

端局の利用について

竹田, 宏
九州大学教養学部物理教室

<https://doi.org/10.15017/1467987>

出版情報 : 九州大学大型計算機センター広報. 4 (4), pp.2-7, 1971-08-02. 九州大学大型計算機センター
バージョン :
権利関係 :

端局の利用について

竹 田 宏[※]

端末機という、私ども「遠隔地」の利用者にとってまことにありがたい(はずの)機械が、わが九大教養部に設置されてから半年になる。はじめのうちは10分動けば20分ダウンするといったありさまで、私のようなよほどの物好き以外は、とうてい使う気が起こらない状態であったが、最近では、少なくとも利用者の立場から見限りでは、システムも安定し、端局がわれわれにとって真にありがたいものになりつつあるのは嬉しい。そこで私がこれまで端局を使って感じてきたことを述べるとともに、経験から体得した「端局利用の我が家の秘伝」を公開して、これから端局を利用される方々のご参考に供したい。

(1) デマンド・ジョブ

端局の利用には大きくわけて2種類ある。一つは今から述べるデマンド・ジョブであり、他の一つは次に述べるリモート・バッチである。デマンドは「会話型処理」などと言われるようだが、そんなことを言われても私ども初心者にはピンとこない。要するにここではBACCUSという特別の言語でプログラムを書くのである。この言語はALGOLによく似ていて、AL-GOLによるプログラミングの経験者ならBACCUSの文法書を読まなくても、センターで配布している「T S S利用のための手引」の中の例題をみるだけで、大体のところは理解されるであろう。

さて、このBACCUSで書いたプログラムを端局のタイプライタから入力するのであるが、もちろん一つの命令を入れるたびに機械が直ちにこの命令を実行するわけではない。プログラムを全部入れ終って、実行開始の命令を入力して始めて計算が始まるわけで、この辺のところはFORTRAN などと同じである。(もともとBACCUSには直接命令というものがあって、それらの命令はそれを入力した時点で実行される。上述の実行開始命令がその一例である。)命令の入力中、例えばミスタイプなどで機械がその命令を理解しえなかったときは、すぐ「EH?」と聞き返してくる。そこで正しく打ち返すと、前の間違った文は正しい文で置きかえられる。また「EH?」と聞かれても何を間違えたかわからないときは、逆にこちらから聞き返す(「?」を入力する)とエラーメッセージが印字される。なるほど「会話型」とはこういうことかとわかったような気になる。

このような場合に限らずプログラムの修正はいつでも簡単にできる。すなわちBACCUSの命令にはすべて番号、それも整数ではなく小数の番号がついていて、計算の実行は命令を入力した順序ではなく、番号の小さい順に行なわれる。だから命令を追加したければ前後の命令の中間の番号をつけて入力

※九州大学教養部物理教室

すればよいし、またすでに使われている番号と同じ番号の文を入れると、古い文は新しい文で置きかえられる。もちろん不要になった命令を削除することもできる。それから実行を開始し、計算が一通り終って結果を打ち出し、FORTRANのSTOP文にあたるPAUSE文までくると、機械は一時停止するが命令や数値は記憶されたまま残っているので、出てきた結果を見て、もし必要とあればすぐにいま述べた方法でプログラムを修正し、再び実行開始の命令を出せば、たちどころに修正されたプログラムでの結果がえられる。なるほどこれも「会話型」ということかと、ますますわかった気になる。

このあたりがデマンドの最大の特長であるが、一方BACCUSではあまり複雑な構造のプログラムは組めないという短所もあるので、そこをよく心得て高性能卓上計算機ぐらいのつもりで気軽に利用するというのが、デマンドをうまく利用することとでも言うことになるだろう。SQRTやEXP等の組み込み関数はもちろんあるが、FORTRANのSLLサブルーチン呼び出して使うこともできるので、例えば行列の固有値を求めたり、連立一次方程式を解いたりするには威力を発揮するだろう。さらに、自分で書いた(FORTRANで)サブルーチンでも、前もって機械語に翻訳された形(RB)で専用ファイルに登録(記憶装置に記憶させておくこと)しておけば、デマンドで呼び出して使えるから、相当複雑な計算でもさせられるかもしれない。ただこの場合のサブルーチンには、いくつかの条件がつく。例えばメインプログラムとの間にCOMMON変数で情報の受け渡しをしているようなものはだめだし、入出力命令を含むものもだめである。ただしその入出力命令を適当な条件で飛び越すようになっていれば使えるようである。一つのデマンドで呼び出すことのできるサブルーチンはただ一つだけであるが、この呼び出したサブルーチンがその中でさらに他のサブルーチン呼び出すのは構わないし、またそれらサブルーチン相互の間でCOMMON変数を使うことも差支えない。もっともこれらのことは私も一度ためしただけだから、あるいは間違っているかもしれない。というのは、ある日そういう複雑怪奇なことばかりやらせていたところ、突然計算機が怒り出し、タイプライタがガチャガチャ言い出したかと思うと、全く無意味な記号をとめどなく印字し出したのである。驚いて割り込みをかけようとしてもテンデ受け付けてくれず、ホトホト困り果ててとうとうセンターに電話してコンソールからの指令でABORTしてもらおうという破目になった。それにこりて、以後そのような「デマンド・ジョブの曲芸的利用法に関する実験」は二度とやっていない。

