

IC-2

Concurrent Pascal 言語の入出力アダプターについて

佐藤文一 田中英彦 元岡達
(東京大学工学部)

1. はじめに

複数プロセスの並列処理機能をもつ Concurrent Pascal システムが、ポリプロセッサシステム PPS-1 上に、PDP11/45 の Emulator (一部の Virtual Code は直接ファームウェア化されている) を作成することにより、1つのプロセスとして稼働されている。また、PPS-1 は、室内計算機網 TECNET の構成員である。今回、Concurrent Pascal システムに入出力アダプターを設け、Kernel などの部分を拡張し、他のシステムと通信できる機能をもたせた。

2. 入出力アダプターの内容

システムの構成は、図1である。

PPS-1・OSでは、プロセス間通信プリミティブ(ハードウェアの拡張とみよすことのできる手続き)として、
COM: 通信要求の提出
COMWAT: 通信終了の問い合わせ
COMCAN: 通信要求の取り消し
が用意されているので、新設した入出力アダプターのための道具としては、上記のプリミティブを利用し、Concurrent Pascal の IO 命令を拡張することにより、イニシアメントした。(図2)

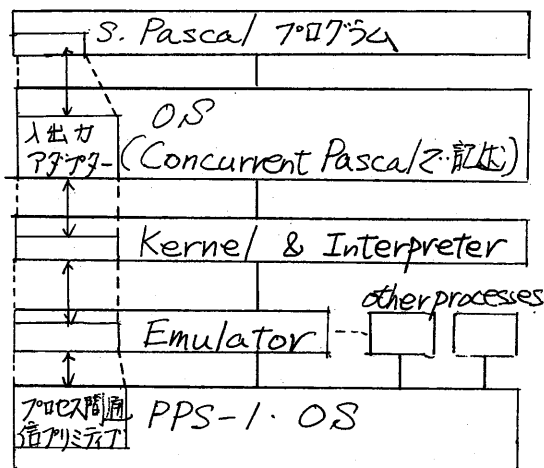


図1 システムの構成

```

*****
* IO TYPES *
*****
TYPE IODEVICE=(TYPEDEVICE, DISKDEVICE, TAPEDEVICE, PRINTDEVICE,
CARDDEVICE, PTRDEVICE, TW2DEVICE, TW3DEVICE, PPSJP0,
PPSJP1, PPSJP2, EXTANYJP);
TYPE IOOPERATION=(INPUT, OUTPUT, MOVE, CONTROL, COMSEND, COMRECEIVE,
COMWAT, COMCANSEND, COMCANRECEIVE);
TYPE IOARG=(WRITEOF, REWIND, UPSPACE, BACKSPACE);
TYPE IORESULT=(COMPLETE, INTERVENTION, TRANSMISSION, FAILURE,
ENDFILE, ENDMEDIUM, STARTMEDIUM);
TYPE IOPARAM=RECORD
OPERATION: IOOPERATION;
STATUS: IORESULT;
ARG: IOARG;
BYTELENGTH: INTEGER
END;
    
```

図2 IO命令の拡張

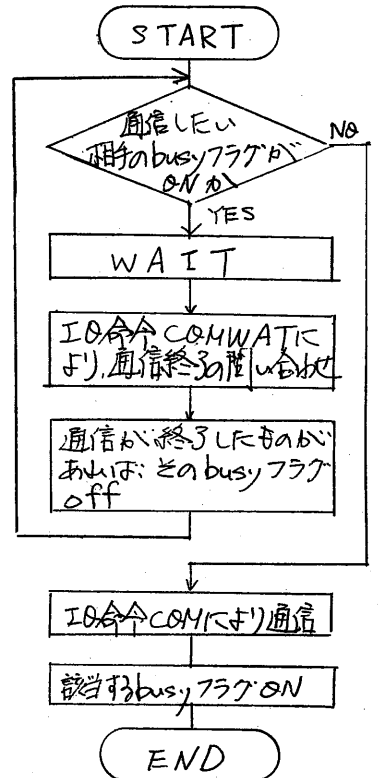
他のシステムと通信を行おう場合、通信完了まで処理が止まるブロック指定モードで通信を行おうと、Concurrent Pascal システム OS は、その通信が終了するまで、完全にストップしてしまいます。これでは、能率が悪く、複数のユーザを管理する場合には具合の悪い点となる。従って、非ブロック指定モードによって通信を行おうことにした。非ブロック指定モードで通信を行おう場合、以前出

