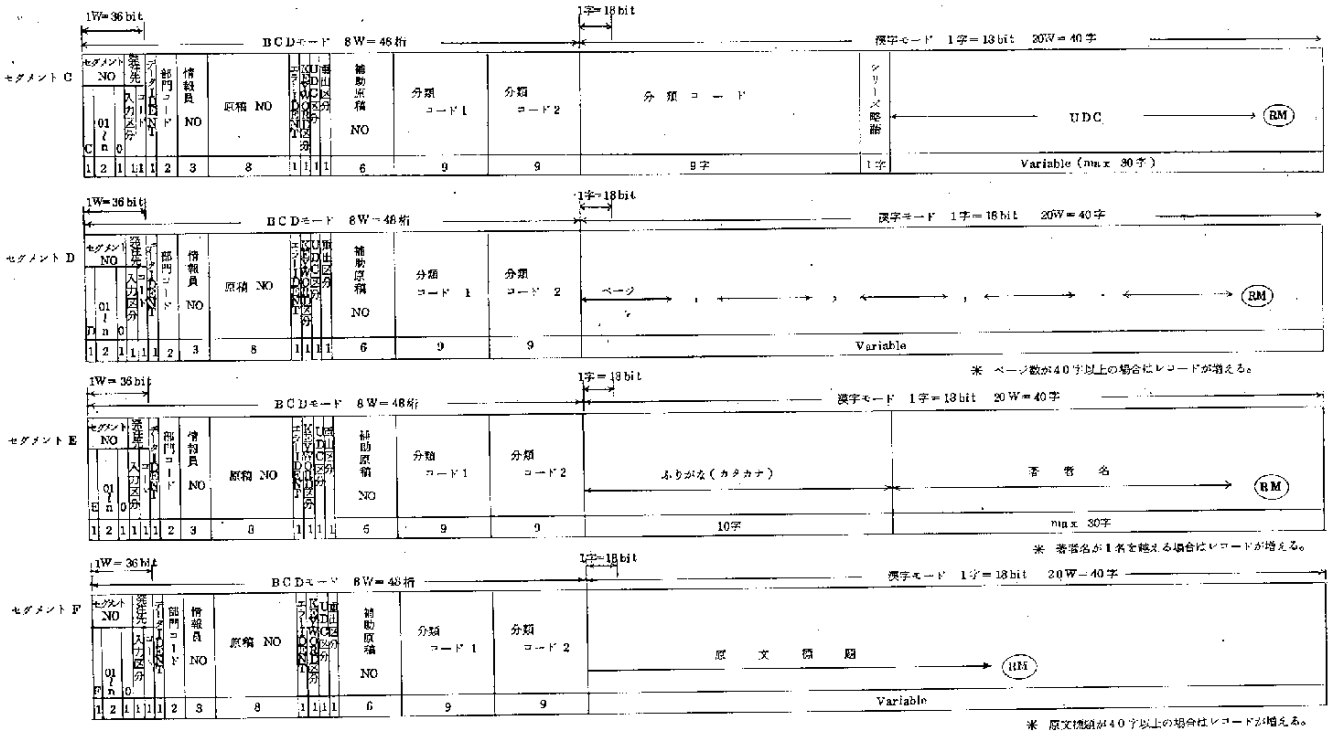
セグメントNO	分類区分	データIDENT	部門コード	情報員NO	原稿NO	訂正区分	訂正区分	訂正区分	補助原稿NO	分類コード1	分類コード2	年月日	抄録者	記事番号	記事区分	料金判定
B01	0															
1 2	1	1	2	3	8	1	1	1	6	9	9	6字	5字	9字	2字	2字

書架NO	使用頻度1	使用頻度2	使用頻度3	年
5字	3字	3字	3字	2字

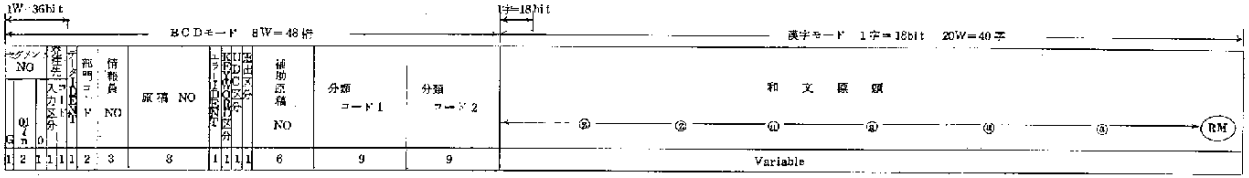
1W=36bit　　BCDモード　　1字=18bit　　漢字モード　1字=18bit　20W=40字

セグメントNO	分類区分	データIDENT	部門コード	情報員NO	原稿NO	訂正区分	訂正区分	訂正区分	補助原稿NO	分類コード1	分類コード2	挿入図コード	写図	表	鈔	発行国	整理NO
B02	0																
1 2	1	1	2	3	8	1	1	1	6	9	9	4字	3字	3字	3字	3字	9字

15字


セグメントC
1W=36bit
BCDモード 8W=48桁
1字=18bit
漢字モード 1字=18bit 20W=40字
セグメントNO
発注入力区分
データ区分コード
部門コード
情報員NO
原稿NO
補助原稿NO
分類コード1
分類コード2
分類コード
シリーズ略語
UDC
RM
C
01/n
0
1 2 1 1 1 1 2 3 8 1 1 1 6 9 9 9字 1字 Variable (max 30字)
セグメントD
ページ
Variable
※ ページ数が40字以上の場合はレコードが増える。
セグメントE
ふりがな(カタカナ)
著者名
10字
max 30字
※ 著者名が1名を越える場合はレコードが増える。
セグメントF
原文標題
HM
※ 原文標題が40字以上の場合はレコードが増える。

セグメントG

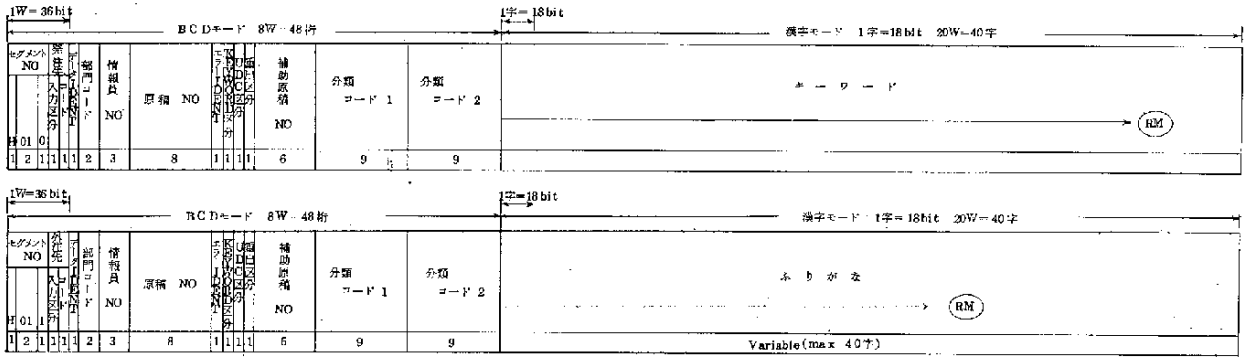


②：分離禁止記号

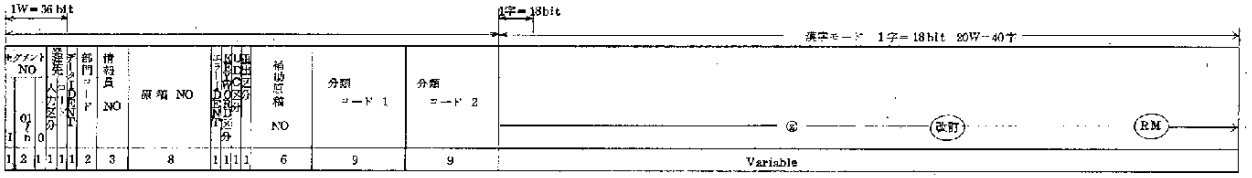
④：キーワード抽出記号

※ 40字以上の場合コードが増える。

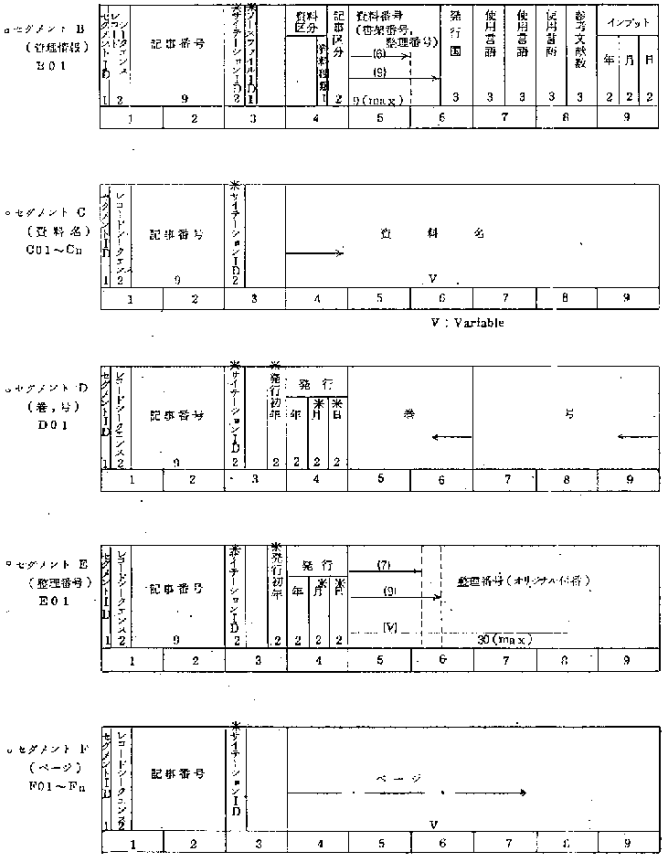
セグメントH



セグメントI



IR BCDファイル



V：Variable

(注) 1. ● 印で示したセグメントはＪＩＣＳＴでインプットされるもの。
※ 印で示した項目はＪＩＣＳＴではインプットされない項目。

2. ＪＩＣＳＴでインプットされる書誌データの内，資料番号，資料名，巻号，整理番号は資料種別（書架番号の４桁目に相当）によりその収録項目が異なる。

(a) 定期刊行物（資料種別：J）
資料番号（書架番号）（B01）　資料名（C01～Cn）　巻号（D01）

(b) 原子力レポート（N）
資料番号（ブランク）（B01）　整理番号（オリジナル付番：max 30 ch）（E01）

(c) 会議資料（K）
資料番号（整理番号：9ch）（B01）　資料名（C01～Cn）

(d) パンフレット（M）
資料番号（整理番号：9ch）（B01）

(e) 特　許（P）
資料番号（整理番号6ch）（B01）　整理番号（オリジナル付番：9ch）（E01）

(f) PB,ADレポート（R）
資料番号（ブランク）（S01）　整理番号（オリジナル付番：9ch）（E01）

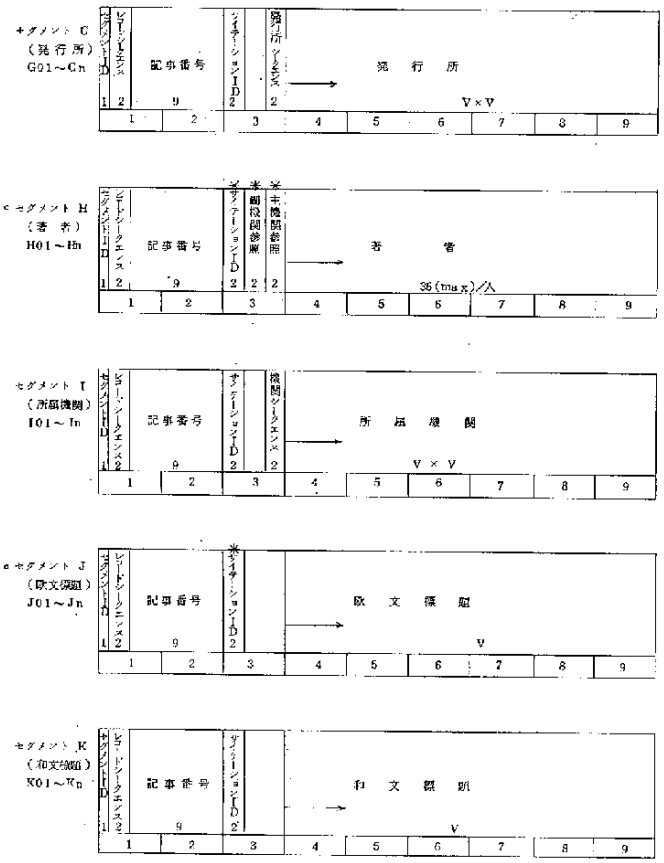
3. ファイルマージの扱いについて
幾種類かのファイルをマージする場合，その記事のデータ源を明示するためソースファイルコードを記録する。記事が増設した場合には核心となるファイル（例えばＪＩＣＳＴ－ＢＣＤファイル）をコアファイルとし，他システムからのファイル（補足ファイル）に関してはその記事番号をソースファイル参照セグメントに，キーワードを補足キーワードセグメントに収録する。

4. 引用文献数（参考文献数）の扱いについて
各記事について引用文献情報（セグメントC,D,E,F,H,）を同一記事番号でインプットする。（ファイル構成はＢＣＤファイルと同一）引用文献情報セグメントについては，サイテーションＩＤに引用文献番号に同一記事内でのシークエンスNOを付る。

5. サイテーションＩＤの付番方法

1	01	120	C0
〃	〃	〃	〃
99	99	129	C9
100	A0	130	D0
〃	〃	〃	〃
109	A9	139	D9
110	B0	140	E0
〃	〃	〃	〃
119	B9	149	E9

以下同様



6. 発行初年

発行年が2年以上にまたがる場合その初年を記入する。

7. 発行所シークエンス（機関シークエンス）

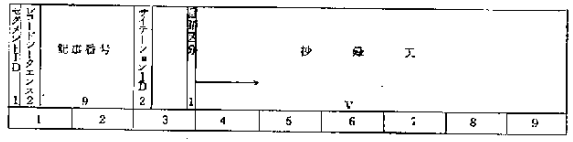
n レコード／1 機関 or 1 発行所
n 機関 or n 発行所／SEG.G or SEG.I
} である故に、

各機関 or 発行所のデータ区分を明確にするために各機関 or 発行所単位に通番する。

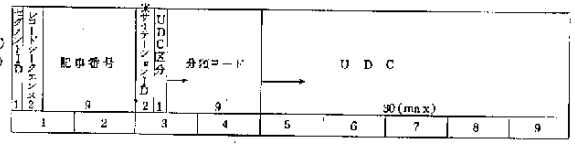
※ レコードシークエンスはレコード単位の通番である。

※ 機関参照はその著者が所属する機関シークエンスを記録する。

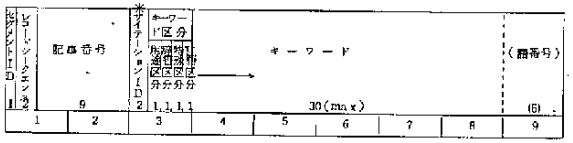
セグメント L
（抄録文）
L01～Ln



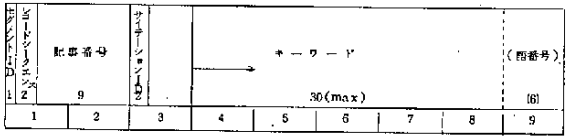
*セグメント M
（分類コード）
M01～M10



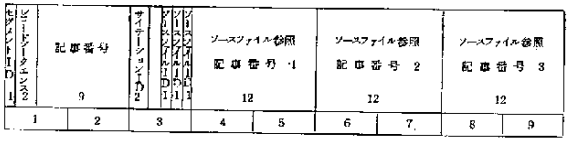
*セグメント N
（キーワード）
N01～N15



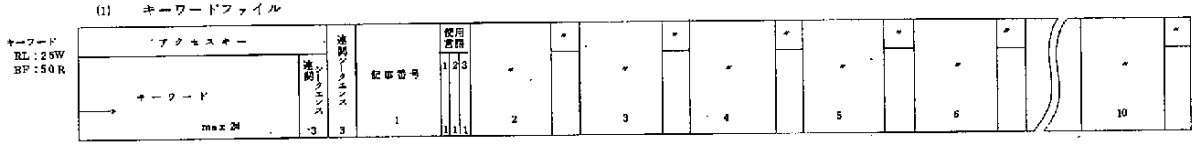
セグメント O
（補足キーワード）
O01～On



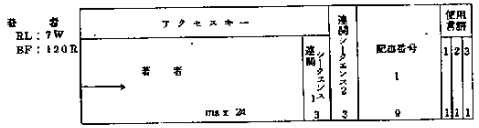
セグメント P
（ソースファイル参照）
P01～Pn



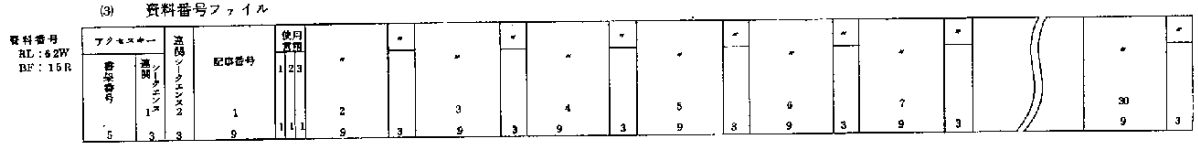
(1) キーワードファイル



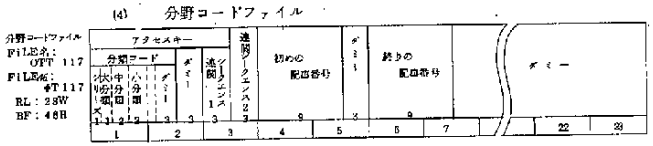
(2) 著者ファイル



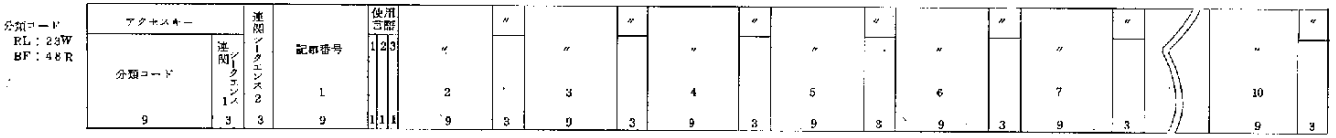
(3) 資料番号ファイル



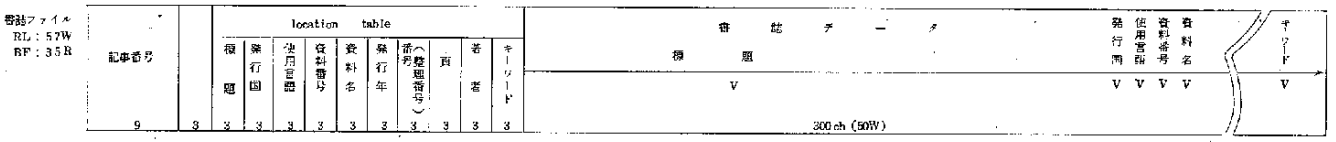
(4) 分野コードファイル



(5)　分類コードファイル



(6)　書誌ファイル



に関する情報をいくつかのレコードに分けてある。

1レコードは28語の固定長とし，各レコードの頭8語はレコードの種別を表わすデータを含んでいる。文献に関する情報は残りの20語に含まれている。

(ii) ＩＲ　ＢＣＤファイル

これはＩＲ漢字ファイルから，情報の表現をＢＣＤモード（英字，カナ文字表現）に変え，ＢＣＤモードの文献検索システム用に作り替えたマスタファイルである。各レコードは9語の固定長で，頭の2語はレコードを区別するためのデータが含まれ，残りの7語にＢＣＤ表現の情報が格納されている。

ＪＩＣＳＴのオンラインＩＲシステム（ＩＲＯＮ）ではこのマスタファイルから次に示す6つのトランザクションファイルを作成し，検索作業の際はこれらのファイルを磁気ディスクの中へ磁気テープよりロードして使用している。

質問の中でキーワードに関する部分はキーワードファイルを探索し，著者名に関する部分は著者ファイルを探索するといったようにする。資料番号ファイル，分野コードファイル，分類コードファイルも同様に扱う。これらのファイルを探索して得られた「記事番号」から書誌ファイルを探索して回答を出力する。さらに利用者の要求に応じてＩＲ漢字ファイルをも探索して回答を出力するようになっている。

1.3.6　ファイル構造の数学的理論

1.3.6.1　データベースの作成

ＣＯＤＡＳＹＬの言語構造グループが提案した情報代数学の理論を導入して，ファイル構造を取扱う。ただし，いわゆる階層構造（hierarchical structure）のみに限定する。

「個体」，「属性」，「属性の値」，「レコード」，「ファイル」を1.3.2の定義で与えたとして，ファイルは次のような一意写像として表現される。

$$\varphi : E \rightarrow A$$

ここにEは個体の集合，Aは属性値集合の直積で，

$$A = A_1 \times A_2 \times \cdots\cdots\cdots \times A_n$$

である。

A_jは，個体のj番目の属性がとりうる属性値を意味する。すなわち，

$$\varphi = (\varphi_1, \varphi_2, \cdots\cdots\cdots, \varphi_n)$$

$$\varphi_j : E \rightarrow A_j$$

ファイルのどのような構造もφの成分の一つとして表現できる。たとえば，個体のある列は，

$$\varphi_f : E \rightarrow A_f = E$$

で表現できる。

φ_f はある個体をその先行する個体（successor）に対応させる。このような対応は，ファイルの"構造"写像とよばれる。構造写像をいうとき，フィジカルには"アドレス"という個体の集合を意味するのが普通である。

ここで図1.3.4に示されるような階層構造を考え，同一の属性集合を持つような一つのレベル中のすべての個体の集合を，SIMSCRIPT言語でいう「集合」とよぶ。図1.3.4では，「集合」A，「集合」B，「集合」Cの三つがあることになる。「集合」A，「集合」Bはそれぞれ「集合」B，「集合」Cの"マスター"集合である。「集合」C，「集合」Bはそれぞれ「集合」A，「集合」Bの"スレーブ"集合である。情報代数学の原論文では，各属性集合に"未定義のΩ"なる値を導入しているが，それでは一般化しすぎるので，このように「集合」なる概念を導入したのである。

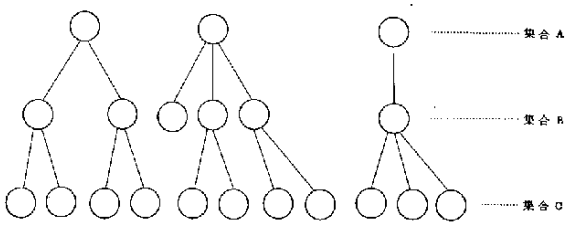


図1.3.4　階層データ構造

先述したように，各集合は個体の列を持つ。

$$\varphi_{fj} : E_j \rightarrow E_j$$

これはフィジカルなレコードシークエンスで表現できるが，大容量記憶装置上で乱更新を有効に行うためには，"次の個体のアドレス"を属性値とするがよい。

場合により，φ_f の逆写像

$$\varphi_{bj} = \varphi_{fj}^{-1} : E_j \rightarrow E_j$$

が，"直前の個体のアドレス"という属性として陽に表現される方がよい。

E_1 をマスター集合，E_2 をスレーブ集合と仮定すれば，階層構造はもう一つの"構造写像"の存在を主張する。

$$\varphi_h : E_1 \rightarrow 2^{E_2}$$

ここに φ_h は一意写像ではない。一意でない写像を表現するための一つの方法は，φ_h を二つの一意構造写像で表現することである。

$$\varphi_d : E_1 \rightarrow E_2$$

$$\varphi_f' : E_2 \rightarrow E_2$$

ここに φ_d はマスター個体に対する長男の個体を指示し，φ_f' はすぐ次の弟を指示する。まず φ_d を適用し，ついで φ_f' を数回反復することによって，どのスレーブ個体にも到達できる。注意すべきことは，長幼関係は用途に応じ選択される適当な列に関するもので，必ずしも φ_f の列には一致しない。

もし必要ならば，次の逆写像

$$\varphi_b' = \varphi_f'^{-1} : E_2 \to E_2$$

は，ある属性によって陽に表現できる。この方法は，もしあるスレーブ個体が2個のマスター個体を持つような可能性があるときは必須条件ではない。このような場合というのは，$e_1, e_2 \in E_1$ として，

$$\varphi_h(e_1) \neq \varphi_h(e_2)$$

$$\varphi_h(e_1) \wedge \varphi_h(e_2) \neq \phi$$

で表わされる。このような場合には対応者（スレーブ個体）のアドレスのリストが各マスター個体に対して用意されねばならない。スレーブ個体からマスター個体を見出すための φ_h の逆写像は，一意でない写像となる。

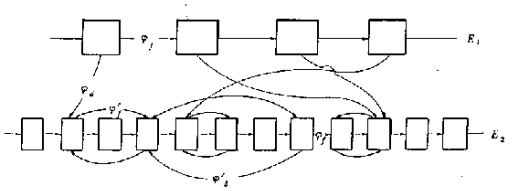


図 1.3.5　φ_f, φ_d, φ_f', φ_b'

$$\varphi_h^{-1} : E_2 \to 2^{E_1}$$

なぜなら2個ないしそれ以上のマスター個体が，共通のスレーブ個体の集合をもつからである。

φ_h^{-1} もまた二つの一意写像に分解できる。

$$\varphi_u : E_2 \to E_1$$

$$\varphi_f'' : E_1 \to E_1$$

ここに φ_u は個体をその第一マスター個体に対応させ，φ_f'' はマスター個体を，同じスレーブ個体を持つ次のマスター個体に対応させる。

φ_f' の場合と同じく φ_f'' の逆写像

$$\varphi_b'' = \varphi_f''^{-1} : E_1 \to E_1$$

も，ある属性で陽に表現できる。

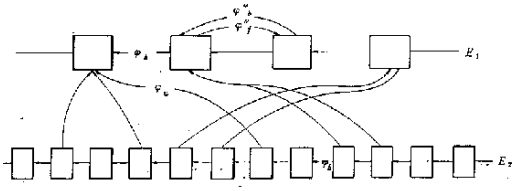


図 1.3.6　φ_h, φ_u, φ_f'', φ_b''

リストの方法も φ_h と同様に適用できる。もし φ_h^{-1} が一意なら，φ_d, φ_f', φ_b' は φ_h^{-1} とともにただ一つの属性で表現できる。これがリング構造である。リング構造はメモリの節約に役立つ。

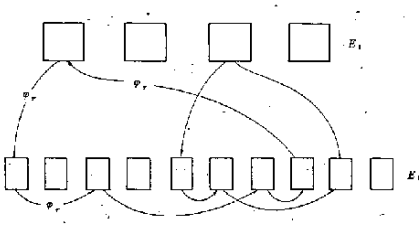


図 1.3.7　リング構造

2個ないしはそれ以上のスレーブ集合が共通のマスター集合を持つ場合，マスター対スレーブ関係を表現する属性を各スレーブに対し与えなければならない。このような構造の例を図 1.3.8 に示す。

階層構造において

$E_i \wedge E_j \neq \phi \quad (i \neq j)$

およびマスター対スレーブ関係は，あらかじめ厳密に定められている。しかし集合の名称をアドレスに沿って持たせることによって，この制限はいくらか緩和される。それは，集合の名称と作成ずみの辞書によって個体のどの属性を検索すべきかがわかるからである。かくして，1マスター個体は異なる集合に属する複数個のスレーブ個体を持てるし，1スレーブ個体は異なる集合に属する複数個のマスター個体を持つことができる。

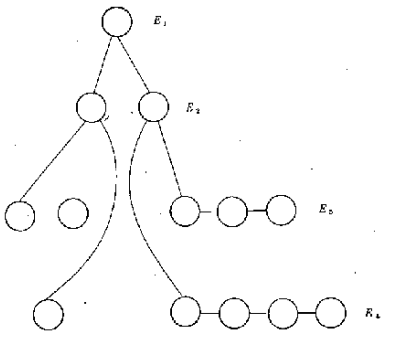


図 1.3.8　複数個のスレーブ集合を持つマスター集合

個体のランダム検索は非構造写像 φ_j の逆写像で特性づけられる。

$$\varphi_j : E \rightarrow A_j$$

まず，アイデンティファイヤによる検索を扱うこととし，対応する写像を1対1としよう。すると

$$\varphi_j^{-1} : A_j \rightarrow E$$

は一意写像ではない。φ_j^{-1} の表現は個体が大容量記憶装置上でどのように配置されているかに関係している。通常，Eは個体が蓄積されているメモリ番地の集合と考えられる。

もしアイデンティファイヤの集合Aと，自然数の列からの密な（ギャップがごく少数の）集合Iとの間の1対1対応を与える簡単なアルゴリズムχが存在すれば，φ_j^{-1} は二つの写像χ，φに分解される。

$$\chi : A_j \rightarrow I \qquad \varphi : I \rightarrow E$$

φは

$$e = \varphi(i) = i \times l + b$$

と定義される。ここに，$e \in E$，$i \in I$で，lは個体の大きさ，bはファイルの先頭アドレスである。

この方法によってメモリを無駄にしないで個体を蓄積するためのアドレスを決定できる。またアイデンティファイヤを指定すれば，検索が迅速かつ容易になる。

しかし多くの場合，便利なアルゴリズムを見出しえない。アイデンティファイヤが数個の独立したコードからなり，各コードは別の意味を持つからである。

この困難を避ける方法の一つは多対1対応χを用いることである。

$$\chi : A_j \rightarrow I$$

このとき，複数個のアイデンティファイヤが1整数に対応することになる。すなわち，複数個の個体が同じ場所に割当てられることがある。対策は，(a)その場所に続く空き地を用意するか，(b)ランダムメモリ上のある領域のコモン・スペース・プールからスペースを確保するかのいずれかである。(a)は簡単で，ランダムメモリのアクセス時の"ブロッキング"を可能にするが，ファイル密度が成長するとき空き地の探索に長時間を要する。

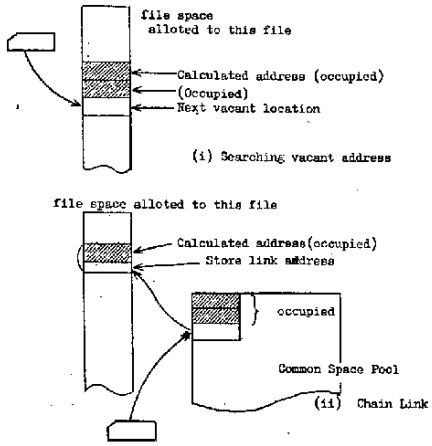


図 1.3.9　多対1対応 χ の蓄積機構

(b)の方法が一般的である。ある新しい個体の占めるべき場所が別の個体によって先取りされたとき，新しい個体のアドレスは本来の場所の一部に記録されている。これを"リンクアドレス"という。2個をこえる場合，リンクアドレスによって次々にチェイニングされる。アイデンティファイヤによる検索も可能である。コモン・スペース・プールは，制御プログラムの"Get and Release"機構によって制御される。すなわちユーザが新しいスペースを必要とするときは，制御プログラムが使用可能なスペースのアドレスを取出し，ユーザがそれを先行する個体にリンクする。不要の個体ができるとリンクをはずし，解放されたスペースを制御プログラムにもどす。

空き地の探索にしても個体のチェイニングにしても，ランダムファイルを呼ぶ回数と処理時間を増し，従って計算機の効率を低下させる。容易にわかるように，各場所に対応するアイデンティファイヤが一様分布であること，すなわち各 $\chi^{-1}(i)$，$i \in I$ の密度ができるだけ同じであることが計算機効率の低下をふせぐために必要である。

アイデンティファイヤは数種類のコードからなるから，その分布には強いひずみがある。ひずみを除去して一様分布を得るために"ランダム化"の技法がある。これには，中央2乗法，素数で割り剰余を求める方法，ラディックス変換法およびこれらの組み合わせ法がある。

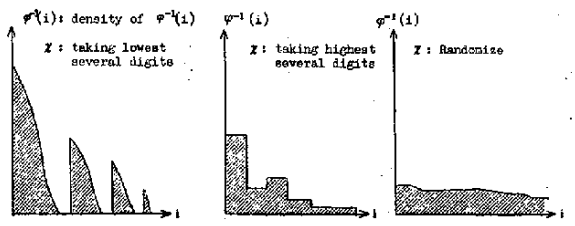


図 1.3.10　ランダム化の効果

アイデンティファイヤからアドレスを得るもう一つの方法は，

インデックスファイルを用意することである。このとき個体は内容的な順序に関係なく作られた順序に蓄積する。この完全にランダムな蓄積法は，コモン・スペース・プールとその制御プログラムを必要とする。しかしこの場合，全個体は一つのチェインに沿ってならんでいる。検索するたびにチェインの第一個体からたどってファイルをスキャンせねばならない。φ_j^{-1} の表現が用意されていないので，検索効率は最低となる。それゆえ，アイデンティファイヤと個体アドレス間の対応テーブルによって，φ_j^{-1} の実際の対応を表現するインデックスファイルを別に作成することが望ましい。

インデックスファイルが大き過ぎてコアの１ロードでおさまらない場合は，数個のインデックスバケットに分割する。インデックスバケットの逐次探索は効率を低下させるので，アイデンティファイヤの集合をバケットアドレスの集合Bに対応させる写像βを考慮せねばならない。

$$\beta \ : \ A_j \ \rightarrow \ B$$

この写像は多対１であるから，再びランダム化を考慮する必要がある。

図1.3.11にインデックスバケットを作るアルゴリズムのサンプルを示す。

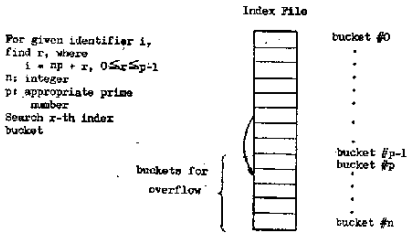


図1.3.11　インデックスバケット作成のアリゴリズム

インデックスエントリーの更新に際し，最初の順序を保つために再配列が必要である。再配列の時間を極小化するには，追加用の別のインデックスエリアを用意し，そのエリアは逐次探索で探すとよい。追加用のインデックスエリアは，ある時点でインデックスファイルを更新するまでそのままにしておく。

さて，ファイルは写像φの表現

$$\varphi \ : \ E \rightarrow A = A_1 \times A_2 \times \cdots\cdots\cdots \times A_n$$

であり，アイデンティファイヤは

$$\varphi_j \ : \ E \rightarrow A_j$$

が１対１であるような属性であった。ところでアイデンティファイヤ以外の属性を使用した検索がありうるが，そのときは必ずしも対応が１対１写像を与えない。このようなとき迅速な検索を実行する方法はクロスインデックスを作成することである。すなわち，検索の対象となる属性ごとに別々のインデックスファイルを作成する。φ_j は多対１であるから，φ_j^{-1} は属性値をEの部分集合に写像する。

$$\varphi_j^{-1} \ : \ A_j \ \rightarrow \ 2^E$$

この写像を表現するために，(a)リストインデックス，(b)チェインインデックスの二方法がある。リ

ストインデックスは，インデックステーブルの関数として個体のアドレステーブルのアドレスを持つ。インデックステーブルのアーギュメントは，個体の対応する属性値である。チェインインデックスは，インデックステーブルの関数として第一個体のアドレスを持つ。インデックステーブルのアーギュメントは，個体の対応する属性値である。

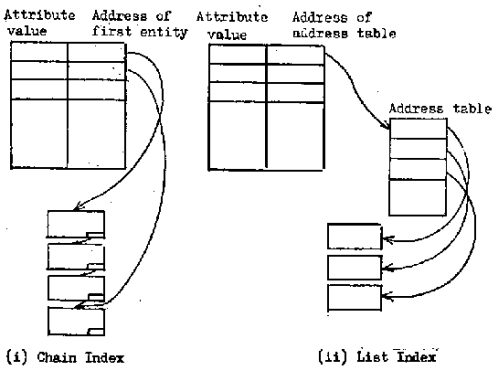


図1.3.12　$\varphi_j^{-1} : A_j \rightarrow 2^E$の表現

1.3.6.2　検　索

検索機構には逐次検索とランダム検索とがある。逐次検索は，与えられた階層構造に沿って逐次処理によって検索を行うものである。ランダム検索は，ある属性が特定の値を持つような個体を見出すことである。一般化していえば，"特定の論理式に対し真なる値を与える個体を検索すること"である。

論理式すなわち検索条件のもっとも簡単な形式は，主題を表わす個体の属性名と定数を"関係演算子"で結合したものである。関係演算子は，

$$= \neq > \geq < \leq$$

である。

〔例〕　Programmer－Apptitude－Test＝'A'

さらに，算術演算子を用いて主題を表わす個体の属性名と定数と変数とを結合した算術式を導入する。算術演算子は，

$$+ \quad - \quad * \quad / \quad ** \quad (\quad)$$

である。

〔例〕　Work－in－main office ＜ 3 * Work－in－branches

ここで，

≪算術式≫ ≪論理演算子≫ ≪算術式≫

なる形式の論理式を基本条件とよぶ。論理演算子は，

$$\wedge \quad \vee \quad \neg$$

である。いくつかの基本条件を結合して複合論理式が作られる。

〔例〕　((Age ≥ '25') ∧ (Material－status＝'SINGLE') ∧
(Location＝'NY')) ∨
((Age ≤ '30') ∧ (Number－of－children ≤ '3') ∧ ¬
(Location＝'NY'))

ランダム検索を迅速に実行するために，サブルーチンでは，インデックス構造が利用される。

$$\varphi^{-1} = (\varphi_1^{-1}, \varphi_2^{-1}, \cdots\cdots, \varphi_n^{-1})$$

もし論理式がただ一つの属性のみに関する基本条件からなるならば，所与の属性値集合A_j中のある値aまたはある部分集合αに対して，Eの部分集合$\varphi^{-1}(a)$または$\varphi^{-1}(\alpha)$を見出すのが検索である。検索条件を，

≪属性名≫ ≪関係演算子≫ ≪算術式（属性名を含まず）≫

なる形式に再編成するならば，検索はφ^{-1}のインデックス機構を参照して達成される。

基本条件C_1で検索される個体の集合をE_1，C_2で検索される集合をE_2として，

$E_1 \wedge E_2$，$E_1 \vee E_2$，$E - E_1$ は，それぞれ，$C_1 \wedge C_2$，$C_1 \vee C_2$，$\neg C_1$に対応する集合であることはすぐわかる。これに，$\neg(\neg C) = C$とド・モルガンの法則

$$\left.\begin{aligned}\neg(C_1 \vee C_2) &= (\neg C_1) \wedge (\neg C_2)\\ \neg(C_1 \wedge C_2) &= (\neg C_1) \vee (\neg C_2)\end{aligned}\right\}$$

とを加えれば，いくつかの基本条件を組み合せる実際の方法が得られる。

しかし，この探索機構は以下のものには適用できない。(a)インデックステーブルが用意されていない属性，(b)2個以上の属性，(c)ある関数を使用する複雑な条件。これらの条件で検索するためにファイルのインデックス構造を利用する一般化したアルゴリズムの発見は，困難ないし不可能であろう。このときは逐次検索しか方法がない。

検索条件をBackus－Naur記法で書くと，

≪条件≫ :: = C

C :: = $e \mid C \vee C \mid C \wedge C \mid \neg C \mid (C)$

ここでCだけが末端記号でなく，e（基本条件），$\wedge$，$\vee$，$\neg$，（，）は末端記号である。$\vee$，$\wedge$，$\neg$の順序のあいまいさを避けるために，文法に制限を加えると，

≪条件≫ :: = $e \mid (C_1) \mid (C_2) \mid (C_3)$

C_1 :: = $e \vee C_1 \mid C_1 \vee (C_2) \mid (C_2) \vee C_1 \mid C_1 \vee (C_3) \mid (C_3) \vee C_1$

C_2 :: = $e \wedge C_2 \mid C_2 \wedge (C_1) \mid (C_1) \wedge C_2 \mid C_2 \wedge (C_3) \mid (C_3) \wedge C_2$

C_3 :: = $\neg e \mid \neg(C_1) \mid \neg(C_2) \mid \neg(C_3)$

$\neg(\neg e) = e$とド・モルガンの法則とを適用し，$\neg e$を基本条件とみなすことによって，文法は少し簡単になる。

≪条件≫ :: = $e \mid (C_1) \mid (C_2)$

C_1 :: = $e \vee C_1 \mid C_1 \vee (C_2) \mid (C_2) \vee C_1$

C_2 :: = $e \wedge C_2 \mid C_2 \wedge (C_1) \mid (C_1) \wedge C_2$

条件に関するある属性のインデックス構造が条件のために使用でき，かつその時に限り，基本条件を"単純"とよぶ。そうでなければ，"複雑"とよぶ。また

$C_1 ::= e \vee C_1 \mid C_1 \vee (C_2) \mid (C_2) \vee C_1$

において，かっこの外に複雑基本条件があるとき，C_1 を"複雑"論理和ブロックとよぶ。

〔例〕

$e_1 \wedge e_2 \wedge e_3$	複雑論理和ブロックでない
$e_1 \vee (e_2 \wedge e_3)$	
$(e_1 \vee e_2) \wedge e_3$	
$e_1 \wedge (e_2 \vee e_3)$	
$e_1 \vee e_2 \vee e_3$	複雑論理和ブロックである
$(e_1 \wedge e_2) \vee e_3$	

ただし，e_1 と e_2 は単純，e_3 は複雑とする。

複雑論理和ブロックに対しインデックス機構は役に立たない。

もし論理和ブロックが単純であるが，複雑基本条件を含むか，または論理和ブロックが入れ子構造をしているとき，論理和記号で結合した条件を分けて検索することが，実際的である。図1.3.13(iv)で e_4 と e_8 が複雑なら，図1.3.14のように分割する。

単純条件については，演算子∨，∧を適用して条件を満足する個体（または個体アドレス）の集合を見出す。これは，アドレス集合のマージングとマッチングとを意味する。もし複数個の条件が同一属性を探す場合は，インデックステーブルを同時に引く方がよい。

