

Title	ACOSのファイル（その2）：IIテンポラリファイルとシステムファイル
Author(s)	多喜, 正城
Citation	大阪大学大型計算機センターニュース. 1977, 25, p. 39-54
Version Type	VoR
URL	https://hdl.handle.net/11094/65347
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

ACOS のファイル (その2)

Ⅱ テンポラリ ファイルとシステムファイル

研究開発部 多喜正城

- 1) ファイルの編成
- 2) テンポラリファイル
- 3) システムファイル

この章では簡単にファイルの編成法を示し、次に、プログラム管理の面からプログラムの流れと、その時のテンポラリファイルについて述べています。

1) ファイルの編成方法

データ管理のもとでは次の4種類の編成法を成しています。

- ① 順編成
- ② 直接編成
- ③ 索引順編成
- ④ IDS (ここでは省略)

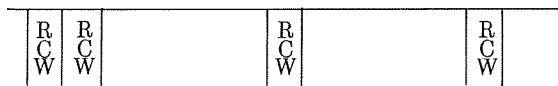
①, ②のアクセスはFRC (File Record Control)にまた③はISP (Index Sequential Processor)により行なわれます。

① 順編成ファイル

これは、データが処理される順序と各レコードの物理的順序が一致し、すべての周辺装置と端末に対して適用が可能であり、データを格納する場合の最も基本的なファイルです。レコード形式は、実際には4種類ありますが、“システム標準形式”を用いることにより、FRC (ファイル・レコード・コントロール)はプログラムの実行時実際に割り当てられた装置の形式に従った処理をしますので、利用者はプログラム作成時、ファイルの定義をシステム標準形式で定義しておきますと、そのプログラムを変更することなく、そのジョブ実行時の状況に応じて周辺装置のタイプを変えたり、割り当てが可能です。(あまり周辺装置を意識せずに使えると言うことですネ。)

システム標準形式の規則

- 最大ブロックサイズ……………320語以内
- レコード形式 ……………可変長レコード (図に示す)



論理レコード 論理レコード

BCW ; ブロック制御語

RCW ; レコード制御語

- ブロック制御語……………各ブロック（物理レコード）の第1語目に必要
- レコード制御語……………各論理レコードの第1語目に必要
- レコーディングモード……………磁気テープの場合，バイナリモード（奇数パリティ）
高密度
- ラベル……………磁気テープの場合，標準ヘッダラベルと標準トレーラ・ラ
ベルをもつ

② 直接編成ファイル

まずファイルの媒体は，磁気ディスクに限られます。ファイルは64語のブロックに分割され，各ブロックに相対アドレスが付けられます。直接編成ファイルは，相対ブロック番号と入出力データ長を指定した物理レコード処理命令（READ/WRITE）で処理することができます。すなわち，FRCでは，ブロックの形式およびブロック内のレコード形式を規定していないので，利用者は自由な形式のファイルを作成することが可能です。

（①，②の詳細はFRC説明書を参照されたし。）

③ 索引順編成ファイル

一つの索引順編成ファイルは，データファイルと索引ファイルの2種類で構成されている。データファイル，索引ファイルともページと呼ぶ一定長の単位に分割されており，ページサイズは利用者が指定します。索引順編成ファイル処理においては，主記憶と直接アクセス装置間のデータ転送は，ページ単位に行われます。索引部は，コースインデックスとファインインデックスよりなり，コースインデックスより，ファインインデックスのページ番号を示し，ファインインデックスは，データファイルのページ番号を示す。データファイルはキーの小さい順に並んでおり，データファイルの各ページには，1語のデータページ制御語と1つ以上のレコードが含まれている。各レコードの前には，レコード制御語が，後には，ポインタ語がついていて，ポインタはファイル中に存在するレコードの中で，このレコードの次に大きいキー値をもつレコードの位置を指定している。

