

HITAC 5020 ソフトウェアシステム(3)

HISAP 5020

酒井博敬，原田昌孝，三橋荘一郎

まえがき

HITAC 5020 のアセンブリーシステム HISAP 5020 は HARP 5020 と共にモニタシステムを構成するものであるが，その擬似命令は機能上(1)一般的な擬似命令，(2)入出力制御に関する擬似命令，(3)プログラムの手直しのための擬似命令の3種に分類される．一般的な擬似命令としてはデータの変換，プログラマズマクロ命令，サブルーチンの入口と出口の作製，サブルーチンの呼出し列の作製などに関する命令のほか，異なる言語で書かれたプログラムやライブラリールーチンの結合，チェインジョブのための擬似命令などが重視されている．入出力制御に関する擬似命令としてはデータ記述，ファイル記述，入出力マクロ命令，内部データ処理のためのマクロ命令などから構成され，またチェックポイントと再運転の処理機能をもつ．プログラムの手直しのための擬似命令としてはプログラムのテストランで誤りを発見するためのものと，原始プログラムを修正するためのものからなる．本文では特に(1)および(2)の擬似命令の主な機能と処理方法について略述する．

§ 1 命令様式について

HISAP 5020 の命令語は機械命令，擬似命令に共通な次の様式で表現する．

制御欄					
	場所欄	操作欄	アドレス欄	ID	通し番号
1	←10ケタ	←7ケタ	←54ケタ	←3ケタ	←5ケタ

制御欄はモニタの制御命令であるか，HISAP 5020 の命令語であるか，または注釈(comment)であるかの区別を指定する．場所欄は一般に命令語の格納場所を表わす記号を書くが，付値に関する命令における左辺の変数を表わす

記号を書くこともある。操作欄には機械命令および擬似命令の暗記用コードやプログラムの定義したマクロ命令のコードを書く。アドレス欄には命令に必要な変数をコンマで区切って並べる。アドレス欄にはこれらの変数の後に続けて注釈を書くことができる。ID欄は記号変換の際は全く無視されるもので何を書いてもよい。最後の通し番号欄には命令語の一貫番号を書く。原始プログラムは card to tape 操作の際この番号順に並べ変えられるほか、プログラム修正用の擬似命令でもこの番号が使用される。

§ 2 記号と数値の取扱いについて

2.1 記号と記号式

記号とは 10 字以下の英数字からなる連系であるが、記号変換の都合上次の 8 種類に分けて扱う。

- (1) 記憶場所を表わす記号：場所記号
- (2) 絶対的な値をもつ記号：絶対記号
- (3) プログラマズマクロの命令コードを表わす記号
- (4) セグメントを表わす記号
- (5) デックを表わす記号
- (6) 仕事を表わす記号
- (7) ファイルを表わす記号
- (8) その他：入出力区域およびその副項目を表わす記号など

この中 (1)~(4) について記号の扱い方を説明する。

(1) 場所記号：プログラム中の語、字、ビットなどのアドレスを表わす記号で、記号変換の段階でセグメントの原点に関して相対的な値が定義され、ローダによつて記憶装置の割付けが行われたとき絶対アドレスが定められる。この記号は内部記号表に場所カウンタの値と共に記録する。

(2) 絶対記号：付値命令 `BEQUAL` によつて絶対的な値を与えられる記号でやはり内部記号表に記録する。

場所記号と絶対記号には擬似命令 `HEAD` によつて 1 字の頭文字を付け、局所的な意味をもたせることができる。ただし 10 字からなる記号は `HEAD` の影響を受けない。頭文字は記号と結合した形で内部記号表に記録する。

内部記号表では各記号表の値を21ビットの2進整数で表わす。記号が記憶場所のアドレスを表わす場合は21ビットの中、上位16ビットが語アドレスに対応し、下位5ビットが1語の中の字またはビット位置に対応する。記号の値としてはその記号が語アドレスを扱う命令で用いられたときは上位16ビットが、また字またはビットアドレスを扱う命令で使用されたときは21ビットがとり出されて参照される。

字アドレスやビットアドレスの表記法としては一般に

$$x)y)z$$

という形を用いる。ここで x は語アドレスを示すもので記号または整数、 y と z はそれぞれ字アドレスとビットアドレスを示すもので絶対記号または整数である。この値は21ビットの2進整数として

$$x)y)z = x \times 2^5 + y \times 6 + z$$

によつて定められる。(1字は6ビット、1語は32ビットで構成される。)

