統計数理 第33巻 第1号（1985）

PROFISS

——FORTRAN ユーザーのための プログラム開発・ファイル管理支援システム——

統計数理研究所　石　黒　真木夫

（1985年5月　受付）

1. はじめに

PROFISS は FORTRAN ユーザーのためのプログラム作製・ファイル管理システム（PROgramming and FIling Support System）である．現在 HITAC-VOS3 システムの上で稼動している．

以下の特徴を備えている．

a） 強力なファイル管理機能

一人の研究者が作製あるいは利用する（実質的にすべての）プログラムおよびデータを識別して管理することが出来る．ここでいう管理には

i） ディスク上のファイルとして存在するデータセットを磁気テープに記録すること．

ii） ディスク上のデータセットを消去すること．

iii） 磁気テープの記録を読み取ってデータセットをディスク上のファイルとして回復すること．

iv） ディスクあるいは磁気テープ上のデータセットのインデックスを用意して検索を助けること．

v） 磁気テープの物理的な保全を図るための定期的な複製作製をソフト的に支援すること．

が含まれる．

b） DEBUGER, EDITOR などのプログラム開発ツールの使用が容易に行なえる．

c） 拡張も容易である．プログラム開発ツールを新しく開発すれば，ほとんどただちにこのシステムに組み込むことができる．

d） 移植しやすく設計されている．

i） 5節で説明するように PROFISS は FORTRAN プログラムとコマンドプロシージャ文で構成されている．コマンドプロシージャ文は意味が明確に定義された高級言語であり，そのまま移植するのは無理にしても翻訳する形での移植に原理的困難は無いと考えられる．FORTRAN 部分が移植できるのは当然といって良いだろう．HITAC-VOS3 システムはもちろんのこと TSS が稼働しているある程度以上の大型機であれば個人ユーザーのレベルで導入することも可能であろう．

ii） ユーザーが既に財産として持っているファイルがカードイメージであってディスク上に在る場合には簡単な登録手続きによって PROFISS の管理下に組み込むことができる．

もとより，いかなるOSのもとであろうとも，上記a）のi）ii）iii）は可能にちがいない．ある程度以上のOSになればDEBUGERも備えていよう．

しかし，これらの機能が使えるということと，使いやすいということは必ずしも等価ではない．たとえば，あるデータとそれを処理するプログラムを磁気テープに記録して，後日それを再びディスク上のファイルとして回復することは，やさしいことではあるが，同時に，何本かあるテープのどれに記録したのかを忘れてしまうことも同様にやさしい．略号のようなファイル名をつけておいたためにどのファイルに何が入っているのか忘れてしまうことも多い．

このような形でデータやプログラムをなくしてしまうことを避ける唯一の手段は，ファイルを作るたびに名前と内容をメモし，消去した時にはメモにその旨を記入し，磁気テープに転送した時には，またメモし，といった具合にまめにメモを残すことである．

完全なメモを残すということは必要にはちがいないが，これを紙と鉛筆で実現しようとするのは非現実的である．まめにノートをとることによって，完全な記録が残せたとする．記録が完全であるとすると，それはかなり大量になるはずであって，この大量の記録の中からある時に必要な部分をさがし出すというのは難事業である．そもそも紙と鉛筆に頼って完全な記録を残すことがまず不可能であるといってよい．ある仕事をしようとしている時の心理状態というのは，じっくりと記録を残すという作業になじまない．したがって，ほとんど論理的な帰結として，次の結論が得られる．

「データの転送にあたって，その記録が自動的に残るシステムが必要である」

この自動記録システムに要求される事項をかぞえあげれば次のようになろう．

1. 各データセットごとに，データセットを識別するコード，その所在（ディスク上のファイルとして保存されているのか，磁気テープに記録されているのか，等），内容の説明，キーワード等が記録されていること．
2. データ，それを処理するプログラム，そのプログラムをコントロールするパラメータなどがばらばらにならないようにひとまとめに管理できること．
3. 記録を読んで必要なデータセットをさがし出す検索機能．
4. あるデータセットに関してメモを書き込み，あるいは書き直しをする記録保守機能．

PROFISSは以上の要求を満たしている他に，上記第1項と2項が満足されていることを利用して，FORTRAN原始プログラムなどのデータセットとプログラム開発ツールを容易に結びつけることができるように作られている．上記第3項の検索機能と4項の記録保守機能はもともと名簿管理，文献リスト管理等を目的として開発されたものであってかなり強力なものである．検索機能を利用してスケジュール表を読んで起動された日の予定を表示する機能も備えている．

2. ファイル

PROFISSにおけるデータ管理の単位をPROFISSファイル，PROFISSファイルのリストをPROFISSインデックスと名づける．

PROFISSの管理下で新たに作成されるファイルはすべてPROFISSファイルでありPROFISSインデックスに登録される．

ディスク上の既存のデータセットがカードイメージであればPROFISSインデックスに登録することによってPROFISSの管理下におかれる．

以下簡単のために特に必要ないかぎりPROFISSファイル，PROFISSインデックスの'PROFISS'を省略して単にファイル，インデックスということにする．

2.1. PROFISSファイル

すべてのファイルに

ｄｄｄｄ．ｉｉｉｉ

の形をした「ファイル名」が与えられる．ｄｄｄｄをユーザー指定名，ｉｉｉｉを識別子という．識別子はPAIR, FORT, CNTL, DATAのいずれかにするのが普通であるが，この他にも親システムの識別子として許されるものならなんでもよい．

ファイルはディスク上又は磁気テープ上いずれにおくこともできる．ファイルｄｄｄｄ．ｉｉｉｉがディスクにあるときの実体は表1の通りである．

表1．ディスク上のPROFISSファイル　ｄｄｄｄ．ｉｉｉｉ

識別子　ｉｉｉｉ	親システムにおける実体
PAIR	ｄｄｄｄ．CNTL　と　ｄｄｄｄ．FORTの対
その他	ｄｄｄｄ．ｉｉｉｉ

識別子PAIRをもつファイルは，親システムにおけるデータセットの対で構成されるのである．これらがつぎの項で説明するPAIRファイルの規格にあっていれば，さまざまなツールが簡単に使用できる．

PROFISSファイルを収めた磁気テープをPROFISSテープという．一本のPROFISSテープに複数のファイルが記録されるが，親システムの側から見たPROFISSテープはラベルなしで単一データセットが書きこまれた単純な構造である．

ディスク上のファイルのコピーを，ディスク上あるいはPROFISSテープ上のファイルとして作製すること，PROFISSテープ上のファイルのコピーをディスク上のファイルとして作製することが自由に出来る．

2.2. PAIRファイル

PAIRファイル（たとえばABCD. FORTとABCD. CNTLの対）が次の規格に従って作られているとき(正規)PAIRファイルという．対になっているものの規格から外れている場合には非正規PAIRファイルという．

PAIRファイル規格

i) ABCD. FORTの内容が固定フォーマットのFORTRAN原始プログラムである．
ii) プログラム中でファイル機番90～99を使っていない（PROFISSのDEBUGルーチンが予約している）．
iii) ABCD. CNTLの内容がABCD. FORTに収められているプログラムをコンパイル，リンク，そして実行するバッチ・ジョブストリーム制御データであってそのままサブミットできる形になっている．
iv) ABCD. CNTLによって起動されるバッチ・ジョブは，以下の例に示すように，入出力ファイル・作業用一時ファイルともにディスク上のファイルであって，機番5の入力装置においたデータでコントロールされる形になっている．

```
//USERIDXB  JOB  ********
//     EXEC FLG
//FORT.SYSIN   DD  DSN=ABCD.FORT, DISP=SHR
//GO.FT10F001  DD  DSN=AAAA.DATA, DISP=SHR
//GO.FT20F002  DD  DSN=AAAA.DATA, DISP=MOD
//GO.FT20F001  DD  DSN=BBBB.DATA, DISP=OLD
//GO.FT30F001  DD  UNIT=DISK, +
    SPACE=(TRK, (10, 10)), DISP=NEW 〔一行におさめる〕
//GO.FT40F001  DD  DUMMY
//GO.SYSIN   DD  *
 〔機番5に割り当てたデータ〕
//
```

このような形式でPAIRファイルが用意されていると，DEBUGにあたって必要となる入出力ファイルをPROFISSがALLOCATEしてくれる．2行目のFLGはプロッターを使うジョブの場合のFLGXYKPであってもよい．

