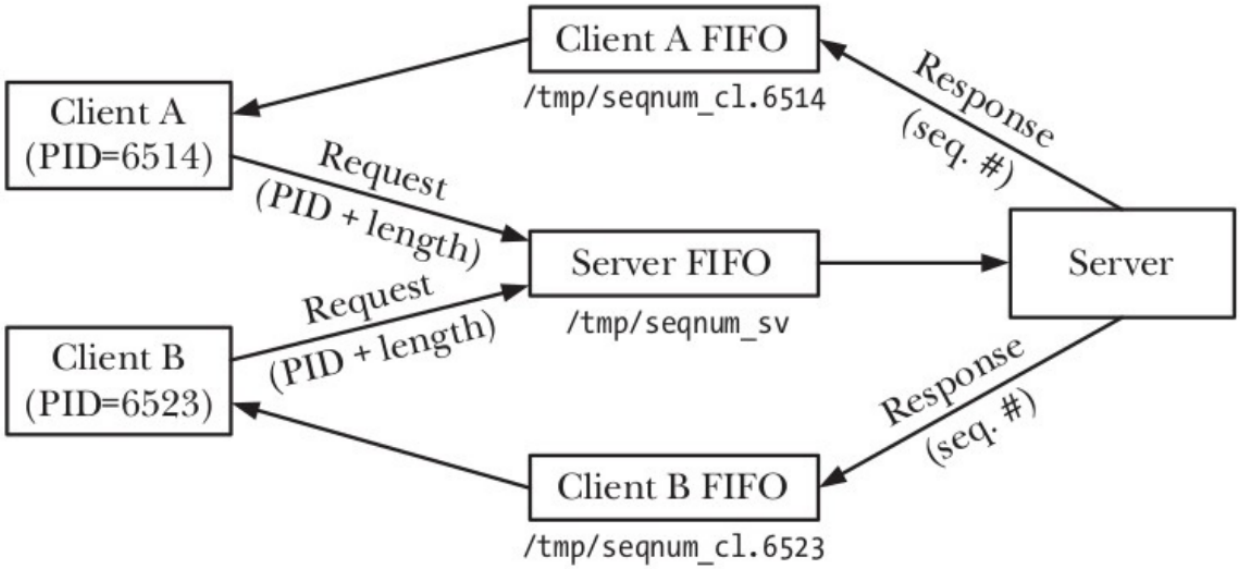
CSE 344
MIDTERM
REPORT

Ömer Faruk SAYAR
171044038

OverAll Design

In this project I used the structure of single-server , multiple-client application:



There is a one server FIFO with named server.PID that accepts the connection requests.

```c
typedef struct connectionReq {
    pid_t client_pid;
    int try_flag;
} connectionReq;
```

My connection requests are consist of client_pid (needed to open client FIFO which is named client.PID) and try_flag for the determine if the request is tryConnect or connect.

Server continuously reads the incoming requests in a while-true loop, and if there are clients their number is less than max_number then server accepts the connection, otherwise adds request to queue if it is try_flag is 0, else rejects the request.

Server Side:

```c
while (1) {
    connectionReq req;
    ssize_t num_read = read(server_fifo_fd, &req, sizeof(connectionReq));
    if (num_read == -1) {
        printf("Error: Failed to read connection request from server FIFO: %s\n", strerror(errno));
        continue;
    }

    else if (num_read == 0) //There is no connection request
        continue;

    if ((*child_num) == max_client_num) {
        printf("Connection request PID %d ...  Queue is FULL\n", req.client_pid);
        if (req.try_flag == 1)
           handle_rejected(req.client_pid);

        else
            enqueue(&connection_queue, req.client_pid);
    }
    else
        handle_client(req.client_pid);
}
```