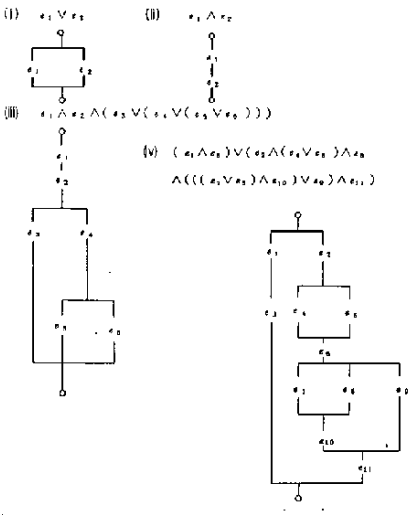


図1.3.13　条件グラフ

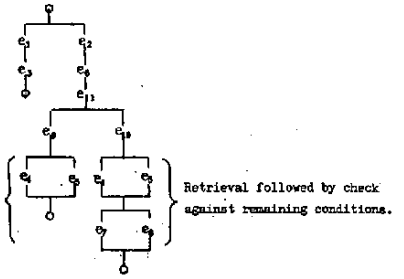


図1.3.14　複雑条件を含む単純論理和ブロックの分解

このようにして見出された個体集合に対して，さらに複雑条件（もしあれば）のチェックを実行する。図1.3.14の探索アルゴリズムを図1.3.15に示す。

i　$(e_1 \wedge e_3) \vee ((e_2 \wedge ((\textcircled{e_4} \vee e_5) \wedge e_6 \wedge (((e_7 \vee \textcircled{e_8}) \wedge e_{10}) \vee e_9) \wedge e_{11})$

iia　$e_1 \wedge e_3$　Found $E_1 \wedge E_3$ and retrieve

iib　$e_2 \wedge e_6 \wedge e_{11}$　Found $E_2 \wedge E_6 \wedge E_{11} = E_0$

　　e_9　Found $E_0 \wedge E_9$, retrieve and check against $e_4 \vee e_5$.

iic　e_{10}　Found $E_0 \wedge E_{10}$, retrieve and check against $(e_4 \vee e_5) \wedge (e_7 \vee e_8)$

図1.3.15　検索アルゴリズムの例

1.3.7　レコード構成の数学的解析[27)〜32)]

1.3.4でファイルの探索時間の短縮という観点からファイルの構成について考えたが，ここでは，記憶装置の無駄使いという観点からファイルのレコードの構成について考えてみる。

1.3.7.1　可変長レコード

レコードの長さは，固定長と可変長の2つの場合が考えられる。レコードの長さが固定されているときには，蓄積すべき領域の大きさは一般に簡単にきめられるが，可変長の場合には詳細な分析が必要である。

可変長のレコードを取り扱う最も簡単な方法は，固定長の概念を導入してレコードの最大長をレコードの長さとすることである。また，磁気テープなどのシークウェンシャル・ファイルの場合には，レコードの長さを示すヘッダーレコードを使用して，完全な可変長として扱える方法もある。しかし磁気ディスクやドラムでは，トラックのような固定領域を組みあわせざるをえない。また，レコードが固定長部と可変長部に分割することができるときには，各々を別々にファイルさせる方法もある。

たとえば，固定長領域に可変部がしまわれている番地を記憶しておいて，2つのファイル間に関係をもたせておく方法である。

可変長レコードの取扱い方には上述のようにいくつかの方法が考えられるが一長一短がある。

レコードの最大の長さを蓄積領域としてすべてのレコードに割合てる方法は各レコードの長さ間にばらつきが少なく，ほとんど一点に集中しているようなときには，有効な方法であるといえるが，一般には無駄なスペースが多くなると考えられる。またヘッダーレコードを使って完全なる可変長として扱うときには，無駄なスペースはないが，その処理が複雑になるであろう。たとえば，コアメモリーの場合，その利点は如何なる場所へも即時呼出しが可能であることであり，レコードを蓄積するときには，任意の場所を指定して行なう。またそれが必要とされたときには，格納してある場所そのものを指定することによって呼出しが行なわれる。そこで，コアメモリーにおいてレコードを可変長として扱おうとすると，2つの困難なことがおきる。一つはレコードがどのくらいの長さをもっているかを計算機が知る前に格納場所を指示しなければならないことである。したがって，蓄積場所として指示した場所とそれに続く空のスペースが，レコードの長さより小さいということがおこるかもしれない。たとえ，レコードの長さが始めからわかっていたとしても，レコードが完全に入りうる空のスペースがメモリー内にあるという保証はなにもない。また，もう一つの困難なことは，格納場所を見失わないことである。（それは，レコードが必要とされたときに，格納場所を指示しなければならないから，蓄積時の格納場所を見失なわないことが必要なのである。）

このように，無駄なスペースの軽減をはかると，その処理が複雑になってしまい，処理を簡単にすると無駄なスペースが増大してしまう。

レコードの最大の長さを割合てる方法の最大の利点は"固定長"という概念をもりこんだことにあり，ヘッダーレコードをつけて完全なる可変長として扱う方法の利点は"無駄なスペースが０"であることであった。この両者のアプローチによって，ここに『セグメント方式』を考える。

1.3.7.2　セグメント方式

(i)　セグメント方式とは，

セグメント方式とは，蓄積すべき領域をあらかじめ固定長領域に分割し，そこにレコードを割付ける方式である。したがってレコードは１個またはそれ以上の固定長領域の連に蓄積される。

この方式によってレコードは固定長の場合と同様に扱えるが，問題は，固定長領域 —— セルと名付ける —— の長さをどのくらいにしたらよいか，ということである。この方式は，２つの無駄なスペースを伴う。一つはレコードがいくつかのセルにまたがっているときに，各セルが同一のレコードのものであることを示すためのスペースであり，もう一つは，最後のセルの残りのスペース（レコードの長さがセルの整数倍になっているときにはゼロ）である。

セルの長さを短かくとれば，最後のセルに残るスペースは少なくてすむが，セルの数が多くなり，セル間のつながりを示すためのスペースが多くなってしまう。反対にセルの長さを長くとると，セルの数は少なくてすむが，最後に残るスペースが多くなってしまう。２つの無駄なスペースをバランスさせて，その和を最小にする事を考える。無駄なスペースを最小にする，セルの長さを最適セル長として定義する。

したがって，セグメント方式とは，『蓄積すべき領域を無駄なスペースをできる限り少なくするような，固定長領域に分割し，そこに可変長のレコードを割付ける方式である。』

(ii)　無駄なスペース

蓄積媒体として，磁気コアメモリーを仮定して話を進める。

セルを，即時呼出しが可能なように１語あるいは数語（machine words）からなる番地づけ可能な（addressable）領域と定義する。

長いレコードに対しても，大きなスペースを確保する必要がないように１つのレコードのすべてのセルをリスト構造によって結んでおく。

また，レコードの位置を見失わないように，レコードの名前とそのレコードが蓄積された最初のセルの名前からなる"リスト"をメモリー内に作っておく。

したがって，いかなる長さのレコードがきても最初のセルを指定するだけで，長さに応じていくつかのセルを使って蓄積することができるし，またレコードが必要とされた時には，レコードの名前でリストを探してレコードの最初のセルの位置を知って，その位置（番地）を直接指定してやれば，後はいもずる式に１つのレコードを取り出すことができる。

レコードで占められるセルの長さをa語（machine words）とし，セルをリスト構造にするために，次に続くセルの名前を入れるためのスペースの長さをb語とする。メモリー内のレ

コードを図示すると図1.3.16のようになる（セルの名前は今の場合はセルの最初の語の番地（adress）とする）。

図1.3.16において，第1列目はレコードのリストでレコード名とレコードで占められた最初のセル名（N_1）からなっている。第2列目はレコードで占められた最初のセルである。セルの最後は，次のセル名を入れるためのスペース（b語）が続いている。最後の列はレコードで占められる最後のセルで，最後に使用されない残りの部分が存在している。

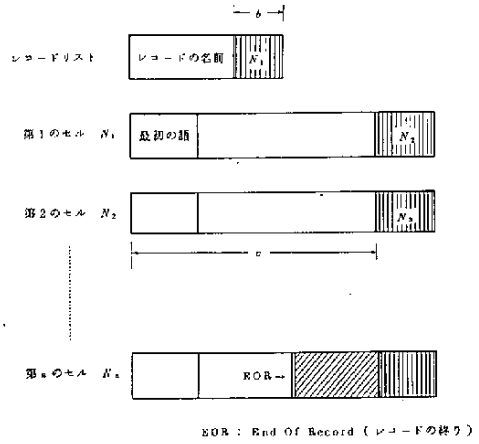


図1.3.16　メモリー内のレコード

いま，レコードの長さをlとし，$n(l)$個のセルからなっているとすると，無駄なスペースは図1.3.16における縦線部（次のセル名を入れるスペース）

$$\{ b \times (n(l) + 1) \}$$

と，斜線部（最後のセルの残りの部分）

$$\{ n(l)\, c - l \}$$

である。従って無駄なスペース$w(c)$は

$$w(c) = \{ n(l)\, c - l \} + b \{ n(l) + 1 \} \qquad (1)$$

と書くことができる。

実際には，lは分布が既知の確率変数と考えることができるから，(1)式をlに関して平均して

$$W(c) = \{ E[n(l)]\, c - L \} + b \{ E[n(l)] + 1 \} \qquad (2)$$

（Lはlの期待値）

$W(c)$を最小にするcをみつけるのが目的である。（このときのcが最適セル長Cになる。）種々の分布形について最適セルの長さを求めてみる。

(iii)　連続モデルにおける最適セル長

レコードの長さl，セルの長さcが共に連続変数である場合を考えよう。lの期待値をL，cの最適値をCとすると，$L \gg 1$，$C \gg 1$と考えることができる。

$p(l)$をレコードの長さの確率密度関数とする。lが$[c(n-1),\ cn]$の間にある確率は$\int_{c(n-1)}^{cn} p(l)\,dl$で，そのとき$n(l)$個のセルが必要である。したがって(2)式は次のように書き換えることができる。

$$W(c)=(c+b)\sum_{n=1}^{\infty}n\int_{c(n-1)}^{cn}p(l)dl-L+b \tag{3}$$

(3)式を微分すると，

$$W'(c)=\sum_{n=1}^{\infty}n\int_{c(n-1)}^{cn}p(l)dl+(c+b)\sum_{n=1}^{\infty}n$$

$$\{np(cn)-(n-1)p(c(n-1))\}$$

右辺の第2項を書き換えて，

$$W'(c)=\sum_{n=1}^{\infty}n\int_{c(n-1)}^{cn}p(l)dl-(c+b)\sum_{n=1}^{\infty}np(cn) \tag{4}$$

ここで，Wは微分可能でcが0と∞（無限大）に近づくとき，Wは限りなく増大するから，cがその最適値Cをとるときは，$W'=0$となる。

したがって，(4)式より最適値Cは次の関係を満たさねばならない。

$$\sum_{n=1}^{\infty}n\int_{C(n-1)}^{Cn}p(l)dl=(C+b)\sum_{n=1}^{\infty}np(Cn) \tag{5}$$

最小期待無駄スペース$W(C)$は(5)式を(3)式に代入して得られる。

$$W(C)=(C+b)^2\sum_{n=1}^{\infty}np(Cn)-L+b \tag{6}$$

(5)式を満足するCの値がいくつかあるときには(6)式に各Cを代入し，(6)式を最小にするCの値を最適値とすればよい。

例　1　　指数分布の場合

レコードの長さlが指数分布に従う場合，その確率密度関数は

$$p(l)=\frac{1}{L}\exp\left(-\frac{l}{L}\right) \tag{7}$$

と書くことができる。

$$\therefore\ \int_{C(n-1)}^{Cn}p(l)dl=\left\{\exp\left(\frac{C}{L}\right)-1\right\}\exp\left(-\frac{Cn}{L}\right)$$

$$=L\left\{\exp\left(\frac{C}{L}\right)-1\right\}p(Cn)$$

この関係を(5)式に代入すると，

$$L\left\{\exp\left(\frac{C}{L}\right)-1\right\}=C+b \tag{8}$$

(6)式に，(7)式と(8)式を代入すると，最小無駄スペースは，

$$W(C)=C+2b \tag{9}$$

と書ける。また，(8)式より　$y = \dfrac{C}{L}$　とおいて，指数級数を使って書くと，

$$\frac{b}{L} = e^{y} - y - 1 = \sum_{i=2}^{\infty} \frac{y^{i}}{(i!)} \tag{10}$$

ここで，$C = \sum_{k=0}^{\infty} a_k b^{\frac{k}{2}}$　（a_k は実数）とおいて(10)式において未定係数法でとくと，

$$C = \sqrt{2bL} - \frac{b}{3} + \frac{b^{3/2}}{9\sqrt{2L}} - \frac{2b^{2}}{135L} + \cdots\cdots\cdots \tag{11}$$

となる。いま考えているのは連続モデルの場合であるから　$L \gg 1$　と考えることができる。したがって(11)式において右辺の第2項以下は無視できると考えられる。

また，$(C + b)$ は整数でなければならないから，$b = \dfrac{1}{4}$　のとき，$(L,\ C + b)$ は(11)式より（16，3），（68，6）などの値をとる。

例　2　　一般の連続分布の場合

$p(l)$ を ある連続分布の密度関数とする。p を $l = cj$（$j = 0,\ 1,\ 2,\ \cdots\cdots$）において p に一致する関数 $\bar{p}$ で近似しよう。

p は $\bar{p}$ と近似誤差 e の和で表わされるから

$$p(l) = \bar{p}(l) + e(l) \tag{12}$$

ここで，$e(l)$ は $\bar{p}$ が p と一致する c の格子上，つまり $l = cj$（j は非負整数）において $e(cj) = 0$ である。

(12)式を(4)式に代入すると，

$$W'(c) = \sum_{n=1}^{\infty} n \int_{c(n+1)}^{cn} \bar{p}(l)\,dl + \sum_{n=1}^{\infty} n \int e(l)\,dl$$

$$-(c + b) \sum_{n=1}^{\infty} n p(cn) \tag{13}$$

ここで，$\bar{p}$ を折れ線とすると p は c の格子上で多角形的に近似されるわけである。

l が $[\,c(n-1),\ cn\,]$ にあるとき

$$\bar{p}(l) = \alpha + \beta l$$

とおくと

$$\begin{aligned} \bar{p}(c(n-1)) &\equiv \alpha + \beta c(n-1) = p(c(n-1)) \\ \bar{p}(cn) &\equiv \alpha + \beta cn = p(cn) \end{aligned} \tag{14}$$

系（System）は行列式 c をもつ線型変換と考えられるから，α と β はともに求めることができる。

$$\int_{c(n-1)}^{cn} \bar{p}(l)\,dl = \alpha c + \beta c^{2}\left(n - \frac{1}{2}\right) = \frac{c}{2}\{p(c(n-1)) + p(cn)\}$$

したがって，(13)式の右辺の第1項は次のように書き換えられる。

$$\frac{C}{2}\sum_{n=1}^{\infty} n\{p(c(n-1))+p(cn)\}=C\sum_{n=1}^{\infty} np(cn)+\frac{C}{2}\sum_{n=0}^{\infty} p(cn)$$

$$\therefore\ W'(c)=\frac{C}{2}\sum_{n=0}^{\infty} p(cn)-b\sum_{n=1}^{\infty} np(cn)+\sum_{n=1}^{\infty} n\int_{c(n-1)}^{cn} e(l)dl$$

この式はもはや$\bar{p}$を伴っていない。前述の指数分布の解析において，レコードの長さの平均値Lが適切なパラメーターであることがわかった。そこで，Lについて考えてみよう。

$$L=\int_0^{\infty} lp(l)dl=\int_0^{\infty} le(l)dl+\sum_{n=1}^{\infty}\int_{c(n-1)}^{cn} l\bar{p}(l)dl$$

$$\int_{c(n-1)}^{cn} l\bar{p}(l)dl=\alpha c^2\left(n-\frac{1}{2}\right)+\beta c^2\left(n^2-n+\frac{1}{3}\right)$$

$$=\frac{c^2}{2}\left\{\left(n-\frac{2}{3}\right)p(c(n-1))+\left(n-\frac{1}{3}\right)p(cn)\right\}$$

$$\therefore\ \sum_{n=1}^{\infty}\int_{c(n-1)}^{cn} l\bar{p}(l)dl=\frac{c^2}{2}\left\{2\sum_{n=1}^{\infty} np(cn)+\frac{1}{3}p(0)\right\}$$

$$L=C^2\sum_{n=1}^{\infty} np(cn)+\frac{c^2p_0}{6}+\int_0^{\infty} le(l)dl$$

$$(\because p_0\equiv p(0))$$

したがって，(15)式の第2項は次のように書き換えられる。

$$b\sum_{n=1}^{\infty} np(cn)=\frac{bL}{c^2}-\frac{bp_0}{6}-\frac{b}{c^2}\int_0^{\infty} le(l)dl \qquad (16)$$

$$1=\int_0^{\infty} p(l)dl=\int_0^{\infty} e(l)dl+\sum_{n=1}^{\infty}\int_{c(n-1)}^{cn}\bar{p}(l)dl$$

最後の項は次のように書き換えられる。

$$\sum_{n=1}^{\infty}\int_{c(n-1)}^{cn}\bar{p}(l)dl=c\sum_{n=0}^{\infty} p(cn)-\frac{cp_0}{2}$$

$$\therefore\ \frac{c}{2}\sum_{n=0}^{\infty} p(cn)=\frac{1}{2}+\frac{cp_0}{4}-\frac{1}{2}\int_0^{\infty} e(l)dl \qquad (17)$$

(15)式に(16)式と(17)式を代入して，$e(l)$の項をもとめると，

$$2W'(c)=\left(1-\frac{2bL}{c^2}\right)+\left(\frac{c}{2}+\frac{b}{3}\right)p_0+\sum_{n=1}^{\infty}\int_{c(n-1)}^{cn}\left(2n+\frac{2bl}{c^2}-1\right)e(l)dl \qquad (18)$$

(18)式は，$W'(c)$がc自身の代りに$e(l)$とpの2つの性質（原点における値p_0とその平均

値）によって表わされたことを示している。

$$2W'(c)=1-\frac{2bL}{C^2}+E_1(c) \tag{19}$$

ここで，

$$E_1(c)=\left(\frac{c}{2}+\frac{b}{3}\right)p_0+\sum_{n=1}^{\infty}\int_{c(n-1)}^{cn}\left(2n+\frac{2b\ell}{C^2}-1\right)e(\ell)d\ell \tag{20}$$

ここで，$W'(c)=0$ とおくと，

$$\frac{2bL}{C^2}=1+E_1(C) \tag{21}$$

ここで $C^2=2bL$ となるためには，$E_1(C)=0$ とならねばならない。が$E_1(C)$はCがわからないかぎり計算できない。また，(20)式からは$E_1(C)\ll 1$となることも，関数pと線型関係にあるかどうかということもいえそうにない。そこで，視点を変えて，$E_1(C)=0$なるためには，pがどのような条件を満足すべきかを考えてみよう。

$E_1(c)=0$が成立つためには，$p_0=0$ が成立ち，いかなるℓに対しても$e(\ell)=0$ が成立つことである。

もっと一般的にいうと；p_0と$e(\ell)$が十分に小さければCは$\sqrt{2bL}$に近づく。すなわちpが原点において十分小さく，Cの格子上でpに一致するような線型関数によって近似できるならば，$E_1(c)=0$となる。

しかし，この関係は0ではじまり，線型であるいかなる密度関数も満足するわけではない。（十分条件であるが，必要条件にあらず）そこで，この関係を満足する範囲をはっきりさせるために，実際に条件に適合するような密度関数を作ってみよう。

$C^2=2bL$ の関係を満足するCとLの組（pair）を選んで，Cが最適値である密度関数pを作る。

原点を出発点として，長さCのℓ軸上に写影をもつ線分を連続的に描く。このようにして作られたpが密度関数になるための条件（$p\geq 0$，$\int_0^{\infty}p(\ell)d\ell=1$）を満足し，平均値$L$をもつならばそれが求めるものである。

$p_0>0$で$e(\ell)<0$ なるpは(20)式の右辺を0にすることができる。つまり$E_1(C)=0$。また，$C_0^2=2bL$ を満足する与えられた（C_0，L）に対して，$C=C_0$ 上につくられた密度pを，E_1 を小さく保ちながら少し動かすことによって，Cを十分に$\sqrt{2bL}$に近づけることができる。したがって$E_1(C)=0$を満足する$p(\ell)$の分布形は上述の$p_0=0$，$e(\ell)=0$であるような形の他に，いろいろな形が考えられる。つまり，たくさんの密度関数が同一の性質（$E_1(C)=0$）をもつわけである。

したがって，pが与えられても，Cをみつける一般的な方法は定義できない。それで，pの性

質をもっと詳しく調べるために$C \fallingdotseq \sqrt{2bL}$なる$p$の性質について考えよう。

与えられたpに対して$C \fallingdotseq \sqrt{2bL}$ということは，$E_1(C)$が0でないということである。(20)式より$E_1(C)=0$とならない原因は2つの基因によるものであることがわかる。一つはp_0に基因する項$\{\frac{c}{2}+\frac{b}{3}\}$と，もう一つは$e(l)$に基因する項$\sum_{n=1}^{\infty}\int_{c(n-1)}^{cn}(2n+\frac{2bl}{c^2}-1)e(l)dl$である。そして，$e(l)$はとりもなおさず$p$の近似の仕方によるものである。そこで$p$の近似方法を変えることによって$e(l)$がどのような影響をうけるかを調べてみよう。

いままでpの近似は線型（一次）であったが，これからは，3次式で近似してみよう。$\bar{p}=\alpha+\beta l+\gamma l^2+\delta l^3$とおくと，系（System）は行列式$C^4$をもつ。線型に近似した場合と同様に解析していこう。重要結果は，

$$\int_{c(n-1)}^{cn}\bar{p}(l)dl=\frac{C}{2}\{p(c(n-1))+p(cn)\}+\frac{1}{12}\{p'(c(n-1))-p'(cn)\}$$

$$\int_{c(n-1)}^{cn}l\,\bar{p}(l)dl=\frac{C^2}{2}\{(n-\frac{7}{10})p(c(n-1))+(n-\frac{3}{10})p(cn)\}$$

$$+\frac{C^3}{12}\{(n-\frac{3}{5})p'(c(n-1))-(n-\frac{2}{5})p'(cn)\}$$

(19)式と同様な形式で書くと，

$$2W'(c)=1-\frac{2bL}{c^2}+E_3(c)$$

ここで，

$$E_3(c)=(\frac{c}{2}+\frac{3b}{10})p_0+(\frac{c^2}{6}+\frac{2bc}{15})(\sum_{n=1}^{\infty}p'(cn)+\frac{p_0}{2})$$

$$+\sum_{n=1}^{\infty}\int_{c(n-1)}^{cn}(2n+\frac{2bl}{c^2}-1)e(l)dl \qquad (22)$$

線型近似の場合のように解析していくと，$E_3(C)=0$を満足する密度関数を作る事ができるが，$E_3(C)$を0にする一番簡単な方法は，$p_0=0$，$\sum_{n=1}^{\infty}p'(cn)+\frac{p_0}{2}=0$とすることである。しかし，この条件を満足する密度関数はたくさん存在し（線型の場合以上に）一意的に求めることはできない。その他に$E_3(C)$を0にするには，前述のように個々の項を0にする以外に，次のような固有の項を0にしてもなりたつ。

$$\frac{C}{3p_0}\{\sum_{n=1}^{\infty}p'(Cn)+\frac{p_0'}{2}\}=\frac{5C+3b}{5C+4b} \qquad (23)$$

$c=C$のとき，$W'(C)=0$なるためには，$E_1(C)=0$あるいは$E_3(C)=0$が成立たねば

ならない。したがって$W'(C)\neq 0$ということは，$E_1(C)$，$E_3(C)\neq 0$ なることであり，$W'(C)$の式において，$E_1(C)$，$E_3(C)$は丁度，誤差項のようにはたらくと考えられる。そこで，線型から三次近似へと，近似方法を高次にすることによってその誤差項がどのような影響をうけるか調べたわけである。もし，"近似を高次にすればする程，誤差項が減少する"ということが成立つなら，しめたものである。解析結果(20)式と(22)式を比較すると，確かに減少してはいるが，一般的な結論を導きだすまでにはいたらない。結局，高次近似をおこなったことによってpに関する情報はふえたが問題の本質的な解はなにもえられなかった。

これまでにわかったことは，

① 種々の分布に対して（密度関数がいろいろな形をもつ）最適セル長Cは$\sqrt{2bL}$に近づく。

② ある与えられた密度関数において，Cが$\sqrt{2bL}$に等しくなるかどうかの評価は，原点におけるpの値が重要な意味をもつ。

(IV) 離散モデルにおける最適セル長

実際には，レコードの長さlもセルの長さCも離散変数であると考えられる。

lを語（words），kを字（characters）と考えると，$l=kd$とおける。但しdは長さd語（words）の"データキャラクター"（data character）である。たとえば42 bit一語で6 bit一字とすると，$d=\frac{1}{7}$ である。したがってlはdの整数倍（$k=1$，2，3，……）と考えられる。

いま，$f(l)$をレコードが長さl語をもつ確率とすると関数$f(l)$は，整数kに対して$l=kd$がなりたつときだけゼロでない値をとる。

長さkd（$=l$）のレコードを蓄積するのにn個のセルを使うkの値の集合は次のように書ける。

$$K_n=\{k:c(n-1)<kd\leq cn\}$$

したがって，1レコード当りの平均無駄スペースは，

$$W(c)=(c+b)\sum_{n=1}^{\infty}n\sum_{k\in K_n}f(kd)-L+b \qquad (24)$$

である。この式は，連続型の(3)式に相当するものであるが，これから先は，連続型のような解析はおこなえない。なぜなら(24)式をCで微分すると，$f(kd)\neq 0$で，$c=kd/n$を満足する整数kとnが存在する以外の点では，$W'(c)=1$となってしまう。したがって，これから先は，$f(kd)$に実際の分布をあてはめて考えていくことにする。

例 1　幾何分布の場合

$L=Kd$とおく。但しKは1レコード当りのデータキャラクターdの平均値である。レコードの長さkdが幾何分布に従うとき，1レコードがk個のデータキャラクターをもつ確率は，

$$f(kd)=\frac{1}{K}\left(1-\frac{1}{K}\right)^{k-1} \qquad K>1,\ k=1,\ 2,\ 3\cdots\cdots \qquad (25)$$

と書ける。いたずらに代数学をふりまわさなくてもすむように実際に数値を与えて考えることにする。

30bit－1語（word）と考える。セルの名前を入れるスペースが15bitからなるとすると $b=\frac{1}{2}$。そしていまデータキャラクターを30bitとすると $d=1$。簡単のため $m=c-\frac{1}{2}$ とおくと，

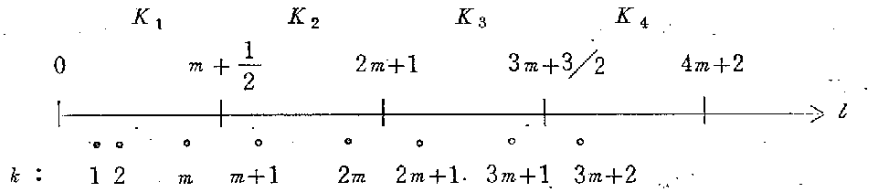
$$c+b=c+\frac{1}{2}=m+1$$

$(c+b)$ は整数でなければならないから，m は非負整数と考えられる。$\beta=1-1/K$ とおくと，いま $d=1$ であるから(25)式は，

$$f(kd)=f(k)=(1-\beta)\beta^{k-1}$$

とかける。

レコードの長さ l 軸上に k の値をプロットしてみると，下図のようになっている。



縦の線はセルの境界を示している。最初のセルは $(m+\frac{1}{2})$ 語からなっている。第2のセルは $(m+\frac{1}{2})$ から $(2m+1)$ まで，第3のセルは $(2m+1)$ から $(3m+3/2)$，等々……。各セルにおちる長さ k のレコードを"。"で示してある。また，K_i は i 番目のセルにおちるレコードの集合である。上図より，

$$\sum_{k\in K_1} f(k)=(1-\beta)\sum_{k=1}^{m}\beta^{k-1}=1-\beta^m$$

$$\sum_{k\in K_2} f(k)=(1-\beta)\sum_{k=m+1}^{2m+1}\beta^{k-1}=\beta^m(1-\beta^{m+1})$$

この式は $m=0$ のときもなり立つ。（$m=0$ のとき，$\sum_{k\ \in K_1} f(k)=0$，$\sum_{k\ \in K_2} f(k)=1-\beta=f(1)$ で，$C=\frac{1}{2}$ となる）

K_1 と K_2，K_3 と K_4 の関係から，一般式

$$\sum_{K_{n+2}} f(k)=\beta^{2m+1}\sum_{K_n} f(k)$$

が導かれる。

$$Y(m) = W\left(m+\frac{1}{2}\right) + K - \frac{1}{2} = W(c) + L - b$$

とおくと，(24)式から，

$$\therefore \quad Y(m) = (m+1)\sum_{n=1}^{\infty} n \sum_{K_n} f(k) \tag{26}$$

$\sum_{n=1}^{\infty} n \sum_{K_n} f(k)$ を奇数部と偶数部にわけてその和を計算する。

奇数 n ： $\sum_{j=0}^{\infty} (1+2j)(1-\beta^m)\beta^{(2m+1)j}$

偶数 n ： $\sum_{j=1}^{\infty} (2j)(1-\beta^{m+1})\beta^{m+(2m+1)(j-1)}$

和は $(1+\beta^m)/(1-\beta^{2m+1})$ となる。

$$\therefore \quad Y(m) = (m+1)\frac{1+\beta^m}{1-\beta^{2m+1}} \tag{27}$$

$$Y'(m) = \frac{(1+\beta^m)(1-\beta^{2m}) + \log\beta(\beta^m+\beta^{2m})(m+1)}{(1-\beta^{2m+1})^2} = 0$$

$$\beta^m = e^{m \log \beta}$$

$$= \sum_{i=0}^{\infty} \frac{(m\log\beta)^i}{i!}$$

$$1+\beta^m - \beta^{2m+1} - \beta^{3m+1} + m\log\beta\beta^m + m\log\beta\beta^{2m+1}$$

$$+ \log\beta\beta^m + \log\beta\beta^{2m+1} = 0$$

ここで，$m = \sum_{k=0}^{\infty} a_k (\log\beta)^k$ とおいて解く。

いま，$b = \frac{1}{2}$ であるから，$p - \frac{1}{2} \leqq C \leqq p + \frac{1}{2}$ なる p を考えると，

$$p = \sqrt{2bK} - \frac{b}{3} + \frac{3/b^{3/2}}{9\sqrt{2K}} + \cdots\cdots \tag{28}$$

$$\therefore \quad W(C) = C + \frac{3}{2}b + \cdots\cdots \tag{29}$$

例　2　　特別な分布 I

$f(25) = 1 \qquad d = 1$

すべてのレコードの長さがきっちり 25 words である場合を考えよう。

$b = \frac{1}{2}$ とすると $\sqrt{2bK} = 5$

したがって，$(C+b)$が整数であるためには$4\frac{1}{2}$か$5\frac{1}{2}$のどちらかの値をCがとればよい。しかしここで　$C=25\frac{1}{2}$とすると，$C=5\frac{1}{2}$　の場合より無駄スペースが少なくてすむ。さらに，$C=12\frac{1}{2}$とすると，各レコードは丁度2個のセルを使ってストアされのこりは0となる。当然　$C=5\frac{1}{2}$　のときよりセル数が少なくてすむから，無駄スペースは，理論値$C=5\frac{1}{2}$の場合より小さい値をとる。なぜこのようなことがおきたのであろう，理由は簡単である。われわれの目的は，無駄スペースの2つの源をバランスさせて最小にすること（入ってないセルと，セルの数を減らすこと）であった。一般に，2つの源は確率変数lの関数である量に関係するものである。この例の場合には，前もってlがわかっているので，最後のセルの残りの部分をなくして完全に入りうるスペースを割当て，その上でレコード1個当りのセルの数を最小にすることが目的となる。そしてそれが最小無駄スペースを与えるCをきめることになる。この例のように，ある単純な関係をもって小さな区間にfが集中する（分散が小さい）ような時には$C=\sqrt{2bL}$にはならない。

例　3　　特別な分布Ⅱ

$$\begin{cases} f(\ 50)=\frac{1}{2} \\ f(150)=\frac{1}{2} \end{cases}$$　なる分布を考える。

$d=1$　とすると　$L=K$

答は解析するまでもなく　$C=50$　とすべきであるが　$C=\sqrt{2008}$　としてこれが50になるためにはbが極端に小さい（$b\leq 0.058$）とき以外はなりえない。この分布の分散は2500で非常に大きい。例2において分散の小さい一点に集中するような分布は　$C=\sqrt{2bL}$　がなりたたないとした。この例と比較すると，分散が大きいとか小さいとかだけでは，　$C=\sqrt{2bL}$　の可否に対する一般的結論がいえないことがわかる。

要するに，例2も例3も，われわれが考えた2つの無駄スペースのうち，1つの無駄（最後のセルの残り）を最小にするCの値が，明らかにわかる場合である。

したがって，もはや2つの無駄スペースのバランスによってその和を最小にする方法ではなくて，1つの無駄スペース（セルの残り）を最小にするCの値の中から，全体のセルの数（あるいは平均セル数）を最小にするCを選べば，無駄スペースが最小にできる。

(V)　結　　論

可変長レコードを扱う1方法として，セグメント方式を考えた。セグメント方式とは，データ領域をあらかじめC語のセルに分割しておき，レコードの大きさに応じて，このセルをリスト構造によって結んで蓄積する方法である。そしてメモリースペースの利用度の面から最も効率の高いセルの長さを最適セル長と定義し，この大きさについて論じた。

セグメント方式は，2つの無駄なスペースを伴う。一つは，リストとして各セルをリンクするのに要するセル名（セルの先頭アドレス）を示す部分と最後のセルで使われない部分とからなる。

ここでレコードの長さl，使用されるセル数$n(l)$，セルの名の長さb，セルの大きさcとすると，無駄なスペースは，

$$w(c)=\{n(l)c-l\}+b\{n(l)+1\}$$

であらわされる。lが既知の分布に従う確率変数であるとき，$w(c)$を平均して1個当りの無駄なスペースの期待値をえることができる。この期待値を最小にするCの値を最適セル長と定義し，lのいろいろな分布に対してこの値を求めた。lが指数分布に従う場合には，$C=\sqrt{2bL}-\frac{b}{3}$となり，ここでbが小さな値（たとえば$\frac{1}{2}$）をとる場合にはCはほぼ$\sqrt{2bL}$と考えられる。同様にして，一般的な連続分布の場合にもこの関係が成り立つことがわかった。一方，lが離散分布に従う場合には，指数分布の場合と同様な解がえられる場合と，そうでない場合もあった。$C\fallingdotseq\sqrt{2bL}$となるのは，レコードの長さの分布が一点に集中しているような場合であった。この原因は簡単で，定義された最適セル長を与える式（$C=\sqrt{2bL}$）は，2つの無駄スペースの和を，両者をバランスさせて最小したことによってえられたものであることによる。つまり，レコードの長さが1点に集中するような場合には，2つの無駄なスペースのうち，最後のセルの残りを0にするセル長はすぐ求められる（集中点をxとすると，xとその約数達）。一方が0になったら，その上でもう一方の無駄なスペース，つまりセルの名前の入るスペースを，最小にするようにすればよい。したがって，最後のセルの残りを0にするCの値の中で，セルの数が1番少なくてすむものが最適セル長となるわけである。結局，ある特別な分布に対して，2つの無駄なスペースのうち明らかに一方を0（あるいはそれに近い値）にするCの値が求められるときには，もはや，2つの無駄なスペースをバランスさせることを考えないで，求められたCの上で，もう一方を最小にすることを考えればよい。

(VI) 考　察

われわれの得た結果は，レコードの長さlがどのような分布に従っても，最適セル長は$\sqrt{2bL}$に近くなるというものであったが，ここに興味ある事実がある。

いま，無駄なスペースのうち，最後のセルに残る空のスペースを平均的に$\frac{C}{2}$であると仮定しよう。Lをレコードの長さlの平均値，$[x]$をxを越える最小整数をあらわすものとする。

$$(L/C)-E([l/C])=\frac{1}{2}$$

したがって1レコードの期待セル数は，

$$E([l/C])+1=\left(\frac{L}{C}\right)+\frac{1}{2}$$

と書ける。

$$\therefore\ W(c)=\frac{C}{2}+b\left\{\left(\frac{L}{C}\right)+\frac{3}{2}\right\}=\frac{C}{2}+\frac{bL}{C}+\frac{3b}{2}$$

ここで，$W'(c)=0$　とおくと，

$$W'(c)=\frac{1}{2}-\frac{bL}{C^2}=0 \qquad \therefore \quad C=\sqrt{2bL}$$

そして，$W(C)=\frac{C}{2}+\left(\frac{C}{2}+\frac{3}{2}b\right)=C+\frac{3}{2}b$

これらの結果は，分布を考えて解析したものと一致している。

解析において，蓄積媒体をコアメモリーと仮定したが，蓄積媒体（テープ，ディスク，ドラム）を変えた場合には，どのようなことを考慮しなければならないか，結果はどのような影響をうけるかを考えてみよう。

テープの場合は，その物理的性質上（コアメモリーのように番地という概念がない）領域を分割するというより，レコード自身を分割して蓄積すると考えた方が素直である。

また，1レコードのセル間の関係は，リスト構造にはできないので，連続的なものである。それで，レコード名をしまうリストもいらないし，次のセル名を入れる領域もいらない。しかし，処理（ソート，サーチ）上の必要性から1つのレコードがいくつかに分けられたときには，その各々にレコードのキィーが繰り返えされねばならない。そして，各セルを区別するためのサブキィーももたねばならない。レコードのキィーとセルのサブキィーの和が"b"に相当する。したがって"b"の意味は，コアメモリーの場合と異質のものとなる。また，リストがいらないので，1レコードの期待無駄スペース$W_T(c)$は，

$$W_T(c)=\{E(n(l))c-l\}+b\cdot E(n(l)) \qquad (30)$$

と定義される。しかし，最適セル長Cは$\left(\sqrt{2bL}-\frac{b}{3}\right)$になることにかわりはない。（$b$の値が大きいであろうと思われるので$-\frac{b}{3}$をくっつけて考えた方がいい。）そのとき，

$$W_T(C)=C+\frac{b}{2} \qquad (31)$$

と考えられる。

ディスクやドラムの場合には，トラックのような固定領域を考慮する必要がある。（なぜなら，読込／書出の単位はトラックである）計算されたセルの大きさCの整数倍がトラックの大きさとなっているときはいいがその他のときには，その差は空のスペースとなり，無駄なものと考えられる。

レコードの平均セル数を$E[n(l)]$とし，トラック内の残りをeとすると，トラック内のレコードの数は

$$M=\frac{T-e}{(c+b)E[n(l)]}$$

であらわされる（ここで，Tはトラックの大きさをあらわす）。

したがって，トラック内の無駄スペースは，

$$W_D=\{cE[n(l)]-L\}\times M+b(E[n(l)]+1)\times M+e \qquad (32)$$

であらわされるが，これを最小にするCの値を求めればよい。しかし，Cをトラックの大きさ（T）の約数としてとれば，コアメモリーの場合と同様になる。（(32)式において$\varepsilon \equiv 0$）したがって，コアメモリーの式（$C=\sqrt{2bL}$）で求めたCを少し変化させるだけでトラックの大きさの約数になるような場合には，その値を最適セル長としても，あまりかわりないと思われる。

1.3.7.3 実験モデル

(i) 序

前節において，セグメント方式を理論的に解析し，一応の結論を得た。そこで実際の数値を与えて検討を加えてみようと思う。

日本科学技術情報センター（ＪＩＣＳＴ）では，「文献速報」誌の編集テープ作成の際にセグメント方式を使っている。そこで「文献速報」誌から抽出したデータで，簡単な実験モデルを組んで考察する。

なお，このテープは「文献速報年間索引」の編集にも使われる。年間索引は項目索引，著者索引，レポート索引，収録雑誌リストからなっている。

(ii) 実験モデル

1. モデルの概要

このモデルの目的は「文献速報」誌の編集テープをセグメント方式を使って，作成することである。

データは「文献速報」４１年度版電子工学編から任意抽出したものであり，文献の原文標題，著者姓名，ページ，固定ＵＤＣ（Universal decimal classification）の４種である。

次にデータの概略を表で示す。

	総件数	min 範囲 max	標本平均値
原文標題	3,389	5 ～ 251	62.1字
著者姓名	3,389	4 ～ 108	14.1
ページ	3,389	1 ～ 92	6.4
固定ＵＤＣ	11,999	2 ～ 36	9.67

度数分布図を図1.3.17～20に掲げる。

図1.3.21は入力テープフォーマットで，最初の8語が共通項目のためのエリアであり，次のC_i語が情報エリアである。データはそのままレコードとなる（データの長さ≡レコードの長さ）ので，レコードは可変長である。そこで，セグメント方式によって，C_i語に分割して蓄積するものとする。たとえば，レコードがn個に分割されたときには，テープ上では，$n(8+C_i)$語を使って蓄積される。共通項目の原稿ＮＯ，補助原稿ＮＯ，分類コード1，