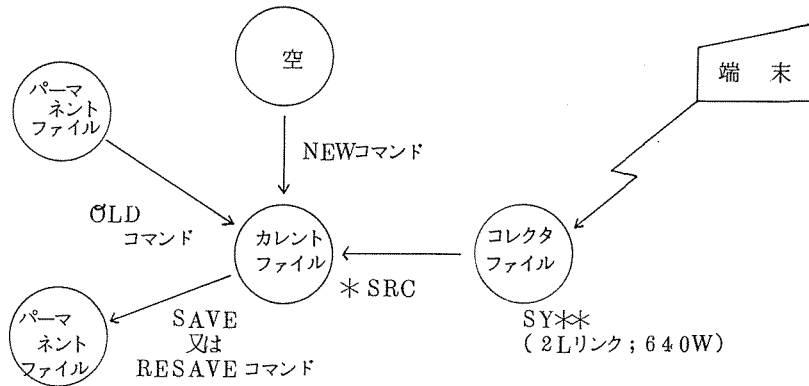
（③の詳細は，ISP説明書を参照されたし。）

2) テンポラリィファイル

利用者のJobは各アクティビティ毎に，それぞれのテンポラリィファイルを使う。ここでは少しバッチ系の方面に目を向け，そこでのテンポラリィファイル（システムファイル）の使われ方をながめてみることにし，T.S.S.でのRUNコマンド，又，会話形リモートバッチにおけるCARDINの効果的な使い方を考える。

- T.S.S.における標準テンポラリィファイル

T.S.S.におけるファイルシステムには，標準テンポラリィファイルにより利用者がそれらを意識せずにシステムとの会話ができます。



○ニューファイル

利用者がNEWコマンドを用いたときに利用者に用意されるテンポラリィファイルである。

○オールドファイル

既に作成され、そしてセーブされたファイルで、利用者はOLDコマンドを指定してカレントファイル上にコピーされる。

○カレントファイル(*SRC)

ファイル名は*SRCである。これは利用者に割り当てられたテンポラリィファイルである。このファイル上にニューファイルが作成されたり、指定されたオールドファイルがコピーされたりする。カレントファイルには最も新しいNEWあるいはOLDコマンドで選び出されたファイルが、その後の修正とともに入っている。従って修正はファイルがSAVEコマンドまたはRESAVEコマンドでセーブされるまでは一時的なものにとどまる。

○コレクタファイル (SY**)

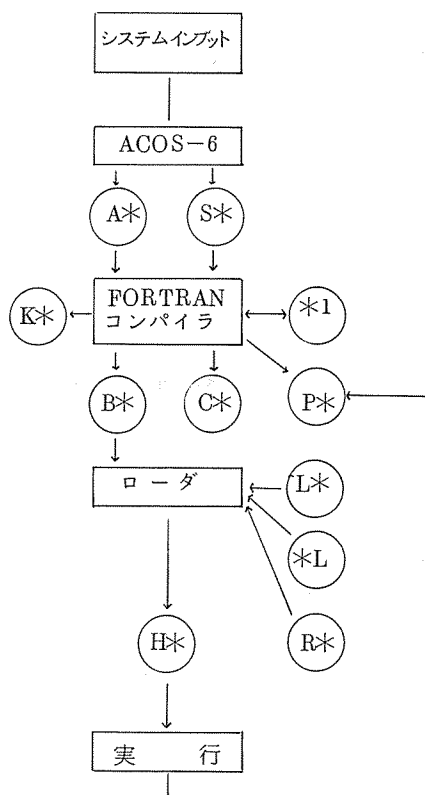
ファイル名はSY**である。これはログオンするとき利用者に対して割り当てられるテンポラリィファイルである。カレントファイルに対する追加更新データの格納ファイルである。

3) システムファイル

ACQS-6が自動的に作成し利用するファイルです。例えばあるジョブストリームをシステムインプットが読み込み、プログラムのソースプログラムを見つけるとACOS-6は“S*”名をもつコンパイラ・ソース・ファイルというシステムファイルを作成し、すべてのソースプログラムをS*の中に格納する。このファイルは後にコンパイラによって利用され、オブジェクト形式に変換され“B*”名をもつオブジェクト・プログラム・ファイルに格納されローダへの入力となります……といったようにシステムが自動的に作成し利用するファイルですので、利用者はこの種のファイルにあまり注意をする必要はありませんが、それぞれのファイルへの流れを知ることにより、ジョブ制御言語のファイル定義文により有効な利用方法を見つけ

