

بسم الله الرحمن الرحيم



دانشکده مهندسی برق و کامپیوتر

شبکه‌های پیشرفته - تمرین دوم

سید مهدی رضوی

استاد : آقای دکتر یزدانی

آبان ماه ۱۴۰۲

فهرست مطالب

۳	۱ تمرین تشریحی
۳	۱.۱ تمرین اول
۴	۲.۱ تمرین دوم
۵	۳.۱ تمرین سوم
۶	۴.۱ تمرین چهارم
۷	۲ تمرین عملی
۱۱	۳ تمرین امتیازی
۱۱	۱.۳ امتیازی اول
۱۱	۲.۳ امتیازی دوم

فهرست تصاویر

۳	active passive	۱
۷	bind server listen client	۲
۷	checking username and password	۳
۸	ls command	۴
۸	ls execution	۵
۹	mk mv rm commands	۶
۹	Retrieve Execution	۷
۱۰	Quit Execution	۸

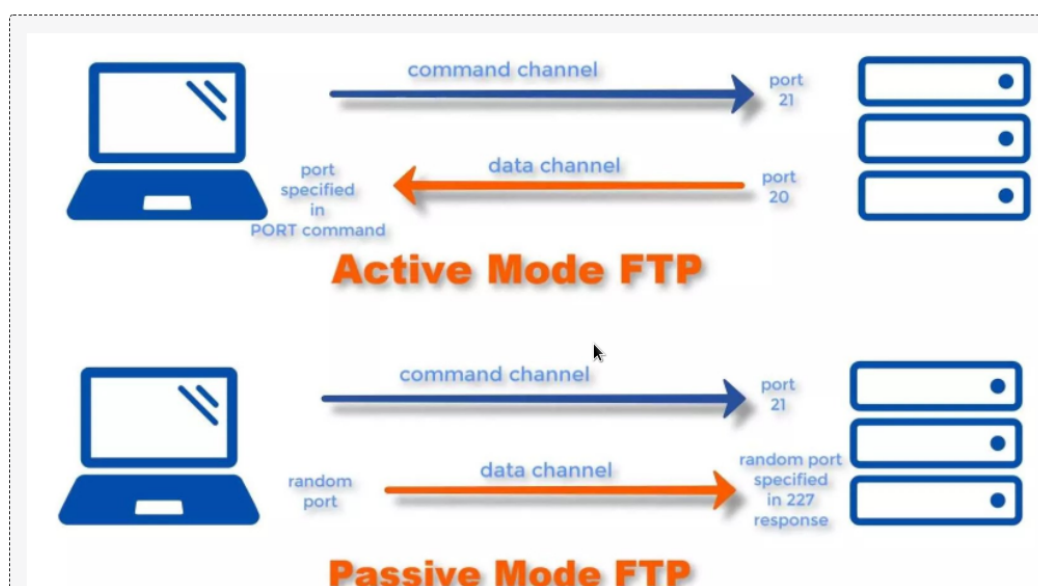
۱ تمرین تشریحی

۱.۱ تمرین اول

تفاوت اصلی بین این دو روش انتقال در FTP در واقع سمت برقراری ارتباط است. در حالت Passive این سرور است که در پاسخ به کلاینت ارتباط برقرار می‌کند که در بسیاری از فایروال‌ها ارتباط از بیرون به داخل مسدود شده‌است. در حالت Active این کلاینت است که با دسترسی که در همان پورت اول دریافت کرده‌است به پورت تصادفی جدیدی که در سرور در نظر گرفته شده‌است متصل می‌شود. بنابراین Session اضافی نیاز نیست از بیرون به داخل شبکه تعریف شود. در واقع یک بحث امنیتی است که تفاوت این دو روش را تعیین می‌کند.