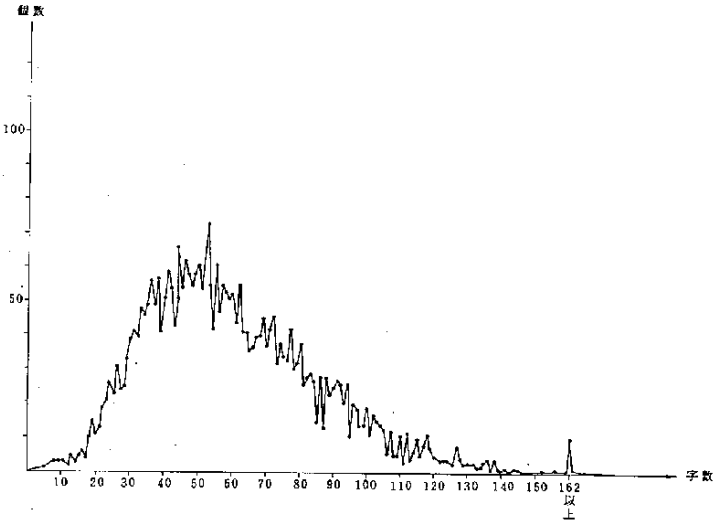


図1.3.17　原文標題・度数分布図

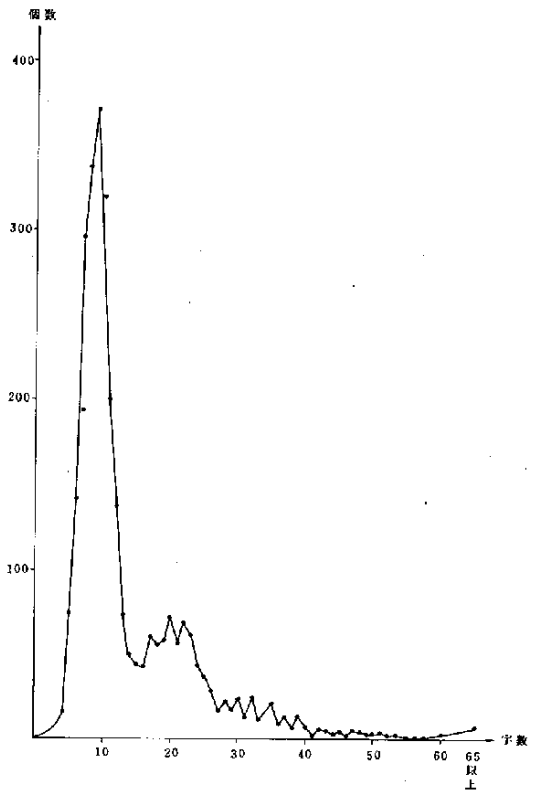


図1.3.18　著者姓名・度数分布図

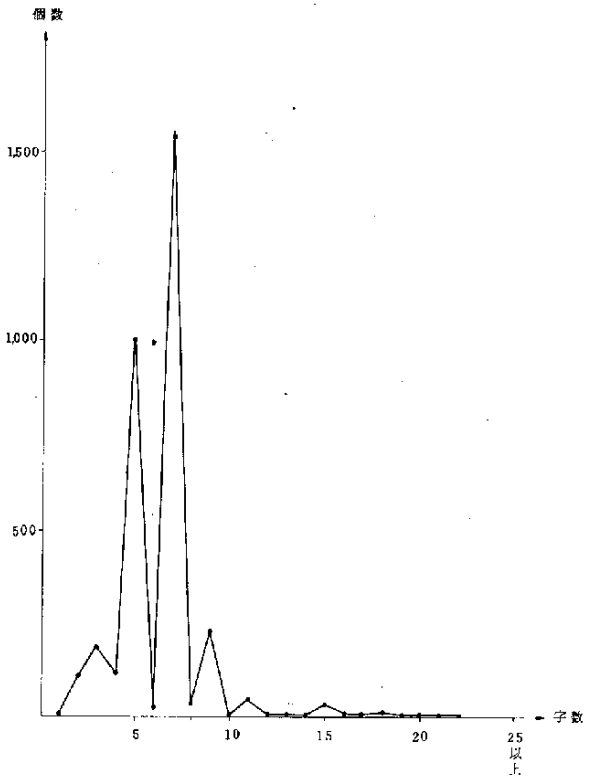


図1.3.19　ページ・度数分布図

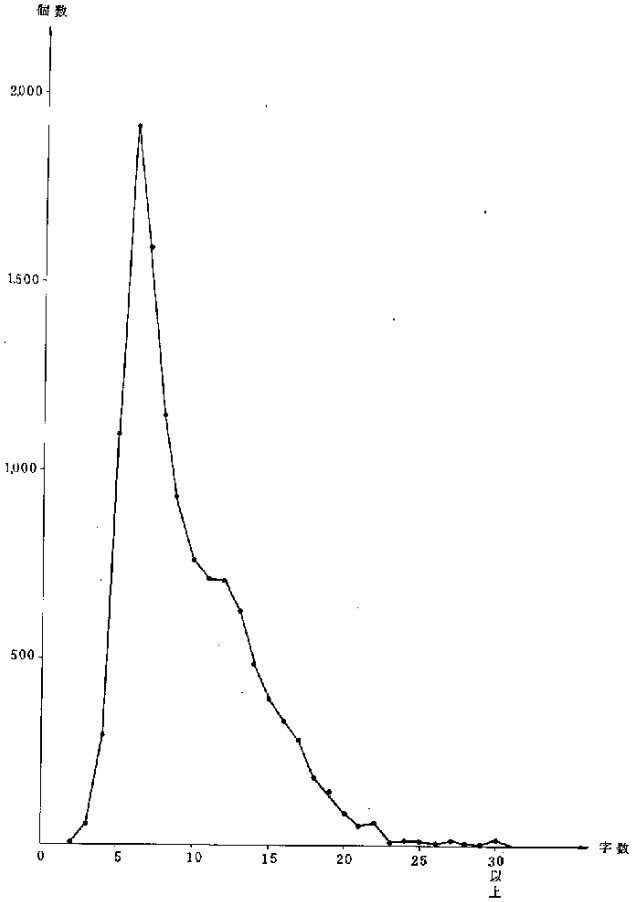


図1.3.20　固定ＵＤＣ・度数分布図

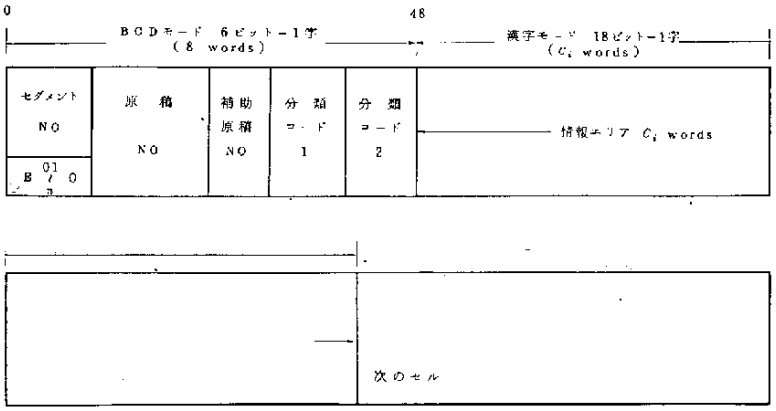


図1.3.21 入力テープフォーマット （1個のセル）

2は，レコードを区別するためのもので，ソートやサーチに使われる。（C_i+8）をセルとすると，各セル間の区別はセグメントNOによって行なわれ，1レコードのいくつかのセルは，通し番号がふられる。

"C_i"を決定するのに理論値と実際の計算値の両方を考えてみよう。そしてその結果を検討しよう。

2. 理 論 値

$b=8$（語）であるから

$$C_i=\sqrt{2bL_i}-\frac{b}{3}$$ として解く。

原文標題（$L_1=62.1$字） $C_1=20$（語）

著者姓名（$L_2=14.1$字） $C_2=8$（語）

ペ ー ジ（$L_3=6.4$字） $C_3=4$（語）

固定UDC（$L_4=4.8$字） $C_4=7$（語）

3. 計 算 値

いま，総レコード数をnとして，総無駄スペース

$$\left(\sum_{i=1}^{n}w_i(c)=\sum_{i=1}^{n}\left[\left\{cn(l_i)-l_i\right\}+bn(l_i)\right]\right)$$

を，Cの値を変えて計算する。

計算結果をグラフによって示す。（縦軸が総無駄スペース，横軸がCの値である。）

原文標題 図1.3.22 著者姓名 図1.3.23

ペ ー ジ 図1.3.24 固定UDC 図1.3.25

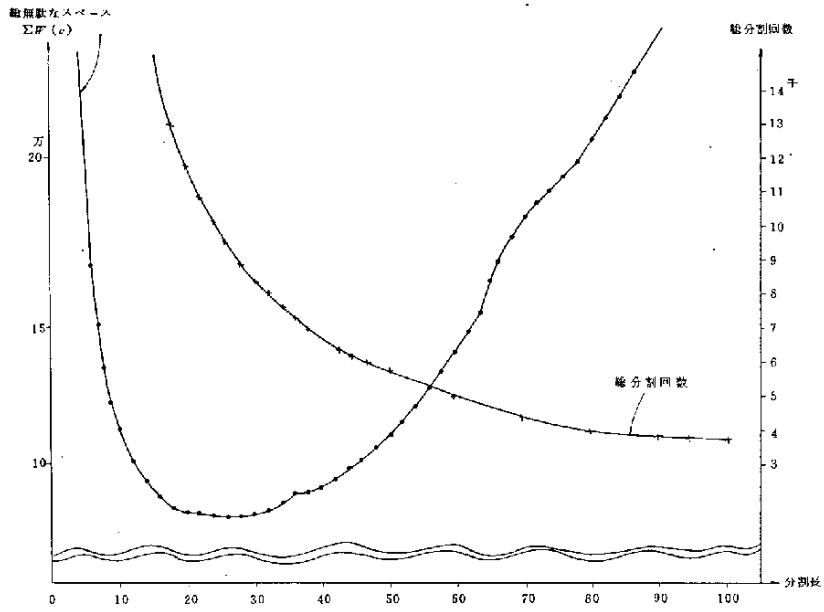


図 1.3.2 2　原文標題・総無駄スペース

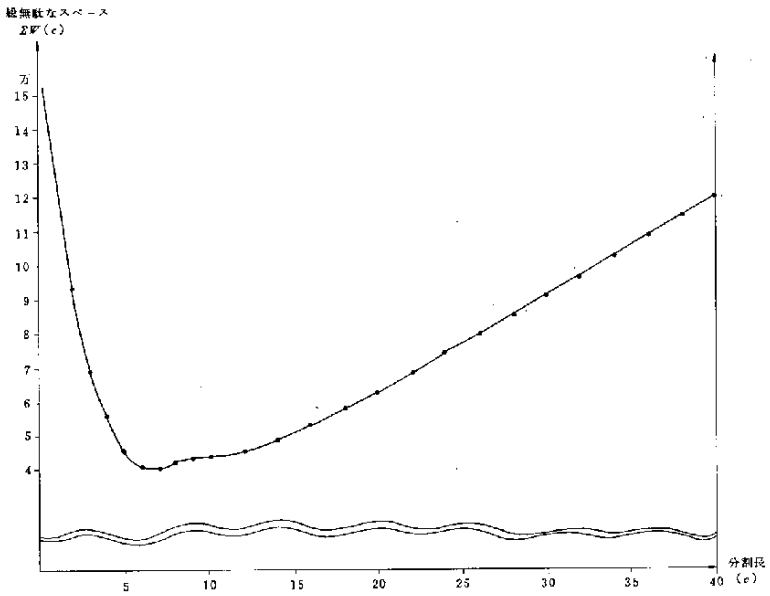


図 1.3.2 3　著 者 姓 名

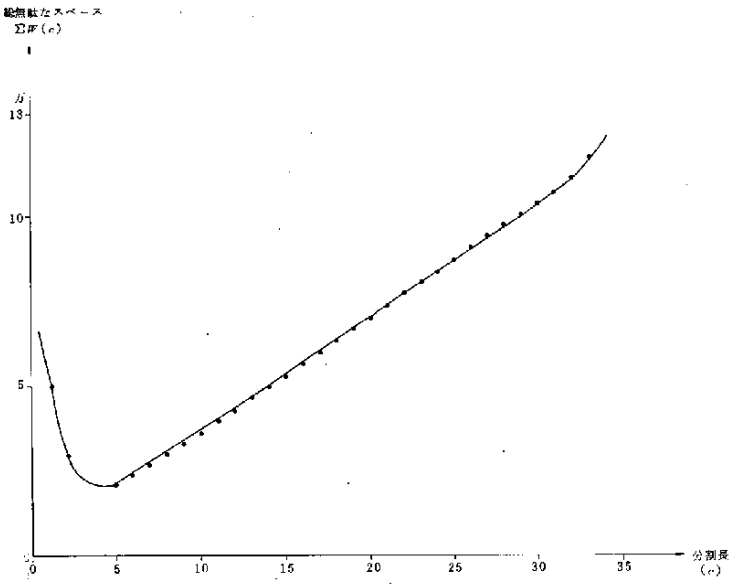


図1.3.24 ペ ー ジ

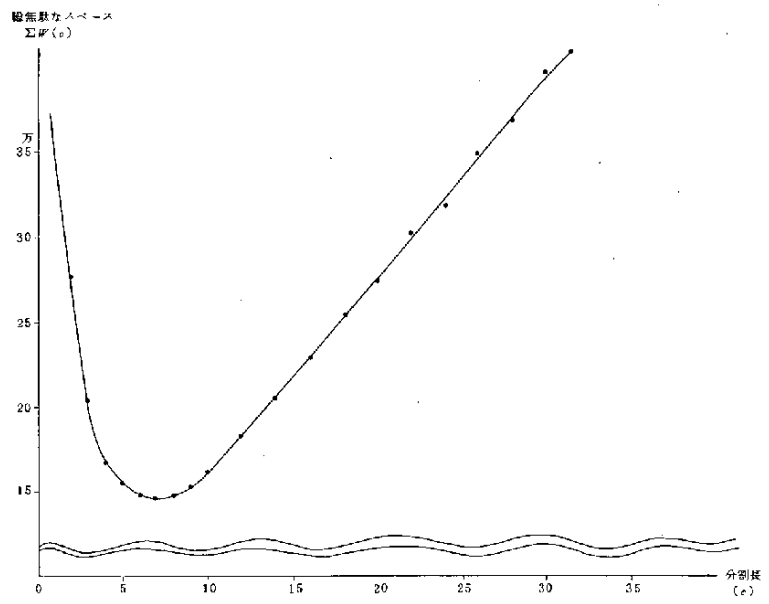


図1.3.25 固定UDC

総無駄スペースを最小にするCの値は各々次のようになっている。

原文標題	$C_1' = 21$	（語）
著者姓名	$C_2' = 7$	（語）
ページ	$C_3' = 4$	（語）
固定UDC	$C_4' = 7$	（語）

理論値とほとんど一致している。

なお，1レコード当りの期待無駄スペースの理論値と計算値を対応させて書くと次のようである。

	理論値（C）	計算値（C'）
原文標題	24語（20）	24.8語（21）
著者姓名	12（8）	12.3（7）
ページ	8（4）	7.9（4）
固定UDC	11（7）	11.8（7）

以上が理論値と計算値の比較である。ここでこの実験モデルをもう少し解析して考察してみよう。

考察—1　（残りの平均スペース）

1.3.7.3において，期待無駄スペース$W(c)$のうち残りの無駄スペース（$=E[n(l)]C-L$）を$\frac{C}{2}$とおくことによって$W(c)$を最小にするcの値を$\sqrt{2bL}$で定義できることを示した。ここで，反対に$C=\sqrt{2bL}$とおいたとき，各レコードの最後のセルの残りの部分がどのような値をとるか，このモデルを使って調べてみよう。

総残りのスペースを

$$N_i = \sum_{j=1}^{m} (n(l_j)C - l_j)$$

と定義すると，計算結果は下のようになった。

原文標題	$N_1 = 36,379$	（$C_1 = 22$）
著者姓名	$N_2 = 17,625$	（$C_2 = 10$）
ページ	$N_3 = 12,266$	（$C_3 = 7$）
固定UDC	$N_4 = 41,219$	（$C_4 = 8$）

したがって，各々の平均残りのスペースと$\frac{C}{2}$の値を対比してみると次のようになる。

	平均残りスペース	$\frac{C}{2}$
原文標題	10.8　語	11　語
著者姓名	5.0	5
ページ	3.6	3.5
固定UDC	3.4	4

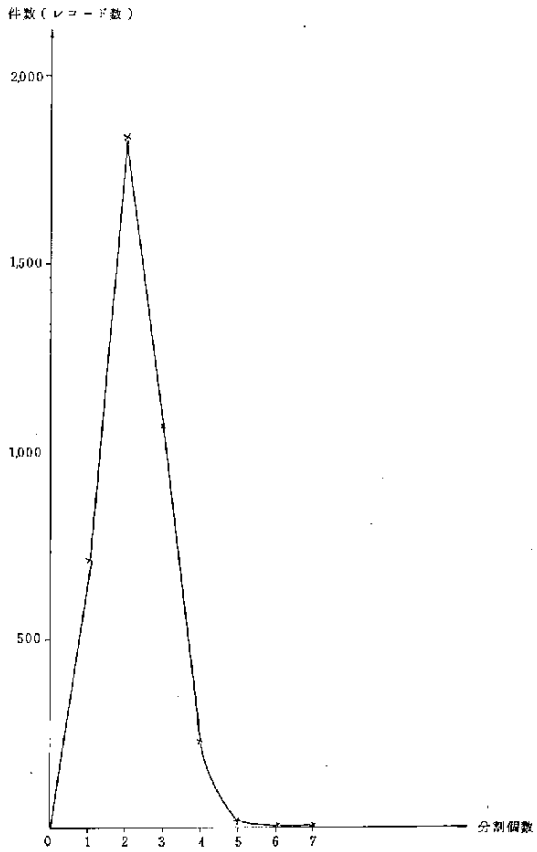


図1.3.26　原文標題・分割個数の分布

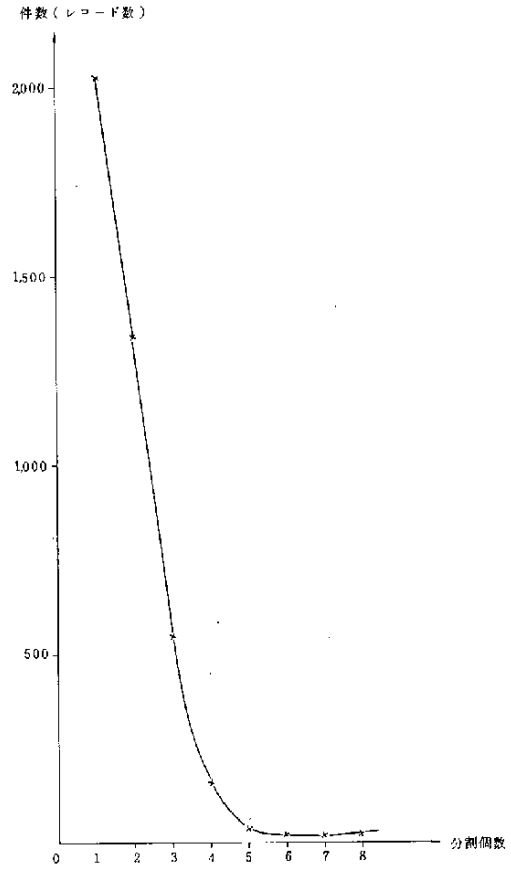


図1.3.27　著者姓名・分割個数の分布

この数値例では，Cを$\sqrt{2bL}$としたとき，各レコードの最後のセルの残りは平均的に$\frac{C}{2}$になる事がいえる。（しかし，反対，つまり残りを$\frac{C}{2}$とおくと，$W(c)$を最小にする値は$C=\sqrt{2bL}$となり，計算値より少しではあるが大きな値となっている。）

また，$C=\sqrt{2bL}$としたときの分割個数とレコードの数との関係は，図1.3.26（原文標題の場合）と図1.3.27（著者姓名）のようになっている。

このときの平均分割個数は，

原文標題　$S_1=2.209$

著者姓名　$S_2=1.22$　　となっている。

理論的には，残りの無駄スペースの期待値は，

$$\{E[n(l)]C-L\}$$

であらわされる。これを$\frac{C}{2}$とおいて

$$E[n(l)]=\frac{L}{C}-\frac{1}{2}$$

としてとくと，

原文標題　$E_1[n(l)]=1.3$

著者姓名　$E_2[n(l)]=0.9$

となる。

したがって，期待無駄スペース$W(c)$のうち最後のセルに残るスペースを平均的に$\frac{C}{2}$とおいて，$W(c)$を最小にするCの値を求めても，最適値に近い値を得ることができるし，おおざっぱな意味で期待セル個数$E[n(l)]$を求めることができることがわかった。

この方法は，多少の誤差を伴うにしても，大変簡単に解析を行なえるのが良い点である。

考察—2　（数種のレコードを同一フォーマットで扱う場合）

実際には，原文標題，著者姓名，ページ，固定ＵＤＣ（本当は，この他に抄録，雑誌略名などが加わる）が全部集ってはじめて1文献の情報となりうるわけであるから，テープ処理の上からも，レポート（雑誌）作成上からも，同一フォーマットで蓄積しておくべきである。つまりテープフォーマットにおける固定情報領域C_iは，項目別ではなく，一意に定めなければならない。

原文標題の1レコード当りの期待無駄スペースをW_1，期待セル数$E[n_1(l)]$，著者姓名のそれをW_2，$E[n_2(l)]$，ページW_3，$E[n_3(l)]$，固定ＵＤＣW_4，$E[n_4(l)]$とする。4つを同一フォーマットで扱ったときの1レコードの期待無駄スペースは，

$$W=\frac{1}{4}\{W_1(c)+W_2(c)+W_3(c)+W_4(c)\}$$

$$=\frac{1}{4}\{(c+b)(E_1+E_2+E_3+E_4)-(L_1+L_2+L_3+L_4)\}$$

ここで，前節の考察で述べた考えを使って，最後のセルに残る無駄スペースを $\frac{C}{2}$ としよう（簡単に解けるから）。

$$\therefore \quad E\left[n_1(\ell) \right] = \frac{L_1}{C} - \frac{1}{2}$$

$$E\left[n_2(\ell) \right] = \frac{L_2}{C} - \frac{1}{2}$$

$$E\left[n_3(\ell) \right] = \frac{L_3}{C} - \frac{1}{2}$$

$$E\left[n_4(\ell) \right] = \frac{L_4}{C} - \frac{1}{2}$$

とかけるので，Wは次のようになる。

$$W = \left\{ \frac{C}{2} + \frac{b}{4} \left(\frac{L_1+L_2+L_3+L_4}{C} \right) \right\}$$

したがって，$W'=0$ とおくと，

$$C = \sqrt{2b \frac{(L_1+L_2+L_3+L_4)}{4}}$$

したがって，理論的には，

$$C = \sqrt{2\times 8\times \frac{(31+7+3.2+4.8)}{4}} \fallingdotseq 13 \quad (\text{語}).$$

となる。ここで，実際に4つの総無駄スペースとセルの長さCとの関係をグラフに示すと図1.3.28のようになっている。実際の値は，図1.3.28より$C=10$（語）となっているが，しかし，Cが8語～14語の範囲では，総無駄スペース間にはあまり相違がみられない。また各項目別に求めたC_1，C_2，C_3，C_4 を使ってその平均をとってみると

$$\overline{C} = \frac{1}{4}(C_1+C_2+C_3+C_4)$$

$$= \frac{1}{4}(21+8+4+7)$$

$$= 10$$

となって，実際値と一致している。

一般的に，幾つかの異なる項目を同一フォーマットで蓄積しなければならないときには，最適セル長Cは次の式で与えられると考えてよいと思われる。

$$C = \sqrt{2b}\times \frac{(L_1+L_2+\cdots\cdots+L_n)}{n} \qquad (\text{ここで}n\text{は項目数})$$

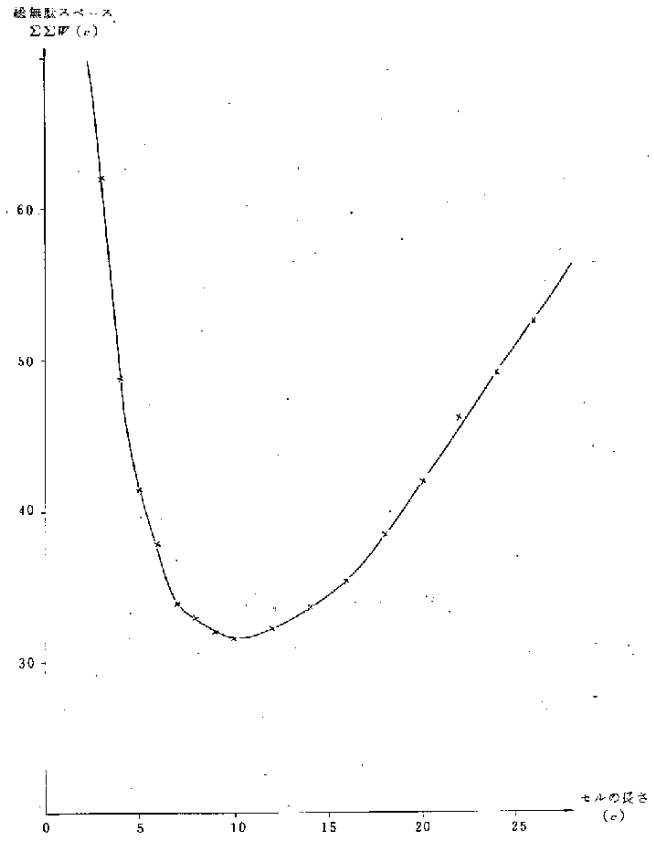


図1.3.28　4項目統計無駄スペース

考察—3　（初期条件がある場合）

蓄積される内容によっては，"どうしてもこれ以下には分けてほしくない"という要求があるかもしれない。

たとえば，ＪＩＣＳＴでは，固定情報領域として２０語（４０characters）とっている。これは，この領域に蓄積する内容が，どうしても４０字分連なっている必要がある場合があるためである（つまり，初期条件として"分割長は２０語以上でなければならない"が与えられたことになる）。

このように，蓄積される内容によって分割長に制限がある場合はどうしたらよいか。

ここで，図1.3.22～25をみていただくとわかるように，無駄なスペースと分割長さの関係は，Cの最適値以上のCの値に対して，線型になっている（つまり，最適値よりおおきいCの値に対して，無駄なスペースは，線型的に増加する）。

したがって，初期条件として与えられた制限長が最適値Cより小さいときは，そのままCを最

適値にとればいいが，大きいときには，制限長を最適値とすればよい。その時，確めるまでもなく，最小無駄スペースとなる（ただし制限付）。

考察―4　（時間値）

これまでは，メモリースペースの軽減だけを考えて，時間に関しては何の考察も加えずにきた。しかし，最適化のパラメーターは容量と時間であり，どちらか一方の最適化をはかっても全体として最適である保証はない。一般には，この両者は排反的な関係にあり，一方の軽減をはかると他方が増大してしまう。最適化は両者のかねあいを考えて行なわれるのが普通である。つまり，なるべく容量を少なく，なるべく時間を速くということになる。

セグメント方式によって容量の軽減をはかることができたが，時間はどうなっているであろう。

そこで，簡単な検索ロジックを考えて，その時間値を調べてみよう。

セグメント方式で編集されたテープと，レコードの最大長（容量が前者よりも大きい）で編集されたテープがある。この2つのテープ上のレコードの検索時間を比較しよう。セグメント方式のレコードは，その長さにおうじていくつかのセル（＝サブレコード）からなっている。

各レコードは原稿ＮＯ（キィー）とデータ部からなっているものとする（サブレコードも同様であるが，データ部の長さが最大長の場合よりも当然短かい）。

検索は原稿ＮＯで行なうものとする。

レコードを読み込んできて，検索し，目的のレコードを印字するまでのフローチャートを次に示す。ここでWはワーク・エリア（work area）を示すものとする。セグメント方式の場合には，ここも分割されている。

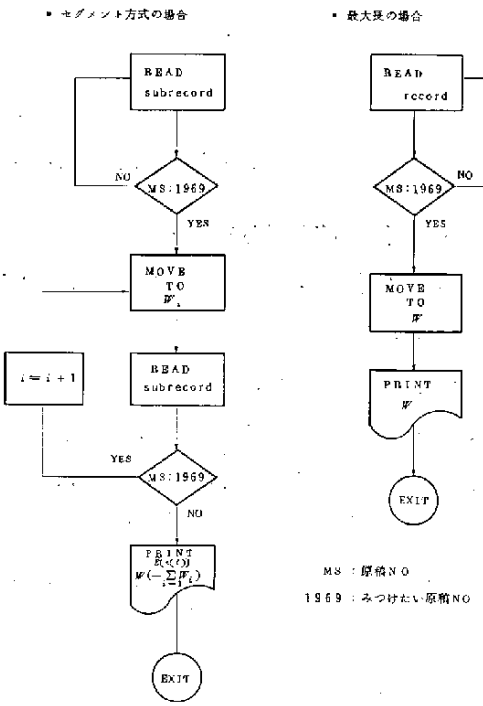


図 1.3.2 9　目的レコードの検出手順

平均検索時間は，次のように書くことができる。

＊セグメント方式の場合

（1レコードは平均的に，$E[n(\ell)]$ のサブレコードからなるとする）

$$T_S = T_{R_1} + T_{P_1}$$

＊最大長の場合

$$T_M = T_{R_2} + T_{P_2}$$

ここで

T_{R_1}，T_{R_2}＝検索時間

T_{P_1}，T_{P_2}＝処理時間

とする。（印刷時間は除いて考えた）

検索時間は，テープからの読込時間と，比較時間からなり，処理時間は，ワーク・エリアへの移動の時間である。

読込回数の軽減はブロッキング（Blocking）によっておこなわれる。（読込み開始に先立ってテープを始動するのに要する時間と，データ転送終了後テープを停止するのに用する時間がどうしてもかかってしまうので，1論理レコード当りの読込時間を短かくするには，ブロック形成を大きくとればよい。）

ここで，セグメント方式による読込み時間と最大長の読込み時間を比較するために，ブロックの大きさ（B）を等しくして考える。セグメント方式の場合の平均レコード長は，$(c+b)E[n(\ell)]$ であり，最大長の場合 ℓ'（レコードの最大長）である。

1ブロック内のレコードの数（ブロック形成率）は，それぞれ

$$f_1 = \frac{B}{(c+b)E[n(\ell)]}\ ,\quad f_2 = \frac{B}{\ell'}$$

と書ける。

全部のレコードを蓄積するのにそれぞれ N_1 個，N_2 個のブロックを使うものとすると，1レコード当りの平均読込時間は，

$$t_{r1} = \frac{\frac{N_1(N_1+1)}{2N_1}}{(c+b)E[n(\ell)]}(t_1+t_2)$$

$$= \frac{(N_1+1)}{2f_1}(t_1+t_2)$$

$$t_{r2} = \frac{(N_2+1)}{2f_2}(t_1+t_2)$$

ここでは，t_1 は1語当りの転送時間

t_2 は読込・書出時間

である。

t_c を比較時間，t_m を移動時間とすると

$$T_S = \frac{(N_1+1)}{2f_1}(t_1+t_2)+\frac{N_1+1}{2}t_c+E[n(\ell)]\times t_m$$

$$T_M = \frac{(N_2+1)}{2f_2}(t_1+t_2)+\frac{N_2+1}{2}t_c+t_m$$

＊＊数値例（原文標題の場合）＊＊

$(c+b)E[n(\ell)=(42+48)\times 2=180$字

ℓ' $=251$字

いま，$B=45180$とすると，

$$f_1=\frac{B}{(c+b)E[n(\ell)]}=251 \quad , \quad f_2=\frac{B}{\ell'}=180$$

$$N_1=\frac{3389}{251}-14 \quad , \quad N_2=\frac{3389}{180}-19$$

$$\therefore\ T_S=\frac{(14+1)}{251\times 2}(0.01+7.8)+\frac{(14+1)}{2}\times 25.5+2\times 24.0$$

$$=473.55\mu\text{sec}$$

$$T_M=\frac{(19+1)}{180\times 2}(0.01+7.8)+\frac{(19+1)}{2}\times 25.5+24.0$$

$$=716.4\mu\text{sec}$$

したがって，$T_S<T_M$

セグメント方式を使うことによって内部処理は複雑になるが，スペースが軽減されたことによって，目的のレコードを得るのに要する読込み時間が最大長の場合よりも，ずっと短かくてすむ。これが，もっと多量のレコードを扱う場合には，読込み時間の軽減は大きな利得になるであろう。時間に関しても保証がえられる。

1.3.7.4 結　論

可変長レコードを取扱う時に，蓄積する領域をどのように構成するかが大きな問題となる。そこで『セグメント方式』を提案した。

『セグメント方式』とは，データ領域をあらかじめC語のセルに分割しておいて，データの大きさに応じてこのセルをリスト構造によって結んで蓄積する方法である。そして，メモリースペースの利用度の面から最も効率の高い最適なセルの大きさ（C）を1.3.7.2において理論的に求め，どのような分布に関してもCは$\sqrt{2bL}$に近づくという結果をえた。1.3.7.3において，実際に数値を与えて，理論値の裏付けを行なった。また，実際問題としておこることにも考察を加えた。

N個からなる可変長レコードをセグメント方式で蓄積するときの手順をフローチャートで示す。

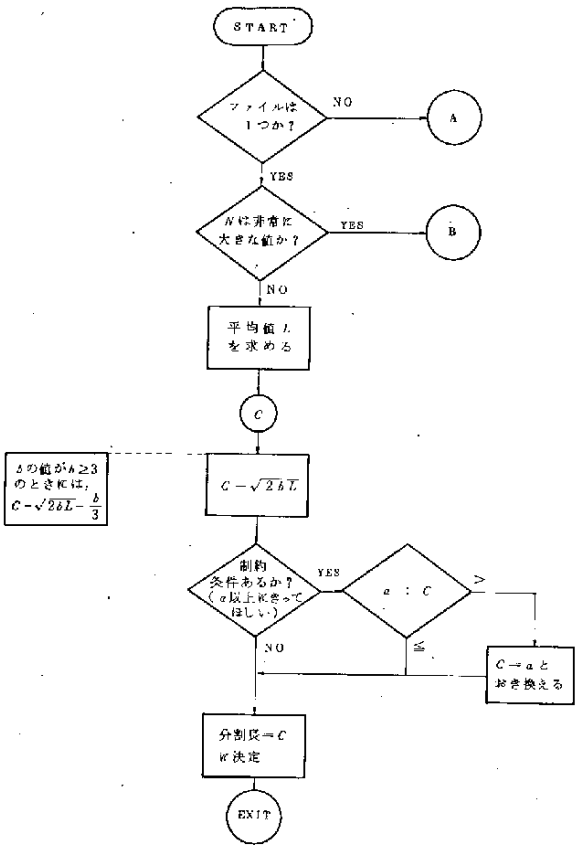
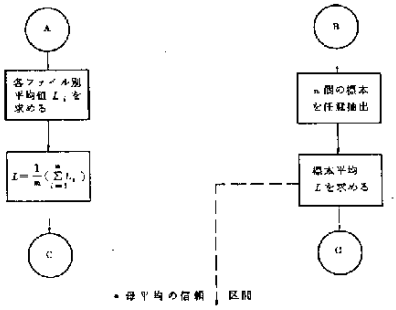


図1.3.30　セグメント方式による蓄積手順



母集団を正規母集団$N(\mu, \sigma^2)$とする。

(σも，μも未知である。)

大きさnの任意標本平均をLとすると，

不偏分散の平方根は，

$$\hat{\sigma}=\sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}(l_i-L)^2}$$

ここで

$$t=\frac{L-\mu}{\hat{\sigma}/\sqrt{n}}$$

は，自由度$(n-1)$のt分布に従う。

自由度$(n-1)$のt分布に従う確率変数tに対して

$$P_r\{|t|\leqq t(n-1,\ 0.05)\}=0.95$$

$$\therefore\quad P_r\{|t|>t(n-1,\ 0.05)\}=0.05$$

となるから，$t(n-1,\ 0.05)$をきめると，

μの95%信頼区間は，

$$\left[L-t(n-1,\ 0.05)\frac{\sigma}{\sqrt{n}},\ L+t(n-1,\ 0.05)\frac{\sigma}{\sqrt{n}}\right]$$

図 1.3.30 (つづき)

参考文献と注

1) 太田文平，黎明期を迎えた情報産業の将来。ビジネス43.12月号：20—24。

2) 化学工学協会編，物性定数第3集，丸善1965。

3) 古小路四朗　コンピューターによる人事管理，日本経営出版会，昭43，p. 84—85。

4) Lesk, Michael E. Performance of automatic information systems. Inf. Stor. Retr. 4, 201—218 (1968)

5) SMARTシステムの概要は次の文献にも詳細な説明がある。

昆野誠司，菊池敏典，情報検索システムの具体例，情報処理7，327—336(1966)

SMARTシステムのファイル規模，プログラム，ハードウエア，人員，将来構想などについては本書第3章で扱う。

6) たとえば，

Resnick, A. Relative effectiveness of document title and abstracts for determining relevance of documents. Science 134, 1004—1005 (1961)

7) たとえば，

Hargerty, Katherine Abstracts as a basis for relevant judgement. master thesis. Univ. Chicago. Graduate Library School. 1967. 36p. (Grant NSF—GN—380. PB174·394)

8) このような方法は種々の人によって示唆されている。たとえば，

Luhn, H. P. Auto-encoding of documents for information retrieval systems. modern trends in documentation, ed. by M. Boaz. Pergamon Press, 1959. Doyle, Lauren B. Indexing and abstracting by association, Amer. Doc. 13, 378—390 (1962)。

9) Salton, G. A flexible automatic system for the organization, storage, and retrieval of language data (SMART) Harvard Univ. Comp. Labo. ISR—No. 5. 1964.

10) 中井浩 構文分析法：Kuno-Oettinger's multiple-path syntactic Analyser. 情報管理11, 464—469；664—676 (1969)

11) Sussenguth, Jr. E. H. Structure matching in information processing Harvard Univ. Comp. Labo ISR-No. 6. 1964.

12) たとえば，

Tritchler, R. J. Effective information-searching strategies without "perfect"indexing. Amer. Doc. 15, 179—184 (1964)

13) Treu, Siegfried は当時 Goodyear Aerospace Corporation, Abron, Ohio の所属である。

14) Treu, S. The browser's retrieval game. Amer. Doc. 19, 404—410 (1968)

15) Goodyear Aerospace Corporation. Text based information retrieval. Internal report GER—11907 Akron, Ohio, 1965.

16) Armitage, J. E. and Lynch, M. F. Some structural characteristics of orticulated subject indexes. Inf. Stor. Retr. 4, 101—111(1968)

17) Armitage, J. E. and Lynch, M. F. Articulation in the generation of subject indexes by computer. J. Chem. Doc. 7, 170—178(1967)

18) Weisenbaum, J. Symmetric list processor Com. ACM. 6, 524—544(1963)

19) Russell, D. B. On the implementation of SLIP. Com. ACM. 8, 263— (1965)

20) Kimber, R. T. Computer applications in the fields of library houskeeping and information processing. Program 6, 5—25(1967)

21) Freeman, R. R. The management of a classification. J. Doc. 23, 304—320(1967)

22) Soergel, Dagobert. Mathematical analysis of documentation systems: an attempt to a theory of classification and search request formulation. Inf. Stor. Retr. 3, 129—173(1968)

23) Weaver, George. Analysis of chemical notation project. 2. a mathematical model for chemical cipher systems. 1. Univ. of Pennsylvania, 1968. iii, 31p.

