GTU Department of Computer Engineering
CSE 344 – Spring 2023
Final Project Report

Hasan Mutlu
1801042673

General Workflow

The general workflow of the client-server program is as follows:

- ➔ Server opens/creates the specified directory.
- ➔ Server creates a pool of threads and they go on to wait.
- ➔ Server opens the socket with specified port number.
- ➔ Clients start and open/create specified directory.
- ➔ Clients connect to server via socket.
- ➔ When connection is established, first synchronization is made from server directory to the client directory.
- ➔ Then, the server thread and client exchange the last modification time of their directories and the files in it.
- ➔ If there is a change in at least one side, the latest modified side's directory is syncronized to the others.
- ➔ If there is no change, server thread and client sleep for 1 seconds before checking each others latest modification time again.

Interprocess Communication: Sockets

For the interprocess communication, **stream sockets** are used. The socket is opened in **AF_INET** domain. Its address is determined with **INADDR_ANY** flag.

The server and the client communicates with a determined path. So, all the blocked recv() calls are eventually unblocked. Synchronization is established by following this strict path throughout the program run.

Detecting the Changes

Latest modification times in directories are detected using the get_latest_modification_time() function. This function recursively checks all files and subdirectories, and returns the latest modification time among them.

After each file synchronization, the server and the client send the latest modification time of their directories to each other. Other side's and this side's value are saved in the initial synchronization. Then, during checking, in each iteration, this value is compared with the new modification time sent by the other side or detected from the current directory as follows:

- ➔ my_difference = new_mod - latest_mod;
- ➔ cli_difference = otherside_new_mod - otherside _latest_mod;

If both of their differences, from last saved time to last checking time, it means there is no change. However, if one side is positive, it means that its directory is changed.

When such change is detected, send_directory() and receive_directory() functions are called from each side. The changed files are sent and received using send_file() and receive_file() functions. The lately modified side sends its directory contents to the older side.

Then, the changed files are detected using compareDirectories() function to print information to the standart output and logfile.

Thread Pool Synchronization

The thread pool is synchronized using condition variables and mutexes. When a new connection is established, client sockets are assigned to each thread's index in the client_sockets list. Threads check the value of their client_socket value. If it is 0, they are blocked until main thread assigns a new client to the list.

When all threads are assigned to a client, the main thread is blocked until one of the threads is done with its client. During this time, new clients that try to connect wait for connection. When the main thread is unblocked, it assigns the next client to an available worker thread.

By this way, an efficient synchronization is established and no busy waiting is used.

Termination: Signal Handling

Only way of termination in both the server and the client side is by sending SIGINT signal. When server receives SIGINT signal, it closes the client sockets, free its resources and terminate. The clients's communication process stops and they eventually figure out that the server is disconnected, when they return to their main loop.

Similarly, when client receives SIGINT, it closes the server socket, frees its resources and terminate. The server thread eventually figures out that the client is disconnected in its main loop and marks itself as free.

Test Scenario and Example Outputs

The program is tested with different scenarios. In the example outputs below, to provide clean and readable report, the test scenario consists of these 5 primary cases, using 3 as poolsize:

- ➔ Connecting a new client on queue after current client disconnects
- ➔ Creating file on client side
- ➔ Updating file on client side
- ➔ Deleting file from the client side.
- ➔ Updating file from the server side.

Similar output to logfile is also printed to standard output, without the timestamps