特に $x)0)0$, $x)y)0$, $x)0)z$ の省略形としてそれぞれ x , $x)y$, $x)z$ を使用する。 $x)y)z$ の形のもの、または記号や整数を乗算記号 $*$ で結合したものを要素とし、これらの要素を加減算の記号 $+$, $-$ で結合したものを記号式といひ、命令語のアドレス欄に書くことができる。

(3) プログラマズマクロの命令コードを表わす記号：これは擬似命令MACROによつて定義されるマクロ命令のコードを表わすもので、マクロ命令コード表に形式パラメータと共に記録される。

(4) セグメントを表わす記号：セグメントとは§3で説明されるように記憶装置割付けの際まとめて扱われる最小単位であつて、他のデックから参照可能な外部名をもつ。セグメント記号は内部記号表とセグメント表の2ヶ所に記録される。内部記号表に記録される場合は相対値0をもつ場所記号と同様に扱われ、擬似命令HEADによる影響を受ける。セグメント表に登録するのは別のデックからセグメント記号を外部名として参照するためである。

このほか(5)のデックを表わす記号、(6)の仕事を表わす記号は適当な制御命令によつてモニタの制御表に記録されるものであり、(7)のファイルを表わす記号、(8)の入出力区域やその副項目を表わす記号などはFCP制御表に記録される。

2.2 数の表現

内部で2進表示の形に変換されるデータとしての数はALGOLで定義されるnumberを加減乗除の記号+, -, *, /で結合したもの(これをnumeric expressionとよぶ。)の前に, それに変換されるべき形成を指定する指定詞を付けて表現する. 指定詞としては次のようなものが使われる.

(1) 整数形 I/ または IO/ (8進) または IB/ (2進)

(2) 固定小数点表示の数

a. 符号つきの場合 S(n)/ または SO(n)/ (8進) または SB(n)/ (2進)

b. 符号なしの場合 P(n)/ または PO(n)/ (8進) または PB(n)/ (2進)

ただし a は符号ビットをもつ一般の形の場合で, b は符号ビットを符号と考えず数値ビットとして扱う場合である. また n は小数点の位置を指定するための整数で, 小数点の前のビット数(符号ビットを除く)を示す. 特に n=0 ならば(n)は省略してよい.

(3) 浮動小数点表示の数: F/ または FO/ (8進) または FB/ (2進)

指定詞の次には一般に numeric expression をコンマで区切って任意個並べてよい. 即ち指定詞は別の指定詞が現われるまで有効である.

データとしての数が内部で何ビットの長さに変換されるかは, データ変換のための擬似命令によつて区別する. たとえば命令 DATA では1語に, DATAD では2語に, VDATA では任意ビットの長さを単位とする内部形成に変換される.

§3 プログラムの分割制御

プログラムは次の3段階の単位に分割され, 作業の各処理過程で各々が或程度互に他と独立に取扱われる.

セグメント(segment) < デック(deck) < 仕事(job)

計算機で処理されるあるまとまつた作業の単位を仕事という. 1つの仕事のプログラムはいくつかのデックから構成され, 更にデックはいくつかのセグメントから構成される.

(1) セグメント：セグメントとは次のようなものである。

a. 命令またはデータのある区域 (area) である。

b. 次のいずれかの方法で定義される。

(i) 擬似命令 SEG と ENDS で囲まれた区域

(ii) 擬似命令 SEGV と ENDS で囲まれた区域

(iii) 擬似命令 COMMON で指定された区域，これを特に共通セグメントという。

(iv) (i)~(iii) で定義されたセグメントの間にある区域

(v) ライブラリルーチンはセグメントである。これを特にライブラリーセグメントという。

c. 上の (iv) の場合を除くセグメントは全て外部名をもつ。(iv) で定義されるセグメントは外部名をもたない。外部名はデックの外部から参照可能な名前のことである。上の (i)~(iii) で定義されるセグメントでは命令 SEG, SEGV または COMMON の場所欄に書かれた記号が外部名となり，またライブラリーセグメントはそれぞれ固有な外部名をもつ。(i)~(iii) の場合のセグメントの外部名は内部記号表に記録されるほか，セグメントに関する他の情報と共にセグメント表にも記録される。

d. セグメント内部の場所欄に書かれた記号には，セグメントの先頭からの相対アドレスが割当てられる。

(2) デック：デックとは次のようなものである。

a. デックはいくつかのセグメントの集まりであつて，原始プログラムの翻訳，変換を行う際の単位となる。

b. デックは次のいずれかの方法で定義される。

(i) 制御命令 HARP および END ステートメントで囲まれたセグメントの集まり。

(ii) 制御命令 HISAP および擬似命令 END で囲まれたセグメントの集まり。

(iii) 制御命令 LOAD および制御命令 DKEND で囲まれたセグメントの集まり。

(i) は HARP 5020 言語で，また (ii) は HISAP 5020 言語で書かれたセグメントから構成されるデックで，これらはそれぞれ HARP 5020, HISAP

