

```

1: #include <stdio.h>
2: #include <sys/socket.h>
3: #include <netinet/in.h>
4: #include <stdlib.h>
5: #include <string.h>
6: #include <fcntl.h>
7:
8: #define DMAX 200
9: #define MAX 256
10: #define SERVER_PORT 45000
11: #define PENDING 10
12:
13: int DATA_SEND_RECV(int a);
14: void s_Chats(int a);
15: void s_Upload(int a);
16: void s_Download(int a);
17: void s_ListFile(int a);
18: void s_ListUser(int a);
19: int s_Exit(int a);
20:
21: void Chat(void);
22: void Upload(void);
23: int Download(int a);
24: void ListFile(void);
25: void ListUser(void);
26: int Exit(void);
27:
28: void error_handling(char* message);
29: int Receive(int a);
30:
31: char* Flaglist[6] = {"FLAG_CHAT", "FLAG_UPLOAD", "FLAG_DOWNLOAD", "FLAG_LIST_FILE",
"FLAG_LIST_USER", "FLAG_EXIT"};
32: char* tag[3] = {NULL,};
33: char user[12] = "";
34: char buf[MAX];
35: char data[DMAX];
36:
37: int main(int argc, char* argv[])
38: {
39:     strcpy(user, argv[1]);
40:     int sock_flag, conn_flag;
41:     struct sockaddr_in server_addr;
42:
43:     if((sock_flag = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0){
44:         printf("Socket i\203\235i\204± i\213±i\214" ... \n");
45:         exit(0);
46:     }
47:     else
48:         printf("Socket i\203\235i\204± i\204±ê³µ ... \n");
49:
50:     server_addr.sin_family = AF_INET;
51:     server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
52:     server_addr.sin_port = htons(SERVER_PORT);
53:
54:     if((connect(sock_flag, (struct sockaddr*)&server_addr, sizeof
(server_addr))) < 0){
55:         printf("i\204\234ê²\204-i\201'ê\235%i\235'i\226,i\212, i\227°ê²°
i\213±i\214" \n");
56:         exit(0);
57:     }
58:     else
59:         printf("i\204\234ê²\204-i\201'ê\235%i\235'i\226,i\212, i\227°ê²°
i\204±ê³µ \n");
60:
61:     DATA_SEND_RECV(sock_flag);
62:
63:     close(sock_flag);

```

```

64:
65: }
66:
67: int DATA_SEND_RECV(int sock_flag)
68: {
69:
70:     char sBuf[MAX];
71:     int sflagnum = 0;
72:     int flagnum = 0;
73:     int checkfinish = 0;
74:     int count = 0;
75:
76:     //char temp[10];
77:     //while(true){...}
78:
79:     while(1){
80:         for(int i = 0; i < 6; i++){
81:             printf("(%d)%s, ", i+1, Flaglist[i]);
82:         }
83:         while(1)
84:         {
85:             printf("\nEnter flag number: ");
86:             if(scanf("%d", &sflagnum) != 0 && sflagnum < 7)
87:                 break;
88:             if(count > 4)
89:                 exit(1);
90:             count++;
91:
92:             }sflagnum--;
93:
94:             memset(buf, 0x00, MAX);
95:             memset(sBuf, 0x00, MAX);
96:
97:             switch(sflagnum)
98:             {
99:                 case 0:
100:                     fgets(sBuf, MAX, stdin);
101:                     s_Chats(sock_flag);
102:                     break;
103:                 case 1:
104:                     s_Upload(sock_flag);
105:                     printf("i\227\205ê; \234ê\223\234ê\201\235");
106:                     break;
107:                 case 2:
108:                     fgets(sBuf, MAX, stdin);
109:                     s_Download(sock_flag);
110:                     break;
111:                 case 3:
112:                     s_ListFile(sock_flag);
113:                     break;
114:                 case 4:
115:                     s_ListUser(sock_flag);
116:                     break;
117:                 case 5:
118:                     s_Exit(sock_flag);
119:                     break;
120:                 default:
121:                     break;
122:             }
123:
124:             //i\206;i\213
125:
126:             printf("\ni\210\230i\213 i\213\234i\236\221\n\n");
127:             memset(buf, 0x00, MAX);
128:             read(sock_flag, buf, sizeof(buf)); //i\210\230i\213
129:             printf("i\210\230i\213 i\231\204êf\214\n\n");
130:             if(read == NULL)

