小樽商科大学データ・ステーションにおけるオープン・リモート・バッチについて

戸島　潮

1. まえがき

本稿は小樽商科大学データ・ステーションからオープンでリモート・バッチ・ジョブを行なう利用者のリファレンス・マニュアルとして役立つことを目的としている。しかし，一般的興味からもよめるように配慮してある。何分にも紙数がかぎられているためこまかいヴァラィアティに十分言及しえなかったが，ひと通りのことは述べてあるので全くの初心者でもこの記述をたよりに独力でリモート・バッチ・ジョブを行なうことができるであろう。本稿の読者には DŌS 45 モニタのジョブ制御文の知識があることを一応前提しているが，実用だけのためならそれすらなくてもよい。

図1に小樽商科大学データ・ステーションの機器構成を示す。小樽商科大学データ・ステーション設置の顛末は戸島[1) 2)]が述べている。また，そこで使用されている伝送制御と入出力装置制御のための通信制御プログラム（Communication Control Program）の内部構造の概略は戸島・稲田[3)]が述べている。

なお，以下ではカードはすべて EL コードでパンチされるものとする。

---

1) 戸島　潮，「TSS 講座(4)—小樽商科大学データ・ステーション顛末記（その1）—」，北海道大学大型計算機センター・ニュース，VŌL. 6, NŌ. 1 (1974年3月)，pp. 9–16.

2) 戸島　潮，「TSS 講座(5)—小樽商科大学データ・ステーション顛末記（その2）—」，北海道大学大型計算機センター・ニュース，VŌL. 6, NŌ. 2 (1974年6月)，pp. 13–21.

3) 戸島　潮，稲田信幸，「小樽商科大学データ・ステーションの通信制御プログラム」，北海道大学大型計算機センター・ニュース，VŌL. 6, NŌ. 4 (1974年8月)，pp. 18–32.

