PROGRAM CODE

server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <string.h>
#include <dirent.h>

void serverRecv(int client_fd) {
      char str[100];
      char str1[1000];

      if(recv(client_fd, str, 100 * sizeof(char), 0) <= 0) {
            printf("Connection lost!\n");
            return;
      }

      if(!strcmp(str, "GET")) {
            DIR *d;
            struct dirent *dir;
            d = opendir(".");
            str1[0] = '\0';

            if (d) {
                  while((dir = readdir(d)) != NULL) {
                        if(strcmp(dir->d_name, ".") && strcmp(dir->d_name, "..")
&& strcmp(dir->d_name, "server.c") && strcmp(dir->d_name, "server")) {
                              strcat(str1, dir->d_name);
                              strcat(str1, "\n");
                        }
                  }
                  closedir(d);
            }

            if(send(client_fd, str1, 1000 * sizeof(char), 0) <= 0) {
                  printf("GET failed!\n");
                  return;
            }

            if(send(client_fd, "200 OK", 7 * sizeof(char), 0) <= 0) {
                  printf("GET failed!\n");
                  return;
            } else {
                  printf("GET successful!\n");
                  return;
            }
      } if(!strcmp(str, "UPLOAD")) {
            if(recv(client_fd, str, 100 * sizeof(char), 0) <= 0) {
                  printf("UPLOAD failed!\n");
                  return;
            }

            if(recv(client_fd, str1, 1000 * sizeof(char), 0) <= 0) {
                  printf("UPLOAD failed!\n");
                  return;
            }

            FILE* fp = fopen(str, "w");
            fputs(str1, fp);
            fclose(fp);

            if(send(client_fd, "200 OK", 7 * sizeof(char), 0) <= 0) {
                  printf("UPLOAD failed!\n");
                  return;
            } else {
                  printf("UPLOAD successful!\n");
                  return;
            }
      } else if(!strcmp(str, "DOWNLOAD")) {
            if(recv(client_fd, str, 100 * sizeof(char), 0) <= 0) {
                  printf("DOWNLOAD failed!\n");
                  return;
            }

            FILE* fp = fopen(str, "r");

            if(fp == NULL) {
                  str1[0] = '\0';

                  if(send(client_fd, str1, sizeof(char), 0) <= 0) {
                        printf("DOWNLOAD failed!\n");
                        return;
                  }

                  if(send(client_fd, "400 BAD", 8 * sizeof(char), 0) <= 0) {
                        printf("DOWNLOAD failed!\n");
                        return;
                  }

                  return;
            }

            str1[0] = '\0';

            while(fgets(str, 100, fp) != NULL) {
                  strcat(str1, str);
            }

            fclose(fp);

            if(send(client_fd, str1, 1000 * sizeof(char), 0) <= 0) {
                  printf("DOWNLOAD failed!\n");
                  return;
            }

            if(send(client_fd, "200 OK", 7 * sizeof(char), 0) <= 0) {
                  printf("DOWNLOAD failed!\n");
                  return;
            } else {
                  printf("DOWNLOAD successful!\n");
                  return;
            }
      } else if(!strcmp(str, "RENAME")) {
            if(recv(client_fd, str, 100 * sizeof(char), 0) <= 0) {
                  printf("RENAME failed!\n");
                  return;
            }

            if(recv(client_fd, str1, 100 * sizeof(char), 0) <= 0) {
                  printf("RENAME failed!\n");
                  return;
            }
            if(!rename(str, str1)) {
                  if(send(client_fd, "200 OK", 7 * sizeof(char), 0) <= 0) {
                        printf("RENAME failed!\n");
                        return;
                  }
            } else {
                  if(send(client_fd, "400 BAD", 8 * sizeof(char), 0) <= 0) {
                        printf("RENAME failed!\n");
                        return;
                  } else {
                        printf("RENAME successful!\n");
                        return;
                  }
            }
      } else if(!strcmp(str, "DELETE")) {
            if(recv(client_fd, str, 100 * sizeof(char), 0) <= 0) {
                  printf("DELETE failed!\n");
                  return;
            }