出せるでしょう。

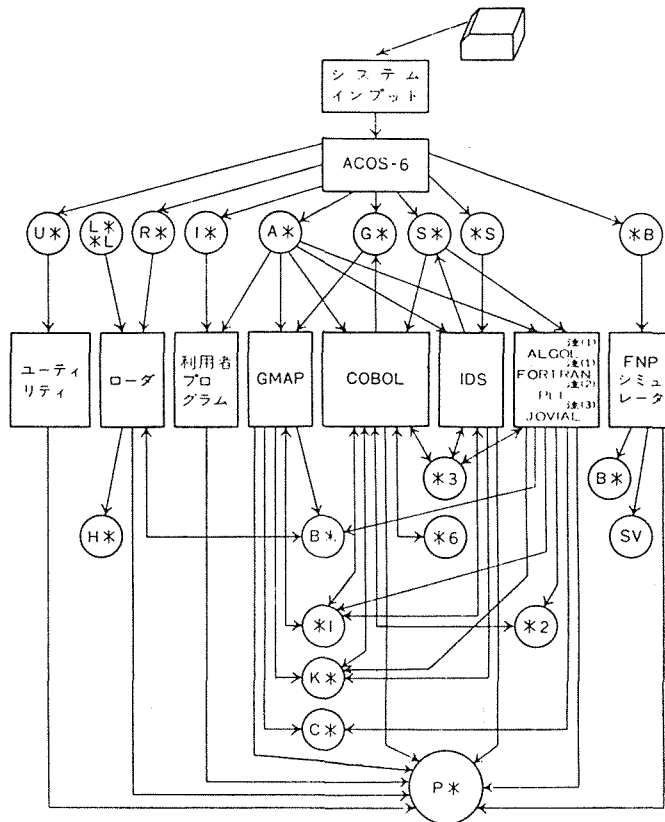
例えば FORTRAN の場合次図のようになります。



システムファイルは2文字のファイルコードで参照され、その2文字のうち一方がアスタリスク“*”であるということである。

一方L1やA4、AB、等は利用者^{*}のファイルに対して用いられるファイルコードである。システムファイルはシステム標準形式である。この章ではそれぞれのシステムファイルについて記述を行う。(以下システム管理説明書より抜粋する)

*) 利用者ファイルとも呼ばれ \$ P R M F L 文等のパラメータとして使われ、同じく2文字のファイルコードである。



- 注(1) *2, *3は使用しない。
 (2) *1, *2は使用しない。
 (3) *3, は使用しない。

図 システムファイルの相互関係

- A* オルタファイル (alter file)
- B* オブジェクトプログラムファイル
- *B シミュレータインプットファイル
- C* バイナリデックファイル
- D* ストレンジャオプションファイル
- G* GMAPソースファイル
- H* プログラムリンクファイル
- I* データストレージファイル
- K* コンプレスデックファイル
- L* システムサブルーチンライブラリファイル
- *L 設置システムサブルーチンライブラリファイル
- P* システムアウトプットファイル
- R* ローダインプットファイル
- S* コンパイラソースファイル
- *S IDSソースファイル
- U* ユーティリティファイル
- *1 中間ファイル (FORTRAN, COBOL, ALGOL, JOVIAL IDS および GMAP)
- *2 中間ファイル (COBOL, JOVIAL)
- *3 " (IDS, COBOL)
- *4 } 中間ファイル (エディタ)
- *5 } 中間ファイル (エディタ)
- *7 } 中間ファイル (エディタ)
- *6 " (COBOL)

表 システムファイルとプログラムとの関連

プログラム	使用ファイル	サイズ	タイプ
ALGOL	A*	可変	順編成 ^{注(1)}
	B*	2リンク	順編成 ^{注(2)}
	C*	—	
	K*	—	
	P*	—	
	S*	可変	順編成 ^{注(1)}
	*1	4リンク	順編成 ^{注(2)}

