

## リアルタイム性を強化したGKSの開発

## 1K-3

○菊池康之・古家俊幸・松本好司(三菱電機東部コンピュータシステム株式会社)  
渡部明洋(三菱電機株式会社)

## 1. はじめに

グラフィックの標準化は近年活発に実施され、その中でもGKS(Graphical Kernel System)は、初めて国際規格(ISO)となったものである。

GKSは機能が豊富で汎用性に富み、これに準じ作成したアプリケーションプログラムは可搬性が高い。

そこで当社では、GKS・2Cレベル+3Dを実現しているMELGKSにいくつかの機能を追加し、また高速、小容量化等の工夫をし、リアルタイムシステムに合ったGKS(ICSGKS)を開発した。

## 2. リアルタイムシステムに必要な機能

GKSをリアルタイムシステムに利用するとき、まず問題となることは応答性である。リアルタイムシステムでは、ある事象が発生してから、なるべく早くオペレータに知らせなければならない。又、その事象の数が多い場合等は複数のタスクからの要求を同時に処理しなければならない。この場合、通常のGKSライブラリでは1台のグラフィックディスプレイに1つのユーザプログラム(タスク)が表示制御を行なうため、複数タスクからの同時動作要求には非常に制限が多い。又、容量の点でも多くのユーザが同時に動作することは、メモリ不足を招く可能性が高い。そして、オペレータに対しては状態表示中に於いても表示を中断したキー入力が可能で、オペレータの要求に瞬時に応える必要がある。

これらをまとめ、グラフィックシステムにおけるリアルタイム性とは次のものがあげられる。

- ① ソフトウェア・オーバーヘッドの軽減
- ② 複数タスク同時処理が可能
- ③ 小容量
- ④ 非同期入力が可能

## 3. 実現方式

## 3.1 ソフトウェア・オーバーヘッドの軽減

リアルタイムシステムにおける表示処理の多くは、固定的な画面の上に時々刻々と変化する要因を追加・変更するものである。

例えば、地図を表示した上に等圧線や雨量情報を加えたり、プラント制御図の上に製品の流れを画く等である。

これらの場合、固定的な画面は変化することがほとんどなく、表示に関してはその都度ユーザプログラムからベクトル情報等を送出する必要がない。通常のGKSでは、メタファイル等に固定画面を作成しておき、それを必要に応じ出力する方法がある。しかし、メタファイルは中間的な情報であり、グラフィックディスプレイの命令に展開する処理が必要となり時間がかかる。

ICSGKSではここに着目し、グラフィックディスプレイの命令に展開した形式でのファイル生成を実現し高速化を図っている。(図1)

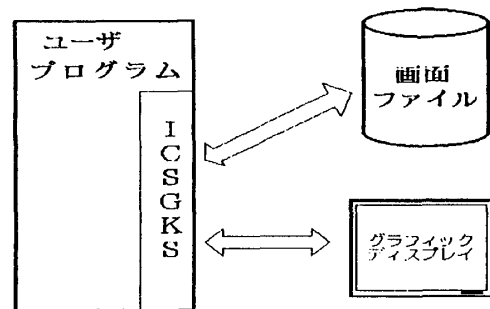
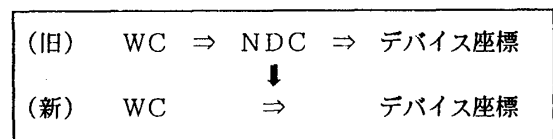


図1 画面ファイル

GKS規格における論理的もしくは数学的厳密さをもって定義された処理過程を、実用システムに適合するように最適化を行なった。即ち、デバッグ終了後、実運転に入った時には毎回チェックする必要のない箇所、及び座標変換方式の処理パスを簡略化した。



座標変換過程

又、従来のGKSでは拡張機能(エスケープ)利用時にデータのバック等が必要であるが、データのバックによる処理時間を短縮するために、データのバック・アンバックを必要としない拡張機能の利用手段を提供している。

これらの方式で高速化を実施し、従来製品の約3倍(当社比)の性能向上が期待される。

Industrial Control System's GKS

Y. Kikuchi<sup>1)</sup> T. Furuya<sup>1)</sup> Y. Matsumoto<sup>1)</sup>  
A. Watabe<sup>2)</sup>

1) MITSUBISHI ELECTRIC COMPUTER SYSTEMS(TOKYO) CORPORATION

2) MITSUBISHI ELECTRIC CORPORATION

### 3. 2 複数タスク同時処理

1 台のグラフィックディスプレイに対して、複数タスクによる図形の同時入出力を可能にしている。(図2) リアルタイムシステムでは、各種センサーから入力される情報の変化を、即時にグラフィックディスプレイに表示しなくてはならない。表示画面は1つでも、部分要素を変更する箇所は数多く、従来のGKSライブラリ構成では多数の情報の変化を同時に処理することはできない。

