ネットワーク接続の方法

山本昌志*

2007 年 10 月 17 日

概 要

簡単なチャットプログラムをとおして，コンピューター間をネットワークで通信する方法を学ぶ．

1 本日の学習内容

本日の授業では，通信のコンピューター間で通信を行う場合の手続きについて学習する．教科書 [1] では，p.258–276 である．以下，学習のゴールを示す．

- 通信の手続きのプログラムの書き方が分かる．

2 コンピューターをネットワークを使って通信する方法

2.1 通信を行うプログラムの例

コンピューターを使って，会話を行うチャットプログラムをリスト 1 とリスト 2 に示す．リスト 1 は，端末よりテキストを指定の PC へ送信するプログラムである．一方，リスト 2 は，送られてきたテキストを端末に表示するプログラムとなっている．これら 2 つのプログラムを使って，チャット (会話) ができる．

リスト 1: テキストを送信するクライアントプログラム

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <sys/types.h>
4  #include <sys/socket.h>
5  #include <netinet/in.h>
6  #include <arpa/inet.h>
7
8  int main(int argc, char *argv[])
9  {
10   struct sockaddr_in server;
11   char message[80], ip_address[16];
12   int fd;
13
14   strcpy(ip_address,argv[1]);           // コマンドライン引数のIPアドレスのコピー
15   fd=socket(PF_INET, SOCK_STREAM, 0);   // ソケットの作成
16
```

*独立行政法人　秋田工業高等専門学校　電気情報工学科

```
17    memset((char *) &server, 0, sizeof(server));    // アドレス構造体の初期化
18    server.sin_family=AF_INET;
19    server.sin_port = htons(5320);
20    server.sin_addr.s_addr=inet_addr(ip_address);
21    connect(fd, (struct sockaddr *) &server, sizeof(server));
22
23
24    while(1){
25      printf("message:");
26      fgets(message, 80, stdin);
27      if(strncmp(message,"bye",3)==0)break;
28      send(fd, message, strlen(message),0);
29    }
30
31    close(fd);
32
33    return 0;
34  }
```

リスト 2: テキストを受信するサーバープログラム

```
1   #include <stdio.h>
2   #include <string.h>
3   #include <sys/types.h>
4   #include <sys/socket.h>
5   #include <sys/socket.h>
6   #include <netinet/in.h>
7   #include <arpa/inet.h>
8
9   int main(void)
10  {
11    struct sockaddr_in client, server;
12    char read_str[80];
13    int fds, fda, length;
14    fds=socket(PF_INET, SOCK_STREAM, 0);
15    memset((char *) &server, 0, sizeof(server));
16
17    server.sin_family=AF_INET;
18    server.sin_addr.s_addr = htonl(INADDR_ANY);
19    server.sin_port = htons(5320);
20    bind(fds, (struct sockaddr *) &server, sizeof(server));
21
22    listen(fds, 1);
23
24    length=sizeof(client);
25
26    fda=accept(fds, (struct sockaddr *) &client, &length);
27
28    while(1){
29      int rn;
30      rn = recv(fda, read_str, 80, 0);
31      printf("%d\n",rn);
32      printf("%s",read_str);
33    }
34
35    close(fda);
36    close(fds);
37
38    return 0;
39  }
```

2.2 通信のプログラム概要

教科書 **??**の p.259–260 に書いてある通り，通信のプログラムはファイルの読み書きと似ている．ここでは述べないが，デバイス (キーボードなどの外部機器) との通信もファイルの読み書きに似ている．コンピューターからみて，データを入出力するものは，すべてファイルとして取り扱われ，ほとんど同じ方法で読み書きができる．

通信を行うためには，通信相手のコンピューターとプログラムを特定しなくてはならない．IP アドレスを指定することで，コンピューターを特定する．通信するプログラムはポート番号により決める．コンピューターは多くの通信を同時に行っており，通信をするプログラムまで指定しなくてはならない．

3 システムコール

3.0.1 概要

それでは，先ほどの例のプログラム，リスト 1 とリスト 2 の実際の通信方法を確認する．

通信には，さまざまなシステムコールが使われる．通信で使われるシステムコールを表 1 を示す．

表 1: 通信に使うシステムコール一覧．参考文献 [2] より，引用

ソケットの作製	socket()
ソケットに名前を付ける	bind()
接続の受付準備	listen()
接続要求	connect()
データ転送	read(), write() recv(), send() recvfrom(), sendto()
ソケットのクローズ	shutdown, close
その他	ioctl(), socketpair() setsockopt(), getsockopt() getsockname(), getpeername()

3.1 初期化処理

初期化処理は，ソケットを作製する `socket()` システムコール[1]と `bind()` システムコールから成っている．

[1]OS に仕事を依頼する関数をシステムコールと言う．システムコールの説明を読みたければ，ターミナル上で「man socket」とする．ほかも同様．

socket() システムコール　ソケットと呼ばれる通信ポートを作製する役割がある．サーバーおよびクライアントで通信を行うための口 (ソケット) を作製すると考える．socket() システムコールを使うために必要なヘッダーファイルとそのプロトタイプ宣言を以下に示す．

socket() システムコール

