

授業名	情報科学演習 C
課題名	課題 4
学籍番号	09B18075
所属コース	計算機科学コース
名前	山口雄翔

1 課題 4-1

1.1 課題内容

資料で示したプロトコル仕様にしたがって、サンプルプログラムと正しく通信できるようなチャットプログラム (クライアントおよびサーバプログラム) を作成する。

1.2 プログラムの仕様

サーバを立ち上げ、そこにクライアントから接続要請を送るとサーバが名前などをチェックしたのちユーザーとして登録する。登録済みのクライアントから入力があればそれがサーバに届き、サーバはその入力の内容を登録されているすべてのクライアントに送信する。クライアントはそれを受信すると画面へ出力する。クライアントから EOF がサーバに送られてくると、サーバはそのクライアントを離脱させる。

1.3 実行結果

1.3.1 サンプルサーバと作成したクライアント

サーバ

```
./chatserver
ICS Exercises C sample program chatserver
user<yusho1> connected
user<yusho2> connected
user<yusho3> connected
user<yusho4> connected
connection failed. username already in use:yusho4
user<yusho5> connected
user<yusho1> closed
user<yusho2> closed
user<yusho3> closed
user<yusho4> closed
user<yusho5> closed
^C
```

クライアント

```
./chatclient exp002 yusho4
connected
accepted
yusho1 >dsafdsafads
sdfsfadf
yusho4 >sdfsfadf
sdfafdsafsdfa
yusho4 >sdfafdsafsdfa
yusho1 >sdfafsa
```

最大参加可能数を超えていた時のクライアント

```
./chatclient exp002 yusho6
connected
error1
```

同じ名前のユーザーが既にいた時のクライアント

```
./chatclient exp002 yusho4
connected
accepted
error : USERNAME REJECTED
```

1.3.2 作成したサーバとサンプルクライアント

サーバー

```
./chatserver
the username of the new member is yusho1

the username of the new member is yusho2

the username of the new member is yusho3

the username of the new member is yusho4

yusho4
have already exist
the username of the new member is yusho5

too many clients
left from chat : yusho1
left from chat : yusho2
left from chat : yusho3
left from chat : yusho4
left from chat : yusho5
^C
```

クライアント

```
./chatclient exp002 yusho1
ICS Exercises C sample program chatclient.c
connected to exp002
join request accepted
user name registered
sadsadfasd
yusho1 > sadsadfasd
yusho2 > dsafgsadfdsa
yusho2 > sadsadfsad
sdasadffads
yusho1 > sdasadffads
```

最大参加可能数を超えていた時のクライアント

```
./chatclient exp002 yusho6
ICS Exercises C sample program chatclient.c
connected to exp002
join request rejected
```

同じ名前のユーザーが既にいた時のクライアント

```
./chatclient exp002 yusho4
ICS Exercises C sample program chatclient.c
connected to exp002
join request accepted
USERNAME REJECTED
```

1.3.3 作成したサーバと作成したクライアント

サーバー

```
./chatserver
the username of the new member is yusho1

the username of the new member is yusho2

the username of the new member is yusho3

the username of the new member is yusho4

yusho4
have already exist
the username of the new member is yusho5

too many clients
left from chat : yusho4
left from chat : yusho1
left from chat : yusho2
left from chat : yusho3
left from chat : yusho5
^C
```

クライアント

```
./chatclient exp002 yusho4
connected
accepted
asdffasd
yusho4 > asdffasd
wwwwww
yusho4 > wwwwww
yusho1 > safdasdfasfasdffsdafsd
```

最大参加可能数を超えていた時のクライアント

```
./chatclient exp002 yusho6
connected
error1
```

同じ名前のユーザーが既にいた時のクライアント

```
./chatclient exp002 yusho4
connected
accepted
error : USERNAME REJECTED
```

1.4 実現方法

1.4.1 サーバー

サーバーはソケットを生成しクライアントからの入力待つ。新規のクライアントから接続要請が来たら名前のチェックをしたのち登録し、新たなソケットを `accept` してそのクライアントに割り当てる。そこで登録したソケットから受信があった場合、それは登録済みのクライアントから入力があったということなのでその内容を `read` してそれをどのユーザーが送ったかの情報を加えた後登録した各ソケットに `write` することによってすべての登録済みクライアントへ送信する。このソケットからの受付待ちの処理は `select` を用いる。クライアントから EOF の入力があればそのソケットを `close` することによってそのユーザーの離脱処理を行う。名前の登録の解除や接続ユーザー数の変更も同時に行う。

1.4.2 クライアント

`sock` 関数で生成したソケットを用いてサーバーに接続要請を行う。受理されると、`select` 関数によってソケットと標準入力を見張る。標準入力に入力された文字列はソケットへ `write` する。ソケットから送られてきた文字列はサーバーが配信したものであるため `read` で読み取り画面に出力する。サーバープログラムが終了されるとそれをソケットからの入力から読み取り、ソケットを閉じて自分のプログラムも終了する。

2 課題 4-2

2.1 課題内容

課題 4-1 のチャットに機能を付け加えて改良する。

2.2 プログラムの仕様

`ctrl-C` でサーバープログラムを終了したとき、3 秒後にプログラムが終了するというメッセージをクライアントの画面に出力し、3 秒待機した後終了する。

2.3 実行結果

2.3.1 サーバー

```
./server2
the username of the new member is yusho2

^CTerminated
```

2.3.2 クライアント

```
./chatclient exp017 yusho2
connected
accepted
afdsfas
yusho2 > afdsfas
df
yusho2 > df
This program will end in 3 seconds.
closed
```

2.4 実現方法

signal 関数により ctrl-C が押された時シグナルハンドラ内の処理が行われるようにする。シグナルハンドラの中身はメッセージをクライアントに送信した後 3 秒 sleep し、そのあと sigterm を自分のプロセスに送ることによってプログラムを終了する。なおクライアントプログラムは一切変更していない。

3 考察、工夫した点、強調したい点

chat 型配列の送受信のところで、サンプルプログラムと自作プログラムで噛み合わずエラーが起こることが多かったので、サンプルプログラムはそのこのプロトコルに関して厳格に作られていると考えられる。

4 感想、意見、疑問

select を使う際の各種設定 (FD_SET など) の意味が分かっていなかったため、はじめのうちはそこに関するエラーが多かった。あまり使ったことのない関数を用いる際はその関数の man ページなどをよく調べる必要があると感じた。