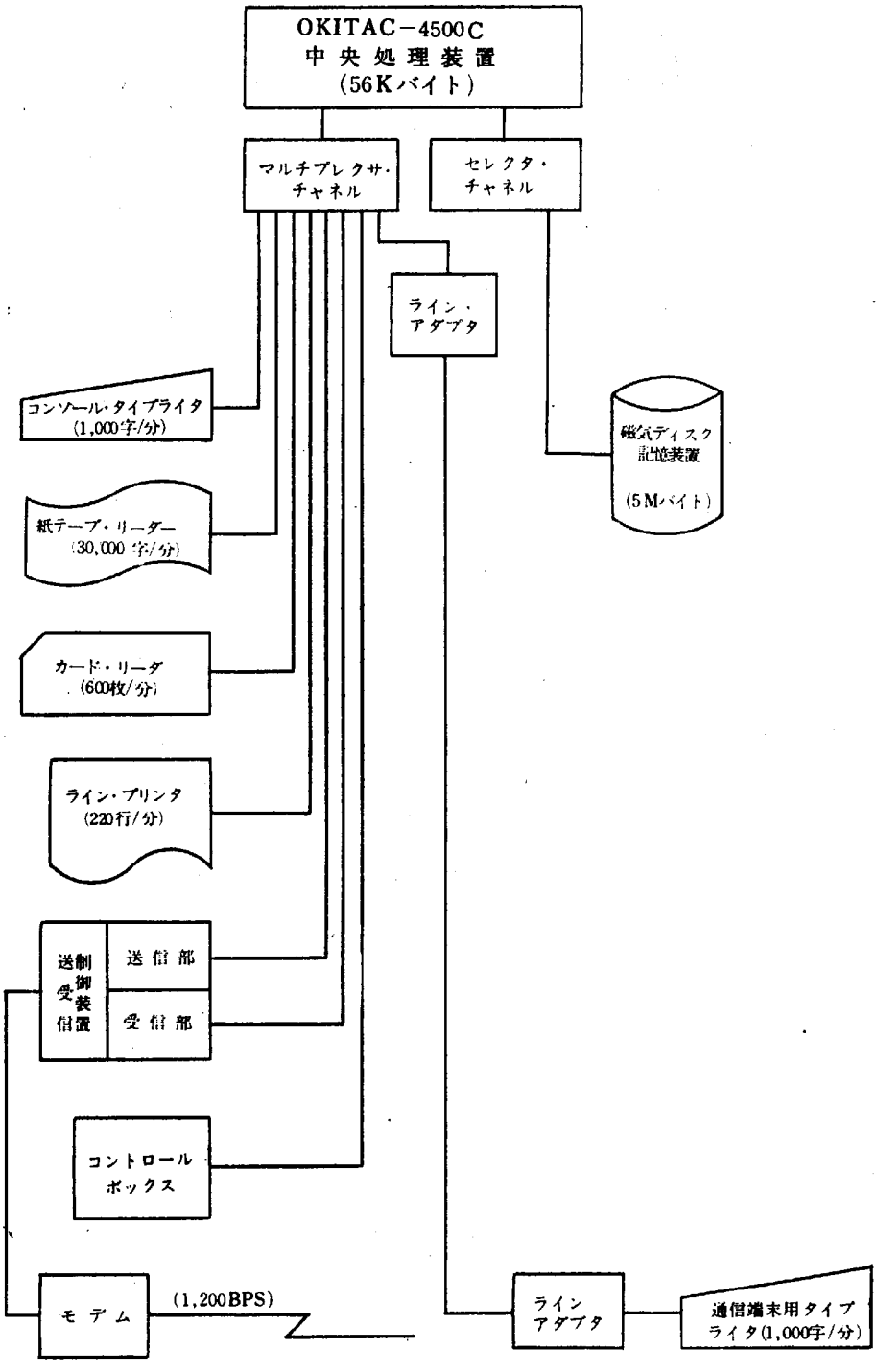


図1　小樽商科大学データ・ステーション機器構成図

2. 小樽商科大学データ・ステーションの概要

小樽商科大学データ・ステーションの端末制御装置 ŌKITAC-4500C 用の通信制御プログラムは CCP45 とよばれる。これはこのプログラムのロード

・モジュールのディスク上におけるデータ・セット名でもある。CCP45 は ŌKITAC-4500 の ŌS である DŌS45 の管理プログラム（モニタ）の管理下で動作する処理プログラムである。DŌS45 のモニタはとくに CCP45 のために用意された管理プログラムではなく，いかなるときもモニタ自身はどんな処理プログラムが走っているかということには一切関知しないといういみにおいて汎用である。通常ミニコンを端末制御装置に使用した端末においては，通信制御プログラムは端末制御装置と一体となって通信制御装置（CCU: Communication Control Unit）の役割をはたすので，端末がセンターに対してオンラインである時とオフラインである時の区別は明瞭であるが，小樽商科大学データ・ステーションではこの事情は多少ちがっていて，モデム・インタフェースである送受信制御装置（TRC: Transmission Reception Control Unit）があたかも入出力装置のようになっているので ŌKITAC-4500C のプログラムは通常の入出力に関するシステム・マクロを使用して通信回線に対する送受信を行なうことができる。ただし，DŌS45 のデータ管理プログラムは伝送制御を全くサポートしていないので，ここで言っている送受信は文字通り回線への送信と回線からの受信そのものである。したがって，伝送制御は利用者がすべて適当に行なわなければならない。この点が TRC に対する入出力マクロと他の入出力装置に対する入出力マクロの大きなちがいであるが，ともかくも TRC が入出力装置となっていることには変りがない。CCP45 は 15.2K バイトの大きさをもっているが，オーバレイをしているので実行のためには主記憶装置が 13.2K バイトあればたりる。昭和49年8月現在小樽商科大学データ・ステーションの ŌKITAC-4500C は 56K バイトの主記憶装置をもっているが，このうちモニタの常駐域と制御表のために 19K バイトが必要である。そこで利用者のためのエリアは 37K バイトということになる。DŌS45 のモニタではマルチジョブ制御が可能であるので，これをたとえば 14K バイトと 23K バイトのふたつのパーティションにわけて，前者に CCP45 を格納し，後者に他の処理プログラムを格納して並行に走らせることができる。また，CCP45 の起動を必要とするジョブ

と他の処理プログラムの起動を必要とするジョブをひとつのインプット・ジョブ・ストリームの中に任意の順番に配列しておいて連続的に処理させることもできる。このように，小樽商科大学データ・ステーションにおいては，オンライン時とオフライン時の区別がかならずしも明瞭ではない。これは端末制御装置として計算機を使用しているという特徴を十分発揮させるものである。

現在小樽商科大学データ・ステーションにおいては，リモート・バッチ・ジョブとデマンド・ジョブを行なうことができる。前者はセンターの業務受付に計算依頼する時に出すのとほとんど同じカード・デックの内容を端末からセンターに一括送信し，計算結果を一括受信するものである。このかぎりにおいて，リモート・バッチの利用者にとって端末の入出力装置はセンターの入出力装置と異なるところがない。ただし，通信回線の速度の制約によって端末の入出力装置はセンターのものにくらべて低速である。実際，センターでのリモート・バッチの処理の仕方はセンターのローカル・バッチの処理の仕方と大差はないのである。したがって，リモート・バッチは遠隔地の利用者のそばにセンターをひきつけるという効果をもつ以外には，センターに出かけて行って業務受付に計算依頼するのと原理的には変りがないといえよう。これに対してデマンド・ジョブは端末の利用者がタイプライタなどを介してセンターの計算機と会話をかわしながら処理をすすめて行くもので，この時利用者はあたかも自分ひとりでセンターの計算機を占有しているような錯覚におち入る。実際にはセンターの計算機は時分割（TS：Time Sharing）を行なっているにすぎないが，このデマンド・ジョブとリモート・バッチジョブの大きな相違点は前者がとにかくジョブの実行を端末から制御できるということである。リモート・バッチで一括送信したジョブはもはや一括受信をする以外には端末からいかなる制御を行なうこともできない。これに反してデマンド・ジョブではセンターのコンソールから利用者のプログラムにいろいろ働きかけるのとほぼ同じようなことを端末から行なおうと思えばかならずしも不可能ではない。そこでデマンド・ジョブは遠隔地の利用者のそば

にセンターをひきつけたうえ，計算機を占有して直接制御させるという効果をもつ。これはセンターの業務受付に計算依頼するというセンターの利用方法とは本質的にちがうものである。端末からデマンド・ジョブを行なう時の唯一の欠点は，錯覚ではなしに実際にひとりの利用者が端末を占有してしまって，その間他の利用者のジョブが待たされてしまうことである。したがって，データ・ステーションの利用者が多数いればデマンド・ジョブはきわめて効率の悪いセンター利用方法ということになる。データ・ステーションに端末タイプライタを何台設置しても端末としてはひとつであるから何ら事態を改善することはできない。

以上の説明を通じて，データ・ステーションとして端末制御装置に計算機を使用することがひとつも本質的ないみをもっていないことに気付かれたと思う。実際，伝送制御と端末の入出力装置制御のためだけならば計算機使用の必然性はないのである。センターも端末制御装置が計算機であることを何ら意識していない。端末制御装置として計算機を使用するのは急速なミニコンの価格低下を背景として伝送制御手順変更に対する適応の容易性によるものであろう。それとともに何といっても単体の計算機としての利用可能性が魅力であることはいうまでもない。このいみでデータ・ステーションの端末制御装置に計算機を使用することはもっぱらデータ・ステーション設置側の個別的事情から由来するものにすぎず，端末にとって本質的に重要なのは端末の入出力装置とセンターの計算機である。それゆえ，データ・ステーションの端末制御装置が計算機であってもそれはセンターの計算機との関係において計算機結合または計算機ネットワークというものとはいささか趣を異にすると考えられる。ただし，プログラムを適当に開発すればセンターの計算機と端末の計算機の間でかなりインタラクティヴなことができるのは事実である。