Client Side:

```c
// create connection request struct
connectionReq req;
req.client_pid = getpid();
if (strcmp(argv[1], "connect") == 0)
    req.try_flag = 0; // client will wait in the queue

else if (strcmp(argv[1], "tryConnect") == 0)
    req.try_flag = 1; // client will not wait in the queue

else {…

char client_fifo[256];
snprintf(client_fifo, sizeof(client_fifo), "/tmp/client.%d", getpid());
if (mkfifo(client_fifo ,0666) == -1) {…

char server_fifo[256];
snprintf(server_fifo, sizeof(server_fifo), "/tmp/server.%d", server_pid);
// send connection request to server
int server_fifo_fd = open(server_fifo, O_WRONLY);
if (server_fifo_fd == -1) {…
if (write(server_fifo_fd, &req, sizeof(connectionReq)) == -1) {…
```

To be able to keep any number of requests, queue was implemented as linked-list. There is also a log file for the server that logs the client requests. To provide any data corruption on the log file I used unnamed semaphore as mutex and kept it on the shared memory.

```c
void log_message(const char *message) {

    time_t now = time(NULL);
    struct tm *tm_info = localtime(&now);
    char timestamp[20];
    strftime(timestamp, 20, "%Y-%m-%d %H:%M:%S", tm_info);
    char log_msg[1024];
    snprintf(log_msg, sizeof(log_msg), "[%s] %s\n", timestamp, message);
    sem_wait(log_mutex);
    write(log_fd, log_msg, strlen(log_msg));
    sem_post(log_mutex);
}
```

After the connection established, client and child-server use the client FIFO respectively. First one side opens the FIFO for writing other side opens for reading then, writing side closes and opens for reading, reading side closes and opens for writing.
Before client sends any commands to server it waits the user input.

```c
while(running)
{
    // open client fifo for writing
    int client_fifo_fd1 = open(client_fifo, O_WRONLY | O_CREAT, 0666);
    if (client_fifo_fd1 == -1) {…

    Request req;
    printf("Enter a command: ");
    char input[BUFSIZE];
    fgets(input, sizeof(input), stdin);
    fflush(stdin);
    input[strlen(input) - 1] = '\0';
    //parse command according to space character
    char *token = strtok(input, " ");
    if (token == NULL){…

    char* command = malloc(strlen(token) + 1);
    strcpy(command, token);
    command[strlen(token)] = '\0';

    if (strcmp(command, "help") == 0)…

    else if(strcmp(command,"list") == 0){…

    else if(strcmp(command,"readF") == 0){…

    else if(strcmp(command, "writeT") == 0)…

    else if(strcmp(command, "quit") == 0)…

    else if(strcmp(command, "killServer") == 0){…

    else if(strcmp(command, "download") == 0){…

    else if(strcmp(command, "upload")== 0){…

    else
        printf("Invalid command!\n");

    free(command);
}
```

For synchronization between the server's child processes I used named semaphores to avoid any race condition and data corruption.

```
char semaphore_name[256];
strcpy(semaphore_name, "/");
strcat(semaphore_name, req.filename);
sem_t* semaphore = sem_open(semaphore_name, O_CREAT, 0644, 1);
sem_wait(semaphore);
transferred = copy_file(fp, destination);
sem_post(semaphore);
sem_close(semaphore);
```

I named the semaphores using file names on the server to be able to sure correct semaphore(mutex) is locked and opened. Before the termination of the server program unlink them using the files names on the server.

```
void close_server(){

    DIR* dir = opendir(working_dir);
    if (dir == NULL) {
        perror("opendir");
        exit(1);
    }
    struct dirent* entry;
    while ((entry = readdir(dir)) != NULL) {
        if (strcmp(entry->d_name, ".") == 0 || strcmp(entry->d_name, "..") == 0) {
            continue;
        }
        char semaphore_name[256];
        strcpy(semaphore_name, "/");
        strcat(semaphore_name, entry->d_name);
        sem_unlink(semaphore_name);
    }

    closedir(dir);
    munmap(&child_num, sizeof(child_num));
    munmap(&client_num, sizeof(client_num));
    sem_destroy(log_mutex);
    close(server_fifo_fd);
    unlink(server_fifo);
    close(log_fd);
}
```

I also used shared memory to keep client number and currently running child number.