            if(!remove(str)) {
                  if(send(client_fd, "200 OK", 7 * sizeof(char), 0) <= 0) {
                        printf("DELETE failed!\n");
                        return;
                  }
            } else {
                  if(send(client_fd, "400 BAD", 8 * sizeof(char), 0) <= 0) {
                        printf("DELETE failed!\n");
                        return;
                  } else {
                        printf("DELETE successful!\n");
                        return;
                  }
            }
      } else {
            printf("Invalid request from client!\n");
      }
}

void main(int argc, char* argv[]) {
      int PORT;

      if(argc == 2) {
            PORT = atoi(argv[1]);
      } else {
            printf("Enter FTP server port!\n");
            exit(1);
      }

      int server_fd, client_fd;
      struct sockaddr_in address;
      int addrlen = sizeof(address);

      printf("FTP Server\n");

      if((server_fd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
            printf("Socket creation failed!\n");
            exit(1);
      }

      address.sin_family = AF_INET;
      address.sin_addr.s_addr = INADDR_ANY;
      address.sin_port = htons(PORT);
```

```
	if(bind(server_fd, (struct sockaddr*) &address, addrlen) < 0) {
		printf("Socket binding failed!\n");
		exit(1);
	}

	if(listen(server_fd, 5) < 0) {
		printf("Listening failed!\n");
		exit(1);
	}

	while(1) {
		if((client_fd = accept(server_fd, (struct sockaddr*) &address,
(socklen_t*) &addrlen)) < 0) {
			printf("Connection failed!\n");
			exit(1);
		} else {
			printf("Connected to client.\n");
		}

		serverRecv(client_fd);

		close(client_fd);
	}

	close(server_fd);
}
```

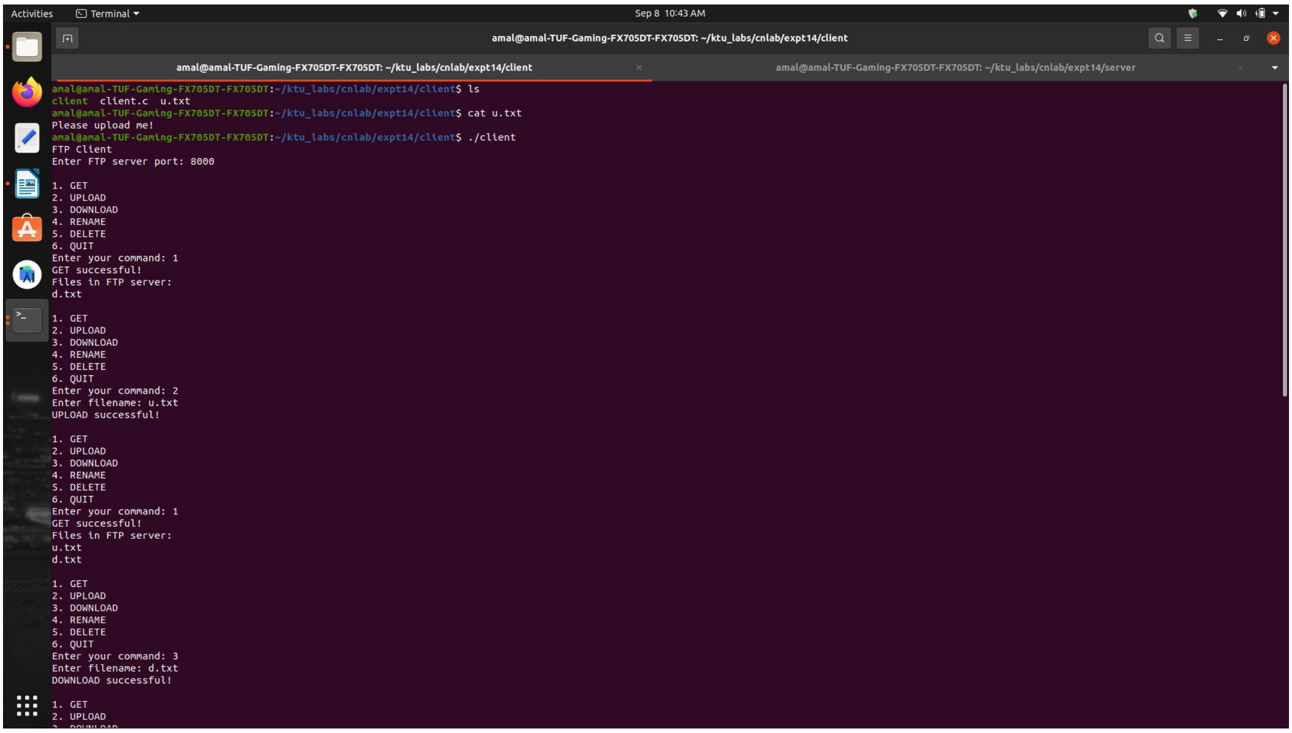
client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <string.h>

void ftp_get(int client_fd) {
	char str[8];
	char str1[1000];

	if(send(client_fd, "GET", 4 * sizeof(char), 0) <= 0) {
		printf("GET failed!\n");
		return;
	}

	if(recv(client_fd, str1, 1000 * sizeof(char), 0) <= 0) {
		printf("GET failed!\n");
		return;
	} else {
		if(recv(client_fd, str, 8 * sizeof(char), 0) <= 0) {
			printf("GET failed!\n");
			return;
		}

		if(!strcmp(str, "200 OK")) {
			printf("GET successful!\nFiles in FTP server:\n%s", str1);
		} else {
			printf("GET failed!\n");
		}
	}
}
```

