

寄 書

割り出しの方式について*

高橋 秀俊** 亀田 壽夫**

いろいろの人のプログラムを一括して、順番に自動的処理をするために、また、時分割使用のために、主記憶装置に常駐するモニタというプログラムが使われる。その場合に、一般使用者のプログラムが、モニタ・プログラムのはいつている記憶場所を書きこわしたり、周辺装置を直接動かす命令を出して、システムを混乱させたり、何もしない停止状態にはいつたりすることを防ぎ、そのような場合、制御をモニタに移すために考え出されたハードウェアの機構を“割り出し”と呼んでいる。割り出しの機能（特にメモリ保護など）は、単に、モニタを保護するというだけでなく、使用者プログラムの誤り検出にも、非常に有効なものであることがわかってきた。

すなわち、複雑な大きいプログラムを完成するのは容易でなく、一応“通った”積りでも、時々悪い動作をし、その原因がわからないということがよくある。特に困るのは、知らない間にプログラムの一部を書きこわすという誤りで、その場合には書きこわしがあって、しばらくしてからでないと、悪い動作が表面に現われない。そのようなときに、プログラムの誤りを検出するのに、割り出しはきわめて有効なわけである。

初期にモニタの概念が導入されたときに採用された方式では、計算機には privileged mode, normal mode の二つの mode があり、前者では割り込み、割り出しは抑制され、また、privileged instruction と呼ばれる幾つかの命令は前者でだけ働き、そしてモニタはいつも前者で動作するようになっていた。

ところでモニタが privileged mode で働くということは、モニタの誤りに対しては保護がないということである。もちろん、モニタはよく注意して作られてい

るはずであるが、速く作る必要にせまられるためなどから、実際には誤りが残る可能性が多い。中でも、一番始末に困るのは、モニタ自身を書きかえてしまうような誤りである。そこで、これを避けるために、普通はモニタについても、privileged mode で働く部分を、できるだけ少なくするという手段がとられるが、そうすると、入出力などの privileged instruction を簡単に出すことができなくなり、また、privileged mode で働く部分に、やはり問題が残る。最近の大型計算機では、もっと多くの mode が設けられてあり、そのような困難は避けられるようである。

ところで、モニタによる動作は、いままで主として大型の計算機に採用されていたが、中型機以下でも、処理速度が向上し、また、補助記憶装置やディスプレイなど、外部装置がいろいろ付けられるようになるであろうことを考えると、モニタ動作が必要になってくる。

その場合、小型・中型機にも割り出しの機構が必要となる。一方、これらの計算機では、普及性や用途の広さを考えると、ハードウェアの設計はできるだけ簡単なことが望ましい。

そのようなわけで、小・中型機にも使える割り出しの方式として、つぎのような方式を提案する。

すなわち、privileged mode では割り込みは禁止するが、メモリ保護などの割り出しは、normal mode のときと同様に起こるようにするのである。また、privileged instruction (メモリ保護境界レジスタをセットする命令も含めて) は、やはり privileged mode だけで用いることができるものとする。このようにしてもモニタは、書きうる領域を、自分が書く積りの最小限の範囲だけいつもひろげておいて、読み書きを行なうようにすれば、支障は起こらない。これにより論理的な redundancy が具現化されたことになり、error detection が容易になるというわけである。

* A Note on Trapping System of Computers, by Hidetosi Takahashi and Hisao Kameda (Dept. of Physics, Fac. of Science, Univ. of Tokyo)

** 東京大学理学部物理学教室

一方、この方式にすると、幾つかの注意事項が出てくる。まず、この方式では、上限と下限とを別々に指定するメモリ保護境界レジスタが必要である。どちらか一方では非常に不十分である。また、一般に、normal mode で働くプログラムのために、ソフトウェア・システムとして、許された範囲内で、メモリ保護の様相を自由に変えるためのモニタ・コールを設けてあれば、debug に役に立つであろう。特に、アセンブラを用いる場合に、このことは重要である。また、この方式では、実行に関する保護は、常に命令カウンタを調べる方法よりも、jump 命令の行先番地だけ調べるやりの方が簡単であり、また、それで十分であるといえる。命令カウンタを調べるやり方では、モード

が変わるときに、たとえば、同時に保護境界レジスタを入れかえるようなハードウェア機構が必要になる。ついでながら、大きな計算機で segment 化するためのハードウェアが付いているものでも、1 segment 内のメモリ・アドレスは、論理的には linear であるので、上のように、メモリ保護の上限・下限を制限内で自由に変えられるようにすることはプログラムの誤り検出を容易にするために有効であろう。

なお、privileged mode でも、メモリ保護の割り出しが起りうるシステムについては、M. V. Wilkes 教授も賛同されたことを付記する。

(昭和 43 年 6 月 7 日受付)