در حالت Active FTP کلاینت از طریق یک پورت تصادفی یا Random Port دستور خودش را به سمت پورت Command Channel یا همان پورت ۲۱ ارسال می‌کند. دلیل اینکار این است که نیازی نیست الزاماً در سمت کلاینت از همان پورتهی که سرور استفاده می‌کند برای ارسال کردن دستور به پورت ۲۱ استفاده شود. بعد از اینکه دستور از طریق پورت ۲۱ در حالت Active FTP دریافت شد، سرور از طریق پورت ۲۰ که Data Channel است، یک شماره پورت تصادفی برای ارتباط‌گیری کلاینت با پورت ۲۰ خودش به سمت کلاینت ارسال می‌کند. (در واقع کلاینت از روی پورت ۲۰ خودش برای برقراری ارتباط با سرور درخواست می‌دهد که کلاینت با یک پورت تصادفی دیگر که معمولاً یک شماره بالاتر از پورتهی که کلاینت در زمان ارسال درخواست به پورت ۲۱ ارسال کرده‌است به سرور متصل شود و شروع به انتقال داده کند.)

قطعا در این میان یک فرآیند Port Forwarding انجام می‌شود در دو مرحله (یکبار در سمت سرور که پورت تصادفی کلاینت را دریافت و به سمت پورت ۲۱ منتقل می‌کند و یکبار در سمت برگشت در سمت کلاینت که پورت تصادفی سرور برای ارتباط‌گیری را باز و ارتباط را برقرار می‌کند). و در نهایت کلاینت بر روی پورت ۲۰ سرور ارتباط برقرار کرده و شروع به انتقال فایل می‌کند. در قدیم این امکان وجود داشت که پروتکل‌ها بصورت مستقیم ارتباط بگیرند اما این روزها با وجود فایروال‌ها این امکان وجود ندارد.



شکل ۱: active passive

۲.۱ تمرین دوم

حالت انتقال در FTP به شیوه‌ای است که داده‌ها بین سرور و کلاینت منتقل می‌شوند. دو حالت انتقال در FTP عبارتند از :

۱. حالت باینری (Binary Mode) : در این حالت ، داده‌ها به صورت بایت به بایت منتقل می‌شوند ، بدون هیچ تغییر یا تبدیلی. این حالت برای انتقال فایل‌های دودویی مانند تصاویر ، ویدیوها و فایل‌های صوتی استفاده می‌شود.

۲. حالت ASCII (ASCII Mode) : در این حالت ، داده‌ها به صورت متنی منتقل می‌شوند و تغییراتی در آن‌ها اعمال می‌شود. برای مثال در این حالت کاراکترهای خاص مانند تب ، فاصله ، علامت نقطه و خط جدید به صورت مخصوص تبدیل می‌شوند.

انتخاب هر یک از حالات به شرایط زیر بستگی دارد:

- اگر فایل‌های دودویی مانند تصاویر، ویدئوها و فایل‌های صوتی را منتقل می‌کنید، حالت باینری را انتخاب کنید.
 - اگر فایل‌های متنی را منتقل می‌کنید و نیاز به تغییر کاراکترهای خاص دارید ، حالت ASCII را انتخاب کنید.
- انتخاب اشتباه حالت انتقال ممکن است عواقب زیر را به همراه داشته باشد:
- اگر فایل‌های دودویی را در حالت ASCII منتقل کنید ، ممکن است تغییراتی در داده‌ها اعمال شود و فایل خراب شود.
 - اگر فایل‌های متنی را در حالت باینری منتقل کنید، ممکن است کاراکترهای خاص به صورت نادرست تبدیل شوند و مشکلات خواندن فایل ایجاد شود.

۳.۱ تمرین سوم

برای ارسال فایل‌های بزرگ یا چندین فایل با استفاده از پروتکل FTP به صورت کارآمد ، می‌توان از موارد زیر استفاده کرد:

۱. استفاده از حالت باینری : برای انتقال فایل‌های دودویی مانند تصاویر و ویدئوها، حالت باینری را انتخاب کنید. این حالت داده‌ها را بایت به بایت منتقل می‌کند بدون هیچ تغییر یا تبدیلی. با استفاده از این حالت، می‌توانید فایل‌های بزرگ را به صورت کارآمد انتقال دهید.
۲. استفاده از فشرده‌سازی : با استفاده از الگوریتم‌های فشرده‌سازی مانند RAR ، ZIP یا GZIP حجم فایل‌ها را کاهش داده و زمان انتقال آن‌ها نیز کاهش می‌یابد.
۳. استفاده از پروتکل FTPS یا SFTP : برای ارسال فایل‌ها به صورت امن و رمزنگاری شده ، می‌توانید از پروتکل FTPS با SSL/TLS استفاده کنید.
۴. استفاده از نرم‌افزارهای مدیریت FTP : برخی از نرم‌افزارهای مدیریت FTP قابلیت‌های خاصی برای ارسال فایل‌های بزرگ یا چندین فایل دارند. به عنوان مثال، برخی از نرم‌افزارها قابلیت تقسیم فایل‌های بزرگ به قطعات کوچکتر را دارند و هر قطعه را به صورت جداگانه ارسال می‌کنند. در سمت سرور، این قطعات بازیابی و به صورت کامل فایل را تشکیل می‌دهند.
۵. استفاده از اتصالات پرسرعت : برای ارسال فایل‌های بزرگ ، استفاده از اتصالات پرسرعت مانند اتصالات با فیبرنوری یا شبکه‌های LTE یا 5G می‌تواند کارآمد باشد. با استفاده از این اتصالات ، زمان انتقال فایل‌ها به صورت چشمگیری کاهش می‌یابد.

۴.۱ تمرین چهارم

۱. SFTP (SSH File Transfer Protocol)

یک پروتکل امن که بر روی Secure Shell تکیه دارد. این پروتکل از رمزنگاری و احراز هویت برای انتقال فایل استفاده می‌کند که امنیت بالایی را برای انتقال فایل فراهم کند.

۲. SCP(Secure Copy Protocol)

یک پروتکل به مانند پروتکل SFTP می‌باشد با این تفاوت که بیشتر برای کپی کردن فایل بین کلاینت و سرور استفاده می‌شود و قابلیت‌های کمتری به نسبت SFTP دارد. این پروتکل نمی‌تواند مانند SFTP فایلی را بسازد و یا حذف کند.

۳. TFTP(Trivial File Transfer Protocol)

یک پروتکل ساده و کوچک برای انتقال فایل است که در برخی از سیستم‌ها و دستگاه‌های شبکه استفاده می‌شود. این پروتکل کمترین تعداد قابل قبولی از ویژگی‌ها را دارد در مقایسه با پروتکل FTP. این پروتکل کمترین تعداد برای ارسال فایل‌های کوچک و بدون نیاز به امنیت استفاده می‌شود.

۲ تمرین عملی

برای سهولت پیاده‌سازی در قسمت کلاینت این پروتکل یک سری کامندهایی را تعریف کرده‌ایم و سپس از کاربر خواسته‌ام که اگر هر کدام از این کامندها را درخواست کند با توجه به آن کامند ، عملیات مدنظر را در سمت سرور انجام بدهند.

در نتیجه ما با استفاده از یک Switch Case مدیریت خطا را نیز کنترل کرده‌ایم.

همانطور که در تصاویر کدها نیز مشخص است ، کلاینت ما باید به عنوان یک کاربر معتبر mahdi و mahdi-pass را وارد کند.

سپس در صورت تایید احراز هویت ، کاربر می‌تواند دستور مدنظر خودش را وارد کند .

در تصاویر زیر کد به زبان Cpp و همچنان نتایج اجرای دستورات را در ترمینال مشاهده خواهید کرد.

```

rtp-server.cpp
35     exit(1);
36 }
37 server.sin_port = htons(9999);
38 server.sin_addr.s_addr = 0;
39 k = bind(sock1, (struct sockaddr*)&server, sizeof(server));
40 if(k == -1)
41 {
42     printf("Binding error \n");
43     exit(1);
44 }
45
46 k = listen(sock1, 1);
47 if(k == -1)
48 {
49     printf("Listen failed \n");
50     exit(1);
51 }
52
53 len = sizeof(client);
54 sock2 = accept(sock1, (struct sockaddr*)&client, (sockl
55 i = 1;
56
57 send(sock2, "Please enter your username : ", 100 , 0)
58 rcv(sock2, username , 100 , 0);
59 printf("username : %s\n", username);
60
rtp-client.cpp
38 server.sin_family = AF_INET;
39 server.sin_port = htons(9999);
40
41 server.sin_addr.s_addr = 0;
42 k = connect(sock, (struct sockaddr*)&server,
43     sizeof(server));
44 if(k == -1)
45 {
46     printf("Connect Error");
47     exit(1);
48 }
49 int i = 1;
50
51 char username[100] , password[100];
52 rcv(sock , buf , 100 , 0);
53 printf("%s" , buf);
54
55 scanf("%s", username);
56 send(sock , username , 100 , 0);
57
58
59 rcv(sock , buf , 100 , 0);
60 printf("%s" , buf);
61
62 scanf("%c" , password);

