3R−6

マルチスレッド対話型COMETシミュレータ

山本洋彦*　宮武明義**　今宮淳美*
*山梨大学電子情報工学科　**詫間電波工専情報工学科

1．はじめに

ユーザインタフェースとしてマルチスレッド対話を導入しようとする場合，ユーザインタフェース設計に長時間が費やされてきた．そこで，マルチスレッド対話を実現する共通のモジュール構造に従う設計を行うことにより，設計，実現段階および機能拡張時においても，ソフトウェア開発者の負担を軽減することが期待できる．また，マルチスレッド対話に適応するＵＩＭＳ[1]（ユーザインタフェース管理システム）の実現を容易にすることも期待できる．

本稿では，COMET シミュレータのマルチスレッド対話型インタフェース作成を述べて，そのようなモジュール構造の実現例を示す．

2．マルチスレッド対話

Mach や OS/2 におけるスレッドとは，UNIX でのプロセスの概念のうちの実行に関する部分を指すが，本稿では，ユーザの視点に立ったときの，共有データにアクセス可能かつ並行実行可能な操作単位をスレッドと呼ぶ．一つのソフトウェア中に複数のスレッドを設けることによりマルチスレッド対話[2]が実現し，ユーザは複数のスレッドを任意の順序で操作しながら全体として一つの仕事を達成することができる．

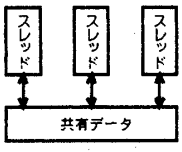

図1　マルチスレッド

3．COMET シミュレータ

COMET は通産省が実施する情報処理技術者試験で用いられている仮想計算機である．ここでは，作成したマルチスレッド対話型 COMET シミュレータを紹介し，その実現方法を示す．

3.1　仕様

COMETシミュレータは，次の４つのスレッドで構成される．

①仮想端末スレッド：プログラムのロード，実行
②メモリ・スレッド：メモリ内容の表示，変更
③レジスタ・スレッド：レジスタ値の表示，変更
④デバッグ・スレッド：トレース，ブレークポイントの設定，一行アセンブル，逆アセンブル

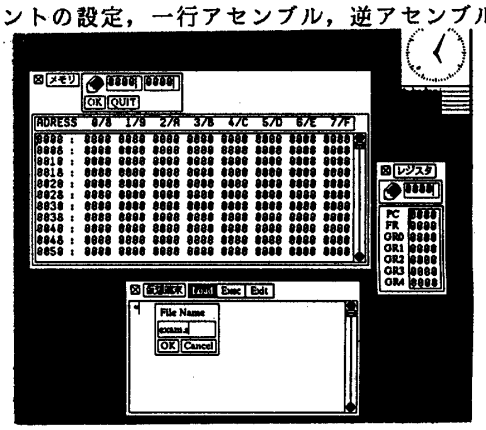

図2　COMETシミュレータ

COMET のメモリとレジスタが共有データとなる．スレッド間のユーザの操作順序は任意であり，各スレッドは並行実行可能である．但し，同時に同じデータ領域に書き込もうとする場合は制限される．また，あるスレッドにおけるデータの変更は，直ちに関連する別のスレッドに影響を及ぼす（例えば，仮想端末スレッドでプログラムをロードすると，メモリ・スレッドのメモリ内容表示部に影響を及ぼす）．

3.2　シミュレータの構成

各スレッドはユーザインタフェース部とアプリケーション部から成る．また，各スレッドは共有データ（メモリ，レジスタ）へ随時アクセスしながらコマンド実行に関する処理を行うので，排他制御のた

The Multi-Threaded Dialog for COMET Simulator
Hirohiko Yamamoto[1], Akiyoshi Miyatake[2], and Atsumi Imamiya[1]
[1]Yamanashi University　[2]Takuma National College of Technology

