Course COMP-8567
Project
Winter 2023
Due Date: Apr/15/2023
100 Marks

- The project work can be carried out alone or in teams of two students.
- Only students from the same section can form a team.
- Along with the file submission, the working of the project must be demonstrated during the scheduled slot (TBA)

Introduction

In this client-server project, a client can request a file or a set of files from the server. The server searches for the file/s in its file directory rooted at its ~ and returns the tar.gz of the file/files requested to the client (or an appropriate message otherwise). Multiple clients can connect to the server from different machines and can request file/s as per the commands listed in section 2

- **The server, the mirror and the client processes** must run on different machines and must communicate using sockets only.

Section 1 (Server)

- The **server** and an identical copy of the server called the **mirror** [see section 3] must both run before any of the client (s) run and both of them must wait for request/s from client/s
- Upon receiving a connection request from a client, the server forks a child process that services the client request exclusively in a function called processclient() and (the server) returns to listening to requests from other clients.
  - The processclient() function enters an infinite loop waiting for the client to send a command
  - Upon the receipt of a command from the client, processclient() performs the action required to process the command as per the requirements listed in section 2 and returns the result to the client
- Upon the receipt of **quit** from the client, processclient() exits.
- **Note:** for each client request, the server must fork a separate process with the processclient() function to service the request and then go back to listening to requests from other clients

Section 2 (Client)

The client process runs an infinite loop waiting for the user to enter one of the commands.

Note: The commands **are not** Linux commands and are defined(in this project) to denote the action to be performed by the server.

Once the command is entered, the client verifies the **syntax of the command** and if it is okay, sends the command to the server, else it prints an appropriate error message.

List of Client Commands:

- **findfile** *filename*
  - If the file *filename* is found in its file directory tree rooted at ~, the server must return **the filename, size(in bytes), and date created** to the client and the client prints the received information on its terminal.
    - Note: if the file with the same name exists in multiple folders in the directory tree rooted at ~, the server sends information pertaining to the first successful search/match of *filename*
    - Else the client prints "File not found"
  - **Ex: C$ findfile sample.txt**

- **sgetfiles** *size1 size2* <-u>
  - The server must return to the client **temp.tar.gz** that contains all the files in the directory tree rooted at its ~ whose file-size in bytes is >=*size1* and <=*size2*
    - *size1 < = size2 (size1>= 0 and size2>=0)*
  - -u unzip temp.tar.gz in the pwd of the client
  - **Ex: C$ sgetfiles 1240 12450 -u**

- **dgetfiles** *date1 date2* <-u>
  - The server must return to the client temp.tar.gz that contains all the files in the directory tree rooted at ~ whose date of creation is >=*date1* and <=*date2* *(date1<=date2)*
  - -u unzip temp.tar.gz in the pwd of the client
  - **Ex: C$ dgetfiles 2023-01-16 2023-03-04 -u**

- **getfiles** *file1 file2 file3 file4 file5 file6* <-u >
  - The server must search the files (file 1 ..up to file6) in its directory tree rooted at ~ and return temp.tar.gz that contains at least one (or more of the listed files) if they are present

  - If none of the files are present, the server sends “No file found” to the client (which is then printed on the client terminal by the client)
  - -u   unzip temp.tar.gz in the pwd of the client
  - **Ex: C$ getfiles new.txt ex1.c ex4.pdf**

- **gettargz** <extension list> <-u> //up to 6 different file types
  - the server must return temp.tar.gz that contains all the files in its directory tree rooted at ~ belonging to the file type/s listed in the extension list, else the server sends the message “No file found” to the client (which is printed on the client terminal by the client)
  - -u   unzip temp.tar.gz in the pwd of client
  - The extension list **must have at least one file type** and can have up to six different file types
  - **Ex: C$ gettargz c txt pdf**

- **quit**
  - The command is transferred to the server and the client process is terminated

Note:

- It is the responsibility of the client process to **verify** the syntax of the command entered by the user (as per the rules in Section 3) before processing it.
  - Appropriate messages must be printed when the syntax of the command is incorrect.
- It is the responsibility of the client process to unzip the tar files whenever the option is specified.

Section 3 Alternating Between the Server and the Mirror

- The server and the mirror (the server’s copy possibly with a few additions/changes) are to run on two different machines/terminals.
- The first 4 client connections are to be handled by the server.
- The next 4 client connections are to be handled by the mirror.
- The remaining client connections are to be handled by the server and the mirror in an alternating manner- (ex: connection 9 is to be handled by the server, connection 10 by the mirror, and so on)

Submission:

You are required to submit three files with **adequate and pertinent comments** briefly explaining/describing various parts of the programs.

- server.c
- mirror.c
- client.c