そんなこともあるので、デマンドの実行にあたっては「強制打ち切り」の方法をいつも頭に入れておかねばならない。もちろんすなおな利用法をする限り暴走の心配は少ないが、それでも格言にもあるように、計算機はプログラマが思った通りに動くのではなく、書いた通りに動くものであるから、思わぬ事態が発生して早くこれを打ち切らねばと思うこともあろう。そのときになってあわてて「手引き」を開いても必要な記事はなかなか見付からず、そこで苦しまぎれにいろんな指令を送ってみることになるが、機械からは「こんな指令は今は受け付けられない」という世にも無情な返事が返って

くるだけで、異常事態はちっとも止まってくれないのである。全く、魔法をかけたのはよいが、まだその魔法を解く術を習っていなかったという魔法使いの弟子の話さながらである。しかし強制打ち切りの指令を出してもABORTされなかったことがだいぶ前だが二三度ある。このときは機械はだまり込んでしまって、いつまで待ってもOFF LINE(一仕事終って、次のジョブが要求できる状態)になってくれない。そんなとき端末機の電源スイッチなど切ってみても何にもならないが、システムダウンだとすべてはご破算になるので、このときばかりは早くダウンしないかと秘かに祈りを捧げたりした。しかし、この祈りはついに神にも悪魔にもとどかなかったようである。

(2) リモート・バッチ

これはFORTRANでもALGOLでもとにかく利用者がカード・デッキを持って計算依頼に行く代りに、そのカード全部の内容を端局から入力して計算依頼をするということである。たくさんのカードをタイプすることは大変だし、またデマンドと異なり一たん間違った文を送ってしまうと取り消しや修正はできない。一つの文を打ち終ると(改行)・(復帰)という2つのキーを押すのだが、この(復帰)を押す前にミスに気がつけば修正することができる。しかし(復帰)を押したあとではどうにもならない。そこで必要なプログラムやデータを前もって私用ファイルに登録しておき、あとはこれらのプログラムを呼び出すための制御文やFD文、あるいはまだ登録してない短いプログラムだけを入力してやればよい。このようなファイルの利用についてはまだシステムに不安な点もあり、また私どもも不慣れのため制御文を入れ違えることもしばしばで、せっかくの計算依頼が目的を達しないまま返されることが多いのは残念である。それにしてもカードを何箱も抱えて行くところを、わずか数十枚、いや十数枚のカードで事がすむのだから、全く「いちど使ったらやめられない」(京大センター広報 vol. 4 No. 2)ということになる。このようにファイルの利用は端局の利用と切りはなしても非常に便利なものであるが、逆にファイルの利用を抜きにした端局の利用はあまりぱっとしない。しかし一方ファイルの利用と結びついた端局の利用は、遠隔地に居ながらにして計算依頼ができるわけで、こんなありがたいことはない。実際、リモート・バッチがうまくいった日は、日頃のストレスも一度に解消する思いである。

このようなありがたいリモート・バッチにもまだまだ欠点は多く、実はストレス解消どころかストレス増大の要素も多分に含んでいる。実際、気の短かい人(私もその一人だが)がリモート・バッチをやればイライラさせられ通しということが多であろう。デマンドの場合は、当然のことながら、入力に対する計算機の応答は極めて速く、待たされることなどほとんどない。せいぜい一番始めのBACCUSを入力したときだけ10秒ほど、特に私用ファイルのサブルーチンを利用したいときなどは1分ほど待たされるだけである。他の応答が余り速いので、この10秒でさえ長く感じられるのだが、リモート・バッチとなるとそんなものではない。入力が終わってから出力が開始されるまでふつう2分