2.3. PROFISSインデックス

(PROFISS) インデックスは項目の集まりであり，原則として各項目が一つのPROFISSファイルまたはPROFISSテープを代表する．またキーボード入力やPROFISSテープ以外の磁気テープからの入力によって新しいファイルを作ることが，仮想的なデータセットからのデータ転送として処理できるようにするための仮想的データセットを代表する項目も用意されている．

各項目はシステムによって書き込まれる4行を含む4行以上の行で構成される．原則として，

1行目にはデータの所在
2行目には識別コード
3行目には名前ｄｄｄｄ
4行目には識別子ｉｉｉｉ

が記入される．ユーザーは5行目以降にシステム使用分を含めて合計2000文字を越えない範囲で任意の情報を書き込むことができる．

以下でインデックスの項目の形式を説明するが，説明中の「・・・」は記号列・・・が必ず使われる場所を示す．

2.3.1. PROFISSファイル

a) ディスク上のファイル

インデックスの項目の形式は

「DISK」
「NOID」
ｄｄｄｄ
ｉｉｉｉ

である．たとえば，

DISK

NOID
XYZ
DATA

で代表されるPROFISSファイルは，親システムにおいてXYZ. DATAというデータセット名を持つファイルとして存在する．

b) PROFISSテープ上のファイル

インデックスの項目は

mmmm
n n n n
d d d d
i i i i

の形をとる．ここで，

mmmm＝PROFISSテープ名（イニシャライズ時に与える）．
n n n n＝識別コード（テープ上に転送された時点の年，月，日，分，秒の値で作られる）

である．PROFISSテープ上のファイルは識別コードn n n nによって識別されるので同じ名前d d d d．i i i iのファイルがいくつあっても差し支えない．もちろん同名のファイル2つが同時にディスク上に存在することは許されない．

PROFISSテープからディスクにデータを転送する際には名前がチェックされ，同じ名前のファイルが既にある場合には別の名前を付けることが要求される．PROFISSが要求してくるのである．

c) 仮想的データセット

インデックスに

「WORLD」　「BRAIN」
「NOID」　「NOID」
「INTERESTING」と「FANTASTIC」
「DATA」　「DATA」

という二つの項目が用意されている．INTERESTING. DATA, FANTASTIC. DATAという2つのファイルを代表しているわけである．INTERESTING. DATAはPROFISSテープ以外の磁気テープからデータを読み込むための，FANTASTIC. DATAはユーザの頭の中にあるデータをとりこむための仮想的PROFISSファイルである．詳細はデータ転送コマンドの項にゆずる．

2.3.2. PROFISSテープ

PROFISSインデックスの項目は

「MYTAPE」
j j j j
mmmm
「TAPE」

の形でとる．ここで，

jjjj＝10桁の数字から成る磁気テープの識別コード
mmmm＝PROFISSテープ名

である．jjjjの正体はそのテープに記録されているレコードの数である．十進表現の上位桁に0を補って10桁の数字にしてある．

2.3.3. インデックス仕様のファイル

PROFISSインデックス自身が一つのPROFISSファイルとしてインデックス（すなわち自分自身）に登録されている．インデックスを代表する項目は

「DISK」
「NOID」
「INDEX」
「DATA」

である．

PROFISSインデックスと同じ仕様で作られたファイルをインデックス仕様のファイルといい，PROFISSの記録・検索機能が使える．名簿や文献リストをインデックス仕様のファイルとして作っておくと便利である．

PROFISSにはインデックス仕様のファイルとして作成されたスケジュール表が用意されていて記録検索機能を利用したスケジュール管理ができる．スケジュール表を代表する項目は，

「DISK」
「NOID」
「SCHDL」
「DATA」

である．

3. 作動モード

PROFISSの作動モードには，主としてPROFISS自体の点検・管理のためのコマンドが用意されているTSSモード，検索機能を主とするREADモード，および記録保守のためのWRITEモードがある．DEBUGERあるいはEDITORが起動された状態もPROFISSのモードと考えてそれぞれDEBUGモード，EDITモードと名づけておく．これらの間の相互の関係を図示すれば図1のようになる．

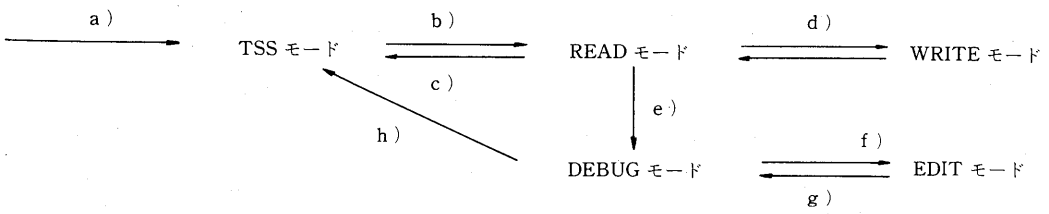

図1．PROFISSの作動モード

a）　会話処理の開始（LOGONコマンド）によってTSSモードのコマンド待ちとなる．

（注） TSS のコマンド待ちモードであるが PROFISS のモードの一つと見なすことができる．この立場をとると，TSS コマンドすなわち PROFISS のコマンドと言うことになる．

b) TSS モードコマンド

BRING　INDEX

の発行によって READ モードが起動される．

（注）「INDEX」は BRING コマンドのオペランドであり，検索の対象として PROFISS インデックスを指定している．

- 「INDEX」の代わりにインデックス仕様のファイルのファイル名を指定してもよい．
- BRING のオペランドで指定されたファイルを BROUGHT インデックスという．
- その日初めての起動あるいはファイル名を変更しての起動を '新規起動' といい，新規起動の場合にはまずスケジュール表が検索されてその日の予定が表示される．それ以外の '継続起動' ではスケジュールの表示はない．
- WHAT　CAN　I　DO　FOR　YOU?
  のメッセージが READ モードのコマンド待ち状態に入ったことを示す．

c) READ モードコマンド

¥BYE

を送信することによって TSS モードに戻る．

（注）
- READ モードと WRITE モードのコマンドには原則として頭に¥がつく．
- 端末によっては¥のないものがある．READ モードのコマンド待ち状態で空行を送信するとヘルプ機能が働いてコマンド表が出力される．表を見れば¥のかわりに使用すべき記号がわかる．
- READ モードではヘルプ機能が使える。
  ¥　HELP（ｘｘｘ）
  を投入することによってｘｘｘに関する情報が得られる．
- たとえば¥HELP（¥HELP）で¥HELP コマンドの使いかたが分かる．

d) WRITE モードを起動するには，READ モードコマンド

¥MODIFY　または　¥ADD

を発行する．READ モードに戻る方法については WRITE モードコマンドの項を参照されたい．

e) READ モードコマンド

¥DEBUG

の発行によって DEBUGER が起動される．

f) ソースプログラムを修正するために，コマンド２つ

END
SHOW

を発行することによって EDITOR が起動される．

（注） ENDコマンドでDEBUGERから抜け，SHOWコマンドでEDITORが起動されるのである．DEBUGERの中でコマンドプロシージャが使えないために2段階の操作が必要になっている．

g) EDITモードコマンド

GO

を発行するとDEBUGモードが起動される．

h) VOS3のDEBUGERのサブコマンド

END

を発行することによってTSSモードにもどる．

4. コマンド

表2に示すように，TSSモード，READモード，WRITEモードおよびEDITモードのコマンドに分類される．

以下で各コマンドの機能と形式，および注意事項，を説明するが，説明のなかでつぎの記号をもちいる．

ｘｘ 等連続する英小文字： 記号列をあらわす変数
ｘｘ｜ｙｙ ：ｘｘ または ｙｙ の選択
〔ｘｘ〕 ： ｘｘ｜空白

4.1. TSSモードのコマンド

a) PHASE

機能：PROFISSの状態表示およびイニシャライズ

形式：PHASE 〔INITIALIZE〕

説明：・オペランド INITIALIZE を付けなければ単なる状態表示．
・INITIALIZE を付けると状態表示，確認手続きをへてイニシャライズ．
・表示内容は
i) LOCK. DATA ：システム保護の状態
ii) STATUS. DATA ：最近のBROUGHTインデックス
iii) FAILSAFE. DATA：BROUGHTインデックスのバックアップ
iv) DEBUG. FORT ：デバック中のプログラム
v) IDCARD. DATA ：PROFISSテープの状態
である．LOCK. DATAが「CLOSED」の場合にはFAILSAFE. DATAとCLOSEされた時点のBROUGHTインデックスの異同を調べて適切な処置を取ってからイニシャライズする．

b) BRING

機能：READモードの起動およびBROUGHTインデックスの指定

形式：BRING ｘｘ

説明：PROFISSインデックス を検索したければ ｘｘ＝INDEX
スケジュール表 を検索したければ ｘｘ＝SCHDL
同じBROUGHTインデックスをもう一度 検索したければ ｘｘ＝C

