

会話型リモートバッチシステム利用法

黒坂, 輝彦
九州大学大学院工学研究科情報工学専攻

荒牧, 重登
九州大学情報処理教育センター

<https://doi.org/10.15017/6767922>

出版情報 : 情報処理教育広報. 4 (1), pp.13-35, 1981-05. Educational Center For Information Processing, Kyushu University

バージョン :

権利関係 :



会話型リモートバッチシステム利用法

黒坂輝彦^{*}，荒牧重登^{**}

1. システムの概要	14
1.1 TSSとバッチ処理	14
1.2 会話型リモートバッチ処理	14
2. 使用例	14
3. 使用法	18
3.1 会話型リモートバッチ処理のためのサブシステム群	18
3.2 コマンド一覧	18
3.3 予備知識	21
3.3.1 ファイルについて	21
3.3.2 パンチカードの形式	21
3.3.3 SYSOUTの形式	21
3.3.4 タブ機能	22
3.3.5 ACOSのバッチシステムについて	23
3.4 各コマンドの説明	26
3.4.1 CARDIN	26
3.4.2 PRINTとRUN	26
3.4.3 JMONIとJSTS	28
3.4.4 JABT	29
3.4.5 JOUT	29
3.4.6 SCAN	31
附録1. \$SELECTA制御文	33
附録2. バッチ処理と会話型リモートバッチ処理との関係	34

この説明は、九州大学情報処理教育センターで、タイムシェアリング処理の利用経験のある者を対象に、会話型リモートバッチシステムの基本的な利用法を述べたものである。説明は、「とにかく使える」を主目標にしているため、表現の一部に不正確な点のあることを注意しておく。

本文中、次の略語を用いてある。

タイムシェアリング→TS (Time Sharing)
九州大学情報処理教育センター→(本)センター
九州大学大型計算機センター→大型センター
オペレーティングシステム ACOS-6 → ACOS
カード読取り装置 → CR
ラインプリンタ装置 → LP

本文中の「注」や「備考」は、説明を正確にするためのものであるから、最初は読み飛ばしてもよい。

* 工学研究科情報工学専攻

** 情報処理教育センター

1. システムの概要

1.1 TSSとバッチ処理

ACOSでは、TSSとバッチ処理（→3.8.5）の二つの利用形態がよく使われる。

TSSとは、通常、端末を一人が専有し、利用者は端末を通して計算機と「会話」しながら処理を進めていく利用形態で、応答時間の早いのが特長である。

一方、バッチ処理では、計算機に対する指示を紙カードにパンチして、CRから読み取らせ、出力はLPから行う。入力してから出力が行われるまでの間には、人間の判断が介入しないのが普通である。バッチ処理は、扱うデータ量の多い場合に向いている処理形態である。また、ある種の言語（例：ACOSのアセンブリ言語GMAP）は、バッチでなければ利用できない。

しかしながら、バッチ処理には、

- 紙カードは修正や保管がめんどろ。（床に落して順番が狂ったり、水にぬらして使えなくなったり…）
- 入出力を行うたびに、わざわざ入出力装置の所まで行く必要がある。
- 必要以上にLP用紙を使う。（重い紙束をかかえてうろうろしなければならない）

などの問題がある。

1.2 会話型リモートバッチ処理

バッチ処理の上のような欠点を改善した利用形態が「会話型リモートバッチ処理」である。会話型リモートバッチでは、入力として紙カードを使う代りに、「ファイル」を使用する。すなわち利用者は、TSS端末の前に座り、通常のTS処理でプログラムを作成すると同様に、紙カードを想定したファイル（「カード・イメージ」と呼ぶ）を作成する。パンチしたカードをCRに読み取らせる代りに、端末から"RUN"を入力する。結果を見にLPの所に行く代りに、端末に座ったまま"JOUT"と入力して、LPが出力すべき文字を見ることができる。また、どこまで処理が進んでいるかを監視したり、処理を打ち切ったりすることも、端末から指示できる。

このような機能に加えて、会話型リモートバッチでは、ファイルの内容を8進でダンプして、バッチする機能（参考資料[1]の4章参照）、機械語レベルでのデバックを行う機能（参考資料[1]の5章）、バッチジョブが端末と入出力する機能（参考資料[1]の6章）などがあるが、ここでは説明しないので、必要のある方は参考資料[1]を見られたい。

会話型リモートバッチは、つまり、TSSとバッチ処理の中間的な処理形態と見ることもできるし、両者のインタフェイスをとる機能と見ることもできる。この処理形態は、大型センターの利用者には「FIBジョブ」として知られているものに対応している。

2. 使用例

下の計算機との会話例は、FORTRAN プログラムの翻訳と実行を会話型リモートバッチで行う時のものである。（下線の引かれた部分は、利用者からの入力を示す。ただし省略している部分もある。）

\$\$\$ CON, TSS キーを押す。

ACOS-6 TS1(R7. 1) ON 03/30/81 AT 13:42:28 CHANNEL 2722
USER ID - X0001A0007;X0001 (送)
PASSWORD -- JA5QDE (送)

} ログオン手続き。

* * *

WELCOME TO CONNECT ACOS-6, MR./MISS X0001A0007
PLEASE WAIT HERE TILL MAKING AN END OF THE PROSECUTION OF
CONNECTION

- ① SYSTEM ? CARDIN NEW ②
- ② *AUTOX ②
- ③ *0010##N, J ② タブ文字 (3.34節参照)
- ④ *0020\$:JOB:X0001A0007, X0001, JA5QDE ②
 *0030\$:QFORT:NFORM ②
 *0040 READ(5, 500)A, B ②
 *0050 WRITE(6, 600)"A=", A, "B=", B ②
 *0060 WRITE(6, 610)"A+B=", A+B, "A-B=", A-B, "A*B=", A*B, "A/B=", A/B ②
 *0070 500 FORMAT(2F15. 7) ②
 *0080 600 FORMAT(1H, 2(A2, G15. 5, 5X)) ②
 *0090 610 FORMAT(1H, 4(A4, G15. 5, 5X)) ②
 *0100 STOP ②
 *0110 END ②
 *0120\$:QEXEC ②
 *0130 10.5, 3.72 ②
 *0140\$:ENDJOB ②
- ⑤ *0150 ②
- ⑥ *PRINT ②

入力
ファイル
作成

04/17/81 10:15:30

```
0020  $      JOB      X0001A0007, X0001, JA5QDE
0030  $      QFORT    NFORM
0040  READ(5, 500)A, B
:
:
0140  $      ENDJOB
```

作成
した
ファイル
の
内容
が
表示
される。

- ⑦ *SAVE JCL2 ②
 DATA SAVED-JCL2

説 明

- ① CARDINサブシステムに移行する。
- ② 行番号の自動発生を行う。(AUTOコマンドでなく, AUTOXを使うこと。また, ある種の端末では, AUTOXを使用して発生させた行番号の後ろに, 空白が表示されることもあるが, この空白は実際には入力されないので安心されたい。)
- ③ CARDINで使用するファイルの第一行には, このような「##行」(正式には「編集情報行」)という特別な行を入れた方がよい。この行の意味については, 後に説明する。よく理解できない場合は, この例をそっくりそのまま, まねればよい。
- ④ この行から先が, 実際にバッチの入力となる行である。\$ JOB文から, \$ ENDJOB文までが, 1つのジョブとして扱われる。
- ⑤ 送信 キー又は RETURN キー(以下②と書く)を押して, 行番号の自動発生を止めさせる。
- ⑥ 作成したファイルの内容を確認する。
- ⑦ 後の使用にそなえて, 作成したファイルを "JCL2" という名で保存する。

⑧ *RUN ②

SNUMB # H012T

⑨ *JMON ②

H012T READING RMT @ 10. 259

H012T EXECUTING @ 10. 259

...

