ソフトウエア工学 17－9
（1981. 2. 12）

マイクロコンピュータのパーソナルコンピュータ化への一過程

土居範久（慶應義塾大学情報科学研究所）
越秋雄，松井弘樹（慶應義塾大学工学部）

現在，我々の研究所における個人用計算環境（personal computing environment）の整備を目的とした研究を進めているが，これは68000系あるいはそれと類似のシステムを用いて開発し，最終的にはLSIあるいはVLSIを用いた専用の環境を開発することを目指している。我々のような環境では個人用計算機システムとしては，マイクロコンピュータ（＋局所的ファイル＋表示システム＋通信機能），基本言語およびエディタが必須であって，さらに，清書系，テスト系，動的解析系，静的解析系，デバッグ系，増殖的に開発するための諸機能，プログラム向きファイル系，版の管理，最適化コンパイラといった各種の道具および各種のパッケージが必要であろう。

ところで，現在，我々の研究施設としては，Dec-2020，Cromenco があり，さらに計算センターのFacom M-180 II ADの表示装置端末が1台ある。M-180は混んでいるときにはログインさせてくれず，運よくログインさせてくれたときには応答時間が5分位かかり，処理系を呼出すと3分位かかり，エディタを使っているときには行単位で30秒位かかる。不幸なことに，通常は混んでいる。Dec-20には端末を6台しか接続していないので，通常は不都合を感じないが，Simula-67とPascalを走らせ，裏でゲームの解析などをしていると，エディタを使っていても，さすがに違和感を感じる。もっとも，エディタを単独で使っていてもページにまたがる操作をするとがタッとリズムが崩れる。Cromencoのスクリーン・エディタはよくできていて，Dec-20の標準のエディタより使い勝手がよいし，Runoffもよくできているが，Dec-20との共通媒体がない。さらに，CromencoにはEP刷装置としてバドミントン・プリンタしかないので不便を感じることが多々ある。また，Cromencoは漢字処理システム，Simpl 処理系の開発など使用者が多い。

そこで，

(1) 操作の程度に見合った応答時間を保証する
(2) 資源の共用をはかる
(3) 主計算機（Dec-20）の負荷を軽減する
(4) Cromencoでの作業環境を改善する

そして，"敵は本能寺"にあるが，

(5) 個人用計算環境で必要とするソフトウェアの開発技術を，既存のソフトウェアを利用して蓄積する

ことを目的として，マイクロ・コンピュータCromencoを，CDOSと互換性を保つようにして，通信機能をもった多重プロセス・システムに組み替えることにした。もちろん，(1) および (2) は個人用計算環境で必須の条件である。

このための段取りは次の通りで，典型的なボトムアップである。

[I] まずは，Cromencoの端末をDec-20の端末として使用できるようにする。

[II] CDOSの下でCromencoとDec-20間のデータの転送ができるようにする。
[III] CDOSの下で非常駐プログラム（エディタ，使用者プログラムなど）と常駐プログラム（データ転送手続き，CDOSユティリティ）が動く環境を実現する。
[IV] CDOSを多重プロセスが支援できるように書き換えると同時に，Dec-20からの割込みを処理できるようにする。
[V] バンク切換機構を付加し，多重プロセス環境を実現する。
[VIa] streamの概念[1][2]を導入し，それに応じたファイル系を実験的に試作する。
[VIb] ラスタ・スキャン方式の表示装置を接続し，多重画面システムを試作する。
[VII] ・・・

そして，ここでは，少くとも[V]の段階までの過程を述べる予定であったのだが，見事に目論見はずれ，[III]の段階をデバッグ中で，[IV]の段階を作成しているというのが現状である。たかがマイコンと高を括っていたのだが，
(イ) Z-80の基本的知識（特に割込み機構）に欠けていたこと
(ロ) 原始リストのないCDOSを逆アセンブラを使って暴くことに手間どったこと
(ハ) デバッグ機能を組込みながらプログラミングしなければならなかったことなどが，この目論見のはずれた主な原因であろう。何もないシステムとの互換性を保つのがいかにいやらしいか改めて思い知らされた次第である。

でき上ったものについての[III]の段階までの話は簡単で，「Cromenco と Dec-20 をつなげました」のひと言につきるが，それでは身もふたもないので，簡単に過程をたどってみることにしよう。

[I]の段階は次の五つの手続きによって実現している。†

```
Terminal :  disable interrupts ;
            terminate := false ;  set baud rate ;
            define interrupt vector ;
            enable CRDA and DRDA ;
            while not terminate do od ;
            reset all interrupts

consoleinput :  disable interrupts ;
                Acc := consoleinputport ;
                if Acc = "^K" then
                   terminate := true fi ;
                enable DTBE ;
                return

consoleoutput :  disable interrupts ;
                 consoleoutputport := Acc ;
                 enable CRDA and DRDA ;
                 return
```