```

شکل ۲: bind server listen client

```

rtp-server.cpp
49     printf("Listen failed \n");
50     exit(1);
51 }
52
53 len = sizeof(client);
54 sock2 = accept(sock1, (struct sockaddr*)&client, (sockl
55 i = 1;
56
57 send(sock2, "Please enter your username : ", 100 , 0)
58 rcv(sock2, username , 100 , 0);
59 printf("username : %s\n", username);
60
61 send(sock2, "Please enter your password : ", 100 , 0)
62 rcv(sock2, password , 100 , 0);
63 printf("password : %s\n", password);
64
65 if(username != "mahdi" || password != "mahdi-pass")
66 {
67     printf("This is username : %s\n", username);
68     printf("This is password : %s\n", password);
69     // printf("Authentication failed \n" );
70     // send(sock1, "Authentication failed \n", 100 , 0)
71     //exit(1);
72     //break;
73
rtp-client.cpp
45 {
46     printf("Connect Error");
47     exit(1);
48 }
49 int i = 1;
50
51 char username[100] , password[100];
52 rcv(sock , buf , 100 , 0);
53 printf("%s" , buf);
54
55 scanf("%s", username);
56 send(sock , username , 100 , 0);
57
58
59 rcv(sock , buf , 100 , 0);
60 printf("%s" , buf);
61
62 scanf("%s", password);
63 send(sock , password , 100 , 0);
64
65 // rcv(sock , buf , 100 , 0);
66 // printf("\n This is buf : %s\n", buf);
67
68 // if(buf == "Authentication failed \n")
69 // {

```

شکل ۳: checking username and password

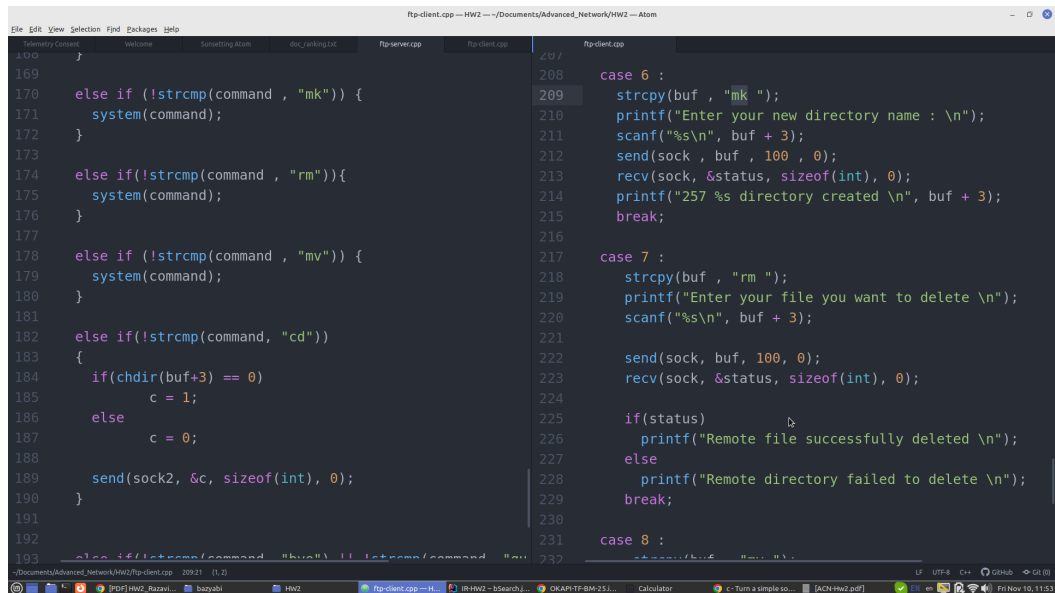
```
89
90 while(1)
91 {
92     recv(sock2, buf, 100, 0);
93     sscanf(buf, "%s", command);
94     if(!strcmp(command, "ls"))
95     {
96         system("ls -l > temps.txt");
97         i = 0;
98         stat("temps.txt", &obj);
99         size = obj.st_size;
100         send(sock2, &size, sizeof(int), 0);
101         filehandle = open("temps.txt", O_RDONLY);
102         sendfile(sock2, filehandle, NULL, size);
103     }
104
105     else if(!strcmp(command, "get"))
106     {
107         sscanf(buf, "%s%s", filename, filename);
108         stat(filename, &obj);
109         filehandle = open(filename, O_RDONLY);
110         size = obj.st_size;
111         if(filehandle == -1)
112             size = 0;
113
167 break;
168
169 case 4:
170     strcpy(buf, "ls -l");
171     send(sock, buf, 100, 0);
172     recv(sock, &size, sizeof(int), 0);
173     f = (char*)malloc(size);
174     recv(sock, f, size, 0);
175     printf("The remote directory listing is as follows:\n");
176     // system("chown temp.txt");
177     system("cat temps.txt");
178
179     // filehandle = creat("temp.txt", O_CREAT | O_WRONLY);
180
181     //fstream my_file;
182     // my_file.open("temp.txt" , ios::out);
183     // my_file << f;
184     // my_file.close();
185     // write(filehandle, f, size, 0);
186     // close(filehandle);
187
188     // printf("The remote directory listing is as follows");
189     // // system("chown temp.txt");
190
191
```

