

ネットワーク接続の手続き

山本昌志*

2007年10月31日

概要

ネットワークを接続するための手続きの内容を示す。

1 本日の学習内容

コンピューターをネットワークに接続する場合、さまざまな手続きが必要である。ひとつひとつその手続きの内容を示す。教科書 [1] では、p.258–276 である。以下、学習のゴールを示す。

- 通信の手続きの内容が分かる。

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

先週と同じ、チャットプログラムをリスト 1 とリスト 2 に示す。本日は、これを改造して、インターネット通信のプログラムの学習を進める。

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
17     memset((char *) &server, 0, sizeof(server)); // アドレス構造体の初期化
```

*独立行政法人 秋田工業高等専門学校 電気情報工学科

```

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("%s",read_str);
32     }
33
34     close(fda);
35     close(fds);
36
37     return 0;
38 }

```

2.2 通信の順序

ここでの通信は、以下の順序で行われている。

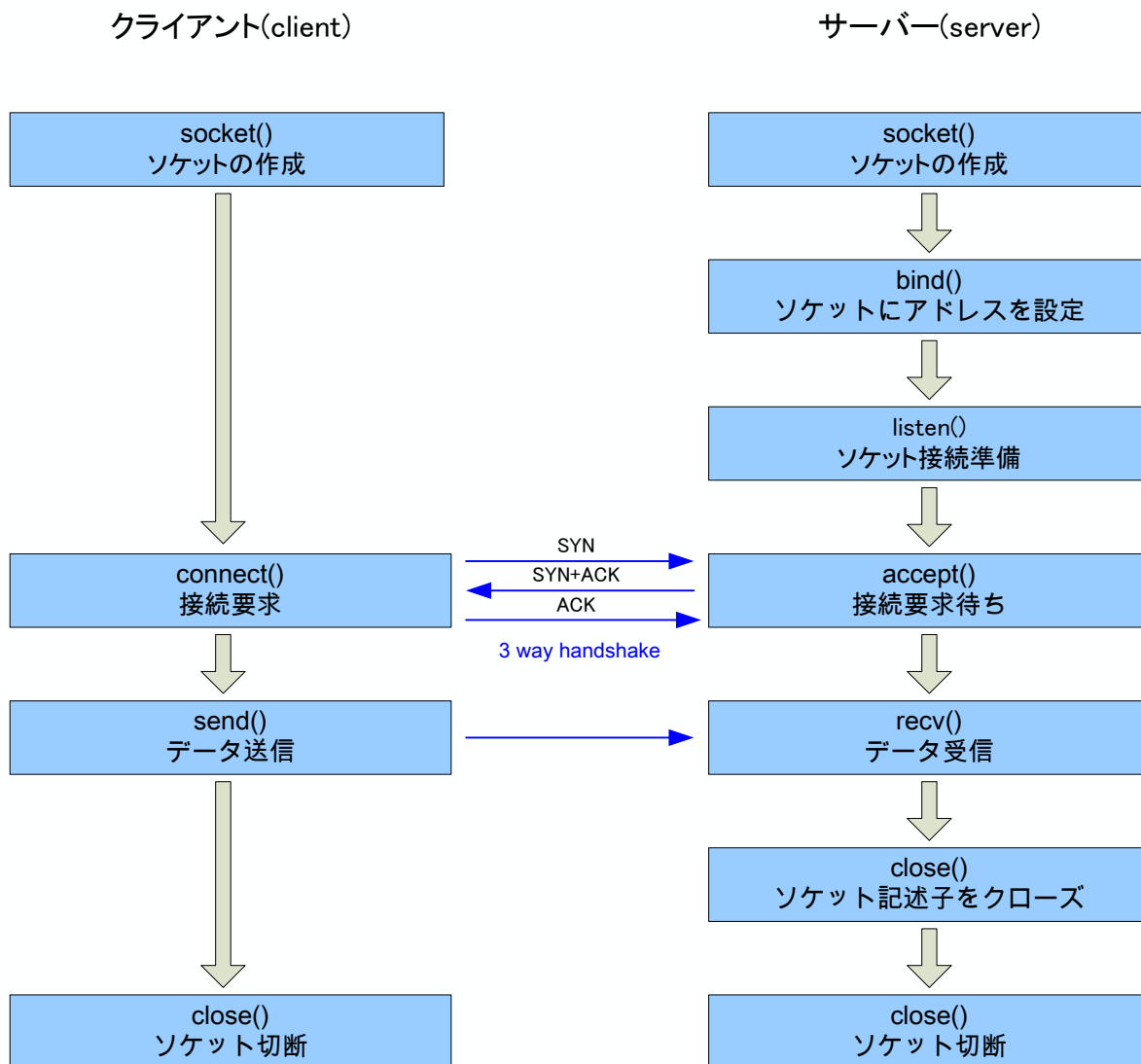


図 1: 通信の順序

3 通信の手続き

3.1 システムコール

通信の様々な処理は、OS; 諸君の学習環境では linux の仕事である。OS にその処理を任せることにより、通信のプログラムは格段に簡単になる。なぜならば、コンピューター間で通信を行うためには、かなり煩雑な処理が必要で、それをいちいちプログラム中に記述すると、とてもプログラムが長くなる。OS に一括して処理をお願いするのである。

この OS に処理を依頼する命令をシステムコールと呼ぶ。システムコールは普通の関数と似ており、ほとんど区別できない。しかし「man socket」¹とすると、