H012T OUTPUT WAITING ID=K8. @ 10. 264

NORMAL TERMINATION

} 実行状況が表示される。

⑩ *JOUT ②

⑪ FUNCTION? PRINT ②

\$\$ H012T ENTERED ACOS-6 AT 10:15:33 ON 04/17/81 TSS/S 0-08-02

0001 \$ SNUMB H012T

0002 \$ COMMENT X0001A0007 TSS CARDIN

...

0021 \$ ENDJOB 00000140

TOTAL CARD COUNT THIS JOB=000030

* ACTY-01 \$ CARD #0014 FORTY 04/17/81 REAL MODE SW=210200000000

* NORMAL TERMINATION AT 005440 BA=000000200000 I=4060 SW=210200000000

* ACTY-02 \$ CARD #0018 GELOAD 04/17/81 REAL MODE SW=000000000000

* NORMAL TERMINATION AT 013505 BA=000000200000 I=5020 SW=000000000000

END OF \$\$

⑫ FUNCTION? LIST ②

ACTIVITY 1

REPORT CODES

\$\$

74

} LISTの出力。

⑬ FUNCTION? PRINT 74 ②

H012T 01 04-17-81 10. 261 LABEL PAGE 1

ACOS-6 R008 FORTRAN (02-15-80) OPTIONS: LSTIN, JIS, BIN

1 READ(5, 500)A, B 00000040

2 WRITE(6, 600) "A=", A, "B=", B 00000050

3 WRITE(6, 610) "A+B=", A+B, "A-B=", A-B, "A*B=", A*B, "A/B=", A/B 00000060

4 500 FORMAT(2F15. 7) 00000070

5 600 FORMAT(1H , 2(A2, G15. 5, 5X)) 00000080

6 610 FORMAT(1H , 4(A4, G15. 5, 5X)) 00000090

7 STOP 00000100

8 END 00000110

THERE WERE NO DIAGNOSTICS IN ABOVE COMPILATION

32K WORDS WERE USED FOR THIS COMPILATION

END OF 74

アクティビティ (翻訳) のレポート 74 の内容

⑭ FUNCTION? ACTIVITY_2 ②

⑮ FUNCTION? LIST ②

ACTIVITY 2

REPORT CODES

\$\$

06

⑯ FUNCTION? PRINT_06 ②

A = 10.500 B = 3.7200

A+B = 14.220 A-B = 6.7800 A*B = 39.060 A/B = 2.8226

} アクティビティ 2 (実行)
} のレポート 06 の内容

END OF 06

⑰ FUNCTION? RELE ②

⑱ *DONE ②

SYSTEM ?

説 明

- ⑨ 作成したファイルをカード形式に変換して、バッチ処理を起動する。(起動するだけで、終了を待たない点に注意。起動すると、すぐに端末からのコマンド入力が可能となる。) "H012T" は、このジョブにつけられた名前 (snumb 名という) である。snumb 名は、控えておいた方がよい。
- ⑩ ジョブの実行状況を見る。
- ⑪ ジョブの実行結果 (SYSOUT ファイル (→ 3.3.5)) を見るため、JOUT サブシステムに入る。
- ⑫ まず、ジョブ全体の実行報告を見る。このジョブが、2 つのアクティビティから成っており、各アクティビティとも "NORMAL TERMINATION" (正常終了) であることを確かめる。(波線の部分。)
- ⑬ アクティビティ 1 (翻訳) の、出力のレポートコードを知るため、"LIST" と入力する。
- ⑭ アクティビティ 1 のレポートコードは "74" だけなので、その内容を見る。
- ⑮ 次に、アクティビティ 2 (実行) の結果を見る準備をする。
- ⑯ アクティビティ 2 の出力のレポートコードを知る。
- ⑰ レポートコード "06" のレポートを見る。
- ⑱ 全部見終わったので、SYSOUT ファイルを計算機の中から消去する。同時に JOUT サブシステムを終える。
- ⑲ CARDIN サブシステムモードからシステム選択レベルに戻る。

3. 使 用 法

3.1 会話型リモートバッチ処理のためのサブシステム群

ACOSでは、会話型リモートバッチを行うために、種々のTS用サブシステム(コマンド)⁺が用意されている。
CARDINサブシステムは、会話型リモートバッチの中核となるサブシステムで、行番号エディタによるカード・イメージ・ファイルの作成と、バッチへの投入・起動を行う。

JOUTサブシステムは、CARDINによって起動されたジョブの実行結果(SYSOUTファイル(→3.3.5))を端末から見ることを可能にする。CARDINとJOUTの2つのサブシステムだけで、基本的な会話型リモートバッチの操作を行うことが可能である。

起動したバッチジョブの実行状況を端末から監視するには JMONI または JSTS コマンドを使う。実行を打ち切るには JABT コマンドを使う。

その他、PRINT, JPRINT, BPRINT*, SCAN, FDUMP*, JDAC*, JISJIS, JISBCD, BCDJIS などのサブシステム(コマンド)が用意されている。(注: *について、説明しないので参考資料[1]を参照されたい。)

なお、大型センターのFIBジョブのためのコマンドとの対応は、次のようになる。

大型センター (FACOM OS IV)	本センター (NEAC ACOS-6)
SUBMIT	CARDIN(RUN)
STATUS	JMONI, JSTS
OUTPUT	JOUT
CANCEL	JABT

3.2 コマンド一覧

会話型リモートバッチ処理に関連するコマンドの一覧を以下に掲げる。各コマンドの、典型的な使用の順序は、図1に示してある。(太字は、特に重要なコマンドである)

	コマンド名 (注1)	パラメータ (注1)	意 味	記述箇所 (注2)
	CARDIN	{ <u>N</u> EW <u>O</u> LD file 1 [;file 2]... <u>S</u> AME }	CARDINサブシステムのビルドモードに入る。 file 1, file 2, ...は、使用するファイルの名前。	
入力ファイルを作成するためのコマンド	LIST	{ 無指定 行番号の指定 ファイル名 詳しくは、3.4.2を参照。 }	指定されたファイル(ファイル名が無指定の場合は、カレントファイル)の内容を端末に表示する。 PRINTを使うと、タブ文字と行番号処理を行った後の、カード・イメージが表示される。	[3] 56 ----- [1] 7 [1] 12
	PRINT			
	応答のしかた	CARD FORMAT ?	<u>N</u> ORM 標準指定。	
			<u>A</u> SIS } 行番号は処理しない。 又は(⊗)	
			<u>M</u> OVE 行番号は後方に移す。	
			<u>S</u> TRIP 行番号は削除。	
		TAB CHARACTERS AND SETTINGS ?	(3.3.4 参照)	
	AUTOX	[<i>m</i>][, <i>n</i>]	行番号の自動発生。 <i>m</i> から始めて、 <i>n</i> ずつ増える。	[8] 55

+ 「サブシステム」と「コマンド」の明確な区別はない。"RUN"を除くすべてのコマンドは、システム選択レベル("SYSTEM?"と計算機から質問してくる)でも、ビルドモード(星印 "*" が出力される)でも使用できる。