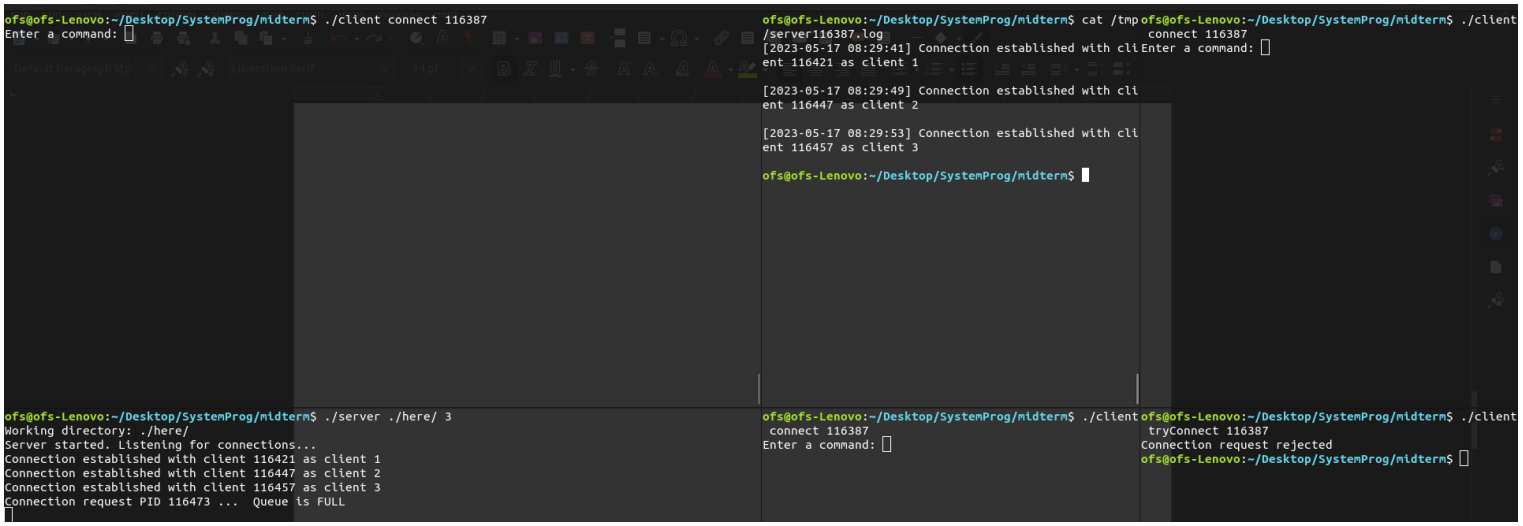
```
void init_server()
{
    struct stat st;
    if (stat(working_dir, &st) == -1) {
        if (mkdir(working_dir, 0777) == -1) {
            perror("mkdir");
            exit(EXIT_FAILURE);
        }
    }
    else…

    pid_t pid = getpid();
    char log_path[256];
    snprintf(log_path, sizeof(log_path), "/tmp/server%d.log", pid);
    log_fd = open(log_path, O_WRONLY | O_APPEND | O_CREAT, 0666);
    if (log_fd == -1) {
        perror("open_log");
        exit(EXIT_FAILURE);
    }

    log_mutex = mmap(NULL, sizeof(*log_mutex), PROT_READ | PROT_WRITE, MAP_SHARED | MAP_ANONYMOUS, -1, 0);
    child_num = mmap(NULL, sizeof(child_num), PROT_READ | PROT_WRITE, MAP_SHARED | MAP_ANONYMOUS, -1, 0);
    (*child_num) = 0;
    client_num = mmap(NULL, sizeof(*client_num), PROT_READ | PROT_WRITE, MAP_SHARED | MAP_ANONYMOUS, -1, 0);
    (*client_num) = 1;
    child_pids = malloc(max_client_num * sizeof(pid_t));
    init_queue(&connection_queue);
    sem_init(log_mutex, 1, 1);

    printf("Server started. Listening for connections...\n");

    snprintf(server_fifo, sizeof(server_fifo), "/tmp/server.%d", pid);
    mkfifo(server_fifo, 0666);
    // open server FIFO for reading
    server_fifo_fd = open(server_fifo, O_RDONLY);
    if (server_fifo_fd == -1) {
        perror("open");
        exit(1);
    }
}
```

