GEBZE TECHNİCAL UNIVERSITY

CSE344

System Programming

Midterm Report

1. How to Run?

To initiate the server, execute "./neHosServer <directory_name> <max_clients>" in the terminal. For example, "./neHosServer here 5" starts the server with the specified directory name "here" and a maximum of 5 clients. For the client, execute "./neHosClient <Connect/TryConnect> <pid_of_server>".

2. Server Part

- **Server Startup:**
  - Sets up signal handler for SIGINT.
  - Validates and stores command-line arguments (server directory and max clients).
  - Creates the server directory (if it doesn't exist).
  - Changes the working directory to the server directory.
  - Creates a log file.
  - Creates the server named FIFO for client communication.
  - Opens the server FIFO for reading with non-blocking mode.
  - Opens the server FIFO for writing (dummy operation to prevent EOF).
  - Prints server startup message with PID.

- **Main Loop:**

  - Continuously reads client requests from the server FIFO.
  - If the request is `KILL_SERVER`, initiates shutdown (removes server FIFO and exits).
  - Checks if there are waiting clients and connected clients.
    - If so, moves a client from the waiting queue to the connected queue and notifies the client.
  - Forks a child process to handle each client request.

- **Child Process:**

  - Handles the received client request using the `handle_request` function.
  - `handle_request` function:
    - Opens the client FIFO for writing.
    - Processes the request based on operation type and connection type:
      - `CONNECT`: Adds the client to the connected list if there's space, otherwise adds to the waiting list and sends a waiting message.
      - `TRY_CONNECT`: Similar to `CONNECT` but doesn't wait in a queue and directly rejects if full.
      - Supported operations (`HELP`, `LIST`, `READ_FILE`, `WRITE_FILE`, `UPLOAD`, `DOWNLOAD`, `ARCHIVE_SERVER`, `QUIT`):
        - Calls the appropriate function (e.g., `read_file`, `write_to_file`, `upload_file`) to perform the operation.
        - Sends a response message to the client FIFO indicating success/failure and any relevant data.
      - `QUIT`: Kills the client process.
  - The child process exits after handling the request.

- **Parent Process:**

  - The parent process continues listening for new client requests in the main loop.

3. Client Part

- **Initialization:**

  - Sets up the signal handler for SIGINT.
  - Validates command-line arguments (connection type - "Connect" or "tryConnect" - and server PID).
  - Creates a unique client named FIFO path based on the client process ID.
  - Creates the client FIFO for communication with the server.
  - Parses the connection type and converts the server PID to an integer.
  - Constructs the server FIFO path based on the server PID.
  - Opens the server FIFO for writing only.
  - Creates a `connect_request` struct containing the client PID, connection type, and initial "NONE" operation type.
  - Sends the `connect_request` to the server through the server FIFO.
  - Opens the client FIFO for reading responses from the server.