	コマンド名	パラメータ	意味	記述箇所	
入力ファイルを作成するためのコマンド	<u>RESEQUENCE</u>	$[m][,n][\left\{\begin{matrix} X-Y \\ X- \\ -Y \end{matrix}\right\}]$	行番号のつけ直し。もとの行番号XからYを、 m から始めて n ずつ増える行番号に変える。	[3] 56	
	<u>ASEQ</u> ----- <u>RESEX</u>	$[m][,n]$	行番号をつける。ASEQは、行番号の後ろに、空白を1文字挿入する。	[3] 56	
	<u>DSEQ</u> ----- <u>STRIPX</u>	なし	行番号をはずす。DSEQは、行番号の次の空白を1文字消去する。	[8] 21	
	<u>SAVE</u> ----- <u>RESAVE</u>	ファイル名	カレントファイルの内容を、パーマネントファイルに保存する。	[3] 54	
	C	$\left\{\begin{matrix} i \\ i-j \end{matrix}\right\} / \text{str 1} / \text{str 2} / [\text{ALL}]$	カレントファイルの <i>i</i> 行(から <i>j</i> 行まで)の文字列 str 1 を(全て) str 2 で置き換える。	[8] 20	
	EDIT	EDITOR サブシステム。ファイルの細かい修正に向いている。		[7] 44	
	SE	Screen Editor。ファイルの修正が画面上で行える。		[8] 74	
ジョブの起動	RUN	$\left\{\begin{matrix} \text{無指定} \\ \text{file 1} [; \text{file 2}] \dots \end{matrix}\right\}$	指定されたファイルのタブ文字処理や行番号の処理等を行った後、SYSIN入力プログラムに渡して、バッチジョブを起動する。	[1] 7 [1] 12	
		応答	CARD FORMAT, ----- DISPOSITION ?	(PRINTと同じ。ただしⓈは不可。)	
			<u>W</u> AIT	ジョブの終了を待つ。	
			<u>J</u> OUT	JOUTシステムを使う。	
			<u>R</u> OUT(<u>R6</u>), <u>R</u> OUT(<u>R7</u>)	ジョブの結果を医歯薬地区(R6)又は教養部地区(R7)のラインプリンタから出力する。	
			Ⓢ	ジョブの結果は、センターLPから出力する。	
		TAB CHARACTERS AND SETTINGS ?	(3.3.4 参照)		
監視	<u>JMON</u> I JSTS	$\left\{\begin{matrix} * \\ \text{snumb 1} [; \text{snumb 2}] \dots \end{matrix}\right\}$	CARDINのRUNコマンドで起動したジョブの実行状況を見る。"*"を指定すると、最後に起動したジョブを見ることができる。	[1] 8 [1] 11	
打切	JABT	$\left\{\begin{matrix} * \\ \text{snumb} \end{matrix}\right\}$	指定されたジョブを、強制的に打ち切る。	[1] 10	
出力を見る	JOUT	$\left\{\begin{matrix} * \\ \text{snumb 1} [; \text{snumb 2}] \dots \end{matrix}\right\}$	ジョブの実行結果(SYSOUT)を、端末から見る。このコマンドを使うには、"RUN"の時に、計算機からたずねてくる質問に対し、"JOUT"と答えておく必要がある。	[1] 11 [1] 42	
	SCAN	ファイル名	バッチジョブが出力したファイルを見る。特定の文字列をさがしたい時とか、エラーのある行だけ見たい、という場合に使う。	[1] 11 [1] 27	

	コマンド名	パラメータ	意味	記述箇所	
ファイルの印字	JPRINT	ファイル名	TS向け JIS ファイルを、センターのラインプリンタから出力する。	[1] 11 [1] 20	
		応答	\$ IDENT ?		アカウント番号, 利用者登録番号 (ユーザー ID)
			RUNOFF FORMAT ?		N O と答えるのが普通
			LINE COUNT 55 ?		YES } 1 ページに 55 行印字。 (送)
					n 1 ページに n 行印字。
ファイルの形式変換	JISBCD	jisfile;bcdfile	JIS ファイルを BCD ファイルへ変換。	[1] 10 [1] 15	
		応答	\$ IDENT ?		アカウント番号, 利用者登録番号。
			LABELS ?		NORM 標準指定。
					ASIS 行番号は処理しない。
					MOVE 行番号は後方へ移す。
					STRIP 行番号は削除。
					str ₁ (i ₁ , j ₁)[;str ₂ (i ₂ , j ₂)] ... ファイルの行番号 i ₁ から j ₂ までの行に対し、 73~80カラムに、str ₁ という文字列と行番号とを書く。
			TAB CHARACTERS AND SETTINGS ?		(3.3.4 参照)
	BCDJIS	bcdfile;jisfile	BCD ファイルを JIS ファイルに変更。	[1] 10 [1] 16	
		応答	LINE NUMBERS ?		MOVE 73~80カラムを行番号とする。
					AUTO [m, n] 行番号自動発生。
					ASIS, [m] BCD レコードの n 桁を BCD から JIS に変換。
					(送) 行番号は、つけない。
			TAB CHARACTER AND SETTINGS ?		t, s ₁ , s ₂ , ..., s _n タブ文字として t, タブ位置として s ₁ , s ₂ , ..., s _n を使う。
		(送) タブは使わない。			
		JISJIS	batchfile;tsfile		バッチ向 JIS ファイルを TS 向 JIS ファイルへ変換。
	(応答のしかたは BCDJIS の場合と同じ)				
	JISJIS	tsfile;batchfile	TS 向 JIS ファイルをバッチ向 JIS ファイルへ変換。		
	(応答のしかたは BPRINT の場合と同じ)				

- 注 1 : 「」は省略形を表わす。例えば PRINT は、PRIN と入力してもよい。
「」はパラメータの省略可能なことを示す。
... は反復を示す。「」... は、そのパラメータをいくつかつづけて書けることを示す。
{ } は、どれかひとつを選択することを示す。
送 は、送信 キー又は (RETURN) キーのみによる応答を示す。
- 注 2 : [m] n は、参考資料 m の n ページ、ということを示す。

3.3 予 備 知 識 （ 3.3.4 以外は飛ばしてよい。）

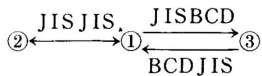
3.3.1 ファイルについて

ACOS-6 では、次の 3 種のファイルを区別して取り扱う。

- ① TS 向 JIS 形式ファイル
- ② バッチ向 JIS 形式ファイル
- ③ BCD 形式ファイル

利用者が、TS 端末で作成するファイルは①の形式をしている。一方（たとえ CARDIN で起動したとしても）、バッチジョブが直接に取り扱うファイルは、②か③の形式でなければならない。従って、ファイル形式の変換が必要になる。

ファイル形式の変換には、次の 3 つのコマンドが用意されている。

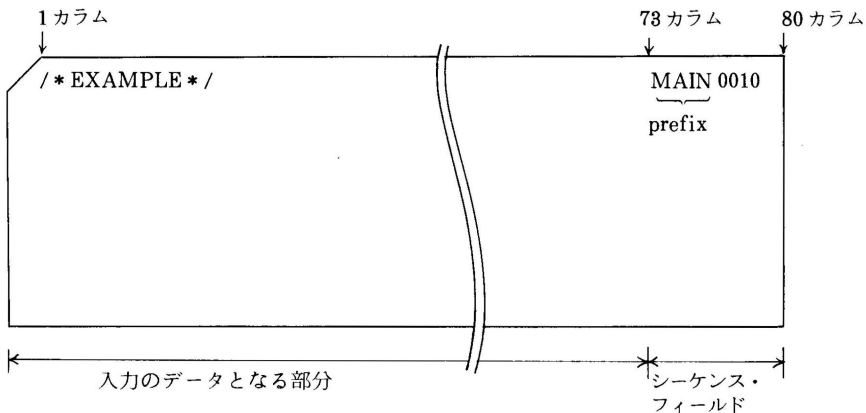


また、RUN コマンドも、暗に形式変換を行っている。そのため、①のファイルを会話型リモートバッチの入力データとして使いたい時は、\$SELECTA 文（附録 1 参照）を使って、入力ファイルと併合してから、RUN コマンドを打ち込むことにより、JIS JIS コマンドを使用せずに②→①の変換を行うことができる。