表2. PROFISS コマンド

TSS モード		PHASE	PROFISS の状態表示、イニシャライズ
		BRING	READ モードの起動
		NEWTAPE	PROFISS テープイニシャライズ
		BACKUP	PROFISS テープバックアップ
		UPDATE	スケジュール表生成
		PACKGE	プログラムパッケージ作成
		INTRO	既存ファイルの登録
		SHOW	DEBUG 時の EDITOR 起動
R E A D モード	制御コマンド	¥HELP	ヘルプ機能
		¥BYE	TSS モードに復帰
		¥MODIFY	WRITE モードの起動、記録の修正
		¥ADD	WRITE モードの起動、記録の追加
		¥BRING	BROUGHT ファイル交換
		¥PON, ¥POFF	ラインプリンタ制御
	検索コマンド	(xx)	記号列 xx を含む項の抽出
		¥REPEAT	同じ検索コマンドの繰りかえし
		¥SAVE	抽出された項目の保存
	計算コマンド	¥TOTAL	パラメータの値の合計
	ファイル操作コマンド	¥EDIT	EDITOR の起動
		¥SHOW	ファイル内容の表示
		¥SIZE	ファイルサイズのチェック
		¥ERASE	ファイル消去
		¥CNAME	ファイル名のつけかえ
		¥TOOL	ユーザー自作ツールの起動
	PAIR ファイルコマンド	¥LIBRARY	ユーザーサブルーチンの取込み
		¥DEBUG	DEBUGER の起動
		¥RUN	デモンストレーション開始
		¥SUBMIT	ジョブのサブミット
	データ転送コマンド	¥PRINT	ファイルの印刷
		¥COPY	ディスク内データ転送
		¥STORE	PROFISS テープへの転送
		¥GET	ディスクへの転送
		¥SYSBACKUP	PROFISS テープのコピー
WRIT モード		¥HELP	ヘルプ機能
		xx	記録 xx の追加
		¥CH (xx)	記録 xx の修正
		NULL, ¥OK	READ モードに復帰、書きこみ終了
		¥MK, ¥MKGB	READ モードに復帰、マーク記入
EDIT モード		ENC	原テキスト保存
		GO	DEBUGER 起動

注) 表中のアンダーラインはコマンドの省略形を示す。

¥SAVE で保存したスタックの中で　　検索したければ　　x x =SAVED
新しいインデックス仕様のファイルを作りたければ　　x x =NEW

c) NEWTAPE

機能：PROFISS テープのイニシャライズ

形式：　NEWTAPE

説明：PROFISS の指示に従ってテープの名前を入力すると

・新しいPROFISSテープが作られ，そのテープにPROFISSインデックスが転送される．
・この転送によって作られたファイルもインデックスに登録されるのはもちろんのことである．

d) BACKUP
機能：PROFISSテープのコピーおよびバックアップされたファイルの登録
形式： BACKUP
説明：以下の順に処理が進む．
1. PROFISSテープから書き込み許可用のリングをはずしてあることの確認．
2. バックアップ用の磁気テープが用意されていることの確認．
3. PROFISSテープであると思った磁気テープに記録されているレコード数がシステムに記録されている値と違う時は処理を打ち切る．
4. レコード数に間違いがなければ，全体をバックアップテープにコピーする．
5. PROFISSテープに初めてBACKUPされたファイルが含まれていたときはこれらのデータセットを代表する項目をインデックスに追加する．

e) UPDATE
機能：スケジュール表の延長．
形式： UPDATE
説明：SCHDL. DATAのスケジュール表を百日分延長する．

f) PACKAGE
機能：プログラムパッケージ作成．
形式： PACKAGE
説明：このコマンドを発行してからPROFISSの指示に従って行動すれば磁気テープとその説明書からなるプログラムパッケージが完成する．
磁気テープのフォーマット：DENSITY＝1600 BPI (PE), 9TRACKS
CODE ＝EBCDIC
LABEL ＝STANDARD

g) INTRO
機能：既存のファイルをインデックスに登録する．
形式： INTRO ｄｄｄｄ ｉｉｉｉ
説明：ｄｄｄｄ＝ファイルのユーザ指定名
ｉｉｉｉ＝識別子（FORT, CNTL, DATAまたはPAIR等）
注意：識別子FORTとCNTLで区別されるがユーザ指定名を共通にする2つのファイルの対がPAIRファイル規格を満たしているときはｉｉｉｉ＝「PAIR」とする．

h) SHOW
機能：デバッグ中にソースプログラムを修正するためのEDITOR起動．
形式： SHOW 〔ｘｘ〕〔ｙｙ〕
説明：a) ｘｘ ｙｙが行番号の場合
ｘｘからｙｙ行までを表示してEDITORのサブコマンド待ちとなる．
b) ｘｘが英字を含む記号列の場合
ｘｘをFIND後その前後を表示してサブコマンド待ち
c) ｘｘもｙｙもない場合
ソースプログラムの先頭を表示してサブコマンド待ち

注意：DEBUGER から END サブコマンドで抜けてから投入すること．

4.2. READ モードコマンド

制御コマンド，検索コマンド，計算コマンド，
ファイル操作コマンド，PAIR ファイルコマンド，データ転送コマンド
の 6 種類に分けられる．制御コマンド，検索コマンドおよび計算コマンドは常に使える．ファイル操作コマンド以降は，唯一の例外である ¥EDIT をのぞいて BROUGHT インデックス が PROFISS インデックスでないと使えない．ファイル操作コマンド，PAIR ファイルコマンド，データ転送コマンドをまとめてファイルコマンドということがある．

4.2.1. 制御コマンド

a) ¥HELP

機能：ユーザーにコマンドの機能，形式，使用上の注意などを与える．

形式： ¥HELP（ｘｘ）|〔¥HELP〕

説明：a）¥HELP（ｘｘ）

- （ｘｘ） のかわりに ｘｘ を前後から同じ記号，たとえば空白，で挟んだ記号列が使える．
- MANUAL 項目を検索して ｘｘ を含む項を抽出・表示する．
- MANUAL 項目とは記号列 ¥PLEH を含む項目のことである．
- PROFISS インデックス および「BRING NEW」コマンドでつくったインデックス仕様のファイルには MANUAL 項目が組込まれている．
- ¥HELP（ｘｘ） の投入は検索コマンド
  （ｘｘ） ¥＊（¥PLEH） ¥RES
  の発行と等価である．

b) ¥HELP |「READ モードのコマンドとして無意味な記号列，たとえば空行」

- 状態を表示する．表示内容は次の通り．
  BROUGHT インデックス
  ラインプリンタ接続の有無
  コマンド表
  スタックの状態（スタックに関しては検索コマンドの項を参照のこと）
  最後に発行した検索コマンド

b) ¥BYE

機能：TSS モードへの復帰

形式： ¥BYE

c) ¥MODIFY, ¥ADD

機能：WRITE モードの起動

形式： ¥MODIFY | ¥ADD

説明：
- ¥MODIFY は ¥MOD でよい．
- ¥MODIFY はスタックに積まれた項目を修正するときに使う．
- ¥ADD は項目を追加するときに使う．

d) ¥BRING

機能：BROUGHT インデックスの交換

形式： ¥BRING ｘｘ

説明：・TSSモードにおける「BRING ｘｘ」の発行に等価.
・ただし，このコマンドによるBROUGHTインデックスの交換の場合は，「BRING」による場合と異なってREADモードの'新規起動'とは見なされず，スケジュールの表示が無い.

e) ¥PON, ¥POFF

機能：ラインプリンタの制御

形式： ¥PON | ¥POFF

説明：・¥PON によってラインプリンタが接続される．その意味に関しては検索コマンドの説明を参照のこと.
・¥PON を発行してから ¥DEBUG を発行するとデバッガのラインプリンタ出力が得られる.
・¥POFF の発行によってラインプリンタは切り離される.
・一般に ファイルコマンドを発行するとラインプリンタは切り離される.

4.2.2. 検索コマンド

機能：BROUGHTインデックスを検索して，所定の条件を満たす項目を順次画面に表示するとともにスタックに積み上げる．検索コマンドを繰り返すとスタックに積まれた項目の中で検索が行われてスタックが次第に小さくなる．この過程はスタックが空になるまで繰り返すことも出来るし，スタックを解消して，検索をはじめからやり直すことも出来る.

