

4U-5

マイクロコンピュータにおけるシリアルラインを用いた

TCP/IPの実現

神田薫, 矢吹道郎, Robert Deiters

上智大学

1. はじめに

近年、ローカル・エリア・ネットワークの構築が盛んである。このような状況の中で、現在広く利用されているマイクロ・コンピュータをネットワークの構成要素として利用したいという要求が生じてくる。そこで問題となるのは、通進路即ちハードウェア及びプロトコルの選択である。プロトコルとしては現在最も広く利用されているのはTCP/IP及びXNSであろう。マイクロ・コンピュータ用の高速のインターフェースとしてはイーサネットが普及しつつあるが、それらはいまだ数も少なく高価である。しかしマイクロ・コンピュータはシリアルラインをほぼ標準で備えている。そこで、我々は、プロトコルとしてTCP/IPを、ハードウェアとしてシリアルラインをも利用できる形で、ネットワークの機能をマイクロ・コンピュータにおいて実現したので、その設計、実装、並びに問題点について報告する。

実装において、ネットワークとしては、バークレー版UNIX 4.2BSDのネットワークを、マイクロ・コンピュータのオペレーティング・システムとしてはMS-DOSを採用した。

2. 設計

このシステムの設計においては、将来の拡張性、並びにUNIX 4.2BSDとの相性を考慮し、UNIX 4.2BSDと同じ手法をとった。即ち、以下の設計方針を持って設計を行った。

- ◆デバイスの多重化を可能とする。
- ◆プロトコルの多重化を可能とする。
- ◆IPの機能として、ルーティング、データ分割を実現する。
- ◆TCPの機能として、データ転送、信頼性、フローコントロール、回線化、回線多重化を、実現する。
- ◆ユーザインターフェースをUNIX 4.2BSDと同様に設計し、基本的にUNIX 4.2BSDのソース・プログラムが大幅な変更なしに実行されるようにする。

3. 実装

マイクロ・コンピュータでは、たとえネットワークに接続されたとしても、本来のスタンドアロンの利用も重要である。従って、ネットワークを利用しない場合、ネットワークの機能が本来の利用に影響を与えるよう

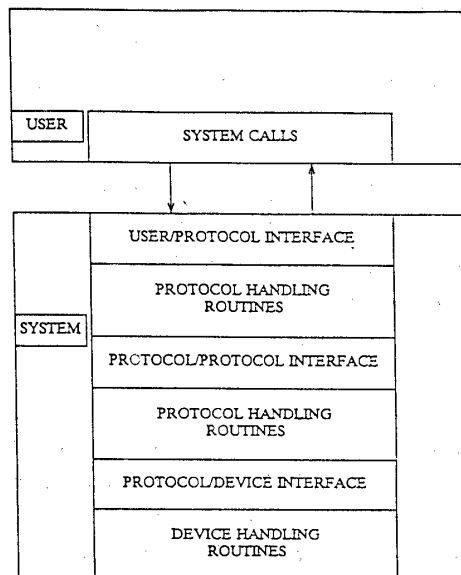


Fig. 1. Structure

あってはならない。

本システムにおいて、TCP/IPのハンドリングはすべてMS-DOSの常駐プロセスにおいて処理される。従って、本来のMS-DOSの変更は必要なく、用意されたプログラムを常駐させることにより、ネットワークの機能が利用できるようになる。

本システムの構造を[Figure 1]に示す。常駐プロセスは、基本的に三つのルーチン、(1)システムコール、(2)プロトコル・ハンドラー、(3)デバイス・ハンドラーからなっている。それぞれのルーチンについて以下に示す。

(1) システムコール

システムコールはsocket, bind, connect, accept, send, recv等、UNIX 4.2BSDと同様のルーチンが用意されている。システムコールはすべて常駐プロセスに対するインタラプトにより実行される。