3.3.2 パンチカードの形式

バッチ処理の入力としては、伝統的にパンチカードが使われてきた。そのため会話型リモートバッチ用の各種コマンドも、パンチカードのことが強く意識されている。例えば、RUN コマンドでは、TS 向ファイルからパンチカード形式の BCD ファイルへの変換が行われていると考えられる。

パンチカードは、1 枚に 80 文字を書くことができるようになっている。そのうち、73～80 カラムは「シーケンス・フィールド」（sequence field）と呼ばれ、TS のファイルの行番号に相当する欄として扱われる。（つまり、計算機は普通、この欄を入力データとは扱わない。）シーケンス・フィールドには、普通、カードの並ぶべき順に増加する数を書く。また、任意の文字（prefix と言う）を、数の先頭につけることもできる。



3.3.3 SYSOUT （ラインプリンタ出力）の形式

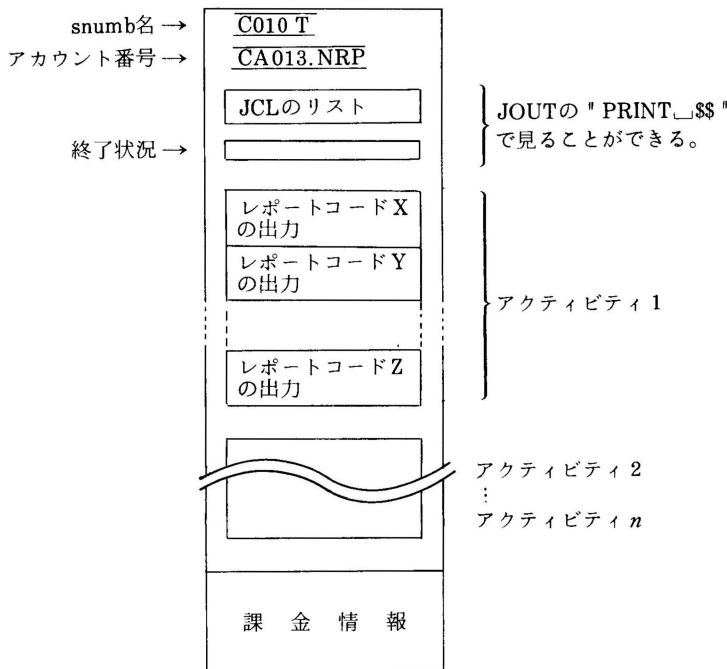
バッチジョブの標準出力の形式は、次の図のようになっている。

最初に出て来るのは、入力された JCL と、各アクティビティの終了状況（正常に終了したかどうか）である。この部分は、JOUT サブシステムでは、レポートコード "\$ \$" で参照される。（→ 3.4.5）

次に、各アクティビティ毎の実行結果が、実行の順に出力される。

各アクティビティの出力は、一般に、いくつかに分かれている。この分類された出力のそれぞれには、それらを区別するため、レポートコードという 2 文字の名前がつけられている。例えば、FORTRAN の翻訳という

アクティビティは、1種のレポートしか出力せず、そのレポートコードは"74"である。レポートコードがわからない時は、JOUTを使って"LIST"と入力すれば、使われているレポートコードが全部出力される。最後に、全体の課金情報が印字される。



3.3.4 タブ機能

タブとは、空白の入力を省く機能のことである。例えば、CARDINのRUNコマンドの入力用ファイルでJCLの部分では本来

```
0010$ JOB X 0001A 0007, ...
  ↑      ↑      ↑
 1カラム 8カラム 16カラム
```

と書くべきである。ところが、このように空白が連続していると、入力をまちがいがしやすい。そこで、ACOS-6では、このように書く代りに、

```
0010$:JOB:X0001A 0007, ...
```

と書くことを(ある条件下で)許している。このように入力すると、":"はいくつかの空白に置き変わり、":"の次の文字が、8カラムか16カラムになるよう調整される。この場合、":"を「タブ文字」、8カラムと16カラムを「タブ位置」という。

タブ文字には":"以外の文字も使用できるし、8、16カラム以外のタブ位置も設定できるようになっている。そこで、利用者は計算機に対し、自分がどのようなタブ文字とタブ位置を採用しているのかを知らせてやる必要がある。このことは、普通、「##行」(後述)によって行う。もし##がないと、ある種のコマンドを使う時に、計算機側から、次のように質問してくる。

TAB CHARACTERS AND SETTINGS?

これに対しては、次のように答える(いずれかひとつ選ぶ)。

$\left\{ \begin{array}{l} \text{NORM} \quad (\text{normal}) \text{ 標準タブ文字 (":") と, 標準タブ位置 (8, 16, 32, 73カラム) を使う場合。} \\ \text{②} \quad \text{タブを使わない場合。} \end{array} \right.$
 $t_1, s_{11}, s_{12}, \dots, s_{1n_1} [; t_2, s_{21}, s_{22}, \dots, s_{2n_2}] \dots$

(ただし, t_i は " # " を除く任意の 1 文字。 s_{ij} は, 数字。)

文字 t_i をタブ文字とし, そのタブ位置を s_{ij} とする。

例

% , 8, 16, 24 (" % " がタブ文字。)

% , 8, 16, 24 ; ? , 7 (複数タブの指定。)

タブを使用したファイルを端末で見るとは, LIST コマンドではなく, PRINT コマンドを使用した方が見やすい。

タブは, TS 向 JIS 形式ファイルにしか使用できない。

3.3.5 ACOS のバッチシステムについて (バッチ利用経験者は読み飛ばしてもよい)

ACOS のバッチシステムについて, 簡単に説明しておく。

用語の解説

ジョブ (Job) ……利用者がシステムに依頼する仕事の単位である。

アクティビティ (Activity) ……資源の割当の対象となる仕事の単位である。ジョブは, 複数のアクティビティで構成される。アクティビティの概念は, 大型センターの「ジョブステップ」に該当する。

snumb 名 ……ひとつのジョブを他のジョブから区別するためにつけられる, ジョブの名前のこと。通常, システムが自動的に割り当てる。

ファイル (File) ……関連するデータの集りをファイルという。大型センターの利用者には, 「データセット」として知られている。(大型センターで言うところの「ファイル」とは多少意味が異なる。)

SYSIN, SYSOUT …… バッチ処理では CR (カードリーダー) から読み込まれたデータの集合を SYSIN (ファイル), LP に書き出されるべきデータの集合を SYSOUT (ファイル) という。上記の事を行うプログラムのことも, それぞれ SYSIN (プログラム), SYSOUT (プログラム) という。

JCL (Job Control Language)

バッチ処理を行わせるには, SYSIN (つまり CR) に計算機に対する指示やデータを与えてやらなければならない。このうち, 仕事の大きな内容や, 実行の順番を知らせたり, 資源 (ファイル, メモリ, 紙や実行時間) の確保を指示したりするカード (行) は, JCL と呼ばれている。ACOS の JCL の代表的な例をあげると,

欄	1	8	16
	\$	JOB	X 0001A 0007, X 0001, JA 5QDE

この制御文は, 支払責任者番号が " X 0001 " , パスワードが " JA 5 QDE " , 課題登録番号が " X 0001A 0007 " であるジョブの始まりを示すものである。