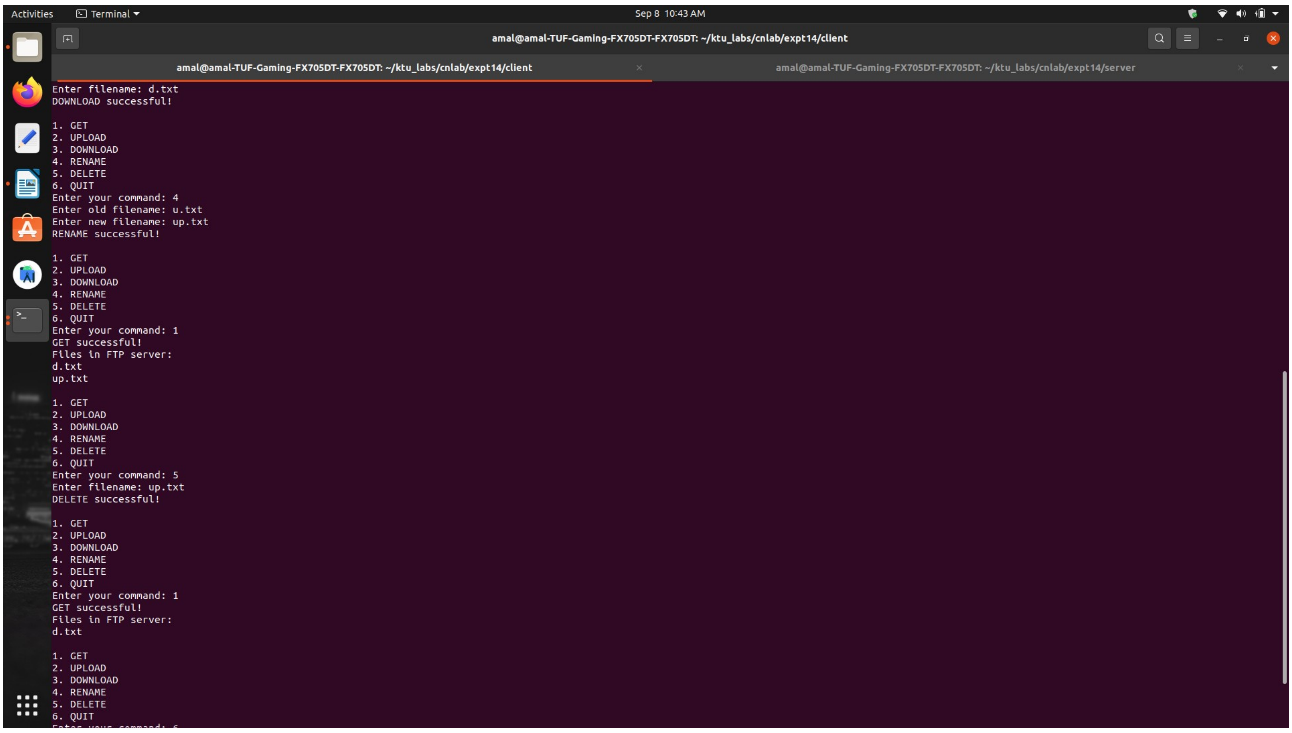
```
void ftp_upload(int client_fd) {
      if(send(client_fd, "UPLOAD", 7 * sizeof(char), 0) <= 0) {
            printf("UPLOAD failed!\n");
            return;
      }

      char filename[100];
      char str[100];
      char str1[1000];

      printf("Enter filename: ");
      scanf("%s", filename);

      FILE* fp = fopen(filename, "r");

      if(fp == NULL) {
            printf("File doesn't exist!\n");
            return;
      }

      if(send(client_fd, filename, 100 * sizeof(char), 0) <= 0) {
            printf("Send failed!\n");
            fclose(fp);
            return;
      } else {
            str1[0] = '\0';

            while(fgets(str, 100, fp) != NULL) {
                  strcat(str1, str);
            }

            if(send(client_fd, str1, 1000 * sizeof(char), 0) <= 0) {
                  printf("UPLOAD failed!\n");
                  fclose(fp);
                  return;
            }

            if(recv(client_fd, str, 7 * sizeof(char), 0) <= 0) {
                  printf("UPLOAD failed!\n");
                  fclose(fp);
                  return;
            }

            if(!strcmp(str, "200 OK")) {
                  printf("UPLOAD successful!\n");
            }

            fclose(fp);
      }
}

void ftp_download(int client_fd) {
      if(send(client_fd, "DOWNLOAD", 9 * sizeof(char), 0) <= 0) {
            printf("Send failed!\n");
            return;
      }

      char filename[100];
      char str[100];
      char str1[1000];
      printf("Enter filename: ");
      scanf("%s", filename);

      if(send(client_fd, filename, 100 * sizeof(char), 0) <= 0) {
        printf("DOWNLOAD failed!\n");
        return;
    } else {
        if(recv(client_fd, str1, 1000 * sizeof(char), 0) <= 0) {
            printf("DOWNLOAD failed!\n");
            return;
        }

        if(recv(client_fd, str, 100 * sizeof(char), 0) <= 0) {
            printf("DOWNLOAD failed!\n");
            return;
        }

        if(!strcmp(str, "200 OK")) {
            FILE* fp = fopen(filename, "w");
            fputs(str1, fp);

            printf("DOWNLOAD successful!\n");

            fclose(fp);
        } else {
            printf("DOWNLOAD failed!\n");
        }
    }
}

void ftp_rename(int client_fd) {
    if(send(client_fd, "RENAME", 7 * sizeof(char), 0) <= 0) {
        printf("RENAME failed!\n");
        return;
    }

    char filename[100];
    char filename1[100];
    char str[8];

    printf("Enter old filename: ");
    scanf("%s", filename);

    printf("Enter new filename: ");
    scanf("%s", filename1);

    if(send(client_fd, filename, 100 * sizeof(char), 0) <= 0) {
        printf("RENAME failed!\n");
        return;
    } else {
        if(send(client_fd, filename1, 100 * sizeof(char), 0) <= 0) {
            printf("RENAME failed!\n");