شکل ۴: ls command

```
mahdi@MahdiRazavi:~/Documents/Advanced_Network/HW2$ gcc ftp-server.cpp -lstdc++ -o server.o
mahdi@MahdiRazavi:~/Documents/Advanced_Network/HW2$ ./server.o
username : mahdi
password : mahdi-pass
This is username : mahdi
This is password : mahdi-pass

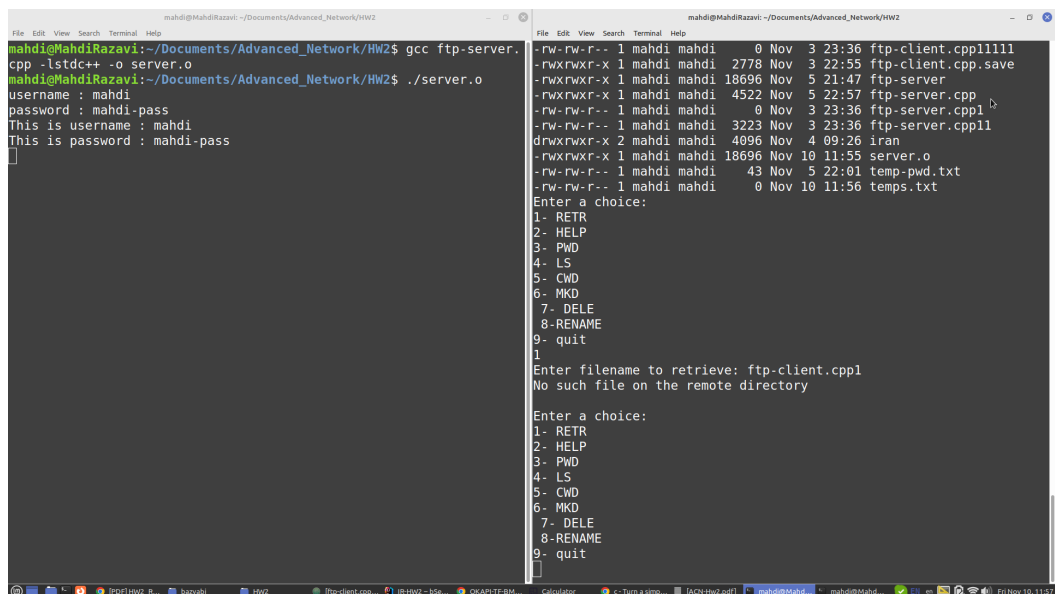
8-RENAME
9- quit
4
The remote directory listing is as follows:
total 144
-rwxrwxr-x 1 mahdi mahdi 18520 Nov  5 09:44 a.out
-rwxrwxr-x 1 mahdi mahdi 18520 Nov 10 11:55 client.o
-rw-rw-r-- 1 mahdi mahdi    57 Nov  5 09:59 configuration.txt
-rwxrwxr-x 1 mahdi mahdi 18520 Nov  5 21:33 ftp-client
-rwxrwxr-x 1 mahdi mahdi 6038  Nov 10 11:55 ftp-client.cpp
-rw-rw-r-- 1 mahdi mahdi    0 Nov  3 23:24 ftp-client.cpp1
-rw-rw-r-- 1 mahdi mahdi 3871 Nov  3 23:26 ftp-client.cpp11
-rw-rw-r-- 1 mahdi mahdi    0 Nov  3 23:30 ftp-client.cpp111
-rw-rw-r-- 1 mahdi mahdi 3882 Nov  3 23:31 ftp-client.cpp1111
-rw-rw-r-- 1 mahdi mahdi    0 Nov  3 23:36 ftp-client.cpp11111
-rwxrwxr-x 1 mahdi mahdi 2778 Nov  3 22:55 ftp-client.cpp.save
-rwxrwxr-x 1 mahdi mahdi 18696 Nov  5 21:47 ftp-server
-rwxrwxr-x 1 mahdi mahdi 4522 Nov  5 22:57 ftp-server.cpp
-rw-rw-r-- 1 mahdi mahdi    0 Nov  3 23:36 ftp-server.cpp1
-rw-rw-r-- 1 mahdi mahdi 3223 Nov  3 23:36 ftp-server.cpp11
drwxrwxr-x 2 mahdi mahdi 4096 Nov  4 09:26 iran
-rwxrwxr-x 1 mahdi mahdi 18696 Nov 10 11:55 server.o
-rw-rw-r-- 1 mahdi mahdi   43 Nov  5 22:01 temp-pwd.txt
-rw-rw-r-- 1 mahdi mahdi    0 Nov 10 11:56 temps.txt
Enter a choice:
1- RETR
2- HELP
3- PWD
4- LS
5- CWD
6- MKD
7- DELE
8-RENAME
9- quit
```