欄	1	8
	\$	ENDJOB

この制御文は, ジョブの終わりを示す。\$JOB と \$ENDJOB の 2 つの制御文は, いかなるジョブ依頼にも不可欠である。

欄	1	8
	\$	QFORT

FORTRAN プログラムの翻訳を示す。

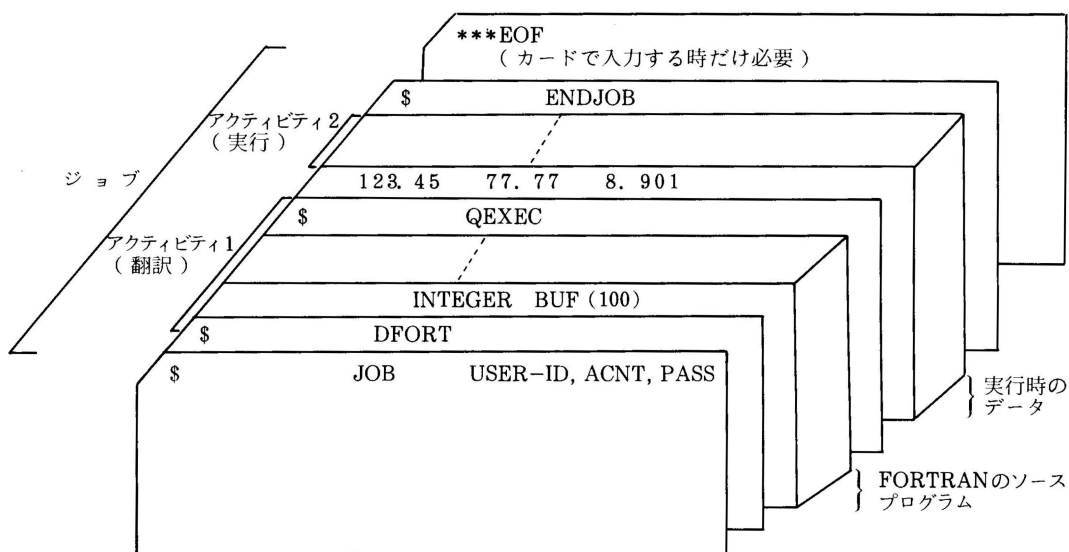
欄	1	8
	\$	QEXEC

翻訳されたFORTRANプログラムの実行を指示。

例でわかる通り、すべてのJCLは、1桁目に"\$"があり、8桁目からコントロール・フィールド("JOB", "ENDJOB"等を書く欄)が始まり、16桁目からオペランド・フィールドが始まる、と決まっている。この桁の指定は、1桁でも間違えると計算機は受け付けないので注意を要する。

よく使うJCLとしては、この他に、\$PRMFL文(永久ファイルの割当)、\$FILE文(作業用ファイルの割当)、\$GMAP文(アセンブラの呼出)等がある。詳しくは参考資料[5],[6]を参照されたい。また、会話型リモートバッチ専用の\$SELECTA文があるが、これについては付録Aを参照のこと。

上の4枚のジョブ制御文を使うと、FORTRANの翻訳と実行を行わせるためのバッチジョブを依頼するためのデック構成は、次のようになる。



```

graph TD
    Start([ログオン]) -- "SYSTEM? に対し" --> NewCardin["CARDIN NEW"]
    Start -- "SYSTEM? に対し" --> OldCardin["CARDIN OLD ファイル名"]
    NewCardin -- "ファイルの作成" --> Edit["AUTO, LIST, ASEQ, RESEQ, DSEQ, C, PRINT コマンド等により,  
ファイルの作成・修正を行う。  
エディタサブシステム (EDIT コマンド), スクリーンエディタサブシステム (SE) 等も使用可。"]
    OldCardin -- "古いファイルの修正" --> Edit
    Edit -- "(必要な場合)" --> PrintTest["PRINT コマンドによって, タブ文字がうまく機能するか, 確かめる。"]
    Edit --> Run["RUN"]
    PrintTest --> Run
    Run -- "(ジョブの終了を待つ時)" --> Jmon["JMON * と入力。  
終了を待つ。"]
    Run -- "(ジョブの終了を待たない時)" --> Snumb["snumb 名を控える。"]
    Jmon --> Jout["JOUT * と入力。  
ジョブの実行結果を見る。  
必要なら, LPから出力。  
(図2を見よ。)" ]
    Jout -- "修正" --> Edit
    Jout --> Done["DONE"]
    Jout --> Bye1["BYE"]
    Done --> Work1["他の仕事をする。"]
    Bye1 --> Work1
    Snumb --> Bye2["BYE"]
    Bye2 -- "しばらくして" --> Logon2["ログオンして  
JSTS snumb 名を入力し,  
ジョブが終了したかどうか見る。"]
    Logon2 -- "ジョブ終了" --> Work2["ジョブ終了"]
    Logon2 -- "ジョブはまだ終了していない。" --> Snumb
    
```

- 25 -

3.4 各コマンドの説明

以下に会話型リモートバッチ処理に関係があり、よく使われるコマンドについて説明する。都合により、一部の機能の説明を行っていない場合があるが、これについては、他の文献を参照されたい。説明の順番は、使用順である。

3.4.1 CARDIN

CARDIN コマンドは、CARDIN サブシステムへの移行を行う。行番号エディタによるファイルの作成や修正を行う前や、次に述べる "RUN" を入力する前に、必ず CARDIN サブシステムのモードに移行しておく必要がある。

使用例

```
SYSTEM ? CARD NEW      新規にファイルを作成
SYSTEM ? CARD OLD FILE1 過去に作成・保存した "FILE 1" という名のファイルを使用する
                        場合。
```

3.4.2 PRINTとRUN

PRINT

PRINTは、LIST コマンドと同様、ファイルの内容を端末に表示するために用いる。LISTとの違いは、タブ文字の処理（タブ文字を空白で置き換える）を行った後の画面を表示することにある。

PRINT コマンドのパラメータの指定は、LISTと同様である。例えば、

```
PRINT   カレントファイルを表示。
PRINT   10, 100-200   カレントファイルの行番号10と100～200の行を表示。
PRINT FILE 1         "FILE 1" という名のファイルを表示。
PRINT FILE 1 (10, 100-200)  FILE 1 の指定された行を表示。
PRINT FILE 1; FILE 2  FILE 1 と FILE 2 を表示。
```

RUN （CARDIN サブシステムのビルドモードで使う事。）

RUNは、指定されたファイルのタブ処理や行番号の処理を行って、パンチカード形式のファイル（カードイメージファイル）を作り出し、それをバッチ処理の入力として、バッチジョブを起動する。

RUNのパラメータとしては、ファイル名を指定する。例えば、

```
RUN           カレントファイルを指定。
RUN FILE 1     FILE 1 を指定。
RUN FILE 1; FILE 2  FILE 1 と FILE 2 とを併合した後、カードイメージに変換し、バッチジョブを起動。
```

PRINTとLISTの違い

下は、同じファイルの内容をPRINとLISTで見た場合の違いを示している。

LISTの場合

```
010 ##N, J
020 THIS: IS: A EXAMPLE OF: TABULATION.
030 THIS
040 : IS
050 :: ANOTHER
060 ::: EXAMPLE.
```

←「##行」はPRINTでは表示されない。

↑タブ文字

PRINTの場合

```
04/ 11/ 81      9:58:02
020   THIS   IS   A EXAMPLE OF   TABULATION.
030   THIS
040           IS
050           ANOTHER
060           EXAMPLE.
```

↑タブ文字はいくつかの空白に展開される。

計算機からの質問とその答え方

RUNおよびPRINTコマンドを入力すると、ファイルの第1行に##行(後述)がない場合、次の質問がなされる。

CARD FORMAT DISPOSITION? (RUNの時)

又は、

CARD FORMAT? (PRINTの時)

この質問に対しては、次の①のひとつと、②のいくつかを、この順に、コンマで区切って答えればよい。

例えば、

- N, J 最も典型的な答え。
- S, W 行番号削除。ジョブ終了を待つ。
- N(%) タブ文字は"%"。結果はLPに出力。

のように答える。

① CARD FORMAT? に対する答え。行番号の取り扱い方を指定する。

ASIS (as is ...そのまま) 行番号に対する処理を行もしない。この指定は普通、行番号なしのファイルを使用する際に行う。

STRIP 行番号を削除する。

MOVE 行番号を73~80カラムに移動する。

NNORM (normal) 行番号を73~80カラムに移動するのに加え、標準タブ文字(":")と標準タブ位置(8, 16, 32, 73カラム)を使用する。最も一般的な答えである。