3. リモート・バッチ・ジョブ

データ・ステーションの運用をどのようにしたらよいかについては種々の

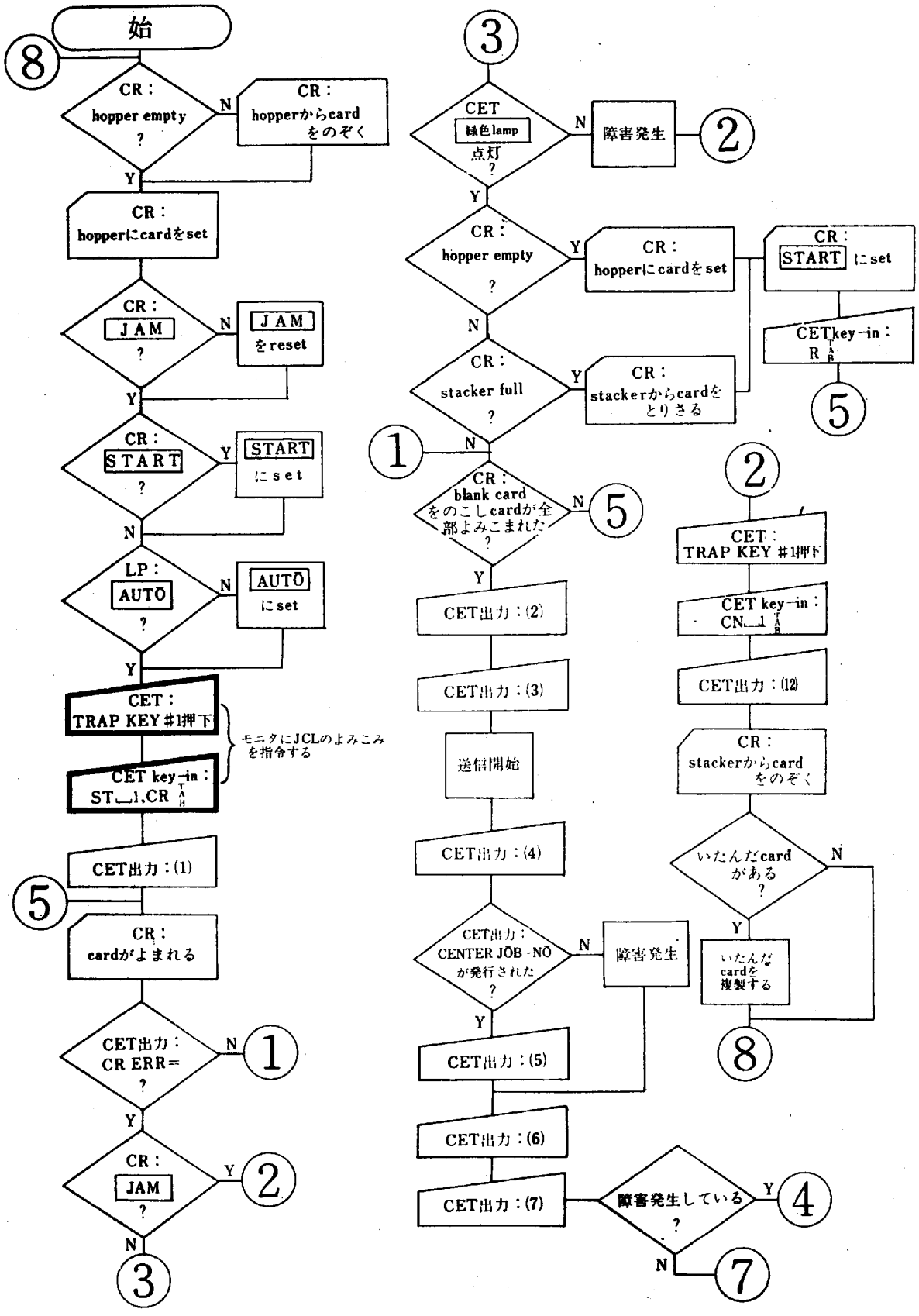
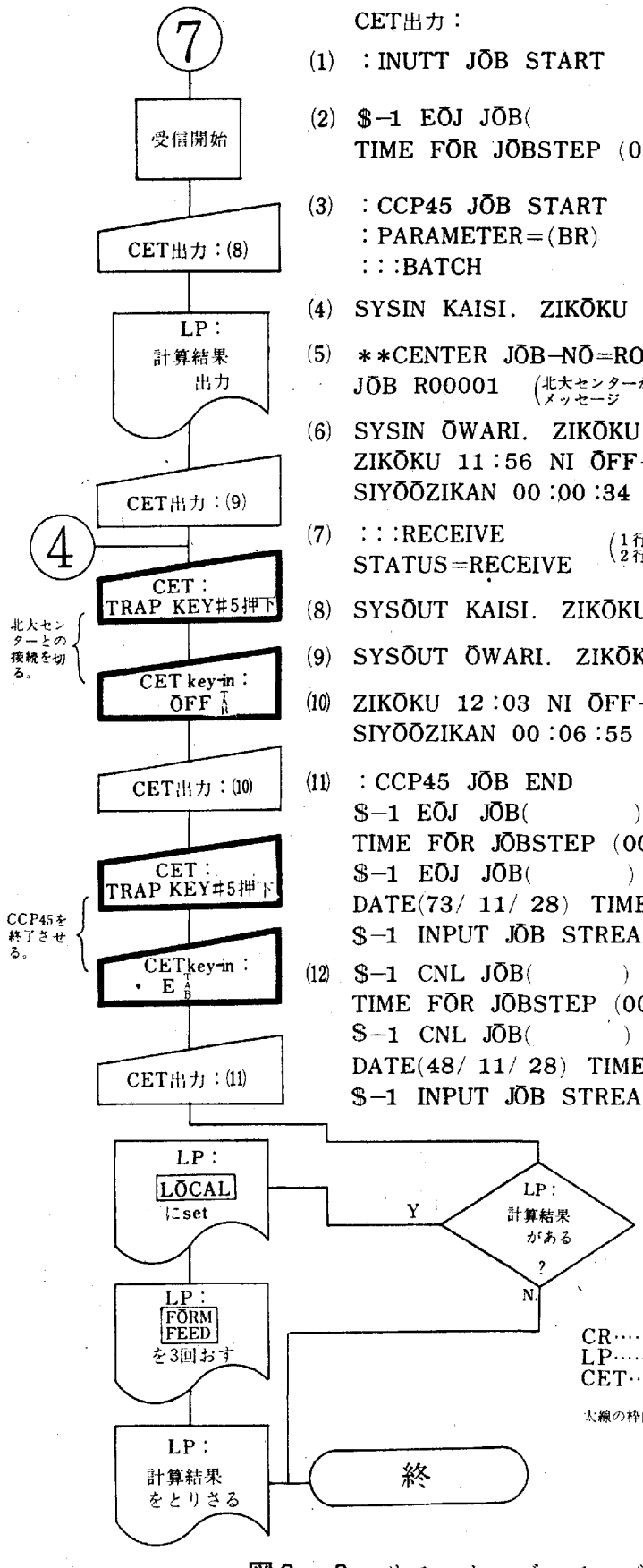


