# **Test Documentation file for Client-Server FTP project**

#Overview

This Project is implemented using C++ programming language, it has two folders named Client\_prg and Server\_prg where all client and server related cpp files and makefiles are stored respectively.

//To run Server

Type make

./iftp\_server -sr <protocol> <ip> <port>

//To run Client

./iftp\_client -cl <protocol> <ip> <port>

//After connecting to server client will be allowed to enter basic commands related to file operations like  
  
-ls  
-read  
-create

-rm

-cp

# Server-Side Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Testcase no | Command | Description | Steps to Replicate | Expected Behaviour | Status |
| Test-C1 | 1. -sr | Calls Server side main function | 1. Run makefile 2. Run the executable file along with ‘-sr’ has commandline argument. | Server should run | PASS |
|  | 1. -tcp | Calls Server main function with TCP protocol | 1. While running the executable file, pass ‘-tcp’ as commandline argument. | TCPServer must run | PASS |
|  | 1. -sr -tcp <ip> <port> | Assigns that ip and port to the server | 1. Run the executable file along with the command line parameters like  -sr  -tcp  <ip>  <port> | TCP Server running on <ip>:<port> | PASS |
| Test-C2 | -sr -tcp <ip> <port> | If port is in use already | - | Error binding socket. | PASS |

# Client-Side Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Testcase no | Command | Description | Steps to Replicate | Expected Behaviour | Status |
| Test-C1 | -cl | Calls Client side main function | 1. Run makefile 2. Run the executable file along with ‘-cl’ has commandline argument. | Client should run | PASS |
| Test-C2 | -tcp | Calls Client main function with TCP protocol | 1. While running the executable file, pass ‘-tcp’ as commandline argument. | TCPClient must run | PASS |
| Test-C3 | -ls | a) Lists all the files within the provided path | 1. Enter ‘-ls’ command 2. Enter the directory path | Must list all the files in specified directory | PASS |
|  | -ls | b) If the directory path specified is a root directory then permission must be denied | 1. Enter ‘-ls’ command 2. Enter ‘/root’ as directory path | “Permission Denied !“ | PASS |
|  | -ls | c) If the path specified does not exist or is not a directory | 1. Enter ‘-ls’ command 2. Enter any invalid path. | “Invalid Path or Directory not found.” | PASS |
| Test-C4 | -rm | a) Removes the specified file | 1. Enter ‘-rm’ command 2. Enter the file name. | “File removed successfully.” | PASS |
|  | -rm | b) If the file specified does not exist then it should through an error. | 1. Enter “-rm” command. 2. Enter any invalid filename or file which doesn’t exist. | “Failed to remove file.” | PASS |
| Test-C5 | -read | a) Reads the content inside the file . | 1.Enter “-read” command.  2. Enter the path of the file. | Must read the content inside the specified file. | PASS |
|  | -read | b) If the file specified is an invalid filename. | 1.Enter “-read” command.  2. Enter the path of the file that doesn’t exist or invalid. | File not found. | PASS |
| Test-C6 | -cp | a) Copies content of one file to another. | 1.Enter “-cp” command  2.Enter the source file path from where the content needs to be copied.  3.Enter the destination file path where the content has to be copied. | File copied successfully | PASS |
| Test-C7 | -create | a) A File will be created in the specified path. | 1.Enter “-create” command.  2. Enter the path where the file need to be created along with file name. | File created successfully | PASS |
|  | -create | b) If file already exists with same file name. | 1.Enter “-create” command.  2. Enter the path where the file need to be created along with file name. | “Error File already Exists” | FAIL |
|  | -create | c) If file is being created inside root directory. | 1.Enter “-create” command.  2. Enter “/root/filename” as the path. | “Permission Denied !“ | PASS |
|  | -create | d) If the path provided to create the file is invalid. | 1.Enter “-create” command.  2. Enter an invalid file path. | “File Path not found” | PASS |
| Test-C8 | -rename | a) To rename a file. | 1.Enter “-rename” command.  2. Enter the old filename.  3.Enter the new filename. | File Renamed Successfully. | FAIL |
| Test-C9 | -cl -tcp <ip>  <port> | a) Connecting to respective server | 1.Run the executable file along with the command line parameters like  -cl  -tcp  <ip>  <port> | “Connected to Server on port <port>” | FAIL |
| Test- C10 | exit | 1. Client gets disconnected | 1. Enter ‘exit’ command. | Client should be Disconnected | PASS |
| |  | | --- | | **Test-C11** | | -cp | a) Copy file to invalid path | 1.Enter “-cp” command.  2. Enter the source filepath and filename.  3.Enter any invalid destination filepath. | “Destination path not found.” | FAIL |
|  | -cp | b) Copy from and to the same file | 1. Enter ‘-cp’ command. 2. Enter the source file name. 3. Enter the same source filename as destination filename. | Source and destination are same. | FAIL |
| Test-C12 | -write | a) To write content inside a file | 1.Enter “-write” command.  2. Enter the file path where the content needs to be written.  3.Type the content and click enter. | “Content written Successfully.” | FAIL |
|  | -write | b) Write to non-existent file | 1. Enter ‘-write’ command. 2. Enter the filename that doesn’t exist. 3. Type the content and click enter. | File not found | FAIL |
|  | |  | | --- | | -write | | 1. Write empty string | 1. Enter ‘-write’ command. 2. Enter the filename. 3. Type the empty string “” and click enter | Empty content written successfully | FAIL |