データベース構築促進及び技術開発に関する報告書

バイオノモデルに基づく先端的文書検索システムの開発

平成2年3月

財団法人　データベース振興センター

委託先　シャープ株式会社

本報告書は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて作成したものである。

序

データベースは、わが国の情報化の進展上、重要な役割を果たすものと期待されている。今後、データベースの普及により、わが国において健全な高度情報化社会の形成が期待される。さらに海外に対して提供可能なデータベースの整備は、国際的な情報化への貢献および自由な情報流通の確保の観点からも必要である。しかしながら、現在わが国で流通しているデータベースの中でわが国独自のものは１／４にすぎないのが現状であり、わが国データベースサービスひいてはバランスある情報産業の健全な発展を図るためには、わが国独自のデータベースの構築およびデータベース関連技術の研究開発を強力に促進し、データベースの拡充を図る必要がある。

このような要請に応えるため、(財)データベース振興センターでは日本自転車振興会から機械工業振興資金の交付を受けて、データベースの構築および技術開発について民間企業、団体等に対して委託事業を実施している。委託事業の内容は、社会的、経済的、国際的に重要で、また地域および産業の発展の促進に寄与すると考えられているデータベースの構築とデータベース作成の効率化、流通の促進、利用の円滑化、容易化などに関係したソフトウェア技術・ハードウェア技術である。

本事業の推進に当って、当財団に学識経験者の方々で構成されるデータベース構築・技術開発促進委員会（委員長　東京工科大学教授　西野博二氏）を設置している。

この「バイナリモデルに基づく先端的文書検索システムの開発」は平成元年度のデータベースの構築促進および技術開発促進事業として、当財団がシャープ株式会社に対して委託実施した課題の一つである。この成果が、データベースに興味をお持ちの方々や諸分野の皆様方のお役に立てば幸いである。

なお、平成元年度データベースの構築促進および技術開発促進事業で実施した課題は次表のとおりである。

平成２年３月

財団法人　データベース振興センター

平成元年度データベース構築・技術開発促進委託課題

分野	課題名
社会	1　気候情報データベースの構築 2　電磁波環境障害に関するデータベースの構築 3　災害情報シソーラスの構築 4　意味情報を中核とした医療評価データベースとコミュニケーションシステムの構築 5　ハンディキャップパーソンの情報ニーズに即したライフサポートデータベースの構築 6　博物館情報データベースシステムの構築 7　中央省庁での電子計算機利用に関する報告書のデータベース化
地域活性化中小企業振興	8　沖縄地域における文化情報データベースの構築 9　九州地域の人材情報データベース構築に関する調査研究 10　高岡市商圏データベースの構築 11　地域の物産・人材・文化情報のデータベース構築と新しい地域間交流推進に関する調査研究
地図	12　マルチメディア型地図データベース構築のための調査研究
エネルギー・資源	13　燃焼技術と燃焼装置設計のデータベース作成
部品・材料	14　技術支援システムにおける産業機械部品データベースの構築 15　マイクロコンピュータのプログラマブル周辺デバイスのデータベース構築
標準化	16　イオンクロマトグラィー・データベースの構築 17　CD－ROMマルチメディアデータフォーマットの調査
海外	18　データベース構築のためのターミノロジーの調査研究
技術	19　異種データから構成されるデータベースの総合的処理技術に関する調査研究 20　バイナリモデルに基づく先端的文書検索システムの開発

＜目次＞

１．本開発の概要

１．１　目的

オフィスの情報は、事務の経過はファイルされているが情報間の関連付けがされていないことが問題である。この問題を解決するために、当社は昭和６２、６３年度、(財)データベース振興センターより補助事業を受託し集合論に基づく知的情報検索システムＳＴＤＢ(Ｓet　Ｔheory　Ｄata　Ｂase)を開発した。昭和６２年度、キーワードの同義語とシソーラスに基づき検索するシステムを開発し、昭和６３年度、同義語とシソーラスを定義するツールを開発した。ＳＴＤＢを使い医薬品情報や修理伝票情報を検索するシステムを試作し、データベース'８９等にて見学者の好評を博した。

ある文書を検索しその文書を読んで、他の関連した文書も必要であることが初めて分かることも多い。従って、検索した文書に現れる語句をキーワードとし他の文書を検索するための、再検索機能が強く求められている。

検索作業者がキーワードを指定すると、ＳＴＤＢはそのキーワードを含む文書を検索する。検索された文書はキーワードを含んでおり、更に他のキーワードも文書内に含んでいる。文書内の他のキーワードを新たに指定すれば、元の文書に関連する他の情報を容易に追加検索できる。又、文書ファイルグループを指定する画面であるメニューや、キーワードの意味と関係を指すファイルを文書と考えれば、ＳＴＤＢの機能についてもその操作を単純化できる。ここで、ＳＴＤＢの基礎理論にキーワードのチェーンをテキストとみなす全く新しい考え方(キーワードチェーンテキスト)を導入し、ＳＴＤＢを文書とメニューとを統合するさらに単純で使い易いシステムとする。

１．２　内容

本開発の要旨は次の通りである。キーワードチェーンテキストを効率良く取り扱う検索システムをバイナリモデルに基づき試作した。その理論的背景、データモデル、検索プログラムのそれぞれについて検討を加え、高度な実用的システムの一部を開発した。

個々の作業内容を以下に述べる。

(１)順序を持つ集合(オーダードセット)を基礎理論とした。

オーダードセットは情報の順序に注目した表現方法であり、キーワードチェーンテキストを表現するために適した基礎理論と言えよう。ＳＴＤＢが持つ検索情報に順序性を取り入れテキスト検索の基礎理論を拡張し、さらに一般化した。

(２)バイナリモデルに関して調査し検討を加えた。

バイナリモデルは実体と関係を基本要素とし実体間の関係を双方に与える点が特徴で、記述力と柔軟性に富み、明解にモデルを表現が可能である。この為、本モデルは概念モデルの内で注目を集めている。一般に文書とキーワードの関係は「文書はキーワードを含んでいる」と「キーワードは文書に含まれている」の双方向性を考慮する必要があろう。この

関係をバイナリモデルであらわしインデックス情報を単一化した。又、バイナリモデルに会話的かつ動的な機能を追加する新たなデータモデルを作成した。

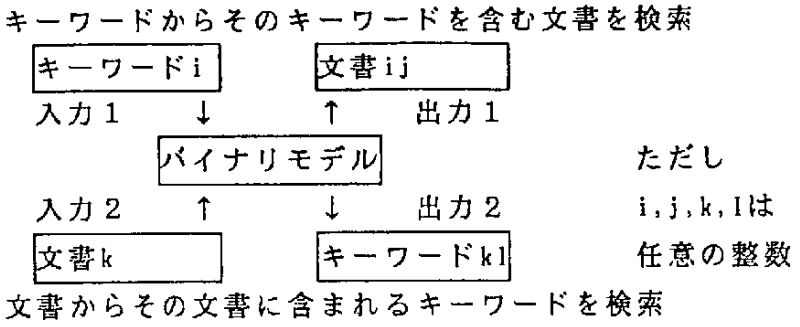


図1－1　バイナリモデルの文書検索システムへの適用

(3)バイナリモデルのデータを高速に検索するプログラムを開発した。

バイナリモデルは複雑かつ情報量に富むため、これを会話的かつ動的に取り扱う為に並列処理が適していることがわかった。マルチプロセッサを用い、複数のディスク上に存在するデータを検索するシステムを検討した。高速応答かつ柔軟な会話処理を実現した。大規模データに対応する為、他のキーワード管理手法を考察し、システムを試作した。

(4)開発したプログラムを基に文書検索システムを試作した。

"薬品名"、"識別コード"、"製造会社名"、"販売会社名"、"効能効果"等の情報を持つ医薬品に関する文書を対象とし、医薬品情報検索システムを試作した。薬品識別コードに基づき検索を行い、その薬品を販売する会社名を知り、その識別コードと得られた会社名をキーワードとし新たに他の医薬品についての情報を検索可能となった。

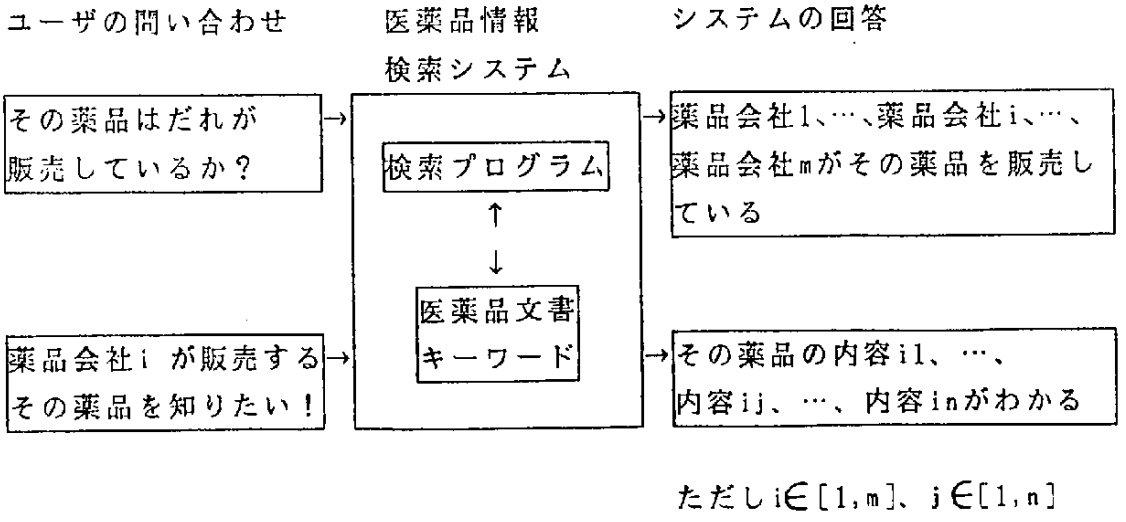


図1－2　医薬品情報検索システムでの検索例

"機種名"、"受付内容"、"処理内容"などの情報を持つ障害報告に関する文書を対象とし、修理情報検索システムを試作した。品名と品番の対応表を利用し、シソーラスを関係登録する手間を省くことを可能とした。又、ＯＥＭ製品に対しての品番の対応表を利用し、同義語(シノニム)を入力する手間を省くことが可能となった。

開発には当社のミニメインフレームコンピュータＩＸ－１１を用いた。ＩＸ－１１は一般のＵＮＩＸにデータベース管理システムに適した機能を追加したコンピュータであり、プログラム開発及びデータベース運用に最適な環境を提供している。

ＩＸ－１１の主たる特徴を下記に記す。

(1)標準ＵＮＩＸ　ＯＳ

独自のデュアルユニバース機構により、現在ＵＮＩＸの２大標準である、バークレイ版４ＢＳＤとＡＴ&Ｔ版ＳＹＳＴＥＭ　Ｖを統合し完全にサポートしている。双方のプログラミング環境を同時に使用したり、一方の環境で開発されたプログラムを他方の環境でも使用可能である。

