寄　　書

プロセス・コンピュータのソフトウェア*

木　下　常　雄**

1. プロセス・コンピュータとリアルタイムプロセス

近年，コンピュータの分野では科学用コンピュータと事務用コンピュータのほかに，プロセス・コンピュータが大きな一つの分野として成長してきた．科学用・事務用コンピュータが，それぞれの対象を持っているように，プロセス・コンピュータもはっきりとその目的，用途，構造を持って生れた．今は汎用コンピュータが出まわってきているので，それぞれにあまり区別はなくなっているかもしれないが，プロセス・コンピュータはリアルタイム・プロセスシステムという形で成長している．

いうまでもなく，コンピュータの動作速度は非常に速いものだが，リアルタイムシステムを採用しないかぎり，突然起こる外界の変化や要求を受け入れながら，その都度最適な制御や解答を出す柔軟なプログラムのジェネレーションは困難である．

計算機の能力の中心はオペレータの指定に従った演算の実行という機能から，外部条件，すなわち生産量，原料，周囲条件，エネルギー源などの変化に基づく割込機能の管理をするということに移ってきている．

リアルタイム・プログラムの特徴をあげてみると，第一に各プログラムは割り込みという手段で開始，停止を行なうので，外から見ていると細切れに実行されるということである．割り込みが発生したとき，その優先順位に従ってプログラムが実行されるということ，内部データ転送が何回かの動作をつなぎ合わせて完了するということ，したがって各割込発生時にそれまで行なわれていたプログラムの必要な情報をしまっておくテーブルがいくつか用意されてこのテーブルの扱いによけいな時間を要することなどが考えられる．

第二にリアルタイム・プログラムの中では能動的にプログラムが動作し，リアルタイムが守られ，あらかじめ決められた時間割に従って，あるいはこの時間割を更新しながら流れて行くことである．第三にリアルタイム・プログラムは入出力装置の世話をしてはいけないということである．最近の汎用，科学用，事務用コンピュータではデータチャンネル機構によって計算機本体の流れと入出力の実行とが切り離されているのが普通だが，この目的は本体の稼動率を高めるのが主な目的であるのに対し，プロセス・コンピュータの場合には目的はもっと深く，単に時を失くして費用をむだにするだけでなく，このプロセス・コンピュータが稼動しない時があるということは，制御されるプラントに莫大な損害，あるいは故障発生という不測の事態を起させることになるからである．プロセス・コンピュータの備えるべき点は，以上のことから効果的なプログラム制御ということであり，演算能力はその次ということになる．

2. プロセス・コンピュータの適用例と分類

プロセス・コンピュータの代表的な用途としては，次のような分類ができる．

オンラインとして：

プロセスデータのしゅう集，制限値を超過した時の警報，運転効率とプロセス特性の計算，自動作表，オペレータへの運転指令，プロセス運転制御，その他．

オフラインとして：

技術解析，最適計算，原料計画，データ計算，その他．

プロセス・コンピュータはその機能から3段階に分けられ，直接プロセスに取りつけられているものをレベル1，それらの各ユニットを制御するレベル2，そしてスーパーバイザとしてスケジューリング，プランニング，プロダクション制御を行なうレベル3とに分れる．このような分類を Hierarchy system と呼ぶ．

プロセス・コンピュータの信頼度を上げるため，各レベルごとに機種を変える方法があって，この場合ソ

---

* The Softuare of Process Computer, by Tuneo Kinoshita (Tokyo Shibaura Electric Co., ltd, Computer Engineering Dept.)

** 東京芝浦電気株式会社電子計算機技術部

フトウェアは単機種で制御する場合より複雑になる.

3. モ ニ タ

先にも述べたとおり，プロセス・コンピュータの中心機能は割り込みの管理ということであった．このことの実行を司るシステムをモニタと呼ぶことにする.

東芝ではプロセス・コンピュータとしてこのほど開発した TOSBAC-7000 シリーズがあるが，この TOSBAC-7000 は徹底してモニタシステムを採用している．ここではそのモニタの紹介を兼ね，プロセス・コンピュータのプログラム管理の一方法を述べてみたい.

モニタは大別して，管理ルーチン，IOCS，UTILITY の三つから成っている．管理ルーチンはいくつか用意されたプラント制御プログラムの，優先順位，時間割をアレンジし，各プログラムからの割り込みに従って動作の開始，停止，廃止，遅延などを行なわせ，時計の管理をし，各種テーブルの操作，内部データの転送を行ないメモリプロテクトの管理を行なう.