NNORM(t) tは"#"を除く任意の1文字。タブ文字としてtを用いることを除いて、上記と同じ。

② PRINTの場合、ASISの指定と同じ。RUNでは、処理を中止する。

② DISPOSITION? に対する答え。PRINTの場合は答えなくてよい。ジョブの出力先の指定等を行う。

WAIT ジョブが終了するまで、端末からの入力を禁止する。この指定がないと、ジョブ起動後ビルドモードに戻り、端末からの入力が可能となる。

JOUT^{注)} ジョブの出力を、後でJOUTサブシステムを使って、見ることを指定。この指定があると、ジョブが終了しても、結果は印刷出力されず、計算機内に残る。

R₆OUT^{注)} (R6), R₇OUT (R7)

ジョブの出力を、医歯薬地区(R6)又は教養部地区(R7)のリモートプリンタに向ける。

この指定があると、ジョブ終了後、結果は自動的に、決められたプリンタから出力される。

③のみ ジョブの結果は、終了後、センターのラインプリンタから出力される。

注) これらの指定は、WAITと組み合わせて指定できる。(コンマで区切る事)

以上の応答後、もし①で"NORM"を指定しなかった場合は、タブについての質問、

TAB CHARACTERS AND SETTINGS?

がなされる。これについては、3.3.4節で述べたように応答すればよい。

全部の質問に答え終わると、PRINTコマンドでは、ファイルの内容が表示される。また、RUNコマンドでは、起動したジョブのsnumb名(識別名)が表示される。snumb名は、以後使うことがあるので、控えておくといよい。

行

RUNやPRINTコマンドを使用するたびに、上記のような質問に答えるのは、やっかいである。そこで予想される質問に対する答えを、予めファイル中に書いておくことができるようになっている。それにはファイルの第1行に、"##"で始まる次のような文を書けばよい。

0010##NORM, JOUT (行番号つきファイルの例)
 "CARD FORMAT, DISPOSITION" に対する応答
 ##A, J, %, 8, 16, 24, 32, 40 (行番号なしファイルの例)
 "TAB CHARACTERS AND SETTINGS?" に対する答え
 必ずひとつだけ空白を入れる。

もし、ファイルの第1行に、このような、"##"で始まる行があると、システムは前述の質問をしない。
 必要ならば、ひとつのファイルに、複数の##行を入れることができる。(詳しくは参考資料[1]の21
 ページ参照。)

備考

もし、①で" NORM "以外の応答をし、" TAB CHARACTERS AND SETTING ? " に対して②
 のみの応答をしたい(つまりタブを使用したくない)という時には、下のように書く事。

0010 ##M, J, ファイル作成時に空白を1文字だけ入れる。
 この空白がないと、" TAB CHARACTERS AND SETTINGS ? " の質問がなされる。また2個以上の
 空白を入れると、空白をタブ文字と扱うので注意。

3.4.3 JMONIとJSTS

JMONI (job monitor) は、CARDINの RUN コマンドで投入されたジョブの実行状況の監視を行う。入
 力の形式は、次の例の通り。

JMONI * 最後に投入したジョブを見る。

JMONI B 034 T; * B 034 T という snumb 名のジョブと、最後に投入したジョブとを見る。

JMONIは、いったんコマンドを入力すると、監視しているすべてのジョブが終了するまで監視を続ける
 ため、他のコマンドが入力できなくなる。従って、もう見たくないという時は、" BREAK " キーを使用する
 必要がある。これに対し、JSTS (job status) コマンドは、コマンドを入力した時点での実行状況を表示し
 て、直ちにビルドモードに戻る。JSTSのパラメータは、JMONIと同じである。

JMONIおよびJSTSは、下のようなメッセージを出力する。

STATUS CHANGING : リモート入力中。

READING-CR : カードSYSINより読み込み中。

" -RMT : リモート入力ファイルより読み込み中。

WAIT-ALOC : スケジューラへの登録待ち。

" - PERIP : 周辺装置の割り当て待ち。

" - CORE : 主記憶の割り当て待ち。

IN HOLD : 特権命令を使用するため、オペレータからの許可を待っている。

IN LIMBO : 使用する資源が大き過ぎるため、オペレータからの許可を待っている。

EXECUTING : 実行中。

SWAPPED : スワップされている(一時的に、主記憶から追い出されている)。

WAITING-MEDIA : 磁気テープまたはディスクのかけかえ要求中。

TOO BIG : 資源の要求量が大きすぎる。

IN RESTART : リスタートされたジョブである。

TERMINATING : 終了処理中。

OUTPUT WAITING : 出力待ち。

" STARTED : 出力中。

" COMPLETE : 出力終了。

JOB NOT ACCESSIBLE : ジョブをアクセスできない。

SYSOUT NOT IN MEMORY : SYSOUTがスワップされているため、実行終了が確認できない。
 (しばらく待てばよい。)

3.4.4 JABT

JABT (job abort) は、ジョブを異常終了させる。アボートコードは " X 2 " である。

使用例

JABT * : 最後に投入したジョブを終了させる。

JABT S 011 T : S 011 T という snumb 名のジョブを終了させる。

3.4.5 JOUT

JOUT (job output) は、指定されたジョブの SYSOUT ファイル (LP から出力されるべきデータ) を端末から見たり、LP に出力させたりするためのサブシステムである。指定できるジョブは、次のいずれかである。

- (1) CARDIN で投入され、かつ " DISPOSITION ? " の質問に対し " JOUT " を応答したジョブ。
- (2) リモートバッチ又はセンタで投入され、\$REMOTE 制御文で指定があるジョブ。

使用例 (システム選択レベル又はビルドモード時に入力可)

JOUT * : 最後に投入したジョブを見る。

JOUT S 001 T : S 001 T という snumb 名のジョブを見る。

これに対して、正常な場合、

FUNCTION ?

というメッセージが返ってくる。

エラーメッセージには次のようなものがある。(詳細は参考資料 [1] を参照)

OUTPUT IS BUSY : ジョブの結果が、ラインプリンタあるいはどこかの TS 端末に出力中である。

OUTPUT NOT FOUND : 指定ジョブが存在しない。

NOT YOUR JOB : 端末接続時の課題登録番号とジョブの課題登録番号 (\$JOB 文で指定した課題登録番号) が一致していない。

BATCH SYSTEM FULL--TRY LATER SORRY
PAT SPACE FULL--TRY LATER

: 両方とも、しばらく待ってから、再度コマンドを入力すればよい。

もし、指定したジョブが終了してない場合は、JMONI と同じメッセージが出る。

FUNCTION ? に対する応答

JOUT コマンドの入力に成功し、" FUNCTION ? " という質問が返ってくると、次のコマンド (そしてこれらのみ) の入力が可能となる。

ACTIVITY *n* (*n* は自然数)

アクティビティ *n* を読み出す状態にする。

LIST

現在表示しようとしているアクティビティ (ACTI *n* で指定する。初期状態は、アクティビティ 1) に関するレポートコードを出力する。

PRINT rc 又は EPRINT rc (rc は、2 文字のレポートコード)

現在表示しようとしているアクティビティのレポートコード rc の内容を端末に表示する。

PRINT は、連続する空白を詰めて表示するのに対し、EPRINT は、そのまま表示する。

普通の、1 行 80 文字の端末では、PRINT を使用した方が、見やすいであろう。EPRINT は、プリンタ付の端末や、高密度のグラフィック端末等で、1 行あたり 136 字以上表示できる端末に対してのみ用いるのが適当である。この場合、LENGTH コマンドを実行しておく必要がある。