(2)シンメトリックマルチプロセッサ

独自のＲＩＳＣ(縮小命令セットコンピュータ)プロセッサを採用し、プロセスの実行を終了したプロセッサが次の処理待ちプロセスを実行するアーキテクチャを構成している。プロセッサの増減に容易に対応でき、耐故障性や拡張性に優れている。

(3)プロセス管理機能の拡張

プロセスの実行を特定のプロセッサに固定する機能や、プロセスの実行優先度を常に最上位にする機能を付加している。これらの機能により、データベースアプリケーション実行処理能力ではメインフレームクラスに匹敵する。

(4)仮想ディスク機構

仮想ディスクの採用により、ストライプディスク、コンキャティネイトディスク、ミラーディスク、メモリー(ＲＡＭ)ディスクを実現している。これにより、既存のプログラムを変更することなく、ファイルシステムについての実行効率や機能を向上できる。

１．３　効果

１．３．１　キーワードチェーンテキスト

文書検索ではシソーラスから検索に必要なキーワードを探すのが一般的と言えよう。キーワードチェーンの考えに基づくとキーワードを構造化して記述した文書、例えば用語集は、それに示された用語をキーワードについてのシソーラスとみなすことができる。この

ため、キーワードについての一覧表など、情報を集約した文書はシソーラスを代替出来る。以下に例を示す。

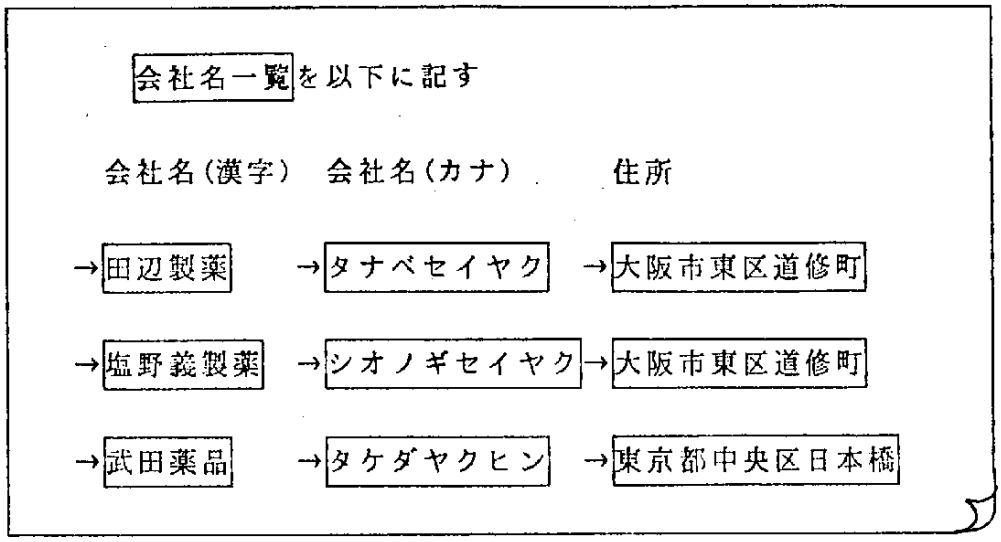


図１－３　キーワードチェーンの例

この機能を使用し、以下に示すような検索が可能となる。これは、開発システムは、キーワードシソーラスを文書化し利用できるだけでなく、文書中にあらわれるキーワードを使い再検索可能であることも示している。

表１－１　医薬品文書を検索する例

	操作１	操作２	操作３
検索操作	キーワード"＊０２６"を用いて文書を検索	キーワード"会社名一覧"を用いて、会社名を一覧表にした文書を検索	キーワード"解熱"と"田辺製薬"をＡＮＤに組み合わせ文書を検索
検索文書	識別コード＊０２６ 会社　塩野義製薬 効能効果　解熱 ：	会社名一覧 田辺製薬 塩野義製薬 ：	識別コードＡＳＡ２００ 会社　田辺製薬 効能効果　解熱 ：
結果内容	キーワード"塩野義製薬"、"解熱"が得られる	シソーラスを使わずに会社名のキーワード"田辺製薬"が得られる	"解熱"、"田辺製薬"の両方のキーワードを含む文書が得られる

１．３．２　バイナリモデルの特徴

情報の関係を表現する一般的モデルであるネットワークモデルを用いて実体間に双方向

の関係を表現すると、それぞれ独立した２つの関係が必要である。これらの関係を会話的(リアルタイム)に更新する場合では２つの関係の一貫性を保証されにくい。

バイナリモデルはデータモデルの１つであり、実体とそれらの関係を用いて情報を表現する。実体間の関係を常に対にして与えることからバイナリモデルと呼ばれる。バイナリモデルを用い双方向の関係を１つに表現すると、会話的な処理において更新時の一貫性が保証できる。

文書検索では、あるキーワードを含む文書を検索することとが一般的である。しかし、上で示したように、文書に含まれるキーワードを知る機能も極めて重要な意味を持っている。バイナリモデルを用いると文書とキーワードの関係を単一に出来、インデックスファイルを格納するために必要な記憶容量を小さく出来る。バイナリモデルに基づきシステムを作成し、以上の点が確認された。

表１－２　バイナリモデルとネットワークモデルの比較

	キーワードと文書の関係ダイヤグラム	特　　徴
バイナリモデル	キーワード ― キーワードと文書は包含関係を持つ ― 文書	・１つの関係が双方向の情報を表現 ・変更時に一貫性を保ち易い ・関係の情報は複雑かつ密度が高い ・効率良く取り扱う為に並列処理が必要
ネットワークモデル	キーワード → キーワードは文書に含まれる → 文書 キーワード ← 文書は キーワードを含む ← 文書	・双方向の関係をそれぞれ独立して表現できる ・表現の自由度が大きい ・変更時に一貫性を保ちにくい ・１つの関係を容易に効率良く取り扱える

１．３．３　開発プログラムの効果

現在、情報検索の新しい方法としてハイパーテキストシステム(以下ハイパーテキストと略す)が開発、販売されている。主たるものとしてＡpple社のＨyperＣard、ＸＥＲＯＸ社のＮoteＣards等がある。ハイパーテキストシステムはネットワーク状にリンクされた文書間を移動し必要な情報を得る検索システムである。文書間のリンク情報はユーザがあらかじめプログラムする。ハイパーテキストは検索条件を使用者が指定する文書検索とは機能が異なるが、関連する文書をたどる機能は文書情報の検索に有効な手法である。本開発で作成したプログラム(以下これをＳＴＤＢ２と呼ぶ)はＳＴＤＢとハイパーテキストの両者の特徴を持ち、かつそれらの問題点を解決した画期的システムである。ＳＴＤＢ２では文書内容がキーワードシソーラスやメニューファイルであり、キーワードが次の情報を得るためのリンク情報となる。この為、ＳＴＤＢ２を使えばユーザが新たな情報を作成しなくても従来のシステムの機能を兼ね備えた文書検索システムが構築可能で、上記の試作

システムと同様の検索があらゆる文書情報を対象に可能となる。

表１－３　ＳＴＤＢ、ハイパーテキストとＳＴＤＢ２との比較

	ＳＴＤＢ	ハイパーテキスト	ＳＴＤＢ２
検索方法	登録されているキーワードを一覧表やシソーラスから選択し検索条件とする	文書中にあらかじめ設定されたリンクをたどる	登録されているキーワードを一覧表、シソーラス、検索した文書から選択し検索条件とする
データベース作成	文書とキーワードからインデックスを自動作成 シソーラスとシノニムは独自に作成	文書間のリンクをあらかじめプログラミング 検索内容をあらかじめ決定	文書とキーワードからインデックスを自動作成 シソーラス、シノニムは文書でも指定可 文書間のリンクも自動作成

医薬品データに対してその実証を行った。従来医薬品データは膨大で複雑すぎ薬品名や会社名等の単純な指定でしか検索できなかった。当システムはキーワードやシソーラスをユーザがほとんど定義せずに錠剤に刻印された識別コードに他の情報、例えば会社名や症状名を条件に加え検索できるようになり病院等で実用に耐えうるシステムといえる。

１．３．４　他のキーワード管理手法との比較

大量のデータを管理する為にリレーショナルデータベース管理システム(ＲＤＢＭＳ)が使われている。キーワード数が大量になった場合、ＲＤＢＭＳにてキーワード管理する方法が効果的であると考え、システムを試作した。ＲＤＢＭＳとして米国カリフォルニア大学バークレー校にて開発されたパブリックドメイン版ingresを使用した。

キーワードについての情報をリレーショナルテーブルに表現し、これらのキーワード情報を取り出すための関数を作成した。これにはＲＤＢＭＳに付属のＣ言語インタフェースを使用した。このライブラリ関数を使い、ＳＴＤＢとＲＤＢＭＳを組み合わせた。本試作プログラムを実行し以下のことが確認された。

・ソートや集計処理は遅い。
・アクセス時間は管理するキーワード数から影響を受けにくい。
・キーワード管理部分と画面表示部分と分離しても実行スピードはさほど低下しない。

これから、キーワード管理について以下の結果を得た。

・キーワード数が少ないときはＳＴＤＢ独自の方式(独自方式)が有利である。一方、キーワード数が多い場合にはＲＤＢＭＳを使用した方式(ＲＤＢＭＳ方式)が有利である。両方式の有利性が交差する点は実行環境に依存する。
・決められたデータをまとめて取り出したり書き込んだりする場合はＲＤＢＭＳ方式が有

利である。
・全体を集計する処理や可変長データの処理は独自方式が有利である。
・総括して、どちらか一方のみに頼らず、両者を融合し状況により使い分ける機能が必要と考える。

1．4　今後の展開

今後以下の機能拡張を計画している。

(1)グループウエアへの対応

ＯＡシステムは個人の業務を向上させるために使用されている。コンピュータがネットワークシステムにて接続されるようになり、データの共有が可能となっている。これにより、グループでの業務に使用できるコンピュータシステム(グループウエア)についての要望がある。文書検索システムはグループでの文書共有化を進めるために役立ち、グループウエアの要素技術の一つと考えられる。今後、以下の既存技術と組み合わせ、グループでの協調作業を支援する。

・電子メール
・リレーショナルデータベースシステム

これら機能と組み合わせるために以下の開発課題がある。

・クライアントサーバアーキテクチャの採用
・システム稼働率の向上
・他アプリケーションとのインタフェース

(2)クライアントサーバアーキテクチャへの対応

システムの実行効率を向上させ、インタフェース部分に自由度を与えようとする場合、プログラムをサーバとクライアントにそれぞれ機能分割することが有効であることを説明した。開発プログラムを改良し、文書検索サーバとしての機能を実現することを計画している。以下の開発課題がある。