そこでICSGKSでは、表示済みのディスプレイに対し画素を追加できるモードを加え、追加表示を可能とすると共に複数タスクからの同時アクセスができる環境を提供している。

これにより、1画面を構成する機能分割をタスク単位で行うことが可能となり、システム設計への展開が柔軟となった。

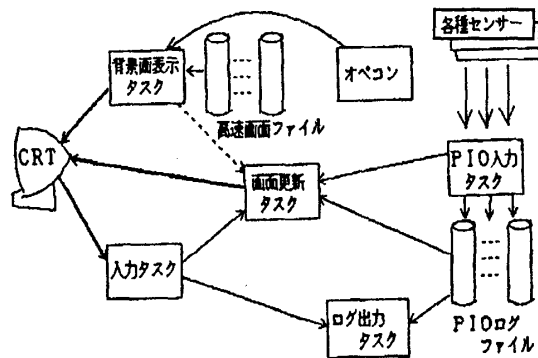


図2 複数タスク同時処理例

### 3. 3 小容量化

通常のGKSは、ライブラリとしてアプリケーションプログラムに抱え込まれる。複数のタスクを同時に動作させ、グラフィックディスプレイへの描画を行なうシステムでは同じGKSライブラリを抱え持ったプログラムが多く立上り、メモリ不足を招く可能性がある。

この時、仮想記憶システムを導入したところでシステム全体の応答性能は下降傾向(スワッピング処理等によるディスクI/Oが増加)となる。

そこでICSGKSは、これらのシステム性能を考慮した結果、GKSライブラリの主要部分を主記憶上に常駐させることで応答性能を確保し、又、その常駐ロジックの複数タスクによる共用化を可能とした。(図3)

これらの方法で、システムの応答性を落とす事無く、プログラム1本に対して組み込まれるGKSライブラリ部のメモリ容量を従来サイズの15%にまで縮小している。

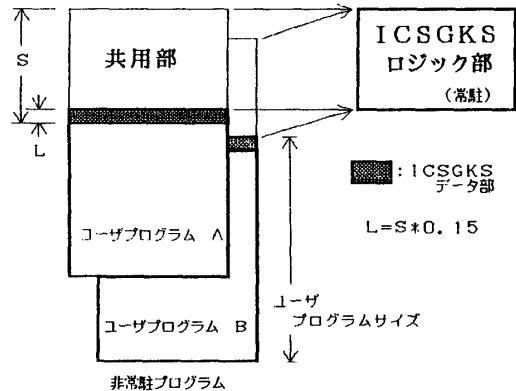


図3 ICSGKS構成

### 3. 4 非同期入力

一般のGKSソフトウェアでは、入力要求が発生するとディスプレイに対する入出力動作を占有してしまい、以降のデータ入出力はその入力動作が終了するまで待たされる。特に要求/事象入力は、オペレータからの入力があるまで待たなくてはならない。リアルタイムシステムの処理では、入力中であっても時々刻々と変化する情報を表示することが必須である。

そこで、オペレータからの入力タイミングを割り込みとして発生させる機能をグラフィックディスプレイに盛込んだ。ICSGKSでは、予め入力割り込み処理の手続きを登録し(要求/事象待ち使用時に自動的に割り込み登録される)、割り込みがグラフィックディスプレイから通知されるまで、他タスクからの図形の出力動作を制限なく行なえることを可能とした。そして、割り込み発生時には以前に登録されていた入力動作を瞬時に実施することとした。

尚、入力を要求したプログラムは割り込みが通知されるまでWAIT状態にスケジューリングされ、他の処理プログラムへのCPU占有を増大させている。

### 4. まとめ

ICSGKSは、一般的なGKS使用ユーザに対しても十分な機能提供ができるものであり、これらの機能強化を行なった事項に対して何ら意識する事無く使用することができる。又、従来のGKSソフトウェアからの移行が容易である。

尚、これらの機能強化項目は、リアルタイム機能としてまとめているが、グラフィックの一般的な利用分野においても要求される項目であり、今後もより高速処理の可能なライブラリを考えてゆく必要がある。