形式：（ｘｘ）〔¥＊（ｙｙ）| ¥−（ｙｙ）| ¥+（ｙｙ）〕
〔¥RESTART | ¥RES〕
〔¥SLOWLY | ¥SLO〕
〔¥GE（１１）〕〔¥LE（ｕｕ）〕〔¥ORD（ａａ）〕〔¥DIC〕

説明：・検索コマンドの基本的な形は （ｘｘ） である.
・これを送信することによって ｘｘ という文字列を含む項目のすべてが抽出，表示されてスタックに積まれる.
・抽出された項目が15以上あると画面への表示は途中で打ち切られて AND SO ON. が表示されるが，スタックには全項目が積まれる.
・ラインプリンタが接続されていると抽出された項目はスタックに積まれるだけでなくラインプリンタ印刷用のバッファにも送られる.
・ラインプリンタへの出力にあたって ¥GE～¥DIC オプションの項で説明する「項目の値」の順に項目が並べかえられる.
・抽出の途中で印刷用バッファがいっぱいになると，そこまでの分を印刷してバッファを空にするので全体を通しての並べかえが出来ないことがある.

注意：検索は原則として既につまれたスタックの中で行われる.
¥RESTART オプションを参照のこと.

オプション ：基本的な形に以下のオプションをつけ加えることが出来る.
オプションはどういう順に並べてもかまわない.

¥＊（ｙｙ） ：文字列ｘｘおよびｙｙを含む項が抽出される.

¥−（ｙｙ） ：文字列ｘｘを含むがｙｙは含まない項が抽出される.

¥+（ｙｙ） ：文字列ｘｘまたはｙｙを含む項が抽出される.

¥RESTART　：スタックを解消して，検索をはじめからやりなおす．このオプションを付加しないとスタックの中での検索が行われる．

¥SLOWLY　：指定した条件を満たす項目が一つ発見されるごとに一時停止して検索を続けるか否かの指示を求める。検索の打切りを指示するとコマンド待ちになる。スタックに残るのは，打切った時点までに抽出された項目である．

¥GE（１１）：項目の最初の英数字５文字を英数字からなる５桁の36進数として読み，この値（項目の値と名づけておく）が同じ36進数として読んだ１１の値以上の項目のみを表示する．英数字の間の順序は　0<1<…<9<…<A<…<Z　である．これ以外の記号や空白は無視される（例外として，最初の５桁の中にとじかっこ')'が含まれているとそれまでの桁で項目の値が評価される．）

¥LE（ｕｕ）：項目の値がｕｕ以下の項目のみを抽出する．

¥ORD（ａａ）：項目の値を決める英数字の列を項目の最初の文字列ではなく，項目中に含まれる最初のａａに続く文字列とする．

¥DIC　：項目の値を評価するにあたって，英数字の列が５桁に足りない場合は，下の桁に０をおぎなってかならず５桁の数として評価する．このオプションによって

A<AB<B

という辞書式の順序づけができる．

b)　¥REPEAT

機能：検索コマンドの繰りかえし

形式：　¥REPEAT | ¥REP 〔¥SLOWLY | ¥SLO〕〔¥RESTART | ¥RES〕

説明：最後に発行した検索コマンド〔にオプションを追加したもの〕を発行する．

c)　¥SAVE

機能：スタックの内容を保存する．

形式：　¥SAVE

説明：これの利用法に関しては　〔¥〕BRING コマンド　を参照のこと．

4.2.3.　計算コマンド

スタックに積まれた項目に記入されているパラメータの値に関する計算をする．計算コマンドはただ一つである．

a)　¥TOTAL

機能：スタックに積まれた項目に記入されているパラメータｘｘの値の和を求める．

形式：　¥TOTAL〔(ｘｘ)〕

説明：　i)　¥TOTAL（ｘｘ）　の場合

正または負の整数　n　（十進法で表現された８桁以下の数）が

ｘｘ（n）

の形で項目中に記入されていると，その和が得られる．パラメータｘｘを含まない項目は

ｘｘ（0）

があるものとして処理される．一つの項目に　ｘｘ（mm）　が二つ以上あると前の値が計算される．

ii)　¥TOTAL　だけの場合

スタックに積まれた項目を数える.

4.2.4. ファイル操作コマンド

PROFISS ファイルの操作に関するコマンド群である. ¥EDIT に例外的な使いかたがあるが, 原則としてスタック最後の項目によって代表されるファイルを操作の対象とする. このファイルを BROUGHT ファイルという. BROUGHT ファイルは(オペランドなしの)¥HELP コマンドまたは空行の送信で表示される.

ファイル操作コマンドは原則としてオペランドをとらないので, 以下の説明では特別な場合を除いて「形式」の項を省いた. 表題の形そのものがコマンドとなる.

a) ¥EDIT

機能: EDITOR (VOS3 の EDIT) の起動

説明: EDIT の対象が場合によってことなる.

- BROUGHT インデックスが PROFISS インデックスであれば, BROUGHT ファイルが EDIT の対象になる.
- それ以外の場合は BROUGHT インデックスが EDIT の対象になる.

b) ¥SHOW

機能: BROUGHT ファイルの先頭部分が画面に表示される.

説明: PAIR ファイルのばあいは ｄｄｄｄ.FORT の方が選ばれる.

c) ¥SIZE

機能: ディスク上の BROUGHT ファイルの大きさを調べる.

説明: PROFISS テープに転送する場合に送られるレコードの数である. テープにまだ余裕があるかどうか調べるのに使える.

d) ¥ERASE

機能: BROUGHT ファイルおよびそれを代表するインデックス項目を消去する.

説明:
- BROUGHT ファイルがディスク上にあるときのみ実行される.
- 消去は次の段階を踏んで行われる. 途中の段階で処理の進行を止めることができる.
  - i) 消去の手続きをとることの再確認
  - ii) インデックス項目の消去
  - iii) ファイルの内容の表示 および 消去することの再再確認
  - iv) ファイルの消去

e) ¥CNAME

機能: ディスク上のファイルの名前の変更

説明: 殆んど不用

f) ¥TOOL

機能: ユーザーが開発して PROFISS に付け加えたツール ｘｘ の起動

形式: ¥TOOL (ｘｘ)

説明: このコマンドを発行するとコマンドプロシージャ BRING に制御が渡される. 「記号パラメタ」の値が以下のように設定されているのでこれを手がかりに ｘｘ を実行するルーチンを付け加えておけばよい.

&GOSTOP=NEWTOOLxx
&MTNAME=ファイルの所在

&FILE＝ファイル名
&KIND＝識別子

補足：目下この形で追加されているルーチンは以下の2つである．

f′）　¥TOOL (FORT)

機能：BROUGHT ファイルをコンパイルする．

説明：BROUGHT ファイルが PAIR ファイルのときにのみ有効なコマンドである．

・コンパイル・リストはソース・リストの最後にコメントの形式に変換されて付け加えられるので，コンパイル結果を見ながらソース・リストの編集ができる．

・SAF によるクロスリファレンスの図がコンパイル・リストの後に付加される．

f″）　¥TOOL (FILE)

機能：ジョブの実行に必要なファイルがディスク上に在るか否か調べる．

説明：・BROUGHT ファイルが PAIR ファイルのときにのみ有効．

・ジョブ制御文に引用されているファイルがディスク上に在るか否か調べる．

4.2.5. PAIR ファイルコマンド

BROUGHT ファイルが PAIR ファイルであるときにのみ有効なコマンド群である．この節も「形式」の説明がいらない．

a）　¥LIBRARY

機能：私的サブルーチンパッケージからのサブルーチン取り込み．

説明：・インデックスの項目に

```
LIBRARY
ｓｓ
ｔｔ
END
```

を記入してからこのコマンドを発行すると，パッケージから2つのサブルーチン　ｓｓ　と　ｔｔ　が BROUGHT ファイルに付け加えられる．

・サブルーチンパッケージは MACRO7. FORT という名のファイルに作っておく必要がある．

b）　¥DEBUG | ¥RUN

機能：DEBUGER の起動

説明：　i）　¥DEBUG　の場合

・インデックスの項目に

```
DEBUG
ｓｓ
ｔｔ
END
```

が記入してあれば，BROUGHT ファイルにおかれた FORTRAN 原始プログラムのメインプログラムと2つのサブルーチン　ｓｓ，ｔｔが TEST オプションで，その他のルーチンが NOTEST オプションでコンパイルされて

DEBUGER に引きわたされる.