プログラム	使用ファイル	サイズ	タイプ
COBOL	A*	可変	順編成 ^{注(1)}
	C*	—	
	G*	4リンク	順編成 ^{注(2)}
	K*	—	
	P*	—	
	S*	可変	順編成 ^{注(1)}
	*1	4リンク	順編成 ^{注(2)}
	*2	1リンク	順編成 ^{注(2)}
	*3	15リンク	直接編成
*6	10リンク	直接編成	
FORTRAN	A*	可変	順編成 ^{注(1)}
	B*	3リンク	順編成 ^{注(2)}
	C*	—	
	K*	—	
	P*	—	
	S*	可変	順編成 ^{注(1)}
	*1	4リンク	順編成 ^{注(2)}
ローダ	B*	可変	順編成 ^{注(1)}
	H*	5リンク	順編成，直接編成
	L*	—	
	*L	—	
	P*	—	
	R*	可変	順編成 ^{注(1)}
PLI	S*	可変	順編成 ^{注(1)}
	A*	可変	順編成 ^{注(1)}
	B*	3リンク	順編成 ^{注(2)}
	C*	—	
	K*	—	
	P*	—	
	*3	6リンク	直接編成
GMAP	A*	可変	順編成 ^{注(1)}
	B*	2リンク	順編成 ^{注(2)}
	C*	—	

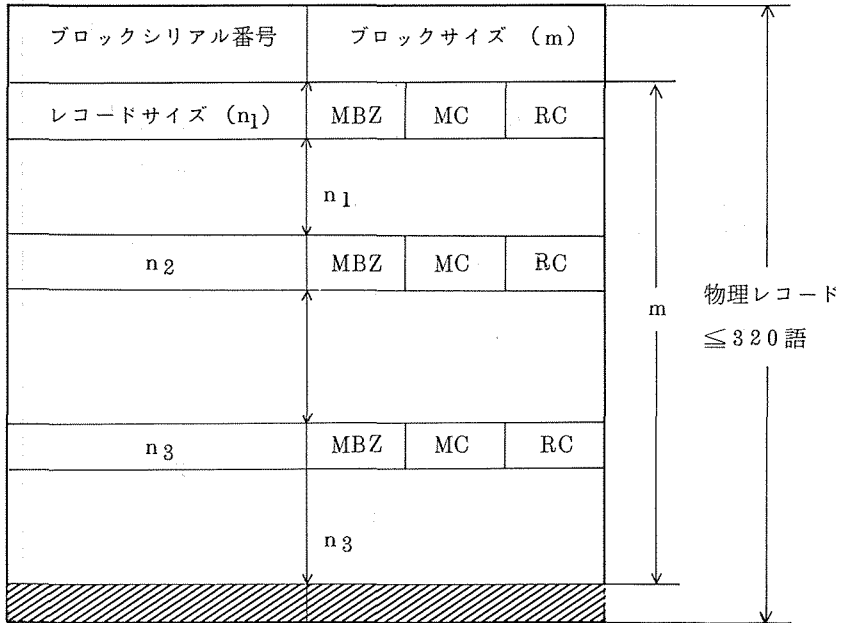
プログラム	使用ファイル	サイズ	タイプ
	G* K* P* *I	可変 — — 4リンク	順編成 ^{注(1)} 順編成 ^{注(2)}
IDS	A* C* K* P* S* *S *I *3	可変 — — — 6リンク 可変 4リンク 15リンク	順編成 ^{注(1)} 順編成 ^{注(2)} 順編成 ^{注(2)} 順編成 ^{注(2)} 直接編成
さらに、COBOLの項で述べられているファイルを必要とする			
JOVIAL	A* B* C* K* P* S* *I *2	可変 2リンク — — — 可変 4リンク 1リンク	順編成 順編成 順編成 ^{注(2)} 順編成 ^{注(2)} 順編成 ^{注(2)}
ユーティリティ	P* U*	— —	
FNPシミュレータ (FNPSIM)	*B P* SV	2リンク — —	順編成 ^{注(2)}
ソースライブラリエディタ	F* J* *M* *Z *K *K*	1リンク 可変 可変 5リンク 可変 可変	順編成 ^{注(2)} 順編成 ^{注(1)} 順編成 ^{注(1)} 直接編成 順編成 ^{注(1)} 順編成 ^{注(1)}