24) Uhlmann, Wolfram. Document specification and search strategy using basic intersections and the probability measure of sets. Amer. Doc. 19, 240—246(1968)

25) 菊池, 笹森, 高橋, 情報内容の処理, 情報処理7, 303—317(1966)

26) Doyle, Lauren B. Indexing and abstracting by association. Am Doc. 13, 378-390(1962)

27) ① 著者: Eric Wolman;
② 表題: a Fixed Optimum cell—size for Records of Various Lengths,
③ 出典: Journal of ACM, vol12 №1 p. 53 JAN. 1965

28) ① 著者: IBM Sales and System Guide,
② 出典: File Organization Techniques for Direct Access

Storage Device

29) ① 著者：Ivan Flores,

② 表題：Computer Time for Address Calculation Sorting

③ 出典：Journal of ACM, vol7, 1960

30) ① 著者：Werner Buchholz

② 表題：File Organization and Addressing

③ 出典：IBM Systems Journal, p. 86, Jun. 1963

31) ① 著者：W. P. Heising

② 表題：Note on Random Addressing Techniques

③ 出典：IBM Systems Journal, p. 112 Jun. 1963

32) ① 著者：淵　一博

② 表題："ファイルコントロール システム" プログラム技術

③ 電子通信学会, 1967

第 2 章　検索システム—(I)

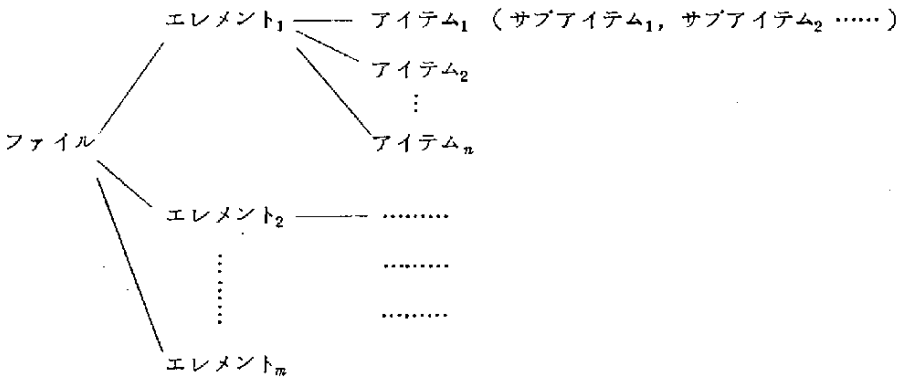
現実のＩＲシステムは，第3章にみるように，特定のハードウエア，ソフトウエアを使用してシステム化されている。一方，すでに発表された汎用ＩＲシステムとしては，ＣＤＣのＩＮＦＯＬ，ＩＢＭのＧＩＳ，ＡＩＳなどがある。以下にその内容を説明する。

2.1 ＩＮＦＯＬ

2.1.1 システム概説

ＩＮＦＯＬ[1]（INFormation Oriented Language）はＣＤＣ３６００／３８００用に開発された汎用情報検索システムである。

ＩＮＦＯＬで扱う最大の情報単位はファイルである。一つのファイルはいくつかのエレメントからなる。そして各エレメントはデータアイテムの同型のリストからなる。



たとえば，従業員に関するファイルでは，個人，個人がエレメントであり，個人の名前，年令，給料などがアイテムである。アイテムは一価または多価になりうる。多価の場合は，そのアイテムをマルティプル・アイテムという。文献ファイルにおける著者アイテムは，その例である。そして一つ一つの値をサブアイテムという。

ＩＮＦＯＬのデータファイルはテープであり，検索はシークエンシャルサーチである。

ＩＮＦＯＬシステムを稼動させるには次のようなハードウェアおよびソフトウェアが必要である。

(1) ハードウェア

本　体　　ＣＤＣ３６００／３８００

主記憶装置　　３２０００語

1語　　４８ビット

カードリーダー

ラインプリンター

磁気テープ装置

磁気ドラム装置

(2) ソフトウェア

SCOPE オペレーティングシステム（テープまたはドラム）

(3) テープの構成

SCOPE オペレーティングシステムがテープの場合とドラムの場合とで，テープの構成が異なる。SCOPE に必要なものを除いて，

テープ SCOPE の場合……最大５本

ドラム SCOPE の場合……最大２本

のテープが必要である。

2.1.2 機 能 概 説

INFOL は６つのフェイズから成立している。各フェイズはつぎのようなコントロールワードを持つ。

(1) ESTABLISHMENT

(2) INTERROGATION

(3) UPDATE

(4) REVISION

(5) BOOKKEEPING

(6) VALIDATION

(1) ESTABLISHMENT

ファイルを作るためのフェイズである。つぎの２種類の情報を与えて，ファイルを作成する。

① ファイルを記述するための情報

② ファイルに挿入するデータ

(2) INTERROGATION

このフェイズは，つぎの２つの部分からなる，

① 検索基準を与えて検索すること。

② 検索されたデータを編集して出力すること。

(3) UPDATE

ファイルの中のデータの変更を行なうフェイズである。つぎの２つの場合がある。

① discrete updates

限られたエレメントの変更，除去，挿入が行なわれる。

② selective updates

アイテムの値が selective criteria を満足するエレメントの変更，除去，挿入が行なわれる。

(4) REVISION

データを記述する情報の変更を行なうフェイズである。エレメントにアイテムをつけ加えるというような，ファイルの構造の変更を行なう。

(5) BOOKKEEPING

ファイルの中のデータの統計的な値を計算して出力する。すなわち，アイテムの最大字数とか，サブアイテムの最大数とかいうものである。これらの値は，出力形式を決めるのに役立つ。

(6) VALIDATION

ファイルの中の情報が，一定の条件を満足するか否かを調べる。ESTABLISHMENT フェイズや，UPDATE フェイズでは，このフェイズは自動的に行なわれる。REVISION フェイズでは validation criteria を変更したあとでのみ，このフェイズが使われる。

以上6つのフェイズのうち，ESTABLISHMENT フェイズと REVISION フェイズは，他のフェイズとともに実行することはできない。他の4つのフェイズは，どのような組合せでも，実行可能である。そのときの実行の順序は以下の如くである。

1. UPDATE
2. VALIDATION
3. BOOKKEEPING
4. INTERROGATION

以下に並べるのは INFOL で使われるコントロールワードの一覧表である。

```
ESTABLISHMENT
        ITEM DESCRIPTIONS
        CATEGORY－TYPE
        CODES
        VALIDATIONS
        INITIAL INPUT
INTERROGATION
        RETRIEVAL CRITERIA
        EXTRACTIONS
UPDATES
        ITEMIZE
        DISCRETE UPDATES
        SELECTIVE CRITERIA
```

SELECTIVE UPDATES

BOOKKEEPING

REVISION

VALIDATION

一段目のコントロールワードを，トップレベルのコントロールワードといい，二段目のコントロールワードをセカンドレベルのコントロールワードという。

2.1.3 機能各論

各フェイズについて，もう少し詳しく述べよう。

2.1.3.1 ESTABLISHMENT フェイズ

ファイルを作るためのフェイズである。ファイルを記述するための情報は，つぎの4つのセカンドレベルのコントロールワードではじめる。

ITEM DESCRIPTIONS
CATEGORY—TYPE
CODES
VALIDATIONS

ファイルに挿入するデータは，セカンドレベルのコントロールワード

INITIAL INPUT

ではじめる。

ファイルの作成，変更にともなって，ファイル中の各アイテムに対する日付，ENTRY DATE が自動的に蓄積される。

(1) ITEM DESCRIPTIONS

アイテムの名前を定義し，各名前に番号を与える。

例） ITEM DESCRIPTIONS 8
EMPLOYEE NUMBER *1* NAME *2* START DATE *3*
MONTHLY SALARIES *4* SALALY MONTHS *5*
PERFORMANCE REVIEWS *6* REVIEW MONTHS *7*
DEPENDENT STATUS *8*

以後アイテムを参照するときは，この番号を使う。

(2) CATEGORY－TYPE

各アイテムに関して，以下8つのうちの1つを指定する。

UNARY ALPHANUMERIC
UNARY NUMERIC
UNARY DATE

UNARY CODED

MULTIPLE ALPHANUMERIC

MULTIPLE NUMERIC

MULTIPLE DATE

MULTIPLE CODED

・UNARY はそのアイテムが一価であることを示し，MULTIPLE は多価であることを示す。

・ALPHANUMERIC は＊と）と（を除く文字列である。

・NUMERIC は数値データである。整数部，小数部または両方含んでもよい。符号と小数点を除いて１０桁まで扱える。

・DATE は日付を表わすアイテムである。以下の表現が扱えるが，内部表現はすべて yymmdd と変換される。

外部表現	内部表現
3 Sept. 64	64 09 03
7 August 1965	65 08 07
Apr. 18, 1965	65 04 18
1938	38 00 00
JAN. 1942	42 01 00
Feb. 1951	51 02 00
5−18−63	63 05 18
6／20／63	63 06 20
7. 21. 65	65 07 21
66 09 25	66 09 25
TODAY	(Current date)

・CODED は，そのアイテムの値をコード化することを意味する。コードにはナンバーとニモニクの２種類がある。ナンバーのときはシステムが異なる値が現われる順に１，２，…と番号をつける。ニモニクの場合は，ことなるすべての値に対してコードを定義する。１つのアイテムに対して許されるコードの数は５１１までである。

(3) CODES

ニモニクコードの定義をする。

(4) VALIDATIONS

アイテムやサブアイテムがファイルに挿入されるまえに満足すべき条件を指定する。つぎのような条件がある。

① NECESSARY

アイテムが必ず存在すること。

② MAXIMUM m

マルティプルアイテムのサブアイテムの最大数。

③ RANGE

数値データ，日付データの満足すべき区間。

④ CHARACTERS X

字数制限　最大4,096字。

⑤ NON－NUMERIC

アイテムに0～9の数字を含まないこと。

⑥ ALPHABETIC

アイテムの値がA～Zとピリオドとコンマからなること。

⑦ INTEGER

整数データであること。

以上の条件に合わないときは，そのアイテムまたは，そのエレメント全体が拒否される。

例）

4.7. PUNCHING EXAMPLES

```
    ITEM DESCRIPTIONS 21
  SUBPROGRAM NUMBER *1* SUBPROCRAM NAME *2* PURPOSE *3* LEVEL *4*
  PROGRAMMER *5* INPUT PARAMETERS *6* INPUT PARAMETER CLASSES *7*
  OUTPUT PARAMETERS *8* OUTPUT PARAMETER CLASSES *9* CELLS DESTROYED
  *10* CELLS-DESTROYED CLASSES *11* CONSTANTS USED *12* CONSTANT CLASSES
   *13* SUBPROGRAMS CALLED BY THIS SUBPROGRAM *14* SUBPROGRAMS WHICH CALL
THIS SUBPROGRAM *15* INDEX REGISTERS USED *16* LOCATIONS REQUIRED FOR SUBPROGRAM
*17* ERRORS CHECKED FOR *18* START AND COMPLETION DATES *19*
  PHASES USED IN *20* ABSTRACT *21*
```

a. Item Descriptions （このとき次のitem numbersの後ろにdataがくる）

```
    CATEGORY-TYPE
  *1* UNARY NUMERIC *2* UNARY ALPHANUMERIC *3* UNARY ALPHANUMERIC *4*
  UNARY NUMERIC *5* UNARY CODED
                *6* MULTIPLE ALPHANUMERIC ASSOCIATION L *7* MULTIPLE NUME
RIC ASSOCIATION 1 *8* MULTIPLE ALPHANUMERIC ASSOCIATION 2 *9* MULTIPLE NUMERIC A
SSOCIATION 2 *10* MULTIPLE ALPHANUMERIC ASSOCIATION 3 *11* MULTIPLE NUMERIC
ASSOCIATION 3 *12* MULTIPLE ALPHANUMERIC ASSOCIATION 4
   *13* MULTIPLE NUMERIC ASSOCIATION 4 *14* MULTIPLE ALPHANUMERIC *15* MU
LTIPLE ALPHANUMERIC *16* MULTIPLE NUMERIC *17* UNARY NUMERIC *18* MULTIPLE ALPHA
NUMERIC
```

b. Category-Type

```
  CODES
*5* MNEMONIC TWO T. W. OLLE * MPO M. P. O'TOOLE * JWE J. W. EUSEBIO *
JRM J. R. MARSHECK * BLV R. L. VENEZKY * EHM E. H. MUELLER *
PLM P. L. MCNATR * WPC W. P. CEAGLIO *
*20* NUMBER FILE ESTABLISHMENT * SPECIFICATIONS * FILE PASS * OUTPUT *
FILE REVISION * DISCRETE UPDATE MERGE **
```

c. Codes （最後のcodeのあとの*に注意）

```
  VALIDATIONS
*1* NECESSARY *2* CHARACTERS 8 NECESSARY *3* NECESSARY *4* INTEGER
*5* NECESSARY *7* )NTEGER *9* INTEGER RANGE 1 6 *11* INTEGER
*13* INTECER *16* MAXIMUM 6 *17* INTECER NECESSARY *19* RANGE
640800 660100
```

d. Validations

```
1A *1* 1 *2* PEELOFF *3* PEEL BLANKS FROM ENDS OF A STRING.  *4* 1 *5*
1B TWO *6* POSI * LOC1 * POS2 * LOC2 * *7* 3 * 3 * 3 * 3 * *8* REALPOSI
1C * REALOC1 * REALPOS2 * REALOC2 * NOCHMINI * *9* 3 * 3 * 3 * 3 * 3 * *12*
1D SIX * INT48 * *13* 2 * 2 * *15* SCANBA * PICKINFO * *16* 1 * 2 * 3 *
1E *17* 41 *18* NO NON-BLANK CHARACTER IN STRING.* *19* 640810 * 640903* *
1F 20* 1 * 2 * *21* PROGRAM TREATS FIRST LEFT THEN RIGHT END OF STRING. SCA
1G NS UNTIL A NON-BLANK CHARACTER OR THE OTHER END OF THE STRING IS FOUND. THE
1H END CHARACTERS OF THE INPUT STRING ARE NOT TESTED. PEELOFF SETS NOCHMINI T
1I O -1 IF NO NON-BLANK CHARACTER IS FOUND. *
```

e. First Element in Initial Input （最後のitemのあとの*に注意）

2.1.3.2 INTERROGATION フェイズ

このフェイズは，検索基準を与えて検索を行ない，検索基準を満足するものを抜粋するという2つの機能を持つ。その2つの機能はつぎのセカントレベルのコントロールワードで指示する。

RETRIEVAL CRITERIA

EXTRACTIONS

(1) RETRIEVAL CRITERIA

（1－1） 検索基準を表わす用語

① ITEM SUB－CRITERION

・Existence Sub-Criterion

EXIST……アイテムの値がエレメント中に存在すること。

DOES NOT EXIST……アイテムの値がエレメント中に存在しないこと。

・Relational Sub-criterion

equal, not equal, greater than, less than, greater or equal, less or equal の6つの条件

② ITEM CRITERION

1つのアイテムに課せられる item sub-criterion の集合。AND, OR, カッコを用いて item sub-criteria を結びつけたもの。

③ MULTILEVEL SUB－CRITERION

DEFINE ステートメントによって，単一の sub-criterion を簡略化した表現に置き換えたもの。この簡略化した表現を identifier という。

④ MULTILEVEL CRITERION

いくつかの multilevel sub-criterion に対する identifier を AND や OR で結びつけたもの。相異なるアイテムに対する sub-criteria を結びつけるのに，OR を用いるというのが multilevel criterion の特徴である。

普通の item criteria は，暗黙のうちに AND で結ばれている。すなわち，数個のアイテムに対する item criteria は，すべてが満足したときのみエレメント全体として満足する。

⑤ RETRIEVAL CRITERION

1つのエレメントに関する item criterion の集合。multilevel criterion は，1つだけ許されている。

（1－2） QUALIFIERS

アイテム中のどの値に検索基準を適用するのかを示すものを qualifier という。

(1) QUALIFIERS ON UNARY ITEMS

1個のアイテムに対する qualifier は item value か，または ENTRY DATE のどちらかである。item value qualifier は実際には書かない。

(2) QUALIFIERS ON MULTIPLE ITEMS

多価のアイテムに対する qualifier には，explicit, derived, group がある。

① explicit qualifier

FIRST …… 最初のサブアイテム

LAST …… 最後のサブアイテム

SUB－ITEM n …… n番目のサブアイテム

LAST BUT n …… 最後からn＋1番目のサブアイテム

ENTRY DATE …… アイテムを作成または変更した日付

② derived qualifier

TOTAL …… サブアイテムの数

MINIMUM …… 数値または日付サブアイテムの最小値

MAXIMUM …… 数値または日付サブアイテムの最大値

SUM …… 数値サブアイテムの和

MEAN …… 数値サブアイテムの平均

③ group qualifier

ALL …… すべてのサブアイテムが同一の relational sub－criterion を満足すること。

ANY n …… 利用者の与える値の集合と，サブアイテムの集合の照合。nは最小限n個の共通値を持たなければならないことを示す。eqnal と not eqnal だけが使える。

any sub－item …… qualifier を書かない。その意味は ANY 1と同じ。

例） RETRIEVAL CRITERIA

21 DOES NOT EXIST

* 7 * ENTRY DATE LE 1－12－66

12 ALL GT 12 Feb.61

17 EQ * PROGRAMMING LANGUAGE *

* 4 * GT 6 AND LT 12

* 5 * TOTAL GE 12 OR SUM LE 175

* 6 * EXIST OR ENTRY DATE GT SEPT. 1964

* 7 * (GT 6 AND LT 12) OR EQ 15

```
*10*  DEFINE (EQ 5) ID1
*11*  DEFINE (EQ *ABC*) ID2
MULTILEVEL      ID1   OR   ID2
```

(2) EXTRACTIONS

検索した情報の出力を指定する。つぎの4つの出力形式がある。

Automatic format selector

Optional format selector

Automatic report generator

complete report generator

report generator は，選ばれたアイテムを印刷するときに用いられる。format selector は，FORTRAN や COBOL プログラムに対する入力のテープファイルを作るときに使う。

automatic 手法が選ばれたときは，抜粋されたアイテムはシステム中に内蔵された標準フォーマットにしたがって出力きれる。

（2−1） AUTOMATIC FORMAT SELECTOR

① Necessary Word

EXTRACT …… automatic format selector であることを示す。アイテム全体を抜粋する。

② Optional Words

FLYLEAF …… このあとに，表紙に書きたい文字を書く。

LISTCODES …… コード化されたアイテムのコードと原文のリストを作る。

SORTKEY n …… nは1〜31の整数。たとえば4つの sortkey が使われているときは，それらに1, 2, 3, 4 と順序番号をつける。ソートのときの文字の大小関係を次に示す。

① 数

② = ≒ +

③ A〜1

④ < ・) −

⑤ J〜R

⑥ ∨ $ blank /

⑦ S〜Z

⑧ , (→

（2−2） OPTIONAL FORMAT SELECTOR

① Necessary Word

RECORD LENGTH n …… 抜粋されたアイテムが挿入されるロジカルレコードの長さを定義する。

POSITION n …… アイテムの最初の文字がロジカルレコードで占める位置。

CHARACTERS n …… マルテイプルアイテムのときだけ。サブアイテムまたはderived quantity に対して予約する文字数を決める。

② Optional Word

ACCEPT FORMAT …… 指定されたフォーマットがアイテムに対して十分のスペースをとっているかどうかのチェックを行なわない。

BLOCK LENGTH n …… n個のロジカルレコードが1つのフィジカルレコードを作る。

FLYLEAF

LISTCODES

SORTKEY n

(2－3) AUTOMATIC REPORT GENERATOR

① Necessary Word

REPORT …… 自動的にアイテム全体を抜粋する。

② Optional Word

HEADLINE n …… このあとにつづく文字列が各頁の上段に印刷される。

PAGEWIDTH n …… 印刷の巾の指定。

NEWPAGE …… 通常は SORTKEY n と共に用いる。1個のアイテムまたはderived quantity の値または explicit qualifier のついたアイテムの値が，すぐ前のエレメントの同じアイテムの値と異なるとき，エレメントは新しい頁から印刷される。

SUPRESS …… 印刷しようとする値が，まえに印刷されたエレメントの該当するアイテムの値と等しいときは，印刷を避ける。

DECODE …… コードを原文に直して印刷する。

FLYLEAF

LISTCODES

SORTKEY n

③ Special Option

COMPUTE MEAN …… 指定されたアイテムのエレメント全体の平均値を計算する。

COMPUTE SUM …… 指定されたアイテムのエレメント全体の和を計算する。

COMPUTE MEANS …… sortkey アイテムの値が変わるごとに，指定されたアイテムの平均値が計算される。

COMPUTE SUMS …… sortkey アイテムの値が変わるごとに，指定されたアイテムの和が計算される。

COUNT …… 指定されたアイテムがとる異なる値を調べその頻度を計算する，

INVERT …… 指定されたアイテムがとる異なる値を調べ，それらとエレメントナンバーの対応表を作る。

（2−4） COMPLETE REPORT GENERATOR

① Necessary Word

ELEMENT LENGTH n …… 1つのエレメントから抜粋されるアイテムのために，予約する行の数を指定する。

ROW n …… 抜粋されたアイテムの最初の文字が印刷される行を指定する。

COLUM n …… 抜粋されたアイテムの最初の文字が印刷される列を指定する。

② Optional Word

COLUMNWISE …… マルティプルアイテムに関するもので，サブアイテムを1行に印刷するかわりに，行を変えてそろえて印刷する。

WITH DESCRIPTION …… 抜粋されるアイテムとともに，アイテムディスクリプションを印刷する。

WITH TEXT …… このあとに続く文字列が，アイテムを印刷するまえに印刷される。

FLYLEAF

LISTCODES

SORTKEY n

HEADLINE n

NEWPAGE

SUPRESS

DECODE

PAGEWIDTH n

COMPUTE MEAN

COMPUTE MEANS

COMPUTE SUM

COMPUTE SUMS

マルティプル・アイテムに関しては，次のような指定ができる。この指定は 2.2.1 ～ 2.2.4 の4つの出力形式すべてで使える。

TOTAL	
SUM	
MEAN	
MINIMUM	
MAXIMUM	
FIRST	FIRST TO SUB-ITEM n
SUB-ITEM n	SUB-ITEM n TO SUB-ITEM m
LAST	LAST TO LAST BUT n
LAST BUT n	LAST BUT n TO LAST BUT m
	ALL

例)

INFOL Information Form

APPLICATION: .....

Check phase and purpose:

ESTABLISHMENT	UPDATE	INTERROGATION
ITEM DESCRIPTIONS CATEGORY-TYPE VALIDATIONS INITIAL INPUT	SELECTIVE CRITERIA SELECTIVE UPDATES DISCRETE UPDATES NEW ELEMENT MODIFY ELEMENT REMOVE ELEMENT	RETRIEVAL CRITERIA EXTRACT ITEMS X

Item Description	Item No.	Information
Reference Number	# 1 #	EXTRACT
Authors	# 2 #	EXTRACT
Title	# 3 #	EXTRACT

図 2.1.2 Automatic Format Selector

INFOL Information Form

APPLICATION: .....

Check phase and purpose:

ESTABLISHMENT	UPDATE	INTERROGATION
ITEM DESCRIPTIONS CATEGORY-TYPE VALIDATIONS INITIAL INPUT	SELECTIVE CRITERIA SELECTIVE UPDATES DISCRETE UPDATES NEW ELEMENT MODIFY ELEMENT REMOVE ELEMENT	RETRIEVAL CRITERIA EXTRACT ITEMS X BLOCK LENGTH 3 RECORD LENGTH 400

Item Description	Item No.	Information
Reference Number	# 1 #	POSITION 1
Authors	# 2 #	POSITION 6 CHARACTERS 43 FIRST TO SUB-ITEM 4
Title	# 3 #	POSITION 178

図 2.1.3 Optional Format Selector

```
                                                        9 MAY 1986          PAGE NO.      1

FILE...NBS FILE/HR23 01          MAGNETIC TAPE FILE     2

        ITEM DESCRIPTION          ITEM NO    POSITION    CHARS/SUB-ITEM

        REFERENCE NUMBER              1          1
        AUTHORS                       2          6          45
        TITLE                         3        393

        NUMBER OF LOGICAL RECORDS.......     926
        NUMBER OF PHYSICAL RECORDS......     948
        RECORD LENGTH (CHARACTERS)......     606
        BLOCKING FACTOR.................       1
```

図 2.1.4 Automatic Format Selector Sample (Short Report)

```
                                                        9 MAY 1986          PAGE NO.      1

FILE...NBS FILE/HR23 01          MAGNETIC TAPE FILE     1

        ITEM DESCRIPTION          ITEM NO    POSITION    CHARS/SUB-ITEM

        REFERENCE NUMBER              1          1
        AUTHORS                       2          6          45
        TITLE                         3        178

        NUMBER OF LOGICAL RECORDS.......      44
        NUMBER OF PHYSICAL RECORDS......      15
        RECORD LENGTH (CHARACTERS)......     402
        BLOCKING FACTOR.................       3
```

図 2.1.5 Optional Format Selector Sample (Short Report)

INFOL Information Form

APPLICATION: INFOL Subprograms .....

Check phase and purpose:

ESTABLISHMENT	UPDATE	INTERROGATION
ITEM DESCRIPTIONS CATEGORY-TYPE VALIDATIONS INITIAL INPUT	SELECTIVE CRITERIA SELECTIVE UPDATES DISCRETE UPDATES NEW ELEMENT MODIFY ELEMENT REMOVE ELEMENT	RETRIEVAL CRITERIA EXTRACT ITEMS x ELEMENT LENGTH 25

Item Description	Item No.	Information
Subprogram Number	*1*	WITH DESCRIPTION ROW 1 COLUMN 1
Subprogram Name	*2*	WITH TEXT NAME * ROW 1 COLUMN 25
Purpose	*3*	WITH TEXT -- * ROW 1 COLUMN 40
Level	*4*	
Programmer	*5*	WITH DESCRIPTION DECODE ROW 2 COLUMN 25
Input parameters	*6*	WITH DESCRIPTION ROW 4 COLUMN 8
Input parameter classes	*7*	
Output parameters	*8*	WITH DESCRIPTION ROW 5 COLUMN 1
Output parameter classes	*9*	
Cells destroyed	*10*	
Cells-destroyed classes	*11*	
Constants used	*12*	
Constant classes	*13*	
Subprograms called by this subprogram	*14*	
Subprograms which call this subprogram	*15*	
Index registers used	*16*	
Locations required for subprogram	*17*	
Errors checked for	*18*	WITH DESCRIPTION COLUMNWISE ROW 7 COLUMN 1
Start and completion dates	*19*	
Phases used in	*20*	
Abstract	*21*	WITH DESCRIPTION ROW 7 COLUMN 53

図 2.1.6 Complete Report Generator

INFOL Information Form

APPLICATION: INFOL Subprograms .....

Check phase and purpose:

ESTABLISHMENT	UPDATE	INTERROGATION
ITEM DESCRIPTIONS CATEGORY-TYPE VALIDATIONS INITIAL INPUT	SELECTIVE CRITERIA SELECTIVE UPDATES DISCRETE UPDATES NEW ELEMENT MODIFY ELEMENT REMOVE ELEMENT	RETRIEVAL CRITERIA EXTRACTIONS X

Item Description	Item No.	Information
Subprogram Number	*1*	REPORT
Subprogram Name	*2*	REPORT
Purpose	*3*	REPORT
Level	*4*	
Programmer	*5*	DECODE
Input parameters	*6*	REPORT
Input parameter classes	*7*	
Output parameters	*8*	REPORT
Output parameter classes	*9*	
Cells destroyed	*10*	
Cells-destroyed classes	*11*	
Constants used	*12*	
Constant classes	*13*	
Subprograms called by this subprogram	*14*	
Subprograms which call this subprogram	*15*	
Index registers used	*16*	
Locations required for subprogram	*17*	
Errors checked for	*18*	REPORT
Start and completion dates	*19*	
Phases used in	*20*	
Abstract	*21*	REPORT

図2.1.7 Automatic Report Generator

```
EXTRACTIONS
ELEMENT LENGTH 25
FLYLEAF THIS IS AN EXAMPLE OF THE USE OF THE COMPLETE REPORT GENERATOR *
HEADLINE 3 THIS HEADLINE WILL APPEAR ON LINE 3 OF EACH PAGE OF THE REPORT *
*1* WITH DESCRIPTION ROW 1 COLUMN 1
*2* WITH TEXT NAME * ROW 1 COLUMN 25
*3* WITH TEXT --* ROW 1 COLUMN 40
*5* WITH DESCRIPTION DECODE ROW 2 COLUMN 25
*6* WITH DESCRIPTION ROW 4 COLUMN 2
*8* WITH DESCRIPTION ROW 5 COLUMN 1
*18* WITH DESCRIPTION COLUMNWISE ROW 7 COLUMN 1
*21* WITH DESCRIPTION ROW 7 COLUMN 53
```

図2.1.8 Sample Complete Report Generator Keypunched

```
EXTRACTIONS
FLYLEAF THIS IS AN EXAMPLE OF THE USE OF THE AUTOMATIC REPORT GENERATOR *
HEADLINE 3 THIS HEADLINE WILL APPEAR ON LINE 3 OF EACH PAGE OF THE REPORT *
*1* REPORT
*2* REPORT
*3* REPORT
*5* DECODE
*6* REPORT
*8* REPORT
*18* REPORT
*21* REPORT
```

図2.1.9 Sample Automatic Report Generator Keypunched

THIS HEADLINE WILL APPEAR ON LINE 3 OF EACH PAGE OF THE REPORT

SUBPROGRAM NUMBER.. 5 NAME CONVNU -- TO CONVERT A NUMERIC ITEM FROM BCD TO FLOATING-POINT BINARY.
PROGRAMMER.. T. W. OLLE

INPUT PARAMETERS.. REALPOS1 * REALOC1 * NOCHMIN1
OUTPUT PARAMETERS.. LABEL * DECIMALS

ERRORS CHECKED FOR
MORE THAN 10 DIGITS.
ILLEGAL CHARACTERS.

ABSTRACT.. TREATS EACH CHARACTER IN TURN FROM LEFT TO RIGHT. IF CHARACTER IS NUMERIC, IT IS ADDED TO THE PREVIOUS TOTAL AND THE SUM MULTIPLIED BY 10 UNTIL THE LAST CHARACTER IS MET, THE LAST CHARACTER IS SIMPLY ADDED. + SIGN AS FIRST CHARACTER IS PERMITTED BUT THE COUNT OF CHARACTERS IN THE STRING IS REDUCED. . (DECIMAL POINT) STARTS A COUNT OF THE NUMBER, N, OF FRACTION DIGITS. AFTER THE LAST CHARACTER IS PROCESSED, A SCALE FACTOR IS COMPUTED FROM N. THE INTEGER PREVIOUSLY FORMED IS FLOATED AND MULTIPLIED BY THE SCALE FACTOR TO OBTAIN THE FINAL RESULT. IF THERE ARE MORE THAN 10 NUMERIC CHARACTERS OR AN ILLEGAL CHARACTER THEN DECIMALS IS SET TO -1 AS AN ERROR INDICATION.

SUBPROGRAM NUMBER.. 96 NAME SRCHS1 -- TO LOCATE THE LAST SUBITEM BUT N IN A MULTIPLE ITEM, N GE 0.
PROGRAMMER.. J. W. EUSEBIC

INPUT PARAMETERS.. PIB * PIE * N
OUTPUT PARAMETERS.. PSIB * PSIE * RC

ERRORS CHECKED FOR
N GREATER THAN THE NUMBER OF SUBITEMS LESS ONE.

ABSTRACT.. IF THE ITEM IS NOT ALPHANUMERIC, THEN BOTH PSIB AND PSIB ARE SET EQUAL TO PIE-N = ADDRESS OF DESIRED SUBITEM. IF THE ITEM IS ALPHANUMERIC, THE SUBROUTINE SEARCHES BACKWARD FROM LOCATION PIE TO LOCATE THE FIRST WORDS OF THE LAST N+1 SUBITEMS. IF PSIE BECOMES SMALLER THAN PIB, THE ERROR FLAG, RC, IS TURNED ON. ON ENTRY, THE ITEM NUMBER MUST BE IN INDEX REGISTER B6.

図2.1.10 Sample Complete Report Generator Output

SUBPROGRAM NUMBER	204
SUBPROGRAM NAME	MVERST
PURPOSE	MOVE ERROR STRING TO R3
PROGRAMMER	J. R. MARSHECK
INPUT PARAMETERS	BSW • LASTSYN • RAWLOC
OUTPUT PARAMETERS	EMCOUNT
ERRORS CHECKED FOR	
ABSTRACT	MVERST COMPUTES NUMBER OF WORDS NEEDED FOR ERROR MESS, CHECKSIF IT WILL FIT R3, CALLS BUFCON IF NECESSARY, MOVES THE ERROR STRING TO R3 AND CONSTRUCTS THE MESSAGE LEADER.
SUBPROGRAM NUMBER	205
SUBPROGRAM NAME	OVITON
PURPOSE	OBTAIN VALUE OF ITEM ONE
PROGRAMMER	J. R. MARSHECK
INPUT PARAMETERS	
OUTPUT PARAMETERS	
ERRORS CHECKED FOR	
ABSTRACT	OVITON OBTAINS THE VALUE OF ITEM ONE AND RETURNS IT IN REGISTER A. ENTER WITH B1 = ABS ADDRESS OF ELEMENT LEADER.
SUBPROGRAM NUMBER	224
SUBPROGRAM NAME	ESTABLSH
PURPOSE	TO ESTABLISH THE INFORMATION FILE
PROGRAMMER	M. P. OTOOLE
INPUT PARAMETERS	STLAB
OUTPUT PARAMETERS	
ERRORS CHECKED FOR	ILLEGAL CONTROL WORD
ABSTRACT	USES SUBROUTINES TO PROCESS BASEFILE DATA. WRITES BASEFILE, THEN USES PREPELEM TO PROCESS INITIAL INPUT. REWINDS TAPE AND REWRITES BASEFILE AFTER BOOKKEEPING WORDS HAVE BEEN SET UP.
SUBPROGRAM NUMBER	230
SUBPROGRAM NAME	ADVCRIT
PURPOSE	ADVANCES CRIT POINTER BY 6 AND ZEROES OUT 6 WORDS OF CRIT.
PROGRAMMER	P. L. MCNAIR
INPUT PARAMETERS	
OUTPUT PARAMETERS	CRIT
ERRORS CHECKED FOR	CRIT TABLE OVERFLOW
ABSTRACT	ADVANCES THE POINTER TO THE CURRENT CRIT ENTRY (B3) BY 6, THEN ZEROES OUT THE 6 WORDS OF CRIT BEGINNING WITH THE ENTRY POINTED TO BY R3. IT MAY BE ENTERRED AT THE ENTRY POINT ZEROCRIT JUST FOR ZEROING OUT 6 WORDS OF THE TABLE. THIS ROUTINE IS A PART OF DO.ACTS.
SUBPROGRAM NUMBER	250
SUBPROGRAM NAME	SELCRI
PURPOSE	TO INTERFACE WITH INTERROGATION SPECIFICATIONS
PROGRAMMER	J. R. MARSHECK
INPUT PARAMETERS	BGSCAN • NXBLOK • RAWLOC
OUTPUT PARAMETERS	BADCRIT • BEGIT • CURCH • ELOK • LASTSYN • NXWORK • PCENUM • PREBLOK • RAWLOC • RAWPOS • SYNMASK • SYNSTART
ERRORS CHECKED FOR	
ABSTRACT	CALLS INTERR TO PROCESS SELECTIVE CRITERIA, CALLS PROCEL TO MOVE PREVIOUS SELECTIVE UPDATE TO R1. IF ERROR IN CRITERIA, SETS BASCHIT EQUAL ZERO, ELSE TO NON-ZERO.

☒ 2.1.11 Sample Automatic Report Generator Output

INFOL Information Form

APPLICATION: .....

Check phase and purpose:

ESTABLISHMENT	UPDATE	INTERROGATION
ITEM DESCRIPTIONS CATEGORY-TYPE VALIDATIONS INITIAL INPUT	SELECTIVE CRITERIA SELECTIVE UPDATES DISCRETE UPDATES NEW ELEMENT MODIFY ELEMENT REMOVE ELEMENT	RETRIEVAL CRITERIA EXTRACTIONS X

Item Description	Item No.	Information
Authors	#24	COUNT

図2.1.12 Example of Count

INFOL Information Form

APPLICATION: .....

Check phase and purpose:

ESTABLISHMENT	UPDATE	INTERROGATION
ITEM DESCRIPTIONS CATEGORY-TYPE VALIDATIONS INITIAL INPUT	SELECTIVE CRITERIA SELECTIVE UPDATES DISCRETE UPDATES NEW ELEMENT MODIFY ELEMENT REMOVE ELEMENT	RETRIEVAL CRITERIA EXTRACTIONS X

Item Description	Item No.	Information
Authors	#24	INVERT

図2.1.13 Example of Invert

NRS FILE COUNTED BY AUTHOR

KENNA, B.T.	1
KENNA, L.A.	1
KENT, R.A.N.	1
KETELLE, B.H.	1
KIENBERGER, C.A.	1
KIENLE, P.	1
KIGOSHI, K.	2
KIM, C.K.	3
KING, F.R.	5
KOCH, H.J., JR.	1
KOCH, R.C.	2
KOHN, H.W.	1
KRAMER, H.H.	1
KRUGER, P.	1
KURODA, R.	1
KUYKENDALL, W.F.	2
LAING, K.M.	2
LAKSHMANAN, S.	1
LARSON, D.V.	1
LASCH, J.E.	1
LAUTTMAN, R.G.	1
LEAVITT, W.Z.	1
LEBOEUF, M.B.	2
LEDDICOTTE, G.W.	42
LEE, C.H.	1
LEE, W.	1
LEVINE, C.A.	1
LEWIS, J.E.	2
LEWIS, J.W.	1
LEWIS, W.W.	1
LINDNER, W.	1
LINNENBOM, V.J.	1
LIVINGOOD, J.J.	1
LOCKHART, L.B.	1
LOVETT, J.E.	1
LOVE, D.L.	1
LOWE, L.F.	1
LUKENS, H.R.	1
LUKENS, H.R., JR.	2
LUNDGREN, S.	1
LYKINS, J.W.	1
LYON, W.S.	1
MACKINTOSH, M.E.	1
MACKLIN, M.L.	1
MADDOCK, R.S.	1
MAHLMAN, H.A.	4
MAHONY, J.D.	1
MANNEY, T.R.	1
MAPPER, D.	1
MARCUS, Y.	1
MARKER, R.C.	1
MARX, W.	1
MARTIN, D.S., JR.	2
MARTIN, T.C.	2

図2.1.14 Example of Count Output

NBS FILE INVERTED ABOUT AUTHOR

9 KING, E.R.	136 487 488 489 730
1 KOCH, H.J., JR.	243
2 KOCH, R.C.	643 676
1 KOHN, H.W.	247
1 KRAMER, H.M.	1150
1 KRUGER, P.	652
1 KURODA, R.	572
2 KUYKENDALL, W.E.	273 590
2 LAING, K.M.	274 275
1 LAKSHMANAN, S.	738
1 LARSON, Q.V.	72
1 LASCH, J.E.	439
1 LAUTTMAN, R.G.	21
1 LEAVITT, W.Z.	446
2 LEBOEUF, M.B.	112 509
42 LEDRICOTTE, G.W.	43 54 55 56 79 80 81 82 277 282 286 287 288 289 290 291 292 293 294 295 296 297 298 300 329 641 694 695 722 759 763 946 954 973 974 981 1031 1035 1060 1088 1189 1190
1 LEE, C.M.	749
1 LEE, M.	931
1 LEVINE, C.A.	1160
2 LEWIS, J.E.	399 979
1 LEWIS, J.M.	53
1 LEWIS, M.M.	491
1 LINDNER, A.	828
1 LINNENBOM, V.J.	315
1 LIVINGOOD, J.J.	443
1 LOCKHART, L.B.	1123
1 LOVETT, J.E.	321

図 2.1.15 Example of Invert Output

2.1.3.3 UPDATE フェイズ

ファイルの変更を行なうフェイズである。セカンドレベルのコントロールワードとしてつぎのものがある。

ITEMIZE

DISCRETE UPDATES

SELECTIVE CRITERIA

SELECTIVE UPDATES

(1) DISCRETE UPDATES

エレメントナンバー（アイテム1の値）を指定して，ファイルの変更を行なう。つぎの3種類がある。

NEW ELEMENT …… 新しいエレメントはエレメントナンバーの値の同一のものがない限りファイルに挿入される。

REMOVE ELEMENT …… 指定したエレメントナンバーをもつエレメント全体を除去する。

MODIFY ELEMENT …… エレメント全体の挿入や除去ではなく，アイテムやサブアイテムの変更を行なう。つぎの6種類がある。

① DELETE …… 1つのアイテム全体を除去する。

② INSERT …… 1つのアイテム全体を挿入する。

③ SUBSTITUTE …… ファイルにすでに存在するアイテムに，別の値を与える。

④ ELIMINATE …… マルティプルアイテムから，指定した値を持つサブアイテムを除去する。

⑤ ADD …… マルティプルアイテムにサブアイテムを付加する。

⑥ REPLACE …… マルティプルアイテムにおいて，指定された値を持つサブアイテムを，指定された別の値に置き換える。

(2) SELECTIVE UPDATES

retrieval criteria の働きは，そのまま UPDATE に利用できる。すなわち，指定した retrieval criteria を満足するエレメントに対して，変更を行なうことができる。

例） SELECTIVE CRITERIA 1

18 EQ *GREY CODE*

SELECTIVE UPDATES 1

MODIFY ELEMENT

18 SUBSTITUTE GRAY CODE *

(3) ITEMIZE

Updates で行なわれた修正を順序に従って，箇条書きにしてリストする。

2.1.3.4 BOOKKEEPINGフェイズ

つぎのような情報を計算して印刷するフェイズである。

① マルティプルアイテムのサブアイテムの最大数

② 英数字アイテムの最大文字数

③ コード化されたアイテムのコードの原文の最大の長さ

④ 数値アイテムの整数部および小数部の最大桁数

これらの値は ESTABLISHMENT フェイズおよび UPDATE フェイズでは，自動的に計算され印刷される。

2.1.3.5 REVISION フェイズ

ファイルの構成を変更するフェイズである。つぎのような種類がある。

① 新しいアイテムを定義する。

② 既定義のアイテムディスクリプションを変更する。

③ 既存アイテムのカテゴリーを unary から multiple に変更する。

④ 新しいコードを付け加える。

⑤ 既存のコードを変更する。

⑥ validation criteria を付け加える。

⑦ 既存の validation criteria を変更する。

2.1.3.6 VALIDATION フェイズ

validation criteria の挿入や変更がなされた REVISION フェイズの後で，UPDATE によって修正すべきエレメントを決定するためのフェイズである。通常はつぎのような手順で行なう。

REVISION : 新しい criteria を指定する。

VALIDATION : 新しい criteria で古いエレメントをチェックする。

UPDATE : 実際のデータの変更を行なう。

2.2. ＧＩＳ

2.2.1 ＧＩＳの概要（Generalized Information System）

システム／３６０は情報ファイルをあらゆる部門で共通にまた簡単なプログラムで使用できるように考えられたプログラミング言語である。実際の使用はオペレイテングシステム／３６０の管理のもとでＧＩＳ Task として行われMulti-Tasking Mode と Tele-Processing Mode で使用可能である。ＧＩＳはＯＳ／３６０アセンブラー言語で書かれているが，ＧＩＳで作られたファイルはＯＳ／３６０アセンブラー，ＣＯＢＯＬやＰＬ／Ｉ等の言語で書かれたプログラムで使用することができる。すなわち，ＧＩＳはＩＢＭシステム／３６０用の汎用情報システムである。

なお，文献検索用のプログラム言語としてはDocument Processing System があり，これは不定型データを扱い，検索のために論理判断を行う。

この言語は次の２つの項目に分類される。

(1) Descriptive Statement …… システム分析をし，全体的にファイルを構成するためのもので，情報のデータの形式と関連性を規定する。規定されたものをData Descriptive Table（ＤＤＴ）といい，後の処理とデータファイルの媒体となるものである。

(2) Procedural Statement …… ファイルの作成と更新，判断機能によりデータを選択し検索し要約後，出力データを作るものである。

2.2.2 ファイルとデータの定義

データのファイルをつくるのはData Descriptive Statement でField, Segment（Record）とFileという形で論理的に情報を形成する。これはＯＳ／３６０の一つのデータ・セットをつくる。そのためこのファイルの検索，更新はＧＩＳの指示に従いＯＳ／３６０のデータ・マネジメントの機能によりなされる。ＧＩＳ使用者が使えるＯＳ／３６０のアクセス方法は順次，索引順次の２方法である。実際は次の形で定義する。