- 機番5の入力装置から読むよう指定されたデータはdddd.CNTLファイルから読み込まれる.
- DEBUGERのマニュアルは参考文献[3]の「VOS3最適化FORTRAN端末使用の手引き」にある.

ii) ¥RUN の場合

- TESTオプションでコンパイルされるのはメインプログラムだけである.
- 機番5の入力装置がキーボードに割り当てられる.

注意: デバッグ終了後は TSSモード になる. 4.1節のSHOWコマンドの説明を参照のこと.

c) ¥SUBMIT

機能: PAIRファイルで定義されたジョブをサブミットする.

4.2.6. データ転送コマンド

データ転送コマンドとして ¥PRINT, ¥COPY, ¥STORE, ¥GET および ¥SYSBACKUP が用意されている. これらはデータの所在と送り先によって表3のように使い分けられる.

表3. データ転送コマンド

		転送先		
		ディスク	PROFISSテープ	ラインプリンタ
データの所在	ディスク上のPROFISSファイル	¥COPY	¥STORE	¥PRINT
	その他	¥GET	¥SYSBACKUP	

データの所在の「その他」にはPROFISSテープ, それ以外の磁気テープ および ユーザーの頭脳が含まれる.

a) ¥PRINT

機能: BROUGHTファイルの内容がラインプリンタに出力される. ファイルの内容がFORTRAN原始プログラムであれば, サブルーチンごとに改ページされ, 最後にサブルーチンの表が印刷される.

b) ¥COPY

機能: ディスク上のBROUGHTファイルをコピーし, それを代表する新しいインデックス項目をつくる.

説明:

- このコマンドを投入すると新しいファイルにつけるユーザー指定名と識別子を問い合わせて来るので適当に入力する.
- あたえたユーザ指定名が現にディスク上にあるいずれかのPROFISSファイルと一致する場合には, 別の名前が要求される.
- PAIRファイルのコピーの場合には, FORT, CNTLを識別子として持つ2つのデータセットそれぞれがコピーされるが, 単純なコピーでなく次の処理が行われる.

i) FORTRAN原始プログラムの中に ENC コマンドを用いて残して

あった古いテキストを消去する.

ii) ジョブストリーム中の FORTRAN 原始プログラムが存在するファイルの名前を新しいファイルの名に書き替える.

iii) コピーとオリジナル両者の先頭にコピーを作った時点の年, 月, 日, 時, 分, 秒および両者のファイル名が記録される.

c) ¥STORE

機能：ディスクから PROFISS テープへのデータ転送

説明：このコマンドを投入すると, 次の段階を追って処理される.

1. インデックスを検索して　MARKEDITEM　と書き込まれた項目をさがす. なければ READ モードにもどる.
2. 記号列　MARKEDITEM　を消去する.
3. この項目によって代表されるファイルの先頭部分に, この時点の年, 月, 日, 時, 分, 秒からなる識別コードを挿入する(PAIR ファイルの場合には FORTRAN 原始プログラムの先頭のみ).
4. データセットに含まれるレコード数を数えて, PROFISS テープに既に記録済のレコード数との和が限界に達するか否かを調べて, レコード数が多ければ処理を中断して READ モードに戻る.
5. テープに余裕がある場合は, テープを磁気テープ装置にマウントすることを要求する.
6. テープにデータを書き込み, PROFISS テープのレコード数の記録を書き換える.
7. READ モードに戻る.

注意：・MARKEDITEM　は　WRITE モードの　¥MK (または¥MKGB) コマンドであらかじめ書き込んでおく (¥MKGB で書いたマークから　MARKEDITEM が消えると GARBAGE　が残る. ¥STORE　してから　¥ERASE　するときに便利である).

・¥STORE を実行しただけでは PROFISS テープに上のファイルを代表するインデックス項目は作られない. TSS モードの BACKUP コマンドを発行してバックアップテープを作ったときにはじめてインデックスに登録される.

・インデックスに新しく作られる項目には

PROFISS テープの名前
¥STORE　ルーチンで与えられた識別コード
オリジナルのユーザー指定名
識別子
データセットの最初の数行

が書き込まれる.

・まとめて　¥STORE　しておいて最後に BACKUP してよい.

d) ¥GET

機能：ディスクへのデータ転送

説明：BROUGHT ファイルによってデータを取りに行く場所がちがう.

i) BROUGHT ファイルが PROFISS テープ上のファイルの場合

・PROFISS テープを読んで新しいファイルを作るジョブがサブミットされる.

・ジョブが終了するとファイル内容のリストがラインプリンタに出力され，新しく作られたファイルを代表する項目がインデックスにつけ加えられる．

ii) BROUGHT ファイルが 仮想的ファイル INTERESTING. DATA の場合
・一般の磁気テープからデータを読み込んで PROFISS ファイルを作るルーチンが起動される．

iii) BROUGHT ファイルが 仮想的ファイル FANTASTIC. DATA の場合
・キーボード入力による PROFISS ファイルの作成ができる．
・新しい PROFISS ファイルが作られて EDITOR が起動される．

e) ¥SYSBACKUP

機能：PROFISS テープのコピー

説明：このコマンドを投入すると，次の段階を追って処理される．

1. インデックスを検索して MARKEDITEM と書き込まれた項目をさがす．なければ READ モードにもどる．
2. 記号列 MARKEDITEM を消去する．
3. この項目によって代表される PROFISS テープを磁気テープ装置にマウントすることを要求する．
4. マウントされたテープに記録されているレコード数を数えて，PROFISS テープの識別コードとして登録されているレコード数と一致することを確認する．一致しなければ作業を打切る．
5. 新しい磁気テープを磁気テープ装置にマウントすることを要求する．
6. PROFISS テープから新しいテープにデータを送る．
7. READ モードに戻る．

注意：MARKEDITEM は WRITE モードの¥MK コマンドであらかじめ書き込んでおく．

4.3. WRITE モードのコマンド

記録の追加と修正のためのコマンド群である．以下の規則を頭にいれておいた方がよい．

WRITE モードと READ モードにおける画面表示

・画面上の1行を「表示行」と呼ぶ．
・WRITE モードのコマンドによる処理では，項目全体がひとつながりの記号列として扱われる．連続する表示行は空白1つをおいて接していると見なされる．
・WRITE モードで表示される ¥ は改行マークである．
・READ モードの表示でもこの ¥ の位置で改行されるが ¥ は画面上に現れない．
・¥ を追加あるいは消去することによって改行位置をかえられる．
・¥ から ¥ までの間隔が長いときは自動的に改行される．
・WRITE モードで#と$で表示される位置は，READ モードにおける表示に際してそれぞれ1ケのスペースと¥で置き替えられる．ラインプリンタへの印字も同様．
・WRITE モードで表示される¥が2つ以上連続した¥¥以降の部分は READ モードでは表示されない（スタックには全体が積まれる）．

a) テキストの追加

形式：以下のコマンドとして解釈できないすべての入力

説明：・項目の最後に追加される．
・¥ によって改行位置を指定できる．
・2つ以上連続した空白は1つの空白で置きかえられるのでREADモードの表示にスペースを確保するためには#を置く必要がある．

b) 記号列の消去，書き替え

形式：¥CH（ｘｘ）〔¥TO〔(ｙｙ)〕〕

説明：i) ¥CH（ｘｘ）：項目中に表われる最初の記号列ｘｘが消去される．
ii) ¥CH（ｘｘ）¥TO（ｙｙ）：項目中の最初のｘｘがｙｙに書き替えられる．
iii) ¥CH（ｘｘ）¥TO：項目中の最初のｘｘ以降がすべて消される．

注意：上の（ｘｘ）の部分はｘｘを任意の（ｘｘ中に含まれない）同じ記号で前後からはさんだもので代用できる．

c) 終了

機能：テキスト追加，修正の終了

形式：〔¥OK | ¥MK | ¥MKGB〕

説明：i) 空行の送信
・修正を終了して一時停止し，スタック中の次の項目以降の修正を続けるか否か指示を待つ．修正終了を指示すればREADモードに戻る．
・この時スタックに残るのは，修正のすんだ項目までである．
・終了に際して，項目の最後に¥NEWITEMという記号列がつけ加えられる．
ii) ¥OKの送信
・¥NEWITEMをつけ加えずに（もし¥NEWITEMという記号列がすでに存在すれば，これを消去して）終了する．
・以降の処理は空行を送信した場合と同様．
iii) ¥MKの送信
・項目の最後に ¥MARKEDITEM が追加されて終了．
・この利用法に関しては¥STORE, ¥SYSBACKUPの項を参照されたい．
iv) ¥MKGBの送信
・項目の最後に ¥GARB¥MARKEDITEMAGE を追加して終了する．
・利用法に関しては，¥STORE, ¥ERASEの項を参照されたい．

d) ¥HELP

機能：WRITEモードコマンドの機能表示 および 誤消去テキストの回復

説明：¥CH（ｘｘ）¥TO で項目の後半を消去した直後であれば消去する前の状態を回復する．

4.4. EDITモードコマンド

a) ENC

機能：ENC (losure). EDITに際して修正部分をあらかじめ「囲い込ん」でおく．

形式：ENC ss ee

説明：ｓｓ行からｅｅ行までをコピーしてコメント行として残すとともに，このコマンドを発行した年月日時分秒の記録をコメント行として加える．

説明：この形で残された原テキストとコメントは¥COPYコマンドによるファイルのコ

ピーでは複写されない.

b) GO

機能：EDIT モードから DEBUG モードへの切りかえ

形式：GO

注意：EDITOR の END サブコマンドを発行しないでいきなり GO とすること.