شکل ۵: ls execution



```
169
170     else if (!strcmp(command, "mk")) {
171         system(command);
172     }
173
174     else if (!strcmp(command, "rm")){
175         system(command);
176     }
177
178     else if (!strcmp(command, "mv")) {
179         system(command);
180     }
181
182     else if (!strcmp(command, "cd"))
183     {
184         if(chdir(buf+3) == 0)
185             c = 1;
186         else
187             c = 0;
188         send(sock2, &c, sizeof(int), 0);
189     }
190 }
191
192
193     else if (!strcmp(command, "bye")) { if (!strcmp(command, "bye")) {
207
208     case 6 :
209         strcpy(buf, "mk ");
210         printf("Enter your new directory name : \n");
211         scanf("%s\n", buf + 3);
212         send(sock, buf, 100, 0);
213         recv(sock, &status, sizeof(int), 0);
214         printf("257 %s directory created \n", buf + 3);
215         break;
216
217     case 7 :
218         strcpy(buf, "rm ");
219         printf("Enter your file you want to delete \n");
220         scanf("%s\n", buf + 3);
221
222         send(sock, buf, 100, 0);
223         recv(sock, &status, sizeof(int), 0);
224
225         if(status)
226             printf("Remote file successfully deleted \n");
227         else
228             printf("Remote directory failed to delete \n");
229         break;
230
231     case 8 :
232         strcpy(buf, "mv ");
```