(1) Field-Oriented Data

Fieldはデータの最小単位である。Fieldの形態は次のものが使われる。

	最　小	最　大	IncreWent
２進数表示	８ビット	３２ビット	８ビット
パック十進数	２桁	３２桁	２桁
バ　イ　ト	１バイト	２５６バイト	１バイト
浮動小数点	４バイト	８バイト	４バイト

指定していく順番は次の通り。①Field名…… シンボリック名で参照するのに用いる。②Synonym名……Alternate名。③Coding……Field の単位を指定し上記4種類を使う。④Fieldの桁数……固定長か可変長か。⑤Error Checking…… バリディティ・チェックとデータの変換。⑥Dependency Function……桁数のテスト，Header編集のパターンを求める。⑦Security Code……QueryとMaintenance の保護に用いる。1～128のCodeがある。⑧Output File。

(2) Record（Segment）－Oriented Data

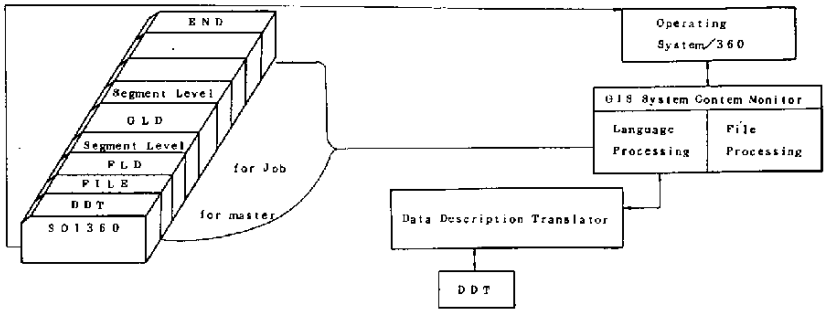
連続したFiledをSegmentとして定義し，このSegmentをOS／360 で定義したものを物理的レコードとして扱う。Segment の第一番目はMaster Segment といいLevel 0 がつく。2番目からはSubordjnae Segment といいLevel（1～15）をつける。指定の順番は次の通り。① Segment 名，②Segment Level……Tree構造に（0～15）のLvvelをつける。③分類欄の指定……最大7 Field まで可能， ④分類方法……正順か逆順か，⑤Parameter ……Mastr Segmentのみに使用する。これはData Management OS／360への指定に使用，⑥Count Field の指定

(3) File-Oriented Data

関連性のあるSegment を統括するためにFileとして定義する。①File名， ②Synonym名……Alternate名，③Security Control Key …… 最小のSecurityとなる。Query と Maintenance に用いる。④File Storage Control ……OSにおけるEvent の処理

Data Descriptive Language は次の様に与える。

プログラムの実行の際にはGIS System Control Monitor を呼び出すためにOS／360 Job Control Language がまず入る。GIS Monitor は使用者のTask Specification の仕事を識別するものである。GIS MonitorはLanguage Processor と File Processor に分れている。



2.2.3 ファイル処理のための言語（Procedural Statement）

ＤＤＴを媒体にして，実際のファイルをつくり，必要な情報を検索するための言語である。Procedural Statement は，(1) Procedure Control Statement，(2) Data Modification Statement，(3) I／O Control Statement，(4) Processing Statement の４種に大別できる。

(1) Procedure Control Statement

(A) Procedure Declaration Statement

QUERY, MODIFY, UPDATE と CREATE とあり，仕事の種類の大別をする。

A－1） QUERY …… 情報検索を要求することを示す。

```
QUERY A
LOCATE RECORD
WHEN X EQ `15'
LIST Y
EXHAUST
QUERY B
  ⁞
END PROCEDURE
```

Subprocedure

A, B は filename

A－2） MODIFY …… Field 単位でデータを変更するときに用いる。一般形の Statent は，

MODIFY filename, filename, filename

MODIFY master file FROM Source file

master file は MODIFY されるファイルであり，条件付探索をする。Source file は MODIFY する内容が入っているもので正順探索を行う。

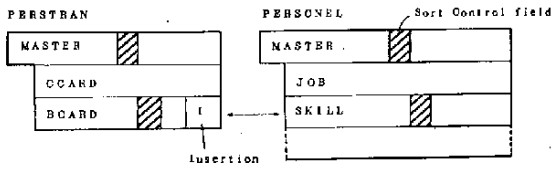
A－3） UPDATE …… Segment Level でファイルの変更を行う。Creation,

Addition と Deletion も UPDATE の一部で RE－FORMATING も UPDATE で行う。

UPDATE Masterfile FROM Source file

(PERSONAL)　　(PERSTRAN)

⊛ STRUCTURE SKILL　FROM BCARD



STRUCTURE により MASTER, SKILL に MASTER, BCARD が Mapping される。

⊛ INSERT SKILL …… 指定された Level の Segment 以下の Subordinate Segment 全てが Insert される。

A－4） CREATE …… UPDATE と同じだが新しくファイルをつくるときに作る。

CREATE A FROM B

↑

Master file　Source file

STRUCTURE JOB FROM Y

(B) END PROCEDURE …… 全ての Procedure の終りを示す。

(C) HALT …… Subprocedure の終りを示す。

(D) GO TO Statement Label …… Subprocedure から Subprocedure への Transfer は許されない。又 Locate partition と STRUCTURE partition 内のみで行うこと。

(E) IF …… 条件により Condition の流れが変る。

E－1） IF Conditional Expression

↓

Operand　Conditional Operand / Operator

Conditional Operator としては, GE, EQ, GT, LT, NE, LE, Operand としては Numeric Constant, Literal Constant を書き, File, Field 名, Define Work Area, Arithemetic Expression (＋, －, ＊, ／), Logical Expression (NOT, AND & Concatination ||)が書ける。

⊛ Conditional Expression　　TRUTH

FALSE

① FALSE …… 次の Conditional Delimitor までとばす。

② TRUTH …… IF の次の Statement から実行 Conditional Delimitor とは ELSE, IN ANY CASE, END PROCEDURE, SORT, RUN, STRUCTURE, EXAUST, FINALLY, 他の IF, WHEN 等がある。

E−2） IF n …… n＝1～999 n回目に TQUE となる。

E−3） IF SW n …… SET, RESET statement で ON 又は OFF にする。

$1 \leqq n \leqq 32$

(F) RUN Procedure name …… 別のGIS Task となる。後に述べる SAVEX でGIS Library に Compiled の形で登録されているプログラムを実行する。このプログラムはアセンブラー又はコンパイラーで書いたものでもよい。

(G) LINK routine name …… GIS Library に登録されていないもので, アセンブラーやコンパイラーで作成された Module との連結をする。

(2) DATA MODIFICATION STATEMENT

(A) Field Modification Statement

A−1） INCREASE / DECREASE / MULTIPLY / DIVIDE　1st Operand BY 2nd Operand

1st Operand: File−field name / VARIABLE n

2nd Operand: File−field Name, Arithemetic Expression / VARIABLE n / Numeric Constant

この Statement は QUERY, MODIFY の中でのみ使用可能である。但し, 第1オペランドが VARIABLE n のときは UPDATE, CREATE Procedure 内で使用してよい。

A−2） CHANGE 1st Operand TO 2nd Operand

1st Operand: VARIABLE n / LITERAL n / File−field

A−3） ERASE Operand …… Operand で指定した Field を ABSENT の状態にする。

(B) File Conversion Statement

UPDATE, CREATE Subprocedure 内においてのみ使用可能で File Conversion Declaration Statement (STRUCTURE), Action Statement (INSERT, REPLACE, DELETE, REMOVE, INCLUDE & etc.) がある。

B−1） STRUCTURE Masterfile Segment Name FROM Sourcefile Segment Name …… データ構造の Mapping を行う。新しくデータを附加するとき，どの Level に入れるかを指示するために用いる。

MASTER	Source
M1	S_1
M2	S_2
M3	Source に対応する Segment なし

B−2） EQUATE Master field Name TO Source field name

Numeric, Literal Constant

END EQUATE

STRUCTURE statement の直後に置くこと，そしてデータを移す。

B−3） EXCEPT …… Implicit に対応しているものだけ対応関係をはずす。（CREATE で対応させたものは除く。）

B−4） FINALLY …… ある条件が起きた時，Structure Partition から Exit させたい時に使う。Source File の Exhaust statement の次の Statement から実行。

B−5） INSERT master-file Segment name …… STRUCTURE で Mapping されている Segment Name を挿入する。これは Data Base 上にない Segment である。指定された Level の Segment 以下の Subordinate Segment の全てが Insert される。Sort Control Field が一致してすでに存在しておれば挿入しない。そして Error Message を出す。

B−6） STORE Masterfile Segment Name …… Insert と同じ。ただし Control field に同じものがあると挿入しない。

B−7） 以下まとめて表にする。

ACTION WORD	System の実行	
	Segment Data のある時	Segment Data の無い時
INSERT	Error Message を出す	新しい Segment を挿入する
STORE	何も実行しない	新しい Segment を挿入する
APPEND	追加 Segment を挿入する	新しい Segment を挿入する
DELETE	Segment を削除する	Error Message を出す
REMOVE	Segment を削除する	何も実行しない
REPLACE	Segment を置き換える	Error Message を出す
INCLUDE	Segment を置き換える	新しい Segment を挿入する
REPLACEF	field 毎に置き換える	Error Message を出す
INCLUDEF	field 毎に置き換える	新しい fields を挿入する

(C) IGNORE statement …… その Subprocedure 中のみに有効でそのファイルに対してDDT中で EDIT, ENCD, DECD を Field 毎に指定しているのを無視させる。

EDIT／ENCODE

IGNORE filename DECODE

(3) INPUT／OUTPUT CONTROL STATEMENT

(A) DATA LOCATION STATEMENT

LOCATE Segment Name

READ RECORD

LOCATE PERS : RECORD …… とすると毎回 File の最初から Sequential Search (IS の Direct Search は別)

READ PERS : RECORD と指定すると直前に Read した RECORD の次から Search する。

(B) EXHAUST …… END procedure と Subprocedure の終りでは Implicit に Exhaust する。与えるときは

EXHAUST Segment Name
RECORD
or n ;

(C) WHEN Condition Statement …… LOCATE 又は READ に続いて Coding し GIS File から取り出す情報を指定する。

LOCATE PERS : RECORD ;

WHEN SKILL EQ SOURCE : SKILL ;

(D) HOLD …… Temporary Sequential File の作成をする。

① HOLD Hold filename, element, ……

② HOLD Hold filename, RECORD,

③ HOLD Hold filename, element = field name, ……

(4) PROCESSING STATEMENTS

次のことを行う。

COUNT …… その Field の個数をかぞえる。

TOTAL …… その Field の合計をとる。

AVERAGE …… その Field の平均値をとる。

UNICOUNT …… その Field の値が変ったとき＋1をする。

TALLY n …… Logical に Statement が Execute されたとき＋1する。

Statements の書き方は次の通り。

COUNT
TOTAL
AVERAGE
UNICOUNT
} Field Name

SUBTOTAL
SUBCOUNT
SUBAVEGE
} Field 1 BY Field 2

COUNT n
TOTAL n
AVERAGE n
UNICOUNT n
VARIABLE n
} { VARIABLE n / Arithemetic Expression / field Name / Numeric Constant }

TALLY n : no operand

```
QUERY
COUNT MANNO
LOCATE RECORD
WHEN MANNO EQ A
      ⟨
EXHAUST
```

全ファイルについて MANNO の Field を持っている RECORD の個数をかぞえる。

(5) DATA DRESENTATION

DATA を出力して印刷したいときに用いる。

(A) LIST { LOCAL / OFFLINE } RECORD element, element, ……

element は field name, numeric 又は Literal constant, Defined work area name そして Arithemetic Expression ならよい。

(B) Formal Reporting

Output の形式を指定して印刷したときに使う。

B−1) REPORT …… Report の始めを指定する。

① 〔Label〕REPORT WIDTH×××, BODYLINE××× 〔, SPACE×〕 〔, OUTLINE〕 〔, { LOCAL / OFFLINE }〕

② 〔Label〕REPORT STANDARD

③ 〔Label〕REPORT STANDARD 以下①に同じ。

WIDTH …… 1行の桁数の指定で最大120桁まで。

BODYLINE ……最後の HEADER から最初の TRAILLER までの Line の数。

SPACE …… 印刷の Space の指定で1, 2, 3とある。指定しないと1をとる。

OUTLINE ……指定しないと1桁に一杯印刷する。

B−2) END REPORT …… REPORT の指定の終りを示す。

B−3) TITLE PP Operand …… 1st 頁のみに表題を印刷する。PP は Print Position を示す。

HEADER PP Operand ……各頁の表題になる。

DETAIL PP Operand ……各桁に示す。

EJECT …… 紙送り。

SUMMARY ……最後の頁のみ印刷。

TRAILER ……各頁の終りに印刷。

2.2.4 SYSTEM UTILITY

自由にデータを検索できなくするために SECURITY Code を使用できる。又GIS Task をあらかじめ GIS Library に登録することもできる。GISは SYSTEM UTILITY としてこれを行う。

(1) SECULITY …… DDT

① Procedure	Data Base
この Procedure を実行する User はどの Security Category を使用するかを指定する。 ＊SEC＝Security Code 最大8桁まで。	DDTで規定しておく File Level　この2つで Security Category で規定 Field Level　（1～128）

File Level についたものを Minimum Security という。

② SECURITY CODE は色々の場合に変えて使うこともできる。このとき SECURITY CODE を TABLE として与えることができる。

```
SECURITY TABLE
MASTER ACCESS IS 00000000
MASTER ACCESS IS 789ABCDE
362234AB   1, 3, 5, 6, 7, 12
31624409   1, 2, 3, 11
GIRISK42   1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12,
#13, 15
```

col 1に#は Continuation Card を示す。

(2) GIS Task の指定

GISで使用するプログラムを後で呼び出すために Task として貯えておく。これには SAVE と SAVEX（実行できる形で貯える）と ACC Statement で62桁の Accounting Information も貯えることができる。

例）

使用者	登録
Procedure	UTILITY
	SAVE PROCNAME
	ACC = (SMITH, J. B. EXT. 37 ACNT NO = 147298 Q3)
CALL PROCNAME	QUERY FILEA
	⋮
	etc
	SAVEX UPDATE14
	ACC = (BROWN T. C. …… etc.)
CALL UPDATE4	UPDATE FILEA FROM FILEB
	⋮
	etc.

2.2.5 ＧＩＳプログラム例

DATA DESCRIPTIVE LANGUAGE の例

```
ex. DDT;
    FILE  NAME = PERSONEL
    SYNM  NAME = GUIDE
    FLD   NAME = YEAR, UNITS = EBCD, LENGTH = 2, JUST = L
    FLD   NAME = RANK, UNITS = EBCD, LENGTH = 3, JUST = L
    FLD   NAME = CORPNAME, UNITS = EBCD, LENGTH = 35, JUST = L
    FLD   NAME = COUNT, UNITS = PACD, LENGTH = 2, JUST = R
    SEGM  NAME = CORPSEGM, LEVEL = ϕϕ,
    #SORT = CORPNAME, A, TYPE = RECORD
    DATM DSORG = IS, BLKSIZE = 8ϕϕ, DSNAME = DEMOFILE,
    #LRECL = 8ϕϕ, RECFM = F, UNIT = 2311, OPTCD = W,
    #KEYNAME = CORPNAME, CYLOFL = ϕ, INDXSIZE = ϕ,
    #NTM = ϕ, SPACE = 1, VOLUME = SER = OSL1B1
    FLD   NAME = YOURNAME, LENGTH = 35, UNITS = EBCD,
    #JUST = L
    FLD   NAME = YOURCITY, LENGTH = 25, UNITS = EBCD,
    #JUST = L
    FLD   NAME = URSTATE, LENGTH = 16, UNITS = EBCD,
    #JUST = L
    SEGM  NAME = YOURDATA, LEVEL = ϕ1, TYPE = TRAILER,
    #OPTION = CNT, OPTFNM = COUNT, SORT = YOURNAME, A
    END
```

Creating Data Files（データファイルの作成）

```
CREATE PERSONEL FROM CARDPRSL
STRUCTURE MASTER FROM ACARD
INSERT MASTER
END PROCEDURE
```

Updating Data Files（データファイルの変更）

```
UPDATE PERSONEL FROM PERSTRAN
STRUCTURE SKILL FROM BCARD
EQUATE
MANNO TO EMPNO
END EQUATE
IF BCODE EQ 'I'
INSERT SKILL
IF BCODE EQ 'D'
REMOVE SKILL
IF BCODE EQ 'C'
REPLACEF SKILL
STRUCTURE JOB FROM CCARD
EQUATE
MANNO TO EMPNO
RATE TO SALARY
END EQUATE
IF CCODE EQ 'D'
REMOVE JOB
IF CCODE EQ 'C'
INCLUDE JOB
END PROCEDURE
```

Query（探索）

```
QUERY PERSONEL
LOCATE RECORD
LOCATE JOB (LAST)
REPORT WIDTH 110, BODYLINES 56, SPACE 2
TITLE
30 'EUREKA TOY MANUFACTURING CORPORATION'
SPACE 25
TITLE
30 'PERSONEL INVENTORY - ALL DIVISIONS'
SPACE 5
HEADER
35 'EUREKA EMPLOYEE DATA - ALL DIVISIONS'
SPACE 2
HEADER
```

```
10 'EMP. NAME'
35 'EMP. NUMBER'
50 'LOCATION'
65 'MARITAL STATUS'
80 'SALARY'
SPACE 2
DETAIL
10 NAME
37 MANNO
50 LOCATION
71 MARSTAT
80 RATE
TALLY1 ON MARSTAT EQ 'S'
TALLY2 ON MARSTAT EQ 'M'
SUMMARY
20 'TOTAL NUMBER OF SINGLE EMPLOYEES IS'
55 TALLY1
SPACE 5
SUMMARY
20 'TOTAL NUMBER OF MARRIED EMPLOYEES IS'
56 TALLY 2
SPACE 3
TRAILER
50 'PAGE'
55 SYSPAGE
END REPORT
EXHAUST RECORD
END PROCEDURE
```

アウトプット例 — 第一ページ

```
                EUREKA TOY MANUFACTURING CORPORATION

                PERSONNEL INVENTORY - ALL DIVISIONS
```

アウトプット例 — 第二ページ以降

```
          EUREKA EMPLOYEE DATA - ALL DIVISIONS

EMP.  NAME      EMP.NUMBER   LOCATION   MARITAL STATUS   SALARY

U.S.  BUTLER      236853     NEW YORK         M            800
O.T.  REDDING     397447     NEW YORK         S            675
J. C. WILSON      480112     NEW YORK         M           1015

TOTAL NUMBER OF SINGLE EMPLOYEES IS 20
TOTAL NUMBER OF MARRIED EMPLOYEES IS 39

                          PAGE 29
```

2.3 AIS

2.3.1 AISの概略

AISは要求されたレポートをリアルタイムで遠隔表示装置に表示するマネージメントのための情報システム[2]でAutomatic Information Systemの略である。

レポートの種類は会社の営業状況，市場の状態，自社製品の内容およびその使用状況などよりなる。AISは汎用性があるので，その仕様に合うデータベースを使って，どのような種類のレポートでも要求に応ずることができる。

端末装置としてはIBM2260遠隔表示装置を使用する。プログラムは検索用のオンラインプログラムとファイル作成用のオフラインプログラムとがある。1968年に日本IBMシステムセンターで開発したものである。インプリメンテーションには約100 MAN－DAY および130 MACHINE－HOUR を費している。

2.3.2 機能上の特徴

(1) リアルタイム処理

質問に対するレスポンスはリアルタイムで応答される。レスポンスタイムは，ほとんど数秒以内である。

(2) ソフトコピーとハードコピー

オンラインリアルタイム検索は，ハードコピーを印刷するよりも，必要なレポートをその場で一見するだけでよい場合が多い。AISでは，検索された情報を"ソフトコピー"として，IBM2260遠隔表示装置上にディスプレイすることになっている。ただしハードコピーが必要ならば，2260キーボード上の PRINT キーを押すだけで，直ちにIBM1053プリンタ上にプリントアウトできる。

(3) 対話形式

一般に選択方式と直接方式との2つがあるが，AISは選択方式を採用している。選択方式は，"メニュー方式"であって，ユーザが画面上の指示に従って必要項目を選択し，その番号をキー・インすればよいようになっている。

(4) コントロール機能

ページングのために，Next Page，Back Page，Back Level がある。Back Level は，すぐ上のレベルのファイルの内容をコールする。Next Page，Back Page は，同一レベルのファイルの次のページ，直前ページの呼び出しである。

ページングの機能に対して，使用経験を重ねるに従っていろいろの要求が出てくるであろう。予想されるものの一つに，"multidisplay"というべきものがある。これは，連続的なページめめくり（現在は，離散的である）やページの合成（あるページの一部と別のページの一部を同一画面に出す）である。これらの要求をみたすためには，ディスプレイ技術の発達が必要である。

このほかに，特定ファイルの検索終了および全検索の完了を示す機能，さらに最終レコードの全キーを知っている場合には，全キーをインプットし直接最終レコードを検索する機能も用意されている。

(4) セキュリティ・チェッキング

各ユーザは5桁のパスワードを持つ。このパスワードに対してシステムは2桁の Previleged Code（PV）を割当てる。一方，ファイルにはデータセット（＝ファイルの単位）ごとに2桁の Security Code（SC）が与えられている。PV≧SCの条件で，ファイルは検索可能である。

(6) ファイルの作成と更新

オフラインプログラム（PL／1を使用）によってISファイル(Indexed Sequential File)が作成または更新される。ファイルは，画面単位のレコード構成，すなわち1画面＝1レコードである。1レコードは960文字からなる。そのうち，10文字をファイル上のキーとして使用する。

キー＝LLIIJJKKPP

LL ＝ Security Code

II ＝第1レベルのファイルのキー

JJ ＝第2 〃 〃 〃

KK ＝第3 〃 〃 〃

PP ＝各レベルにおけるページ番号

2.3.3 検索操作手順

選択方式であるから，操作手順は常に画面に指示され，次の手順で行われる。

(1) パスワードのキー・イン

最初の画面（AIS START Image という）に対してパスワードをキー・インする。

(2) アイテムコードおよびコントロール文字のキー・イン

パスワードが正しければ，次の画面に DATA SET SELECTION LIST がディスプレイされる。その中から，検索の対象となるデータ・セットを選択し，そのアイテムコードをキー・インする。

以後，そのデータ・セットに関し，第1レベル，第2レベル，第3レベルの順で検索する。

コントロール文字は次のとおりである。

① ／N （Next Page）

同一レベルの次のページをコール

② ／B （Back Level）

すぐ上のレベルのファイルの内容をコール

③ ／BP （Back Page）

同一レベルの前ページをコール

④ ／E （End of Operation）

全検索の終了

⑤ ／D （Direct Key Input）

直接，最終レベルのファイルを検索することを指示する。／Dに続いて，該当するキーをインプットする。

⑥ ENTER Key のみ（End of Data Set）

そのデータセットの検索終了

⑦ PRINT Key with SHIFT

画面の内容をプリントアウト

⑧ ENTER Key with SHIFT

コントロールをシステムにもどす（キー・インの最後）

2.3.4 システム・プログラムおよびデータセット

(1) オンラインプログラム（コアレジデント）

Line Control Program	（8.7Kバイト）
Message Processing Program	（7.3Kバイト）
File Accessor	（7Kバイト）

(2) オフラインプログラム

File Creation／Update Program

Data Set Print Program

(3) データセット

ＡＩＳファイルとしてインプットされるデータは，１２レコード（カード１２枚）が１ロジカルレコードとして処理される。

最初の６文字は，キーの中の JJKKPP に対応するものでなければならない。１２×８０バイトは，そのまま２３１４上にＩＳファイルとして作成される。ＡＩＳのファイル作成手順は，図2.3.1のようになる。

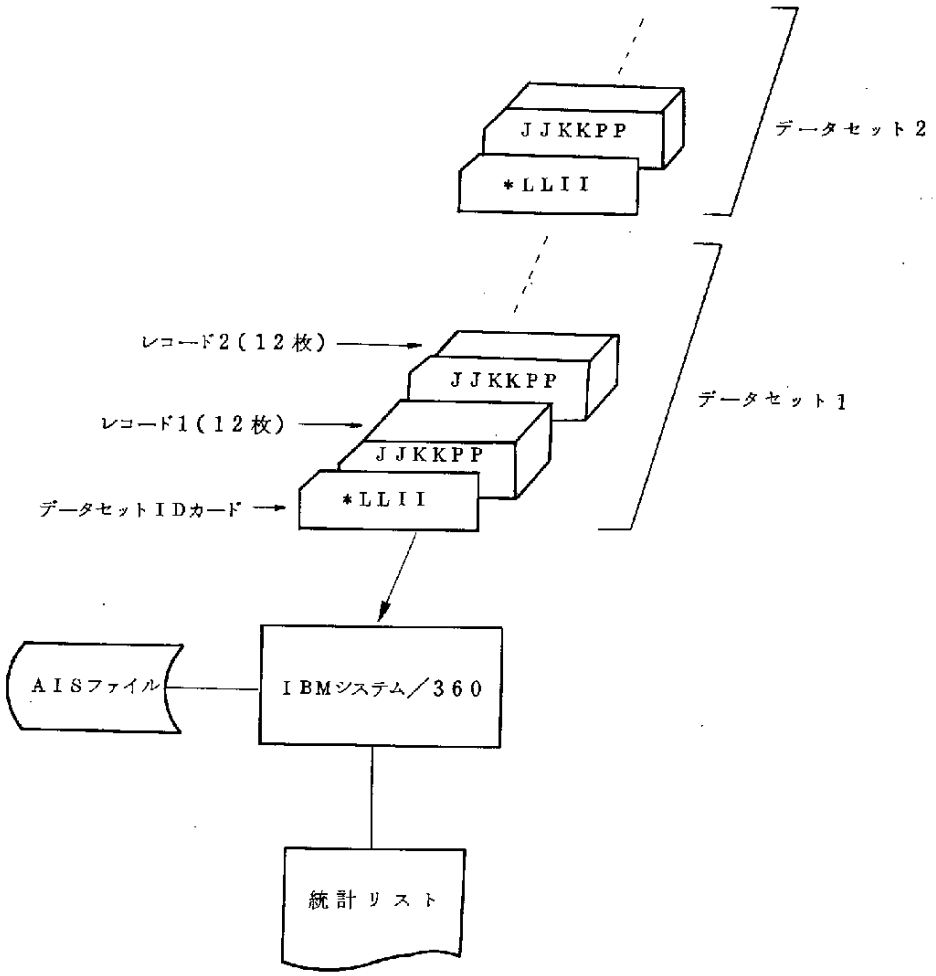


図 2.3.1

2.3.5 検 索 例

Program product Profile を対象とした検索例を図 2.3.2 に示す。

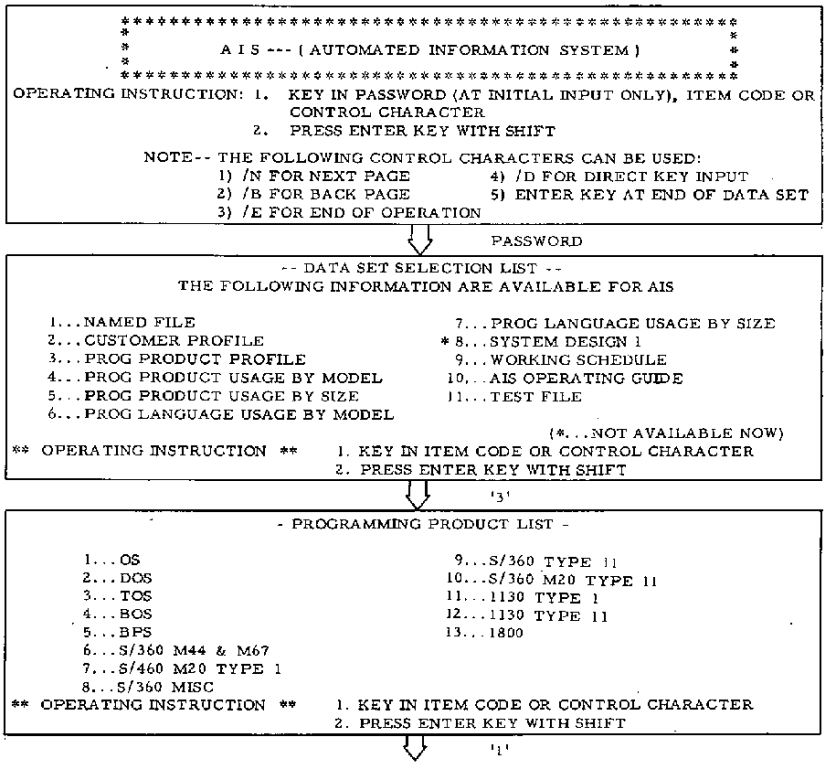


図 2.3.2　ＡＩＳの検索例

2.3.6 適 用 例

日本ＩＢＭ社は，社内用として次のファイルを使用している。

① Named File

プログラム・マニュアル類など名前をつけておいたもの。

② Customer Profile

③ Program Product Profile

④ Program Product Usage by Model

⑤ 〃 〃 〃 〃 Size

⑥ Program Languagc Usage by Model

⑦ 〃 〃 〃 〃 Size

⑧ Working Schedule File

⑨ AIS Operating Guide

⑩ Test File

参 考 文 献

1） 伊藤忠電子計算サービス株式会社

３６００／３８００計算機システム ＩＮＦＯＬ説明書

１９６８年9月 ８５Ｐ.

2） 深沢士郎：ＡＩＳ（Automated Information System）—情報検索処理システム，第１０回プログラミングシンポジウム報告集，Ｄ－１２～Ｄ－２３，１９６９

第３章　検索システム—(Ⅱ)

１９６０年代の前半は，どちらかというとＩＲの手法が開発された時代であるが，後半は実用ＩＲシステムの開発に特徴があるといえよう。ＮＡＳＡのＩＲシステムやＩＢＭ技術情報センター（ＩＴＩＲＣ）のＳＤＩシステムは，その代表的な例である。一方，ＭＥＤＬＡＲＳ，ＲＩＮＧＤＯＣ，ＣＡＳなどのように磁気テープを提供するサービスが次々と開始されてきた。

次の時代の特徴は，おそらくシステムのオンライン化であろう。現にオンライン化を検討中のところが少くない。しかし，まずバッチ処理で大量の情報を処理する実用的手法が確立されねばならない。ＩＲとくに文献検索の著しい特徴は，マスファイルを扱うことにあるからである。

本章では，すでに実用化されたシステムおよび実用化を目前にひかえたシステムについて，情報の量と種類，検索の手法とハードウエア，費用と人員などを知れる限りまとめてみた。

多くの資料に眼を通したが，主なものは次のとおりである。

ＮＳＦ（米国科学財団）レポート類

Directory of Computerized Information in Science & Technology

(Science Associates International, Inc.)

海外視察団報告書

（ほとんど毎年，日本から視察団が欧米を訪問している）

専門雑誌

American Documentation

情報管理

ドクメンテーション研究

各種パンフレット

ＰＲ資料，カタログなど

3.1 MITのPROJECT TIP

3.1.1 概　　要

マサチューセッツ工科大学（MIT）では，ＭＡＣ計画の一環としてＰＲＯＪＥＣＴ　ＴＩＰというオンラインＩＲシステムを開発した。これは１９６６年１月から実際に動いている。用途は研究目的のためである。対象は物理学雑誌３２誌であるが，これらは雑誌間の相互引用のマトリックスに基い

て選定されたものである。TIPはTechnical Information Programの略である。

3.1.2 ファイル

1967年現在，デイスクに30,000件，磁気テープに30,000件収録されている。いずれはすべてディスクに蓄積される。シリアルファイル構成である。

収録項目は，

雑誌名

巻

ページ

標題

著者（複数）

機関

引用文献

である。

3.1.3 検　索

3.1.3.1 検索タグ

検索の手がかりとなる検索タグは，

雑誌名

巻

標題中のキーワード

著者

機関

引用文献

である。

3.1.3.2. 手　　法

(1) 範囲指定

探索すべき蓄積範囲を指定する。

命　令	探　索　範　囲
SEARCH:ALL	蓄積全体
SEARCH:ALL NEW	各雑誌の最近号
SEARCH:PHYREV	Physical Reviewの全号
SEARCH:PHYREV V120 to V125	〃　〃　Vol 120～125
SEARCH:PHYREV V120	〃　〃　Vol 120
SEARCH:JOURNAL LIST	雑誌所蔵目録

(2) 論理関係

1命令文中に，AND，OR，NOTの論理関係を組合せて使用できる。

3.1.4 計算機システム

IBM　7094　　1台

IBM　1050　150台（電話回線を使用）

3.2 SMARTシステム

3.2.1 概　　要

SMART (Salton's Magical Automatic Retriever of Texts) システムは，完全自動化情報検索システムの研究，検索技術の評価を目的としている。従って種々の手法を併用しているのが特徴である。

はじめ，Harvard 大学で実施されたが，プロジェクトリーダのG. Salton の移動に伴ない1965年からCornell 大学に移管された。

3.2.2 ファイル

磁気テープベースのシリアルファイルである。ファイルの種類は1.2.2参照。

3.2.3 検　　索

このシステムで選択的に使える各種の処理（検索）方式を，次のようにまとめることができる。1.2.2で詳細に説明されている。

① 辞書を使わない
② 統計的相関による同義語
③ シソーラスによる術語のグルーピング
④ 概念体系参照
⑤ 統計的句処理
⑥ 構文分析による句の処理
⑦ 質問と文献の相関
⑧ 文献の群別（クラスタリング）
⑨ 適合性フィードバック（システムとの相互通信）

検索タグは，標題，抄録，全文である。

3.2.4 計算機システム

3.2.4.1 ハードウエア

Harvard大学 ：IBM7094

Cornell大学 ：CDC1604

3.2.4.2　ソフトウエア

多くはＦＯＲＴＲＡＮでプログラムされている。

3.2.5　そ　の　他

3.2.5.1　人　　員

１９６７年現在，スタッフは次のとおりである。

Ｃｏｒｎｅｌｌ 大学　　１３

Ｈａｒｖａｒｄ 大学　　５

3.2.5.2　予　　算

１９６５年の報告によれば，年間予算は７５,０００ドルである。

3.2.5.3　今後の予定

(1)　ランダムアクセス大容量記憶装置を設置する。

(2)　リアルタイム処理

(a)　会話システム

(b)　ディスプレイシステム

(c)　システムパラメータのモニタリング

レスポンスタイム，検索時間，待ち時間など

(d)　同時処理

(e)　検索タスクのバックグラウンド処理

〔例〕　ＳＤＩ

(f)　事項検索をも可能にする

3.3 NASAのIRシステム

3.3.1 概　　要

NASA (National Aeronautics and Space Administration)は1962年から全世界の航空宇宙に関する包括的な文献サービスをはじめた。サービスは主として次の2機関によって提供されている。

(1) NASAは1962年からレポート文献の抄録誌STAR (Scientific and Technical Aerospace Reports)を発行。NASAの技術情報部としての仕事はNASAとの契約によりDocumentation Incorporatedが行っている。

(2) NASAから一部補助を受けて，AIAA (American Institute of Aeronautics and Astronantics) がレポート以外の文献に関する抄録誌IAA (International Acrospace Abstracts) を発行している。STARおよびIAAは，その内容が磁気テープに入れられ，ユーザーに提供されている。NASAのIRシステムは，この磁気テープをファイルとして使用するシステムである。

3.3.2 ファイル

3.3.2.1 情報源

航空宇宙工学関係の世界各国のレポート，雑誌，図書およびその他の文献を対象とする。年間約200,000万件を扱う。レポートはSTARに，レポート以外の文献はIAAに収録される。

3.3.2.2 ファイル構成

linear-file 形式をとる。各レコードは三つの部分からなる。

(1) 固定長データフィールド：各文献について基本データを収録するもので，文献番号，文献および標題の機密分類，主題カテゴリー，使用言語，文献の種類などがこれに入る。

(2) レラテァブ・イメージ：次の可変長データフィールドの相対位置を示す"ロケーションテーブル"である。

(3) 可変長データフィールド：可変長，可変形式のデータが収録されており，その相対位置はレラティブイメージで参照できる。収録するデータは各文献について，機関，標題，著書，レポート番号，契約番号，索引語（またはそのコード）などである。

3.3.3 検　　索

3.3.3.1 サービスの種類

(1) Q－A

(2) SDI

3.3.3.2 検索タグ

(1) Q－A：

文献の機密分類

文献番号の範囲

資料の種類

文献の種類

COSATIの主題カテゴリー

抄録誌の主題カテゴリー

機関著者

契約番号

個人著者

レポート番号

国語（英語かどうか）

キーワード

である。

(2) SDI：

キーワード

契約番号

国語

である。

3.3.3.3 手　　法

AND・OR・NOT

重み

語幹探索

パーセンテージ・マッチ（質問とファイルの両プロファイルの一致率によりHit かどうかをきめる）

3.3.4　計算機システム

3.3.4.1　ハードウエア

ＩＢＭ　３６０Ｍｏｄｅｌ　４０

ＩＢＭ　１４０１

ＩＢＭ　１４０３

ＩＢＭ　２３１１

磁気テープ　　１８巻

８００ｂｐｉ，９トラック

3.3.4.2　ソフトウエア

ＩＢＭ１４１０Ａｕｔｏｃｏｄｅｒ

ＰＬ／１

3.3.5　その他

１９６５年の米国視察チームの報告によれば，予算は年額５００万ドル，人員は３００名とのことである。ただし，予算の大部分は契約者のＤｏｃｕｍｅｎｔａｔｉｏｎ　Ｉｎｃｏｒｐｏｒａｔｅｄに支払われ，人員も大半を同社が占めている。

3.4 IBM技術情報センター（ITIRC）の検索システム

3.4.1 概　　要

ITIRC (IBM Technical Information Retrieval Center)は，IBM全体の技術情報サービスのための中央処理機関で，米国本土および海外諸国のIBM機関に対し，情報サービスを提供している。ITIRCの情報処理システムが，IBMで開発されたNormal Text IR System である。

サービスとしては，遡及探索およびSDI（IBMはCurrent Information Selection＝CIS と呼んでいる）が実施されている。CISのサービス対象は2000～3000名である。

3.4.2 ファイル

シリアルファイル形式で，磁気テープに以下の項目が収録されている。

抄録

著者

出典

文献番号

Normal Text Program によって，語は，文中の位置，大文字小文字の区別，語のチェイン（次の語を示す）がつけられ，ABC順，および語の長さの順に配列されている。これによって全文探索が実行される。

ファイル作成にあたっては，マスター辞書テープとインプットデータとを照合して，綴りチエックその他の編集を施している。

1967年現在で，IBM資料，DDC資料，大学のレポート，IBMプロジェクトファイル，100以上の社外雑誌，IBM発明報告，IBM規格および多くの雑資料について，合計350,000件以上がITIRCのファイルにある。

シソーラスははじめIBMのシソーラスを使用したが，現在はEJCシソーラスを使うことになっている。

3.4.3 検　　索

以下のような手法を用いる

論　　理	定　　義	例
INDIVIDUAL WORDS	どのような語も探索してよい	INFORMATION
OR	同　等	IR OR INFORMATION RETRIEVAL
AND	組合せ	INFORMATION AND RETRIEVAL AND FID AND TOKYO
ADJACENT WORDS	語の並置	BLIND VENETIAN—VENETIAN BLIND DINNER AT SIX—DINNER FOR SIX
NOT	否　定	NOT FID AND ITALY
SECURITY	秘密保持：社内秘資料の提供を有資格者のみに制限	IBM CONFIDENTIAL
WITHIN	所在の指定	著者名でなく標題中のI. R.
ABSOLUTE YES	でなければならない	ALL FID REPORTS AND 1967
MASKING	切　断	MICROF$はMICROFILM, MICROFORM, MICROFICHEを得るが, MICRODOTは得られない
CONCEPTS	論理の組合せ	

3.4.4 計算機システム

3.4.4.1 ハードウエア

IBM 360 Model 50

3.4.4.2 ソフトウエア

中核となるプログラムはNormal Text Program（標準テキストプログラム）である。

3.5 スミソニアン研究所のＳＩＥ

3.5.1 概　　要

米国のスミソニアン研究所は，約２０年前からScience Information Exchange（ＳＩＥ）を行っているが，処理量の増加に伴ない順次機械化をすすめてきた。

本システムは以下に示すように研究情報を扱うものであって，文献検索を行なうものではない。

ＳＩＥの業務概要

(1) 未出版の研究（進行中の研究）に関する情報を扱う。

(2) 生物学，物理学，社会科学，工学を対象とする。

(3) 情報の蓄積・検索のみならず，研究管理業務に重点が置かれている。

3.5.2 ファイル

年間１００,０００件の研究計画または進行中研究に関する記録は，各1ページに納められる。これはNotice of Research Project（ＮＲＰ）と呼ばれる。各記録には次の項目が記載されている。

研究後援者

研究担当者

所在地

期日

予算

抄録（２００語）

索引

このＮＲＰは，フルサイズのハードコピー（２年経過するとマイクロフィルム化）で保存されるが，それとともにコード化されて計算機ファイルとなる。ファイル構成は不明であるが，おそらくシリアルファイルであろう。

3.5.3 検　　索

詳細は不明であるが，ユーザからの質問を受けて年間５５,０００件のアウトプットが出されるという。過去数年間で最も多い需要は，氏名調査，その次は主題別調査である。

3.5.4　計算機システム

１９５０年　ソータ

１９５８年　ＩＢＭ１４０１

１９６１年　ＩＢＭ１４６０

１９６６年　ＩＢＭ３６０　Ｍｏｄｅｌ ３０

3.5.5　そ　の　他

コスト分析の報告によれば，年間５５,０００件のアウトプットはその種類によりばらついた値を示している。

〔例〕　氏名１名あたり　　1.7ドル

　　複合主題調査　　４５ドル

　　複雑データの編集　　１００ドル以上

3.6 MEDLARS

3.6.1 概　　要

MEDLARS (Medical Literature Analysis and Retrieval System) は米国NLM (National Library of Medicine) が開発した情報検索システムで，1964年1月から実際に使用されている。