```

```

131:         {
132:             printf("i\210\230i\213 ä\220\234 ë@224i\204,i$200ë°\200
i\227\220ë\237~");
133:             break;
134:         }
135:         printf("receive success: %s\n", buf);
136:
137:         int i = 0;
138:         char *ptr = strtok(buf, "|");
139:
140:         while(ptr!=NULL)
141:         {
142:             tag[i] = ptr;
143:             i++;
144:             ptr = strtok(NULL, "|");
145:         }
146:
147:         // printf("ë\215°i\235`i\204° ë¶\204ë\215 Flag: %s, User: %s, Data:
%s\n", tag[0], tag[1], tag[2]);
148:
149:         for(int i=0;i<6;i++)
150:         {
151:             if(tag[0]!=NULL)
152:             {
153:                 if(strcmp(tag[0],Flaglist[i]) == 0)
154:                 {
155:                     flagnum = i;
156:                     break;}}
157:
158:         switch(flagnum)
159:         {
160:             case 0:
161:                 Chat();
162:                 break;
163:             case 1:
164:                 Upload();
165:                 break;
166:             case 2:
167:                 Download(sock_flag);
168:                 break;
169:             case 3:
170:                 ListFile();
171:                 break;
172:             case 4:
173:                 ListUser();
174:                 break;
175:             case 5:
176:                 checkfinish = Exit();
177:                 break;
178:             default:
179:                 break;
180:         }
181:
182:         //write(sock_flag, buf, sizeof(buf));//i\206;i\213
183:         if(checkfinish == 1)
184:             break;
185:         memset(buf,0x00,MAX);
186:         tag[0]=NULL;
187:         tag[1]=NULL;
188:         tag[2]=NULL;
189:
190:         printf(
191:
192:
193:         }
194: }

```

```

195:
196:
197: void s_Chat(int sock_flag)
198: {
199:     memset(data, 0x00, MAX);
200:     printf("Enter data: ");
201:     fgets(data, DMAX, stdin);
202:
203:     strcat(buf, Flaglist[0]);
204:     strcat(buf, "|");
205:     strcat(buf, user);
206:     strcat(buf, "|");
207:     strcat(buf, data);
208:
209:     write(sock_flag, buf, sizeof(buf));
210:     printf("send success: %s", buf);
211: }
212:
213: void s_Upload(int sock_flag)
214: {
215:     int fd, n;
216:     int DATA_SIZE = 200;
217:     char DATA[DATA_SIZE];
218:
219:     memset(buf, 0x00, MAX);
220:     memset(DATA, 0x00, DATA_SIZE);
221:
222:     fd = open("2020_NW_Final_Project.pdf", O_RDONLY);
223:     printf("file open i\204±ë³p\n");
224:     if (fd < 0)
225:         error_handling("open() error");
226:
227:     while (1)
228:     {
229:         memset(buf, 0x00, MAX);
230:         memset(DATA, 0x00, DATA_SIZE);
231:
232:         n=read(fd, DATA, DATA_SIZE-1);
233:         strcat(DATA, "\0");
234:         if(n < 0)
235:             error_handling("read() error");
236:         if(n == 0)
237:         {
238:             printf("this file is finish\n");
239:             break;
240:         }
241:
242:         strcpy(buf, Flaglist[1]);
243:         strcat(buf, "|");
244:         strcat(buf, user);
245:         strcat(buf, "|");
246:         strcat(buf, DATA);
247:         write(sock_flag, buf, sizeof(buf));
248:         printf("%s\n", buf);
249:     }
250:
251:     memset(DATA, 0x00, DATA_SIZE);
252:     memset(buf, 0x00, MAX);
253:     strcpy(buf, Flaglist[1]);
254:     strcat(buf, "|");
255:     strcat(buf, user);
256:     strcat(buf, "|");
257:     strcat(buf, "EOF");
258:     write(sock_flag, buf, sizeof(buf));
259:     printf("%s\n", buf);
260:     printf("i \204i\206;i\231\204ë£\214\n");
261:     close(fd);