・サーバが提供すべき機能を使用するためのインタフェース
・クライアントを開発するためのライブラリ
・複数のクライアントからのタスク管理
・競合解消と処理最適化

2．研究開発の詳細

2．1　バイナリモデル文書検索システム

2．2．1　シソーラス型文書検索システム

ＳＴＤＢは、当社が(財)データベース振興センターから受託した補助事業の成果を利用し開発した文書検索システムである。あらかじめ登録されたキーワードについて、そのインデックス情報を使い、文書を高速に検索する。使用者は、検索キーワードを画面上から選択する。検索を行うと画面上に指定したキーワードを含む文書の内容が表示され、使用者は内容を読んで自分の必要な文書を探し出せる。また、検索された文書を他のＵＮＩＸコマンドに渡しワープロやプリンタに出力する為に、ＵＮＩＸとインタフェースを持つ。

キーワード選択・検索操作はスプレッドシートに似通った表画面上にて行う。表の各項目には、ポップアップメニューが設定されており、使用者はメニュー項目上にカーソルキーを移動し、操作を選択する。表と画面が一致しているため、会話的な操作が容易に行える。

以下に機能の詳細を述べる。

(1)文書検索機能

指定したキーワードを含む文書を、キーワード毎にあらかじめ設定されたインデックスを使い検索し、文書中のそのキーワードが出現する箇所を高速に表示する。この為、使用者は自分の必要とする文書かどうか実際の内容を読んで確認できる。

キーワードはＡＮＤとＯＲを使い組み合わせて指定できる。例えば、パソコンとオフコンについて検索する場合は"パソコン　ＯＲ　オフコン"と組み合わせ検索条件とする。又、パソコン用のＵＮＩＸについて検索するときは"パソコン　ＡＮＤ　ＵＮＩＸ"と組み合わせる。

(2)概念登録機能

ＳＴＤＢは検索条件をキーワードに登録する機能を持つ。例えば、"パソコン　ＡＮＤ　ＵＮＩＸ"を"パソコンＵＮＩＸ"というキーワードに登録可能である。この機能を概念登録と呼ぶ。キーワード"パソコンＵＮＩＸ"を使い文書を検索すると、自動的に検索条件"パソコン　ＡＮＤ　ＵＮＩＸ"に展開されて処理される。

この機能を利用して同義語(シノニム)を容易に定義できる。例えば、"オフコン"は"オフィスコンピュータ"の省略形で同一の言葉である。そこで、キーワード"オフコン"に"オフィス　ＯＲ　オフィスコンピュータ"を登録し、同義語を定義できる。又、"オフィスコンピュータ"は文書中で"オフィス・コンピュータ"、"オフィス用コンピュータ"、"オフィス向けコンピュータ"、"オフィスで使用されるコンピュータ"等と表現されることがあり、"オフィス　ＡＮＤ　コンピュータ"と考えると都合良い。更に、"オフィス"と"コンピュータ"についてもそれぞれ"オフィス　ＯＲ　ビジネス　ＯＲ　事務"、"コンピュータ　Ｏ

R　COMPUTER　OR　計算機"と同義語を定義できる。以上の内容を概念登録すれば、使用者は"オフコン"をキーワードとして選択するだけで以下の条件式を満たす文書がすべて検索される。

・オフコン	(オフコンを含む文書)
・オフィス　AND　コンピュータ	(オフィスとコンピュータを含む文書)
・オフィス　AND　COMPUTER	(オフィスとCOMPUTERを含む文書)
・オフィス　AND　計算機	(オフィスと計算機を含む文書)
・ビジネス　AND　コンピュータ	(ビジネスとコンピュータを含む文書)
・ビジネス　AND　COMPUTER	(ビジネスとCOMPUTERを含む文書)
・ビジネス　AND　計算機	(ビジネスと計算機を含む文書)
・事務　AND　コンピュータ	(事務とコンピュータを含む文書)
・事務　AND　COMPUTER	(事務とCOMPUTERを含む文書)
・事務　AND　計算機	(事務と計算機を含む文書)

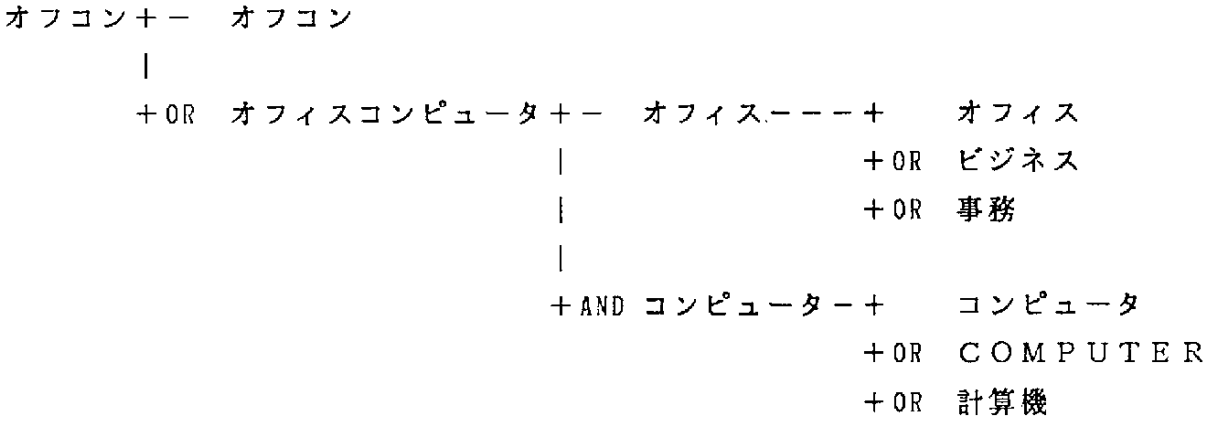
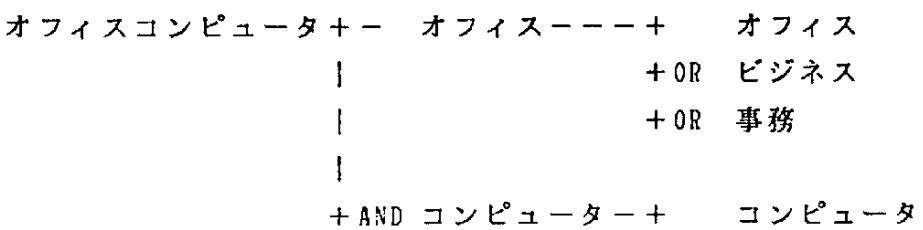


図2.－1　キーワード"オフコン"の概念登録例

キーワード"オフィスコンピュータ"に"オフィス　AND　コンピュータ　OR　オフコン"を定義しても同様の結果を得ることができる。この場合は以下の様に考えられる。



```
|                    +OR  COMPUTER
|                    +OR  計算機
|
+OR  オフコン
```

図２－２　キーワード"オフィスコンピュータ"の概念登録例

ＳＴＤＢは概念登録にＡＮＤ条件を使用できるので、複合語にも柔軟に対応可能である。先の例からも、特に日本語の様に複合語が多い文の全文検索に対して有効であることがわかる。これは一般の同義語検索機能にない特徴である。

(３)関係登録、関係表示機能

検索に使用したいキーワードを一覧表の中から探し出すことは難しい。さらに、自分の思い付いたキーワードが登録されていないこともある。キーワードを分類して登録し、比較的容易に探し出せる機能が求められている。

ＳＴＤＢはキーワードに下位のキーワードを登録する機能を持つ。この機能を使いキーワードを階層構造に分類でき、階層の木構造をたどり目的のキーワードを容易に探し出せる。この機能を関係登録と呼ぶ。又、関係登録されたキーワードの内容を、画面上に階層的に表示する機能を持ち、これを関係表示と呼ぶ。

コンピュータについて下図のような階層構造を定義出来る。

```
上位    ←                                    →    下位

コンピュータ  +－  パーソナルコンピュータ  +－ＡＸパーソナルコンピュータ
              |                            +－ＩＢＭパーソナルコンピュータ
              |                                  :
              +－  オフィスコンピュータ
              +－  スーパーコンピュータ
                        :
```

図２－３　キーワード"コンピュータ"の関係登録例

この機能を使用しキーワードのシソーラスを表現できる。関係登録された内容はあるキーワードについて３階層までを１度に表示できる。つまり、関係表示したキーワードに直接関係登録されたキーワード(子キーワード)と、それらのキーワードの内どれか１つについて関係登録されたキーワード(孫キーワード)、更にそれらのキーワードの内どれか１つについて関係登録されたキーワード(ひ孫キーワード)を表示できる。孫キーワード、ひ孫

キーワードはそれぞれ１度に１つずつ表示する。他の孫キーワード、ひ孫キーワードへ簡単に表示を変更できる。ひ孫以下のキーワードをたどることや、そのキーワードを関係登録しているキーワード(親キーワード)をたどることもできる。親キーワードは複数存在することができる。つまり、１つのキーワードは複数のキーワードに関係登録可能である。

関係登録されたキーワードの木構造を上位、下位にかかわらず容易にたどり、必要なキーワードを選択できる。

```
コンピュータ  ＋－－  オフィスコンピュータ    ←  オフィス、コン
              ＋－－  スーパーコンピュータ        ピュータのどち
                      :                          らの関係を表示
                                                 してもオフィス
オフィス      ＋－－  オフィスコンピュータ    ←  コンピュータが
              ＋－－  オフィスシステム            表示される。
                      :                          ただし、これら
                                                 は別々の内容を
                                                 登録できない。
```

図２－４　複数の親キーワードを持つ関係登録例

単語"コンピュータ"は"パーソナルコンピュータ"の上位概念である。このように、文字列はその文字列を含むより長い文字列の上位概念であることが多い。この現象に注目し、キーワードをそのキーワードに含まれる短いキーワードに自動的に関係登録する機能を用意している。例えば、図２－３、２－４に示した例はいずれも自動作成できる。

(４)ＵＮＩＸインタフェース機能

文書検索の最も重要な目的は文書の再利用である。この為、ＳＴＤＢは検索した文書をＵＮＩＸに引き渡すインタフェース機能を持つ。この機能を使用し、検索した文書をワープロソフトに引き渡せば、ワープロソフト上で文書を加工し、文書を再利用できる。又、文書中にその文書に関係する情報へのリンクをあらかじめ記入し、かつ、そのリンクをたどる手続きを用意すれば、簡単にそのリンクをたどる事ができる。例えば、文書ファイル中に関連する画像情報へのアクセス方法を記入していれば、検索した文書からそれに関連する画像情報を表示できる。

(５)適用分野

文書は重要な情報源であり、文書の共有は情報(知識)の共有を意味する。ＳＴＤＢを用いて、提案書、決裁書、報告書など部門内の文書情報を容易に検索できる。その他、新聞スクラップ情報なども容易にデータベース化できる。個人の知識を部門の知識とし利用可能となる。