NLMは医学文献の二次資料作成に関しては長期間にわたる実績を持っている。1879年に始まり1927年まで続き，一時中断されたが1960年に再発行されたIndex Medicusをはじめ多くの二次資料を発行している。

長い歴史を持つIndex Medicusの発行を電子計算機により行なうためにはかなり長期間にわたる注意深いシステム設計が必要であった。そのため，1960年から1963年は当時のシステムの一部を紙テープ穿孔装置，カード穿孔装置などを利用して機械化し，MEDLARSのシステム設計のための基礎的な情報を与えること，MEDLARSのデータ・ベースを用意することなどが行なわれた。

このように3年間にわたる綿密な計画と開発の末，1964年からこのシステムは実際に稼働を開始したわけである。

MEDLARSではNLMのほかに，MEDLARS Search Centerと称する地域センターが，下記のとおり設置されている。（このほかにも設立ずみ，または計画中のところ数カ所あり）

米国	東北部	ハーバード大学
	中西部	ミシガン大学
	中南部	アラバマ大学
	太平洋岸	カリフォルニア大学（ロサンジェルス）
	ロッキー山脈	コロラド大学
英国	ニューカッスル・アポンタイン大学	
	国立科学技術貸出図書館	
スウェーデン	カロリンスカ研究所	
オーストラリア	シドニー大学	

日本では未定であるが，蓄積（索引作成）については，すでに慶応義塾大学が作業に協力している。

3.6.2 ファイル

シリアル・ファイル方式で，約2,３００の医学雑誌から年間約１８万件を収録している。収録されるのは，書誌事項と件名標目である。現在，７５万件以上の文献が蓄積されている。

主な収録項目は次のとおりである。

著者名

英文標題

原文標題（翻字）

雑誌名

使用言語

タグ（Medical Subject Headings からの件名標目またはそのほかの索引語）

巻数およびページ

発行年月日

インプット日

3.6.3 検　　索

3.6.3.1 検索タグ

探索に用いる要素としては，MeSH から選んだ件名標目が大きなウエートを占めるが，このほかにも表に示すように多くの要素が使用できる。

表3.6.1　ＭＥＤＬＡＲＳの探索要素

要素の記号	要　　素　　名
M	Subject Heading
S	Subheading
G	地名のHeading
F	様式のHeading
T	print指定のSubject Heading
Z	non-print指定のSubject Heading
C	MeSH階層でのカテゴリー番号
A	著者名
J	雑誌名コード
I	計算機に入力された日付
L	言語の略号
Q	発行地名
Y	発行年
X	Subject Heading と Subheadingとの組合せ

3.6.3.2 手　法

通常の論理関係（ＡＮＤ，ＯＲ，ＮＯＴ）と大小関係（≧ ≦）を使用する。ファイルが膨大なので，探索スピードをはやめるために，基底要素テーブルと文献ファイルとの照合によって粗い探索を実施し，検索結果を中間テープにはき出し，この中間テープと決定表との照合によって詳しい探索を実行する。この基底要素テーブルおよび決定表は，探索プログラム中の編集セグメントによって作成される。

基底要素テーブルは，探索速度をはやめるための"スクリーン検索"にあたるもので，次の条件をみたさなければならない。

(a) 少くともその要素を含まなければ，文献は探索要求を満たさない。

(b) このような要素がいくつか論理積の形で結合されているときには，それらの中で使用頻度が最も低いもの

〔例〕

論理式　（Ａ＊Ｂ）＋（Ｃ＊Ｄ＊Ｅ）

使用頻度　Ａ　６２９

Ｂ　１,０１６

Ｃ　５４３

Ｄ　８１６

Ｅ　４３９

基底要素　ＡおよびＥ

決定表は，論理式をリスト表現したもので，これも探索速度をはやめるためのものである。

表3.6.2　決定表の論理

例；　Ｍ１＊（Ｍ２＋Ｍ３＋Ｍ４）　　＊　；　ＡＮＤ　＋　；　ＯＲ

探索要素	非適合コード	適合コード
Ｍ１	5	0
Ｍ２	0	3
Ｍ３	0	3
Ｍ４	5	3

適合コードおよび非適合コードの説明

0　＝　次の要素を調べる。

3　＝　この文献は条件を満たす。

5　＝　この文献は条件を満たさない。

簡単に説明すれば，Ｍ１のとき適合しなければ駄目，Ｍ１を満たしたときには，Ｍ２，Ｍ３，Ｍ４のいずれかを満たしていればよいことが判る。

3.6.4　計算機システム

3.6.4.1.　ハードウエア

種　類	機　器　名	製　造　会　社	台数
入力機器	フレキソライター	Friden, Inc	15
	024　カード穿孔装置	IBM	3
	026　カード穿孔装置	IBM	1
	056　カード検査装置	IBM	2
	557　翻訳機	IBM	1
	082　カード分類機	IBM	1
	カードタブファイル	Diebold, Inc	1
計算機	H－800	Honeywell	1
	中央演算制御装置		
	磁気テープ制御装置		
	3／4インチ磁気テープ装置(7)		
	コア記憶装置8192語（48ビット）		
	H－200	Honeywell	1
	中央演算制御装置		
	磁気テープ制御装置		
	1／2インチテープ装置(2)		
	コア記憶装置8192文字（6ビット）		
	高速印刷装置，制御装置（900行／分）		
	紙テープ読取装置，制御装置		
	カード読取／穿孔装置，制御装置		
	H－800との記憶装置間でのオンラインアダプター		
出力機器	“900”－磁気テープベースのPhototypesetter	Photon	1
	Versamat自動フィルム処理装置	Eastman-Kodak	1

3.6.4.2　ソフトウエア

ＡＲＧＵＳ（Ｈｏｎｅｙｗｅｌｌ　Ｈ−８００　のアセンブリ言語）

3.6.5　そ　の　他

3.6.5.1　人　員

１９６８年1月現在でＭＥＤＬＡＲＳの稼働および管理のために９０人のスタッフが従事している。これらの他に目録作業および技術サービスのための人員が２５名（専門職１５名，事務員１０名）いる。

図にセクション別の人員構成が示してある。ここに掲げたものはＭＥＤＬＡＲＳに直接従事しているものだけで，当然ＮＬＭ内には密接にこれらスタッフと協力しながら働いているセクションがいくつかある。

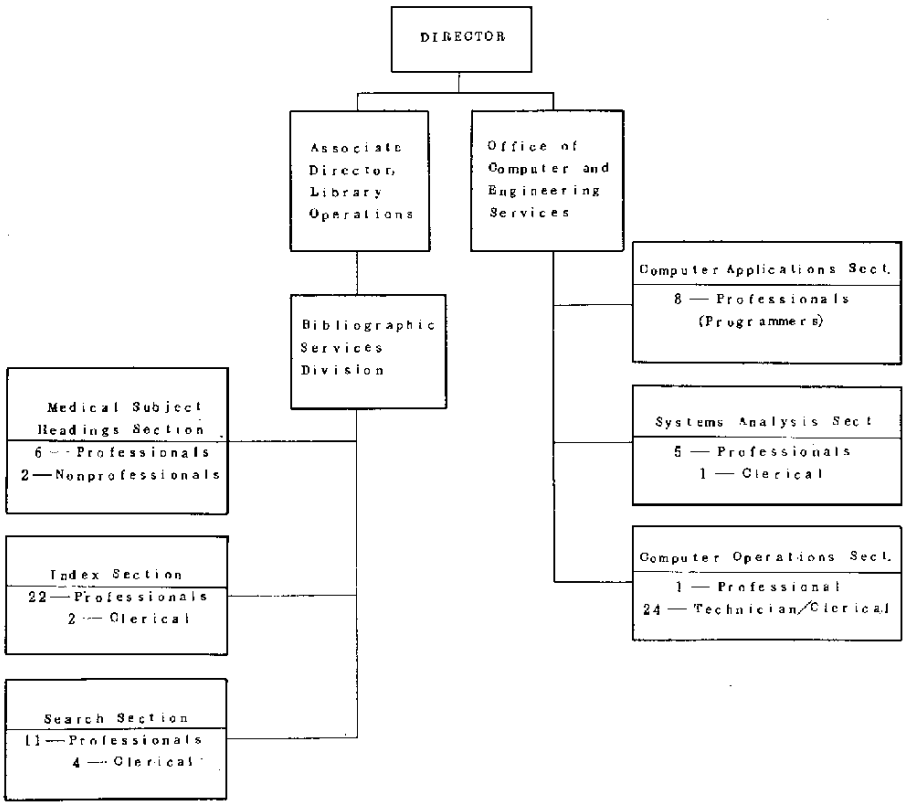


図3.6.1　ＭＥＤＬＡＲＳの人員構成

3.6.5.2. 処理件数

(1) 文献ファイル（１９６８年１月現在）

入力	７００件／日
蓄積量	６４５,７５１件
磁気テープ	２５巻
対象雑誌	2,３００種

(2) 探索質問

１９６４	1,０００件／年		
１９６５	2,０００		
１９６６	3,０００		
１９６７	5,０００	ＮＬＭ本部 地域センター	4,０００ 1,０００
１９６８	8,０００	ＮＬＭ本部 地域センター	3,０００ 5,０００

3.6.5.3 今後の予定

１９６９年７月からＭＥＤＬＡＲＳⅡと呼ばれる第２段階に入ろうとしている。そのためにComputer Science CorporationがＣＯＳＭＩＳ (Computer System for Medical Information Service)というプログラムを開発した。

また，ＲＩＳＣ (Remote Information System Center) 計画を実行し，オンライン検索を行うことを考えている。

3.7 CASのIRサービス

3.7.1 概　　要

60年の歴史をほこるCAS (Chemical Abstracts Service) は，そのサービスを一層充実すべく，1969年夏を目標に着々と機械化をすすめている。CASのねらいは，主題分析は人間にまかせ，以後の多角的処理は機械に引受けさせることである。CASでは，それを，Single Analysis／Multiple Use とよんでいる。

3.7.2 ファイル

CASは以下のようなファイルを製品として提供するとともに，このファイルによる検索サービスを引受けている。また，これらのファイルに収録される有機化合物はCompound Registry System (Atom-by-Atom Topological Technique を利用した化合物登録システム) によって管理されている。

（以下に示す件数は，1967年現在の概算）

(1) 速報サービス ── 冊子体および磁気テープ

(a) CHEMICAL-BIOLOGICAL ACTIVITIES

有機化合物の生物活性に関するダイジェスト誌。約46,000件。

(b) CHEMICAL TITLES

新着化学文献のKWIC索引。約614,000件。

(c) POLYMER SCIENCE & TECHNOLOGY

高分子の文献および特許のダイジェスト誌。約20,000件。

(2) 速報サービス ── 磁気テープ

(a) BASIC JOURNAL ABSTRACTS

化学および化学工学の基本雑誌35種を対象とした抄録テープで冊子体もつく

(b) CA CONDENSATES（冊子体なし）

CHEMICAL ABSTRACTS誌の全記事について，標題 書誌事項 キーワード索引語を収録した磁気テープ

(3) 索引 ── 磁気テープ

(a) CA Patent Concordance

同一内容の各国の特許番号（CAに掲載されたもの）の対照表

(4) 参考資料 ── 磁気テープおよび冊子体

(a) COMPREHENSIVE LIST OF PERIODICALS FOR CHEMISTRY AND CHEMICAL ENGINEERING

CAの収録雑誌に加えてBEILSTEIN'S HANDBUCH DER ORGANISCHEN CHEMIEやCHEMISCHES ZENTRALBLATTおよび19世紀の化学雑誌など，雑誌25,000点，モノグラフ5,300点の所蔵機関つきリスト

3.7.3 検　　索

ファイル別に検索タグ（検索アイテムの種類）と手法をまとめる。

ファイル ＼ 検索	検索タグ						手法			検索対象
	主題語	雑誌名(CODEN)	著者名	化合物名	分子式	化合物番号	語の部分マッチ	論理式	重み	
CBAC	○	○	○	○	○	○	○	○	○	標題，ダイジェスト
CT	○	○	○	○			○	○	○	標題
POST	○	○	○	○	○	○	○	○	○	標題，ダイジェスト
BJA	○	○	○	○	○		○	○	○	標題，抄録文（分子式を含む）
CA CONDENSATES	○	○	○	○			○	○	○	標題，キーワード索引
CA Patent Concordance	検索プログラム　なし									
COMPREHENSIVE LIST	検索プログラム　なし									

3.7.4 計算機システム

3.7.4.1 ハードウエア

IBM 1401　　1台

IBM 360 Model 50　　2台

IBM 2280　　1台

いずれIBM 360 Model 65 2台にレベルアップする予定である。

IBM 2280は，新しいモデルに代えられる。

3.7.5 そ の 他

3.7.5.1 人　　員

1968年現在の人員は次表のとおりである。

Approximate Number of Person of CAS

I Library	Librarian	15
	Chemist	5
II Research and	Developement	
	System Analyst	10
	Programmer	30
	Chemist	10
	Data Processing (Operator)	25
	Key Boarding	*215 (2 shifts)
III Abstract	Assignment	50
	Edit Organic	25
	Biochemical	15
	Physical	10
	Abstractor	40
IV Index	Formula	40
	Subject General	50
	Organic	50
	Biochem	20
	Alphabetize	*80
Total		ca. 1,000

〔注1〕 *印以外は学卒

〔注2〕 Key Boardingは2shiftsでないかもしれない。確認できない。

3.7.5.2 予　　算

		1960	1961	1962	1963	1964	1965	1966	1967 (予定)
予　算（100万ドル）		3.9	4.4	5.4	5.6	6.8	9.7	10.6	12.5
内訳（%）	業　務	89	91	91	89	88	74	79	79
	CASの研究開発	5	6	5	5	6	7	6	6
	研究開発の補助金	6	3	4	6	6	19	15	15

3.7.5.3　今後の予定

オンライン化とダイレクトアクセス記憶装置の使用をはかる。１９７１年までに化合物登録システムは３００万種の化合物に達し，抄録は４０万件／年，索引の見出しは３００万件／年に増加するであろうから，ファイルの分割やスクリーンの使用など効果的な蓄積方法を研究中である。

処理機能によってパッケージ化し，パラメータ指定によって使い分けるようにする。

計算機がもつデータベースからハンドブックやコンペンディウムを編集することも計画している。

3.8 LCのPROJECT MARC

3.8.1 概　　要

米国議会図書館（ＬＣ）は，図書資料の目録データを機械に読める形式に変換し，それを加盟図書館に配布し，変換のフォーマットと方法，配布法，データの有効性などを調査するプロジェクトに１９６５年１２月から取組んでいる。現在，MARCⅡ（Machine-Readable Cataloging）とよばれている。はじめ以下の１６図書館が参加し，１９６７年中頃で約１６,０００件が収録されたが，現在，参加図書館は２０をこえ，データも増加している。

Argonne National Laboratory
Univ. of California
Univ. of Chicago
Univ. of Florida
Georgia Inst. of Technology
Harvard Univ.
Indiana Univ.
Univ. of Missouri
Montgomery County Public Schools
Nassau (County) Library System
National Agriculture Library
Redsone Scientific Information Center
Rice Univ.
Univ. of Toronto
Washington State Library
Yale Univ.

3.8.2 ファイル

4種類のファイルがある。

File 1　目録レコード
File 2　著者／標題レコード
File 3　件名相互参照トレーシングレコード
File 4　記述相互参照トレーシングレコード

3.8.2.1 File 1

シリアルファイルであって，各レコードは固定長フィールドと可変長フィールドに分けられる。可変長フィールドにはデータが，固定長フィールドにはデータ（目録レコード）に関する情報が収録される。

3.8.2.2 File 2

固定長形式のシリアルファイルで，File 1から，著者／標題を固定長化して抽出，作成したものである。

3.8.2.3. File 3，4

相互参照データを収録した可変長形式のファイルで，File 3とFile 4は同一形式である。

3.8.3 検　　索

目録作成を主要目的としたファイルなので，とくに検索プログラムは用意されていない。

3.8.4 計算機システム

3.8.4.1 ハードウエア

参加図書館は，IBM1400シリーズ，7000シリーズ，システム／360がいろいろあるが，LCにはIBM360 Model 30Dが設置されている。

3.8.5 そ の 他

3.8.5.1 人　　員

Project MarcにはLCの多くの部局が協力している。そのほか，フルタイムの職員としては，1966年10月の報告によれば，以下のとおりである。

編集担当（主任目録員）	1
編集補助	2
タイピスト	4
オペレーション要員	1
オペレーション主任	1

3.8.5.2 費　　用

1966年10月の報告によれば，以下のとおりである。

Council on Libray Resources の補助金……10万ドル

LC基金 ……相当額

3.9 ENGINEERING INDEXのPROJECT CADRE

3.9.1 概　　要

米国のENGINEERING INDEX社は，PROJECT CADRE (Current Awareness and Document Retrieval for Engineers) というパイロットプログラムを開発した。（1968年中，テスト）

ENGINEERING INDEX 社は，工学全般を対象とする抄録誌Engineering Indexを発行しているが，本プロジェクトではさしあたり，Engineering Index誌中の電気工学／電子工学部門とプラスチック部門とが対象分野として選択された。

3.9.2 ファイル

(1)マスターファイル，(2)ディスクリプタファイル，(3)辞書ファイルの三つがある。

(1) マスターファイル

シリアルファイル形式で磁気テープに以下の項目を収録する。セグメント化されている。

索引語

ロールまたはサブディスクリプタ（修飾語）

リンク

索引語コード

著者（10名まで）

標題またはNotation Of Contents(N.O.C.フルタイトル)

出典

件名標目

(2) ディスクリプタファイル

インバーテッドファイル形式で，磁気テープにディスクリプタと文献番号が収録されている。

(3) 辞書ファイル

管理ファイルとして辞書ファイルがある。これは，Engineering Index Thesaurus の部分集合をなすもので，以下のものが収録されている。

ディスクリプタ

サブディスクリプタ

同義語

スコープノート（注記）

すべてのディスクリプタおよびサブディスクリプタは，マスターファイルに入る前に辞書ファイルと照合される。

3.9.3 検　　索

3.9.3.1 検索タグ

索引語（ディスクリプタ）

ロールまたはサブディスクリプタ（修飾語）

リンク

索引語コード

3.9.3.2 手　　法

まずディスクリプタファイルをさがし，質問ディスクリプタを含む文献の番号を求める。ついでその文献番号によってマスターファイルを詳細に探索する。

ブール探索が行える。

3.9.4 計算機システム

3.9.4.1 ハードウエア

IBM 1401　　12K

磁気テープ　　4台

3.9.4.2 ソフトウエア

Combined File Search（IBMライブラリー・プログラム）

Autocoder Assembler

3.10 ISIの情報サービス

3.10.1 概　　要

アメリカの民間情報機関ISI(Institute for Scientific Information)は，1960年から情報サービスの機械化に着手している。

主要サービスは以下のとおりである。

CURRENT CONTENTS……目次速報

ASCAⅢ……SDIサービス

SCIENCE CITATION INDEX……引用索引

PERMUTERM SUBJECT INDEX……索引誌

INDEX CHEMICUS……化合物の図式抄録誌

ENCYCLOPAEDIA CHIMICA INTERNATIONALIS

……INDEX CHEMICUSの累積版

ISI SEARCH SERVICE……検索サービス

ISI MAGNETIC TAPES……データテープの提供

ORIGINAL ARTICLE TEAR SHEETS……原文提供

3.10.2 ファイル

中心となるファイルは，

Source Tape ……文献テープ

Citation Tape……引用文献テープ

である。

(1) Source Tape

1600種の科学雑誌から，毎週5400件の文献を収録する。可変長形式である。

収録項目は次のとおりである。

記事番号

第一著者

雑誌

巻

ページ

記事区分

引用文献数

号，補遺，分冊

雑誌の号ごとの受入れ番号

第二著者以下（9名まで）

標題

(2) Citation Tape

5400件の引用源文献から，毎週64,000件の引用文献を収録する。固定長形式である。収録項目は次のとおりである。

引用源文献：
- 雑誌略名
- 巻
- 始まりページ
- 記事区分
- 発行年下1ケタ
- 第一著者　（引用源著者）
- 第一著者　（被引用著者）

雑誌名（第一発明者，特許分類）

巻

ページ

発行年下2ケタ

雑誌であるかどうかの区分

引用源文献との対応番号

3.10.3 検　　索

Source Tape を使用して検索（SDI）を実行する。引用文献に関する検索ができるという特徴をもつ。

3.10.3.1 検索タグ

標題中の語（単語，語幹，句）

引用源文献（特定の文献を引用している文献を探す）

著者

機関

雑誌

3.10.3.2 手　　法

AND，OR，NOTを用いる。

3.10.4 計算機システム

3.10.4.1 ハードウエア

IBM 360

IBM 7044

IBM 1401

3.10.4.2 ソフトウエア

以下のようなプログラムが使われている。このプログラムもユーザに提供される。

(1) Source Tape について

(a) ソートコントロールカード（IBM7044）

(b) 索引リスト作成（IBM1401）

(c) SDI用のプログラム集（IBM7074，1401）

Autocoderで書かれている。

(d) SDI用のプログラム集（IBM360）

(e) SDI用のプログラム集（IBM1401）

(2) Citation Tapeについて

(a) ソートコントロールカード（IBM1401，360，7044）

(b) 引用索引作成（IBM1401）

3.10.5 そ の 他

3.10.5.1 人 員

1965年当時の報告によれば，124名の職員がおり，このうち約半分の60名は科学技術のバックグラウンドをもっている。

3.11 Derwent社のRingdoc

3.11.1 概　　要

英国のDerwent Publication Ltd.は，主として化学関係の情報サービスを種々実施しているが，薬学関係の情報サービスとして1964年7月にRingdoc システムを組織した。これは，会員制度によるシステムで，所定の費用を支払い会員として登録することにより，抄録誌，索引カード，パンチカードまたは磁気テープに収録した薬学情報およびそのほかの資料の提供を受けるものである。現在，全世界の会員は約80社であり，そのうち日本からは11社が参加している。費用は，基本費が年間3,250ポンドである。

3.11.2 ファイル

シリアルファイル方式で情報が磁気テープ（IBM1401）に蓄積されている。

収録項目は次のとおりである。

文献番号

分類（thematic groups）

キーワード（codeless scanning）——電文抄録形式のキーワード群。シソーラスによって統制されている。

書誌事項

著者

標題

3.11.3 検　　索

3.11.3.1 検索タグ

標題中のキーワードを含めてすべての収録項目が検索タグとなる。

3.11.3.2 手　　法

AND，NOT，ORの論理関係を使用する。

3.11.4 計算機システム

IBM 1401

IBM 1460

IBM 7094

3.11.5 そ の 他

3.11.5.1 人 員

Derwent 社の人員構成は次のとおりである（１９６８年現在）

総員　２１０

専門家　５６

事務系　８０

作業員　３３

（データ処理１７）

（印 刷　１６）

その他　４０

3.12 EXCERPTA MARK I SYSTEM

3.12.1 概　　要

オランダの Excerpta Medica Foundation では，3,000種以上の生物医学系雑誌から年間200,000件の記事を採択し，抄録誌"Excerpta Medica"や索引誌を発行しているが，システムの機械化を計画し，このほどExcerpta Mark I Systemを開発した。

3.12.2 ファイル

抄録誌"Excerpta Medica"の情報を磁気テープまたはランダムアクセスメモリ（NCR CRAM－5）に収録している。磁気テープは，シークエンシャルファイル，ランダムアクセスメモリはシークエンシャルファイルまたはランダムファイルである。

収録項目は次のとおり

製造番号

インプット年

著名（4名まで）

データのタイプ

雑誌名

発行年

発行国（雑誌の）

発表国（記事の）

使用言語（記事の）

英文標題

所属機関

原文標題（英文以外）

補足情報（巻数，シリーズ数など）

Codenコード

分類 { セクション番号（医学の専門分野）
　　チャプター番号（複数個。専門分野の細分）

索引 { 重要索引語
　　補助索引語（重要索引語に対する修飾語）
　　アイテム索引（一種のカテゴリー）

抄録文

3.12.3 検　　索

3.12.3.1 検索タグ

次の2種類がある。

タイプ1：ファイル中の全項目

タイプ2：著者（複数）

雑誌のCodenコード

重要索引語（primary indexing terms）

アイテム索引語

分類

3.12.3.2 手　　法

AND，OR，NOTの論理関係を用いる。重要索引語については，計算機シソーラス（Master List of Medical Indexing Terms＝MALIMET）が同義語→標準索引語への変換を自動的に行なう。本シソーラスは，15万語からなり，標準索引語（preferred term）は6,000～7,000語含まれる。

3.12.4 計算機システム

3.12.4.1 ハードウエア

中央処理装置	NCR315－501 RMC
オンライン・メモリ	20K NCR 316／504
バッチ・メモリ	20K NCR 316／505
磁気テープ装置	33KC
制御装置つき	NCR 334／131
磁気テープ装置	3×NCR334／132
CRAMユニット	4×CRAM5，NCR353／5
スイッチング計算機	NCR321／3
アダプタ・ゲージ	2台
RCA Video Comp	
電話回線	100本

3.12.4.2 ソフトウエア

(1) 計算機シソーラス（MALIMET）に基づくシステム・スーパバイザ

(2) ハードウエアの各部分，I／Oルーチン，探索質問およびSDIサービス用の検索オペレーションを結合する各種のプログラム

(3) オンライン検索用のプログラム

3.12.5 そ の 他

3.12.5.1 人 員

生物医学専攻の専門スタッフ	80
編集スタッフの補助	195
編集部に協力する生物医学専門家（部外）	700
抄録者（部外）	4,000
製造 管理 システムアナリスト プログラマ 計算機操作員	230

3.12.5.2 処理件数

生物医学系雑誌（スクリーン対象）	3,000
雑誌冊数（スクリーン対象）	20,000／年
記事数（蓄積および検索）	200,000／年
抄録数（作成および蓄積）	80,000／年
抄録数（出版）	140,000／年
マイクロフイルム化したページ数	2,000,000／年
マイクロフイルム総ページ（1960年以後）	18,000,000
月刊抄録誌数	33
分類表中のカテゴリー数	3,000

3.13　PANDEX社の情報サービス

3.13.1　概　　要

米国のPANDEX　INC.は，1968年から磁気テープサービスを開始している。対象分野は，科学技術および医学で，情報源は下記のとおり。

図書	6,000点／年　以上
雑誌	2,100種／年　以上
特許	5,000件／年　以上
政府レポート	35,000冊／年　以上

3.13.2　ファイル

可変長形式のシリアルファイル（磁気テープ）である。収録項目は次のとおり。

(1) 雑誌の場合

発行年
雑誌略名
巻
号
ページ
主題
著者
標題

(2) 図書の場合

発行年
発行所
版次
L. C. カード番号
ページ
価格
主題
著者
標題

3.13.3 検　　索

詳細は不明であるが，ＳＤＩ（週単位），遡及探索の受託サービスと共に，データテープとプログラムを提供するようになっている。

3.13.4 計算機システム

3.13.4.1 ハードウエア

ＩＢＭ３６０

3.13.4.2 ソフトウエア

ＣＯＢＯＬ

3.14　JICSTのIRシステム

3.14.1　概　　要

日本科学技術情報センター（ＪＩＣＳＴ）は，１９６７年１２月にＦＡＣＯＭ２３０－５０を導入し，情報処理業務の機械化を図った。

文献速報自動作成システム …… １９６８年末から稼働開始

漢字モードＩＲシステム …… テスト中

ＢＣＤモードＩＲシステム …… テスト中

用語管理システム …… 設計中

3.14.2　ファイル構成

主要なファイルは次のとおりである。

イ．文献速報自動作成システム …… 文献速報ファイル

ロ．漢字モードＩＲシステム …… ＩＲ漢字ファイル　（1.3.5.2参照）

ハ．ＢＣＤモードＩＲシステム …… ＩＲ　ＢＣＤファイル（1.3.5.2参照）

ニ．用語管理システム…… 用語ファイル

イとロは，ほとんど同じファイル構成である。ニはまだ発表されていない。なお，オンライン検索では，ＩＲ　ＢＣＤファイルからインバーテッドファイル（磁気ディスク）を作成することになっている。

(1)　ＩＲ漢字ファイル

セグメント方式のシリアルファイル（磁気テープ）である。ＪＩＣＳＴの定義によればセグメント方式は，１文献を複数個の異なるセグメントで表わし，１セグメントを複数個のレコード（各レコードは固定長）で表わす。固定長形式と可変長形式の中間である。各セグメントはそれぞれ１個ないし複数個の収録項目に対応している。

データは漢字コード（１２ビット＋ファンクションコード４ビット）で，以下の項目が収録されている。

セグメントＩＤ	内　　容
A	原稿仏など
B	文献の形式，書誌事項（ページを除く）など
C	分類コード，ＵＤＣ
D	ページ
E	著者名，同ふりがな
F	原文標題
G	和文標題
H	キーワード，同ふりがな，キーワード区分
I	抄録文

(2) ＩＲ　ＢＣＤファイル

これもセグメント方式のシリアルファイルである。セグメント体系は，ＩＲ漢字ファイルの場合と異なる。近くインプットが開始される予定である。

	セグメントＩＤ	セグメント名	内　　容
	B	管理情報	文献の形式に関する情報
	C	資料名	雑誌名，レポート名など
	D	巻，号	発行年，巻，号
	E	整理番号	レポート番号
	F	ページ	ページ
*	G	発行所	発行所
	H	著者	著者名
*	I	所属機関	著者の所属機関
	J	欧文標題	英語またはローマ字化標題
*	K	和文標題	カタカナ化標題
*	L	抄録文	抄録文
	M	分類コード	分類コード，ＵＤＣ
	N	キーワード	キーワード
*	O	補足キーワード	追加インプットのキーワード
*	P	ソースファイル参照	ソースファイルの出典

＊印は，データがインプットされていない。

3.14.3 検　　索

3.14.3.1 検索タグ

	漢字モード	BCDモード
資料区分	○	○
記事区分	○	○
記事番号	○	○
資料番号	○	○
発行国	○	○
使用言語	○	○
発行年	○	○
巻	○	○
号（整理番号）	○	○
著者	○	○
分類コード	○	○
UDC	○	○
キーワード（索引語）	○	○
キーワード（標題中）	○	○
所属機関		○

3.14.3.2 手　　法

AND，OR，NOTの論理関係に加えて，重みを用いる。マッチ方式として，完全マッチ，部分マッチ（前方，後方，中間）の使い分けが許される。発行年，巻などは大小関係が利用できる。

3.14.4 計算機システム

3.14.4.1 ハードウエア

FACOM230-50	1台（65KW）
磁気ドラム装置	1台
磁気ディスク装置	1台
磁気テープ装置	8台
カード読取装置	1台
紙テープ読取装置	1台
ラインプリンタ装置	1台
コンソールタイプライタ	1台

ＦＡＣＯＭ２７０－２０	１台（１６Ｋｗ）
磁気テープ装置	２台
ファコムライタ	１台

ＪＥＭ－３８００漢字プリンタ

ソフトプリンタ	２台
フイルムセッタ	２台

3.14.4.2 ソフトウエア

大部分，ＣＯＢＯＬで書かれている。

3.14.5 そ の 他

実験的規模でＢＣＤモードＩＲシステムのオンライン化を開発中である。

3.15 電気通信研究所のREWDAC

3.15.1 概　　要

日本電信電話公社・電気通信所研究所は，所内情報サービスのためにREWDAC（REtrieval by title, Words, Descriptors, And Classification）システムを開発し，実用に供している。

システムの開発は1963年，標題速報誌（海外文献リストという）は1964年から，検索サービスは1966年から開始されている。

3.15.2 ファイル

シリアルファイル（磁気テープ）形式で以下の項目を収録する。

分類コード（3個まで）
著者（3名まで）
所属機関（3機関）
標題
雑誌名
巻，号，ページ
年月
文献の種類
国
UDC（インプットはされていない）
キーワード

蓄積対象は，海外学術雑誌190種から，月間約2000件を蓄積する。

3.15.3 検　　索

3.15.3.1 検索タグ

分類コード（5個まで）
キーワード（6個まで）{標題中のキーワード / 追加されたキーワード}
著者（2名まで）
所属機関（2機関まで）

3.15.3.2 手　　法

ＡＮＤ，ＯＲ，ＮＯＴの論理関係を用いる。逐字照合（ｃｈａｒａｃｔｅｒ　ｂｙ　ｃｈａｒａｃｔｅｒ）である。機関名は１字誤まりを許す。（１字ミスマッチでもｈｉｔ　とする）

3.15.4 計算機システム

3.15.4.1 ハードウエア

ＮＥＡＣ　２２０６　　（１９６６年まで）

ＮＥＡＣシリーズ　２２００　　モデル　４００　（１９６７年以降）

3.15.4.2 ソフトウエア

主なプログラムとして次のようなものがある。言語はアセンブラ，およびＣＯＢＯＬである。

(1) 文献の紙テープから磁気テープへの転送プログラム

(2) 分類項目別文献リスト作成プログラム

(3) 自然語による文献検索プログラム

(4) 機関別分類項目別文献数一覧表作成のプログラム

(5) 機関別文献リスト作成プログラム

(6) 磁気テープファイル作成プログラム

(7) 著者別検索プログラム

(8) さん孔紙テープの検査プログラム

(9) １字誤まりを許する機関名による検索プログラム

3.15.5 その他

3.15.5.1 費用

表* 文献リスト費用概算例（60種の雑誌の場合）

	費用	算出根拠
雑誌代	1,800,000円	30,000円×60種
電算機関係		（年間 12,000件）
打ちこみ	408,000	0.2円×170パンチ×12,000件
書きこみ	72,000	300円／分×20分×12ケ月
打ち出し	108,000	300円／分×30分×12ケ月
磁気テープ	87,000	29,000円×3
紙テープ	104,000	260円×400
ラインプリンタ用紙	6,000	2円×250枚×12
テープケース	6,000	1,000,000円×3／50
印刷・資料費	1,050,000	3.5円×12,000／12頁×300部
人件費		
主任以上	1,000,000	1,000,000円×1
その他	180,000	300,000円×0.6
	4,821,000円	

文献リスト月1部　　1,339円

6冊分の場合1分冊当り　　223円

＊NEAC 2206によるデータ。

草間基他，電気通信研究所における文献情報の電子検索（REWDAC）について

ドクメンデーション研究　17〔2〕31～41（67）

3.15.5.2 将来の計画

オンライン化を計画している。

附　　録

附－1　大容量記憶装置の歴史

1. 磁気ドラム以前

初期の電子計算機，たとえばＥＮＩＡＣ，ＥＤＳＡＣ，ＭＡＲＫ－ＩＶ，ＩＢＭ６５０などには数百語ないしは数千語の主記憶装置があるだけで，補助記憶装置とか外部記憶装置といったものはなかった。強いて言えば，紙テープやカードが外部記憶装置の役割りをはたしていた。この紙テープとカードは人間の目で見て所定のものを探せるという特長はあるが，計算機での入出力の速度が遅い，人間の手間が多くかかる，また書き直しができないなどの欠点から現在では初期データの入力媒体にのみ使われるようになっている。

2. 磁気ドラム

磁気ドラムは始め主記憶装置として使われていたが，磁気コアの出現以来補助記憶装置としてもっぱら使われるようになった。そしてこの外部記憶装置の出現とともにプログラムやデータを記憶装置に保存しておくという概念が生れた。

磁気ドラムは磁気ディスクや磁気テープに比較して

ａ）呼出し時間が早い

という特長があるが

ｂ）体積が大きい

ｃ）交換できない

などの欠点のため，あまり大容量のものは作りにくく現在では，磁気コアと磁気ディスクや磁気テープの中間記憶装置すなわちバッファのような形式で用いられるようになってきている（例：ＵＮＩＶＡＣ１１０７）。

磁気ドラムは通常 図1.1 のような形式をしており，回転ドラムの表面円周上のトラックの情報をヘッドで読み書きする。ヘッドは円周上を移動しないので，読み書きしたいときに所定の場所がヘッドの下を通過した後では，1回転待たなければならない。この回転遅れの時間を短縮するために，呼出しヘッドだけ複数個つけて，一番近いところにいるヘッドで読むようにしているものがある（磁気ドラムが主記憶装置

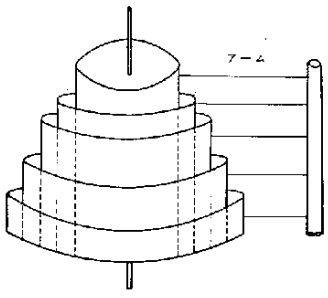


図 1.1　磁気ドラム

として使われている時代にあった）。

普通ヘッドはトラック毎についていて固定のものが多いが，移動するのもある。ドラム1つにヘッド1つだけしかないもの（例：ＵＮＩＶＡＣ　ＲＡＮＤＥＸ），トラックを複数個まとめてバンドとし，このバンドの大きさの数だけヘッドがついて，バンド毎にヘッドが移動する（例：ＦＨ－８８０）ものもある。これは同一ヘッド内ではヘッドは移動せずに読み書きできるが，異なるバンドの読み書きにはヘッドの移動を要する。

3. 磁気テープ

磁気ドラムが取りはずしができず記憶容量が固定だったのに対し，脱着可能な磁気テープの出現により，原理的にはいくらでも記憶容量が増やせることになり，本格的なデータ処理はこの磁気テープの出現により始まったといえる。

磁気テープは巾，¼インチ，½インチ，1インチ，長さ７００ｍ～１２００ｍで，記憶容量も2千万字位入る。磁気テープには番地がついていないことと，その機構上逐次的にしか読んだり書いたりできないので，入力データを磁気テープに入っているファイルの順にそろえておく，いわゆる順ファイル（Ｓｅｑｕｅｎｔｉａｌ　Ｆｉｌｅ）の処理しかできない。

また磁気テープは番地がついていないために，テープ上の記録の一部を変更することはできないという欠点がある。

磁気テープの特長はなんといっても，交換によって，いくらでも記憶容量が増やせること，そのテープの値段が安価（1巻約2万円）なことである　また逐次的にデータを処理する場合には，ブロッキングやバッファリングのテクニックにより非常に高速にできる。

磁気テープの数が多くなると，テープの交換時間（約1分）が問題となるのでカートリッジ式（ＩＢＭ７３４０）のものもある。

4. 磁気ディスク

磁気ドラムの非逐次に読める特長と磁気テープの大容量の特長をそなえた記憶装置として開発されたものである。

回転円板上のトラックの上を呼出し書込み機構（アーム）が移動して，読み書きを行なう。初期のものは（例：ＩＢＭ１４０５）アームがディスク装置全体に1本しかなかったので，異なるディスクをアクセスする場合には，一度アームを引っこめて所定のディスクまで上下に移動し，さらにそのディスク内のトラックまで移動するので非常に時間がかかった（平均６００ｍｓ）。

またアームの数を2～3本として，アクセスの時間の短縮をはかったものもある（例：ＩＢＭ７３００）。

このディスク間にまたがってアームが移動するのはいかにも時間がかかりすぎて遅いので最近のも

のは1面に少なくとも1個のヘッドがついてディスク間でヘッドが移動する形式のものはない。そして各ヘッドはそれぞれ独立になって移動するものはなく，全部いっしょになって動く。このようになって，シリンダーコンセプトが生れた。すなわち，ある面のトラックにデータを書き，続けて次のデータを書くときはその面の次のトラックに書くのではなく，次の面の同じトラックに書くようにする。この場合はアームは移動しないですむ。このようにアームを移動させないで読み書きできる範囲をシリンダーと呼びこの部分はまったく磁気ドラムと同様に使用することができる。すなわち1面に200トラックあるとすれば，200個の磁気ドラムが同心円上にまわっていると考えるのである(図1.2)

円板上のトラックは外側と内側とでは当然その周の長さは異なってくるので，外側と内側を全部同じ記録密度で書くのは難かしい。そこで，外側と内側では記録密度が変えてあるものもある(例：GE DS−20)(図1.3)

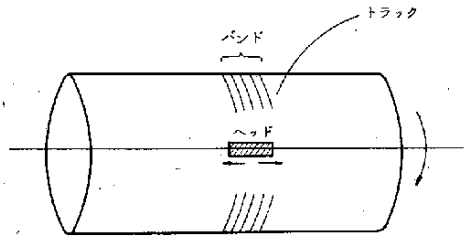


図1.2 シリンダーコンセプト(Cylinder Concept)

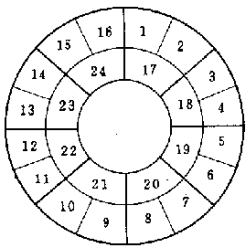


図1.3 外側のトラックと内側と記録密度が異なる例(DS−20)
外側は16セクターあり，内側は8セクターである。
1セクターの記憶容量は内外ともに同じ。

1本のアームに複数個のヘッドがついていてアームを移動せずに同一面の複数個のトラックをアクセスできるものもある（例：GE　DS－20）。シリンダーの容量が増えて，シリンダーの数が減ったと考えられる。

全部の面の全部のトラックにヘッドがついていてアームの固定のものもある（例：B475）。もはやこれはまったく磁気ディスクと同じ考えで使用することができる。

磁気ドラムを交換できるものはなかったが磁気ディスクの革命ともいうべき交換できるディスクパック（例：IBM1311）ができて，磁気ディスクの利用度が増え急速にひろまっている。

また磁気ディスクを集団にしてひとまとめにし記憶容量の大量化をはかっている（例：IBM2314）。これはディスクパックが9個ついていて，常に8個アクセスできる。1個は予備。さらにディスクパックも11枚の円板のうち常時アクセスできるのは18面で予備が2面ある。

