

**12/6/2019**

Individual Report

**System Programming**

**Testing document**

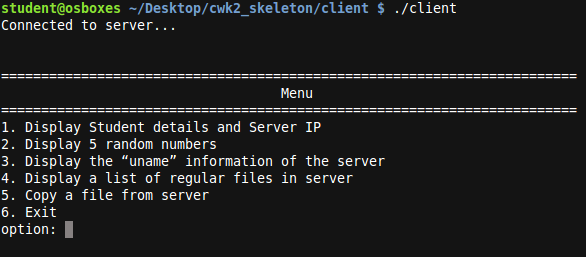
Nname: Tício Torcato Fortuna Victoriano Email: tvicto200@[caledonian.ac.uk](http://caledonian.ac.uk/) Student id: S1803453



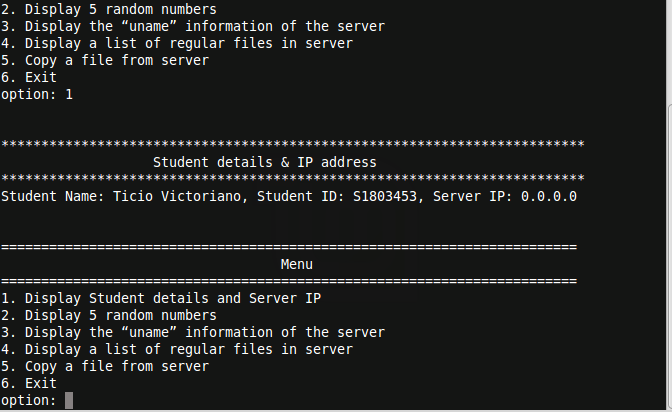
# Testing Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Testing description** | **Expected results** | **Actual result** |
| Display menu with available options | Testing if the menu is displayed when the client process starts executing and always right after an option is executed. | Menu displayed whenever the client process starts executing and always right after an option is executed. | Menu displayed.  Refer to fig. 1, fig. 2 and fig. 3 in the appendix section. |
| Display student details and Server IP address. | Testing whether the client is able to request for the student information and the server address from the established socket connection. | The client gets my hardcoded name and student ID from the server prefixed with the server IP  address and display it. | The client successfully received the information from the server and displayed it on console.  Refer to fig. 2 |
| Display the “uname” information of the server including OS  name. | Testing if the client can request and get from the server the uname information of the server. | The client gets the “uname” information of the server including OS name  and display it on screen. | The client successfully requests and gets the information accordingly.  Refer to fig. 3 |
| Display a list of files in the server. | Testing if the client can request the list of filenames for the regular files present on the server in its “upload” directory. | The client gets a list of the files in the server and displays it on console. | The client successfully gets the information.  Refer to fig. 4 |
| Copy file from the server | Testing if the client sends a request to copy a specified file in the server and have the file copied. | The file gets a copy of the file from the server | The file is copied to the client’s machine.  Refer to fig. 5, fig. 6, fig. 7 and fig. 8. |
| copy file that doesn’t exist. | Testing if the server can handle appropriately requests to copy a file that doesn’t exist in the upload folder. | Not copying the file and inform the client that file doesn’t exist. | Client informed that file don’t exist, and file not copied.  Refer to fig. 9 |
| Exiting client’s process with an option | Testing if the client can shut down the process (program) when the user enters the option 6 to exit the  program. | Have the process terminated when option 6 is entered. | Program successfully terminated.  Refer to fig. 10 |
| Enter an invalid option | Testing if the program in the client machine can handle appropriately when the user enters an invalid option. | Inform the user that the option entered is not valid. | User is informed that the option he entered is invalid.  Refer to fig. 11 in appendix. |
| Request 5 random numbers to the server | Testing if the can request 5 random numbers to the server and the server is able to process the request. | The server should generate the five random numbers and send them to the client. | 5 random numbers sent to the client.  Fig. 12 |

