

CSE344 MIDTERM Report

Akif Safa Angi-200104004079

1. Problem Definition and My Solution Approach

The task is to design and implement a file server that enables multiple clients to connect, access, modify and archive the files in a specific directory located at the server side. The project has implemented as a server side and a client side programs. While the server side processes incoming commands using the child process, the parent process controls incoming connection requests. The client side connects to the server and performs certain operations.

Server side

The server side first handles the necessary handling and creates a folder with the name of the folder received from the user, making the current working directory this folder.

It opens `main_fifo_fd` as `O_NONBLOCK` and `O_RDONLY`. So it won't wait at beginning for any client to open fifo for writing. After opening for reading, also opened for dummy writing. After that it will try read request in a while loop. It will always stay alive, checks if server is not full and any client waits for the connection or just try to read request from `main_fifo_fd`.

The server side continuously reads incoming commands from the `main_fifo_fd` (server FIFO) and accepts new clients based on the server's occupancy status. If the server is full, the clients can be queued by putting them in the `clients[]` array and connecting them when the number of clients on the server decreases.

After a client connects, a child process is opened with a fork. Using the client pid, communication between the client and the child process is provided between this specific client and child process via `client_fifo`.

The client sends the desired command to the child process and the necessary actions are taken according to the requested command.

In any case, if the communication between the client and the child process ends, the parent process on the Server side makes sure that both processes are closed. It reduces the number of clients by 1 and connects the waiting clients to the server.

If the parent process needs to shut down, it sends a shutdown request to all clients. After all child processes and clients are shut down properly, the parent process also shuts down.

The server side also keeps incoming client requests and client outputs in a log file.

Signal Handling

Implemented signal handlers for `SIGCHLD` and `SIGINT` signals. The `SIGCHLD` handler collected the terminated child processes and set the arrays containing the client and child pids to -1 (so that the new client and child can use this slot). It recorded the output message in the log file and reduced the client count by 1. The `SIGINT` handler set a flag to indicate interruption, which was checked at various points in the program to ensure clean termination. Closing files and semaphores. On the parent process deleting fifos, calling `kill_childs()` function to terminate child processes and clients etc.

Error Handling

Error handling mechanisms were implemented to detect and handle errors related to necessary creations, data/command transmission, child process operations and child process execution. While errors on the parent process side were suppressed, only potentially critical errors were suppressed on the client side. Situations such as the absence of a file that would not negatively affect the processes were only sent to the client as a message.

Log Handling

The connection requests, connection status and exits of the clients were written to the log file.

Semaphore Handling

Semaphores were used to prevent race conditions when reading and writing files on the server and to ensure correct communication between the client and child processes. Typical reader and writer semaphore usage handled in the file semaphores.

clients and childPids Arrays

Child process and client pids were kept in arrays to be used in necessary situations such as server shutdown.

Shared Memory Handling

Shared memory was used so that child processes could access file names and the same semaphores.

Client side

On the client side, a request is sent to the server according to the user's connection mode (**tryConnect**, **Connect**). In **tryConnect** mode, if the server is full, the client does not wait to connect and shuts down. **Connect** waits until it connects to the server. After connecting to the server, it starts communicating with the child process created for itself in the Server side, using the client FIFO created using its own pid.

Here, it transmits the operation it wants to be performed to the child process via fifo and waits for the necessary operations to be carried out and a response to be sent. This communication continues unless a quit command is entered or a specific signal is received to close the client.

Available commands can be accessed by sending **help command to child process**.

Signal Handling

Implemented signal handlers for SIGTERM and SIGINT signals. The SIGTERM handler waits for the server parent process to send SIGTERM signal. The SIGINT handler set a flag to indicate interruption, which was checked at various points in the program to ensure clean termination. Closing files and semaphores, deleting client fifo and semaphores and sending SIGINT to child process. So child process will be also terminated if client has terminate.

Error Handling

Error handling mechanisms were implemented to detect and handle errors related to necessary creations, data/command transmission. It prints all errors and terminate the program if there is an error in client side.

Semaphore Handling

Semaphores were used to ensure correct communication between the client and child processes.

2-) How to use

1-) After typing '**make clean**', it will delete files other than c, header file(s) and makefile (to ensure no leftover files from the old program). Later type "**make**" to compile the program.

Or you can compile one by one.

For server file type "make Server"

For client file type "make Client"

2-) To run programs

For the server file usage `./server <nameofserverdirectory> #maxclient`

For the client file usage `./client <tryConnect/Connect> <PidOfTheServer>`.

After connecting to server you can send command to server. For usable commands you can use "help" command.

Warning: If you remove or add file without using program. You should restart server.

3-) The Parts I Have Completed

I have completed all the requirements of the homework and tested.

Errors:

After adding new file to server, can't reach from other child processes using readF. There is an issue with fac_table shared memory. writeT, download and upload works. After reopened server all commands works fine.