かかる。ときにはすぐ出力が始まることがあるが、それはたいていジョブ制御文にエラーがあってABORTされるときだから喜んではいけない。しばしば10分ぐらい待たされるし、ときには1時間も待つことがある。それでも結果が出てくれればまだいいのであって、サービス時間終了まで待ってもついに結果が出なかったということも何度もある。出力を待つ間は何をしてもよい（もし望むなら電源を切って休憩してもよい）ので、例えばデマンドの練習でもしていれば気はまぎれるがそんなことをするとバッチの処理はますます遅れるのかも知れない。

さて待望の出力が始まっても、それでイライラが解消するわけではない。こんどはタイプライタのスピードのノロさが気になるのである。現在NHKの教育テレビでFORTRANの講座をやっている、この番組の後半では、できたプログラムを端局を通して機械にかけているが、そのときのタイプライタの目にも止まらぬ動きを見ていると、私などはただ羨望のタメ息が出るばかりである。この番組を見た人がそれと同じつもりで九大の端局を利用されるならば、そのスピードのノロさに驚嘆されるに違いない。しかし印字の高速化などは望むべくもないので、利用者としては出力量を極力少なくし、印字はなるべく左につめて打ち、FORMATは1行80字以内にしておけば（タイプの1行は80字だから）出力に要する時間を少しでも節約できよう。1行80字以上のFORMATだと80字目に自動的に改行するが、この改行は一つの数値の途中でもおかまいなしに行なわれるので、出力は大変読みづらくなる。FORMATで行間をあけるようにしていても無視されるが、改頁の指定に対しては区切りのための横線が印字される。あまり正確なデータではないが、LP用紙1頁分の出力（といっても空白をあけることの好きな人とそうでない人とで大差があるが）に30分ぐらいかかるようである。だから出力の最中にサービス時間が終るのではないかとハラハラしながら時計とニラメッコしたこともある。よくよく運の悪いときは出力中にダウンという泣きたくするような事態も起きるが（いっそ出力開始前のダウンならさっぱりする）、ダウンと全く同じ症状（タイプがパタリ止まり、開始ボタンを押しても応答なし）なのに実はダウンでないということもあるので早まってヤケを起こしてはいけない。5分ほど待って開始ボタンを押してみても応答があればまだ脈はある。そこで「R(受信の指令)」を入力すればそれがきっかけで出力が再開されることもあるし、それでもだめなときは、余りいろいろな指令を出すあとでタタリがあるので、また数分してから改行、復帰のキーでも押してみるとよい。運がよければそのうち出力が再開されるであろう。真のダウンのときは「5555……」という字が半行ぐらい出ることになっているが、これが出るのはダウンしてから10分ほどしてであるから余り役に立たない。しかし出ないよりまだから止めないでほしい。

ファイルの利用に必要なFD文はとかく長くなってカード1枚に入らないことが多く、継続欄を使わねばならなくなるが、それがFORTRANと違って72欄である。ところが印字用紙は白紙であるため、どこが72欄なのか容易にはわからない。72字目の所に薄くたて線を入れた印字用紙がほしいと言

っていた人があるが、私も同感である。75字まで進んだときリンが鳴るので、そこから3つ後退させればよいのだろうが、私はむしろローカルボタンを押して端局と計算機の連絡を一時切り、1行一杯に123-9012……と印字しこれを目印にしている。ところが目印をつけてもまだ間違えることがある。それは途中ミスタイプを修正したときで、間違った字の上に正しい字を重ね打ちして修正したときはよいが、(訂正)のキーで消したときはこの字がノーカウントになるため、タイプの印字が72欄になっても、そこは実は71欄なのである。

幸いにも(?)TSSはまだ実験期間中なので計算機使用料がいらぬという特典がある。そこで日頃から「こんなことをしたらどうなるだろうか?」というような疑問のあるむきは、この実験時間を利用して小さなプログラムを書いて実験してみるとよい。私もこの4月にシステムが変更になってファイルが利用できなかった時期には、例えばLIEDの制御文の書き方といったまじめな勉強もずいぶんさせてもらったし、また(こんなことがセンターの人に知れると怒られるかもしれないのでこれは内緒だが)バッチにおける曲芸的利用のイタズラなどもだいぶやってみた。またあるときALGOLの練習を試みたが、簡単なプログラムがさっぱり通らない。いろいろ調べているうちに、どうやら「:=」の記号を計算機がまともに処理してないのだということがわかった。そこでために「:=」の代りに「=」など入れてみたがもちろんだめ。ふと思いついて「:」と「=」との2つに分けて入れてみたらやっとう通った。気がついてみれば当たり前なのかも知れないが、タイプライタにはなまじ「:=」のキーがついているばっかりに手を焼いてしまった。似たようなことだが、リモートバッチではジョブ制御文の第1字は「\$」ではなく「¥」であることを念のため注意しておく。