5020 による処理を受ける。(iii)は一旦 HARP 5020 または HISAP 5020 によつて翻訳または変換された再配置可能 2 進プログラムからなるデックである。

c. 制御命令 HARP, HISAP, LOAD の場所欄に書かれた記号はデック名となる。デック名はデックに関する他の情報と共に、そのデックに関するセグメント表の先頭に記録される。

d. 同一デック内では、アルセグメントから同じデック内で定義されている他のセグメントの内部の記号を自由に参照することができる。この参照は内部記号表の情報に基づいて行われる。

e. あるデックから他のデックで定義されたセグメントをその外部名によつて参照することができる。またライブラリーセグメントはどのデックからも自由に参照可能である。

(3) 仕事：仕事とは次のようなものである。

a. 仕事はデックの集まりであつて、一つのまとまつた作業の単位である。

b. 制御命令 EXECUTE と EOF で囲まれたデックの列として定義される。必要ならばこの後にデータのデックを付け、最後に EOF 命令を置く。

c. チェインジョブの場合は各チェイン要素の前に制御命令 CHAIN を置く。あるチェイン要素内のデックから他のチェイン要素を呼出すには HARP 5020 ならばステートメント CALL CHAIN を、HISAP 5020 ならば擬似命令 CALLCH を用いる。

§ 4 FCP (file control processor) について

入出力の際のバッファリング、エラー処理等を統一的に扱うためのシステムとして FCP がある。HISAP 5020 の文法上では FCP が働くためのパラメータの定義、バッファの定義及び確保等を行う擬似命令と、入出力及び演算を行うマクロ命令との 2 種の命令群がある。

4.1 DTF (define tape file)

磁気テープファイルの性格を次の 24 の項目によつて明らかにするものである。

(1) FILENAME

磁気テープファイルの名称を定義するもので、入出力マクロ命令の変数として利用される。

(2) FCHANNEL

磁気テープファイルが接続されるチャネルを指定するもので、一般にはチャネル番号ではなくチャネルの記号名が用いられる。

(3) BASETAPE

磁気テープファイルが接続されるデバイスを指定するものである。

(4) ALT1TAPE

磁気テープファイルが1巻のリールに入りきらないとき2本目のリールをかけるデバイスを指定する。

(5) ALT2TAPE

同様に2巻以上のリールになる場合の3本目のリールをかけるデバイスを指定する。

(6) FILEFORM

磁気テープファイルの形式を定義するものである。このFCPでは次の2つの形式を取扱う。

(形式1)

1記録を構成する語数が一定で且つ集積因数が一定のもの。

(例)

I B G	記録1	記録2	記録3	記録4	I B G
	40語	40語	40語	40語	

集積因数 = 4

記録の長さ = 40

(形式2)

1記録を構成する語数は不定(但し最大はわかっている)であるが集積因数が一定のもの。

(例)

I B G	記録1	記録2	記録3	記録4	I B G	記録5	記録6	記録7	記録8	I B G
	30語	40語	20語	30語		40語	40語	30語	20語	