4-) Testing and Validation

The implemented solution underwent rigorous testing to ensure functionality and correctness.

Opening Server with 2 max clients and connecting with clients

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./server deneme 2
Server Started PID 643...
Waiting for clients...
```

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client tryConnect 643
Waiting for Que..
Connection established:
Enter comment:
```

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client Connect 643
Waiting for Que..
Connection established:
Enter comment:
```

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./server deneme 2
Server Started PID 643...
Waiting for clients...
Client PID 654 connected as "client01"
Client PID 657 connected as "client02"
```

Checking ps -aux for running programs

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ps -aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	9848	564	?	Ssl	07:23	0:00	/init
root	8	0.0	0.0	9856	304	tty1	Ss	07:23	0:00	/init
a	9	0.0	0.0	18212	3684	tty1	S	07:23	0:00	-bash
root	140	0.0	0.0	9856	304	tty3	Ss	07:23	0:00	/init
a	141	0.0	0.0	18212	3668	tty3	S	07:23	0:00	-bash
root	154	0.0	0.0	9856	304	tty4	Ss	07:23	0:00	/init
a	155	0.0	0.0	18212	3688	tty4	S	07:23	0:00	-bash
root	276	0.0	0.0	9856	304	tty2	Ss	07:37	0:00	/init
a	277	0.0	0.0	18212	3672	tty2	S	07:37	0:00	-bash
a	643	96.0	0.0	10748	876	tty4	R	08:35	0:21	./server deneme 2
a	654	0.0	0.0	10708	744	tty1	S	08:35	0:00	./client tryConnect 643
a	655	0.0	0.0	10756	456	tty4	S	08:35	0:00	./server deneme 2
a	657	0.0	0.0	10708	740	tty2	S	08:35	0:00	./client Connect 643
a	658	0.0	0.0	10756	460	tty4	S	08:35	0:00	./server deneme 2
a	659	0.0	0.0	18900	2060	tty3	R	08:35	0:00	ps -aux

Trying to connect using tryConnect as third client

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client tryConnect 643
Waiting for Que..
Server is full...
Exiting program!
```

Trying to connect using Connect as third client

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client Connect 643
Waiting for Que..
```

Server side terminal

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./server deneme 2
Server Started PID 643...
Waiting for clients...
Client PID 654 connected as "client01"
Client PID 657 connected as "client02"
Connection request PID 661... Que FULL
Connection request PID 663... Que FULL
|
```

Quitting from one client

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client Connect 643
Waiting for Que..
Connection established:
Enter comment: quit
Sending write request to server log file
waiting for logfile ...
logfile write request granted
bye..
```

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./server deneme 2
Server Started PID 643...
Waiting for clients...
Client PID 654 connected as "client01"
Client PID 657 connected as "client02"
Connection request PID 661... Que FULL
Connection request PID 663... Que FULL
client02 disconnected..
Client PID 663 connected as "client03"
|
```

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client Connect 643
Waiting for Que..
Connection established:
Enter comment: |
```

Checking ps -aux for running programs

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.0  0.0   9848    564 ?        SsL   07:23   0:00 /init
root         8   0.0  0.0   9856    304 tty1     Ss    07:23   0:00 /init
a           9   0.0  0.0  18212   3684 tty1     S     07:23   0:00 -bash
root       140   0.0  0.0   9856    304 tty3     Ss    07:23   0:00 /init
a          141   0.0  0.0  18212   3668 tty3     S     07:23   0:00 -bash
root       154   0.0  0.0   9856    304 tty4     Ss    07:23   0:00 /init
a          155   0.0  0.0  18212   3688 tty4     S     07:23   0:00 -bash
root       276   0.0  0.0   9856    304 tty2     Ss    07:37   0:00 /init
a          277   0.0  0.0  18212   3672 tty2     S     07:37   0:00 -bash
a           643  92.6  0.0  10748    876 tty4     R     08:35   2:18 ./server deneme 2
a           654   0.0  0.0   10708    744 tty1     S     08:35   0:00 ./client tryConnect 643
a           655   0.0  0.0   10756    460 tty4     S     08:35   0:00 ./server deneme 2
a           663  38.4  0.0   10708    740 tty3     S     08:36   0:22 ./client Connect 643
a           664   0.0  0.0   10756    460 tty4     S     08:37   0:00 ./server deneme 2
a           665   0.0  0.0   18900   2060 tty2     R     08:37   0:00 ps -aux
```

Help command(s)

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client tryConnect 643
Waiting for Que..
Connection established:
Enter comment: help
Available comments are :
help, list, readF, writeT, upload, download, archServer, quit, killServer
Enter comment: help readF
readF <file> <line #>
display the #th line of the <file>, returns with an
error if <file> does not exists
Enter comment: help asd
Invalid command
Enter comment:
```