現在、ＳＴＤＢは技術サービス会社にて修理情報を始めとする技術情報システムの検索システムとして稼働中である。過去の修理記録を修理伝票から検索し、障害発生時の対策方法を知ることができる。全国のサービスマンが常に最新の情報を共有し、技術レベルを向上できる。

ＳＴＤＢを使用し文書を検索していると、検索された文書を見て始めて他の文書も必要となり、再検索を行う場合がある。この時、新たに必要となった文書は元の検索した文書と同じキーワードを含んでおり、そのキーワードを検索条件に指定できれば容易に再検索できる。つまり、表示している文書内容に出現する登録済みキーワードを知り、そのキーワードを新たに検索条件とする機能をＳＴＤＢに付加し、更に使い易い文書検索システムとする必要がある。この機能はＳＴＤＢに関連する文書をたどりながら必要な情報を順次検索する方法を与える。

ＳＴＤＢを用いて文書データベースを構築する場合、シソーラスとシノニムをいかに簡単に定義できるかが問題である。使いやすさを向上させるためには、あらかじめ専門家の定義した内容を、使用者の意見を取り入れて改良する必要があり、その作業は決して簡単ではない。しかし、文書中のキーワードを検索条件とする機能は再検索を容易にするのみならず、シソーラスとシノニムに対しても有効に利用可能である。キーワードを分類し記述した文書を検索すればキーワードシソーラスとして利用できる。同様に、キーワードの同義語を説明した文書を検索すればキーワードシノニムとして利用できる。キーワードのシソーラスやシノニムを定義する作業を簡単にでき、しかも他の文書と利用法を統一できる。更に、検索する文書グループを指定するメニューを文書として登録し、メニュー選択をＳＴＤＢ上から行うこともでき、更に統一を図れる。

２．１．２　バイナリモデルによる拡張

本項は、検索表示した文書に含まれる他の登録済みキーワードを知る機能を従来の検索機能に追加する方法を述べる。この機能を使用すれば、文書を容易に再検索でき、検索した文書を読んでわかった他の必要な文書を簡単に調べられることが確認された。これは、ハイパーテキストシステムが提供する文書間のリンクをキーワードを使い実現できることを示している。

本機能を追加したために、以下の現象が新たに生じた。

・処理が繁雑となり、実行スピードが若干低下した。

・プログラムが扱う情報が増え、実行時に必要とするメモリ空間が増大した。

・以上の理由により、同時に使用できるユーザ数が若干減少した。

・インデックス部分の情報量が増えたため、より多くの記憶容量が必要となった。

この為、より大型のシステムにて運用する必要が生じた。又、

・新しい機能を使用するために操作が増え、複雑化した。

・表示する文書の桁数が少なく、キーワードをたどりにくい。

など、インタフェース部分を使用者毎に自由に設定できる機能が望ましいことが分かった。

具体的な開発内容を以下に示す。

(1)キーワードチェーンテキスト

ＳＴＤＢはキーワードと文書に「文書はキーワードを含む」という１個の集合的な関係を定義する。この定義に基づくインデックスを作成し高速検索を可能とする。本開発では、キーワードと文書について新たに「文書はキーワードを含む」という今までと逆の関係を定義する。

始めに、文書はキーワードが連続するもの、即ちキーワードチェーンテキストという考えを取り入れた。キーワードチェーンテキストは文中に含まれるキーワードをその出現順に並べたものである。例えば「ワープロやパソコン等オフィスでコンピュータが利用されている。」という文は、[ワープロ、パソコン、オフィス、コンピュータ、利用]というキーワードチェーンで表現される。

通常の文書	ワープロやパソコン等オフィスでコンピュータが利用されている。
キーワードチェーンテキスト	ワープロ → パソコン → オフィス → コンピュータ → 利用

図２－５　文書とキーワードチェーンテキスト

この表現法をとるとそれぞれのキーワードからその前後のキーワードを指定でき、文書中にあらわれるキーワードをたどることができる。従来、キーワードが文書中に単に存在す

ると考えていた点を拡張し、その前後のキーワードと順序関係を持つとした。この為、従来の集合的な考えに順序性を取り入れ、順序集合を基礎理論とした。

順序集合は、要素の出現順序も意味を持つ集合の表現方法である。通常の集合論では、集合Ａ＝｛１、２、３｝と集合Ｂ＝｛２、１、３｝は同一の集合である。順序集合では、順序集合Ａ'＝[１、２、３]と順序集合Ｂ'＝[２、１、３]はそれぞれ異なる集合である。(本書では通常の集合をあらわす場合に｛｝を、順序集合をあらわす場合に[]を用いる。)

順序集合は通常の集合と比較し多くの情報を持ち、より大きな記憶容量を必要とする、例えば、順序集合Ｃ'＝[１、２、１、３、２、１]は通常の集合ではＣ＝｛１、２、３｝で表現可能である。この情報量の増加を押さえることは課題の１つである。

キーワードチェーンの考えに基づきインデックスを表現すれば、キーワードから文書を検索し、さらにその文書内のキーワードを検索できる。この順序集合は、文書からキーワードへの関係を示しており、インデックスとして使用できる。従って、キーワードから文書へのインデックスと合わせ、２種類のインデックスを必要とする。

(２)バイナリモデル

キーワードと文書に新たな関係を取り入れたため、文書からキーワードへのリンクを示すインデックスが存在することとなった。キーワードから文書へのインデックスと文書からキーワードへのインデックスは同一の情報を持つ。同一の情報が別々の記述方法により複数存在すると矛盾を生じる可能性がある。また、情報を格納する容量にも無駄が生じる。つまり、インデックスを単一化する必要がある。本開発ではバイナリモデルの考えを採用し、この問題を解決した。

バイナリモデルは、実体間の関係を常に対にして与える。ネットワークモデルを用いると文書とキーワードの関係は、「文書はキーワードを含む」と「キーワードは文書に含まれる」の２つにて表現される。バイナリモデルでは「キーワードと文書は包含関係を持つ」、と表現すれば、ネットワークモデルの２つの関係をあらわすこととなる。これは２つの関係(インデックス)を統一できることを意味する。この双方向のインデックスをバイナリインデックスと呼ぶ。

バイナリインデックスと従来のインデックスそれぞれの特徴を以下に示す。

○バイナリインデックスの特徴

１つのインデックスで双方向の情報を表現可能
情報は１つにまとまっており、変更時に一貫性を保ち易い
インデックス情報は複雑かつ密度が高い
効率良く取り扱うことが比較的困難

○従来のインデックスの特徴

インデックスをそれぞれ独立して表現可能
インデックスを目的に応じて自由に設計可能

変更時にインデックスの一貫性を保ちにくい
効率良く取り扱うことが容易

(3)バイナリインデックス

インデックスを1つに統一するための、具体的な表現方法を説明する。

キーワードから文書へのインデックスは、キーワード毎に、そのキーワードが出現するファイルと行番号リストを記述し表現可能である。この表現方法は、ＳＴＤＢのインデックスとして使用されている。同様に、文書からキーワードへのインデックスは、文書毎に、キーワードが出現する行番号とそのキーワード名を記述し表現可能である。ただし、同じ行に複数のキーワードが出現する場合は、出現する順に並べる。これはキーワードチェーンテキストと同じ情報構造を取る。この2種類のインデックスを1つの形式で表現するため、キーワードと文書の組を定義し、その組毎にキーワードが文書中に存在する行番号のリストを与える。

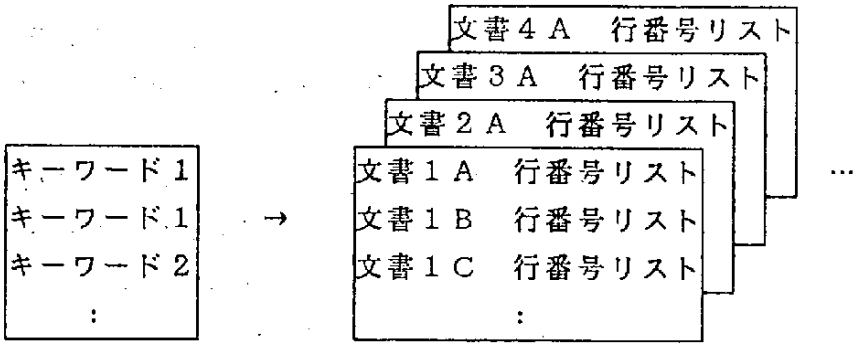


図2－6　キーワードから文書へのインデックス

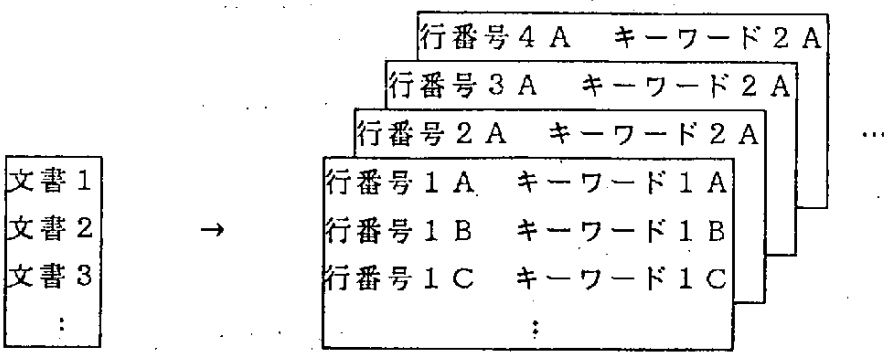


図2－7　文書からキーワードへのインデックス

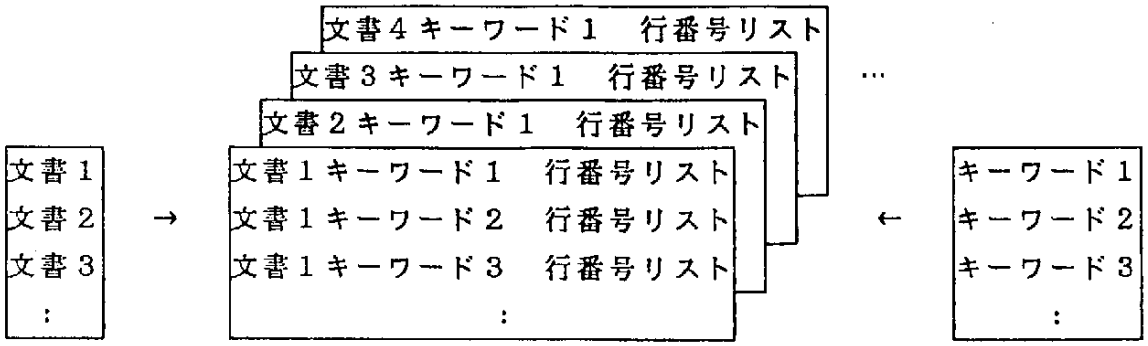


図２－８　文書とキーワードの単一化されたインデックス

このようにインデックスを表現すれば、文書からキーワードインデックスを得ることもキーワードから文書インデックスを得ることも可能であり、両者を単一化できる。

(４)実現方法

バイナリインデックス実現にあたり、文書とキーワードにそれぞれ管理番号を付ける。それぞれの管理番号を使い文書とキーワードのどちらからでも容易にアクセスできる組み合わせ管理を行う。