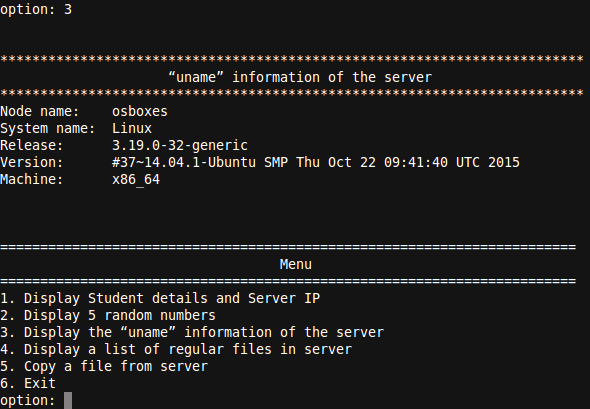
**Appendix**



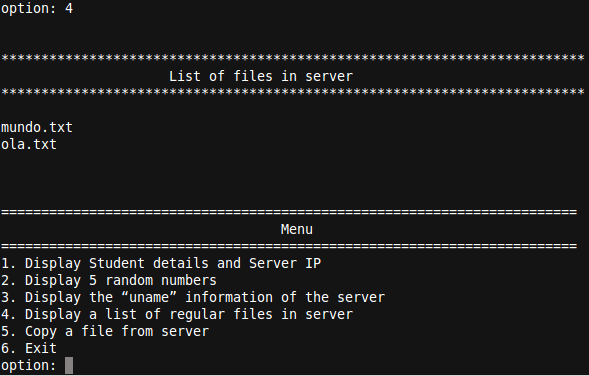
## Fig. 1



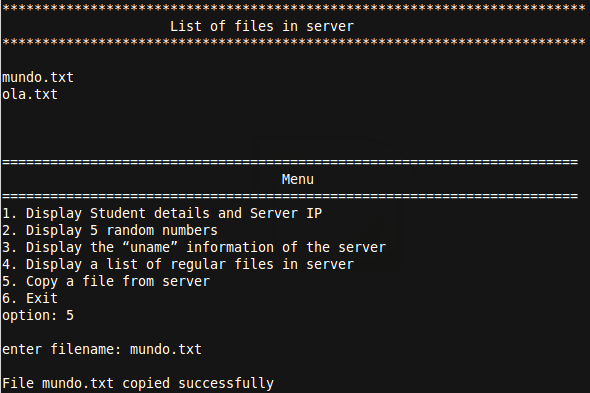
**Fig. 2**



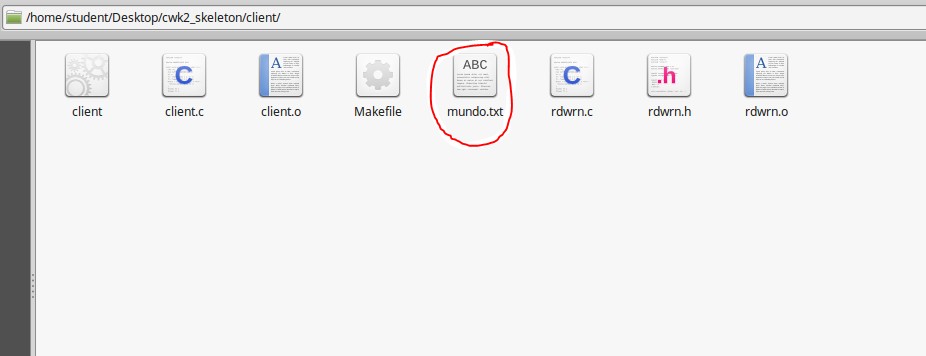
**Fig. 3**



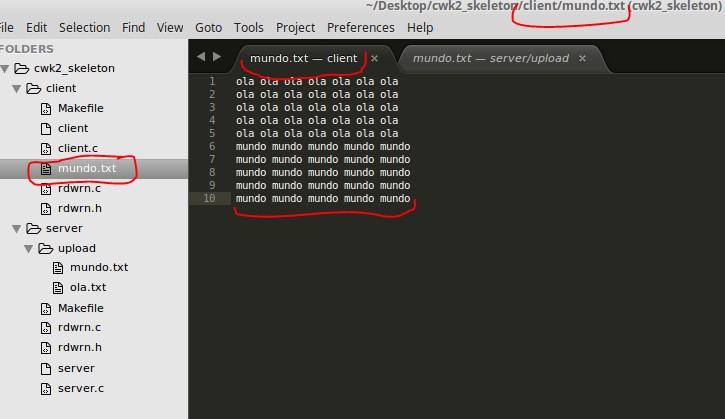
**Fig. 4**



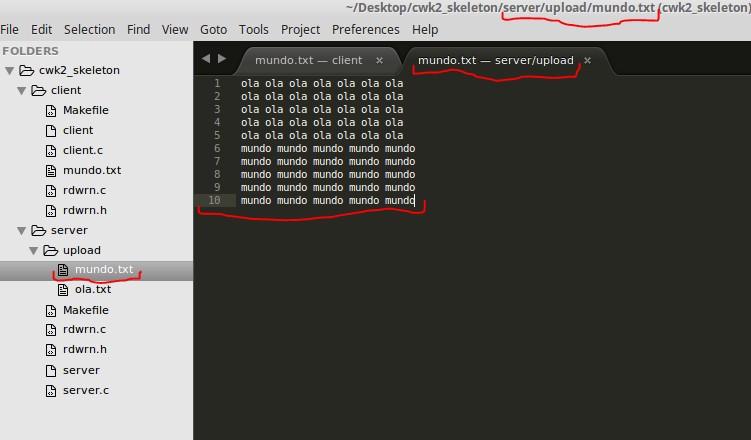