Using writeT command creating file and writing some strings inside it

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client tryConnect 643
Waiting for Que..
Connection established:
Enter comment: help
Available comments are :
help, list, readF, writeT, upload, download, archServer, quit, killServer
Enter comment: help readF
readF <file> <line #>
display the #th line of the <file>, returns with an
error if <file> does not exists
Enter comment: help asd
Invalid command
Enter comment: writeT dene.txt 1 line1
5 byte(s) has been written
Enter comment: writeT dene.txt lineOld
7 byte(s) has been written
Enter comment:
```

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client Connect 643
Waiting for Que..
Connection established:
Enter comment: writeT dene.txt line3
5 byte(s) has been written
Enter comment:
```

Inside of the file

```
dene.txt [x]
1 line1
2 lineOld
3 line3
```

Write to the last line and current line again

```
Enter comment: writeT dene.txt lineNew
7 byte(s) has been written
Enter comment: writeT dene.txt 2 lineNew
7 byte(s) has been written
Enter comment: |
```

```
dene.txt [x]
1 line1
2 lineNew
3 line3
4 lineNew
```

readF

```
Enter comment: readF dene.txt
line1
lineNew
line3
lineNew
Reading file completed
Enter comment: readF dene.txt 3
line3
Enter comment: readF abc.txt
There is no file with this name.
Enter comment:
```



download txt file

```
Enter comment: download dene.txt
27 byte(s) transferred
Enter comment:
```

Create inside the client file and upload txt file



```
up.txt [x]
1 bu
2 dosya
3 upload
4 edildi
```

```
Enter comment: upload up.txt
File transfer request received. Beginning file transfer:
25 byte(s) transferred
Enter comment: |
```





 dene.txt	5.05.2024 08:40	TXT Dosyası	1 KB
 up.txt	5.05.2024 08:43	TXT Dosyası	1 KB

```
up.txt [x]
1 buC:\Users\akif\Desktop\SystemMidterm\v3\program\deneme\up.txt
2 dosya
3 upload
4 edildi
```

upload pdf and big size file from other client, also try non-exist file to upload

 ompare.exe	4.05.2024 15:31	Uygulama	12.757 KB
 report.pdf	4.05.2024 15:31	Opera GX Web Do...	64 KB

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client Connect 643
Waiting for Que..
Connection established:
Enter comment: writeT dene.txt line3
5 byte(s) has been written
Enter comment: upload report.pdf
File transfer request received. Beginning file transfer:
65421 byte(s) transffered
Enter comment: upload ompare.exe
File transfer request received. Beginning file transfer:
13062648 byte(s) transffered
Enter comment: upload nonexist.txt
File transfer request received. Beginning file transfer:
File doesn't exist in client directory.
Enter comment: |
```

 dene.txt	5.05.2024 08:40	TXT Dosyası	1 KB
 ompare.exe	5.05.2024 08:45	Uygulama	12.757 KB
 report.pdf	5.05.2024 08:45	Opera GX Web Do...	64 KB
 up.txt	5.05.2024 08:43	TXT Dosyası	1 KB

Remove pdf file from client and download from server

```
Enter comment: upload report.pdf
File transfer request received. Beginning file transfer:
File doesn't exist in client directory.
Enter comment: download report.pdf
65421 byte(s) transffered
Enter comment: |
```

After removing big size file from client directory and download from server






```
Enter comment: upload ompare.exe
File transfer request received. Beginning file transfer:
File doesn't exist in client directory.
Enter comment: download ompare.exe
13062648 byte(s) transffered
Enter comment: |
```


Server log file

```
server_log.txt
1 May 05 2024 +03 08 35 21: Client PID 654 connected
2 May 05 2024 +03 08 35 25: Client PID 657 connected
3 May 05 2024 +03 08 36 20: Connection request PID 661... Que FULL
4 May 05 2024 +03 08 36 38: Connection request PID 663... Que FULL
5 May 05 2024 +03 08 37 03: Client 657 quit
6 May 05 2024 +03 08 37 03: Client PID 663 connected
7 May 05 2024 +03 08 49 27: Client 663 quit
8 May 05 2024 +03 08 49 40: Client 654 quit
```

(Server and clients reopened with current server files)

Current server file

Ad	Değiştirme tarihi	Tür	Boyut
 abc.txt	5.05.2024 08:51	TXT Dosyası	1 KB
 dene.txt	5.05.2024 08:40	TXT Dosyası	1 KB
 ompare.exe	5.05.2024 08:45	Uygulama	12.757 KB
 report.pdf	5.05.2024 08:45	Opera GX Web Do...	64 KB
 up.txt	5.05.2024 08:43	TXT Dosyası	1 KB

