

# **CSE 344 System Programming Midterm Report**

**Ömer Kaan Uslu**

**1801042642**

# Introduction

In this project, I designed a server client file management system which allows clients to perform various file operations such as listing files, reading files, writing to files, uploading files, and downloading files. The system consists of a server program and multiple client programs that communicate using a named pipe for interprocess communication.

## System Architecture

The system follows a client-server architecture, where multiple client programs connect to a central server program. The server program handles client connections, receives commands from clients, performs the requested file operations, and sends the appropriate responses back to the clients. Each client program can independently execute commands and interact with the server.

## Design Decisions

**Communication Protocol:** The named pipe (FIFO) was chosen as the communication protocol between clients and the server.

**Signal Handling:** Both the server and client programs implement signal handlers for SIGTERM and SIGINT signals. These signal handlers gracefully handle termination requests and perform operations before exiting.

**Shared Resource Access and Synchronization:** To ensure mutual exclusion and prevent concurrent access to shared resources (such as the log file and critical sections), a semaphore mechanism is implemented. The semaphore guarantees that only one client can access the critical section at a time, preventing race conditions and conflicts.

**File Operations:** The system supports various file operations, including listing files, reading files, writing to files, uploading files, and downloading files. The commands are sent by clients to the server, which processes them accordingly and provides appropriate responses.

# Implementation Details

## **Server**

The server program does handle multiple client connections concurrently and processing commands to send back response.

It listens for client connection from fifo at first.

When a new client comes, program creates a new process with fork to handle clients concurrently.

Each client process id is used for creating a new fifo to communicate with client.

The *serveClient* function is used to serve client after those operations. This method serves client to get clients requests and send response them separately for each command. In this method, there is *sem\_post* and *sem\_wait* calls to guard synchronization. The semaphore ensures that only one client can access the critical section at a time, avoiding conflicts and data corruption.

Server log file is created and the server completes its lifetime.

## **Client**

The client program does handle communication with server. It does open fifo, and sends its process id to server.

Program takes command input and sends it to server without doing anything about this string. The response directly comes from server without parsing it on client side etc.

After sending the command, client listens for response and displays the response.

If killServer or quit commands entered, the while loop ends and program sends this commands to server in order to process this commands in server side.

# Test Results

## With one client:

```
lastODEV -- -zsh -- 80x24
kaanuslu@192 lastODEV % ./biboClient connect 31055
Enter comment: list
..
a.txt

Enter comment: readF a.txt 1
erkan hocam midterm zordu

Enter comment: readF a.txt
erkan hocam midterm zordu

midterm baya zordu

bu midterm yapilmaz dedirtti

baya baya olmadı bu is

Enter comment: killServer
Server killing..

kaanuslu@192 lastODEV %

lastODEV -- -zsh -- 80x24
kaanuslu@192 lastODEV % ./biboServer here 2
Server Started PID 31055
directory exists.
: File exists
Client1 connected with pid 31056.
kill signal from client31056.. terminating
zsh: quit ./biboServer here 2
kaanuslu@192 lastODEV %
```

## With two client:

```
lastODEV -- -zsh -- 80x24
kaanuslu@192 lastODEV % ./biboClient connect 31084
Enter comment: list
..
a.txt

Enter comment: quit
Sending write request to server log file
waiting for logfile ...
logfile write request granted
bye..

kaanuslu@192 lastODEV %

lastODEV -- -zsh -- 80x8
kaanuslu@192 lastODEV % ./biboClient connect 31084
Enter comment: readF a.txt 2
midterm baya zordu

Enter comment: killServer
Server killing..

kaanuslu@192 lastODEV %

lastODEV -- -zsh -- 80x24
kaanuslu@192 lastODEV % ./biboServer here 2
Server Started PID 31084
directory exists.
: File exists
Client1 connected with pid 31086.
Client2 connected with pid 31088.
Client1 disconnected
kill signal from client31088.. terminating
zsh: quit ./biboServer here 2
kaanuslu@192 lastODEV %
```

## With three client:

```
lastODEV -- -zsh -- 80x24
kaanuslu@192 lastODEV % ./biboClient connect 31110
Enter comment: readF a.txt
erkan hocam midterm zordu

midterm baya zordu

bu midterm yapilmaz dedirtti

baya baya olmadı bu is

Enter comment: killServer
Server killing..

kaanuslu@192 lastODEV %

lastODEV -- -zsh -- 80x21
kaanuslu@192 lastODEV % ./biboClient connect 31110
Enter comment: readF a.txt
erkan hocam midterm zordu

midterm baya zordu

bu midterm yapilmaz dedirtti

baya baya olmadı bu is

Enter comment: quit
Sending write request to server log file
waiting for logfile ...
logfile write request granted
bye..

kaanuslu@192 lastODEV %

lastODEV -- -zsh -- 80x24
kaanuslu@192 lastODEV % ./biboServer here 2
Server Started PID 31110
directory exists.
: File exists
Client1 connected with pid 31113.
Client2 connected with pid 31116.
Connection request PID 31119... Que FULL
Client3 connected with pid 31118.
Client3 disconnected
Client1 disconnected
kill signal from client31116.. terminating
zsh: quit ./biboServer here 2
kaanuslu@192 lastODEV %

lastODEV -- -zsh -- 80x15
kaanuslu@192 lastODEV % clear

kaanuslu@192 lastODEV % ./biboClient connect 31110
Enter comment: list
..
a.txt

Enter comment: quit
Sending write request to server log file
waiting for logfile ...
logfile write request granted
bye..

kaanuslu@192 lastODEV %
```