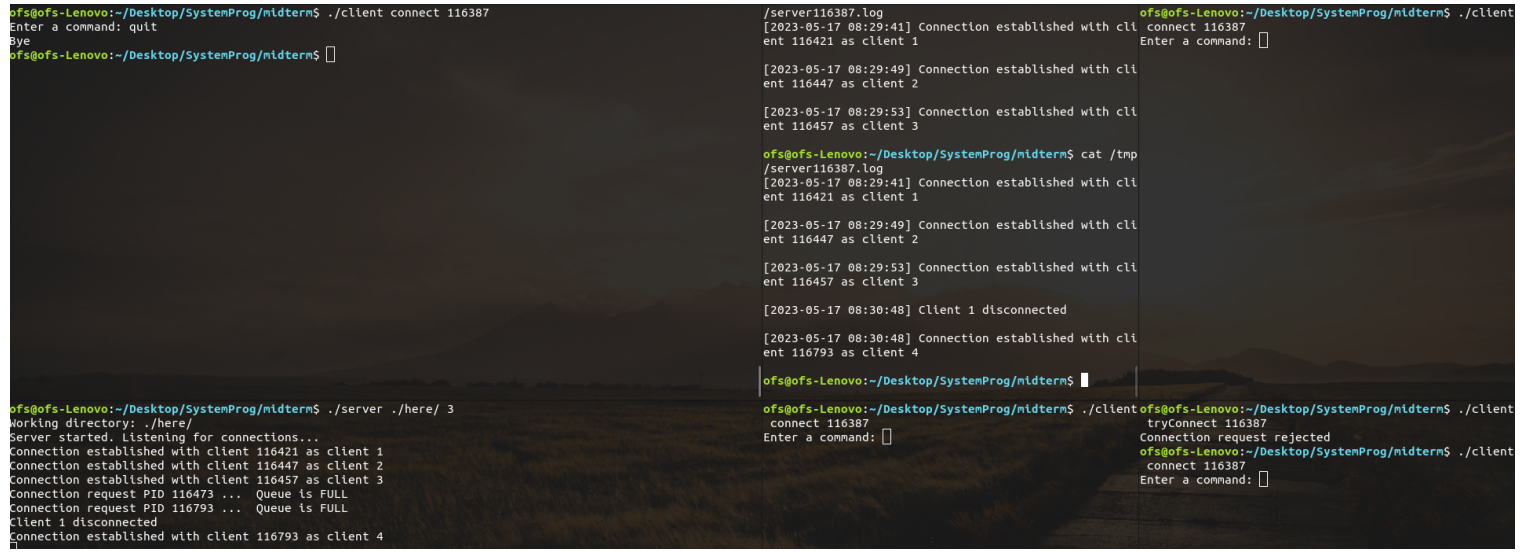
To be able to cleanup child processes when client sends killServer commands to server or any SIGINT, SIGTERM etc. signals arrives the server, Program keeps the array of currently working child pids in an array.