- **Connection and Command Loop:**

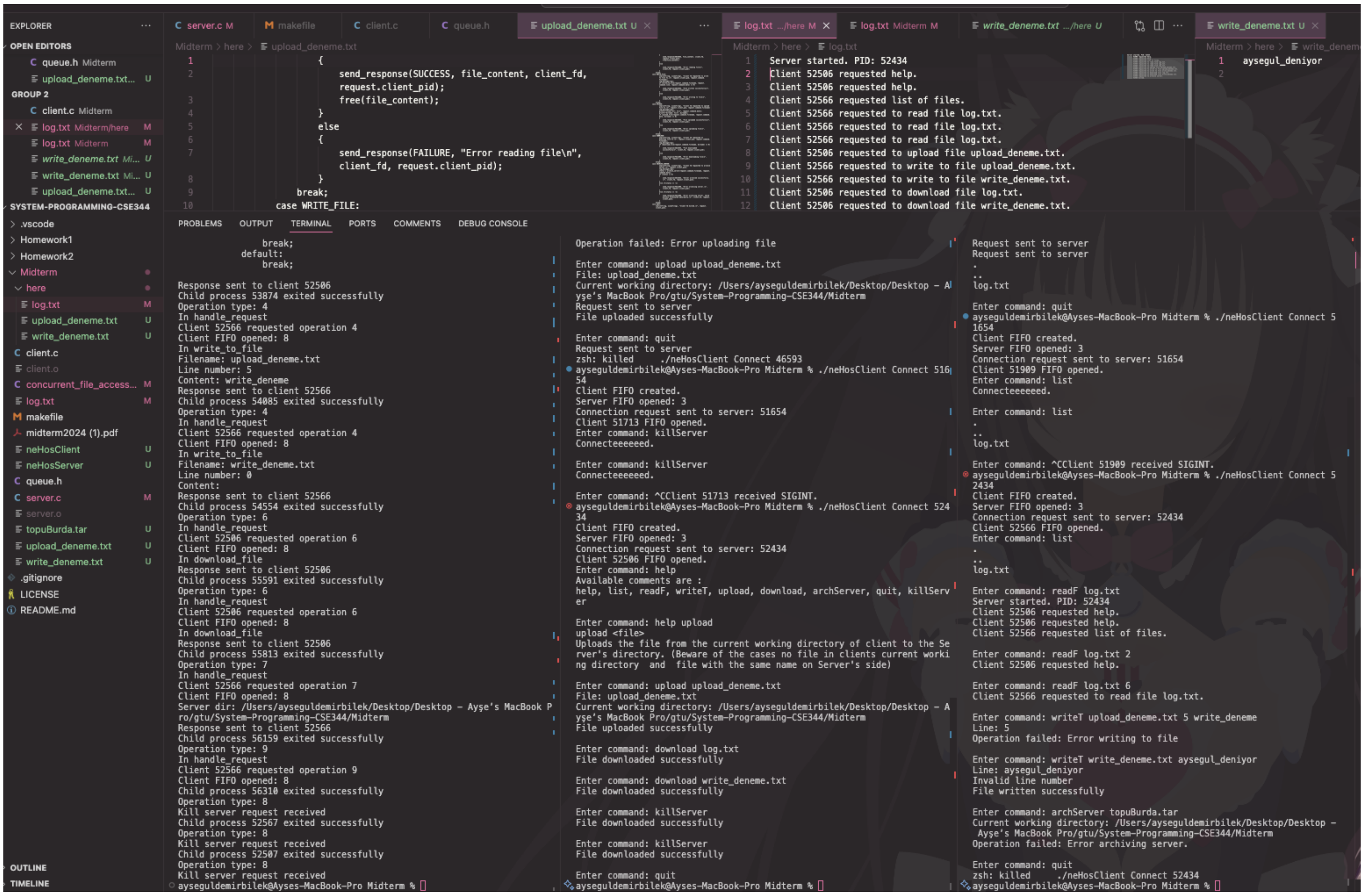
  - Reads a response from the server FIFO.
  - Handles the response status:
    - `SUCCESS`: Enters a loop for sending commands and receiving responses.
    - `FAILURE`: Prints an error message with the reason provided by the server and exits.
    - `WAIT` (for "tryConnect"): Prints the waiting message received from the server and exits (client couldn't connect immediately).
  - Inside the loop:
    - Prompts the user for a command.
    - Reads the command from standard input.
    - Removes the newline character from the command string.
    - Parses the command using the `parse_command` function (defined in `concurrent_file_access_system.h`).
    - If the command is invalid, displays an error message and continues waiting for a valid command.
    - Otherwise, sends the parsed command (including operation type and additional arguments) to the server using the `send_request` function.
    - Reads a response from the server FIFO.
    - Displays the response based on its status:
      - `SUCCESS`: Prints the response body (usually the result of the operation).
      - `FAILURE`: Prints an error message with the reason provided by the server.
      - `WAIT`: Prints a message indicating the server is busy (applicable for certain operations).

- **Cleanup:**

  - Closes the server and client FIFOs.

- Removes the client FIFO.
- Exits the program with success (`EXIT_SUCCESS`).