5. しくみ

PROFISS の機能を実現しているしくみを説明する. とにかく動けば良いという方針で作り上げられたしくみであって, 能率や簡潔性に関しては最小限の考慮しか払われていないことを予め断っておく.

5.1. ソフトウエア構成

PROFISS は表4にまとめた大小19個のルーチンで構成されたシステムである. 各ルーチンは FORTRAN またはコマンドプロシージャ文（あるいは, その両方）で書かれている.

- サイズの F, C, J の後の数値はルーチンを構成する FORTRAN プログラム, コマンドプロシージャ文, ジョブ制御文それぞれの行数を示す.
- ルーチンは原則としてサイズの大きい順に並べてある.
- 表中の入出力ファイル名は略称であり最後に識別子「.DATA」が省かれている.

5.2. コントロールと情報の流れ

表4から軸となる部分を抜きだして図示したのが図2である. コマンドプロシージャ BRING と FORTRAN プログラム RETRIEVE が中心的な役割を果たしている.

形式的には RETRIEVE を起動する（⑲, ⑮）BRING が上位にあるが, 一度コントロールを握った RETRIEVE はユーザーからのコマンドを受けつける窓口として機能し(⑳), ファイル STATUS を通じて BRING ルーチンを制御する信号を流す（⑪, ⑩）ことによってシステム全体を支配下におさめる.

RETRIEVE ルーチンがユーザーからのコマンドを受けつけて実行ルーチンを統括する BRING に橋渡しするわけであるが, WRITE モードコマンドおよび READ モードコマンドのうちの検索コマンドは RETRIEVE ルーチンの中で処理される.

たとえばユーザーが検索コマンドを発行する（⑳）と RETRIEVE がインデックスから所定の条件を満足する項目を抽出（⑰）して端末に表示して次のコマンドを待つ.

検索コマンド以外の READ モードコマンドを処理するのは BRING である.

たとえばユーザーが ¥STORE コマンド を発行する（⑳）と, RETRIEVE はファイル STATUS. DATA に情報を書き込んで(⑪)からコントロールを BRING に戻す(⑯). BRING は STATUS. DATA を読んで（⑩）NEWTAPE ルーチンを起動する（⑦).

NEWTAPE ルーチンは PROFISS ファイルから PROFISS テープにデータを転送(①)して BRING にコントロールを戻す(③)が, BRING は必ず RETRIEVE を起動(⑮)するのでユーザーにコントロールが戻って来る.

¥BYE コマンドによって TSS モードに戻れる（⑯, ⑧）が, この際 RETRIEVE はスタック内容などの状態を STATUS. DATA に退避（⑪）し, 再び READ モードに入る（⑲, ⑮）時に回復（⑨）する. この「しかけ」によって TSS モードを PROFISS のモードの一つと見なすことが可能になっている.

表4.　PROFISSを構成するルーチン

ルーチン名（サイズ）	機　　能	起　　動	主な入出力	
			入　　力	出　　力
RETRIEVE (F: 1284)	コマンド解析	BRING	キーボード INDEX STATUS	ディスプレイ INDEX STATUS
BRING (C: 722)	メインルーチン	ユーザー	STATUS LOCK INDEX	LOCK FAILSAFE
NEWTAPE (C: 129 F: 86)	¥STORE	BRING	ディスク	磁気テープ
	NEWTAPE	ユーザー	INDEX	磁気テープ
UPDATE (C: 9 F: 182)	UPDATE	ユーザー	SCHDL	SCHDL
MTREAD (C: 137 J: 43)	¥GET	BRING	磁気テープ	ディスク
DEBUG (C: 174)	¥DEBUG	BRING	キーボード	ディスプレイ
BACKUP (C: 70 F: 103)	BACKUP	ユーザー	磁気テープ	磁気テープ
	¥SYSBACKUP	BRING	磁気テープ	磁気テープ
PACKAGE (C: 120 F: 26)	PACKAGE	ユーザー	ディスク	磁気テープ
ERASE (C: 133)	¥ERASE	BRING		
DIVIDER (F: 107)	¥LIBRARY	BRING	MACRO 7	ファイル
	¥DEBUG 前処理	BRING	ファイル	DEBUG DEBUGSUB
FETCH (C: 46 J: 16 F: 37)	¥GET	BRING	磁気テープ	ディスク
RENEW (C: 43 F: 26)	¥COPY	BRING	ディスク	ディスク
PHASE (C: 67)	PHASE	ユーザー	キーボード	ディスプレイ LOCK STATUS
PRINT (F: 66)	¥PRINT	BRING	ディスク	ラインプリンタ
ENC (C: 26)	ENC	ユーザー	キーボード	ディスク
INTRO (C: 13)	INTRO	ユーザー	キーボード	INDEX
RSV (C: 21)	RSV	ユーザー	キーボード	ディスク
XREF (C: 32 F: 14)	¥TOOL (FORT)	BRING	ファイル	ファイル
FILES (C: 73)	¥TOOL (FILE)	BRING	ディスク	ディスプレイ

注）個々のルーチンもコマンドプロシージャ文、FORTRAN プログラムなどをくみあわせてつくられている。

TSS モードにおけるコマンド処理は TSS がコマンドプロシージャを起動する形で実行してくれる．たとえば，PROFISS テープを新しく作るときに，TSS モードで　NEWTAPE コマンド　を発行すると直ちに　コマンドプロシージャNEWTAPE　が起動される(②，このコマ