**Fig. 5**



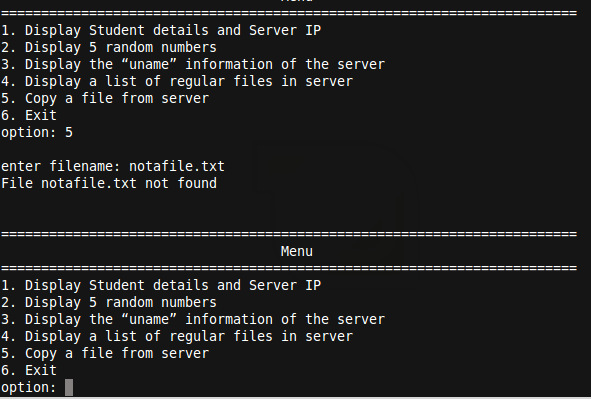
**Fig. 6**



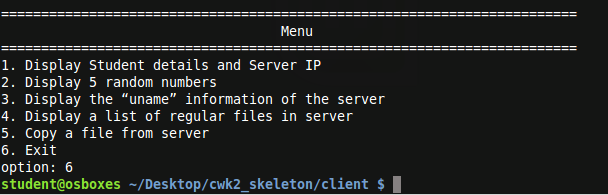
**Fig. 7**



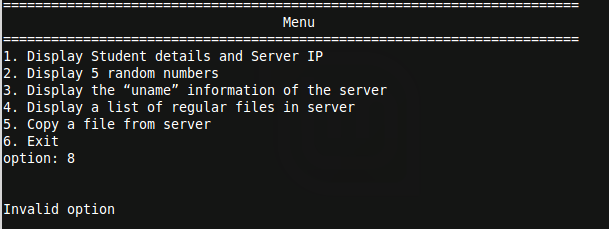
**Fig. 8**



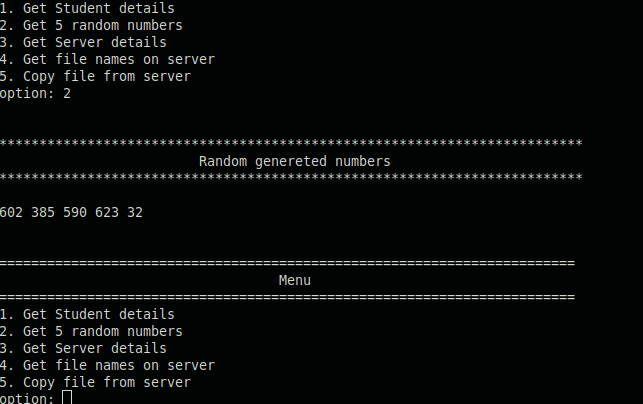
**Fig. 9**



**Fig. 10**



**Fig. 11**



**Fig. 12**