|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case | Description | Precondition | Steps | Exp Result | Actual Res  (Pass/Fail) |
| TC1 | Start program (Server) | (1) On Linux Terminal | (1) Go to folder with the application via terminal  (2) type ./server -t server | Server mode started, program listening on message queue | Pass |
| TC2 | Start program  (Client) | (1) TC1 complete  (2) On Linux Terminal | (1) Go to folder with the application via terminal  (2) type ./server -t client -f “sometext4096.txt” -p 1024 | (1) Server receives request from IPC  (2) Client Application starts receiving data from server IPC | Pass |
| TC3 | Multiple clients connecting at the same time | (1) Server running  (2) TC2 successful  (3) On multiple linux terminals | Client 1:  ./server -t client -f “sometext4096.txt” -p 512  Client 2:  ./server -t client -f “sometext4096.txt” -p 1024 Client 3:  ./server -t client -f “sometext4096.txt” -p 2048 | Server request obtained from all 3 clients, all 3 clients will start reading from message queue after successful init message from server | Pass |
| TC4 | Multiple client requests in progress | TC3 started |  | All clients, with different priorities, will transfer at different speeds | Pass |
| TC5 | Client request finished | TC4 finished |  | All clients will receive end message from server via message queue, and terminate accordingly.  Result from each client with respect to number of message received and bytes received will be printed | Pass |
| TC6 | Server request(s) complete | TC3 started |  | Server will print out after the file requested has finished being read, and will send last message to msg queue | pass |
| TC7 | Incorrect usage of args (server) | (1) On Linux Terminal | (1) Go to folder with the application via terminal  (2) type ./server -t server -f “sometext4096.txt” | Usage dialog will be printed to console, program ends | Pass |
| TC8 | Incorrect usage of args (client) | (1) On Linux Terminal | (1) Go to folder with the application via terminal  (2) type ./server -t client -f “sometext4096.txt” | Usage dialog will be printed to console, program ends | Pass |



Figure 1-TC1 - Starting application, server shows the created/accessed msg queue ID and it’s own PID on terminal

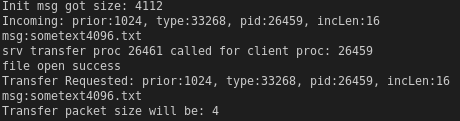
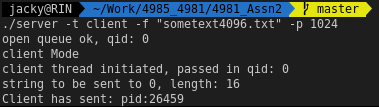


Figure 2- TC2: A single client connected to server (left – client view; right – server view)

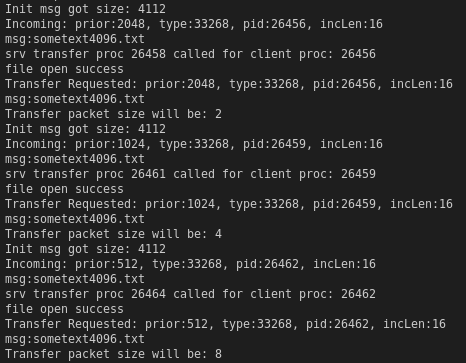


Figure 3- TC3: Multiple Clients request to server (server view)

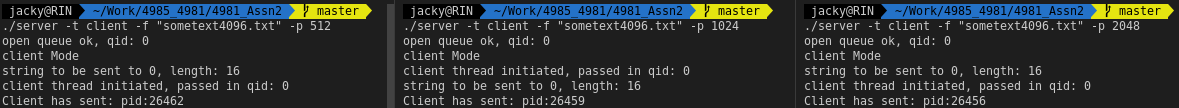


Figure 4- TC3: Multiple Clients requests to server (client view)



Figure 5- TC4: Different speed for different priorities (left to right: priority = 512, 1024, 2048, respectively)



Figure 6- TC5: Different priority, different transfer results (left to right: priority = 512, 1024, 2048, respectively)

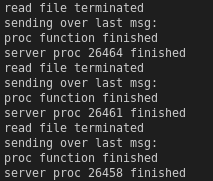


Figure 7- TC6: Server printing out end message,and sending end message to Clients (note in this figure, the server proc ID is the same as figure 3)



Figure - TC7: Running server without correct params



Figure - TC8: Running Client without correct params