            return;
        } else {
            if(recv(client_fd, str, 8 * sizeof(char), 0) <= 0) {
                printf("RENAME failed!\n");
                return;
            }

            if(!strcmp(str, "200 OK")) {
                printf("RENAME successful!\n");
            } else {
                printf("RENAME failed!\n");
            }
        }
    }
}
```

```
void ftp_delete(int client_fd) {
      if(send(client_fd, "DELETE", 7 * sizeof(char), 0) <= 0) {
            printf("DELETE failed!\n");
            return;
      }

      char filename[100];
      char str[8];
      printf("Enter filename: ");
      scanf("%s", filename);

      if(send(client_fd, filename, 100 * sizeof(char), 0) <= 0) {
            printf("DELETE failed!\n");
            return;
      } else {
            if(recv(client_fd, str, 8 * sizeof(char), 0) <= 0) {
                  printf("DELETE failed!\n");
                  return;
            }

            if(!strcmp(str, "200 OK")) {
                  printf("DELETE successful!\n");
            } else {
                  printf("DELETE failed!\n");
            }
      }
}

void main() {
      int client_fd, PORT;
      struct sockaddr_in serv_addr;

      printf("FTP Client\n");

      printf("Enter FTP server port: ");
      scanf("%d", &PORT);

      serv_addr.sin_family = AF_INET;
      serv_addr.sin_addr.s_addr = INADDR_ANY;
      serv_addr.sin_port = htons(PORT);

      while(1) {
            client_fd = socket(AF_INET, SOCK_STREAM, 0);

            if(client_fd < 0) {
                  printf("Socket creation failed!\n");
                  exit(1);
            }

            if(connect(client_fd, (struct sockaddr*) &serv_addr, 
sizeof(serv_addr)) < 0) {
                  printf("Connection failed!\n");
                  exit(1);
            }

            int option = 0;
            printf("\n1. GET\n2. UPLOAD\n3. DOWNLOAD\n4. RENAME\n5. DELETE\n6. 
QUIT\nEnter your command: ");
            scanf("%d", &option);
            getchar();

            switch(option) {
                  case 1:
                        ftp_get(client_fd);
                        break;
                  case 2:
                        ftp_upload(client_fd);
                        break;
                  case 3:
                        ftp_download(client_fd);
                        break;
                  case 4:
                        ftp_rename(client_fd);
                        break;
                  case 5:
                        ftp_delete(client_fd);
                        break;
                  case 6:
                        printf("Exit.\n");
                        exit(0);
                  default:
                        printf("Invalid command!\n");
                        break;
            }

            close(client_fd);
      }
}
```

