

## プログラム作成テクニックについて\*

和田 英 一\*\*

学会の編集幹事会からプログラムテクニックについてなにか会誌にかけという注文である。これはかつて東大大型計算機センター広報(14号)に「200倍スピードアップの記」と称する雑文をかいたのが、幹事の一松先生の炯眼にとまり、なにか秘法でもあれば公開せよとの意向だったかにおもう。200倍スピードアップとは、当時大型計算機センターと併任で(この方はさる7月めでたくびになった。)いろいろもちこまれるプログラムの診断をさせられていたとき、HITAC 5020 E で10時間かかる(自称)のプログラムがもちこまれた。これをプログラムを改良してかけたら3分(正確には175秒)で結果がえられたというはなしである。10時間がどのくらい正確か不明なので、200倍にもその程度の不確かさがあるが、こんなことをかいたために、物議もかもしてしまった。このスピードアップには別にたねもしかけもなく、あたりまえのことをしただけだが、最近われわれがひそかに森口語録とよんでいる森口先生の「JIS FORTRAN 入門」が巷間にでまわり、情報処理大革命をになう FORTRAN 紅衛兵が激増した結果、あたりまえの処理さえもされていないプログラムで大型計算機センターにもちこまれるのがかなりあるようだ。(むかしはもっと情熱をかたむけてプログラムをかいていたとおもう。)

問題となるプログラムには、結果がでないプログラムと、結果はでるけれどまずいプログラムとがある。結果がでないといってももちこまれたプログラムの最たるものに、WRITE文のかきわすれというそそっかしいのがあったが、(ご本人の名誉のために付言すれば一緒に使用したライブラリープログラムがウォーニングメッセージをだしたため、結果がでないのはそのせいだとかんがえたようである。)これは落語研究会で検討してもらうことにして、FORTRAN プログラムでもっとも気をつけなければいけないのが「場外ホームラン型」すなわち DIMENSION 文で配列宣言した

そのそとになにかかいてしまうものである。どうしてこうなるかという、配列はあるきまったおおきさにきっておき、データをそこによみこみたい。そのデータの個数もまたデータカードからよむというようなことをするのが悲劇の発端である。この対策としては、よみこんだデータはかならず印刷するよう勧告している。そうすればどうしてエラーになったかはやくわかるからである。それもよみこんだらすぐに。この勧告にしたがって、すべてのデータをよみこんでからそれを印刷した利用者があったが、ときすでに悲劇はおわっていて、下手人も犠牲者も区別つかなくなっていた。

計算機センターは出力量に対しても料金をとるせいか、大体は印刷をきらう傾向がある。余計なものを印刷するのは悪徳とおもうらしい。プログラムのどこかで無限ループにはいつか時間ぎれで終了したというのをただもちこまれても大変こまる。サブルーチンにとびこんだら、せめてその名称でも印刷するようにデバッグ中のプログラムはつくるのがよいと思う。こうしてどのサブルーチンのなかで堂々めぐりをしているかがわかったらそこに DEBUG をいれればよい。プログラムはこれで完成したとおもっても、まだまだエラーが沢山あるのがふつうだから、これらのデバッグ用の印刷はとりのぞいてしまわずにいれたままにしておき、簡単な操作で印刷する、しないがきりかわるのがよいとおもう。われわれのところの LISP は制御カードの CHECK, UNCHECK で、JOSS は名前付共通ブロックをひとつ用意してその BLOCK DATA 文による初期値設定で、デバッグ情報の制御をおこなっている。この種のプログラムは年中あちこちかきなおしがあるので、デバッグ用の印刷をとりのぞくことは不可能である。

中間の印刷は長大なプログラムの場合、進行の様子をみるために必要である。最初にのべた10時間かかるプログラムをみておどろいたことは、10時間フルに計算するあいだはなにも出力せず、計算がすむと一挙に出力するというプログラムの構成である。かりに10時間のランがみとめられ、10時間半くらいのうちきり

\* How to improve your program, by Eiiti Wada (Faculty of Engineering, University of Tokyo)