集積因数 = 4

記録の最大の長さ = 40

(7) FILETYPE

入力用ファイルか又は出力用ファイルかの区別をするための項目である。

この FILETYPE は、マクロ命令 $\overline{\text{OPENR}}$ (open reverse) によつて書き変えられることがある。

(8) RECLNGTH

記録の長さを規定するもので、形式 1 では記録の長さ、形式 2 では最大記録の長さを表わす。

(9) $\overline{\text{BLOCKING}}$

集積因数を規定する。

(10) $\overline{\text{OPENPR}\overline{\text{OC}}}$

$\overline{\text{OPEN}}$ マクロ命令を実行する際に巻戻しを行うか否かを規定する。

(11) $\overline{\text{CLSEPR}\overline{\text{OC}}}$

$\overline{\text{CLOSE}}$ マクロ命令を実行した後巻戻しを行うか否かを規定する。

(12) $\overline{\text{IOMETH}\overline{\text{OD}}}$

入力ファイル用と出力ファイル用とのバッファーとして同一 3 域バッファーを回転順に使用する場合の指定を行う。

(13) $\overline{\text{TI}\overline{\text{OAREAS}}}$

磁気テープファイル用のバッファーとして何域バッファーを利用するかを規定するもので、1, 2, 3 の指定ができる。

(14) $\overline{\text{PRI}\overline{\text{ORITY}}}$

同一チャネルに接続された別のファイルとの優先度を規定する。

(15) $\overline{\text{CHPTPR}\overline{\text{OC}}}$

検査点記録の有無(入力ファイルの場合)又は検査点記録を書くか否か(出力ファイルの場合)を規定する。

(16) $\overline{\text{RWNDPR}\overline{\text{OC}}}$

後書きラベルの処理が終つてリールのスイッチを行うときの巻戻し手順を規定する。

(17) $\overline{\text{COMD}\overline{\text{LIST}}}$

ファイルとパツプアとの対応をつけるためのもので擬似命令 DRAV の場所
概の記号を書く。

(18) SLRPROC D

短長記録があつた場合の処理ルーチンの場所を指定する。

(19) LLRPROC D

過長記録があつた場合の処理ルーチンの場所を指定する。

(20) SYMPROC D

シンボルを読んだときの処理ルーチンの場所を指定する。

(21) EOFPROC D

ファイルの終りに到達したときの処理ルーチンの場所を指定する。

(22) BTLPROC D

前書きラベルの標準的処理は FCP が行うがそれ以外に使用者が追加処理を
行いたいときその処理ルーチンの場所を指定する。

(23) ETLPROC D

後書きラベルの標準的処理は FCP が行うがそれ以外に使用者が追加処理
を行いたいときその処理ルーチンの場所を指定する。

(24) LABELINF

前書きラベル及び後書きラベルに関する一切の情報は擬似命令 CDATA で
記述しておかなければならないがこの CDATA の場所を指定する。

これら 24 の項目は各ファイルについて記述しなければならないものである。
これらが HISAP 5020 によつて翻訳されて、次の図のような 11 語の情報と
して FCP へ渡される。

0	m_0	1	ABB
1	m_1	1	ABB
2	m_2	1	ABB
3	block counter	CODDD	RECORD LENGTH
4	BLOCKING	EEFF	IOM TIO PRI
5	COMMAND	GGG	HHHH
6	SLRPROC	LLRPROC	
7	SYMPROC	EOFPROC	
8	BTLPROC	ETLPROC	
9	LABELINF		
10	error count	Permanent	noise

BASETAPE m_i はチャヤナル・デバイスを指定する語のアドレス
 ALT1TAPE A は処理の対象となつているテープなら 1
 ALT2TAPE BB で OPEN, CLOSE されたか否かの区別

block counter: 1 プロック処理される毎に 1 増す。現在のテープの位置を示すものである。
 CC: FILE FORM
 DDD: FILE TYPE
 RECORD LENGTH: RECLNGTH

BLOCKING: BLOCKING
 EE: OPENPROC (open procedure)
 FF: CLSEPROC (close procedure)

IOM: IOMETHOD
 TIO: TIOAREAS
 PRI: PRIORITY

COMMAND: COMDLIST (command list)
 GGG: CHPTPROC (check point procedure)
 HHHH: RWNDPROC (rewind procedure)

error count: エラー発生回数を記憶
 permanent: 10 回読み直してみてなおエラーが発生する場合のカウン
 noise: その他 C1 などのカウン

FCPはこれらの情報を参照しながら動く。

4.2 DRAV

入出力マクロ命令によつてデータを入出力するための入出力区域を磁心記憶装置内に定義かつ確保するための擬似命令である。この命令のアドレス欄には記録の長さ l と集積因数 b とが書かれる。これはDTFのLECLNGTH, BLOCKINGの項目の記述と対応している。従つてLECLNGTH= l , BLOCKING= b とすると、 $b \times l + b$ 語の場所が確保される。ただし、その中 b 語は各記録を制御する指令のためのものである。

各記録の副項目を定義し、副項目間の演算を行う。この定義に際してはその副項目の中に入るべきデータの種類(例えば、2進データ、10進データ、字データ等)をも記述する。

4.3 その他の定義擬似命令について

(1) DUF (define unit record file)

セン孔カードファイルをFCPで処理する場合の各パラメータの定義を行う。

(2) DCHPT (define check point)