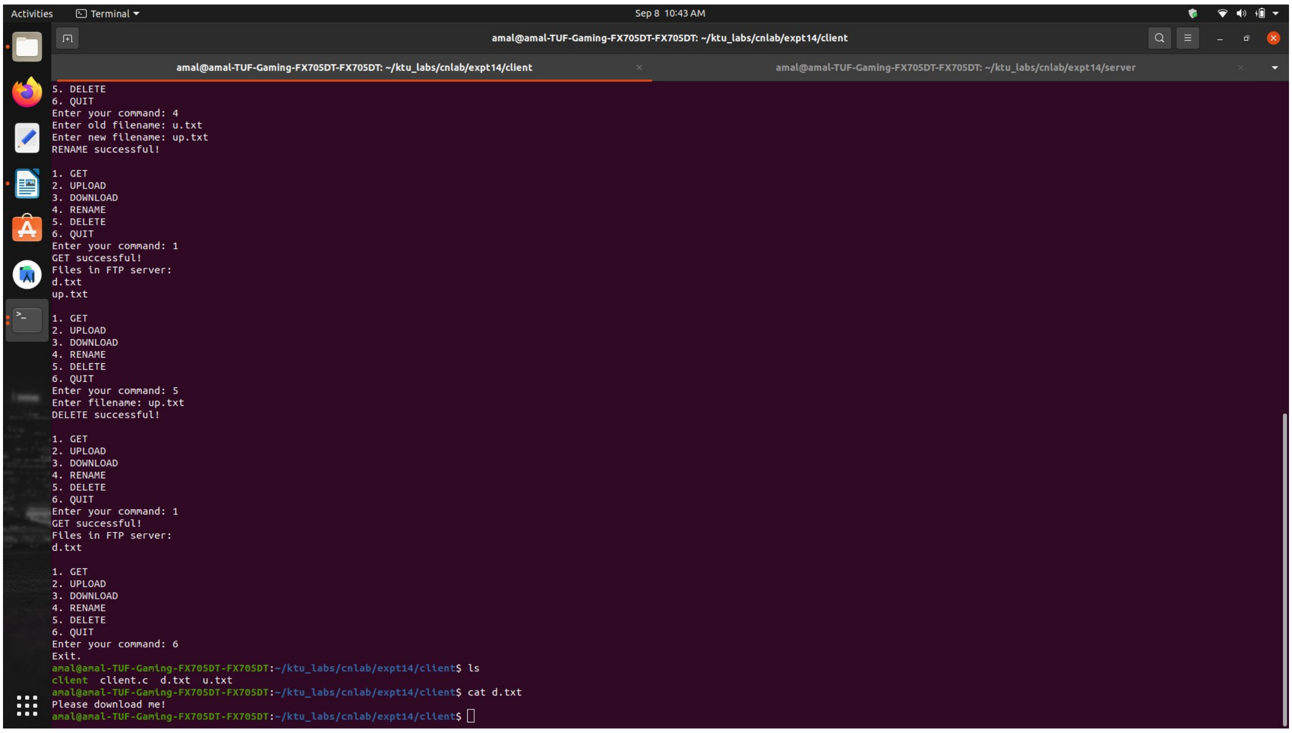
OUTPUT


Activities
Terminal
Sep 8 10:43 AM
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/client
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/client
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/server

