

**SCHOOL OF ENGINEERING AND BUILT ENVIRONMENT**

**Department of Computing**

BSc/BSc (Hons) Computing

**Systems Programming M3I324183-23-A**

**Diet 1 Coursework 2 2023/2024 Tri A ALC**

| **Student Name** | **Student ID** | **Student E-mail** |
| --- | --- | --- |
| **Caroline Vivian Anyango Otuoma** | **S2110913** | **c.otuoma@alustudent.com** |

**Signature: C.V.A.O Date: December 1st 2023**

**Table of Contents**

[Introduction 3](#_66ezidyxl91)

[Creating the Upload Folder and Saving Test Files on Server Side 3](#_ngjwn8ly839d)

[Menu Options 6](#_anlbafowfrio)

[Option 1: Display Name and Student ID 6](#_zdipnpjtj2pz)

[Option 2: Display 5 Random Numbers and Compute Sum 7](#_vczq0fy9wad9)

[Option 3: Display Server "uname" Information 8](#_bicgapstqj8d)

[Option 4: Display a List of Files from the"upload" Directory 9](#_62xkzx8jhw5r)

[Option 5: Request File Copy from Server to Client 10](#_uekfgbjh6h1o)

[Option 6: Exit 13](#_cluckqy0rzhm)

[Multithreaded Server Implementation 14](#_gd96kre2nsz)

[Signal Handling in the Server 17](#_8nkbkknwfrvm)

[Issues Encountered 17](#_cx9vnv9r2cmu)

[Option 5: Request File Copy from Server to Client 17](#_euxkt2v607se)

# 

# 

# 

# **Introduction**

The testing report evaluates the functionality and performance of the Client-Server File Transfer Application. Screenshots are included to visually document the application's behaviour during different scenarios, enhancing the clarity and transparency of the testing process. This report provides a detailed overview of successful implementations and identifies areas that require further attention or improvement.

# **Creating the Upload Folder and Saving Test Files on Server Side**

| **Option Tested** | **Description Of Test** | **Notes(Screenshots)** |
| --- | --- | --- |
| **Server Start and Upload Directory** | Since the Upload directory does not exist initially in the server folder, code creates it automatically when you start the server for the first time.  It also creates three test files.  Other subsequent server starts will just check and confirm the Upload directory exists. | Initial files in server folder:  After the Server starts for the first time:    Test files: |
| **Client-Server connection** | Clients connect to the server. The server registers a connection |  |

# **Menu Options**

## **Option 1: Display Name and Student ID**

| **Option Tested** | **Description Of Test** | **Notes(Screenshots)** |
| --- | --- | --- |
| **Hard-coding student info into the system** | To retrieve the hardcoded name and student ID from the server, prefixed with the server IP address. |  |

## **Option 2: Display 5 Random Numbers and Compute Sum**

| **Option Tested** | **Description Of Test** | **Notes(Screenshots)** |
| --- | --- | --- |
| **Display 5 Random Numbers and Compute Sum** | Retrieve an array of 5 random numbers from the server in the range 100 to 500. Calculated and displayed the sum at the client side. |  |

## **Option 3: Display Server "uname" Information**

| **Option Tested** | **Description Of Test** | **Notes(Screenshots)** |
| --- | --- | --- |
| **Display Server "uname" Information** | To retrieve and display the "uname" information of the server, including OS name, release, and version |  |

# 

## **Option 4: Display a List of Files from the"upload" Directory**

| **Option Tested** | **Description Of Test** | **Notes(Screenshots)** |
| --- | --- | --- |
| **Display List of Files in "Upload" Directory** | To retrieve and display a list of names for regular files in the server's "upload" directory |  |

## 

## **Option 5: Request File Copy from Server to Client**

| **Option Tested** | **Description Of Test** | **Notes(Screenshots)** |
| --- | --- | --- |
| **Request File Copy from Server to Client** | Copy the contents of a specified file from the server's "upload" directory to the client's current working directory | Initial Contents of client folder:  User prompted to enter file name:    File copies to the client folder:   Error handling for copying files that are not in the server’s upload folder: |

## 

## **Option 6: Exit**

| **Option Tested** | **Description Of Test** | **Notes(Screenshots)** |
| --- | --- | --- |
| **Existing with option 6 for the client** | Copy the contents of a specified file from the server's "upload" directory to the client's current working directory |  |

# **Multithreaded Server Implementation**

## 

| **Option Tested** | **Description Of Test** | **Notes(Screenshots)** |
| --- | --- | --- |
| **Multithread server** | Each connected client triggers the creation of a new thread, allowing the server to handle multiple clients simultaneously. The code executed by each thread is identical, utilizing POSIX threads for efficient implementation. |  |

# **Signal Handling in the Server**

## 

| **Option Tested** | **Description Of Test** | **Notes(Screenshots)** |
| --- | --- | --- |
| **<ctrl> + c or SIGINT trigger** | A signal handler in the server to gracefully exit, clean up resources, and display the total execution time. |  |

# **Issues Encountered**

## **Option 5: Request File Copy from Server to Client**

| **Issue Faced** | **Possible Solution** | **Notes(Screenshots)** |
| --- | --- | --- |
| The client freezes during file transfer, and control is not returned to the menu  -Suggest possible causes: Imprope  r socket handling, failure to close sockets, or issues in the file transfer loop. | Recommend reviewing the client and server code for proper socket handling and ensuring sockets are closed appropriately. | Copying file test1.txt successfully but Failing to exit the transfer option:  Forceful exit using control c on the client results in a non-stop loop on the server side: |