終りに、これら2つのジョブに共通した注意を述べよう。それは、慣れるまでは実に呆れるほどのミスタイプをすることを覚悟しておかねばならないということである。あまり大きな期待をもって端局の実験を始めると、恐らく余りのミスタイプに驚いて、「こんな不便なもの2度と使うものか」という感想を抱かれるのではないだろうか。私はある計算機の専門家から現在のTSSなど実用からはほど遠いということを知っていたので、始めての実験でも余りショックを受けなくてすんだ。ただしこの話を聞いたのは3年ぐらい前のことで、日進月歩の計算機の世界のことだから、その方が今どう考えておられるかはわからない。九大センターの利用者の大部分を占める箱崎地区の利用者にとってはこの「不便なもの」という評価も当たっているかも知れない。しかし遠隔地の利用者は計算機の利用についてはかねてから多大の不便を感じておられるので、端局利用の不便さも相対的には小さく感じられるであろう。したがってこのミスタイプでいやになる時期を乗り切るのも易しいのではないだろうか。

ミスタイプのほとんどは上中下段のシフトの誤りと、送信ランプがつく前にキーを押すことである。送信ランプというのは、ついているのがふつうなのだが、端局から計算機の方へ情報を送っていると

き、すなわち一つの文を打ち終って（復帰）のキーを押すと消え、再びつくまで2、3秒かかるのである。これらのミス避けるには、入力すべき文を前もってテープにパンチしておけばよい。端末機のテープ・リーダーは送信ランプが消えると自動的に停止し、点灯すると動き出すので、テープにミスパンチさえなければ入力中はテープのもつれにだけ気をつけていればよい。テープにミスパンチがある場合はそこだけ手入力をすればよいが、そこだけというのはミスパンチしたその字だけということではなく、その字を含む文全部ということである。文の途中でテープから手に切りかえたのではうまくゆかないようである。一つの文の途中で余り何度も修正を繰り返していると、「端末からの入力を読めない」というメッセージが出て、ABORTされることがある。

段のシフトはまことにうるさいもので、英数字や「・」、「,」、「/」は上段、「¥」、「=」、「(」、「)」、「*」は中段、「+」、「-」は下段である。ただシフトの誤りは場合によっては正しく処理されることがあり、「利用の手引」の暫定版の中のデマンド・ジョブの例題では、シフトを間違えても正しく動いている所があったが、現在のものではそこが正しいシフトに直されてしまっている。私自身の経験では、FORTRANのプログラムを打っていて、STOPというのを中段で打ってしまったことがある。いつもだと入力ミスをしたときは、そのあとわざと間違ったジョブ制御文を入れて早いとこABORTしてもらおうのだが、このときはプログラムの最後ではあるし、淡い期待をもってそのまま入力が続けてみた。すると計算機はちゃんとこの期待にこたえ、正しくSTOPと読んでくれたのである。データは、ほとんど上段だけで用が足りるので、送信ランプのことさえ注意すればミスはほとんどしなくてすむ。

以上思いつくままにいろいろ述べてきたが、何しろ私の経験といってもわずかなものに過ぎないから、間違っていることもあるかもしれない。また中には未熟者が何を言うかとお気にざわったむきもあるかも知れないが、その辺のところはご寛容をお願いしたい。また、特にリモート・バッチについては、あくまでも遠隔地の利用者が3原則（1.入出力は端局からのみ。2. Aジョブ。3.40KW以内）を固く守るという立場で書いたので、センター内の端末機を利用されている人や、原則は破るためにあると思っておられる人達から見ると、だいぶ感覚がずれているところもあろう。遠隔地利用者からみて現在のTSSに対する最大の不満は、リモート・バッチの出力に時間がかかりすぎることである。出力開始までに時間がかかるのは始めから覚悟していれば何でもないが、印字に時間がかかるのはかなわぬこととは知りながらも何とかならないかと思う。かってセンター内の掲示板に「TSS時のラインプリンタへの出力リストは翌日取りにすれば渡す」という原則1と矛盾する掲示が出されたことがあり、実は私もときどき遠慮がちにはあるが出力をLPへ出して翌日もらっている。もしもこのLPへの出力が認められ、また真の遠隔地利用者にはリストを郵送するというにでもなれば（同時にファイルに関する安心できるシステムの開発も必要だが）、遠隔地利用者のささやかな夢がかなえられることになるのではあるまいか。