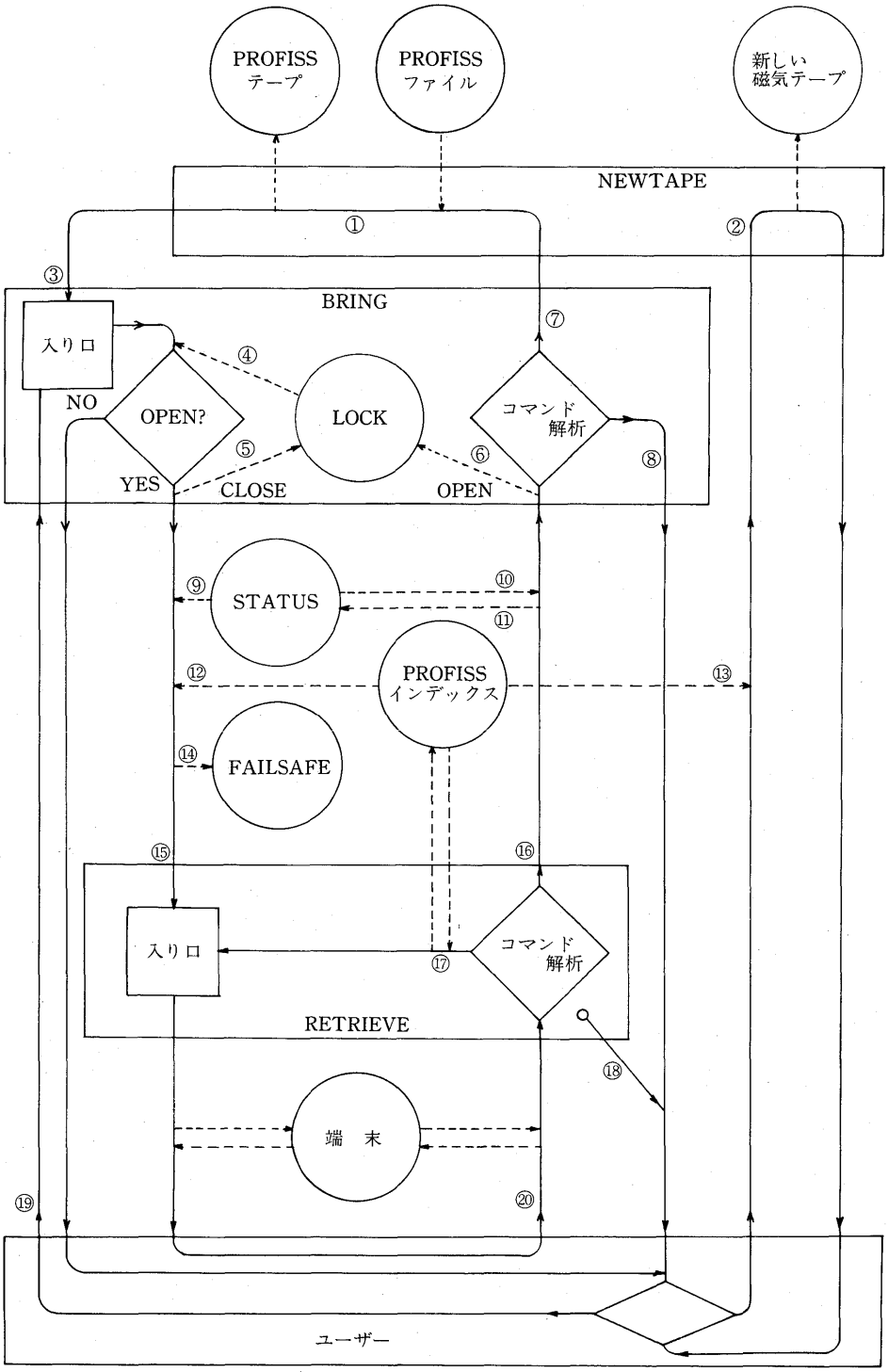

図2. PROFISSにおけるコントロール（→）と情報（⇢）の流れ

ンドプロシージャは　2つのコマンド　¥STORE　と　NEWTAPE　の処理を受け持っている).

5.3.　事故対策

PROFISS において最も大切なものである PROFISS テープとインデックスを事故から守る工夫が為されている.

a)　PROFISS テープの保護

PROFISS テープは私用の磁気テープとして作成されるものであって，親システムの管理運営体制によるバックアップを期待できない．ユーザー自身によるバックアップテープ作成を促し，支援する体制をもって保護対策としている．具体的には以下の方策が講じられている.

i)　¥STORE コマンドによって PROFISS テープにデータを転送しただけでは PROFISS インデックスに登録されない．バックアップテープを作って初めて登録される．インデックスへの登録が済んでいるか否かは　PHASE コマンド　で調べられる.

ii)　¥STORE　後のバックアップとは別に年に一度のバックアップのプロンプトがカレンダー生成ルーチンに組み込んである.

iii)　PROFISS テープが何本であろうとも，紙と鉛筆を使わないでもれなく全テープのコピーが作れるようなコマンドが用意されている．ユーザーはバックアップを取りたいテープを代表するインデックスの項目に　¥MK コマンド　で印をつけておいてから　¥SYSBACKUP コマンド　を発行し，あとは PROFISS の指示に従っていれば良い.

なお，PROFISS テープは書き込まれているレコード数で識別され，テープの掛け違いによるトラブルの確率は低い.

b)　PROFISS インデックスの保護

これはディスク上に常駐しているので管理運営上行われるバックアップの対象となっており PROFISS テープほど気を使わなくて良いが，4つの面で保護対策がとられている.

i)　NEWTAPE コマンドで新しい書きこみ用の PROFISS テープを作るとき必ず PROFISS インデックスのコピーを作る(図 2.⑬, ②)．なおこのとき同時にそれまで書きこみ用だった PROFISS テープが読みだし専用として新たにインデックスに登録される.

ii)　FAILSAFE. DATA という名のファイルが用意されていて READ モードの新規起動(図 2.⑲, ⑮)に際して BROUGHT インデックスのバックアップコピーが作られる(⑫, ⑭)．この FAILSAFE. DATA は次のような形で保護されている.

1)　BRING が RETRIEVE を起動するにあたってまず LOCK. DATA をチェック(④)し，「OPEN」と書かれていなければ中止.

2)　「OPEN」の場合にはこれを「CLOSED」と書き替えてから

3)　BROUGHT インデックスを読み込んで(⑫) FAILSAFE. DATA にコピーする(⑭).

4)　RETRIEVE の処理中に事故がなく無事に終了(⑯)すると，LOCK. DATA の「CLOSED」は再び「OPEN」に書き替えられる(⑥).

RETRIEVE が異常修了(⑱)したときに BROUGHT インデックスが壊されている可能性があるが，この場合 LOCK. DATA には「CLOSED」が残ることになって無傷の BROUGHT ファイルのコピーが残されている FAILSAFE. DATA の書き替え

が禁止される.

iii) PROFISS インデックスにユーザーが直接手を加えるのは, BROUGHT インデックスとして WRITE モードコマンドによって書き替える場合である. 人手を借りて作業を進めることも多い. このような場合の後の確認作業を楽にするために, 修正が加えられた項目に印が残せるようになっている. すなわち,

- 項目の修正を修了する際に ¥OK コマンド でなく空行を送信すれば「印」として文字列「NEWITEM」が項目の最後に付け加えられる（すでにこの「印」が付いている場合を除く）.
- ¥OK コマンドで修正を修了すると「印」は付加されない. もし「印」があれば取り除かれる.

iv) 不用意なアテンション割りこみによる異常終了を避けるために, アテンションキーが押されたときに再確認を求める手続きが組込まれている.

6. 開発経過と現状

6.1. 開発経過

PROFISS を構成する 19 個のルーチンが表 4 にまとめられているが, 表 4 はルーチンのサイズの順に並べてある. ほぼシステムの構成要素としての重要度の順といってよい. この表をルーチンの作成時期の順に並べかえたのが次の表 5 である.

表 5. PROFISS を構成するルーチンの開発経過

ルーチン名	機能	作成時期	備考
NEWTAPE	¥STORE	80年4月25日	
BACKUP	BACKUP	80年4月25日	
FETCH	¥GET	80年4月25日	
MTREAD	¥GET	83年2月21日	
UPDATE	UPDATE	83年6月	
ENC	ENC	84年4月26日	
RENEW	¥COPY	84年4月26日	
RSV	RSV	84年4月26日	
BRING	メインルーチン	84年5月10日	
RETRIEVE	コマンド解析	84年5月11日	
		84年5月25日	検索機能完成
		84年7月 3日	ファイル管理機能追加
		85年1月18日	DEBUGER 組み込み
		85年2月22日	MANUAL 作成
PACKAGE	PACKAGE	84年6月21日	
ERASE	¥ERASE	84年7月	
INTRO	INTRO	84年7月	
PRINT	¥PRINT	84年9月19日	
DEBUG	¥DEBUG	85年1月	
DIVIRER	デバッグ前処理	85年1月18日	
PHASE	PHASE	85年2月	
FILES	¥TOOL (FILE)	85年6月	
XREF	¥TOOL (FORT)	85年6月	

6.2. 現状

システムの現状に関してはしくみの項で説明した通りであるが，PROFISSの利用状況をまとめると以下の表6のようになる．

表6. PROFISSの利用状況（1985年4月12日現在）

ファイル管理	: PROFISSファイル	605個（ディスク上に 82個）
	: PROFISSテープ	10本（レコード数591,727）
		バックアップ 2組
名簿管理	: 登録項目数 259	
文献管理	: 登録項目数 169	
スケジュール管理	: 登録項目数 321	（過去の履歴を含む）
サブルーチンパッケージ	: ルーチン数 102, レコード数 4586	

謝　辞

統計数理研究所の桂康一氏と松野秀夫氏および仁木直人氏にはTSSの使用法に関して一かたならぬお世話になった．謹んで感謝の意を表する．

FORTRANの仕様に関しては緯度観測所の田村良明氏の「BAYTAP-G　使用手引書」の書き方を参考にさせて頂いた．

最後に，適切なコメントを頂いたレフェリーの方々に心からのお礼を申し上げる．

参考文献

〔1〕 HITAC VOS3 TSSコマンド 資料番号 8090-3-120-80
〔2〕 HITAC VOS3 TSS操作 資料番号 8090-9-105-80
〔3〕 プログラムプロダクト VOS3 最適化FORTRAN77端末使用の手引資料番号 8090-3-222-10

