CSE344 – HOMEWORK 4

Yağız Hakkı Aydın
1901042612

Mainly this function starts with starting the server,when server started;client stars with server id.When it happens,server is already created a server fifo to get requests.When client first connected,sends a specific message to that server fifo and server fifo forks itself.Then child process responses to that client and tells new server fifo to get requests it.So,when program starts ; between child process of server and client;there is two fifos for request-response communication.At the start of the server program;user enters num of max threads th the server and then for every request;it’s handled at the available thread from the thread pool.

Function explanations for server side:

1. `handleCtrlC`: This function is a signal handler for the Ctrl+C signal. It is triggered when the user presses Ctrl+C, and it performs cleanup tasks before exiting the program.

2. `main`: The main function of the program. It performs the following tasks:
   - Checks the server program arguments for correctness.
   - Sets up the server's working directory.
   - Opens the log file.
   - Initializes threads for handling client requests.
   - Creates and opens the server FIFO (named pipe) for communication with clients.
   - Enters a loop to continuously read requests from the server FIFO and handle them.

3. `checkServerProgramArguments`: This function checks the validity of the server program arguments. It verifies the number of arguments and the format of the arguments.

4. `continueRunInDirectory`: This function continues running the server in the specified directory. It creates the necessary directories in the given path if they don't exist and changes the current working directory to the specified path.

5. `handleRequest`: This function is executed in a separate thread for each client request. It handles different types of requests received from clients, such as "list," "download," "upload," etc. It performs the required actions based on the request type.

6. `copyFile`: This function copies a file from the source directory to the destination directory. It takes the file name, source directory, and destination directory as input parameters.

7. `readLine`: This function reads a specific line from a file. It takes the file name and the line number as input parameters. If the line number is provided, it sends only that line to the client; otherwise, it sends the entire file.

8. `addClient`: This function is called when a new client connects to the server. It creates a new child process to handle the client's requests. It also generates a new FIFO for communication with the client and sends the necessary information to the client for establishing communication.

9. `openLogFile`: This function opens the log file for writing server activities and client requests. It creates a log file with the server's process ID as the file name.

10. `sendListOfFilesToClient`: This function sends the list of files in the server's working directory to the client. It reads the directory contents, concatenates the file names into a string, and writes it to the client FIFO.

These are just brief descriptions of the functions. Each function performs specific tasks to ensure the proper functioning of the server and handling of client requests.

Function explanations for server side:

1. `generateClientFifo`: This function generates a unique FIFO (named pipe) for the client using its process ID. It creates a FIFO file with the name constructed from the process ID.

2. `openClientFifo`: This function opens the client's FIFO for communication with the server. It uses the FIFO file created by `generateClientFifo` and opens it for both reading and writing.

3. `openServerFifo`: This function opens the server's FIFO for communication with the client. It takes the server's process ID as a parameter and constructs the server FIFO file name based on it. It opens the server FIFO for writing.

4. `writeToServerFifo`: This function writes the content of the `CLIENT_TO_SERVER_BUFFER` to the server's FIFO. It is used to send requests from the client to the server.

5. `handleClientComment`: This function handles the user's comment or command. It tokenizes the comment using `tokenizeGivenCommand` function and checks the command type. It performs specific actions based on the command, such as printing help messages, sending requests to the server, or exiting the client.

6. `printHelpComment`: This function prints help messages for different commands. It takes a comment parameter, which specifies the command for which the help message should be printed. It displays the appropriate help message based on the given comment.

7. `connectServer`: This function sends a request to the server to establish a connection between the client and the server. It constructs a request message using the client's process ID and working

directory, and sends it to the server through the `CLIENT_TO_SERVER_BUFFER`. The server will then fork itself and send a specific response to make the client connect to the server's child process.

These functions are responsible for handling client-side operations such as setting up communication with the server, sending requests, handling user commands, and establishing the initial connection with the server.

Please note that the explanations provided are based on the code's structure and function names, as the code itself lacks detailed comments.

Tests:

mid : biboClient — Konsole

File Edit View Bookmarks Plugins Settings Help

```
yagiz@yagiz-System-Product-Name:~/Desktop/mid$ ./biboClient connect 9783

Enter comment : readF test.c 2
#include <fcntl.h>


Enter comment : quit
yagiz@yagiz-System-Product-Name:~/Desktop/mid$ ./biboClient connect 9783

Enter comment :
```

mid : biboClient — Konsole <2>

File Edit View Bookmarks Plugins Settings Help

```
yagiz@yagiz-System-Product-Name:~/Desktop/mid$ ./biboClient connect 9783

Enter comment : list
3232.log
6207.log
3046.log
3373.log
test.c
2419.log
5798.log
3393.log
7721.log
6336.log
1931.log
3203.log
7524.log
2901.log
6486.log
7631.log
2083.log
6212.log
```

serv : biboServer — Konsole

File Edit View Bookmarks Plugins Settings Help

```
yagiz@yagiz-System-Product-Name:~/Desktop/mid$ ./biboServer /home/yagiz/Desktop/serv/ 10
Server started PID 9783
Client PID 9784 connected as "client01" with fifo -> 5 to pid 9785 server
Client PID 9786 connected as "client01" with fifo -> 5 to pid 9787 server
Client PID 9792 connected as "client01" with fifo -> 5 to pid 9793 server
```