プログラム	使用ファイル	サイズ	タイプ
注 ×印のついて いるシステム ファイルは利 用者が指定す る必要がある×	A*	可変	順編成 ^{注(1)}
	*C	可変	順編成 ^{注(1)}
	B*	可変	順編成 ^{注(1)}
	*1	可変	順編成 ^{注(1)}
	G*	可変	順編成 ^{注(1)}
	*2	可変	順編成
	*3	可変	直接編成
	C*	—	
	P*	—	
	*5	可変	順編成
*7	可変	順編成	
オブジェクトライブラ リエディタ	F*	1リンク	順編成
	B*	可変	順編成
	× *R	可変	順編成
	× R*	可変	順編成
	*4	可変	順編成
	*5	可変	順編成
	× *7	可変	順編成
P*	—		

注(1) システムインプットによって作成されたファイルは、データを格納するのに必要なだけリンクを使用する

(2) 中間データファイルとして使用されるファイルは、表にあるように初めは最少のリンク数を割り当てられるが、言語処理プログラム等によって使用されている間に、MME GEMOREを用いて自動的にその数が増やされる。

図 システムファイルとシステム標準形式

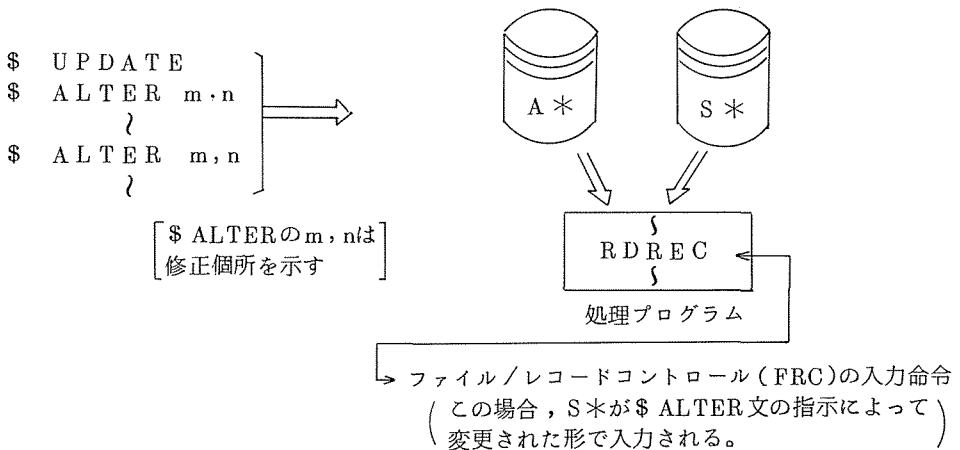


MC : 論理レコードタイプ (メディアコード)
 RC : レポートコード
 MBZ; ゼロ (Must Be Zero の略)

○システムファイルの説明

1 オルタファイル (A*)

オルタファイル (A*) はシステムインプットによって作成され、GMAP, ALGOL, COBOL, IDS, JOVIAL, FORTRANによって使用される。\$UPDATE文を読むとすぐに、システムインプットはその\$UPDATE文の後の\$ALTER文をA*ファイルに入れる。このファイルはアセンブルあるいは翻訳中のプログラム変更で使用される。A*ファイルは、\$ALTER文とソースあるいはシンボリックなランゲージカードイメージで構成され、これらはプライマリインプットファイルのカードとマージされる。