始めに、１文書グループに登録する文書とキーワードをそれぞれ１０万個(５桁)までとする。次に、上５桁を文書の管理番号、下５桁をキーワードの管理番号により構成させ、文書とキーワードの組を１０桁であらわす。この方法を取れば、文書とキーワードの双方からそれぞれ必要な組み合わせの管理番号を即座に知ることができる。更に、この管理番号によって示されるファイルに行番号リストを記述する。以上でキーワードから文書へのインデックスと文書からキーワードへのインデックスを単一化できる。

しかし、これだけではキーワードのファイル内に出現する順序を表現していない。実際には、キーワードと行番号の組を行番号順に再配置し、さらに、同じ行に複数のキーワードが出現する場合や、同一行に２回以上同じキーワードが出現する場合を調べ、キーワードチェーンを表現するインデックスを作成する必要がある。

検索を行う場合、インデックスは以下の手順にて処理される。

(A)キーワードから文書を検索し内容を表示
(A－１)キーワードの管理番号を調べる。
(A－２)管理番号からインデックス番号を調べる。
(A－３)インデックスからキーワードが存在する文書の管理番号と行番号リストを調べる。
(A－４)文書の管理番号から文書名を調べる。
(A－５)文書名とキーワードの出現する行番号がわかる。
(A－６)文書を読み出し、必要な部分についてキーワードを強調して表示する。

(B)その中に含まれるキーワードを選択
(B－1)文書の管理番号からインデックス番号を調べる。
(B－2)インデックスから文書に存在するキーワードの管理番号と行番号リストを調べる。
(B－3)キーワードの管理番号からキーワード名を調べる。
(B－4)行番号とキーワード名の組み合わせを作成し、行番号順に並べる。
(B－5)表示されている部分の行について存在するキーワードを得る。
(B－6)実際の文書にて出現順を確認し、キーワードチェーンを作成する。
(B－7)キーワードチェーンをたどり、順次キーワードを強調表示していく。
(B－8)指定されたキーワードを条件に選択する。

(C)再検索
(C－1)再度キーワードから文書を検索する。(A－1)に戻る。

以上の手順に基づきプログラムを作成した。この追加機能は期待通りの結果をもたらした。しかし、いくつかの問題点も生じた。バイナリモデルの利用によりインデックスを統一化した為、別々にインデックスを用意した場合に比べ、それぞれの処理は複雑となる。又、インデックスを表現する情報量も2つのインデックスの合計よりは少ないが、キーワードから文書への一方向のインデックスと比較すると大きくなる。この為、検索処理スピードは低下し、使用するメモリやディスク等の計算機資源も多くなる。これらの問題はシステム構成を大きくするだけでも対応がとれる。しかし、根本的な解決にはならず、将来的には他の対応を必要とする。又、従来は機能が限定されていたために、操作も単純にでき、個人的な好みを反映する環境ではなかった。しかし、新しい機能を追加したために操作が複雑化し、インタフェースにも個人の好みが表れてくる。この点も今後考察すべき問題となった。これらの問題に対応する方法について次節以降で考察を行う。

２．２　システムの実行環境

２．２．１　基本システムの特徴

本開発で使用したピラミッドコンピュータはサーバプロセスを効率良く実行するための仕組みを標準のＵＮＩＸ上に加えており、アプリケーションを変更することなくその機能を使用できる。追加機能の多くはメインフレームコンピュータにおいても使われている方法でありデータベースの処理に欠かせない機能といえる。各機能について説明する。

(１)オペレーティングシステム

ピラミッドコンピュータはＵＮＩＸをオペレーティングシステムとする。現在ＵＮＩＸは、バークレイ版４ＢＳＤとＡＴ＆Ｔ版ＳＹＳＴＥＭ　Ｖの２つが標準である。ピラミッドコンピュータは独自のデュアルユニバース機能により、両者を統合し完全にサポートしている。このため一方の環境にて作成されたプログラムを他方の環境にて使用可能である。つまり、標準ＵＮＩＸ上の様々なアプリケーションを実行可能である。又、両方の環境を相互に利用してプログラムを開発可能である。ＳＴＤＢの大部分はＡＴ＆Ｔのライブラリを使い実現されている。ただし、全体表示機能についてはバークレイのライブラリが効率良く出来ておりこちらを使用している。

(２)シンメトリックマルチプロセッサ

ＳＴＤＢがそうであるようにデータベースアプリケーションのほとんどはＣ言語を用いて開発されている。ピラミッドコンピュータは、データベースアプリケーションの命令コードの実行に最適な数のレジスタとキャッシュを持ちＣ言語を効率良く実行できる独自のＲＩＳＣプロセッサを利用する。プロセッサは最大１２個まで増設でき、それぞれのプロセッサはシンメトリックに動作する。つまり、プロセッサはジョブの実行を終えると、待ち行列から次に実行されるジョブを選択し実行する。すべてのプロセッサは公平にジョブを実行していく。このため、プロセッサの増減がシステム全体の処理能力に比例する。つまり、プロセッサを追加するだけ処理能力を向上させることが出来る。又、プロセッサが故障した場合、故障したプロセッサをシステムから取り外し再スタートすれば、システムが自動的に取り外されたプロセッサを無視するよう設定を変更する。使用者は何等手を加えずに処理を継続できる。これによりシステムの稼働率を高めることができる。

(３)キャッシュ

シンメトリックマルチプロセッサでは、実行を中断されたジョブが、次にどのプロセッサ上で実行を再開されるか決まっていない。ジョブの実行を中断したプロセッサと異なるプロセッサ上でジョブが再開されると、先に実行したプロセッサが読み込んだキャッシュの内容が無駄になるだけでなく、新しく別のプロセッサが同じキャッシュの内容を読み込む必要があり無駄である。

ＳＴＤＢはＷＹＳＩＷＹＧ(What You See Is What You Get)を基本としている。このため、実行時間のほとんどは使用者が表示内容を読み理解する時間であり、プロセッ

サの負荷は低い。しかし、文書検索時はキーワード概念登録内容を展開し文書を探し出すために非常に負荷が高くなる。ピラミッドのデータキャシュとプログラムキャシュは２５６キロバイトあり、この展開処理を高速に処理可能である。

(４)実行優先度

通常のＵＮＩＸは、資源を平等に分配するために、長い時間実行されてきたプロセスについては、実行する時間の割合を徐々に短くする。このため、１つのプロセス内で同じ処理を２回続けて行う場合でも、２回目の方が１回目より遅くなってしまう。大規模データベースのアプリケーションはデータが増加するごとに、データ量の増加以上に処理時間がかかることとなり、予定した時間内に処理できないこともある。ピラミッドコンピュータはプロセスの実行優先度を固定する機能を持ち、特定のプロセスを常に優先して実行可能である。ＳＴＤＢは検索した文書をファイルに記述する機能を持つ。印刷処理を行うプロセスを最優先し実行すれば、使用者は最優先に印刷物を入手できる。

(５)プロセッサの占有

ピラミッドコンピュータは、特定のプロセスの実行を特定のプロセッサに割り当てる機能を持つ。この機能を使用すると、プロセスを割り当てられたプロセッサは、そのプロセス自身とその子プロセスを実行する。複数のプロセスを１つのプロセッサに割り当てることもできる。実行優先度の機能と組み合わせると、あるプロセスを特定のプロセッサ上で常に最優先に実行できる。この機能をうまく利用すれば、プロセッサを特定の使用者専用に設定可能である。印刷処理を最優先にすれば、他の使用者にとってはサービスが悪くなる。プロセッサを占有し使用すれば、他の使用者への応答性をそこなうことがない。

(６)バーチャルディスク

データベースアプリケーションが大量のデータを取り扱うとき、ディスクシステムがボトルネックとなる。このためディスクシステムについての追加機能は実行効率に大きく影響する。ピラミッドコンピュータは知的ディスクコントローラを使用し、ディスクシステムに仮想ディスク機能を付加した。さらにディスクコントローラが故障した場合は、故障したコントローラに接続されているディスクドライブを他の正常なディスクコントローラからアクセスするよう変更でき、データがアクセス不可能になることを防ぐ。

仮想ディスクのそれぞれについて説明する。

(Ａ)ストライプディスク

単一のディスクシステムに複数の処理要求がある場合、ディスクコントローラを最適に制御し、なるべく無駄のない処理を行わせることが必要である。ピラミッドコンピュータ、の知的ディクスコントローラはある位置から読み出す処理と、ある位置に書き出す処理を同時に行う場合、どちらから実行する方が速く行えるかをあらかじめ調べてから行っている。この様に同時に複数の処理を行う時、ディスクシステムが物理的にいくつかのディス

ク装置にわかれているならば、同時に幾つかの処理を行うことが可能である。上の例でいえば、読み出しする位置と書き出す位置が別のディスク装置にあれば同時に処理可能である。このように、みかけは１つのファイルシステムを複数のディスクシステムに分散する方法をストライプディスクと呼ぶ。この方法でディスクの効率を最大３００％向上できる。ストライプディスクはデータベースのインデックスファイルのような使用頻度の高いファイルを格納するのに適している。

ＳＴＤＢにおいては、使用回数の多い文書はストライプディスク化が有効である。例えば、仕様書説明書等、比較的ページ数が多い文書データベースにおいて有効性を確認した。このような場合は、多数のユーザが同時に１つのファイルへアクセスする必要がある。ストライプディスクでアクセスを分散できる。

(Ｂ)コンキャティネイトディスク

巨大なデータベースでは、ディスク装置より大きいサイズのファイルを作成する必要が生じる場合がある。コンキャティネイトディスクは複数のファイルシステムを連結し１つのファイルシステムとして連結し使う方法である。これにより理論的にファイルサイズは、最大ディスク容量と等しくできる。

ＳＴＤＢでは本機能を使用せず、ストライプディスク機能を使用している。本機能は、独自のファイル管理機構を持つ大規模データベースのアプリケーションには最適な機能であろう。

(Ｃ)ミラーディスク

データベースの一番注意すべき点はディスクの内容を安全に保つ方法である。この為、データベースシステムにはログやロールバックなど、様々な工夫が施されている。しかし、これらの機能はディスク装置の破壊には対応できない。ピラミッドコンピュータは、ミラーディスクの機能を使い、ファイルシステム間で同じ内容を持つことを可能とした。複数のディスクシステムに同時に同じ更新を行え、故障によるデータの消失を避けることができる。ディスクシステム全体にてミラー化を実現するため、ディスクドライブの故障のみならず、ディスクコントローラの故障にも対応できる。