図2—1　リモート・バッチ・ジョブの操作法



CET出力：

(1) ：INUTT JŌB START (入力ユーティリティからのメッセージ)

(2) $−1 EŌJ JŌB( )JŌBSTEP(0001)
TIME FŌR JŌBSTEP (00.00.09) (モニタからのメッセージ)

(3) ：CCP45 JŌB START
：PARAMETER=(BR) (CCP45からのメッセージ)
：：:BATCH

(4) SYSIN KAISI. ZIKŌKU 11:55 (北大センターからのメッセージ)

(5) **CENTER JŌB−NŌ=RO1469 FRŌM TERMINAL
JŌB R00001 (北大センターからのメッセージ)

(6) SYSIN ŌWARI. ZIKŌKU 11:56
ZIKŌKU 11:56 NI ŌFF−LINE NI NATTA.
SIYŌŌZIKAN 00:00:34 (北大センターからのメッセージ)

(7) ：：:RECEIVE (1行目…CCP45からのメッセージ\2行目…北大センターからのメッセージ)
STATUS=RECEIVE

(8) SYSŌUT KAISI. ZIKŌKU 12:00 JN=RO1469 (北大センターからのメッセージ)

(9) SYSŌUT ŌWARI. ZIKŌKU 12:03 JN=RO1469 (北大センターからのメッセージ)

(10) ZIKŌKU 12:03 NI ŌFF−LINE NI NATTA.
SIYŌŌZIKAN 00:06:55 (北大センターからのメッセージ)

(11) ：CCP45 JŌB END
$−1 EŌJ JŌB( ) JŌBSTEP(0002)
TIME FŌR JŌBSTEP (00. 14. 44)
$−1 EŌJ JŌB( ) PRŌGRAMER( )
DATE(73/ 11/ 28) TIME FŌR JŌB (00. 14. 53)
$−1 INPUT JŌB STREAM END (1行目のみCCP45からのメッセージ他はモニタからのメッセージ)

(12) $−1 CNL JŌB( ) JŌBSTEP(0001)
TIME FŌR JŌBSTEP (00. 00. 18)
$−1 CNL JŌB( ) PRŌGRAMER( )
DATE(48/ 11/ 28) TIME FŌR JŌB (00. 00. 18)
$−1 INPUT JŌB STREAM END (モニタからのメッセージ)

CR……Card Reader
LP……Line Printer
CET……Console Electric Typewritter

太線の枠内のoperationはとくに重要なので誤りのないよう注意されたい。

図2—2 リモート・バッチ・ジョブの操作法

考え方があるであろうが，ここでは利用者がオープンで小樽商科大学データ・ステーションを利用するのに必要な事柄をのべる。まず，利用者がどのような操作をしなければならないかを図2に示す。ここでホッパにセットすべきカード・デックはつぎのように構成されていなければならない。1枚目のカードはDŌS45 モニタに対するJŌB 文である。JŌB 文の形式はつぎの通りである。

//JŌB␣〈ジョブ・ネーム〉([〈プログラマ・ネーム〉], [〈打ち切り時間〉], [〈JŌBパラメータ〉])

ジョブ・ネームは英字からはじまる6文字以内の英数字の記号列である。プログラマ・ネームはコンマ，改行，TAB，ブランク以外の任意のコード10文字以内の記号列である。打ち切り時間は分単位で指定するが無記入であれば打ち切り時間は無限大となる。JŌB パラメータとしてはEとFがいみをもつが，Eはライン・プリンタにヘッダとトレイラのプリントを行なうことをやめさせ，FはDŌS45 モニタに対するジョブ制御文をプリントさせる。Eの指定がのぞましい。

例

//JŌB␣AJŌB (TŌSHIMA, , E)