```

```

262: }
263:
264: void s_Download(int sock_flag)
265: {
266:     memset(buf, 0x00, MAX);
267:     printf("Enter file name: ");
268:     fgets(data, DMAX, stdin);
269:
270:     strcat(buf, Flaglist[2]);
271:     strcat(buf, "|");
272:     strcat(buf, user);
273:     strcat(buf, "|");
274:     strcat(buf, data);
275:
276:     write(sock_flag, buf, sizeof(buf));
277:     printf("%s", buf);
278: }
279: void s_ListFile(int sock_flag)
280: {
281:     strcat(buf, Flaglist[3]);
282:     strcat(buf, "|");
283:     strcat(buf, user);
284:     strcat(buf, "|");
285:     strcat(buf, "NULL");
286:
287:     write(sock_flag, buf, sizeof(buf));
288:     printf("send success: %s", buf);
289: }
290: void s_ListUser(int sock_flag)
291: {
292:     strcat(buf, Flaglist[4]);
293:     strcat(buf, "|");
294:     strcat(buf, user);
295:     strcat(buf, "|");
296:     strcat(buf, "NULL");
297:
298:     write(sock_flag, buf, sizeof(buf));
299:     printf("send success: %s", buf);
300: }
301: int s_Exit(int sock_flag)
302: {
303:     strcat(buf, Flaglist[5]);
304:     strcat(buf, "|");
305:     strcat(buf, user);
306:     strcat(buf, "|");
307:     strcat(buf, "EXIT");
308:
309:     write(sock_flag, buf, sizeof(buf));
310:     printf("send success: %s", buf);
311: }
312:
313:
314: void Chat(void)
315: {
316:     char* Data = tag[2];
317:     printf("From server data: %s\n", Data);
318: }
319:
320: void Upload(void)
321: {
322:     char* Data = tag[2];
323:     printf("From server: %s\n", Data);
324: }
325:
326: int Download(int sock_flag)
327: {
328:     int fd;

```

```

329:     char temp[200];
330:     char* Data = tag[2];
331:
332:     if(strcmp("[ERR] Not found file.", Data)==0)
333:     {
334:         printf("From server: %s\n", Data);
335:         return 0;
336:     }
337:
338:     fd = open("download.pdf", O_WRONLY|O_CREAT|S_IXUSR);
339:     if (fd < 0)
340:         error_handling("file open error\n");
341:
342:     memset(temp, 0x00, 200);
343:
344:     int flag_num;
345:     while(1)
346:     {
347:
348:         flag_num = Receive(sock_flag);
349:         if(flag_num!=2)
350:         {
351:             printf("Download Error!");
352:             break;
353:         }
354:         memset(temp, 0x00, 200);
355:         strcpy(temp, tag[2]);
356:         // printf("%s", temp);
357:         if(strcmp(temp, "EOF")==0 || strcmp(temp, "[Download] Done.")==0)
358:             break;
359:         if(write(fd, temp, sizeof(temp)) < 0)
360:             error_handling("write() error");
361:         lseek(fd, 199, SEEK_CUR);
362:     }
363:     close(fd);
364:     printf("finish download\n");
365:     return 0;
366: }
367: void error_handling(char* message)
368: {
369:     printf("%s", message);
370:     fputc('\n', stderr);
371:     exit(1);
372: }
373: void ListFile(void)
374: {
375:     char* Data = tag[2];
376:     printf("From server: %s\n", Data);
377: }
378: void ListUser(void)
379: {
380:     char* Data = tag[2];
381:     printf("From server: %s\n", Data);
382: }
383: int Exit(void)
384: {
385:     char* Data = tag[2];
386:     if(strcmp(Data, "BYE")==0)
387:     {
388:         printf("i\204\234ë²\204 i¢\205ë£\214..\n");
389:         return 1;
390:     }
391:     else
392:         return 0;
393: }
394:
395: int Receive(int sock_flag)

```

```
396: {
397:     int flagnum=0;
398:
399:     memset(buf, 0x00, MAX);
400:     read(sock_flag, buf, sizeof(buf)); //i\210\230i\213
401:     //printf("i\210\230i\213 i\231\204ë\214\n\n");
402:
403:     if(read == NULL)
404:     {
405:         printf("i\210\230i\213 ë\220\234 ë\224i\204,i$200ë°\200
i\227\220ë\237~");
406:         return 6;
407:     }
408:     int i = 0;
409:     char *ptr = strtok(buf, "|");
410:     while(ptr!=NULL)
411:     {
412:         tag[i] = ptr;
413:         i++;
414:         ptr = strtok(NULL, "|");
415:     }
416:
417:     for(int i=0;i<6;i++)
418:     {
419:         if(tag[i]!=NULL)
420:         {
421:             if(strcmp(tag[i],Flaglist[i]) == 0)
422:             {
423:                 flagnum = i;
424:                 break;
425:             }
426:         }
427:     }
428:     return flagnum;
429: }
```