１つのディスク装置やディスクコントローラが故障しファイルシステムが使用できなくても他のディスクシステムを使い処理を継続できる。故障が治れば、同じ内容になるように自動的に復旧できる。両方が稼働している場合には、それぞれに処理を分散し効率を高めることができる。

ＳＴＤＢにおいては、使用頻度が高くかつデータバックアップが必須なデータベースへの利用が有効である。稟議書データベースにて利用し有効性を確認した。ディスクへのアクセスを分散できるほか、バックアップも自動で行える。万一、ディスクシステムが故障しても、利用ができなくなる事態を避けられる。

(Ｄ)メモリー(ＲＡＭ)ディスク

ファイルシステムをさらに高速化するための方法として、ディクス装置の代わりにメモリーを使用する方法がある。これをメモリー(ＲＡＭ)ディスクと呼ぶ。メモリーへのＩＯスピードで処理でき、特に高速処理が必要な場合は有効である。しかし、メモリーディスクは信頼性が極端に低くなり、データベースへは適していない。そこで、メモリーディスクとミラーディスクを併用する方法がある。これにより、メモリーのスピードでデータを処理し、かつデータはディスクシステムに自動的に書き込まれる。

メモリーディスクは容量が比較的小さいため、最も効果が高い一部のファイルのみを扱わせる必要がある。ＳＴＤＢにおいては、使用頻度が極端に多いファイル例えば検索頻度が多いキーワードについてのインデックスファイルをメモリーディスク上に置くことが考えられる。将来実験システムを試作し効果を確認したい。

２．２．２　システムチューンアップ手法の考察

本項は、文書検索プログラムを高効率化する方法を説明する。リレーショナルデータベース管理システムに代表されるデータベース管理システムを調べ、ＳＴＤＢへの利用を考察した。次に、前項で説明したピラミッドコンピュータの特徴機能がいかにＳＴＤＢやリレーショナルデータベース管理システムの実行効率の向上につながるかを考察した。

考察の結果、以下のことがわかった。

・検索システムはバックエンドサーバとフロントエンドの２つから構成すると効率良く実行できる。
・データ管理を受け持つバックエンドとプログラミング言語にインタフェースがあると、使用者は操作インタフェースを自由に定義できる。
・クライアントサーバアーキテクチャのプログラムはピラミッドコンピュータ上で効率良く実行できる。
・ピラミッドコンピュータを使えばアプリケーションを変更せずに、システムの利用度を高められる。

詳細を以下に説明する。

(１)キーワード管理システムの開発

データベースシステムの処理を分けてみると大きく２つにわけることが出来る。一方はユーザへの画面表示やユーザからの操作を処理する部分であり、他方は実際のデータにアクセスしユーザの問い合わせ結果の生成を処理する部分である。これらの部分をそれぞれ別のプロセスにて実行するシステムがある。それぞれのプロセスをフロントエンドプロセス、バックエンドプロセスと呼ぶ。

使用者がデータベースへのアクセスを開始するとき、１つのフロントエンドプロセスが生成され、それに対応するバックエンドプロセスが生成される。ユーザの入力はフロントエンドプロセスにて検索式と変換されバックエンドプロセスに送られる。バックエンドプロセスは検索式を解釈し、必要なデータを集め、加工し、結果をフロントエンドプロセスに戻す。フロントエンドプロセスはその結果を加工(成型)し使用者に表示する。この方式は、以下の特徴を持つ。

・検索インタフェースを変更する場合、フロントエンドプロセスを変更するだけですむ。
・プロセス間の通信手段が確立されていれば、それぞれのプロセスを別のコンピュータで実行できる。
・それぞれが同時並列実行可能であるから、処理時間を短くできる。

フロントエンドとバックエンドにプロセスを分割するアーキテクチャは多くのデータベース管理システムに採用されている。本開発では、米国バークレイ大学にて開発されたパ

ブリックドメイン版のリレーショナルデータベース管理システム(以下ＲＤＢＭＳと略す)を使い、キーワード情報を管理させるシステムを試作し、その評価を行った。プログラムの詳細を「＜付録＞ingresに基づくキーワード管理システム概要書」に示す。

以下の手順で開発を行った。始めに、キーワードに関する情報(検索日付、検索回数、シノニム、シソーラス)をリレーショナルテーブルに表現した。次にこれらの情報を記録するファイルからその内容を読み出し、ＲＤＢＭＳの管理するデータベースに登録するプログラムを開発した。これにはＲＤＢＭＳに付属のデータコンバートツールを使用した。次にＲＤＢＭＳから必要な情報を取り出すためのライブラリ関数を作成した。これにはＲＤＢＭＳに付属のＣ言語インタフェースを使用した。最後に、ＳＴＤＢにこのライブラリ関数を組み込んだ。これにより、従来のプログラムはフロントエンドプロセスとして使用者とのインタフェースを取り扱い、ＲＤＢＭＳはバックエンドプロセスとしてキーワード情報を管理し、フロントエンドの問い合わせに対し必要な情報を返す構造となった。

表２－１　キーワード管理方法の比較

	従来プログラム	試作プログラム
文書管理方法	ＵＮＩＸのディレクトリ機能	ＵＮＩＸのディレクトリ機能
キーワード管理方法	独自管理構造	リレーショナルテーブル
キーワード管理ファイル	可変長ファイル	固定長ファイル
キーワード検索方法	独自アルゴリズム	ハッシングテーブル

本試作プログラムを実行し、以下のことが確認された。

・キーワード管理部分と画面表示部分と分離しても実行スピードはさほど低下しない。ただし、本システムはＵＮＩＸのパイプ機能を使いデータの受け渡しを行っている。
・ソートや集計など等の特殊な処理を必要とする問い合わせはＲＤＢＭＳのパフォーマンスに依存し、一般に遅い。

このことから、フロントエンドプロセスとバックエンドプロセスプロセスに分けて処理を行うアーキテクチャは高効率を実現でき、独自処理を行うバックエンドプロセスを作成し利用することが有効であると考えられる。

しかし、フロントエンドとバックエンドにて構成されるアーキテクチャのプログラムに以下の問題があることがわかった。

・１ユーザにつき２つのプロセスが生成されるため、ユーザ数の増加にともないプロセス数が増加し、システムの負荷が高くなりやすい。
・多数のバックエンドプロセスが１つのデータを共有するため、データ更新時のロック等、非同期処理を効率化する必要がある。
・データベースの問い合わせ処理がＯＳに依存してしまい、処理の最適化が出来ない。

これらの問題は、主に複数のバックエンドプロセスが独立し実行されることが原因である。そこで、複数のフロントエンドプロセスからの問い合わせに１つのバックエンドプロセスで対応する方式が考えられており、これをクライントサーバアーキテクチャと呼ぶ。クライアントサーバアーキテクチャについて次に説明する。

(2)クライアントサーバアーキテクチャ

使用者が新たにアクセスを開始するとき、フロントエンドプロセスのみを生成し、バックエンドプロセスを生成しない構造をクライアントサーバアーキテクチャと呼ぶ。フロントエンドプロセスをクライアントプロセス、バックエンドプロセスをサーバプロセスと呼ぶ。サーバプロセスは各クライアントからの要求をスケジューリングし処理する。それぞれ異なるデータを管理する複数個のサーバプロセスを独立して存在させることができる。クライアントプロセスも異なる複数のサーバにアクセス可能である。

この方式は、以下の特徴を持つ。

・使用者が１人増える毎に１つのプロセスが生成されるのみであり、使用者数の増加にともなう負荷の増加が最小限ですむ。
・１つのサーバプロセスを通じてのみデータにアクセス可能であるため、データの変更について一貫性を保ちやすい。
・サーバプロセスはマルチタスクオペレーティングシステムとみなせる。このため、クライアントプロセスと比較すると非常に大きなプロセスとなる。
・データベース問い合わせ処理の最適化が可能である。

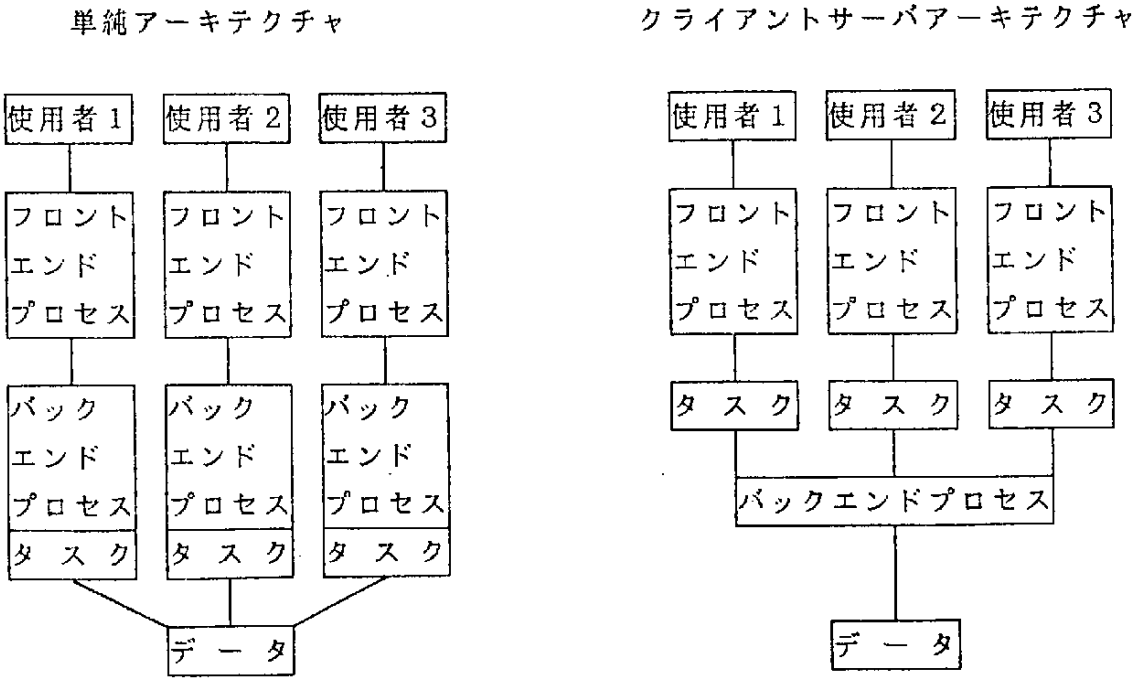


図２－９　データベースプロセスのアーキテクチャ

ＳＴＤＢをサーバアーキテクチャに設計変更することは、効率化を図るに最も有効な手段の１つである。又、それにより画面表示を受け持つインタフェース部分の開発も容易になる。