SCAN rc

レポートコードrcのジョブ出力を検索する(例えば、エラーメッセージの部分だけ表示させたりする)。このコマンド入力後、計算機は"FORM?"と質問してくるので、SCANサブシステム(→3.3.6)と同じように答える。ただし、

(1) BATCH, REM, BYE, LENGの応答はできない。

(2) メモリダンプ形式の出力走査ができる。("FORM?"に対し"DUMP"と答える)という違いがある。

LENGTH n ($8 \leq n \leq 160$)

1行あたりの出力文字数を n 文字にする。初期値は $n=76$ 。

JOUTから抜け出すには、次のいずれかを答える必要がある。

DIRECT $\left\{ \begin{array}{l} \text{ONL} \\ \text{R } 6 \\ \text{R } 7 \end{array} \right\}$

SYSOUTファイルの内容をセンターのLP(ONLの場合)、医歯薬地区(R6)、教養部地区(R7)のリモートプリンタから、それぞれ出力する。

RELEASE

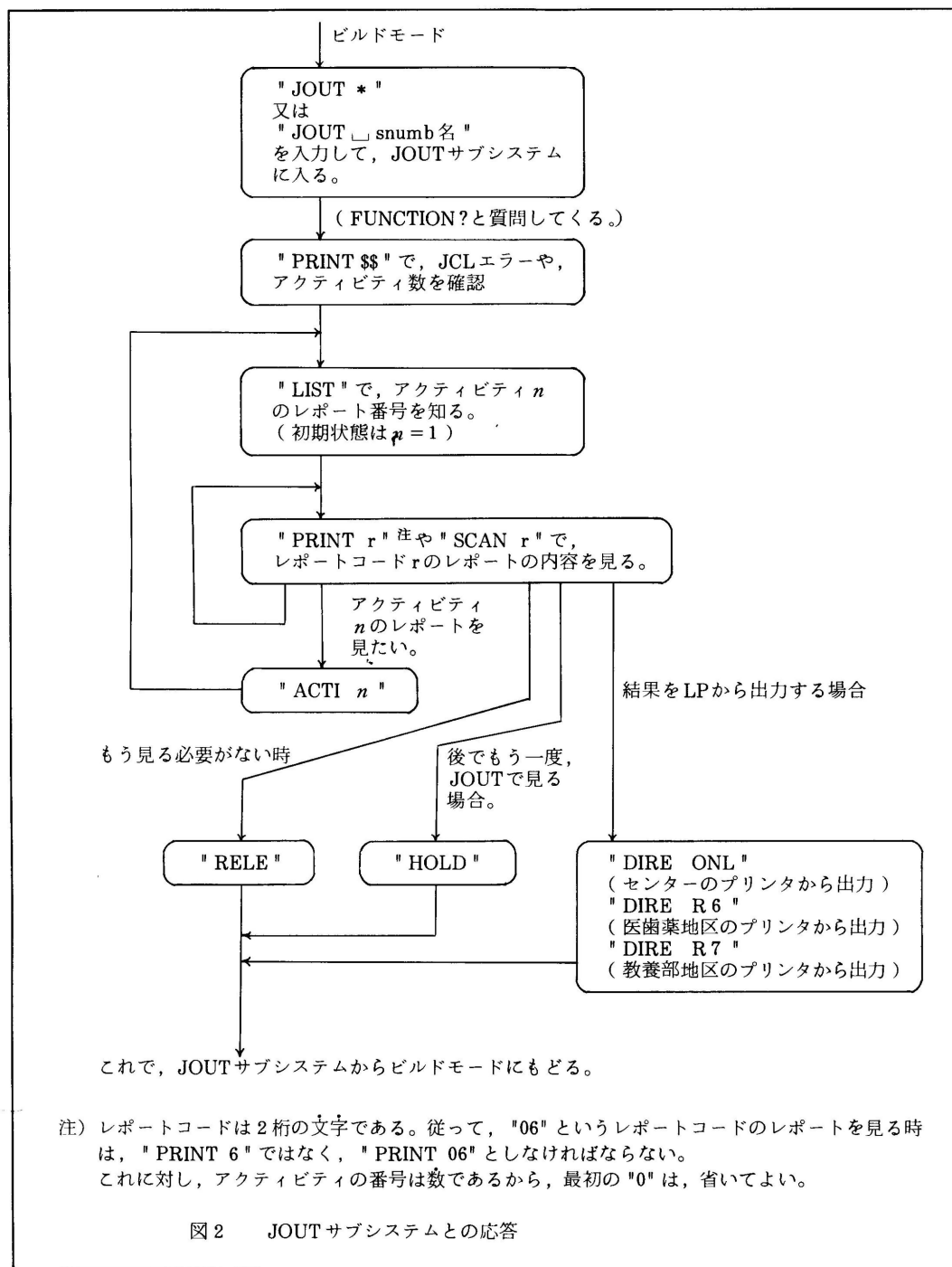
SYSOUTファイルの内容を消す。

HOLD

SYSOUTファイルをそのまま保存しておく。

このように答えた場合、その日のうちにJOUTを再び使用して、消すか、出力するかしなければならぬ。

(備考) SYSOUTファイルを、パーマネントファイルに保存したり、ファイルの一部だけを出力したりすることはできない。



3.4.6 SCAN

SCANは、バッチジョブのパーマネントファイル上の出力（これを行うには、\$PRMFL制御文を使う）を検索するためのサブシステムである。この機能を使うには、まず、システム選択レベル又はビルドモードで、

SCAN ファイル名

と答える。すると"FORM?"という質問がなされる。

FORM?に対する応答は、次のいずれかである。

GMAP : GMAPアセンブラのリスト出力を扱う場合。

FORT : FORTRANコンパイラの //

COBOL : COBOLコンパイラの //

LOAD : ローダ //

USER : 上記以外の出力を扱う場合。

USERの応答に対しては、" CODE ? " という質問がなされるので、⑤のみで応答するか
1～5文字の文字列（ラインコードという）を応答する。もしラインコードの応答を行う
と、以後の検索は、行の2～6カラムがそのラインコードに一致する行についてのみなさ
れる。

つづいて、" EDIT ? " という質問がなされる。

EDIT?に対する応答は、次のいずれかである。以後の出力形式を指定する。

Y 又は YES 連続する空白を詰めて表示。印字行番号（システムが付加する6桁の数字）を付加。

N又はNO又は⑤ 空白はそのまま表示。印字行番号を付加。

Y*又は*Y 空白は詰める。印字行番号は付加しない。

N*又は*N 空白はそのまま。印字行番号はそのまま。

つづいて疑問符"? "が出力されて、入力状態になる。

この段階になって初めて出力検索が行えるようになる。ここでは、「ポインタ」（pointer）という概念が必要になる。利用者が、ある瞬間に見ることのできる部分は、指定されたファイルのどこかの1行に限られている。ポインタとは、その1行を指している針のようなものである。ポインタは、ある種のコマンドの実行によって（ある場合はあらかじめ、ある場合は暗黙のうちに）動かされる。ポインタの指している行を「カレント行」と呼ぶ。

"?"に対する応答は、次のいずれかである。

PRINT [{ $\begin{matrix} n \\ \text{ALL} \end{matrix}$ }] （ n は自然数）

このコマンドは、カレント行から n 行（省略時は1行、ALL指定時は、最後の行まで）
表示する。もしラインコードを用いているならば、一致する行のみ表示される。
ポインタは動かない。