```
#@DESKTOP-SFG0AM7: /mnt
#@DESKTOP-SFG0AM7: /mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./server deneme 3
Server Started PID 679..
Waiting for clients...
Client PID 689 connected as "client81"
Client PID 682 connected as "client82"
Client PID 684 connected as "client83"










#@DESKTOP-SFG0AM7: /mnt
#@DESKTOP-SFG0AM7: /mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client tryConnect 679
Waiting for Que..
Connection established:
Enter comment:

#@DESKTOP-SFG0AM7: /mnt
#@DESKTOP-SFG0AM7: /mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client Connect 679
Waiting for Que..
Connection established:
Enter comment:

#@DESKTOP-SFG0AM7: /mnt
#@DESKTOP-SFG0AM7: /mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client Connect 679
Waiting for Que..
Connection established:
Enter comment:
```

archServer

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client tryConnect 679
Waiting for Que..
Connection established:
Enter comment: archServer buBir.tar
Calling tar utility .. child PID 686
child returned with SUCCESS..
copying the archive file..
removing archive directory...
SUCCESS Server side files are achived in "buBir"
Enter comment: |
```

	deneme	5.05.2024 08:51	Dosya klasörü	
	buBir.tar	5.05.2024 08:56	WinRAR arşivi	12.830 KB
	client	5.05.2024 08:34	Dosya	40 KB
	Client.c	5.05.2024 08:32	C Kaynak Dosyası	24 KB
	FifoHeader.h	4.05.2024 16:21	C Header Kaynak ...	1 KB
	Makefile	4.05.2024 15:39	Dosya	1 KB
	server	5.05.2024 08:34	Dosya	61 KB
	Server.c	5.05.2024 08:33	C Kaynak Dosyası	61 KB
	server_log.txt	5.05.2024 08:56	TXT Dosyası	1 KB

Inside of buBir.tar file

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ tar -tvf buBir.tar
-rwxrwxrwx a/a          25 2024-05-05 08:56 abc.txt
-rwxrwxrwx a/a          27 2024-05-05 08:56 dene.txt
-rwxrwxrwx a/a      13062648 2024-05-05 08:56 ompare.exe
-rwxrwxrwx a/a       65421 2024-05-05 08:56 report.pdf
-rwxrwxrwx a/a          25 2024-05-05 08:56 up.txt
```

killServer

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client Connect 679
Waiting for Que..
Connection established:
Enter comment: killServer
Terminating client
```

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./server deneme 3
Server Started PID 679...
Waiting for clients...
Client PID 680 connected as "client01"
Client PID 682 connected as "client02"
Client PID 684 connected as "client03"
Kill signal from client682.. terminating...
```

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client tryConnect 679
Waiting for Que..
Connection established:
Enter comment: archServer buBir.tar
Calling tar utility .. child PID 686
child returned with SUCCESS..
copying the archive file..
removing archive directory...
SUCCESS Server side files are achived in "buBir"
Enter comment: Terminating client
```

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ./client Connect 679
Waiting for Que..
Connection established:
Enter comment: Terminating client
```

ps-aux

```
a@DESKTOP-5F60AM7:/mnt/c/Users/akif_/Desktop/SystemMidterm/v3/program$ ps -aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	9848	564	?	Ssl	07:23	0:00	/init
root	8	0.0	0.0	9856	304	tty1	Ss	07:23	0:00	/init
a	9	0.0	0.0	18212	3684	tty1	S	07:23	0:00	-bash
root	140	0.0	0.0	9856	304	tty3	Ss	07:23	0:00	/init
a	141	0.0	0.0	18212	3672	tty3	S	07:23	0:00	-bash
root	154	0.0	0.0	9856	304	tty4	Ss	07:23	0:00	/init
a	155	0.0	0.0	18212	3688	tty4	S	07:23	0:00	-bash
root	276	0.0	0.0	9856	304	tty2	Ss	07:37	0:00	/init
a	277	0.0	0.0	18212	3672	tty2	S	07:37	0:00	-bash
a	733	0.0	0.0	18900	2060	tty2	R	09:00	0:00	ps -aux

Last server_log file

```
server_log.txt
1 May 05 2024 +03 08 35 21: Client PID 654 connected
2 May 05 2024 +03 08 35 25: Client PID 657 connected
3 May 05 2024 +03 08 36 20: Connection request PID 661... Que FULL
4 May 05 2024 +03 08 36 38: Connection request PID 663... Que FULL
5 May 05 2024 +03 08 37 03: Client 657 quit
6 May 05 2024 +03 08 37 03: Client PID 663 connected
7 May 05 2024 +03 08 49 27: Client 663 quit
8 May 05 2024 +03 08 49 40: Client 654 quit
9 May 05 2024 +03 08 55 59: Client PID 680 connected
10 May 05 2024 +03 08 56 02: Client PID 682 connected
11 May 05 2024 +03 08 56 07: Client PID 684 connected
```