(３)高利用度システム

データベースシステムに２４時間フル稼働の要求があり、稼働率の高い高利用度システム(フォールトトレラントシステム)が望まれている。フォールトトレラントシステムの構成について様々な提案があり、製品化されている。具体的には機構を２重化３重化あるいは分散化させ、システムが一部故障しても全体の動作に影響しないような構成をしている。

本開発にて使用したピラミッドコンピュータは、以下の機能を備えている。

・ＣＰＵ処理の分散
・ミラーディスクによるディスクシステムの２重化
・ディスクコントローラの２重化

これらの機能により、アプリケーションに変更を加えることなく、システムをフォールトトレラント化できることがピラミッドコンピュータの特徴となっている。

これとは別に、データベースの問い合わせを出来るだけ高速に処理したいとの要求がある。サーバアーキテクチャはこのための１つの解決法である。サーバアーキテクチャアプリケーションの利用度をさらに高めるため、ピラミッドコンピュータが提供する機能を以下に述べる。

・サーバプロセスは長時間実行される為、途中で処理が中断されることもある。ピラミッドコンピュータでは途中で実行が中断されたプロセスは同じプロセッサ上で再実行されるため、キャッシュの内容が有効に利用される。
・ＵＮＩＸは資源を平等に分配する。このため、長時間実行されているプロセスは優先度が下がり、単位時間あたりに実行される時間が次第に短くなっていく。ピラミッドコンピュータは優先度を固定する機能を持ち、常に最優先に実行できる。
・ピラミッドコンピュータは特定のプロセスの実行を特定のプロセッサに割り当てられる。プロセスが割り当てられたプロセッサはそのプロセスを常時最優先に実行可能である。

これら機能によりピラミッドコンピュータはサーバプロセスを最良の状態で実行できるシステムであると言えよう。

３．今後の展開

(１)ハイパーテキストシステムとの比較

本開発プログラムはハイパーテキストシステムと似た機能を持つ。両者を比較し、ＳＴＤＢを更に強化する方法を検討する。

ハイパーテキストシステムはカード状に表現されたテキスト(文書、画像)についてあらかじめプログラミングされた他のテキストへのリンクを選択し、必要な情報をたどっていくシステムである。特徴を以下に示す。

・画面の位置を指示するだけなので操作インタフェースが簡単
・リンク付け言語を使いプログラミングするため自由度が高い
・音声や動画などマルチメディアにも対応
・プログラミングに手間が掛かる

本開発システムではこの点は、以下のようになっている。
・ハイパーテキストに比較すると操作が複雑
・キーワードが他のテキストへのリンク情報となる
・ＵＮＩＸインフェース機能を使い他のメディアへもリンク付け可能
・プログラミングは基本的に不必要

システムをより良い検索システムとするには、上記の弱点を補強する必要がある。具体的に以下の強化を考えている。
・使用者が操作インタフェースを設計可能とする。
・他メディアへのリンクの有無を分かりやすくする

(２)グループウエアへの対応

ＯＡシステムは個人の業務を向上させるために使用されている。コンピュータがネットワークシステムにて接続されるようになり、データの共有が可能となっている。これにより、グループでの業務に使用できるコンピュータシステム(グループウエア)についての要望がある。文書検索システムはグループでの文書共有化を進めるために役立ち、グループウエアの要素技術の一つと考えられる。今後、以下の既存技術と組み合わせ、グループでの協調作業を支援する。

・電子メール
・リレーショナルデータベースシステム

これら機能と組み合わせるために以下の開発課題がある。

・クライアントサーバアーキテクチャの採用
・システム稼働率の向上
・他アプリケーションとのインタフェース

(3)クライアントサーバアーキテクチャへの対応

システムの実行効率を向上させ、インタフェース部分に自由度を与えようとする場合、プログラムをサーバとクライアントにそれぞれ機能分割することが有効であることを説明した。開発プログラムを改良し、文書検索サーバとしての機能を実現することを計画している。以下の開発課題がある。

・サーバが提供すべき機能を使用するためのインタフェース
・クライアントを開発するためのライブラリ
・複数のクライアントからのタスク管理
・競合解消と処理最適化

<参考文献>

[1]Carlo Batini ed., Entity-Relationship Approach, North-Holland, 1989
[2]C.Beeri,J.W.Schmidt,U.Dayal, Proceedings of the 3rd International Conference on Data and Knowledge Bases, Morgan Kaufmann, 1988
[3]F．W．ランカスタ，情報システムのためのシソーラスの構築と利用，情報科学技術協会，1989
[4]J.Biskup, J.Demetrovics, J.Paredaens, B.thalheim, MFDBS '87 1st Symposium on Mathematical Fundamentals of Database Systems, Springer-Verlag, 1987
[5]J.Kalash,L.Rodgin,Z.Fong,J.Anton, Ingres Version 8 Reference Manual, University of california Berkeley, 1984
[6]Jeffrey D.Ullman, Database and Knowledge-Base System Volume1, Computer Science Press, 1988
[7]K.Haviland,B.Salama, UNIX System Programming, Addison Wesley, 1987
[8]M.J.Rochkind、福崎俊博訳，ＵＮＩＸシステムコールプログラミング Advanced UNIX programming，アスキー出版，1987
[9]M.Stonebraker ed., Readings in Database Systems, Morgan Kaufmann, 1988
[10]S.Jajodia,W.Kim,A.Silberschatz eds., Proceedings International Symposium on Databases in Parallel and Distributed Systems, Computer Society, 1988
[11]Salvatore T.March ed., Entity-Relationship Approach, North-Holland, 1988
[12]Stefano Spaccapietra ed., Entity-Relationship Approach, North-Holland, 1987
[13]T.Hasegawa,H.Takagi,Y.Takahashi eds, Performance of Distributed and Parallel Systems, IFIP, 1989
[14]Tosiyasu L.Kunii, Visual Database Systems, North-Holland, 1989
[15]Ulka Rodgers, UNIX Database Management Systems, Yourdon Press, 1990
[16]W.Cellary,E. and Gelenbe,T.Morzy, Concurrency Control in Database Systems, North-Hollannd, 1988
[17]会津　泉，グループウエアが米国で関心の関心の的に、グループの共同作業を支援，日経バイト April 1989 P281～P285，日経ＢＰ，1989
[18]池田克夫，オペレーティング・システム，日本コンピュータ協会，1976
[19]石井　裕，グループウエア技術の研究動向，情報処理 Vol.30 No.12 P1502～P1508，情報処理学会，1989
[20]石井　裕，グループワークのコンピュータ支援に関する研究動向，Human Interface News and Report Vol.4 No.1 P113～P117，計測自動制御学会 ヒューマンインタフェース部会，1989
[21]伊藤哲郎，情報検索，昭晃堂，1986
[22]緒方良彦，インデックス その作り方・使い方 データベース社会のキー・テクノロジー，産業能率大学出版部，1986

[23]尾関周二、クリスティアン・ガリンスキー，ターミノロジー学，文理閣，1987
[24]高山正也，技術情報管理のコンピュータ化とその実例集，応用技術出版，1985
[25]ダニー・グッドマン，THE COMPLETE HYPERCARD HANDBOOK VOL．1，株式会社ビー・エヌ・エス，1988
[26]ダニー・グッドマン，THE COMPLETE HYPERCARD HANDBOOK VOL．2，株式会社ビー・エヌ・エス，1988
[27]日本医療情報学会，医療コンサルテーションシステム国際シンポジウム 医療知識データベース時代の幕明け－－講演を中心にして－－，日本医療情報学会，1989
[28]マーク・フリッセ、山崎俊一，ハイパーテキスト，日経バイト February 1989 P231～P244，日経ＢＰ，1989
[29]前川守，オペレーティングシステム，岩波書店，1988
[30]平川顕名、岡田好一，ＭＵＭＰＳによる医療情報処理入門，朝倉書店，1988

＜付録＞

ingresに基づくキーワード管理システム概要書

１．概要

リレーショナルデータベース(以下ＲＤＢ)を利用しＳＴＤＢが使用するキーワード情報を管理させ、ＲＤＢをバックエンド、ＳＴＤＢをフロントエンドとし実行するシステムを、開発する。ＲＤＢは大量のデータが格納可能、データ量にかかわらず検索効率が良い、データの保証性が高い等のメリットがあり、ＳＴＤＢに生かす。

本開発ではＲＤＢとして米国カリフォルニア大学バークレー校にて開発されたパブリックドメイン版ingresを使用する。ingresはＣ言語インタフェースを用意しており、Ｃプログラム中に独自のデータベース検索言語を記述できる。

２．プログラムの関係構造

ユーザ、プログラム、データの関係構造を示す。

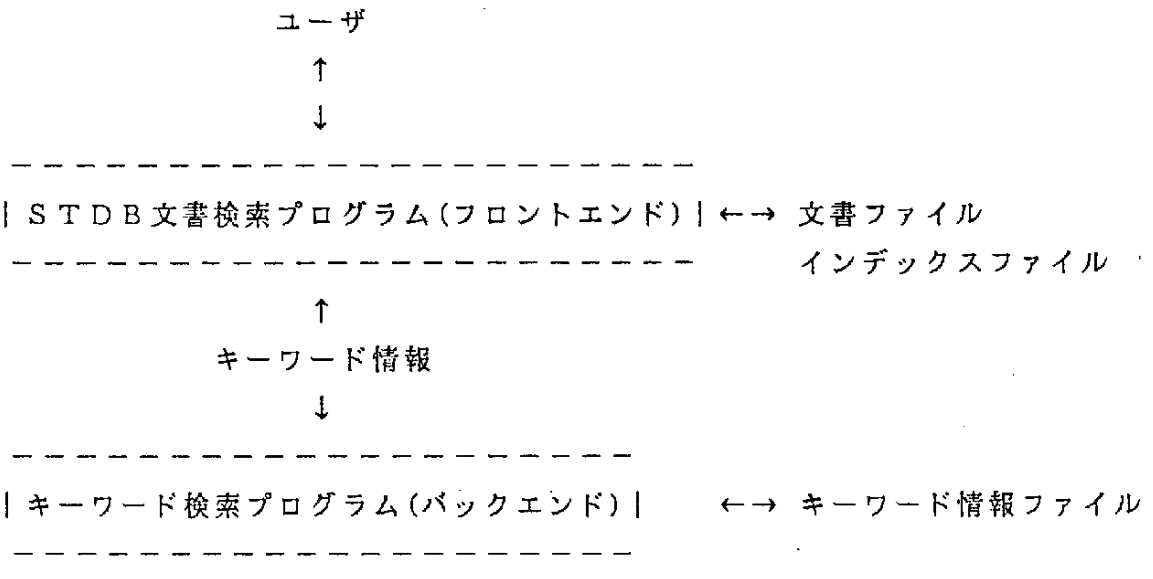


図１　ユーザとプログラムとデータの関係構造

３．ファイル構成

ファイルはそれぞれＳＴＤＢディレクトリ、データディレクトリ、ingresディレクトリにわかれて格納される。以下、各ディレクトリ毎にファイル構成を示す。

(1)ＳＴＤＢディレクトリ

ＳＴＤＢ関連のプログラムはＳＴＤＢディレクトリに格納される。以下に分類を示す。

```
+- bin/ ---+- dbi        (文書検索プログラム)
|         +- makekey    (*.log → *.stdb変換プログラム)
|         +- umakekey   (*.stdb → *.log 変換プログラム)
|         +- keysort    (ソートキーワード作成プログラム)
|
+- lib/ ---+- help       (ヘルプファイル)
          +- list       (全体標示プログラム)
          +- cmd0～cmd9 (ＵＮＩＸインタフェースプログラム)
```