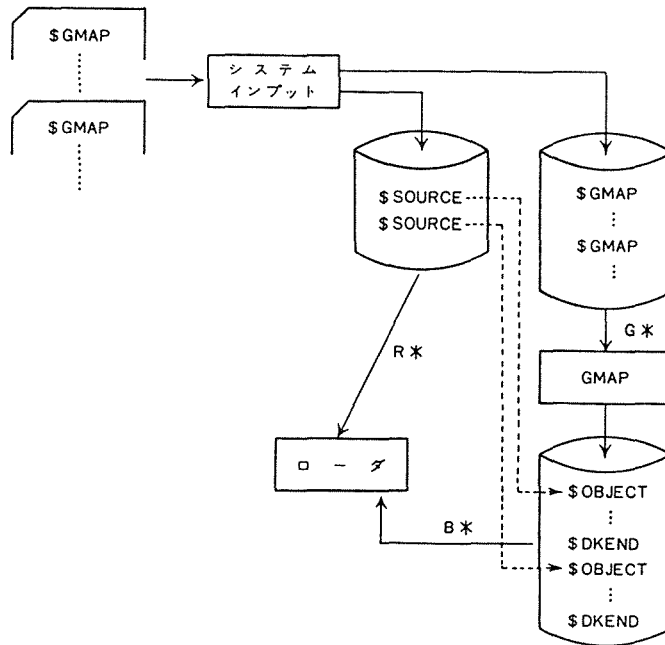
2 オブジェクトプログラムファイル (B*)

オブジェクトプログラムファイル (B*) は, GMAP, JOVIAL, FORTRAN または ALGOL で, \$EXECUTE 文がソースプログラム以降に存在するとき (すなわちプログラムスイッチワードのビット 5 が 1 のとき) 作成され, ローダがオブジェクトプログラムとして入力する。ローダの入力ファイル R* 中に \$SOURCE 文が存在すると, ローダは B* 中にオブジェクトプログラムが存在するとみなし, B* からの入力を始める。

B* ファイルの最初の大きさは 2 リンクである。ただし, 各言語処理プログラムは, 必要に応じて GEMORE を出すことによって, リンクを増やす作業を行っている。

備考 R* 中に \$SOURCE 文を入れるのはシステムインプットが自動的に行う。

ソースプログラム



3 シミュレータインプットファイル (*B)

このファイルは, FNP (N7293) シミュレータ制御文とオブジェクトデッキを入れるために, FNP シミュレータ (FNPSIM) によって使用される。またこのファイルは, N7293 FNP のカード読取装置の入力をシミュレートするためにも使用される。

4 バイナリデックファイル (C*)

このファイルには、GMAP、ALGOLやFORTRAN等によって作成されたバイナリオブジェクトプログラムが、後でカードせん孔されるために入れられる。

\$ OBJECT

```
  .
  . } オブジェクトプログラム
  .
```

\$ DKEND

このC*ファイルは、それぞれのシステムコール文上に、オプションとして「DECK」が指定されると作成される(\$GMAP DECK, \$FORTY DECK等)。

このファイルは通常SY SOUTへ置かれる。

システム標準形式(図を参照)の第1語のレポートコード(ビット30~35)は76₈である。

5 スtrenジャオプションファイル (D*)

このファイルは、システムコール文に指定されているストレンジャオプションに対して、ペリフェラルアロケータによって作成される。ストレンジャオプションは、利用者スイッチワード内のビットで表すことのないオプションである。

ジョブ制御文のオペランドにストレンジャオプションが検出されると、そのカードのイメージおよびそれに関連した\$ETCカードがD*ファイルに書き込まれる。

D*ファイルには1Lリンク(320語)がとられる。これを作成したアクティビティとだけ関連づけられ、アクティビティ間で保存されることはない。

6 GMAPソースファイル (G*)

このファイルはシステムインプットあるいはCOBOLによって作成され、GMAPによって使用される。システムインプットの場合は、\$GMAP文を見つけるとG*ファイルを作成する。COBOLコンパイラは、出力ファイルとしてこのファイルを用いる。このファイルはその後GMAPアセンブラへの入力として使用される。

最初このG*ファイルに4リンクが割り当てられる。しかし、その大きさはシステムインプットまたは言語処理プログラムによって動的に変更される。

7 プログラムリンクファイル (H*)

このファイルはローダによって作成され、メモリへ読み込んだり、メモリから書き込まれたりするテンポラリファイルとして、利用者プログラムによって使用される。H*ファイル内の連続するセグメントをロードする作業はLLINKあるいはLINKサブルーチンによって行われる。

ローダはプログラムがオーバレイされた順にH*に書き出して行く。オーバレイセグメントの最初の2語には、(1)DEBUGテーブルを使うなら、そのエンリポイントと位置および(2)