5. 磁気カード

磁気ディスクのどこでもすぐにアクセスできて交換もできるという特長をそなえ，さらに記憶容量を増加するためにできたのが磁気カードである。

磁気カードは数百枚ないし数千枚あって，その中から所定のカードが選択されて回転ドラムに巻きついて読み書きが行なわれる。カードが回転ドラムに巻きついている間は普通の磁気ドラムとまったく同じである。異なるカードのアクセスには，いま回転ドラムにあるカードをもとに戻して，次の所定のカードを選択して回転ドラムに巻きつけてから読み書きするので，磁気ディスクの異なるシリンダーのアクセスよりもずっと時間がかかる。しかし，カードが磁気ディスクよりもずっと小さいために1つの装置としては記憶容量を大きくすることができる。数百枚が1単位になって1つの箱（マガジン，サブセルなどという）になっていて，この箱単位に交換可能である。

6. むすび

大容量記憶装置の条件としては

a．体積が小さいこと

b．廉価であること

c．non−volatile（電源を切っても記憶が消えない）であること

d．交換可能であること

e．アクセス時間が短いこと

の5つがある。

この条件を満たすものは現在では磁気ディスクと磁気カードで，いずれも電磁気的に情報を記録し，回転する媒体から読み書きヘッドにより，読出し書込みを行なっている。

将来も当分の間はこの方式が続くものと思われるが，物理的に回転しているためある程度以上はア

クセス時間を短かくすることは不可能である。いずれ上の条件を満たし，磁気コア程の早さで，磁気テープのように安価な大容量記憶装置が出現するであろうが，そのときは計算機の利用方法がかなり変ってくることが予想される。

現在ではアクセス時間の早い順すなわち，磁気ドラム，磁気ディスク，磁気カード，磁気テープの順で主記憶装置からはみ出したデータを収容するというような序列ができそうである。

情報検索には格納する情報の量および検索の頻度，形態（リアルタイムかバッチか）によってもっとも適切な記憶装置を採用すべきである。

附－2　大容量記憶装置の性能比較表

I　磁気ディスク

1. 名前

機器名称または番号とモデル番号

2. 使用計算機

同じディスク装置をいろいろな機種で使える場合にはその代表的な計算機名をいくつかあげてある。

3. 発売日

最初に商品として使用者のもとにわたった年月である。

4. 円板数

1個の匡体またはディスクパックに入っているディスクの枚数である。

5. 使用面数

データの記録に使用できるディスク面の数である。外傷からの保護のため最上面と最下面は使用しなかったり（例：IBM2311），予備をもっていたり（例：IBM2314）して，必ずしも円板数の2倍とはかぎらない。

6. 直径

文字通りディスクの直径

7. 回転数（rpm）

1分間のディスクの回転数

8. トラック数／面

一面上のトラックの数で，全部のアームが同時に動く機構のものはこの数がシリンダーの数となる。

9. 文字数／トラック

1トラックに収容可能な文字数である。1文字は6～8ビット（パリティを含まず）である。内側のトラックと外側のトラックでは文字数が異なる場合（例：DS－20）はその平均で示してある。

10. 文字数／面

1面に収容できる文字数である。原則として，（8.トラック数／面）×（9.文字数／トラック）となっている。

11. 全記憶容量

1つの匡体またはディスクパックの記憶容量である。原則として，（5.使用面数）×（10.文字数／面）となっている。

12. 位置ぎめ時間

ヘッドが所定のトラックまで移動するのに要する時間である。無論，距離によって異なるが，平均または最小と最大を示してある。各トラック毎にヘッドがついている場合にはこの位置ぎめ時間はゼロである（例：Burroughs　B475）。

13. 平均回転遅れ（ms）

ヘッドが所定のトラックに達してから，所定のレコードがヘッドの下を通過するまでの時間である。だいたいディスクの1回転の時間（1000ms ×60秒／回転数）の半分である。いわゆる呼出し時間（access time）はこれと，位置ぎめ時間を加えたものである。

14. 転送速度（文字数／秒）

データの読出しまたは書込みの速度である。

15. アーム数／面

いずれも1本のアームでディスク1枚分（2面）を受けもつようになっていて，1／100 というのは，使用面数が100に対してアームが1本だけということを示す，したがって1本のアームが移動して全部のディスクの読み書きを受けもつ。16／32（例：CDC828Disk File）は32面のディスクに16個アームがついている，すなわち全部の面にヘッドがついていることを示す。

16. ヘッド数／面

1面当りのヘッド数である。50（例：B475）は1面50トラック全部にヘッドがついていて，ヘッドは全部固定であることを示している。4（例：GE　M640A）は1面にヘッドが4つあって，それぞれが1面の¼ずつを分担していることを示す。2／100（例：IBM1405）は100面に対してヘッドが2つだけで，この2つのヘッドで全部の面を受けもつことを示している。

17. 交換

ディスクが交換可能かどうかを示す。

18. 賃借価格（万円／月）

1ケ月のレンタル料を示す。コントローラー，ディスクパックの値段は含んでいない。

19. 買取り価格（万円）

買取ったときの値段，コントローラー，ディスクパックの値段は含んでいない。

20. 備考

特長その他

Ⅱ　磁気ドラム

1. 名前

機器名称または番号とモデル番号

2. 使用計算機

同じ磁気ドラムがいろいろな機種で使える場合にはその代表的な計算機名をいくつかあげてある。

3. 発売日

最初に商品として使用者のもとにわたった年月である。

4. ドラム数

1個の匡体の中に何個ドラムが入っているかである。通常は1か2で交換可能のものはない。

5. ドラムの大きさ（直径×長さ）

磁気ドラムの直径と長さを示す。

6. 回転数（rpm）

1分間の磁気ドラムの回転数

7. トラック数／ドラム

1個のドラムに何トラックあるかを示す。

8. 文字数／トラック

1トラックに何文字入るかを示す。1文字は6～8ビット（パリティビットを除く）である。

9. 全記憶容量（万字）

この装置全体の記憶容量，原則として，（4.ドラム数）×（7.トラック数／ドラム）×（8.文字数／トラック）である。

10. 位置ぎめ時間（ms）

ヘッドが所定のトラックに移動するまでの時間である。平均または最小と最大を示す。各トラック毎にヘッドのある場合はこの値はゼロである。

11. 平均回転遅れ（ms）

ドラム1回転の時間（1000ms ×60秒／回転数）の半分である。

12. 転送速度（文字数／秒）

1秒間に何文字読み書きできるかを示す。

13. ヘッド数／ドラム

ドラムに何個ヘッドがついているかを示す。これがトラック数と同一のときはヘッドは固定である。1（例：UNIVAC　RANDEX）はヘッドが1つだけで全部のトラックを移動してカバーすることを示す。

14. 賃借価格（万円／月）

１ケ月のレンタル料を示す。コントローラーは含まず。

15. 買取り価格（万円）

買取ったときの値段，コントローラーは含んでいない。

16. 備考

特長，その他

No. 1

磁気ディスク性能比較表 (1)

	名前	Disk File Storage Module (B475)	B450 Disk File and Data Communication Basic Control	828 Disk File	838 Disk File	828 Disk File	Control Data 6603 Disk File	Control Data 6607 Disk File System	Control Data 6608 Disk File System	Control Data 852 Disk Storage Drive
2	使用計算機	Burroughs 100/200/300 series	Burroughs B5500	CDC3200	CDC3200	CDC3400	Control Data 6000 series	Control Data 6000 series	Control Data 6000 series	Control Data 3000, 6000 series
3	発売日	1964年	1964年	1964年11月	1965年4月	?	1965年	1966年	1966年	1966年
4	円板数	4	4	16	32	16	14	36	72	6
5	使用面数	8	8	32	64	32	24	64	128	10
6	直径（インチ）	26.5	26.5	31	31	31	?	26	26	14
7	回転数（rpm）	1,500	1,500	1,200	1,200	1,200	900 - 950	1,140	1,140	1,500
8	トラック数／面	50	50	256	256	256	512	192	192	100
9	文字数／トラック	24,000	24,000	平均 4,096	平均 4,096	平均 4,096		6,827	6,827	2,000 又は 2,980
10	文字数／面（万字）	120	120	100	100	100		131	131	20.0又は29.8
11	全記憶容量（万字）	960	960	3,300	6,600	3,300	8,080	8,400	16,800	200 又は 298
12	位置ぎめ時間（ms）	0	0	平均 207	平均 207	平均 207	0 - 120	0 - 100	0 - 100	平均 57.5
13	平均回転遅れ（ms）	20	20	25	25	25	30 - 31.5	26.3	26.3	20
14	転送速度（文字／秒）	10^5	10^5	(最大)9,800	(最大)9,800	(最大)9,800	1,342,104	1,666,667	3,333,334	77,730
15	アーム数／面	4/8	4/8	16/32	32/64	16/32	14/28			6/12
16	ヘッド数／面	50	50	4	4	4	4	1	1	
17	交換	不可	不可	不可	不可	不可	不可	不可	不可	不可
18	賃借価格（万円／月）	33	33	101	?	101	?	?	?	?
19	買取り価格（万円）	2,306	2,306	4,392	?	4,392	?	?	?	?
20	備考	各トラック毎にヘッドがついていてアームは固定である	各トラック毎にヘッドがついていてアームは固定である		Model 28を2つ合せたもの		外側のトラックは128セクター，内側のトラックは100セクター入る		CDC6607を2つ合せたもの	

No.2

磁気ディスク性能比較表 (2)

	名前	Control Data 853 Disk Storage Drive	Control Data 854 Disk Storage Drive	M640A(MRADS) Mass Random Access Data Storage	M640A(MRADS) Mass Random Access Data Storage	M640A(MRADS) Mass Random Access Data Storage	DS-20 Disc Storage Unit	DS-25 Disc Storage File Unit	DS-15 Removable Disc Storage Drive	IBM1405 (RAMAC) Disk Storage Unit Model 1
2	使用計算機	Control Data 3000, 6000 Series	Control Data 3000, 6000 Series	GE215	GE225	GE235	GE400 Series GE600 Series	GE400 Series GE600 Series	GE400 Series GE600 Series	IBM1401
3	発売日	1966年	1966年	1963年	1962年	1963年	1964年4月	1966年	1966年	1961年7月
4	円板数	6	6	18	18	18	4	16	1	25
5	使用面数	10	10	32	32	32	8	32	2	50
6	直径（インチ）	14	14	31	31	31	31	21	16	25
7	回転数（rpm）	2,400	2,400	1,200	1,200	1,200	1,170	1,200	1,200	1,200
8	トラック数／面	100	200	256	256	256	256	512	320	200
9	文字数／トラック	4,096	4,096	平均 2,304	平均 2,304	平均 2,304	平均 2,880	6,144	12,288	1,000
10	文字数／面（万字）	41	82	59	59	59	74	315	393	20
11	全記憶容量（万字）	410	820	1,888	1,888	1,888	592	10,080	786	1,000
12	位置きめ時間（ms）	平均 57.5	平均 57.5	0又は70 - 305	0又は70 - 305	0又は70 - 305	0又は70 - 305	平均 90	平均 79	平均 600
13	平均回転遅れ（ms）	12.5	12.5	25	25	25	26	26	25	25
14	転送速度（文字／秒）	208,333	208,333	最大23,700	最大 23,700	最大 23,700	最大 75,000	300,000	260,000	22,500
15	アーム数／面	6/12	6/12	18/36	18/36	18/36	4/8	8/32	1/2	1/50又は2/50
16	ヘッド数／面			4	4	4	4	8	4	1/25又は2/25
17	交換	可	可	不可	不可	不可	不可	不可	可	不可
18	賃借価格（万円／月）	?	?	62			41	169	16	35
19	買取り価格（万円）	?	?	2,736			1,908	8,100	778	1,296
20	備考						円板数は4，8，12，16枚の4種類ある。	円板数は16，24，32枚の3種類ある。		アームは標準は1本だがもう1本追加できる

No. 3

磁気ディスク性能比較表 (3)

1	名前	IBM1405 (RAMAC) Disk Storage Unit Model 2	Disk Storage Drive 1311 Model 2	Disk Storage Drive 1311 Model 4	Disk Storage Unit (DSU) 1301 Model 1	Disk Storage Unit (DSU) 1301 Model 2	IBM7300 (RAMAC) Disk Storage Unit Model 1	IBM7300 (RAMAC) Disk Storage Unit Model 2	IBM1302 Disk Storage Unit Model 1	IBM1302 Disk Storage Unit Model 2
2	使用計算機	IBM1401	IBM1401, 1410, 1440, 1620	IBM1401, 1410, 1440, 1620	IBM1410	IBM1410	IBM7070/7074	IBM7070/7074	IBM7080	IBM7080
3	発売日	1961年7月	?	?	?	?	1960年3月	1960年3月	1965年	1965年
4	円板数	50	6	6	25	50	50	50	25	50
5	使用面数	100	10	10	40	80	100	100	40	80
6	直径 (インチ)	25	14	14	24	24	24	24		
7	回転数 (rpm)	1,200	1,500	1,500	1,790	1,790	1,200	1,200	1,790	1,790
8	トラック数／面	200	100	100	250	250	100	200	500	500
9	文字数／トラック	1,000	2,000	2,000	2,780	2,780	300	300	5,850	5,850
10	文字数／面 (万字)	20	20	20	69.5	69.5	3	6	293	293
11	全記憶容量 (万字)	200	200	200	2,780	5,560	300	600	11,720	11,720
12	位置ぎめ時間 (ms)	平均 600	平均 250 又は 〃 154	平均 250 又は 〃 154	0又は50 - 180	0又は50 - 180			0又は50 - 180	0又は50 - 180
13	平均回転遅れ (ms)	25	20	20	17	17	25	25	17	17
14	転送速度 (文字／秒)	22,500	77,000	77,000	83,000	83,000	7,000	7,000	184,000	184,000
15	アーム数／面	1/100 又は 2/100	5/10	5/10	1/40	1/80	3/100	3/100	2/40	2/80
16	ヘッド数／面	2/100 又は 4/100	10/10	10/10	2/40	2/80	6/100	6/100	4/40	4/80
17	交換	不可	可	可	不可	不可	不可	不可	不可	不可
18	賃借価格 (万円／月)	55	14	36	78	128	35	54	202	284
19	買取り価格 (万円)	1,746	612	1,679	4,288	6,808	2,239	2,693	9,072	12,798
20	備考	アームは標準は1本だが，もう1本追加できる	最初の交換可能なディスク							

No. 4

磁気ディスク性能比較表 (4)

1	名前	IBM1302 Disk Storage Model N1	IBM1302 Disk Storage Model N2	IBM2311 Disk Storage Drive (1316 Disk Pack)	IBM2314 Direct Access Storage Facility	H460, Model 0 Magnetic Disc File Bryant Series 4000	H460, Model 1 Magnetic Disc File Bryant Series 4000	H460, Model 2 Magnetic Disc File Bryant Series 4000	H460, Model 3 Magnetic Disc File Bryant Series 4000	H460, Model 4 Magnetic Disc File Bryant Series 4000
2	使用計算機	IBM System /360	IBM System /360	IBM System /360	IBM System /360	Honeywell 400	Honeywell 400	Honeywell 400	Honeywell 400	Honeywell 400
3	発売日	1965年	1965年	?	1967年	1963年4月	1963年4月	1963年4月	1963年4月	1963年4月
4	円板数	25	50	6	11	3	6	12	18	24
5	使用面数	45	90	10	18	6	12	24	36	48
6	直径（インチ）			14	14	39	39	39	39	39
7	回転数（rpm）	1,794	1,794	2,400	2,400	900	900	900	900	900
8	トラック数／面	500	500	200	200	768	768	768	768	768
9	文字数／トラック	バイト4,984	4,984	3,625	7,188	2,710	2,710	2,710	2,710	2,710
10	文字数／面（万字）	249	249	72.5	144	208	208	208	208	208
11	全記憶容量（万字）	11,205	22,410	725	2,592	1,250	2,500	5,000	7,500	10,000
12	位置ぎめ時間（ms）	平均 165	平均 165	平均 85	平均 75	平均 105	平均 105	平均 105	平均 105	平均 105
13	平均回転遅れ（ms）	17	17	12.5	12.5	34	34	34	34	34
14	転送速度（文字／秒）	156,000	156,000	156,000	312,000	75,000	75,000	75,000	75,000	75,000
15	アーム数／面	2/45	2/90	5/10	10/20					
16	ヘッド数／面	4/45	4/90	1	1	6	6	6	6	6
17	交換	不可	不可	可	可	不可	不可	不可	不可	不可
18	賃借価格(万円／月)	?	?	?	?	?	104	144	184	223
19	買取り価格（万円）	?	?	?	?	?	4,040	6,480	7,920	9,360
20	備考				これはディスクパックが，9個ついていて，そのうち8個アクセスできる。1個は予備					

磁気ディスク性能比較表 (5)

No. 5

	1 名前	H860, Model 1 Magnetic Disc File Bryant Series 4000	H860, Model 2 Magnetic Disc File Bryant Series 4000	H460, Model 0 Magnetic Disc File Bryant Series 4000	H460, Model 1 Magnetic Disc File Bryant Series 4000	H460, Model 2 Magnetic Disc File Bryant Series 4000	H460, Model 3 Magnetic Disc File Bryant Series 4000	H460, Model 4 Magnetic Disc File Bryant Series 4000	260-1 Random Access Disc Storage	260-2 Random Access Disc Storage
2	使用計算機	Honeywell 800	Honeywell 800	Honeywell 1400	Honeywell 1400	Honeywell 1400	Honeywell 1400	Honeywell 1400	Honeywell 200	Honeywell 200
3	発売日	1963年	1963年	1963年4月	1963年4月	1963年4月	1963年4月	1963年4月	?	?
4	円板数	12	24	3	6	12	18	24	1	2
5	使用面数	24	48	6	12	24	36	48	2	4
6	直径（インチ）	39	39	39	39	39	39	39	39	39
7	回転数（rpm）	900	900	900	900	900	900	900	900	900
8	トラック数／面	384	384	768	768	768	768	768	768	768
9	文字数／トラック	21,845	21,845	2,760	2,760	2,760	2,760	2,760	2,760	2,760
10	文字数／面（万字）	839	839	210	210	210	210	210	210	210
11	全記憶容量（万字）	20,136	40,272	1,250	2,500	5,000	7,500	10,000	420	840
12	位置ぎめ時間（ms）	平均 102	平均 102	平均 90	平均 90	平均 90	平均 90	平均 90	平均 95	平均 95
13	平均回転遅れ（ms）	33	33	33	33	33	33	33	33	33
14	転送速度（文字／秒）	34,200	34,200	74,800	74,800	74,800	74,800	74,800	64,300	64,300
15	アーム数／面									
16	ヘッド数／面									
17	交換	不可	不可	不可	不可	不可	不可	不可	不可	不可
18	賃借価格（万円／月）	222	292	?	104	144	184	223	71	77
19	買取り価格（万円）	9,900	13,140	?	4,040	6,480	7,920	9,360	3,200	3,483
20	備考									

No. 6

磁気ディスク性能比較表 (6)

1	名前	260-3 Random Access Disc Storage	260-4 Random Access Disc Storage	260-5 Random Access Disc Storage	260-6 Random Access Disc Storage	260-7 Random Access Disc Storage	260-8 Random Access Disc Storage	260-9 Random Access Disc Storage	
2	使用計算機	Honeywell 200	Honeywell 200	Honeywell 200	Honeywell 200	Honeywell 200	Honeywell 200	Honeywell 200	
3	発売日	?	?	?	?	?	?	?	
4	円板数	3	4	5	6	12	18	24	
5	使用面数	6	8	10	12	24	36	48	
6	直径（インチ）	39	39	39	39	39	39	39	
7	回転数（rpm）	900	900	900	900	900	900	900	
8	トラック数／面	768	768	768	768	768	768	768	
9	文字数／トラック	2,760	2,760	2,760	2,760	2,760	2,760	2,760	
10	文字数／面（万字）	210	210	210	210	210	210	210	
11	全記憶容量（万字）	1,260	1,680	2,100	2,520	5,040	7,560	10,080	
12	位置ぎめ時間（ms）	平均 95	平均 95	平均 95	平均 95	平均 95	平均 95	平均 95	
13	平均回転遅れ（ms）	33	33	33	33	33	33	33	
14	転送速度（文字／秒）	64,300	64,300	64,300	64,300	64,300	64,300	64,300	
15	アーム数／面								
16	ヘッド数／面								
17	交換	不可	不可	不可	不可	不可	不可	不可	
18	賃借価格（万円／月）	84	90	96	103	140	178	216	
19	買取り価格（万円）	3,767	4,050	4,332	4,617	6,318	8,018	9,720	
20	備考								

No. 7

磁気ドラム性能比較表 (1)

1	名前	Storage Drum Model B430	861 Drum Storage Unit	862 Drum Storage Unit	MDS200 Magnetic Drum and Controller	7320 Drum Storage	2301 Drum Storage	Random Access Drum Storage Model 270	Magnetic Drum Unit Model 272	3465 Data Drum Memory Model 1
2	使用計算機	Borroughs 5000	CDC 3200	CDC 3200	GE600 Series	IBM System /360 7090/7094	IBM System /360	Honeywell 200	Rhiko 2000-210/211	RCA3301
3	発売日	1963年4月	1965年4月	1965年8月	1962年9月	1964年	?	1964年12月	1960年6月	1964年
4	ドラム数	2	1	1	1	1	1	1	1	1
5	ドラムの大きさ（長径×長さ）（インチ）	8 x 22	18 x ?	?	24 x 30	12 x 12	10.7 x 12	20 x 25	18.5 x 24	10 x $16\frac{3}{8}$
6	回転数（rpm）	3,600	1,745	3,490	1,800	3,490	3,490	1,200	1,750	3,500
7	トラック数／ドラム	384	832	832	880	400	800	512	384	128
8	文字数／トラック	680	5,030	2,515	6,144	2,075	5,120	5,120	682	2,560
9	全記憶容量（万字）	52	419	209	472	83	410	262	26	33
10	位置ぎめ時間（ms）	0 - 0.04	0	0	0	0	0	0		0
11	平均回転遅れ（ms）	4	17.2	8.6	16.7	8.6	8.6	25	17	8.6
12	転送速度（文字／秒）	30,720	2,000,000	2,000,000	180,000	135,000	1,200,000	102,000	952,000	150,000
13	ヘッド数／ドラム	64	?	?		400	400	512	384	128
14	賃借価格（万円／月）	?	?	?	119	?	?	?	58	?
15	買取り価格（万円）	?	?	?	5,436	?	?	?	2,592	?
16	備考				UNIVAC FH-880に同じ。					

No. 8

磁気ドラム性能比較表 (2)

	名前	3465 Data Drum Memory Model 2	3465 Data Drum Memory Model 3	3465 Data Drum Memory Model 4	3465 Data Drum Memory Model 5	3465 Data Drum Memory Model 6	70/565-12 Drum Memory Unit	70/565-13 Drum Memory Unit	RANDEX Drum Storage Type No. 7965	RANDEX Drum Storage Type No. 7957
2	使用計算機	RCA3301	RCA3301	RCA3301	RCA3301	RCA3301	RCA SPECTRA 70	RCA SPECTRA 70	UNIVAC SS 80/90 Model I	UNIVAC SS 80/90 Model I
3	発売日	1964年	1964年	1964年	1964年	1964年	?	?	1962年1月	1962年1月
4	ドラム数	1	1	1	2	2	1	1	1	2
5	ドラムの大きさ（長径×長さ）（インチ	$10 \times 16\frac{3}{8}$	$10 \times 17\frac{7}{8}$	$10 \times 17\frac{7}{8}$	$10 \times 30\frac{13}{16}$	$10 \times 30\frac{13}{16}$	12 x ?	12 x ?	24.3 x 44	24.3 x 44
6	回転数（rpm）	3,500	3,500	3,500	3,500	3,500	3,600	3,600	870	870
7	トラック数／ドラム	256	512	640	768	1,024	253	506	2,000	2,000
8	文字数／トラック	2,560	2,560	2,560	2,560	2,560	3,053	3,053	2,880	2,880
9	全記憶容量（万字）	66	131	164	187	262	78	156	576	1,152
10	位置ぎめ時間（ms）	0	0	0	0	0	0	0	5 - 540	5 - 540
11	平均回転遅れ（ms）	8.6	8.6	8.6	8.6	8.6	8.6	8.6	34.5	34.5
12	転送速度（文字／秒）	150,000	150,000	150,000	150,000	150,000	210,000	210,000	3,480	3,480
13	ヘッド数／ドラム	256	512	640	768	1,024	253	506	1	2/2
14	賃借価格（万円／月）	?	?	?	?	?	54	80	68	90
15	買取り価格（万円）	?	?	?	?	?	2,700	4,500	4,500	5,040
16	備考								ヘッドはドラムに1個だけ	ヘッドはドラムに1個だけ

No. 9

磁気ドラム性能比較表 (3)

	名前	RANDEX Drum Storage Type No. 7966	Flying Head Magnetic Drum Type 8112	Fastrand Mass Storage Subsystem Type 8206	Fastrand I Mass Storage Subsystem	Fastrand II Mass Storage Subsystem	Fastrand II Mass Storage Subsystem	Fastrand I Mass Storage Subsystem	Fastrand II Mass Storage Subsystem	Fastrand II Mass Storage Subsystem Type 6010
2	使用計算機	UNIVAC SS 80/90 Model I	UNIVAC 490	UNIVAC 490	UNIVAC 1050	UNIVAC 1050	UNIVAC 1108	UNIVAC 418	UNIVAC 418	UNIVAC 490 Series
3	発売日	1962年1月	1961年12月	1963年9月	1964年	1964年	1964年	1963年9月	?	1964年
4	ドラム数	2	1	2	2	2	2	2	2	2
5	ドラムの大きさ（直径×長さ）（インチ）	24.3 x 44	24 x 30	23.8 x 61.2	23.8 x 61.2	23.8 x 61.2	23.8 x 61.2	23.8 x 61.2	23.8 x 61.2	23.8 x 61.2
6	回転数（rpm）	870	1,800	870	870	870	870	870	870	870
7	トラック数／ドラム	2,000	768	3,072	3,072	6,144	6,144	3,072	6,144	6,142
8	文字数／トラック	2,880	5,120	10,560	10,752	10,752	10,752	10,752	10,752	10,560
9	全記憶容量（万字）	1,152	393	6,488	6,605	13,210	13,210	6,605	13,210	12,976
10	位置ぎめ時間（ms）	5 ~ 540	0	平均 58	平均 58	平均 58	平均 58	平均 58	平均 58	平均 58
11	平均回転遅れ（ms）	34.5	16.7	35	35	35	35	35	35	35
12	転送速度（文字／秒）	3,480	300,000	50,300	185,000	185,000	153,750	158,054	158,054	125,750
13	ヘッド数／ドラム	2/2	768	64/2	64/2	64/2	64/2	64/2	64/2	64/2
14	賃借価格（万円／月）	68	72	119	119	137	137	119	139	139
15	買取り価格（万円）	3,060	3,312	4,500	5,760	6,624	6,624	5,760	6,624	6,624
16	備考			ヘッドは移動型である	同左	同左	同左	同左	同左	同左

No. 10

磁気ドラム性能比較表 (4)

	名前	Flying Head 432 Magnetic Drum	Flying Head 432 Magnetic Drum	Flying Head 1782 Magnetic Drum	Flying Head 1782 Magnetic Drum	Flying Head 220 Magnetic Drum	Flying Head 330 Magnetic Drum	Flying Head 880 Magnetic Drum	Flying Head 880 Magnetic Drum Type 7304	
2	使用計算機	UNIVAC 1108	UNIVAC 490 Series	UNIVAC 1108	UNIVAC 494	UNIVAC 418	UNIVAC 418	UNIVAC 418	UNIVAC 490 Series	
3	発売日	1965年12月	1966年	1966年	1966年	1963年6月	1965年1月	1962年9月	1961年12月	
4	ドラム数	1	1	1	1	1	1	1	1	
5	ドラムの大きさ（直径×長さ）（インチ）	10.5 x 9.0	10.5 x 9.0	24 x 36	24 x 36	20 x 28.5	20 x 28.5	24 x 30	24 x 30	
6	回転数（rpm）	7,100	7,100	1,770	1,800	3,600	3,600	1,800		
7	トラック数/ドラム	384	384	1,536	1,536	128	256	880		
8	文字数/トラック	4,096	3,413	8,192	6,827	1,536	3,022	6,144		
9	全記憶容量（万字）	157	131	1,258	1,049	20	77	472	393	
10	位置ぎめ時間（ms）	0	0	0	0	0 0.3	0	0		
11	平均回転遅れ（ms）	4.25	4.25	17	17	8.3	8.3	16.7		
12	転送速度（文字/秒）	1,440,000	1,200,000	1,362,000	1,200,000	23,100	90,000	180,000		
13	ヘッド数/ドラム	384	384	1,536	1,536	128	256			
14	賃借価格（万円/月）	36	36	?	?	36	43	51	72	
15	買取り価格（万円）	4,320	1,440	?	?	1,296	1,440	3,312	3,312	
16	備考									

Ⅲ　磁気カード

1. 名前

機器名称または番号とモデル番号。

2. 使用計算機

この磁気カード記憶装置が接続可能な計算機名。

3. 発売日

最初の商品として使用者のもとにわたった年月である。

4. カードの枚数

1個の匡体の中に収容できるカードの枚数。

5. カードの大きさ

カードのたてとよこの長さ。

6. 回転ドラムの大きさ

磁気カードを巻きつける回転ドラムの直径と長さである。

7. 回転数（rpm）

回転ドラムの1分間の回転数である。

8. トラック数／カード

磁気カードのトラック数，回転ドラムのトラック数と一致する。

9. 文字数／トラック

1トラックに入る文字数を示す。1文字は6～8ビット（パリティビットを除く）である。

10. 文字数／カード

1枚のカードに何文字入るかを示す。（8.トラック数／カード）×（9.文字数／トラック）

11. 全記憶容量（万字）

この装置全体の記憶容量，原則として，（4.カードの枚数）×（10.文字数／カード）である。

12. 平均待ち時間（ms）

カードがドラムに巻きついて，ヘッドが所定の位置に移動し，読み書き可能な状態になるまでの時間で，平均または最小と最大を示す。

13. 転送速度（文字数／秒）

1秒間に何文字読み書きできるかを示す。

14. ヘッド数／ドラム

回転ドラムのヘッド数である。8のトラック数／カードと一致しているときはヘッドは固定である。

15. 交換単位

磁気カードはいずれも交換可能であり，その単位を示す。だいたいカード２００～５００枚単位で交換可能である。

16. 賃借価格

17. 買取り価格

18. 備考

No. 11

磁気カード性能比較表 (1)

／1	名前	MS-40 Storage Cell Drive	2321 Data Cell Drive Model 1	Type 251 Mass Memory File	Type 252 Mass Memory File	Type 253 Mass Memory File	CRAM, Card Random Access Memory Model 353-1	CRAM, Card Random Access Memory Model 353-2	CRAM, Card Random Access Memory Model 353-3	Model 3488 Random Access Computer Equipment
2	使用計算機	GE400 Series	IBM System /360	Honeywell Series 200	Honeywell Series 200	Honeywell Series 200	NCR 315	NCR 315	NCR 315	RCA 3301
3	発売日	1965年7月	?	1966年	1966年	1967年	1962年6月	1962年6月	1962年6月	1964月
4	カードの枚数	2,000	2,000	512	512	2,560	256	128	256	2,048
5	カードの大きさ（インチ）	2.25 x 13	2.25 x 13	? x 7	? x 7	? x 7	3.25 x 14	3.25 x 14	3.25 x 14	4.5 x 16
6	回転ドラムの大きさ（直径×長さ）（インチ）	?	?	?	?	?	3.5 x 7	3.5 x 7	3.5 x 7	6 x 5
7	回転数（rpm）	1,200	1,200	3,600	3,600	3,600	1,235	1,235	1,235	?
8	トラック数／カード	100	100	32	128	128	42	56	56	128
9	文字数／トラック	2,664	バイト 2,000	918	918	918	516	1,120	1,120	1,300
10	文字数／カード（万字）	27	20	3	12	12	22	6.3	6.3	16.6
11	全記憶容量（万字）	54,000	40,000	1,500	6,000	30,000	555	806	1,613	34,000
12	平均待ち時間（ms）	175 - 600	175 - 600				200	200	200	
13	転送速度（文字／秒）	73,300	54,700	50,000	50,000	50,000	100,000	38,000	38,000	80,000
14	ヘッド数／ドラム	20	20	16	16	16	42	56	56	8
15	交換単位	1サブセル	1サブセル	1カートリッヂ	1カートリッヂ	1カートリッヂ	1カードブロック	1カードブロック	1カードブロック	1マガジン
16	賃借価価（万円／月）	?	?	?	?	?	?	?	?	?
17	買取り価格（万円）	?	?	?	?	?	?	?	?	?
18	備考	IBM2321に同じ	磁気カード200枚1組をサブセルと呼び，これが10組入っている	512枚が1つのカートリッジに入っている。カートリッジの数は1個	同左	TYPE252と同様のカートリッジが5個入っている。	最初にできた磁気カード装置			256枚の磁気カードが1組で1マガジンといい，これが8組入っている。

磁気カード性能比較表 (2)

No. 12				
	1	名前	70/568-11 Mass Storage Unit	
	2	使用計算機	RCA SPECTRA 70	
	3	発売日	?	
	4	カードの枚数	256	
	5	カードの大きさ（インチ）	4.5 x 16	
	6	回転ドラムの大きさ（直径×長さ）（インチ）	?	
	7	回転数（rpm）	?	
	8	トラック数／カード	128	
	9	文字数／トラック	2,139	
	10	文字数／カード（万字）	27.4	
	11	全記憶装置（万字）	7,014	
	12	平均待ち時間（ms）		
	13	転送速度（文字／秒）	70,000	
	14	ヘッド数／ドラム	8	
	15	交換単位	1マガジン	
	16	賃借価格（万円／月）	?	
	17	買取り価格（万円）	?	
	18	備考		

附－3　汎用言語とIRに使用できる言語

附－3.1　CISS

1.　システム概説

1.1　目　　的

CISS（Consolidated Information Storage System）は磁気ディスクを対象とした汎用ファイル処理システムである。その目的はつぎのようである。

(1)　データベースの作成

(2)　データベースの保守

(3)　検索と処理

データベースとは，データの集中管理を目的として，多数のファイルを統合したもののことである。CISSはCOBOL 言語にCISS言語を追加したCOBOLの拡張型である。

データベースには，データベースのコントロール情報およびデータが収められている。このコントロール情報とはデータ相互の有機的関係（チェインやリンク）や，データの構成要素（長さ，性質）を含む情報をいう。CISSの特徴は，

①　チェイン，リンクを用いることによりデータの任意の部分に二次インデックスとしての働きを持たせ，データの内容によるアクセスが可能なこと。

②　データベースはプログラムにより独立に作られ，これを使用するときは，その構成についてあまり詳しく知らなくても利用できること。

③　マルティプログラミングに適応できること。

④　機密保持，バックアップが完全にできること。

などである。

1.2　プログラムの構成

CISS プログラムの構成は次のようになっている。

CISS
- CIS—MONITOR
- CIS—TRANSLATOR
- UTILITY プログラム
- JOB プログラム

(1)　CIS—MONITOR

IOFCSに追加されたもので，JOB プログラムからの要求に従いJOB プログラムの

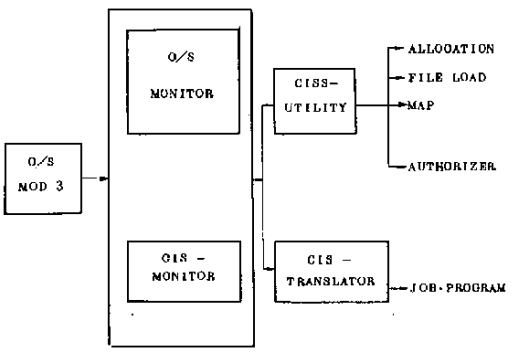


図1.2.1

バッファとデータベースとのデータの移送，データベースのバックアップ処理や検策を行う。

(2) CIS—TRANSLATOR

CISS言語をCOBOL に変換するコンパイラである。

(3) UTILITY プログラム

① CIS—ALLOCATOR

マスストリッジをフォーマットし，エリアの保持をすることにより，CIS—LOADER がデータベースを作成できる準備をする。

② CIS—LOADER

データベースの作成をする。入力データをディスク上に格納してそのロジカルな関係を与え，その他各種テーブルを格納する。

③ CIS—MAP

データベースの内容をチェインまたはリンクに従いプリントアウトする。

④ AUTHORIZER

データベースの機密保護のために Authority テーブルをディスク上に作成する。

1.3 機 器 構 成

CISS は NEAC —シリーズ２２００オペレイティングシステム MOD Ⅲの中に組み込まれたものである。

(1) CIS—TRANSLATOR

磁気テープ …… 2台（CITおよびワーク）

磁気ディスク …… 1台（SOF）

カードリーダー
磁気テープ } ソースデックの入力

ラインプリンタ

CPU …… 40k以上

(2) CIS—ALLOCATOR

カードリーダー …… 1台

ラインプリンタ …… 1台

DISK …… 2～16台（含：SOF）

(3) CIS—LOADER

磁気テープ …… 3台（optionとして5台まで）

カードリーダー …… 1台

ラインプリンタ …… 1台

磁気ディスク …… 16台

2. データ構成

2.1 ロジカルな構成

① フィールド：システムで処理するデータの最小単位。

② レコード：フィールドの集合。1レコードは1つ以上のフィールドから成る。

③ セグメント：同一種類のレコード。

例

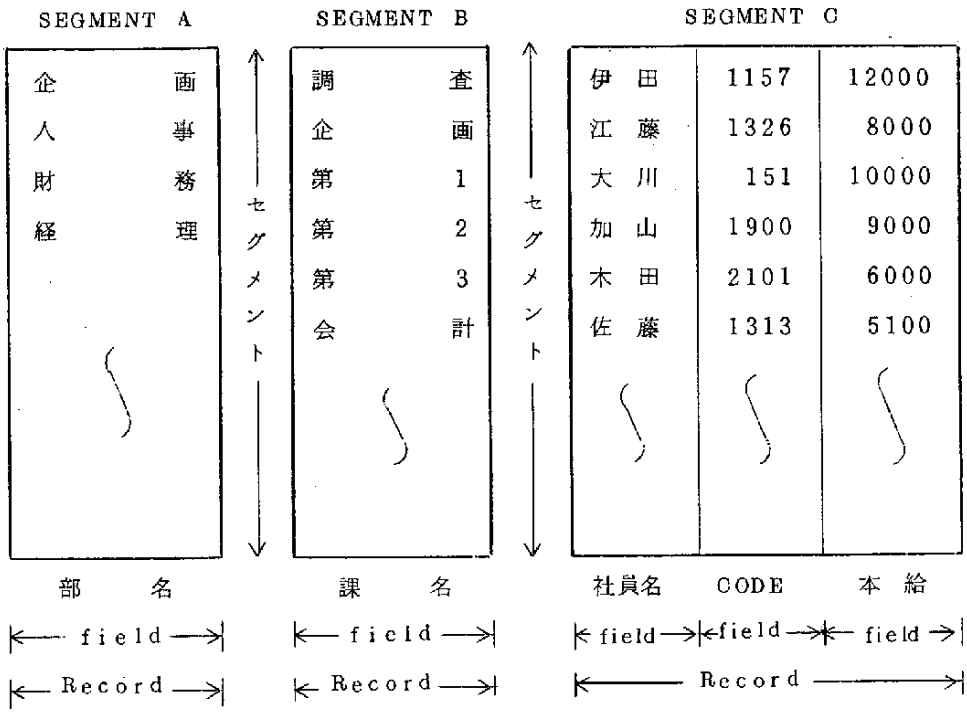


図 2. 1

2.2 フィジカルな構成

① ページ： 入出力の単位で固定長である。データベース内ではページナンバーが0から順次連続的に与えられている。

② パラグラフ： 同一ページ内に関係ある数種類のセグメントを格納することができる。そのためにページを分割した単位をパラグラフという。

③ セクション： セグメントがデータベース内に格納されるページ

例

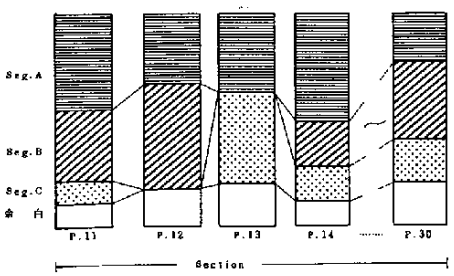


図 2.2

の範囲。

2.3 チェインとリンク

セグモント内外のレコードの論理的な関係を定義するためにチェインとリンクを用いる。

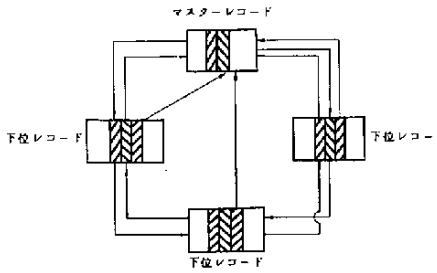


図 2.3

① チェイン

図 2.3のように，チェインには次のレコード，前のレコード，マスターレコードを示すチェインの3種類がある。各下位レコードは同一のセグメントのレコードでなければならず，マスターレコードは異なるセグメントでなければならない。

② リンク

リンクはチェインと同様にポインターによってつぎのレコードを示す。しかしループにならない。図2.4において，nとn＋1とn＋2はmの下位レコードである。これはmをマスターとするチェインと同じ関係となる。しかし，この場合は下位レコードには，ポインターフィールドは不要である。

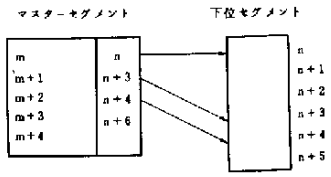


図 2.4

各レコードは任意のチェインまたはリンクのマスターレコードにも，下位レコードにもなれる。

チェインやリンクによって，セグメント間に上下関係ができる。セグメントAをマスターレコードとし，セグメントBを下位レコードとしたとき，図 2.6 のように示す。

2.4 サブファイル

データベース内のセグメントの相互関係は，数個の木構造に分けられる。各木構造をサブファイルと呼ぶ。

木構造を表わすのに，レベル番号を使う。図2.7において，A，A′ はレベル 1，B，B_1'，B_2'，はレベル 2，C_1，C_2，C_3，C′ はレベル 3 である。