(2)データディレクトリ

文書ファイルとそれに付属するキーワードやインデックスファイルはデータディレクトリに格納される。以下に分類を示す。

```
+ 文書/  ------ *          (文書ファイル)
+ 文書.id/  --- *          (インデックスファイル)
|
+ 文書.log/  -+- date       (日付の新しい順にソートしたキーワードファイル)
|             +- count      (使用回数の多い順にソートしたキーワードファイル)
|             +- relay      (関係の多い順にソートしたキーワードファイル)
|             +- program    (概念の多い順にソートしたキーワードファイル)
|                           (以上をまとめてログファイルと呼ぶ)
+ 文書.stdb/ -+- m_index    (文書インデックスファイル)
              +- m_keytbl   (キーワード／日付／使用頻度ファイル)
              +- m_rlytbl   (関係キーワードファイル)
              +- m_protbl   (概念キーワードファイル)
              +- s_date     (日付の新しい順にソートしたキーワードファイル)
              +- s_count    (使用回数の多い順にソートしたキーワードファイル)
              +- s_relay    (関係の多い順にソートしたキーワードファイル)
              +- s_program  (概念の多い順にソートしたキーワードファイル)
              |             (以上をまとめてキーワード情報ファイルと呼ぶ)
              +- admin      (ingres用データ管理ファイル)
              +- attribute  (ingres用データ管理ファイル)
              +- indexes    (ingres用データ管理ファイル)
              +- integrities(ingres用データ管理ファイル)
              +- protect    (ingres用データ管理ファイル)
              +- redelim    (ingres用データ管理ファイル)
              +- relation   (ingres用データ管理ファイル)
              +- tree       (ingres用データ管理ファイル)
```

(3)ingresディレクトリ

ingresディレクトリはingresプログラムとデータファイルを格納する。以下に分類を示す。

```
+- bin/ ---+- creatdb    (データベースディレクトリ作成プログラム)
|         +- sysmod     (データベースディレクトリ補修プログラム)
|         +- printr     (データベースファイル表示プログラム)
|         +- helpr      (データベースファイルステータス表示プログラム)
|         +- usersetup  (ユーザー登録プログラム)
|         +- equel      (EQL言語 → C言語 変換プログラム)
|         +- univingres (データ処理プログラム)
|
+- lib/ ------ libq.a    (ingresライブラリ)
|
+- files/ ---+- dbtmplt8  (creatdb用データファイル)
             +- equel8    (equel用データファイル)
             +- error8    (エラーメッセージファイル)
             +- proctab8  (プロセステーブル)
             +- procx8    (プロセステーブル)
             +- users     (ユーザ登録ファイル)
```

4．リレーショナルデータベース用キーワード情報ファイルの構造

リレーショナルデータベース用キーワード情報ファイルについてファイル(テーブル)毎に、ファイル(テーブル)名、フィールド名、各フィールドのタイプ、サイズ、内容、サンプルデータを示す。

(1) m_keytbl

すべてのキーワードについてキー番号、キーワード名、最終検索日、検索使用回数を示す。

フィールド名	タイプ(サイズ)	内容
no	整数(4バイト)	キー番号
keyword	文字(40字)	キーワード名
kw_date	文字(6字)	キーワード最終検索日
ke_count	整数(4バイト)	キーワード検索使用回数

no(i4)	keyword(c40)	kw__date(c6)	kw__count(i4)
1	ＩＢＭ	890918	10
2	ＡＴ＆Ｔ	890917	32
3	シャープ	890918	1
:	:	:	:

(2) m__rlytbl

関係登録されたキーワードについて、キー番号、キーワード名とその下位関係キーワード名を示す。

フィールド名	タイプ(サイズ)	内容
no	整数(４バイト)	キー番号
keyword	文字(４０字)	キーワード名
kw__child	文字(４０字)	下位関係キーワード名

no(i4)	keyword(c40)	kw__child(c40)
1	米国企業	ＩＢＭ
2	米国企業	ＡＴ＆Ｔ
3	米国企業	ＤＥＣ
4	米国企業	ＳＵＮ
5	日本企業	シャープ
6	日本企業	富士通
:	:	:

(3) m__protbl

概念登録されたキーワードについて、キー番号、キーワード名とプログラムオペレータとキーワード名を示す。

フィールド名	タイプ(サイズ)	内容
no	整数(４バイト)	キー番号
keyword	文字(４０字)	キーワード名
kw__op	文字(１字)	プログラムオペレータ
kw__child	文字(４０字)	プログラムキーワード名

no(i4)	keyword(c40)	kw_op(c1)	kw_program(c40)
1	ＡＴ＆Ｔ		ＡＴ＆Ｔ
2	ＡＴ＆Ｔ		ＡＴＴ
3	ＵＮＩＸ		ＡＴ＆Ｔ
4	ＵＮＩＸ	＆	ＢＳＤ
:	:	:	:

（４）s_date

すべてのキーワードについてキーワード名とキーワード最終検索日をキーワード最終検索日の遅い(近い)順に示す。

フィールド名	タイプ(サイズ)	内容
keyword	文字(４０字)	キーワード名
kw_date	文字(６字)	キーワード最終検索日

keyword(c40)	kw_date(c6)
ＩＢＭ	890918
シャープ	890918
ＡＴ＆Ｔ	890917
:	:

（５）s_count

すべてのキーワードについてキーワード名とキーワード検索利用件数をキーワード検索利用件数が多い順に示す。

フィールド名	タイプ(サイズ)	内容
keyword	文字(４０字)	キーワード名
kw_count	文字(４バイト)	キーワード検索使用回数

keyword(c40)	kw_count(i4)
ＡＴ＆Ｔ	32
ＩＢＭ	10
:	:
シャープ	1

(6) s_relay

関係登録されたキーワードについて、キーワード名と関係登録キーワード数を関係登録キーワードが多い順に示す。

フィールド名	タイプ(サイズ)	内容
keyword	文字(40字)	キーワード名
kw_rlycnt	数字(4バイト)	関係登録キーワードの数

keyword(c40)	kw_rlycnt(i4)
米国企業	4
日本企業	2
:	:

(7) s_program

概念登録されたキーワードについて、キーワード名とプログラムされたキーワード数をプログラムされたキーワードが多い順に示す。

フィールド名	タイプ(サイズ)	内容
keyword	文字(40字)	キーワード名
kw_procnt	数字(4バイト)	プログラムされたキーワード数

keyword(c40)	kw_procnt(i4)
AT&T	2
UNIX	2
:	:

5．検索関数(サブルーチン)

データベースバックエンドとのインタフェースである検索サブルーチンを説明する。

(1)検索関数の使用法

・データの受渡しはメモリバッファを使用する。

・機能と引数は以下を参照のこと。

(2)サブルーチン名と機能

以下に機能と引数を列挙する。

ing setup(char * dir)
ingresの呼び出しと初期化

ing dspdat(int fr, int to, char * ireg)
s__dateファイルからの日付データの取り出し

ing setdat(char * ireg, char * date)
s__date、m__dateファイルへの日付データの書き込み

ing deldat(char * ireg)
s__date、m__dateファイルからの日付データの削除

ing dspcnt(int fr, int to, char * ireg)
s__countファイルからの頻度データの取り出し

ing setcnt(char * ireg, int ct)
s__count、m__countファイルへの頻度データの書き込み

ing delcnt(char * ireg)
s__count、m__countファイルからの頻度データの削除

ing dsprly(int fr, int to, char * ireg)
s__relayファイルからの関係キーワードデータの取り出し

ing getchi(char * ireg)
m__rlytblファイルからの下位関係キーワードの取り出し

ing getpar(char * ireg)
m__rlytblファイルからの上位関係キーワードの取り出し

ing setrly(char * ireg)
m__rlytblファイルへの関係キーワードデータの書き込み

ing delrly(char * ireg)
m__rlytbl、s__relayファイルからの関係キーワードデータの削除

ing dsppro(int fr, int to, char *ireg)
s__programファイルからのプログラムキーワードデータの取り出し

ing getpro(char *ireg)
m__protblファイルからのキーワードプログラムの取り出し

ing setpro(char *ireg)
m__protblファイルへのキーワードプログラムの書き込み

ing delpro(char *ireg)
m__protbl、s__programファイルからのキーワードプログラムの削除

ing getsts(char *ireg)
s__date、s__count、s__relay、s__programファイルからのデータの取り出し

6．変換プログラム

ＳＴＤＢの持つキーワードファイルの情報をingres用のデータベース形式に変換するプログラムを説明する。

(1)makekey

第一引数にて指定したディレクトリについてログファイルからデータベース用ファイルを作成する。

変換元のデータファイル		変換後のデータファイル
.log/date、.log/count	→	*.stdb/m__keytbl
*.log/relay	→	*.stdb/m__rlytbl
*.log/program	→	*.stdb/m__protbl

(ただし、*は第一引数にて指定したディレクトリ名)

(2)umakekey

第一引数にて指定したディレクトリについてデータベース用ファイルからログファイルを作成する。

変換元のデータファイル		変換後のデータファイル
*.stdb/m__keytbl	→	*.log/date、*.log/count
*.stdb/m__rlytbl	→	*.log/relay
*.stdb/m__protbl	→	*.log/program

(ただし、＊は第一引数にて指定したディレクトリ名)

(3)keysort

第一引数にて指定したディレクトリについてデータベース用ファイルを作成する。

変換元のデータファイル		変換後のデータファイル
＊.stdb/m__keytbl	→	＊.stdb/s__date、＊.stdb/s__count
＊.stdb/m__rlytbl	→	＊.stdb/s__relay
＊.stdb/m__protbl	→	＊.stdb/s__program

(ただし、＊は第一引数にて指定したディレクトリ名)

平成２年　３月発行

発行　　財団法人　データベース振興センター
東京都港区浜松町二丁目４番１号
世界貿易センタービル７階
ＴＥＬ　０３－４５９－８５８１

委託先　　シャープ株式会社
大阪市阿倍野区長池町２２番２２号
ＴＥＬ　０６－６２１－１２２１

印刷所　　大蔵印刷工業株式会社
東京都千代田区神田淡路町１丁目９番１号