(2) プロトコル・ハンドラー

プロトコルハンドラーは、各プロトコルに1つずつ用意され、現在はIP [1]、TCP [2]、UDP [3]のプロトコル・ハンドラーが用意されている。

(3) デバイス・ハンドラー

デバイス・ハンドラーは、シリアルラインとイー

サネットのための二つのルーチンが用意されている。

常駐プロセスは三つの要求、即ちユーザ・リクエスト、タイマ・インタラプト、ハードウェア・インタラプトにより駆動される。入力データの流れの様子を [Figure 2] に示す。ハードウェアからの入力、それぞれのインタラプトにより、ハンドラ・ルーチンによって解釈され、ユーザに到達する。

また、システムコールあるいは、その他のユーティリティを利用するためのライブラリを用意し、すべてUNIX 4.2BSDと同じ仕様とした。

4. プログラム例

MS-DOS上で実行されるプログラム例を [Figure 3] に示す。これは、TCPプロトコルを用いたストリームのソケットを開きデータを受信するプログラムで、`sclose`を`close`とする以外は変更なしにUNIX 4.2BSDで実行することが可能である。

5. 問題点

当然のことであるが、TCP/IPはその仕様が小さいとは言えず、マイクロ・コンピュータに移植する場合、メモリスペースとCPUパワーの限界が問題となる。これは、送受信のためのバッファの領域、受信したデータを処理するために処理時間の短縮が要求されるからである。処理時間が長いと無用な再送が行なわれ、システムの負荷が大きくなる。特にIPの分割は危険である。IPにより分割されたパケットは論理的に一つのパケットであるため、連続するパケットの一つでも受信に失敗すれば、すべての分割されたパケットが再送されるからである。

シリアルラインにおいては、その信頼性が大きく問題となる。なぜなら、パケットの再送のメカニズムのため、伝送路の信頼性が低いとパケットの再送信を繰り返すことになり、これもまたシステムへの負荷となる。シリアルラインは安定した伝送速度、及び状態で使用すべきである。また、伝送路の信頼性によって、パケット長も適当に選ぶ必要がある。

6. 結論

マイクロ・コンピュータをネットワークに接続するため、通信路として最も一般的であるシリアルライン並びにイーサネットを、プロトコルとしてTCP/IPを用いた。これにより、シリアルラインではその伝送速度は低い、マイクロ・コンピュータを容易にネットワークの構成要素とすることが可能となり、またそのユーザ・インターフェースをUNIX 4.2BSDと同様に設計したため、ユーティリティの開発が容易となった。

7. 参考文献

- [1] J. Postel, "Internet Protocol," RFC791, SRI International, September 1981.
- [2] J. Postel, "Transmission Control Protocol," RFC793, SRI International, September 1981.

- [3] J. Postel, "User Datagram Protocol," RFC768, SRI International, August 1980.

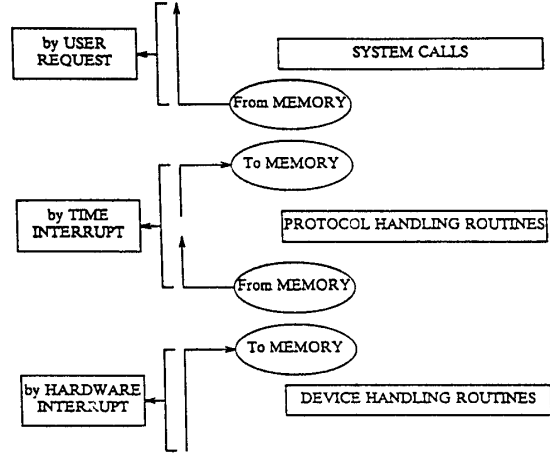


Fig. 2. Motion of input

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <signal.h>
#include <errno.h>
extern int errno;
main(argc, argv)
char **argv;
{
    struct sockaddr_in sin;
    char buf[1024];
    int s, i, ret, new;

    sin.sin_port=htons(atoi(argv[1]));
    sin.sin_addr.s_addr=INADDR_ANY;
    sin.sin_family=AF_INET;
    s=socket(AF_INET,SOCK_STREAM,0);
    bind(s,&sin,sizeof(sin));
    listen(s,3);
    for(;;){
        new=accept(s,0,0);
        while((ret=recv(new,buf,sizeof(buf)))>0){
            printf("%d received\n",ret);
        }
        sclose(new);
    }
    sclose(s);
}
```

Figure 3. Example Program