\*\* 東京大学工学部計数工学科

時間でランをしてみても、10時間の予想がすこしちがって11時間かかるとすると、なんの結果もえられず、どこまで進行したかもわからず、10.5時間の計算時間を無駄にってしまう。すこしはデータ処理のチェックポイント(リランポイント)の精神をとりいれたらよいとおもう。それにはプログラムをする努力が必要だが10時間無駄にすることをかんがえれば努力のしがいもある。センター長の高橋先生にこんなことをお話ししたら、「最後の結果のチェックのためにも、途中でときどき中間結果を出力しておくのがよからう」といわれた。おりあって教養学部の小野周氏の「1時間くらいかかる」と申請してあったプログラムのなかをみせていただいた。小野氏は大型計算機センターの研究開発部長でいらっしやるので、さぞ模範的なプログラムだろうとおもったが、やはり全部の計算がすんでから出力するという構成であった。

このあたりで200倍スピードアップの秘伝を披露すれば、対称性の考慮と、おなじ計算のくりかえしの省略である。もともと対称性を考慮して、全体をいくつかの共通の性質のグループにわけ、そのグループごとの計算をして、グループに属する場合の数をかけてくわえればよいことがわかっていたのだが、計算機はとてつもなくはやいから、計算機の馬力で全部のケースをしらみつぶしにかぞえあげようとしたのが、そもそもこのプログラムの時間のかかっていた原因であった。これを丹念に分解したが第一、これはしかし第二の改良よりは適用例がすくないのであろう。第二は後述するとして、簡単なプログラムで時間をかけて実行する方が、プログラムのエラーがすくないことは事実のようである。現に前述小野氏のプログラムも、内容は10時間かかるはずのプログラムと同類であり、この方は一応おなじパターンのもはまとめて計算するようになっていたが、みせていただいた段階では3箇所ばかり加算すべき項がぬけていたように記憶する。

第二の改良点は10時間のプログラムでは特に極端であったがつぎのようなものである。プログラムには16重の完全入れ子のDO文があり、16箇の制御変数が1から3までかえられる。つまり一番なかのDOの範囲は $3^{16}$ 回実行されるわけである。(実際は、HITAC 5020のFORTRANは15重のDOの入れ子しかできないから一番そとがわは普通のIF文でかいてあった。)ところで16箇の制御変数をI1, I2, …, I16としよう。驚異なのは一番なかのDOの範囲にI1+I2

+……+I16な算術式があらわれることである。I1が一番そとの制御変数なら、I1+I2+……のはじめの方はほとんどかわらないのに、毎回たしなおすのはまったく無駄である。そとから2番目のDOの範囲の最初でI1+I2を計算してJ2にいれ、つぎに3番目のDO文をかき、その最初でJ2+I3を計算してJ3にいれ、……というようにし、一番なかのDOの範囲ではJ15+I16がI1からI16までの和になるようにする。これは第一の修正よりはずっと容易である。ふつうのプログラムはDOの入れ子がこれほどおおくはないのできして問題とならないかもしれないが、おおいプログラムをつくる時には注意したい点である。ちなみにたびたび引用する小野氏のプログラムはやはり一番なかにはいってから全部の制御変数をつかって計算をする形式であった。

計算機のスピードは決して無限大ではなく、ひとつでも演算をすればそれだけ時間がかかる。それがうへの例では比較的わかりよいはずだが、それでも利用者はなんとか努力してスピードアップしようとはしないようである。したがってもっとかくれた部分まで気をくばってプログラムを改良しようという奇特なひとはよいよまれである。この種に属する改善は仮引数を有する副プログラムに例をみることができる。副プログラムに仮引数がなん回もあらわれるプログラムを案外みかけるが、これはHITAC 5020のコンパイルされたプログラムでは、実引数のアドレスを仮引数の部分にさしてむ命令が沢山つくられ、かつ副プログラムにはいるたびにこの部分が実行されて、やはり損である。なん回もでてくる仮引数に対しては、副プログラム内に変数をとって、入力側の変数については副プログラムのいりくちで、出力側の変数についてはでぐちで代入をおこなうのがよい。このことは名古屋大学の近藤氏がずい分まえに指適されていた。

利用者のプログラムがすべて名人芸の域に達し、無形文化財が激増するともおもわないが、しかし大型計算機センターその他の計算能力が不足している今日、プログラムの改善にすこしでもちからをつくすべきとおもう。そうならないのはいろいろな意味での「プログラマーパワー」の不足らしく、ほこさきはこちらをむいているような気がしないでもない。プログラマーパワー養成には計算時間がふんだんにいるし、どこでこのループからとびだしたらよいだろうか。

(昭和43年10月5日受付)