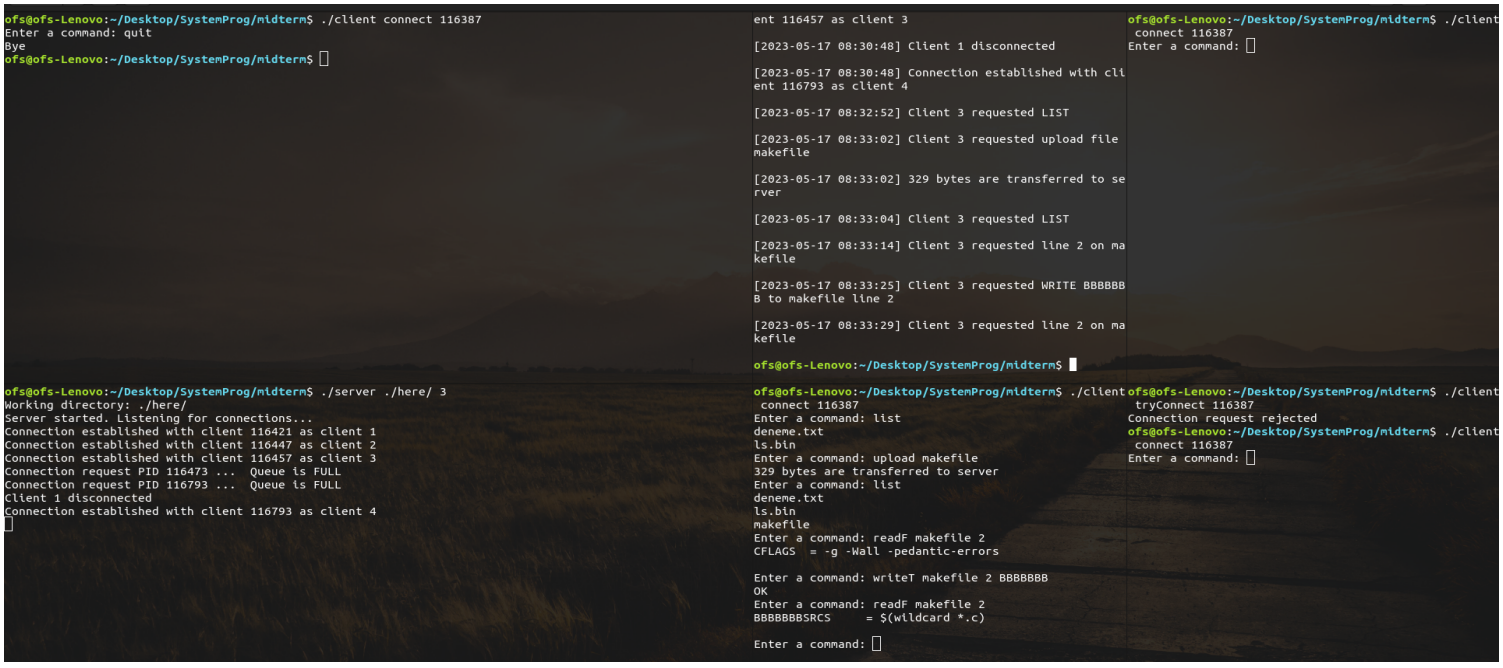
When a child server terminates, program checks the queue if there is any request using SIGCHLD signal handler.

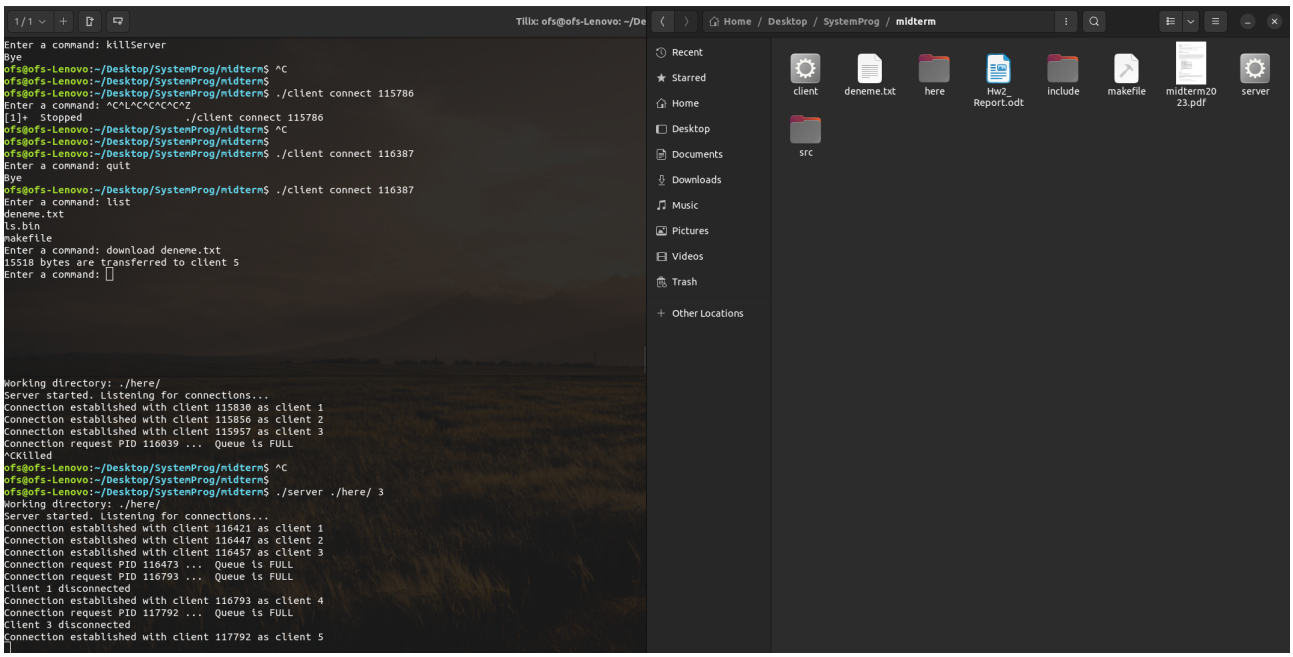
```c
void sigchld_handler(int sig) {
    pid_t pid;
    int status;

    // Wait for any terminated child process
    while ((pid = waitpid(-1, &status, WNOHANG)) > 0)
        (*child_num)--;

    if (!is_queue_empty(&connection_queue) && (*child_num) < max_client_num) {
        pid_t client_pid = dequeue(&connection_queue);
        handle_client(client_pid);
    }
}
```