めの共有データ管理部（ＳＤＭ）が必要である．

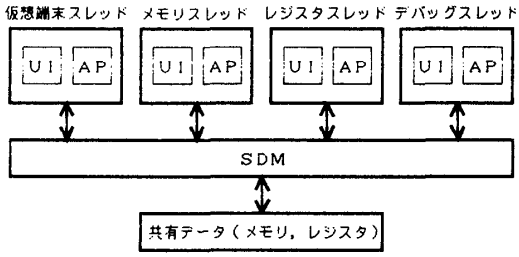

図３　シミュレータの構成

3.3 実現方法

各スレッドを，UNIX 4.3 BSD上でマルチプロセスで実現する．各スレッドは，UIプロセスとAPプロセスで実現し，共有データ管理部はSDMプロセスとして実現する．各プロセスは図４に示すモジュール構造によって構成される[3]．

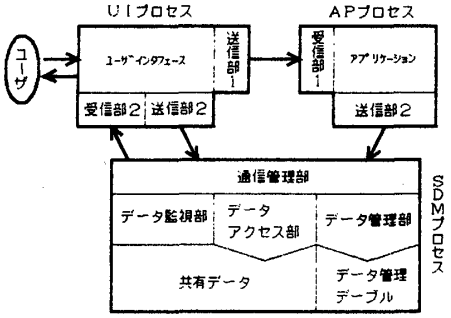

図４　モジュール構造

UI,APプロセスは，各々の処理部分と通信部分から成る．SDMプロセスは，通信管理部，共有データの更新を監視するデータ監視部，共有データへのアクセス（メモリ，レジスタへのリード／ライト）を行うデータアクセス部，排他制御を行うデータ管理部及びそのためのテーブル，そして共有データから成る．

3.4 プロセス間通信

次の形式のメッセージの送受信により，各プロセス間の処理の要求をする．

（命令　引数１　引数２　…）

●UIプロセス→APプロセス のメッセージの例

```
仮想端末スレッド　(LOAD filename)
                  (SAVE filename addr1 addr2)
                  (EXEC filename)
メモリ・スレッド　(M/D addr1 addr2)
                  (M/E addr value)
レジスタ・スレッド(R/D)
                  (R/E id value)
```

●UI,APプロセス→SDMプロセス のメッセージの例

```
各スレッド共通　(RMEM addr)          ;ﾒﾓﾘの読みだし
                (WMEM addr value);ﾒﾓﾘの書き込み
                (RREG id)            ;ﾚｼﾞｽﾀ読みだし
                (WREG id value)      ;ﾚｼﾞｽﾀ書き込み
```

3.4 処理の流れ

あるスレッドで，UIプロセスがユーザからの実行要求を受けると，そのスレッド独自のメッセージをAPプロセスへ送り，APプロセスはそれに従いSDMプロセスへデータ・アクセス要求を行いながら処理を進める．データの内容が更新された場合は関連するスレッドのUIへ通知する．

```
例：UI→AP  (LOAD test)
                ;仮想端末スレッドのLOADを実行
    AP→SDM (WMEM $0000 $9000);メモリへの書込み
           (WMEM $0001 $001A)
           .......
    SDM→UI ($0000 $9000) ;メモリ内容の更新通知
           ($0001 $001A)
           .......
```

４．おわりに

図４のモジュール構造に従ってマルチスレッド対話型 COMET シミュレータを作成した．今後，ソフトウェア依存度の小さいモジュール（送・受信部，ＳＤＭ，データ管理関数・テーブル）をソフトウェア固有のパラメータを与えることにより半自動的に作成するツールを作成する予定である．

参考文献

(1) 今宮淳美：ﾕｰｻﾞｰｲﾝﾀﾌｪｰｽ管理ｼｽﾃﾑ，情報処理ﾊﾝﾄﾞﾌﾞｯｸ 第13編，第10章，ｵｰﾑ社，pp.1194-1201，(1989).

(2) Tanner,P.P.: Multi-Thread Input. Computer Graphics Vol.21 No.2, pp.142-145,(Apr.1987).

(3) 宮武明義，今宮淳美：ﾏﾙﾁﾌﾟﾛｾｽによるﾏﾙﾁｽﾚｯﾄﾞ対話型ﾕｰｻﾞｰｲﾝﾀﾌｪｰｽの設計，ｸﾞﾗﾌｨｸｽとCAD研究会 43-2，pp.1-8，(1990).