4. Output



```
                {
                    send_response(SUCCESS, file_content, client_fd,
                    request.client_pid);
                    free(file_content);
                }
                else
                {
                    send_response(FAILURE, "Error reading file\n",
                    client_fd, request.client_pid);
                }
            break;
        case WRITE_FILE:
```

```
Server started. PID: 52434
Client 52506 requested help.
Client 52506 requested help.
Client 52566 requested list of files.
Client 52566 requested to read file log.txt.
Client 52566 requested to read file log.txt.
Client 52566 requested to read file log.txt.
Client 52506 requested to upload file upload_deneme.txt.
Client 52566 requested to write to file upload_deneme.txt.
Client 52566 requested to write to file write_deneme.txt.
Client 52506 requested to download file log.txt.
Client 52506 requested to download file write_deneme.txt.
```

```
aysegul_deniyor
```

```
            break;
        default:
            break;

Response sent to client 52506
Child process 53874 exited successfully
Operation type: 4
In handle_request
Client 52566 requested operation 4
Client FIFO opened: 8
In write_to_file
Filename: upload_deneme.txt
Line number: 5
Content: write_deneme
Response sent to client 52566
Child process 54085 exited successfully
Operation type: 4
In handle_request
Client 52566 requested operation 4
Client FIFO opened: 8
In write_to_file
Filename: write_deneme.txt
Line number: 0
Content:
Response sent to client 52566
Child process 54554 exited successfully
Operation type: 6
In handle_request
Client 52506 requested operation 6
Client FIFO opened: 8
In download_file
Response sent to client 52506
Child process 55591 exited successfully
Operation type: 6
In handle_request
Client 52506 requested operation 6
Client FIFO opened: 8
In download_file
Response sent to client 52506
Child process 55813 exited successfully
Operation type: 7
In handle_request
Client 52566 requested operation 7
Client FIFO opened: 8
Server dir: /Users/ayseguldemirbilek/Desktop/Desktop – Ayşe’s MacBook P
ro/gtu/System-Programming-CSE344/Midterm
Response sent to client 52566
Child process 56159 exited successfully
Operation type: 9
In handle_request
Client 52566 requested operation 9
Client FIFO opened: 8
Child process 56310 exited successfully
Operation type: 8
Kill server request received
Child process 52567 exited successfully
Operation type: 8
Kill server request received
Child process 52507 exited successfully
Operation type: 8
Kill server request received
ayseguldemirbilek@Ayses-MacBook-Pro Midterm % 
```

```
Operation failed: Error uploading file

Enter command: upload upload_deneme.txt
File: upload_deneme.txt
Current working directory: /Users/ayseguldemirbilek/Desktop/Desktop – A
yşe’s MacBook Pro/gtu/System-Programming-CSE344/Midterm
Request sent to server
File uploaded successfully

Enter command: quit
Request sent to server
zsh: killed     ./neHosClient Connect 46593
ayseguldemirbilek@Ayses-MacBook-Pro Midterm % ./neHosClient Connect 516
54
Client FIFO created.
Server FIFO opened: 3
Connection request sent to server: 51654
Client 51713 FIFO opened.
Enter command: killServer
Connecteeeeeed.

Enter command: killServer
Connecteeeeeed.

Enter command: ^CClient 51713 received SIGINT.
ayseguldemirbilek@Ayses-MacBook-Pro Midterm % ./neHosClient Connect 524
34
Client FIFO created.
Server FIFO opened: 3
Connection request sent to server: 52434
Client 52506 FIFO opened.
Enter command: help
Available comments are :
help, list, readF, writeT, upload, download, archServer, quit, killServ
er

Enter command: help upload
upload <file>
Uploads the file from the current working directory of client to the Se
rver's directory. (Beware of the cases no file in clients current worki
ng directory  and  file with the same name on Server's side)

Enter command: upload upload_deneme.txt
File: upload_deneme.txt
Current working directory: /Users/ayseguldemirbilek/Desktop/Desktop – A
yşe’s MacBook Pro/gtu/System-Programming-CSE344/Midterm
File uploaded successfully

Enter command: download log.txt
File downloaded successfully

Enter command: download write_deneme.txt
File downloaded successfully

Enter command: killServer
File downloaded successfully

Enter command: killServer
File downloaded successfully

Enter command: quit
ayseguldemirbilek@Ayses-MacBook-Pro Midterm % 
```

```
Request sent to server
Request sent to server
.
..
log.txt

Enter command: quit
ayseguldemirbilek@Ayses-MacBook-Pro Midterm % ./neHosClient Connect 5
1654
Client FIFO created.
Server FIFO opened: 3
Connection request sent to server: 51654
Client 51909 FIFO opened.
Enter command: list
Connecteeeeeed.

Enter command: list
.
..
log.txt

Enter command: ^CClient 51909 received SIGINT.
ayseguldemirbilek@Ayses-MacBook-Pro Midterm % ./neHosClient Connect 5
2434
Client FIFO created.
Server FIFO opened: 3
Connection request sent to server: 52434
Client 52566 FIFO opened.
Enter command: list
.
..
log.txt

Enter command: readF log.txt
Server started. PID: 52434
Client 52506 requested help.
Client 52506 requested help.
Client 52566 requested list of files.

Enter command: readF log.txt 2
Client 52506 requested help.

Enter command: readF log.txt 6
Client 52566 requested to read file log.txt.

Enter command: writeT upload_deneme.txt 5 write_deneme
Line: 5
Operation failed: Error writing to file

Enter command: writeT write_deneme.txt aysegul_deniyor
Line: aysegul_deniyor
Invalid line number
File written successfully

Enter command: archServer topuBurda.tar
Current working directory: /Users/ayseguldemirbilek/Desktop/Desktop –
 Ayşe’s MacBook Pro/gtu/System-Programming-CSE344/Midterm
Operation failed: Error archiving server.

Enter command: quit
zsh: killed     ./neHosClient Connect 52434
ayseguldemirbilek@Ayses-MacBook-Pro Midterm % 
```