Server log:

```
1   [Fri Jun 16 22:03:32 2023] SERVER: Directory dir is opened.
2   [Fri Jun 16 22:03:32 2023] SERVER: Server started.
3   [Fri Jun 16 22:03:35 2023] Worker0: Client1 is connected to server.
4   [Fri Jun 16 22:03:35 2023] Worker0: First synchronization is made to client1.
5   [Fri Jun 16 22:03:37 2023] Worker1: Client2 is connected to server.
6   [Fri Jun 16 22:03:37 2023] Worker1: First synchronization is made to client2.
7   [Fri Jun 16 22:03:39 2023] Worker2: Client3 is connected to server.
8   [Fri Jun 16 22:03:39 2023] Worker2: First synchronization is made to client3.
9   [Fri Jun 16 22:03:44 2023] Worker0: Client1 disconnected
10  [Fri Jun 16 22:03:44 2023] Worker0: Client4 is connected to server.
11  [Fri Jun 16 22:03:45 2023] Worker0: First synchronization is made to client4.
12  [Fri Jun 16 22:04:07 2023] Worker1: There is change in client2. Synchronizing...
13  [Fri Jun 16 22:04:07 2023] File created: dir/Makefile copy.
14  [Fri Jun 16 22:04:08 2023] Worker2: There is change in server side. Synchronizing to client3...
15  [Fri Jun 16 22:04:08 2023] Worker0: There is change in server side. Synchronizing to client4...
16  [Fri Jun 16 22:04:15 2023] Worker2: There is change in client3. Synchronizing...
17  [Fri Jun 16 22:04:15 2023] File updated: dir/Makefile copy.
18  [Fri Jun 16 22:04:16 2023] Worker1: There is change in server side. Synchronizing to client2...
19  [Fri Jun 16 22:04:16 2023] Worker0: There is change in server side. Synchronizing to client4...
20  [Fri Jun 16 22:04:20 2023] Worker0: There is change in client4. Synchronizing...
21  [Fri Jun 16 22:04:20 2023] File deleted: dir/subdir2.
22  [Fri Jun 16 22:04:21 2023] Worker2: There is change in server side. Synchronizing to client3...
23  [Fri Jun 16 22:04:21 2023] Worker1: There is change in server side. Synchronizing to client2...
24  [Fri Jun 16 22:04:34 2023] Worker2: There is change in server side. Synchronizing to client3...
25  [Fri Jun 16 22:04:34 2023] Worker1: There is change in server side. Synchronizing to client2...
26  [Fri Jun 16 22:04:34 2023] Worker0: There is change in server side. Synchronizing to client4...
27  [Fri Jun 16 22:15:50 2023] SERVER: Received SIGINT signal. Shutting down the server.
```

- ➔ Line 4-11: First synchronizations to the clients.
- ➔ Line 10: Client on queue connects to the server.
- ➔ Line 12: Client2 created a new file. It is detected and synchronized to the current directory on line 13, then synchronized to the other connected clients on lines 14 and 15.
- ➔ Line 16: Client3 edited the file. Same steps are repeated.
- ➔ Line 20: Client4 deleted a file. Same steps are repeated.
- ➔ Line24: Server edited a file. It is synchronized to all clients.

Example **client log** from client2. They are all similar except their own actions:

```
1   [Fri Jun 16 22:03:37 2023] CLIENT: Directory clidir2 is opened.
2   [Fri Jun 16 22:03:37 2023] CLIENT: Waiting for connection to server.
3   [Fri Jun 16 22:03:37 2023] CLIENT: Connected to server.
4   [Fri Jun 16 22:03:37 2023] File created: clidir2/BibakBOXClient.
5   [Fri Jun 16 22:03:37 2023] File created: clidir2/BibakBOXClient.c.
6   [Fri Jun 16 22:03:37 2023] File created: clidir2/BibakBOXServer.
7   [Fri Jun 16 22:03:37 2023] File created: clidir2/BibakBOXServer.c.
8   [Fri Jun 16 22:03:37 2023] File created: clidir2/filesync.h.
9   [Fri Jun 16 22:03:37 2023] File created: clidir2/Makefile.
10  [Fri Jun 16 22:03:37 2023] File created: clidir2/subdir1/final_project.pdf.
11  [Fri Jun 16 22:03:37 2023] File created: clidir2/subdir2/settings.json.
12  [Fri Jun 16 22:03:37 2023] CLIENT: First synchronization is made from server
13  [Fri Jun 16 22:04:07 2023] CLIENT: There is change in client side. Synchronizing to server..
14  [Fri Jun 16 22:04:16 2023] CLIENT: There is change in server side. Synchronizing from server..
15  [Fri Jun 16 22:04:16 2023] File updated: clidir2/Makefile copy.
16  [Fri Jun 16 22:04:21 2023] CLIENT: There is change in server side. Synchronizing from server..
17  [Fri Jun 16 22:04:21 2023] File deleted: clidir2/subdir2.
18  [Fri Jun 16 22:04:34 2023] CLIENT: There is change in server side. Synchronizing from server..
19  [Fri Jun 16 22:04:34 2023] File updated: clidir2/subdir2/settings.json.
20  [Fri Jun 16 22:15:50 2023] CLIENT: Server disconnected.
```

- ➔ Line 4-11: Initial synchronization.
- ➔ Line 13: A modification is made by the current client and is synchronized to the server.
- ➔ Lines 14, 16, 18: Changes on the server side are synchronized to the current client.