メモリの上限，下限のアドレスが入る。

再使用を考えない場合，H*ファイルは普通磁気ディスク上にとられ，\$LINK文があるならば5リンクが与えられる。

8 データストレージファイル（I*）

このファイルはシステムインプットによって作成され，プログラム自身が使用する。

9 コンプレストデックファイル（K*）

このファイルはソースプログラムを圧縮形にしたもので，COMDKオプション指定のなされた言語処理プログラムが作り出す。ソースプログラムはオルタファイルの内容とマージされながら，このファイルに圧縮された形で出力される。この変更や修正を伴ったプログラム自身は，その後言語処理プログラムの入力として取り扱うことができる。通常K*ファイルはSYSOUTファイルに割り当てられるが，磁気テープファイルとかディスクファイルまたはカードファイルに割り当てて置いて保存しておくことができる。

10 システムサブルーチンライブラリファイル（L*）

システムサブルーチンライブラリを格納しておくファイルである。

FORTRAN I/Oおよび関数ライブラリとかファイル/レコードコントロールといったサブルーチンはこのファイル中に置かれ，ローダの入力ファイルとして利用される。システムサブルーチンライブラリおよび次に述べる設置システムサブルーチンライブラリは，システムスタートアップ時に定義することが可能である。これらのライブラリはオブジェクトライブラリエディタによって作成されるが，さらにランダムライブラリゼネレータ（RANLIB）を通すことによって，直接編成ファイルライブラリの形式にしておいて利用することもできる。

なお，参考までにローダのサブルーチンライブラリ検索順序を以下に示す。

(1) 利用者サブルーチンライブラリ

\$LIBRARY文で指定されるファイルコードに対応するものを指す。長大10個まで指定可能で，そのうちではオペランド上に現れる順に検索がなされる。

(2) 設置システムサブルーチンライブラリ

*Lというファイルコードで参照されるライブラリを指す。

(3) システムサブルーチンライブラリ

L*というファイルコードで参照されるサブルーチンライブラリ。

11 設置システムサブルーチンライブラリファイル（*L）

このサブルーチンライブラリは，前述のシステムサブルーチンライブラリと同様な性格をもつ。システムサブルーチンライブラリは原則としてメーカーが提供するのに対し，このライブラリは設置システム（またはセンタ）が必要に応じて作成するという点が異なるだけである。このライブラリが指定されている場合，ローダはL*の検索に先立ってこのライブラリの検索を行う。

12 システムアウトプットファイル（P*）

このファイルは各種ソフトウェアによって出力ファイルとして使用される。P*ファイルは

通常SYSSOUTに割り当てられ、アクティビティのタイプごとに制限行数が決められている。しかし、制限値を超えると、P*を磁気テープや磁気ディスクに割り当てることができる。その後でバルクメディアコンバージョンプログラムを使って出力できる。

各アクティビティと標準出力行数との関係は、次のとおり。

ALGOL	10,000 行
CONVER (バルクメディアコンバージョン)	1,000 //
COBOL	20,000 //
EXECUTE	5,000 //
PROGRAM	5,000 //
EXTEDIT/FILEEDIT	10,000 //
FORTRAN	12,000 //
FILSYS	1,000 //
GMAP	10,000 //
IDS	20,000 //
JOVIAL	10,000 //
PROGRAM	5,000 //
SYSEEDIT	10,000 //
UTILITY	20,000 //
MAP	10,000 //
FNPSIM	5,000 //

行数の計算は、スレーブプログラムによって出力される場合についてだけ行われる。上記標準出力行数以外のものを使用したいとき、\$LIMITS文で行数を指定することもできるが、いずれの場合でも、指定行数以上出力しようとするプログラムがアボートされる原因となる。

13 ローダインプットファイル (R*)

このファイルはシステムインプットによって作成される。システムインプットは、すべてのローダ制御文およびオブジェクトプログラム(\$OBJECT~\$DKEND)をR*ファイルに入れる。言語処理プログラムがシステムコール文(\$GMAP, \$ALGOL等)によって指定されると、\$SOURCE文が自動的に作成されてR*ファイルに書き込まれる。