TEST CASES:


ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command:
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ cat /tmp/server116387.log
[2023-05-17 08:29:41] Connection established with client 116421 as client 1
[2023-05-17 08:29:49] Connection established with client 116447 as client 2
[2023-05-17 08:29:53] Connection established with client 116457 as client 3
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command:
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client tryConnect 116387
Connection request rejected
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./server ./here/ 3
Working directory: ./here/
Server started. Listening for connections...
Connection established with client 116421 as client 1
Connection established with client 116447 as client 2
Connection established with client 116457 as client 3
Connection request PID 116473 ... Queue is FULL


ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command: quit
Bye
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
/server116387.log
[2023-05-17 08:29:41] Connection established with client 116421 as client 1
[2023-05-17 08:29:49] Connection established with client 116447 as client 2
[2023-05-17 08:29:53] Connection established with client 116457 as client 3
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ cat /tmp/server116387.log
[2023-05-17 08:29:41] Connection established with client 116421 as client 1
[2023-05-17 08:29:49] Connection established with client 116447 as client 2
[2023-05-17 08:29:53] Connection established with client 116457 as client 3
[2023-05-17 08:30:48] Client 1 disconnected
[2023-05-17 08:30:48] Connection established with client 116793 as client 4
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command:
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client tryConnect 116387
Connection request rejected
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command:
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./server ./here/ 3
Working directory: ./here/
Server started. Listening for connections...
Connection established with client 116421 as client 1
Connection established with client 116447 as client 2
Connection established with client 116457 as client 3
Connection request PID 116473 ... Queue is FULL
Connection request PID 116793 ... Queue is FULL
Client 1 disconnected
Connection established with client 116793 as client 4


ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command: quit
Bye
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ent 116457 as client 3
[2023-05-17 08:30:48] Client 1 disconnected
[2023-05-17 08:30:48] Connection established with client 116793 as client 4
[2023-05-17 08:32:52] Client 3 requested LIST
[2023-05-17 08:33:02] Client 3 requested upload file makefile
[2023-05-17 08:33:02] 329 bytes are transferred to server
[2023-05-17 08:33:04] Client 3 requested LIST
[2023-05-17 08:33:14] Client 3 requested line 2 on makefile
[2023-05-17 08:33:25] Client 3 requested WRITE BBBBBB to makefile line 2
[2023-05-17 08:33:29] Client 3 requested line 2 on makefile
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command:
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client tryConnect 116387
Connection request rejected
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command:
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./server ./here/ 3
Working directory: ./here/
Server started. Listening for connections...
Connection established with client 116421 as client 1
Connection established with client 116447 as client 2
Connection established with client 116457 as client 3
Connection request PID 116473 ... Queue is FULL
Connection request PID 116793 ... Queue is FULL
Client 1 disconnected
Connection established with client 116793 as client 4
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command: list
deneme.txt
ls.bin
Enter a command: upload makefile
329 bytes are transferred to server
Enter a command: list
deneme.txt
ls.bin
makefile
Enter a command: readF makefile 2
CFLAGS = -g -Wall -pedantic-errors
Enter a command: writeT makefile 2 BBBBBBB
OK
Enter a command: readF makefile 2
BBBBBBBSRCS = $(wildcard *.c)
Enter a command:
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client tryConnect 116387
Connection request rejected
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command:


Enter a command: killServer
Bye
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ^C
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 115786
Enter a command: ^C^L^C^C^C^C^Z
[1]+ Stopped ./client connect 115786
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ^C
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command: quit
Bye
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command: list
deneme.txt
ls.bin
makefile
Enter a command: download deneme.txt
15518 bytes are transferred to client 5
Enter a command:
Working directory: ./here/
Server started. Listening for connections...
Connection established with client 115830 as client 1
Connection established with client 115856 as client 2
Connection established with client 115957 as client 3
Connection request PID 116039 ... Queue is FULL
^Ckilled
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ^C
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./server ./here/ 3
Working directory: ./here/
Server started. Listening for connections...
Connection established with client 116421 as client 1
Connection established with client 116447 as client 2
Connection established with client 116457 as client 3
Connection request PID 116473 ... Queue is FULL
Connection request PID 116793 ... Queue is FULL
Client 1 disconnected
Connection established with client 116793 as client 4
Connection request PID 117792 ... Queue is FULL
Client 3 disconnected
Connection established with client 117792 as client 5
Home / Desktop / SystemProg / midterm
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
client
deneme.txt
here
Hw2_ Report.odt
include
makefile
midterm2023.pdf
server
src


Enter a command: quit
Bye
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command: list
deneme.txt
ls.bin
makefile
Enter a command: download deneme.txt
15518 bytes are transferred to client 5
Enter a command: killServer
Bye
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ^C
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./server ./here/ 3
Working directory: ./here/
Server started. Listening for connections...
Connection established with client 116421 as client 1
Connection established with client 116447 as client 2
Connection established with client 116457 as client 3
Connection request PID 116473 ... Queue is FULL
Connection request PID 116793 ... Queue is FULL
Client 1 disconnected
Connection established with client 116793 as client 4
Connection request PID 117792 ... Queue is FULL
Client 3 disconnected
Connection established with client 117792 as client 5
Killed
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ith client 117792 as client 5
[2023-05-17 08:35:09] Client 5 requested LIST
[2023-05-17 08:35:18] Client 5 requested download file deneme.txt
[2023-05-17 08:35:18] 15518 bytes are transferred to client 5
[2023-05-17 08:36:57] Client 5 requested KILL
[2023-05-17 08:36:57] Server shutting down
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
Enter a command: list
deneme.txt
ls.bin
makefile
Enter a command: readF makefile 2
CFLAGS = -g -Wall -pedantic-errors
Enter a command: writeT makefile 2 BBBBBBB
OK
Enter a command: readF makefile 2
BBBBBBBSRCS = $(wildcard *.c)
Enter a command: quit
Bye
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
novo:~/Desktop/SystemProg/midtofs@ofs-Lenovo:~/Desktop/Sysofs@ofs-Lenovo:~/Desktop/Sofs@ofs-Lenovo:~/Desktop/Systofs@ofs-Lenovo:~/Desktop/Systemofs@ofs-Lenovo:~/Desktofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ exit^C
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 115786
Enter a command: quit
Bye
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command: list
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client tryConnect 116387
Connection request rejected
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$ ./client connect 116387
Enter a command: list
ofs@ofs-Lenovo:~/Desktop/SystemProg/midterm$