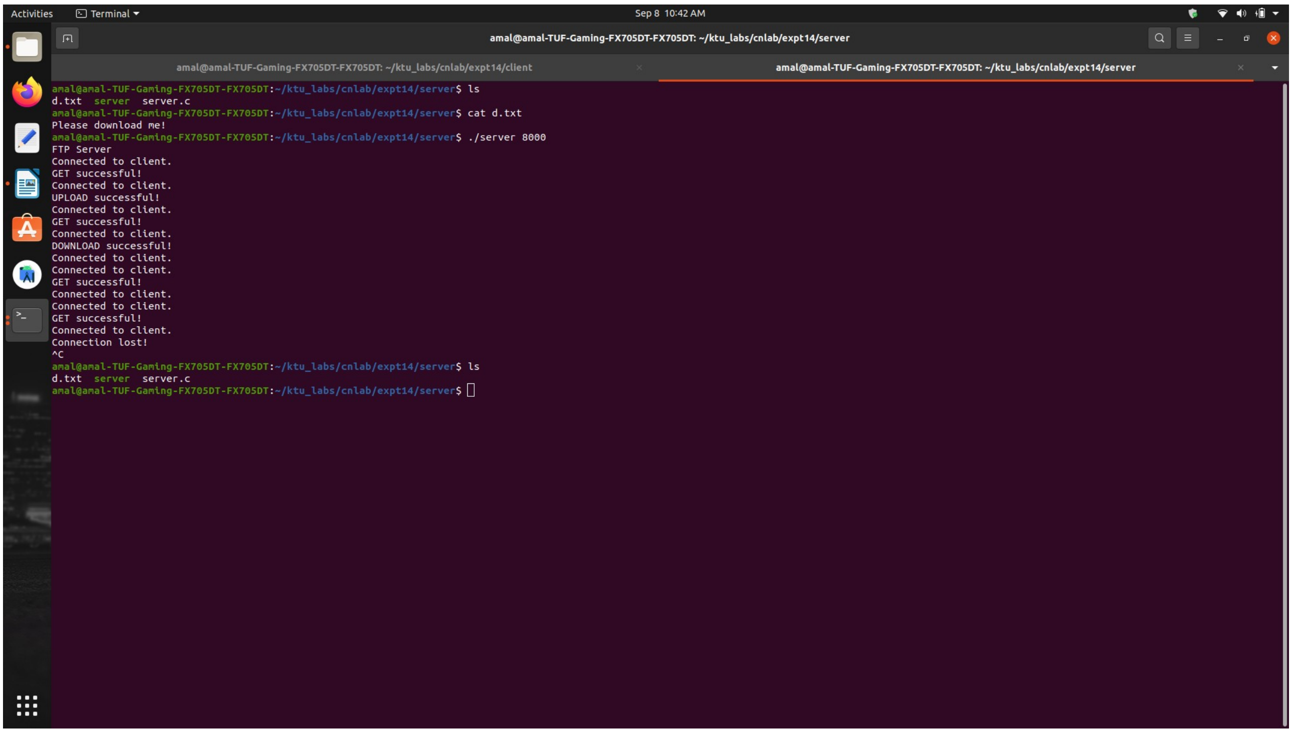
```
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/client$ ls
client  client.c  u.txt
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/client$ cat u.txt
Please upload me!
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/client$ ./client
FTP Client
Enter FTP server port: 8000

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 1
GET successful!
Files in FTP server:
d.txt

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 2
Enter filename: u.txt
UPLOAD successful!

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 1
GET successful!
Files in FTP server:
u.txt
d.txt

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 3
Enter filename: d.txt
DOWNLOAD successful!

1. GET
2. UPLOAD
3. DOWNLOAD
```


Activities
Terminal
Sep 8 10:43 AM
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/client
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/client
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/server

```
Enter filename: d.txt
DOWNLOAD successful!

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 4
Enter old filename: u.txt
Enter new filename: up.txt
RENAME successful!

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 1
GET successful!
Files in FTP server:
d.txt
up.txt

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 5
Enter filename: up.txt
DELETE successful!

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 1
GET successful!
Files in FTP server:
d.txt

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 6
```


Activities
Terminal
Sep 8 10:43 AM
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/client
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/client
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/server

```
5. DELETE
6. QUIT
Enter your command: 4
Enter old filename: u.txt
Enter new filename: up.txt
RENAME successful!

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 1
GET successful!
Files in FTP server:
d.txt
up.txt

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 5
Enter filename: up.txt
DELETE successful!

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 1
GET successful!
Files in FTP server:
d.txt

1. GET
2. UPLOAD
3. DOWNLOAD
4. RENAME
5. DELETE
6. QUIT
Enter your command: 6
Exit.
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/client$ ls
client  client.c  d.txt  u.txt
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/client$ cat d.txt
Please download me!
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/client$ 
```


Activities
Terminal
Sep 8 10:42 AM
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/server
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/client
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt14/server

```
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/server$ ls
d.txt  server  server.c
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/server$ cat d.txt
Please download me!
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/server$ ./server 8000
FTP Server
Connected to client.
GET successful!
Connected to client.
UPLOAD successful!
Connected to client.
GET successful!
Connected to client.
DOWNLOAD successful!
Connected to client.
Connected to client.
GET successful!
Connected to client.
Connected to client.
GET successful!
Connected to client.
Connection lost!
^C
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/server$ ls
d.txt  server  server.c
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt14/server$ 
```