検査点記録を書く際の記述であつて検査点記録を書く頻度や検査点用テープの指定などを行う。

(3) DRA (define record area)

DRAVの場合は指令を附加したがDRAでは指令が見つからない。

(4) DRC (define record control word)

記録制御語(指令)を作り出すためのものである。

(5) 磁気ドラム入出力制御語

(6) DLINE (define line)

行印字機に印字する場合の様式を定義する。

(7) CODE

記録の副項目の内容に対して名前をつけるための擬似命令である。この名前はマクロ命令 $\overline{\text{BOOL}}$, $\overline{\text{DECODE}}$ で利用される。

4.4 入出力マクロ命令について

入出力マクロ命令には次のものがある。

(1) $\overline{\text{OPEN}}$

入出力ファイルの初期設定を行う。即ち、DTFのOPENPROCの記述に従って巻戻しの処理をし次に前書きラベルのチェックを行う。

(2) OPENR

入力テープファイルの逆読込みを行う際の初期設定を行う。

(3) CLOSE

入出力ファイルの処理の終わりにファイルの後処理を行う。DTFのCLSEPROCの記述に従って巻戻しの処理をする。

(4) CLOSEV

すべての入出力ファイルをCLOSEしかつ制御をモニタに移して仕事の終りとする。

(5) READ, WRITE

データの移動(入出力装置 ↔ 磁心記憶装置, 磁心記憶装置 ↔ 磁心記憶装置)を行うマクロ命令である。制御の動きについては後述する。

(6) READD, WRITED

磁気ドラム記憶装置とのデータの移動を行うマクロ命令である。

(7) PRINT, PRINTV

行印字機へデータを印字するマクロ命令である。

(8) PUNCH, PUNCHV

カードセン孔機にセン孔させるためのマクロ命令である。

4.5 演算マクロ命令について

(1) ARITH

各記録の副項目間の演算を行うマクロ命令であり、アドレス欄に式を書く。

(例)

```
ANYLABEL ARITH A=B+C*D
```

加減乗除, 巾乗, 平方根, 正弦, 余弦, 逆正弦, 逆正接, 常用対数, 自然対数, 10の巾乗, eの巾乗, 絶対値などの演算が可能である。

(2) BOOL

論理式の判定を行いその結果によつて枝分れするマクロ命令である。論理式の中には一般の論理式の他に, スイッチのオン, オフ, CODE値の真偽等

がある。

(3) $\overline{\text{COMP}}$

2つの値を比較し(この値は副項目の内容である),その大小によつて枝分れする。

(4) $\overline{\text{DECODE}}$

CODE値によつて枝分れする。

(5) $\overline{\text{FILL}}$, $\overline{\text{FILLV}}$

記録又は副項目の区域をあるパターンで埋めるマクロ命令である。