IOCS はいうまでもなく入出力の管理を一手に引き受けるものであるが，特に周辺機器の故障対策とか，アナログ・ディジタル入出力機器の待ち合わせやデータのオフセットも行なう.

UTILITY ルーチンはオフラインでも動作するが，特にオンラインでプロセスをコントロールしながら，デバックすることが許されるよう設計されている.

これらのうちで特に重要なのは管理ルーチンで，次のものから成っている.

（1） 中央管理ルーチン
（2） 時計
（3） 時計処理ルーチン
（4） 遅延発生ルーチン
（5） プログラム始動ルーチン
（6） プログラム停止ルーチン
（7） 情報保存，復元ルーチン
（8） 内部データ転送ルーチン
（9） メモリプロテクトルーチン

中央管理ルーチンはプログラム群のスケジューラで，各プログラムの優先順位と実行開始時間をにらみ合わせ，時間割を作ったり更新したりする．プログラムはそれぞれに実行中，停止，遅延，廃止の各状態を示すテーブルが与えられ，実行中のものはその時の時刻が入っている.

時計ルーチンは 1/60 秒ごとの割り込み発生に基づいた時刻機能で，この時間に基づいてモニタは動かされて行く.

モニタを構造上の立場から見ると，まずモニタそのものは固定したものではなく，プラント制御プログラムの組み合わせに従ってモニタも構成が変わってくるように作られている.

モニタの各モジュールは全てアセンブラ言語で表わされたシンボリック表現のままで保存管理してあり，この中から必要なルーチンを選び出して，プラント制御プログラムと共にアセンブルしてはじめて一つのモニタ組織ができあがる.

プロセス用コンピュータの場合には，あるプラントが決まればその制御に要するプログラムは固定してしまうものであるから，このようなモニタの構成がたびたび変わるということはなく，したがってある程度手間をかけてアセンブルを行なっても，それほど経費を要したことにはならない.

また次のようなことも考えられる．日本ではコンピュータというと万能という思想が強いが，むしろプロセス・コンピュータに関しては，ある万能コンピュータに全てを期待するより，仕事のレベルに合わせた能力別コンピュータをいくつか用いる方が安全性と稼動率を増すと考えられ，われわれのモニタの場合にもモニタを構成し，アセンブルするコンピュータとそのアセンブルされたオブジェクト・プログラムをランさせるコンピュータとを区別しておくということが考えられている.

科学用コンピュータには FORTRAN, ALGOL があり，事務用コンピュータには COBOL があるようにプロセス・コンピュータにも何か世界共通言語がほしいところだが，先に述べた適用例をみてもわかるとおり，あまりにも多種多様の用途と，そのたびに異なる入出力装置と機構のために，一つの言語を用意して全てをカバーすることはなかなか困難だし，かえって能率の悪いオブジェクト・プログラムができてしまう心配がある．プログラムすることに多少時間はかかっても実際のランタイムで最も有効なプログラムが働いてくれないとプラント制御機能として失格になる.

TOSBAC-7000 ではコンパイラとして FORTRAN II，IVが用意されていて，それぞれモニタの管理のもとにオンラインで動作するようになっている．しかしもともと FORTRAN はプロセス入出力の機能を持っていないから，その部分はアセンブラに頼らなければならない．ここで問題となるのはコンパイラとアセン

ブラとのリンクの部分だが，われわれの場合ははじめからこのことを念頭において，同一プログラム内にコンパイラ言語でもアセンブラ言語でも書けるように設計した．両者の区別はコーディング上の特別な欄に番号の区別をおくだけでよく，リンクのためのパラメータの処理やスタックの処理は一切不要にしてある．

今後も各方面でプロセス・コンピュータ用言語が考え出されるだろうが，今のところこうした言語を考えるより，より優れたモニタの開発の方に力を注ぐ方が得策であると思われる．

参考文献

1) TOSBAC-7000 MONITOR 説明書
2) 田中・鈴木：プロセス用計算機のコンパイラについて，情報処理 Vol. 5, No. 5, 1964, pp. 253.
3) Guide-line and General Information on User's Requirements Concernning Direct Digital Control, 1963. 4, 米プリンストン ISA 化学石油分科会

（昭和41年3月7日受付）