2枚目のカードはDŌS45 モニタに対するEXEC 文である。これはつぎのように指定する。

//EXEC␣CTL=HJŌB

ここでHJŌB はカタログ・ネームであらかじめディスクに登録されているつぎのジョブ制御文のあつまり（カタログ）がよばれる。

//EXEC␣PGM=INUTT, P (I)
//DD␣IPTFIL=DK02 (DSA, 100), R
//DD␣IPTR=CR
//EXEC␣PGM=CCP45, P (BR)
//DD␣ŌPTFIL=LP

このジョブ制御文のあつまりの登録はFILER というプログラムによって行

なうが，それはデータ・ステーション管理者のなすべきことであろう。

リモート・バッチにおいて端末が行なうことは，送信用データをセンターに送信することと，計算結果を受信することであるが，上のジョブ制御文はこれをふたつのジョブ・ステップにわけている。第1は送信用データをいったんディスク上にファイルするジョブ・ステップ，第2はディスク上にファイルされた送信用データをセンターに送信し，計算結果を受信してライン・プリンタに出力するジョブ・ステップである。最初のジョブ・ステップで起動される処理プログラムは入力ユーティリティ（ロード・モジュールのプログラム・ネームは INUTT）である。このプログラムを起動するためのジョブ制御文の一般形はつぎの通りである。

//EXEC␣PGM=INUTT[,P(パラメータ)]

//DD␣IPTFIL=DK××(DSA,{〈スペース長〉[K] / *})[,R]

//DD␣IPTR={CR / PR××}

パラメータにはIとMを指定することができる。Iを指定すると入力ユーティリティはディスク上に新たに入力ファイルのエリアを確保する機能をもつ。この時入力ファイルを指定するDD文で確保すべきファイル領域をK指定するか，DD文がそれ以降のジョブ・ステップでも意味をもつようにR指定するか，いずれかを指定しなければならない。入力ユーティリティは入力ファイル領域の確保とともにファイル・ヘッダのカウンタ類のイニシャリゼーションを行なう。Iを省略すれば入力ユーティリティはディスク上にすでに入力ファイル領域が確保されているものとして処理をすすめる。この時入力ファイルを指定するDD文では＊を指定しなければならない。オープン利用時はかならずI指定をする。Mを指定すると入力ユーティリティの起動後プログラムはコンソール・タイプライタからのオペレータの介入をまつ。この時オペレータはいくつかのコマンドを打鍵することができるが，これはオペレータが特殊な保守のために利用すべきものなので一般のオープン利用者はかならずしも知る必要はない。さて，つぎのジョブ・ステップで起動さ

れる処理プログラムはいうまでもなくCCP45である。このプログラムを起動するためのジョブ制御文の一般形はつぎの通りである。

//EXEC␣PGM=CCP45, P (パラメータ)

$$\text{//DD␣IPTFIL=}\begin{cases}\text{DK}\times\times\text{(DSA,*)}\\ \text{CR}\\ \text{PR}\times\times\end{cases}$$

$$\text{//DD␣}\bar{\text{O}}\text{PTFIL=}\begin{cases}\text{DK}\times\times\text{(DSB,}\begin{cases}\langle\text{スペース長}\rangle\text{K}\\ *\end{cases}\text{)}\\ \text{LP}\end{cases}$$

CCP45の出力媒体としてディスク上の出力ファイルとライン・プリンタのいずれかを指定できるが，後述するように通常のオープン使用ではライン・プリンタを指定すれば十分である。ディスク上の出力ファイルを指定する時，出力ファイル領域が確保されていなければ，パラメータにIを指定するとともに〈スペース長〉Kを指定する。ファイル領域がすでに確保されているならば*の指定だけでよい。出力ファイル上にかきこまれたデータをライン・プリンタに出力するためには出力ユーティリティ（ロード・モジュールのプログラム・ネームは $\bar{\text{O}}$UTUTT）を使用する。出力ユーティリティ起動のためのジョブ制御文の一般形はつぎの通りである。

//EXEC␣PGM=$\bar{\text{O}}$UTUTT[,P(パラメータ)]

$$\text{//DD␣}\bar{\text{O}}\text{PTFIL=DK}\times\times\text{(DSB,}\begin{cases}*\\ *\text{DELE}\end{cases}\text{)}$$

出力ユーティリティ実行後，出力ファイルをのこす時は*，抹消する時は*DELEを指定する。パラメータとしてはMが指定できるが，これは入力ユーティリティのパラメータMと同じいみをもつので一般のオープン利用者は指定しない方がよい。なお，その他のパラメータのいみについては後述する。

さて，カード・デックの3枚目から送信用のデータをパンチしたカードがはじまる。これはセンターの業務受付にだすものと全く同じ構成でなければならないが，つぎの3点がちがうので注意しなければならない。第1は N$\bar{\text{O}}$ 文である。これはカードの1欄目から ¥N$\bar{\text{O}}$␣ とパンチしてありさえすれば

あとは何であってもよい。図2の CET 出力(5)にみられるように，センターは端末からの NŌ 文に対してあらたに一連番号を発行する。第2は送信用データの一番最後が ¥END 文でなければならないことである。これはカードの1欄目からこの通りにパンチする。第3は HJŌB カードのジョブ種別のうしろにコンマ，FR.02 とパンチしなければならないことである。ちなみに，端末から PL/I コンパイラを起動する時のジョブ種別は P である。なお，ジョブは何個連続していてもよい。要するに最後のジョブの ¥JEND のつぎのカードが ¥END でありさえすればよいのである。ついでに間違いやすい点に注意しておけば，センターからの処理プログラムの出力を端末に出すには，センターのジョブ制御マクロのパラメータとして ŌUTPUT=REMŌTE と指定するかまたは FD 文に SYSŌUT=REMŌTE と指定しなければならない。この指定がないとすべてセンター出力となる。最後の送信用データ・カードのうしろには1枚のブランク・カードをつけその後にインプット・ジョブ・ストリームがつきたことを DŌS45 モニタにつげるため /# を1欄目からパンチしたカードをつけさらにそのうしろにブランク・カードをつけるかまたは最後の送信用データ・カードの後のブランク・カードにつづいてつぎのジョブの JŌB カードがつづくかいずれかになる。