した通信要求が終了したかどうかの確認する必要がある。このために、他のシステムごと busy フラグを設け、通信が終了したという時に、そのシステムに対し、次の通信要求を出すことが出来るようにした。この管理は、Concurrent Pascal のモニタによって実現した。すなわち、busy フラグに対する処理をモニタ中の monitor procedure で記述し、Concurrent Pascal 内の各プロセスは、このモニタを介してのみ通信終了かどうかの確認できるようにした。つまり、モニタ自身の排他性を利用し、busy フラグをプロセス間の共有のデータとみわけて管理するのである。この busy フラグに対する操作を示したのが図3である。相手システムと通信する場合の基本的な制御の流れを示したのが図4である。

```

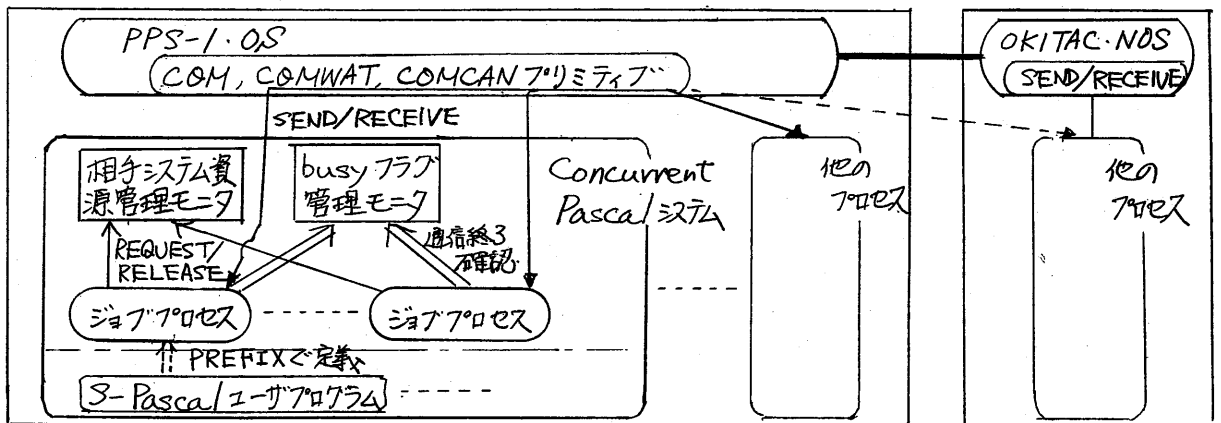
*****
* BUSYFLAGMONITOR *
*****
TYPE BUSYFLAGMONITOR=MONITOR
VAR BUSY:ARRAY(,PPSJPO..EXTANYJP,) OF BOOLEAN;
    J:IODEVICE;
FUNCTION ENTRY CHECKBUSY(I:IODEVICE):BOOLEAN;
BEGIN CHECKBUSY:=BUSY(I,) END;
PROCEDURE ENTRY BUSYON(I:IODEVICE);
BEGIN BUSY(I,:)=TRUE END;
PROCEDURE ENTRY BUSYOFF(I:IODEVICE);
BEGIN BUSY(I,:)=FALSE END;
BEGIN
FOR J:=PPSJPO TO EXTANYJP DO BUSY(J,:)=FALSE
END;
    
```

図3 busy フラグ管理のためのモニタ



この他に、複数のジョブプロセスが相手プロセスと通信する際の排他性を管理するために、相手システム資源管理モニタを設けた。この全体の関係を、以下に示す。

図4 通信の制御の手順



3. おわりに

入出力アダプターのためのプログラムサイズは、マイクログラム138ワード、Concurrent Pascal プログラムで、busy フラグ管理モニタほどを含めて110行である。この程度の大きさですんだのは、Concurrent Pascal によって、かなり近い部分の記述ができたこと、抽象データタイプの利用のためである。