付 録

A.1. FORTRAN とコマンドプロシージャ

FORTRAN は FORTRAN IV（または 66）規格に次の仕様を加えたものを使用している．

a) READ 文における END＝文番号 を用いたジャンプ

b) 単精度変数に 4 文字格納できること（32bit 以上のこと）

コマンドプロシージャは HITAC VOS3 のものであるが，文法的に PL/I や BASIC に近く，1F-THEN-ELSE, GOTO などによる実行制御によって，非常に複雑なプロシージャを作ることができる．インタープリターのもとで動くので，使ってみるのが簡単であり FORTRAN ユーザーなら半日で基本的な使いかたをマスターできよう．

以下の機能を備えている．

a) 文字列データの転送（ファイルや端末との入出力）

b) 文字列演算（文字列定数の設定，結合，部分列の抽出，変数への代入 等）

c) 文字列をコマンドとして発行すること

d) 文字列を数式として評価すること

e) 2 つの文字列を数式，あるいは文字列そのもの，として比較すること．

以上の機能からわかるように，コマンドプロシージャはキーボードやファイルからの文字列入力を加工してコマンドとして発行するユーザーと TSS の間のインターフェースを作るための強力な道具である．

A.2. インデックス仕様のファイル

PROFISS のファイル管理が容易なのはすべてのファイルをカードイメージに統一しているからである．逆にカードイメージのファイルであれば，INTRO コマンド で PROFISS インデックスに登録することによって PROFISS の管理下に置くことができる．

カードイメージというだけでなく，「インデックス仕様」を満足していれば PROFISS の記録保守，検索機能を利用することが出来る．

インデックス仕様

a) カードイメージ（80byte/レコード，すなわち 80 カラム/行）のファイルである．

b) 73-80 カラムの内容が失われても良い．

c) 空白なしで 60 個以上連続する記号列を含まない．

d) 第一カラムが空白である行を先頭行，それ以外の行を継続行，ある先頭行とそれに続く継続行を合わせて「項目」と名付けるが，28 行以上の項目を含まない．

e) 項目の先頭行の第 2-6 カラムに

¥REF¥

が記入されている．

f) 最後はただ一行からなる項目であって，第 2-14 カラムに

¥REF¥LASTITEM

が記入されている．

以上の条件を満足していれば PROFISS の記録保守・検索機能が使えるが，記号 ¥, # および $ は次のように特殊な記号として解釈されるので注意されたい．詳しくは WRITE モードコマンド の項を参照のこと．

特殊記号	意味
¥	改行マーク
#	スペース
$	表示に際して「¥」に変換

またWRITEモードコマンド　や　¥ERASEコマンドを使用するとBROUGHTインデックスのファイルの構造は必ずしも保存されずに,

a) 記号列は各行の第1-60カラムの範囲内に出来るだけ詰めこまれる.

b) その際2個以上連続する空白は1個に詰められる.

c) ただし空白なしで連続した記号列が途中で2行に分けられることはない.

注) 61-72カラムに空白を残すのは　EDITOR　を用いて編集する余裕を与えるためである.編集の結果,行が短くなったり61-72カラムに書きこまれたりしても構わない.
73カラム以降に書き込んだ分は失われるので,行番号を付けてから編集すると良い.

A.3. PROFISSテープの仕様

PROFISSテープの規格は次の通りである.

DENSITY	: 1600　BPI (PE), 9　TRACKS
CODE	: EBCDIC
LABEL	: ラベル無し
RECORD FORMAT	: 固定長ブロック化形式
RECORD LENGTH	: 80　BYTE/RECORD
BLOCK SIZE	: 6240　RECORD/BLOCK
テープ長	: 600フィート

このフォーマットで全ての情報が一つのデータセットに書き込まれている. PROFISSファイルの区切りは第1-20カラムに記入されたエンドマーク

EEEEEEEEEEEEEEEEEEEE

であり,「識別コード」からエンドマークまでが一つのPROFISSファイルになる. 識別コードに関しては以下のPROFISSテープへの書き込みと読み出しの規則を参照されたい.

読み出し規則に合う形の磁気テープを作ってエンドマークで区切られた各部分の識別コードをPROFISSインデックスに登録（¥ADDコマンドが使える.）すれば, この磁気テープはPROFISSの管理下に置かれる.

¥STOREコマンドによるPROFISSテープへの書き込み

a) PROFISSファイル　ｘｘｘｘ.ｉｉｉｉ　からデータを転送するものとする.

b) システムから年月日（ｙｙ-mm-ｄｄ), 時分秒（ｈｈ：ｎｎ：ｓｓ）を読み出す.

c) ファイルｘｘｘｘ.ｉｉｉｉの最初に識別コード

Cyy-mm-dd-hh:nn:ss　xxxx

を追加する. ただし　ｉｉｉｉ=「PAIR」であるPAIRファイルの場合は　xxxx.FORT　の方に書き込まれる.

d) PROFISSテープの最後につぎのレコードを追加する.

i) ｘｘｘｘ.ｉｉｉｉ　の識別コード, すなわち　ｘｘｘｘ.ｉｉｉｉ（ｉｉｉｉ=「PAIR」ならxxxx.FORT）の最初のレコード.

ii) 「COMMENT」 (iiii=「PAIR」の場合のみ第1-7カラムに)
iii) xxxx. CNTL の全体 (iiii=「PAIR」の場合のみ)
iv) 「COMMENT」 (iiii=「PAIR」の場合のみ)
v) xxxx. iiii の残り全部. (iiii=「PAIR」の場合は xxxx. FORT)
vi) 「EEEEEEEEEEEEEEEEEEEE」 すなわちエンドマーク.

PROFISS テープからの読み出し規則

a) ファイルｘｘｘｘ. ｉｉｉｉにデータを転送するものとする.
b) テープの最初から探しはじめて, 第1-20カラムが求める識別コードに合致する行を見付ける.
c) そこから次の順にデータを転送する.
i) 最初のレコードすなわち識別コードをｘｘｘｘ. ｉｉｉｉ（ｉｉｉｉ=「PAIR」ならｘｘｘｘ. ＦＯＲＴ）に.
ii) 次の行が「COMMENT」であったらiii)に行く. そうでなかったらこれもｘｘｘｘ. ｉｉｉｉ（ｉｉｉｉ=「PAIR」ならｘｘｘｘ. ＦＯＲＴ）に転送してからiv)に行く.
iii) 再び「COMMENT」が現れるまで全ての行をｘｘｘｘ. CNTLに転送する. 「COMMENT」は転送せずにiv)へ.
iv) 「ＥＥＥＥＥＥＥＥＥＥＥＥＥＥＥＥＥＥＥＥ」すなわちエンドマークが現れるまでの残り全部をｘｘｘｘ. ｉｉｉｉ（ｉｉｉｉ=「PAIR」の場合は ｘｘｘｘ. FORT）に転送.

なお, PROFISS テープの規格のうち DENSITY とテープ長を変えるのは難しくない. 一定の長さの磁気テープに書き込めるレコード数で比較してみると, 標準ラベル付きのファイルとして記録するのにくらべて PROFISS 方式の記録の効率が高いこと, 一本のテープにあまり多くのファイルを記録することは必ずしも得策ではないことを考慮して現在の規格が採用されている.

A.4. PROFISS 導入手続

以下の手続きをとる.

1. MCATPROC. CLIST というファイルに PROFISS コマンドプロシージャファイルを複写する.
2. コマンド
PROFILE PROC (MACTPROC. CLIST)
を発行する.
3. コマンド
PROFISS
を発行する.

PROFISS

A PROgraming and FIling Support System for FORTRAN Users

Makio Ishiguro

(The Institute of Statistical Mathematics)

PROFISS is a programing and filing support system designed for FORTRAN users. This paper discribes the function and construction of PROFISS.

The features of this system are as follows:

1) The system provides a powerful information keeping and retrieval function which is sufficient to handle hundreds of files of source programs and/or data.
2) Under the supervision of this system, it is easy to apply programing tools such as DEBUGER or EDITOR to a FORTRAN source program. Necessary datasets and working files are allocated automatically by the system.
3) The function of the system is realized by command procedures and FORTRAN programs, which means that every user can impliment this system without any special support from the supervising operating system.
4) Though, FROFISS is now operating only on HITAC machine with VOS-3 operating system, it should be easy to implement this system on IBM or other derivative machines.