以上のように，センターへの送信用データ・カードのほかにオープン利用者が必要とする DŌS45 モニタに対するジョブ制御カードは3枚である（ブランク・カードをふくめると4枚になる)。これはデータ・ステーションの管理者が用意して利用者に便宜をはかるのもひとつの方法であろう。しかし原則としてオープン利用者が用意する。利用者は DŌS45 モニタに対するジョブ制御文によって前後をはさまれた送信用データ・カードから構成されるカード・デックをよくさばいてカード・リーダのホッパにセットする。その後，図2のフロー・チャートにみられるように，カード・リーダとライン・プリンタのランプ表示を確認する。ついでジョブ制御文のよみこみを DŌS45 モニタに指令する。図3にコンソール・タイプライタおよび通信端末タイプライタの鍵盤配列を示す。最上段のキイのうち中央の番号が付されている5

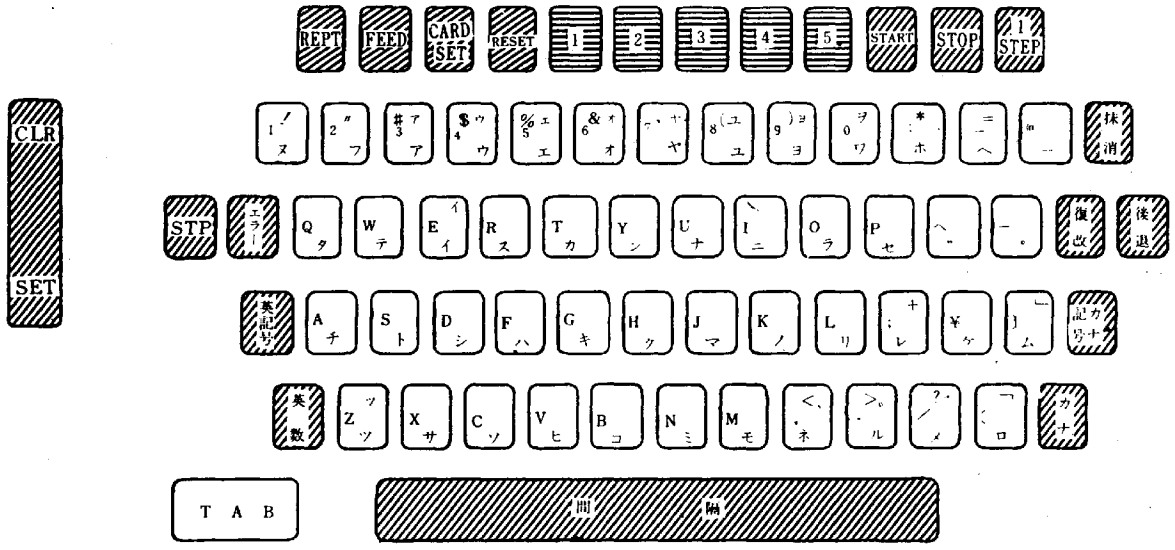


図3　コンソール／通信端末タイプライタ鍵盤配列

つのキイを割込みキイ（Trap Key）という。最上段のキイ以外は鍵盤入力の状態になっていなければロックされていて打鍵ができないようになっている。割込みキイはつねに押下できる。DOS45 モニタにジョブ制御文よみこみを指令するには，まずコンソール・タイプライタの割込みキイ1を押下する。これでコンソール・タイプライタの鍵盤入力が可能となるので（* が出力する）

$$\underline{*}\,\mathrm{ST}\underline{\sqcup}1,\ \mathrm{CR}\ \begin{matrix}\mathrm{T}\\ \mathrm{A}\\ \mathrm{B}\end{matrix}$$

と打鍵する（下線は計算機からの出力を示す）。ST は START の略で，これはカード・リーダからジョブ制御文をよみこんで，それにしたがってパーティション1でジョブを開始せよといういみをもつ。この指令によってカード・デックの先頭の2枚のジョブ制御文がよまれる。EXEC 文はカタログ（ディスク上に登録されているジョブ制御文の集り）に展開される。その結果，すでに示した展開形からあきらかなように，第1ジョブ・ステップとして入力ユーティリティが起動される。ここで図2の CET 出力(1)がえられる。ついで送信用データ・カードが入力ユーティリティによってよみこまれて入力ファイルに登録される。この最中にもっとも起りやすい障害はカード・ジャムである。これが起るとコンソール・タイプライタに

: : CR ERR=

と出力し入力ユーティリティに対して鍵盤入力が可能となる。通常はこの状態でジョブをキャンセルしてジャムしたカードをパンチしなおしてもう一度最初からやり直すのが確実である。ジョブをキャンセルするには図2のフロー・チャート通りに，コンソール・タイプライタの割込みキイ1を押下したあと

$\underline{*}$ CN␣1 $\begin{matrix}T\\A\\B\end{matrix}$

と打鍵する。CN は CANCEL の略で，これはパーティション1のジョブをキャンセルすることを DŌS45 モニタに指令する。この時 CET 出力(12)がえられる。もしカードのジャムの具合がたいしたことがなければカードをセットしなおして