すなわち、\$SOURCE文は翻訳後すぐ実行するタイプのジョブに対して、オブジェクトプログラムファイル(B*)からプログラムをロードするよう指令する働きをもつ。

ローダはおおのこの制御文を読み、R*ファイルまたはB*ファイル(B*ファイルはオブジェクトプログラムがそのときアSEMBルされたものであれば使用される)からオブジェクトプログラムをロードする。

14 コンパイラソースファイル (S*)

このファイルはシステムインプットによって作成され、COBOL, ALGOL, JOVI

AL, FORTRANが使用する。システムインプットは, \$COBOL文, \$ALGOL文, \$ALGOL文等を読み取ると, すぐにこのファイル(S*)を作成する。この中にはシステムコール文の後のソースカード全部が書き込まれる。コンパイルーションが始まると, S*ファイルが読まれコンパイラへのソース入力として使用される。

15 IDSソースファイル(*S)

このファイルはシステムインプットによって作成され, IDS/COBOLデータステートメントが書き込まれる。このファイルは, IDSトランスレータに対する入力情報として使用され, 標準COBOLコンパイラで処理できる形(S*)に変換される。

16 ユーティリティファイル(U*)

このファイルはシステムインプットが作成し, ユーティリティ(UTILITY)が使用する。このファイルには, ユーティリティのための制御文とデータが入る。

17 中間ファイル(*1)

このファイルはFORTRAN, IDS, JOVIAL, COBOL, またはGMAPによって作成され, 作業用のファイルとして利用される。書き込み作業中に, このファイルに最初に割り当てられたリンク数より多くのリンクが必要になったときは, システムストレージに余裕がある限り増加させることができる。

ソースファイルの大きさによって, あらかじめ4, 8または12のリンクが割り当てられる。

18 中間ファイル(*2)

このファイルはJOVIAL, またはCOBOLによって作成され, JOVIALあるいはCOBOLが作業用のファイルとして使用する。書き込み作業中に, このファイルに最初に割り当てられたリンク数より多くのリンクが必要になったときは, システムストレージに余裕がある限り増加させることができる。

COBOLの場合, 初め1リンクが割り当てられる。

19 中間ファイル(*3)

このファイルはCOBOLコンパイラによって作成され, IDSおよびCOBOLが作業用のファイルとして使用する。15リンクの割り当てられたスペースは, 1プログラムにつきカード約1,000枚分に相当する。この制限を超えた場合, 利用者プログラムは打ち切られる。

20 中間ファイル(*4)

このファイルはオブジェクトライブラリエディタによって使用され, 変更事項の記録用ファイルに関する情報が入る。この*4ファイルは初めに1リンクを割り当てられる。

21 中間ファイル(*5)

このファイルはすべてのエディタエグゼキューションレポートを書き込むため, およびエグゼキューションレポートとエディタエグゼキューションレポートを分けて出力させるために使用される。ソース/オブジェクトライブラリエディタの実行の終わりに, この*5ファイルの内容はP*ファイルに転送される。これらのレポートは, *5ファイルをSYSOUTファイ

ルに割り当てることによって分離しておくことができる。

通常、*5ファイルは最初3リンク割り当てられる。

22 中間ファイル(*6)

このファイルはCOBOLが作業用のファイルとして使用する。

23 中間ファイル(*7)

このファイルはエディタによって使用され、その中にはソースプログラムやせん孔指令によって作成されたオブジェクトデックが書き込まれる。このファイルは利用者が割り当てるものである。

このファイルは磁気テープまたは磁気ディスクに割り当てられる。この磁気テープおよび磁気ディスクは、せん孔カード出力のためのバルクメディアコンバージョンプログラム(BMC)に対する入力として使用されるものである。小さなジョブの場合は、*7ファイルをSYS-OUTファイルに割り当てることもできる。この*7ファイルを直接カードせん孔装置に割り当てることはできない。

以上、システム管理説明書より抜粋して記述しましたが、では実際に、ユーザーがパーマネントファイルやシステムファイルをどのように利用すればよいかを次の第三章でFORTRANを中心にバッチ系及びT.S.S.について述べます。