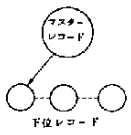


図 2.5

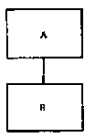


図 2.6

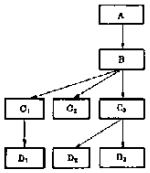


図 2.7

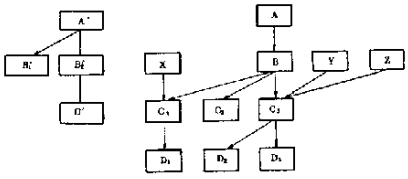


図 2.8

サブファイルは，図2.7の木構造に加えて，外部セグメントと呼ばれるものを持つことができる。たとえば図2.8のＸＹＺは外部セグメントである。

3. データベースの保護

つぎのような目的のために，データベースの保護を行う。

① 特定の利用者の機密保持

② 更新処理の禁止

③ 現在使用中のデータを，他のプログラムで更新したり，他で更新中のデータを利用したりすることの禁止

データベースの保護にはつぎの3通りがある。

(1) AUTHORITY LEVEL (AL)

各セグメントに与えられる64段階のレベルである。

MOVE LEVEL (ML) …… 読み出し

UPDATE LEVEL (UL)…… 更新処理

の2つのレベルがあり，(ML, UL) という形で与える。

(00, 00)≦(ML, UL)≦(64, 64)

利用者のAL(Authority Word で与える)がセグメントのALより低いときのみ，処理が可能である。

(2) PRIVILEGE KEY (PK) および PRIVILEGE LEVEL (PL)

PKとは，ALと同様Authority Word と各セグメントに対して与えられるもので，PKの一致するものに関しては，ALに関係なく，PLによってその範囲のセグメントが参照可能である。

(3) BUSY CHECK

現在処理中のレコードを含むページは，その処理中，他のプログラムの使用を禁ずるものである。

4. データベースの作成

データベースの作成は2つの段階からなる。第1に CIS－ALLOCATORにより，ディスク上にデータベースを格納するためのファイルを割り当て，そして CIS－MONITOR で使用するディレクトリの作成を行う。第2に CIS－LOADER によって，データの定義を行い，実際にファイルにデータを格納する。

4.1 CIS—ALLOCATOR

CIS－ALLOCATOR のパラメーターにはつぎのようなものがある。

① PREP　aaaaaa, a′ a′ a′ a′ a′ a′, type, ALL

機能： volume preparation用のカードである。

- aaaaaa …… パックに登録する Volume Serial Number
- a′a′a′a′a′a′ …… 今までマウントされていた Volume Serial Number をチェックする。
- type …… volume の種類を与える。
- ALL …… 全シリンダーがフォーマットされる。

② NAME *CISS FILE*

機能： このまま準備する。

③ PAGESIZE bbbb

機能： ページサイズを指定する。 例

④ EXPERDAT' yyddd

機能： 保存日数を指定する。

⑤ PASSWORD xxxxxxxx

(Option)

機能： 8文字までのパスワードを指定。

⑥ TRANSFER 6or8

(Option)

機能： 移送のビット数を指定する。

⑦ VOLNAME aaaaaa

(Option)

機能： aaaaaaでアロケイトしたい Volume Serial Number を指定する。

⑧ ALLOCATE ccc, tt, c′c′c′, t′t′

機能： エリア指定の基本単位であるユニット・オブ・アロケイション(UOA)の指定を行なう。

ccc, tt …… UOAの最初のシリンダーおよびトラックのアドレス。

c′c′c′, t′t′ …… UOAの最後のシリンダーおよびトラックのアドレス。

```
STATEMENT LEVEL PARAMETER
  DEFINE          CISS,(0,5)
  SUBFILE         01
  SEGMENT         (6,14),10
  RECORD     01   BU,2,15,NEXT,,,AFTER
  FIELD           BU,ALPHA,15
  SEGMENT         (6,14),50
  RECORD     02   KA,3,12,NEXT,,MASTER,AFTER
  FIELD           KA,ALPHA,12
  SEGMENT         (15,15),1
  RECORD     61   SENKO,2,5,NEXT,,,AFTER
  FIELD           SENKO,ALPHA,5
  SEGMENT         (16,1015),40
  RECORD     03   ZYUGYOIN,4,24,NEXT,,,AFTER
  FIELD           BANGO,NUMERIC,5
  FIELD           NAME,ALPHA,18
  FIELD           SEI,ALPHA,1
  SEGMENT         (1016,2015),15
  RECORD     04   SYUGAKU,1,17,NEXT,,,AFTER
  FIELD           SYUSSHIN,ALPHA,15
  FIELD           SOTHUNEN,NUMERIC,2
  SEGMENT         (1016,2015),15
  RECORD     04   KYUYO,1,18,NEXT,,,AFTER
  FIELD           KIHONKYU,NUMERIC,6
  FIELD           SYOKUNOKYU,NUMERIC,6
  FIELD           TEATE,NUMERIC,6
  SEGMENT         (2016,3015),10
  RECORD     04   GENZYUSHO,1,00,,,,AFTER
  FIELD           GENZYUSYO,ALPHA,00
  SEGMENT         (3016,4015),10
  RECORD     05   HONSEKI,1,00,,,,AFTER
  FIELD           HONSEKI,ALPHA,00
  RELATION        BU,KA,CHAIN
  RELATION        KA,ZYUGYOIN,CHAIN
  RELATION        SENKO,ZYUGYOIN,CHAIN
  RELATION        ZYUGYOIN,SYUGAKU,CHAIN
  RELATION        ZYUGYOIN,KYUYO,CHAIN
  RELATION        ZYUGYOIN,GENZYUSYO,LINK
  RELATION        GENZYUSYO,HONSEKI,LINK
  END
```

図4.1

4.2 CIS—LOADER

CIS－LOADER は CIS－ALLOCATORにより準備されたディスクに対してデータを格納してデータベースを作成するものである。入力ファイルは磁気テープより入力し，その形式には2つある。

形式2： ブロッキングされた固定長レコード形式

形式4： ブロッキングされた可変長レコード形式

形式2の場合は，1つのブロックはサブファイルに含まれるすべてのセグメントのフィールドからなる。

形式4の場合は，1つのブロックは1つのセグメントに対応するフィールドからなる。そして各ブロックはサブファイルのレベル順に並び，これを周期として入力される。

図4.2のようなサブファイルのとき，形式2，形式4の1つのブロックの内容はつぎのようになる。

例

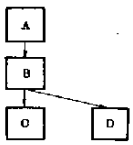
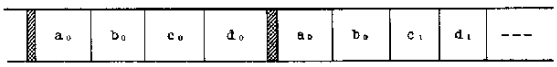


図4.2

形式2



形式4

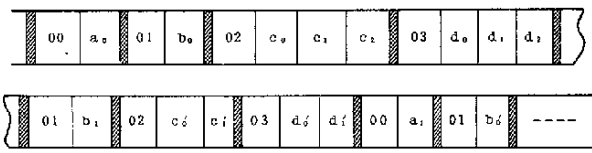


図4.3

(1) 入力ファイルの定義

以下に示すように入力ファイルを定義する。

① DEFINE INPUT

② FORM { 2 / 4 }

③ LABEL { 80 / 120 }, file identification

④ BLOKSIZE 10進数

⑤ RECSIZE 10進数

⑥ PARITY { EVEN / ODD }

⑦ FILE　　sul-file character

⑧ RECORD

形式2の場合と形式4の場合とで異なる。

- 形式2の場合

RECORD　record-name (X_1, X_2)

入力レコードの定義でレコード名とその磁気テープ上の位置を(X_1 , X_2)で表わす。

LOAD　record-name ($\left\{\begin{matrix}M\\L\end{matrix}\right\}$, Y_1 , Y_2)

ディスク上に格納するレコード名の定義。このレコードがすでに格納されている場合はM，格納されていないときはLと指定。Y_1 , Y_2 はこのレコードのソートの対象となるフィールドの指定。

CONDITION　$\left\{\begin{matrix}S\\省略\end{matrix}\right\}$

このサブファイルがソートされているときはS，そうでないときは省略。

- 形式4の場合

00　　record-name, ……, record-name

01　　record-name, ……, record-name

⋮

(2) データの定義

つぎのようなパラメーターを与えてデータを定義する。

① DEFINE　CIS, (P_1 , P_2), (P_3 , P_4 , P_5)

機能：　authority, data name, segment, structure, work area の格納範囲をページ番号で示す。

② SUBFILE　subfile character

機能：　サブファイルを定義。subfile characterはこのサブファイルにつけられる2桁のコード。

③ SEGMENT　(Pa, Pb), record number, {authority level}

機能：　セグメントの定義。Pa，Pbはこのセグメントの格納開始，終了ページ。record numberはパラグラフに格納する最大レコード数。

④ RECORD ij record name, Pointer field の個数, size,

{NEXT}, {PRIOR}, {MASTER}, $\left\{\begin{matrix}\text{BEFORE}\\\text{AFTER}\\\text{FIRST}\\\text{LAST}\\\text{AS (XX, X'X')}\\\text{DS (XX, X'X')}\end{matrix}\right.$

機能： レコードの定義。

- ij このレコードのサブファイル内でのレベル。
- size はこのレコードの長さ。
- NEXT, PRIOR, MASTER はポインタフィールドの指定。
- BEFORE, AFTER, FIRST, LAST, AS, DSはチェイン またはリンク中にレコードを並べる順序を指定する。
- BEFORE …… 下位レコードをチェインまたはリンクのカレントレコードの直前に挿入する。
- AFTER …… 下位レコードをチェインまたはリンクのカレントレコードの直後に挿入する。
- FIRST …… 下位レコードを，そのマスタレコードに対して，チェインまたはリンクにおける最初の下位レコードとして挿入する。
- LAST …… 下位レコードを，そのマスタレコードに対して，チェインまたはリンクにおける最後の下位レコードとして挿入する。
- AS(XX, X'X')…… レコードを Ascending ソートしてチェインまたはリンクを作る。XXはソートキーのフィールドの位置，X'X'はその桁数を示す。
- DS(XX, X'X')…… レコードを Descending ソートしてチェインまたはリンクを作る。

⑤ FIELD field name, {ALPHA / NUMERIC}, size, {UNIQUE}

機能： フィールドの定義。

- ALPHA …… このフィールドがアルファベットである。
- NUMERIC …… このフィールドが数値である。
- size …… このフィールドの長さ。
- UNIQUE …… このフィールドの内容がチェイン内においてユニーク。

⑥ RELATION master record-name, subordinate record-name, {CHAIN / LINK}

機能： 上記セグメントとマスタセグメントとの関係を定義。

- master record-name …… 上記セグメントのマスタになるレコード名。
- subordinate record-name …… 上記セグメントの下位になるレコード名。
- CHAIN, LINK …… 上記セグメントとマスタセグメントとの関係を示す。

5. 言 語 概 説

通常の COBOL の DATA DIVISION の FILE SECTION のあと, WORKING-STORAGE SECTION の前に CIS SECTION を設ける。

```
CIS SECTION
PAGE IS nnnn CHARACTERS
PAGE BUFFER CONTAINS XX PAGES
```

次に CIS 言語の説明をする。

(1) OPEN

機能： データベースの処理を可能にし，そのプログラムが用いる Retrieve table を完成する。

形式： OPEN CIS FOR { UPDATE [WITH BACK UP] / RETRIEVAL }

[AUTHORITY WORD IS { identibier / literal }.]

- WITH BACK UP を指定すると更新データのテープダンプを行なう。

(2) RETRIEVE

機能： OPEN で完成する Retrieve table に情報を与える。

形式： RETRIEVE data-name-1 [AND data-name-2 ……]

WHERE field-name-1 condi { identifier-1 / literal-1 }

[AND field-name-2 condi { identifier-2 / literal-2 }] ……

- condi には EQ(=), LL(<), LE(≤), HH(>), HE(≥), NE(≒)がある。
- OR を用いるときは括弧でくくる。

  A AND B AND (C OR D) AND E

  ((A AND B) OR C) AND D AND E

- condi を用いて範囲を指定できる。

  field-name HE α AND field name LE β は

  $\alpha \leq$ field name $\leq \beta$

  となる。

(3) MOVE

機能： RETRIEVE で求めた data-name をワーキングストリジエリアに移送する。

形式： MOVE〔PRIOR〕 data-name TO identifier

〔FOR UP DATE〕「AT END GO TO procedure-name」

(4) REPLACE

機能： ファイル内のデータの更新処理を行う。

形式： REPLACE data-name BY identifier

・ FOR UP DATE を用いて MOVE により ワーキングストリジェリアに移送された Field の内容を identifier の内容で書き換える。

(5) INSERT

機能： ファイル内に新しいデータを挿入する。

形式： INSERT identifier TO record-name,

(6) DELETE

機能： ファイル内のデータの削除を行う。

形式： DELETE data-name.

・ FOR UP DATE を用いて MOVE により ワーキングストリジェリアに移送されたレコードおよびそのレコードより下位のチェインに属するレコードを削除する。

(7) STORE REFERENCE CODE

JOIN

機能： 下位レコードに対して，そのマスタレコードを変更する。

形式： STORE REFERENCE CODE OF data-name

JOIN { 〔ALL〕 data-name-1 TO data-name-2. | data-name-1 TO 〔ALL〕 data-name-2. }

・ STORE により data-name-1 のリファレンスコードを保存する。そして JOIN により data-name-1なるレコードは data-name-2なるレコードをマスタとするか，または data-name-2なるレコードのマスタとなる。

・ ALL を指定すると，このチェインのすべての下位レコードはすべてマスタが変更される。

(8) CLOSE

機能： ページバッファ上で Write-switch が ON のページをディスク上に格納した後，ファイルを格納する。

形式： CLOSE CISS

例

```
43610        CIS-COBOL                                  PAGE 001

0001        COBOL *HEADER
0002        OPTION                  RETRIE    RLM
0003              IDENTIFICATION DIVISION.
0004              PROGRAM-ID.  CIS-COBOL
0005              ENVIRONMENT DIVISION.
0006              CONFIGURATION SECTION.
0007              SOURCE-COMPUTER.  NEAC SERIES 2200 MODEL 400.
0008              OBJECT-COMPUTER.  NEAC SERIES 2200 MODEL 400.
0009              INPUT-OUTPUT SECTION.
0010              FILE-CONTROL.
0011                  SELECT PRINT-F ASSIGN 10 PRINTER MW1.
0012                  SELECT CARD-F ASSIGN TO CARD-READER MW2.
0013              I-O-CONTROL.
0014                  APPLY PRINTER-CONTROL ON PRINT-F.
0015              DATA DIVISION.
0016              FILE SECTION.
0017              FD CARD-F LABEL RECORD ARE OMITTED.
0018                  RECORD CONTAINS 80 CHARACTERS
0019                  DATA RECORD IS CARD-R.
0020                  01  CARD-R.
0021                      02  CARD10  PICTURE X(10).
0022                      02  CARD20  PICTURE X(10).
0023                      02  FILLER PICTURE X(60).
0024              FD  PRINT-F LABEL RECORD ARE OMITTED
0025                 RECORD CONTAINS 120 CHARACTERS
0026                  DATA RECORD IS PRINT-R.
0027                  01  PRINT-R.
0028                      02  FILLER PICTURE X(10).
0029                      02  PRINT20 PICTURE X(10).
0030                      02  FILLER PICTURE X(10).
0031                      02  PRINT50 PICTURE X(20).
0032                      02  FILLER PICTURE X(70).
0033              CIS SECTION.
0034                      PAGE IS 1000.
0035                      PAGE BUFFER CONTAINS 6 PAGES.
0036               PROCEDURE DIVISION.
0037               OPEN-RTN.
0038                  OPEN INPUT CARD-F OUTPUT PRINT-F.
0039              CIS-OPEN.
0040                      OPEN CIS FOR RETRIEVAL.
0041               READ-RTN.
0042                  READ CARD-F AT END GO TO END-RTN.
0043              RRET-01.
0044               RETRIEVE NAME AND ADDRESS WHERE DIVISI = CARD10 AND
0045               SECTI EQ CARD20 AND (KYURYO HE :50000: OR
0046                                    KYURYO LE :15000: ).
0047              MOV01 MOV.
0048                      MOVE NAME TO PRINT20 AT END GO TO STEP-1.
0049              MOVE01.
0050                      MOVE ADDRESS TO PRINT50.
0051               PRINT-RTN.
0052                  WRITE PRINT-R BEFORE ADVANCING 2 LINES.
0053                  GO TO READ-RTN.
0054              END-RTN.
0055                  STOP RUN.
0056              STEP-1.
0057                  STOP :NOT FOUND NAME:.
0058        END CONV
```

図4.4

参考文献

1) 日本電子工業振興協会：

CISS (Consolidated Information Storage System),

昭和43年3月, 147P.

附－32　IDS

1.　システム概説

IDS（Integrated Data Store）は，磁気ディスク装置を使う汎用のファイルシステムである。IDS では，まず情報の構造をデータレコードの意味内容と関連させて記述する。一度データが記述されると，IDS システムは，磁気ディスク装置のハードウェア側からの要求に合うように，自動的にそのデータを組立てる。意味上の関連にしたがって，データレコードを構成する仕事はIDS システムが行なう。レコードの関連づけにはチェインを用いる。チェインはレコード間のクロスレファランスのリンケージを作る。

IDS 言語は COBOL をベースとしてその機能を拡張したものである。基本的な機能は STORE，RETRIEVE，MODIFY，DELETE の４つである。RETRIEVE は，磁気ディスク装置からレコードを検索して，COBOLで処理ができるようにコアメモリへデータを転送する。

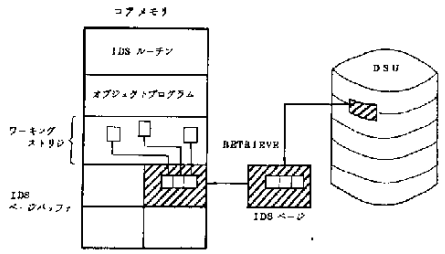


図 1.1

IDS システムに必要なハードウェアの最小構成

TOSBAC-5400 シリーズ

記憶容量	１６K語
磁気ディスク装置	1
磁気テープ装置	4
カード読取り装置	1
カード穿孔装置	1
ラインプリンタ装置	1

GE-600 シリーズ

１つの磁気ディスク装置を含むシステムなら，いずれでも可能。

IDS はユニ・プログラム方式においてのみ働くように作られている。

IDS/COBOLのソースデックは IDSトランスレイターにより COBOL に受け入れられるソースデックに変換される。

2. データ構成

2.1 IDSチェイン

IDS のフィジカルな入出力の単位はページである。ページはプログラマによって指定された一定の数のディスクセクタからなる。関連のあるレコードは，リンクされ同一ページにストアされる。

各ページの最初にはページヘッダレコードがある。その内容はつぎのようである。

① そのページのリファレンス・アドレス

② 余白スペースについての情報

③ ページが，リトリーバルされてから変更を受けたかどうかを示す情報

④ カルキュレイテッド・レコードのチェインの最初のレコードのアドレスを示す情報。このレコードはすべて，このページにランダマイズされて置かれている。

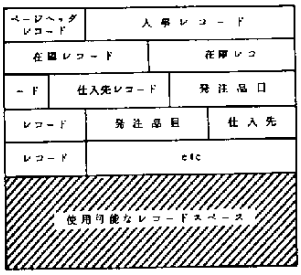


図2.1.1 IDS ページ

⑤ そのページで指定することができるライン・ナンバー。

ページ内の各 IDS レコードは，図 2.1.2 のような構成をしている。

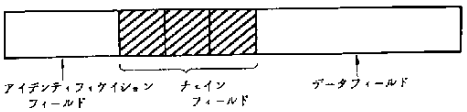


図 2.1.2

(1) アイデンティフィケイション・フィールドの内容

① リファレンスコードのラインナンバー

② レコードタイプ（たとえば在庫レコード，給与レコードなど）

③ レコードの長さ

(2) チェイン・フィールドの内容

チェインフィールドには，別の IDS レコードのリファレンスコードが含まれている。

(3) データ・フィールドの内容

1組のデータ。固定フォーマット，固定長。数字，英字または英数字。

2.2 レコードクラス

レコードの処理において，レコードを他のレコードから一意に区別する方法として，3種類ある。

(1) カルキュレイテッド・レコード

1つ以上のデータフィールドの内容がランダマイズルーチンによって処理され，ページナンバーが決められるレコード。レコードはこの頁または近接のページにストアされる。

(2) セカンダリー・レコード

ある特定のマスタレコードにチェインで結ばれていて，そのチェイン内でのソートコントロー

ルフィールドによって他のレコードと区別されるレコード。

(3) プライマリ・レコード

リファレンスコードを通じて，他のレコードと区別されるレコード。

2.3 IDSチェイン

図2.3.1はIDSチェインをあらわしている。

あるチェインのマスターレコードは，そのチェインのディテールレコードが共通に持っている固定情報を記述する。見方を変えれば，ディテールレコードは，マスターレコードに関係する可変情報を記述している。

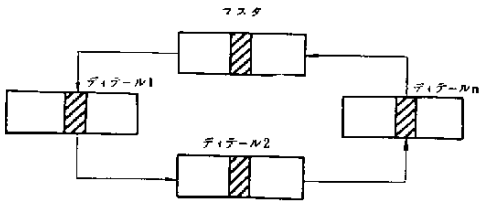


図2.3.1 IDSチェイン

(1) IDSチェインの特徴

① シンボリックネームを持ち，ただ1つのマスターレコードを持つ。

② 1つのチェインに含まれるディテールレコードのタイプと数は任意。

③ チェインのすべてのレコードは循環するループをなす。

④ 1つのレコードは任意個のチェインのマスターまたはディテールとなることができる。

⑤ レコードは，直接的にも間接的にも自分自身のディテールにはなれない。

(2) IDSチェインの種類

IDSチェインには，NEXT，PRIOR，MASTERの3種類がある。

① Chain - NEXT

そのチェインの次のレコードのリファレンスコード。chain-NEXTはすべてのレコードに自動的につけられる。

② Chain - Prior（オプショナル）

そのチェインの1つ前のレコードのリファレンスコード。

③ Chain - Master（オプショナル）

そのチェインのマスターレコードのリファレンスコード。

(3) チェイン・オーダリング

IDSチェインの中のレコードの順序の指定には6種類ある。

① Sorted Within Type …… チェインのレコードは，他のタイプのレコードに関係なく，同タイプで分類した順序になる。

② Sorted …… チェインのレコードは，チェインに数種のレコードが入っていても，それを無視して1例に並べられる。この場合，種々のレコードのコントロール・フィールドは同じ大きさでなければならない。

③ First …… ディテールレコードは，そのマスターレコードに対して，チェインにおけ

る最初のディテールとして，追加される。

④ Last …… ディテールレコードは，そのマスターレコードに対して，チェインにおける最後のディテールとして，追加される。

⑤ Before…… ディテールレコードはチェイン中のカレントレコードの直前に挿入される。

⑥ After …… ディテールレコードはチェインのカレントレコードの直後に挿入される。

(4) プライム・チェイン

磁気ディスク装置のアクセスタイムは，最後にアクセスしたレコードと現在求めているレコードの位置関係に依存している。そこでIDSでは，チェインのマスターレコードにできるだけ近接した位置に，追加のディテールをストアする。1つのレコードが数個のチェインのディテールとして定義されているときは，1つのチェインがプライムチェインとして選ばれる。プライムチェインを選択する基準は，チェインの使用頻度である。

(5) IDS shorthand

レコード間の関係を図示する方法。

図2.3.2の意味

① システムには，マスターレコード(従業員に1つずつ)がいくつも含まれている。

② マスターレコードは，それぞれ定められた控除チェインのマスターである。

③ 各チェインには，ディテールレコードがいくつか含まれている。

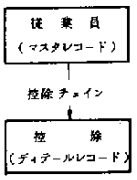


図2.3.2

IDS Shorthandを展開した書き方

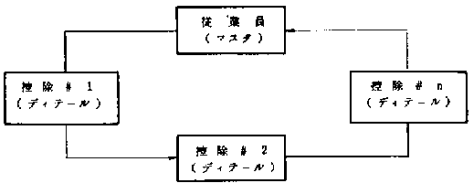


図2.3.3

3. 言語概説

IDS言語はCOBOLの拡張である。したがって，すべてのCOBOLのフォーマットと言語指定にしたがわなければならない。

3.1 IDENTIFICATION DIVISION

目的と用法は通常のCOBOLと同一である。

3.2 ENVIRONMENT DIVISION

通常の COBOL と異なる部分は INPUT/OUTPUT SECTION の FILE-CONTROL のパラグラフである。

形式： FILE-CONTROL. SELECT IDS file-name
ASSIGN TO hardware-device code-1 ;
〔hardware-device code-2 ;〕
〔BASE IS n_1 PAGES ;〕
〔PROGRAM REQUIRES n_2 BUFFERS〕

機能： ① ファイルネームの定義
② ファイルの入出力のチャネルとロジカルな装置番号の指定。
③ ファイルの磁気ディスク上での始点の指定。
④ コアメモリに保持されるべきページ数の指定。

3.3 DATA DIVISION

IDS ファイルは IDS Sectionと呼ばれる特別なセクションで記述される。このセクションは Working-Storage Section と Constant Section の間になければならない。

IDS Section は, File Description および Chain Definition よりなる。

3.3.1 FILE DESCRIPTION

形式： MD file-name 〔; PAGE CONTAINS n_1 CHARACTERS〕
; FILE CONTAINS n_2 PAGES〕

機能： IDS ファイルの物理的構造の記述。
有効なページサイズ
240, 480, 960, 1920, 3840 キャラクタ

3.3.2 RECORD DESCRIPTION

以下に述べる例外を除いて通常の COBOL の Record Description のすべての指定ができる。

OCCURS
EDITING CLAUSES
RENAMES
LEFT SYNCHRONIZED

IDS Record Description では, 実際のデータフィールドしか定義しない。レコードヘッダフィールドやチェインフィールドは定義しない。それらは固定フォーマットでシステム

が付け加える。

Record Description におけるレベル

① 02 レベルのフィールドエントリ

IDS トランスレータは IDS Section において出会った 02 レベルのエントリのそれぞれに対して特別なワーキングストリジェリアを作り出す。レコードが磁気ディスク装置から読み出されたとき，そのレコードがワーキングストリジに移されてからでなければ，データフィールドを使うことはできない。

② 02 REDEFINES

02 レベルにおけるエントリの REDEFINE。 ディスクには REDEFINE の エントリに対するスペースは用意されない。

③ 02 FILLER PICTURE X (n)

ディスク上に，n 字分の余分のスペースを用意する。

④ レベル 96 data-name-X

data-name-X は多重シンボルである。

⑤ レベル 97 data-name

この data-name のためにワーキングストリッヂにエリアをとっておくが，ディスク上には取っておかない。

IDS RECORD DESCRIPTION

形式： 01 data-name TYPE IS n_1

; RETRIE VAL VIA { data-name-1 FIELD / data-name-2 / CALC } CHAIN

〔; PLACE NEAR data-name-3 CHAIN〕

〔 PAGE-RANGE IS n_2 TO n_3 〕

〔 INTERVAL IS n_4 PAGES〕

機能： このレコードを IDS システムで処理するときに用いられる IDS の手法を指定する。

① TYPE IS n_1

IDS におけるレコードフォーマットを識別するために用いられるレコードタイプコードを指定する。n_1 は 1 ～ 9 9 9 の任意の値。

② RETRIEVAL

このレコードが RETRIEVE 動詞によってリトリーヴされる ときに用いられる方法を指定する。

i) RETRIEVAL VIA data-name-1 FIELD

レコードをそのリファレンスコードにより直接リトリーヴすることを用意する。このコードはレコードがディスク上にストアされるとき IDS によって作られ，ストアの直後に参照することができる。

ii) RETRIEVAL VIA data-name-2 CHAIN

レコードをチェインのディテールとしての関連にもとづいてリトリーヴする用意をする。

iii) RETRIEVAL VIA CALC CHAIN

レコードをその定義されたコントロールフィールドで乱数を発生させてリトリーヴする用意をする。

③ PLACE NEAR data-name-3 CHAIN

このレコードがディスク上にストアされるときに用いられる方法を指定する。指定されたチェインにおけるそのレコードの論理的位置の近くにストアする。

④ PAGE-RANGE IS n_2 TO n_3

レコードを IDS システムにおいて，グループにまとめる方法を用意する。n_2 ページから n_3 ページの間のページに，このレコードがストアされる。

⑤ INTERVAL IS n_4 PAGES

与えられたタイプのレコードが IDS システムで均一に分布するように保証する。同じタイプのレコードを n_4 ページ飛びにストアする。

3.3.3 CHAIN DEFINITION

形式1：

```
98 data-name-1 CHAIN MASTER ;
                          { SORTED WITHIN TYPE
                          | SORTED
                          | FIRST
   CHAIN-ORDER IS         | LAST
                          | BEFORE
                          | AFTER
   [; LINKED TO PRIOR]
```

形式2：

```
98 {data-name-1 | CALC} CHAIN DETAIL
   [; RANDOMIZE ON data-name-2 [; RANDOMIZE……] ]
  [                                                  ]
  [ ; DUPLICATES { ARE FIRST                         } ]
  [              { ARE LAST                          } ]
  [              { NOT ALLOWED                       } ]
```

```
[ ; SELECT { UNIQUE  } MASTER ]
  [ ;{ ASCENDING         } KEY IS data name-3
  [ ;{ DESCENDING        }
  [ ;{ ASCENDING RANGE   }
  [ ;{ ASCENDING         }  ..............  ]]
  [ ;{ DESCENDING        }
  [ ;{ ASCENDING RANGE   }
  [ ; MATCH-KEY IS data-name-4
    [ ; MATCH-KEY ..... ]]
  [ ; MATCH-KEY IS data-name-4
    [ SYN data-name-5 ]
  [ ; LINKED TO MASTER ]
```

機能： レコードがチェインのマスタであるかディテールであるかを指定し，チェインの性質を定義する。

① CHAIN ORDER

チェインのディテールレコードを順序づける規準を指定する。（ IDSチェインの種類参照 ）

② LINKED TO PRIOR

定義されたチェインの各データレコードにチェインフィールド PRIOR を用意する。

③ RANDOMIZE ON data-name-2

データファイルにおいてレコードの位置を決めるのに用いられる calculated チェインのレコードのフィールドを指定する。システムは指定されたフィールドすべてを使って計算する。

④ DUPLICATES { ARE FIRST / ARE LAST / NOT ALLOWED }

重複したデータレコードがチェインに存在するかどうかを指定する。

重複が許されているときは，追加されるディテールレコードは重複のつながりの FIRST または LAST に置かれる。

NOT ALLOWED が指定されたときは，重複したレコードがあるとエラー状態となる。

⑤ SELECT { UNIQUE / CURRENT } MASTER

多くの与えられたタイプから指定されたマスタレコードを選ぶ規準を指定する。

UNIQUE オプションは，新しいディテールレコードのデータフィールドの値とマスタレコードの MATCH KEY データフィールドの値が一致するマスタレコードを選ぶ。

CURRENT オプションは，新しいディテールレコードのマスタレコードとして，current マスタを選ぶ。

⑥ MATCH-KEY IS data-name-4

チェインのディテールレコードを一意に識別するためのデータフィールドを指定する。

⑦ { ASCENDING / DESCENDING / ASCENDING RANGE KEY } IS data-name-3

チェインのデータレコードの順序をコントロールするデータフィールドを指定する。

ASCENDING RANGE KEY というオプションは，例

ソーテッドチェインをいくつかの範囲に分割するときに用いる。たとえばディテールレコードが1～10000の間の値をとるとき，10個のレンジマスタを作り，それぞれに1000，2000，……，10000というコントロールフィールドを与える。それらをレンヂチェインとしてつなぎ，その下に実際のディテールレコードのチェインをつなげる。そして，STORE や RETRIEVE を速くするようにする。

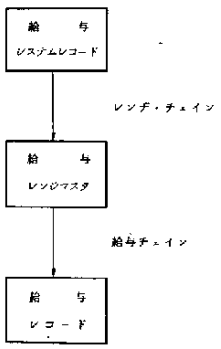


図3.3.1

⑧ SYNONYM

MATCH-KEY IS data-name-4 { SYNONYM / SYN } data-name-5

MATCH-KEY フィールドとして定義されたフィールドの代りの名前を指定する。

⑨ LINKED TO MASTER

各ディテールレコードにチェインのマスタレコードへのチェインフィールドを用意する。

例

PARTIAL DATA DESCRIPTION LISTING

```
01 L-REC;
        TYPE IS 100;
        RETRIEVAL VIA L-CODE;
        PAGE-RANGE IS 20001 TO 20001.
    02 L-CODE; SIZE 8 NUMERIC.
    98 M CHAIN MASTER;
        CHAIN-ORDER IS SORTED.

01 M-REC;
        TYPE IS 101;
        RETRIEVAL VIA M CHAIN;
        PAGE-RANGE IS 20001 TO 21000.
    02 EMPLOYEE-NAME; SIZE 25 ALPHANUMERIC
    02 EMPLOYEE-NO; SIZE 5 NUMERIC.
    02 SKILL-CODE; SIZE 4 NUMERIC.
    02 SKILL-NAME; SIZE 20 ALPHANUMERIC.
    02 UNIT-CODE, SIZE 4 NUMERIC.
    98 M CHAIN DETAIL;
        SELECT CURRENT MASTER;
        ASCENDING KEY IS EMPLOYEE-NAME;
        ASCENDING KEY IS EMPLOYEE-NO;
        DUPLICATES NOT ALLOWED.
    98 N CHAIN MASTER;
        CHAIN-ORDER IS AFTER.

01 N-REC
        TYPE IS 102;
        RETRIEVAL VIA N CHAIN;
        PAGE-RANGE IS 20001 TO 21000.
    02 SKILL-CODE; SIZE 4 NUMERIC.
    02 SKILL-NAME; SIZE 20 ALPHANUMERIC.
    98 N CHAIN DETAIL;
        SELECT CURRENT MASTER.
```

図3.3.2

3.4 PROCEDURE DIVISION

IDS Procedurae statement はディスクに関するデータのストアとリトリーヴを行なう。

ワーキングストリジエリアはIDSコントローラと COBOL Procedurc Division との間のリンケージをとる。これらのエリアは，IDS Section の Record Description により定義された各レコードタイプとフィールドにつき設定される。COBOL で IDS ファイルデータを使用する場合は，このエリアを参照する。

各IDS sentence の前には ENTER IDS clause がなければならない。

例

① ENTER IDS.

{STORE / RETRIEVE} record-name.

② ENTER IDS.

statement-1; 〔statement-2〕 ;……

③ ENTER IDS.

{STORE / RETRIEVE} record-name ; statement-1 〔statement-2〕 …… .

ここで statement-1, statement-2, ……は imperative statement でも conditional statement でもよい。

例2では, statement は有効に RETRIEVE または STORE された最後のデータレコードに作用する。

3.4.1 IDS Procedural Statement

(1) STORE data-name RECORD.

レコードを IDS データファイル中に記憶する。

(2) RETRIEVE

```
{ { data-name-1
  { CURRENT data-name-1 } RECORD

  { { NEXT
    { PRIOR  } RECORD OF data-name-2 CHAIN
    { MASTER }

    EACH AT END GO TO Procedure-name-1

    DIRECT                                 }
```

レコードをメモリバッファ内で利用できるようにする。

① data-name-1 はレベル 01 で定義された record-name.

② data-name-2 はレベル 98 で定義された chain-name.

③ DIRECT はリファレンスコードが DIRECT-REFERENCE と名づけられたコミュニケイションエリア内に記憶されている場合に使う。

④ EACH はリファレンスコードの順にリトリーヴする。

3.4.2 IDS Imperative Statement

Imperative statement は STORE と RETRIEVE の機能を拡張するものである。

(1) MOVE

形式1： MOVE TO WORKING-STORAGE

形式2： MOVE data-name-1 〔; MOVE data-name-2 〕……

形式1はレコードを, 形式2は指定されたフィールドをバッファからワーキングストリジに移す。

(2) HEAD data-name CHAIN ; HEAD ……

リトリーヴするレコードを含む階層構造のどれか高位のマスタレコードをリトリーヴしてワーキングストリージに移す。data-name はレベル 98 の chain-name でなければならない。

(3) MODIFY data-name 〔; MODIFY …… 〕

リトリーヴされたレコードのフィールドの内容を変更する。ワーキングストリジ内の指定されたフィールドの内容とバッファ内のレコードの内容を置き変える。

(4) DELETE 〔ON data-name-1 DETAIL〕

〔MOVE TO WORKING STORAGE〕

〔HEAD data-name-2 CHAIN, HEAD ……〕

〔PERFORM Procedure-name〕

〔GO TO Procedure-name〕

[{ OTHERWISE / ELSE } ON data-name-3 DETAIL 〔; ……〕]

リトリーヴされたレコードを従属するすべてのディテールレコードと共に削除する。

data-name-1 のタイプに従属するディテールに出会ったときは，処理を中断する。そしてそのレコードを処理する前に，つづく各種の imperative statement が実行される。

data-name-1 のタイプでないときは data-name-3のタイプと比べられる。以下同様である。

(5) GO TO Procedure-name-1

命令の通常の実行順序を変える。

(6) PERFORM Procedure-name-1

命令の通常の実行順序から離れ，指定された procedure を実行し，それから通常の順序に戻る。

3.4.3 IDS Conditional Statement

Conditional statement は基本的な IDS statement の論理的延長である。

(1) IF data-name-1 RECORD { GO TO …… / PERFORM …… }

[{ OTHERWISE / ELSE } { statement-1 〔; statement-2; ……〕 }]

指定されたレコードタイプのレコードが出てきたときに通常の処理の順序からコントロールを移す。

(2) IF ERROR statement-1

[{ OTHERWISE / ELSE } { statement-2 〔; statement-3 …… 〕 }]

IDS 機能の実行中に検出された論理的または物理的なすべてのエラーの発生を検出する。

(3) OPEN file-name

IDS ファイルをイニシャライズする。

(4) CLOSE file-name

IDS ファイルの処理を終了させる。

例

PARTIAL PROGRAM LISTING

```
      PROGRAM-ID.  DEMAND-REPORT-2.
      START.
*         OPEN DATA-BASE.
          OPEN OUTPUT REPORT-FILE.
          INITIATE ALL.
          STORE L-REC.
          MOVE 1110 TO SKILL CODE
*         RETRIEVE SKILL-CLASS-REC.
          MOVE 1129 TO CURRENT-CODE.
          GO TO A.
      AA.
          ADD 1 TO CURRENT-CODE.
          MOVE CURRENT-CODE TO SKILL-CODE.
          IF SKILL-CODE IS EQUAL 1140 GO TO F.
*         RETRIEVE SKILL-CLASS-REC.
*             IF ERROR GO TO AA.
      A.
*         RETRIEVE NEXT RECORD OF SKILLEE CHAIN.
*             IF SKILL-CLASS-REC GO TO AA.
          MOVE DIRECT-REFERENCE TO SAVE-REF.
*         IF PERSONNEL-REC GO TO B.
*         RETRIEVE MASTER RECORD OF SKILL CHAIN.
      B.
*         MOVE TO WORKING-STORAGE.
          IF LEVEL IS "HEAD" GO TO E.
          IF SEX IS NOT "M" GO TO E.
*         HEAD SKILLEE CHAIN.
*         STORE M-REC.
*             IF ERROR GO TO E.
      D.
*         RETRIEVE NEXT RECORD OF SKILL CHAIN
*             IF PERSONNEL-REC GO TO E; ELSE
*             HEAD SKILLEE CHAIN.
*         STORE N-REC.
          GO TO D.
      E.
          MOVE SAVE-REF TO DIRECT-REFERENCE.
*         RETRIEVE DIRECT.
          GO TO A.
      F.
*         RETRIEVE CURRENT L-REC.
      G.
*         RETRIEVE NEXT RECORD OF M CHAIN;
*             IF L-REC DELETE; GO TO I; ELSE
*             IF M-REC MOVE TO WORKING-STORAGE.
          GENERATE LINE-M.
*         DELETE;
*             ON N-REC DETAIL MOVE TO WORKING-STORAGE; PERFORM H.
          GO TO G.
      H.
          GENERATE LINE-N.
      I.
          TERMINATE ALL.
          CLOSE REPORT-FILE.
*         CLOSE DATA-BASE.
          STOP RUN.
```

図 3.4.1

参 考 文 献

1) CHARLES W. Bachman

INTEGRATED DATA STORE

GENERAL ELECTRIC Oct. 1967 86P.

2) 東京芝浦電気株式会社

IDS 入門, TOSBAC-5400 システム概説書 56P. 1965年

3) 東京芝浦電気株式会社

IDS/COBOL, TOSBAC-5400 システム概説書 101P. 1966年

附－３３　汎用言語の比較

汎用言語を用いて情報検索システムを作成するときの機能比較表

F：FORTRAN　　1：言語機能として持つ。

C：COBOL　　2：簡単にプログラムできる。

P：PL／1　　3：困難である。

S：SNOBOL　　4：できない。

L：LISP　　－：意味がない。

機　　能	F	C	P	S	L
① 可変長レコードの扱い	4	1	1	－	－
② アレイ処理（dimension）	1	1	1	4	4
② レベル構造の処理	4	1	1	4	4
② リスト構造の処理	4	3	2	1	1
③ 表を引く機能（search 命令）	2	1	2	4	4
③ チェインをたどる機能	2	2	1	4	4
④ 文字処理	4	3	2	1	1
④ 四則演算	1	1	1	2	3
④ マトリックス演算	2	2	1	－	－
⑤ 分　　類	3	1	2	－	－
⑥ 作表（Report Writer）	3	1	2	4	4
⑦ 拡張された入出力（Disk, Display）	4	1	1	－	－
⑧ 非同期処理	4	1	1	4	4

索　　引

（ＡＢＣ順）

（50音順）

昭和44年3月発行

発行所　社団法人　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　三協印刷株式会社
東京都渋谷区渋谷3－11－11
TEL（407）7316

43－S－003