$\underline{: : \text{CR ERR}=}$R $\begin{matrix}T\\A\\B\end{matrix}$

と打鍵するとカードよみこみが再開する。このほか : : CR ERR＝はホッパ・エンプティ，スタッカ・フル，リード・エラーなどによっても出力するがいずれもエラーを起したカードはよみこまれていないので，Rを打鍵する時はホッパにセットするカードによく注意しなければ，入力ファイルに登録されるカードに脱落が生じてしまう。したがって，これらが起きた時スタッカからホッパにカードを1枚かならずもどさなければならない。ところで入力ユーティリティではカードの先よみを行なっているので，リード・エラーを起したカードの後のカードもカード・リーダを走ってしまうがこのカードに対する内部処理はまだ終了していない。この時は単にカードを1枚もどしてRを打鍵して処理を再開するわけにはいかず，実はスタッカからカードをホッパに2枚もどさなければならない。一般にRの打鍵により処理を再開するにはどのような状態から再開すべきかの判断が面倒であるので，なるべく最初からやり直すのがよい。ホッパ・エンプティ，スタッカ・フルもホッパやスタッカの状態を観察していて，それらの障害がおきる前にコンソール・タイプライタの割込みキイ1を押下して

*HL␣1 TAB

と打鍵する。すると実行中のプログラムは一時停止するので，ホッパにカードをつぎたすなり，スタッカからカードをとりさるなりしてから再びコンソール・タイプライタの割込みキイ1を押下して

*RS␣1 TAB

と打鍵すれば一時停止しているプログラムは実行を再開するので，できるだけカード・リーダの障害をさけた方がよい。なお，R以外のもの（Aをのぞく）を：：CR ERR＝ のうしろに打鍵した時には

：TYPE IN MISS

：：RETRY＝

と出力するので正しく打鍵しなおす。Aを打鍵するとそれまでよみこんだデータはすべてキャンセルしたうえ，よみこみを中止してジョブ・ステップを正常終了するので，入力ファイルに何ら送信データがないのにつぎのジョブ・ステップで CCP45 が起動されることになって不都合が生じるので，決してAを打鍵してはならない。このほか，入力ユーティリティからは

：IPTFIL FULL

というエラー・メッセージが出ることがある。これは入力ファイル領域をこえてデータを登録しようとした時に出るもので，このあとジョブ・ステップは正常終了するからただちにジョブをキャンセル（CN）しなければならない。また，誤ってコンソール・タイプライタの割込みキイ5を押下するとパラメータにM指定がないかぎりジョブはキャンセルされるので注意すべきである。

さて，第1のジョブ・ステップが正常終了すると図2の CET 出力(2)がえられる。ここでさらにカタログのジョブ制御文がよまれるので，第2のジョブ・ステップがはじまる。このジョブ・ステップでは CCP45 が起動されるが，EXEC 文のパラメータの BR のいみはつぎの通りである。これはリモート・バッチで送信用データを一括送信しそれが終了したら自動的にセンタ

ーからのデータを一括受信する状態になることをいみする。B と R は単独に指定することもでき，一括送信と一括受信をそれぞれ単独に行なうことを指示する。このほか W をくみあわせて，BW, RW, BRW という指定も可能である。W があると CCP45 が主記憶装置に格納されたあと W の前のパラメータにしたがって B, R または BR を行なわせるにはオペレータの介入が必要である。この時 CCP45 はオペレータの指令まちの状態にあるので，オペレータまたは利用者は，コンソール・タイプライタの割込みキイ 5 を押下して

::: S $^{\mathrm{T}}_{\mathrm{A}}$
$_{\mathrm{B}}$

と打鍵すればよい。CCP 45 がオペレータの指令まち状態にあってもセンターから何か送信されてくれば受信する。これを利用してこの状態でセンターの TSS サービス開始メッセージを受信することができる。CCP 45 が起動されると CET 出力(3)がえられる。ここからいよいよセンターへのデータ送信がはじまり図 2 の CET 出力 (4), (5) がえられる。もし送信中に何らかの原因により送信が不可能となったならば，CCP 45 はコンソール・タイプライタに

: S ERR

と出力しコントロール・ボックスの ATTENTIŌN ランプを点灯し，チャイムを鳴動する。この障害の回復はどのような状態で送信ができなかったかということに応じてそれぞれ適当な方法をとらなければいけないため説明が長くなるのでここで対応策を述べることは省略する。一般に送信不可能の時は原因をたしかめた上で最初からやりなおすほかない。送信中にオペレータの意志により送信を一時中断したい時は，コンソール・タイプライタの割込みキイ 5 を押下して

::: W $^{\mathrm{T}}_{\mathrm{A}}$
$_{\mathrm{B}}$

と打鍵すると CCP45 は処理を一時中断する。なお，これは CCP45 が動作しなくなることではないことに注意しておこう。この一時中断を再開するに

は，コンソール・タイプライタの割込みキイ5を押下して

::: R TAB

と打鍵すればよい。この時打鍵を誤って CCP45 にとって無意味な入力をすると

: CŌMMAND ERR

というエラー・メッセージが出力して処理の一時中断状態がつづく。また，送信中ディスクに障害が発生すれば，コントロール・ボックスの ATTENTIŌN ランプが点灯しチャイムが鳴動して，コンソール・タイプライタに

:: DK ERR=

と出力する。この時はジョブをキャンセル (CN) するほかないが，その前に R TAB と打鍵してデータのよみこみ再開を何度かこころみることが必要であろう。

ところで，CCP45 は直接カード・リーダから送信用データをよみこんでセンターに送信することもできるが，ここでそのような方法をとらずに第1のジョブ・ステップでディスク上の入力ファイルに登録したデータを送信することにしているのは，カードよみこみの時の障害の回復が端末の一方的操作のみで行なえないことによる。端末から一括送信をはじめるとセンターは端末を SYSIN 状態として保持し，¥END 文が送信されてくるまで入力をつづける。入力したデータを端末からキャンセルする手段は全くないし，カード・リーダからカードをよみながらセンターに送信している最中にカード・リーダに何らかの障害がおきてカードよみこみがとまり，したがって送信が中断しても SYSIN 状態は依然としてつづいている。ここで SYSIN 状態を打切るには ¥END 文を何とかしてセンターに送信しなければならないのであるが，たとえ ¥END 文を送信しても途中までセンターに送信したデータはキャンセルされないでそのままセンターにのこる。さらに，もし ¥END 文の送信が不成功に終れば端末からはもはやほどこすすべがない。また，リード・エラーの場合の再開も前述のようにやや判断が面倒である。そこで，カード・リーダから直接送信することをやめて，第1のジョブ・ステップで