شکل ۶: mk mv rm commands



```
mahdi@MahdiRazavi:~/Documents/Advanced_Network/HW2$ gcc ftp-server.
cpp -lstdc++ -o server.o
mahdi@MahdiRazavi:~/Documents/Advanced_Network/HW2$ ./server.o
username : mahdi
password : mahdi-pass
This is username : mahdi
This is password : mahdi-pass

-rw-rw-r-- 1 mahdi mahdi 0 Nov 3 23:36 ftp-client.cpp1111
-rwxrwxr-x 1 mahdi mahdi 2778 Nov 3 22:55 ftp-client.cpp.save
-rwxrwxr-x 1 mahdi mahdi 18696 Nov 5 21:47 ftp-server
-rwxrwxr-x 1 mahdi mahdi 4522 Nov 5 22:57 ftp-server.cpp
-rw-rw-r-- 1 mahdi mahdi 0 Nov 3 23:36 ftp-server.cpp1
-rw-rw-r-- 1 mahdi mahdi 3223 Nov 3 23:36 ftp-server.cpp11
drwxrwxr-x 2 mahdi mahdi 4096 Nov 4 09:26 iran
-rwxrwxr-x 1 mahdi mahdi 18696 Nov 10 11:55 server.o
-rw-rw-r-- 1 mahdi mahdi 43 Nov 5 22:01 temp-pwd.txt
-rw-rw-r-- 1 mahdi mahdi 0 Nov 10 11:56 temps.txt
Enter a choice:
1- RETR
2- HELP
3- PWD
4- LS
5- CWD
6- MKD
7- DELE
8- RENAME
9- quit
1
Enter filename to retrieve: ftp-client.cpp1
No such file on the remote directory

Enter a choice:
1- RETR
2- HELP
3- PWD
4- LS
5- CWD
6- MKD
7- DELE
8- RENAME
9- quit
```

شکل ۷: Retrieve Execution

```
maledi@MahdiRazavi: ~/Documents/Advanced_Network/HW2
maledi@MahdiRazavi:~/Documents/Advanced_Network/HW2$ gcc ftp-server.cpp -lstdc++ -o server.o
maledi@MahdiRazavi:~/Documents/Advanced_Network/HW2$ ./server.o
username : mahdi
password : mahdi-pass
This is username : mahdi
This is password : mahdi-pass
FTP server quitting..
maledi@MahdiRazavi:~/Documents/Advanced_Network/HW2$

maledi@MahdiRazavi:~/Documents/Advanced_Network/HW2$
-rwxrwxr-x 1 mahdi mahdi 4522 Nov  5 22:57 ftp-server.cpp
-rw-rw-r-- 1 mahdi mahdi   0 Nov  3 23:36 ftp-server.cppl
-rw-rw-r-- 1 mahdi mahdi 3223 Nov  3 23:36 ftp-server.cppl1
drwxrwxr-x 2 mahdi mahdi 4096 Nov  4 09:26 iran
-rwxrwxr-x 1 mahdi mahdi 18696 Nov 10 11:55 server.o
-rw-rw-r-- 1 mahdi mahdi  43 Nov  5 22:01 temp-pwd.txt
-rw-rw-r-- 1 mahdi mahdi   0 Nov 10 11:56 temps.txt
Enter a choice:
1- RETR
2- HELP
3- PWD
4- LS
5- CWD
6- MKD
7- DELE
8-RENAME
9- quit
1
Enter filename to retrieve: ftp-client.cppl
No such file on the remote directory
Enter a choice:
1- RETR
2- HELP
3- PWD
4- LS
5- CWD
6- MKD
7- DELE
8-RENAME
9- quit
9
Server closed
Quitting..
maledi@MahdiRazavi:~/Documents/Advanced_Network/HW2$
```

شکل ۸: Quit Execution

۳ تمرین امتیازی

۱.۳ امتیازی اول

راه حل مساله ذکر شده استفاده از متغیر سمافور برای مدیریت چندین Thread می‌باشد.

```
include <semaphore.h>
int semwait(sem_t *sem);
```

۲.۳ امتیازی دوم

در ابتدا باید یک SSL Certificate داشته باشیم که حاوی گواهینامه با کلید خصوصی باشد .

```
include <openssl/applink.c>
include <openssl/bio.h>
include <openssl/ssl.h>
include <openssl/err.h>
```

سپس با استفاده از کتابخانه‌های فوق به ایجاد و حذف یک گواهینامه (SSL (Secure Sockets Layer می‌پردازیم.