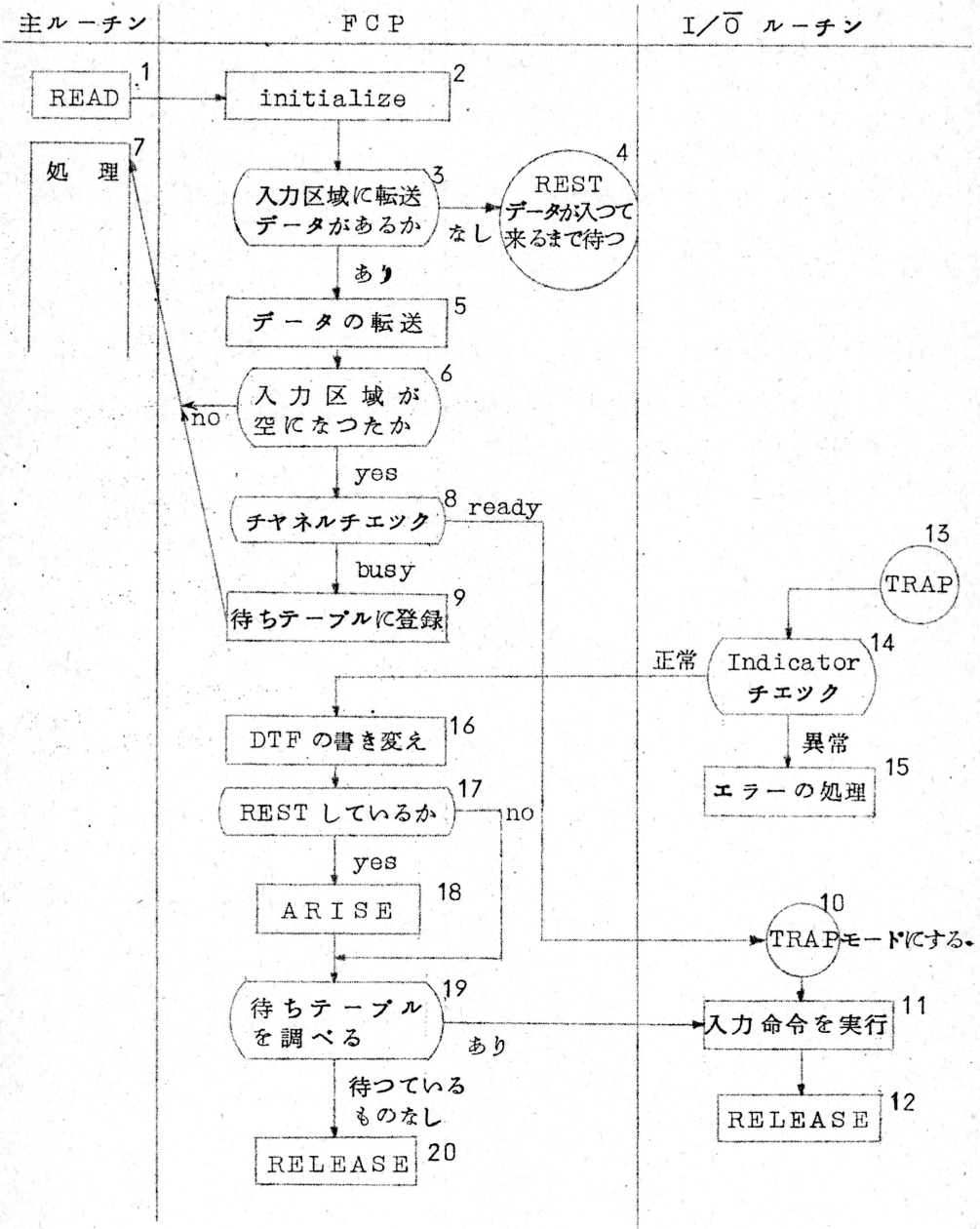
(6) $\overline{\text{MOVE}}$, $\overline{\text{MOVEV}}$

磁心記憶装置対磁心記憶装置の情報の移動を行う。

4.6 制御の動きについて

特に $\overline{\text{READ}}$ マクロ命令が実行された際の制御の動きについて次に略図によつて説明する。

制御の流れ図



1. 主ルーチンからREAD命令を出す。
2. READ命令のアドレス欄によつてファイルが定められているのでそのファイルについてのinitializeを行う。
3. ファイルの入力区域にデータが入っているか否かをチェックする。

4. もし入っていないければ、入つて来るまで仕事は行えないのでRESTして待つ。
RESTしている間に他の仕事を行うことが不可能である。
5. もし入つていればデータの転送を行う。
6. データ転送が終ると入力区域が空になつたか否かのチェックを行う。
7. もし空でなければ主ルーチンへ戻り処理を行う。
8. もし空であれば、そのファイルの接続されているチャネルをチェックする。
9. チャネルがbusyであればそのチャネルの待ちテーブルに登録して、主ルーチンへ戻る。
10. チャネルがreadyであればI/Oルーチンへ行き、TRAPモードにする。RELEASEされると主ルーチンへ戻るようにしておく。
11. 入力命令を実行する。入力が終るとTRAPが発生するようにしておく。
12. RELEASEする。
13. TRAPにより、制御をI/Oルーチンに渡す。このとき主ルーチンが動いているのかその他の仕事が動いているのかは不明である。
14. チャネル表示子のチェックを行う。
15. エラーがあればその処理をする。エラーの中には、シンボル、Long record, tape end, errorなどがある。FCPに戻つてDTFのerror countの処理をしたり、シンボルのときは、SYMPROCDにジャンプする。
16. 正常に読み込みが行われていれば、FCPへ戻り、DTFのblock counterを1増す。
17. TRAP発生時に主ルーチンがこのファイルを待つてRESTしているかどうかをチェックする。
18. RESTしていればARISEさせる。
19. 待ちテーブルを調べこのチャネルを待つているファイルがあるかどうかをチェックする。もしあればその中の最優先のファイルについて入出力を行うためにI/Oルーチンへ行く。
20. もしなければRELEASEする。

本 PDF ファイルは 1964 年発行の「第 5 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者(論文を執筆された故人の相続人)を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思えます。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>