まず送信用データをディスク上の入力ファイルに登録するようにすれば、どんなカード・リーダの障害がおきてもただちにそのジョブをキャンセルすればディスク上の入力ファイルも抹消されてしまうので，もう一度新しく最初からやりなおすことが可能となる。実はセンターから受信したデータをライン・プリンタに直接出力するさいにも同様の問題がおきるが，ライン・プリンタの障害はカード・リーダよりはるかに少いのでまず問題はない。また，ディスクの障害率もカード・リーダよりはるかに低いことはいうまでもない。これが送信用データをディスク上の入力ファイルにいったん登録する理由である。

送信中に障害が起らず，¥END 文までの送信が終了すれば，CET 出力(6)がえられる。このあと CCP45 はただちに一括受信状態に移行し，CET 出力(7)がえられる。通常はこの状態から SYSŌUT 状態になるまで多少の時間が経過する。いよいよ SYSŌUT がはじまれば，まず，CET 出力(8)があったあとライン・プリンタにセンター出力とほぼ同じ形式で出力する。ひとつのジョブの出力が終了すれば CET 出力(9)がえられるが，なお出力すべきものがあれば CET 出力(8)からはじまる過程が何度でもくりかえされる。こうして端末は一括受信状態と SYSŌUT 状態の間を何度も往復する。そして，センターに端末へ送信すべきデータがなくなっても端末は依然として一括受信状態をつづけている。したがって，ここで問題になるのは，いつ一括受信状態を終了させたらよいかということである。一括受信状態を終了させるにはコンソール・タイプライタの割込みキイ 5 を押下してから

: : : ŌFF $\begin{matrix}\mathrm{T}\\ \mathrm{A}\\ \mathrm{B}\end{matrix}$

と打鍵するが，その打鍵のタイミングの判断はもっぱらオペレータまたは利用者にまかせられている。そこで，オープン利用者は端末に出力されるべきものが全部センターから送信されてきたかどうかを慎重に判断して適当な時期に ŌFF を打鍵しなければならない。ŌFF を打鍵すると CET 出力(10)がえられる。なお，SYSŌUT 中にライン・プリンタに障害が起るとコントロール・ボックスの ATTENTIŌN ランプを点灯し，チャイムを鳴動しコンソ

ール・タイプライタに

：：LP ERR＝

と出力し鍵盤入力が可能になるので，R $\begin{smallmatrix}\mathrm{T}\\ \mathrm{A}\\ \mathrm{B}\end{smallmatrix}$ を何度か打鍵してみてそれでもライン・プリンタが回復しなければジョブをキャンセル（CN）するほかない。また，センターから受信したデータに誤りがあった時もコントロール・ボックスの ATTENTIŌN ランプを点灯し，チャイムを鳴動しコンソール・タイプライタに

：R ERR

と出力する。この時はチャイムの鳴動をとめしばらくそのままの状態で放置しておくと出力が再開する。出力が再開しないで：R ERR という出力がつづくならジョブをキャンセル（CN）しなければならない。受信中の一時中断と再開は送信中の場合と同様である。

以上でリモート・バッチ・ジョブは終了したことになるが，実は CCP45 そのものの実行はまだ終了していないのである。これを終了させるにはコンソール・タイプライタの割込みキイ 5 を押下して

：：：E $\begin{smallmatrix}\mathrm{T}\\ \mathrm{A}\\ \mathrm{B}\end{smallmatrix}$

と打鍵する。すると，CET 出力(11)がえられる。あとは図 2 のフロー・チャートにしたがってライン・プリンタから計算結果の出力をとりさればよい。もし，ŌFF を打鍵しないで E を打鍵すると初回だけコンソール・タイプライタに

：CŌMMAND ERR

と出力して，E の打鍵はいみをもたない。しかし，2 回目から E は正常に受付けられて，CCP45 は終了する。それにもかかわらずこの時は端末の一括受信状態は終了していない。したがって，ŌFF の打鍵をわすれてはならない。

ジョブ制御文を EXEC 文のカタログによって指定しない場合は，カタログのジョブ制御文をすべてカードにパンチする。ただし

//DD␣IPTR＝CR

は

//DATA␣IPTR

としてこの後に送信用データ・カードをつける。したがって，カード・デックの構成は，JŌB 文，INUTT の EXEC 文，DD 文，DATA 文，NŌ 文から ¥END 文までの送信用データ・カード，ブランク・カード，CCP45 の EXEC 文，DD 文，/# カード，ブランク・カードという順序になる。

4. あ と が き

データ・ステーションからのセンター利用は利用者がセンターに出向いて行なうセンター利用と全く同等ではなくかなり制限がある。たとえば，端末からのジョブで磁気テープが使用しえないというのはその一例である。これらについてはセンターの刊行物[4) 5)]を参照されたい。また，CCP45 はリモート・バッチ・ジョブの内容には全く関係しないことはいうまでもない。したがって，送信，受信に関する事柄以外は一切が利用者の責任である。このため利用者はたえずセンターのアナウンスメントに気をつけていることがのぞまれる。

---

4) 北海道大学大型計算機センター利用の手引，北海道大学大型計算機センター・ニュース，Supplement NŌ. 5 (1973年 8 月).

5) TSS 利用の手引（暫定版），北海道大学大型計算機センター，1973年10月.