```
#include <sys/socket.h>
#inlcude <sys/types.h>

int socket(int domain, int type, int protocol);
```

戻り値は，ファイルディスクリプターと呼ばれる整数値となる．この整数でソケットを区別する．引数は 3 つある．引数の詳細については教科書 [?] に書いてある[2]ので，ここでは簡単にのべる．

- 引数 domain は，プロトコルファミリーの設定である．普通のインターネット接続であれば,「PF_INET」を設定する．いわゆる IP(internet protocol) である．
- 引数 type は，TCP 接続であれば「SOCK_STREAM」とする．
- 引数 protocol は，0 と指定すればよい．

bind() システムコール　ソケットに名前を付ける役割がある．具体的には，受信ポート番号を指定する．

bind() システムコール

```
#include <sys/socket.h>
#inlcude <sys/types.h>

int bind(int sock, struct sockaddr *addr, socket_t *addrlen);
```

3.2　接続受理準備

listen() システムコール　ソケットが，接続を待つソケットであることを OS に伝える．

listen() システムコール

```
#include <sys/socket.h>

int listen(int sock, int backlog);
```

- 引数 sock は，ソケットのファイルディスクリプター．
- blocklog は，受け付けるコネクションの最大数．

[2]あるいは,「man socket」を見よ

3.3 接続要求

connect() システムコール　クラインと側がサーバーに対して，接続の要求を出す.

connect() システムコール

```
#include <sys/socket.h>
#inlcude <sys/types.h>

int connect(int sock, const struct sockaddr *addr, socket_t addrlen);
```

3.4 接続要求受理

accept() システムコール　ソケットに他のプログラム (プロセス) が接続してくるのを待ち，接続が完了したらファイルディスクリプタを戻り値として返す．受信には，このファイルディスクリプターを使う.

accept() システムコール

```
#include <sys/socket.h>
#inlcude <sys/types.h>

int accept(int sock, struct sockaddr *addr, socket_t *addrlen);
```

3.5 データの送信

send() システムコール

```
#include <sys/types.h>
#include <sys/socket.h>

ssize_t send(int s, const void *buf, size_t len, int flags);
```

3.6 データの受信

recv() システムコール　ソケットからメッセージを受け取る役割がある.

recv() システムコール

```
#include <sys/types.h>
#include <sys/socket.h>

ssize_t recv(int s, void *buf, size_t len, int flags);
```

3.7　終了処理

close() システムコール　ファイルディスクリプターを閉じる．

recv() システムコール

```
#include <unistd.h>

int close(int fd);
```

引数は，ファイルディスクリプター (整数) である．

4　練習問題

[**練習 1**]　プログラムを入力し，実行させよ．

[**練習 2**]　プログラムを実行させながら，システムコールの引数と動作順序を理解せよ．

参考文献

[1] 内田智史監修, (株) システム計画研究所編. C 言語によるプログラミング　応用編　第 2 版. (株) オーム社, 2006.

[2] 金内典充, 今安正和. UNIX ネットワークプログラミング. 株式会社オーム社, 1997.