FIND : /str/[: n] （ n は自然数、strは文字列）

カレント行から始めて、strと同じ文字列を含む行の n 番目の行にポインタを位置づける。
もしstr中に、"/ "を含んでいる場合は、次の例のようにする。

例： FIND : /CALL/ 文字列 " CALL " をさがす。

FIND : " A+B/C " 文字列 " A+B/C " をさがす。

SPACE [n] ポインタを n 行（省略時 $n=1$ ）進める。ファイルの終わりを越えると、メッセ
ージを出力して、ファイルの先頭に位置づける。

BACK [n] ポインタを n 行戻す。 n を指定しないと、ファイルの先頭まで戻る。

LINE n ポインタを印字行番号 n の行に位置づける。

ERROR [n] エラーメッセージのある行を n 行（省略時は全行）表示する。

DONE SCANを終了する。

その他、UNDE, FLAG, LOAD, CODE等のコマンドがあるが、GMAP（アセンブラ）の利用者にしか関
係しないので、説明は省く。

附録1 \$SELECTA 制御文（参考資料[1]の21ページ参照）

\$SELECTA 制御文は、会話型リモートバッチでのみ使える JCL である。この制御文は、この制御文の現れた行をこの制御文のパラメータとして指定されたファイルと置き換える事を指示する。例えば、FORTRAN のプログラム

```
10 READ (5, 500) A, B
20 WRITE (6, 600) " A= ", A, " B= ", B
30 WRITE (6, 610) " A+B= ", A+B, " A-B= ", A-B, " A*B= ", A*B, " A/B= ", A/B
40 500 FORMAT (2F 15.7)
50 600 FORMAT (1 H_, 2( A 2, G 15, 5, 5 X ))
60 610 FORMAT (1 H_, 4( A 4, G 15, 5, 5 X ))
70 STOP
80 END
```

を " FTPROG 1 " という名のファイルに保存し、入力データ

```
10 10.5, 3.72
```

も、" FTDATA 1 " という名のファイルに保存しておいたとする。この時、次のファイルを CARDIN の RUN コマンドの入力ファイルとして用いると、\$SELECTA の部分は（バッチに渡される前に）上の二つのファイルに置き換えられるので、2章の使用例と同じことになる。

```
0010 ##N, J (注1)
0020 $: JOB: X 0001 A 0007, X 0001, JA 5 QDE
0030 $: QFORT: NFORM (注2)
0040 $: SELECTA: FTPROG 1 —————→ FTPROG 1 の内容と置きかわる。
0050 $: QEXEC
0060 $: SELECTA: FTDATA 1 —————→ FTDATA 1 の内容と置きかわる。
0070 $: ENDJOB
```

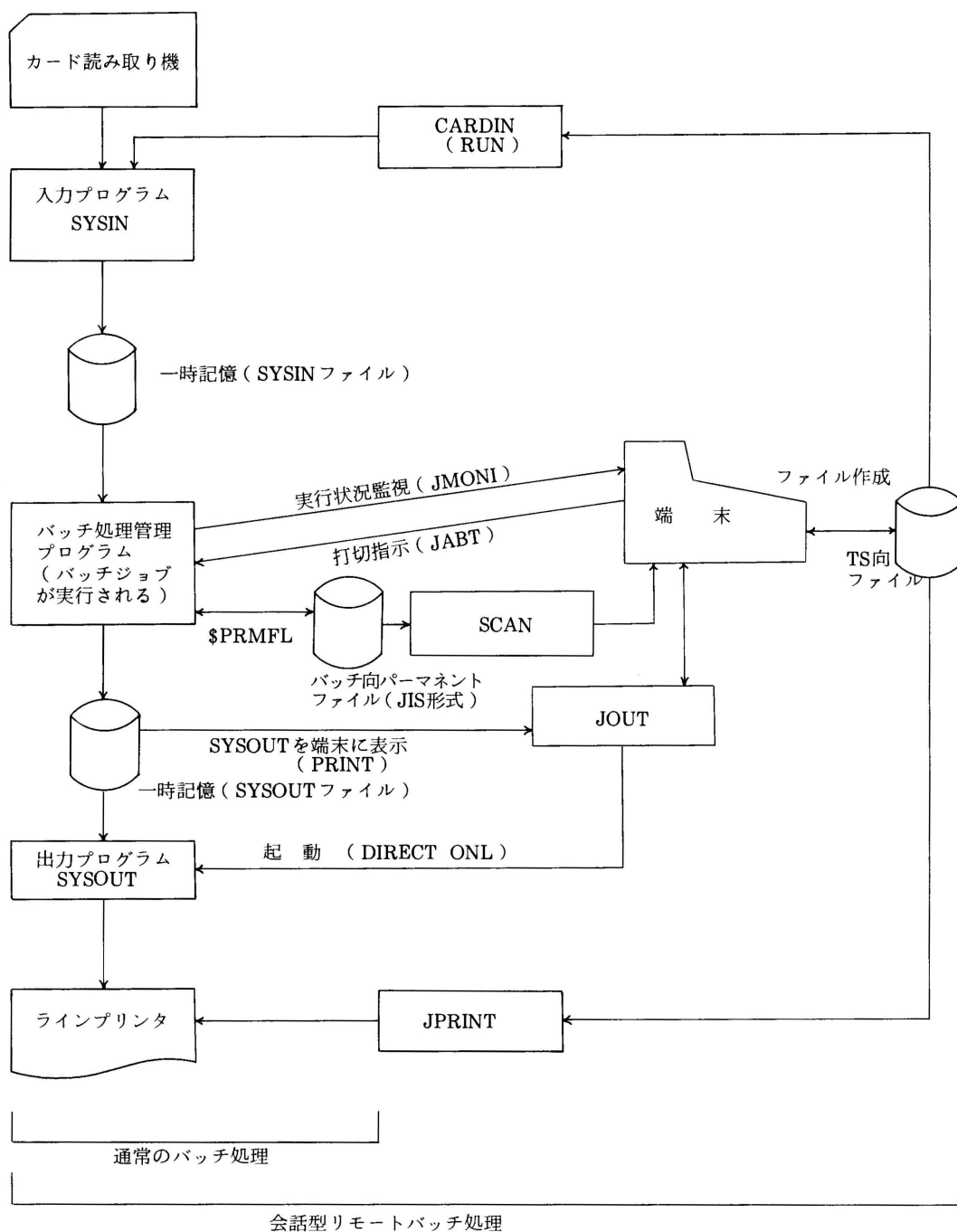
上のようなファイルを作って保存しておけば、他の FORTRAN プログラムやデータを扱いたい場合には \$SELECT 文のパラメータだけ書き換えるだけで、わざわざ JCL を作る必要がなくなるので便利である。

（注1）\$SELECTA 文で呼び出されたファイル（上の例では、FTPROG 1 と FTDATA 1）の方にも「##行」が存在してよい。その場合、その情報は、そのファイルからの入力が行われている間のみ有効である。

（注2）\$SELECTA 以外の JCL では、パラメータにファイル名を書く場合、利用者登録番号（userid）を含めた形で書く必要がある。上の例では、" FTPROG 1 " ではなく、" X 0001 A 0007 /FTPROG 1 " と書く。

（注3）\$SELECTA 文で呼び出されたファイルの中で \$SELECTA 文を使用してもよい。（7重の使用まで許されている。）

附録2 バッチ処理と会話型リモートバッチ処理との関係



参 考 資 料

- [1] 日本電気：会話型リモートバッチ説明書，FEF 06-3
本説明は，これに基いている。
- [2] 日本電気：タイムシェアリングシステム説明書，FEF 02-4
- [3] 広報 Vol. 1, No. 1 「タイムシェアリングシステム利用法」（p 40-68）
一般的な，TS処理についての説明。
- [4] 日本電気：システム管理説明書，FDA 01-5
- [5] 日本電気：ジョブ制御言語説明書，FDA 02-6
- [6] 広報 Vol. 1, No. 1 「バッチシステム利用法」（p 69-91）
バッチジョブについての説明。
- [7] 広報 Vol. 1, No. 2
- [8] タイムシェアリング・システム利用法 第2版
最新のTSS利用法である。