† CRDA : Console Receiver Data Available , CTBE : Console Transfer Buffer Empty
DRDA : Dec Receiver Data Available , DTBE : Dec Transfer Buffer Empty

```
decinput :  disable interrupts ;
            Acc := decinputport ;
            enable CTBE ;
            return

decoutput : disable interrupts ;
            decoutputport := Acc ;
            enable CRDA and DRDA ;
            return
```

ここで，"^K" はCDOSへ制御を移すための制御文字である。Accを際どい資源として，割込み可能命令で同期をとっているが，[IV]の段階で環状バッファを設けるように変更した。この段階で，Dec-20の端末が1台増えたことになる。

次に，CromencoおよびDec-20の双方に，データ転送のための手続きを実現することによって[II]の段階を達成している。これらの手続きの構造は基本的には同じであり，次のようになっている。

```
sender :  set mode ;
          repeat
            prepare data packet ;
            wait (^K) ;
            send data packet
          until last packet

receiver : set mode ;
           repeat
             signal (^K) ;
             receive data packet ;
             construct file
           until last packet
```

ただし，データ転送は，必ず，Cromencoからデータ転送の方向，mode（1バイトのビット長で通常は7または8）およびファイル名を指示した制御パケットを送ることによって始まる。

端末側で実際に転送を指示するには，次のコマンドを用いる。

Send (C , D)　　Receive (C , D)

ここで，CおよびDはそれぞれCromencoおよびDec-20のファイル名である。この段階では，通信文単位を処理するための手続きを前述した1バイト単位の手続きをもとに作り使用している。これらの手続きは次のとおりである。

対コンソール	inputfromconsole	outputtoconsole
対Dec-20	inputfromdec	outputtodec

これによって，Cromencoのエディタを使ってDec-20のプログラムを作成したり，Cromencoのフロッピー・ディスクの内容をDec-20のプリンタに印刷したりすることができるようになったと同時に，Cromencoのプログラムを Dec-20で作成し，Cromencoが使えるときにDec-20から取り込むことも可能になり，一応，前述の(1)から(4)が達成された。たったこれだけのことでソフトウェアの開発環境が著しく改善されたのである。早速，Cromencoでの個人用計算環境の整備を促進するために，Dec-20にCromencoに対するSimpleのクロス・コンパイラを現在開発中である。

しかし，ファイルの転送は9,600ボーで行なっているが，転送方式があまりにも原始的なので時間がかかるが（それ程でもなく1KBを転送するのに約4秒），ファイル転送中はCromencoを他の用途に使用することができない。これは，ある程度予期していたことではあるが，使用者連中の要求を入れ[III]に進むことにした。つまり，転送時間を犠牲にしてCromencoの利用効率を上げるようにしたのである。そのために，二つのプロセスを時分割で割振るスケジューラを作成した。

エディタをはじめとするCDOSの下で動く非常駐プログラムが一つのプロセスを構成し，割込みにもとづく入出力処理のための手続きがもう一つのプロセスを構成するわけである。

現在作成中の[IV]の段階でCDOSの切り分けが完了すると，次は[V]の段階に入るわけであるが，バンク切換機構に加えて記憶保護機構も付加することにした．記憶保護機構の仕様に関しては議論百出で，最終的には松竹梅と出揃ったが，現在のところ「梅の上」を採用することにしている。これは，入出力空間に，上限ページ・レジスタ，下限ページ・レジスタ，バンク番号レジスタ，記憶保護制御レジスタおよび状態を記憶するためのフリップ・フロップを設けるだけのもので，記憶保護制御レジスタは核状態から使用者状態への切換えを保留するCPUサイクル数を指定するためのものである。
(5～6年前にタイムシェアリングシステムを開発したときにしたようなことの繰返しである。大きく違うことは，今度は自分達でいじれるということである。)
これに，イジョウ，多重プロセスを実現するための標準的な核をモニタの概念[3][4]にもとづいて作成する計画である。

計画は遅れに遅れているが，できるだけはやく[VI]の段階までの目処をつける予定でいる。

参考文献

[1] Stoy, J.E. and C. Strachey, "OS6 — An Experimental Operating System for a Small Computer," Computer Journal, 15, 2 and 3 (1972).

[2] Lampson, B.W. and R.F. Sproull, "An Open Operating System for a Single-User machine," Operating Systems Review, 13, 5, 98—105 (1979).

[3] Hoare, C.A.R., "Monitors: An Operating System Structuring Concept," CACM, 17, 10, 549—557 (1974).

[4] Lampson, B.W. and D.D. Redell, "Experience with Processes and Monitors in Mesa," CACM, 23, 2, 105—117 (1980).

[5] Thacker, C.P., E.M. McCreight, B.W. Lampson, R.F. Sproull and D.R. Boggs, "Alto: A Personal Computer," Xerox Palo Alto Research Center (1979).

[6] Deutsch, L.P. and E.A. Taft, "Requirments for an Experimental Programming Environment," Xerox Palo Alto Research Center (1980).