```
SOCKET(2)                                Linux Programmer 's Manual                                SOCKET(2)
```

名前

socket - 通信のための端点 (endpoint) を作成する

書式

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

後は長いので省略

と表示され、最初の行は SOCKET(2) となっており、この 2 からシステムコールと分かる。一方、普通の関数の場合「sin」とすると、

```
SIN(3)                                    Linux Programmer 's Manual                                    SIN(3)
```

名前

sin, sinf, sinl - 正弦 (サイン) 関数

書式

```
#include <math.h>

double sin(double x);
float sinf(float x);
long double sinl(long double x);
```

-lm でリンクする。

後は長いので省略

と表示され、最初の行は sin(3) となっており、この 3 から通常の関数と分かる。

¹man はオンラインヘルプである「f」で次ページ「p」で前ページを見ることができる「q」で終了となる。

3.2 通信方法

インターネットを使った通信では、普通、クライアントプログラムとサーバープログラムのプログラムの2つのプログラムを実行させる。ここでの例では、データを送る方をクライアント、受け取り表示する方をサーバーとしている。大体は、接続を待っている方がサーバーで、接続要求をする方がクライアントとなる。

ここでの例のクライアントとサーバーの通信の手続きをまとめると、図??のようになる。大体、インターネットを使ったプログラムは、これと似たような手続きを行う。

それでは、これから通信手続きの具体的な内容を見よう。

3.2.1 ソケット作製

ソケットは、通信のための端点 (endpoint) のことである。

`socket()` システムコールを使うことによりソケットを作製できる。このシステムコールの戻り値と引数は、

```
ファイルディスクリプター (整数値)=socket(domain, type, protocol)
```

となっている。

戻り値は、ファイルディスクリプターと呼ばれる整数値である。この整数値により、複数のソケットを区別している。ソケットどころか、入出力はすべてこのファイルディスクリプターで区別している。

引数の `domain` では、通信のプロトコルファミリーを指定する。プロトコルとは通信規約のことである。通常のインターネットの通信であれば、`PF_INET` を指定する。

第2引数の `type` は、通信方式をを定める。次に示す2通りの指定ができる。

`SOCK_STREAM` 信頼性の確保された通信が必要なとき。`SOCK_DGRAM` 信頼性は無いが、リアルタイムのデータを送るのに適している。

第3引数の `protocol` には、0を指定する。すると、TCPかUDPの適した方が選択される。

3.2.2 その他のシステムコール

その他のシステムコールの内容は課題とする。

4 プログラム作成の練習

[練習1] リスト1とリスト2を実行させて、自分自身と会話せよ。二つターミナル(端末)を開いて、その間で通信を行う。

[練習2] 通信に使ったファイルディスクリプターの番号を調べよ。`socket()`と`accept()`システムコールの戻り値である。

- [練習 3] `accept()` システムコールの第 2 引数を調べることにより，接続要求を行ってきたクライアントのアドレスファミリーと IP アドレス，ポート番号を調べよ．
- [練習 4] `send()` と `recv()` システムコールではなく，`write()` と `read()` システムコールを使ったプログラムに書き直せ．
- [練習 5] 複数のクライアントから，受信できるようにプログラムを改良せよ．これはかなり難しい．
- [練習 6] 時間の余った者は，教科書を読み，課題のレポートを作成せよ．

5 課題

5.1 内容

以下の課題を実施し，レポートとして提出すること．

[問 1] (復予) 教科書 [1]7 章「ネットワークプログラミング」pp.247-312 を 1 回読め．そして，以下について，答えよ．

- システムコール
- プロトコル
- パケット

[問 2] 次のシステムコールのも動作内容を述べ，戻り値と引数を示せ．

- `bind()`
- `listen()`
- `connect()`
- `accept()`
- `send()`
- `recv()`
- `close()`

5.2 レポート 提出要領

期限	11 月 7 日 (水) AM 8:45
用紙	A4 のレポート用紙．左上をホッチキスで綴じて，提出のこと．
提出場所	山本研究室の入口のポスト
表紙	表紙を 1 枚つけて，以下の項目を分かりやすく記述すること． 授業科目名「情報処理応用」 課題名「課題 ネットワーク接続の手続き」 提出日 2E 学籍番号 氏名
内容	2 ページ以降に問いに対する答えを分かりやすく記述すること．

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.