Tech MANUAL

COMP 4981 Message File Server

Table of Contents

[1 Intro to Linux Message File Server 2](#_Toc442122976)

[2 Starting Client and Server 2](#_Toc442122977)

[3 Sending and receiving files 4](#_Toc442122978)

[4 Notes 6](#_Toc442122979)

# Intro to Linux Message File Server

*Type* ***make*** *in the terminal to compile the files.*

*Type ./msgqcs to run the program.*

This application is a basic **Client** and **Server** system for Linux to demonstrate inter-process communications using message queues. The message file server receives packets of small information sent from a client through the kernel to communicate.

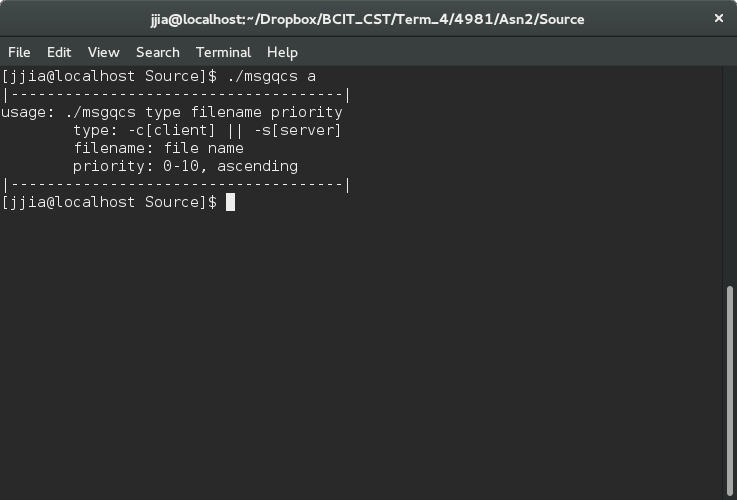
Overview:

* Client specifies the filename and priority on the command line
* Client sends its process ID to the server through message queue
* Server sends its process ID back to the client when a query has been received
* Server creates a child process to handle the client
* Client receives the server’s process ID, and sends the file name and priority level to the server
* Server’s child process receives the filename, attempt to open and read from it, and send each packet with the size based off of the priority level
  + - Priority 1 is 40 bytes
    - Priority 100 is 4000 bytes
* Client receives packets, and print each data on the screen

# Starting Client and Server

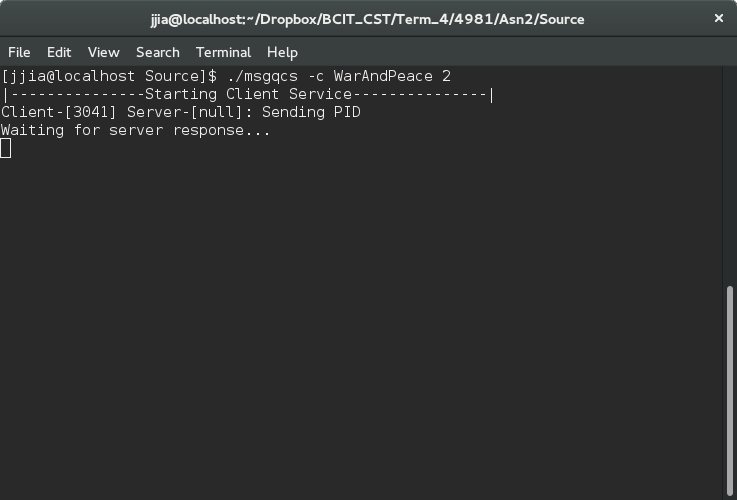
Both the **Client** and **Server** is run by the same program.

Program Usage:

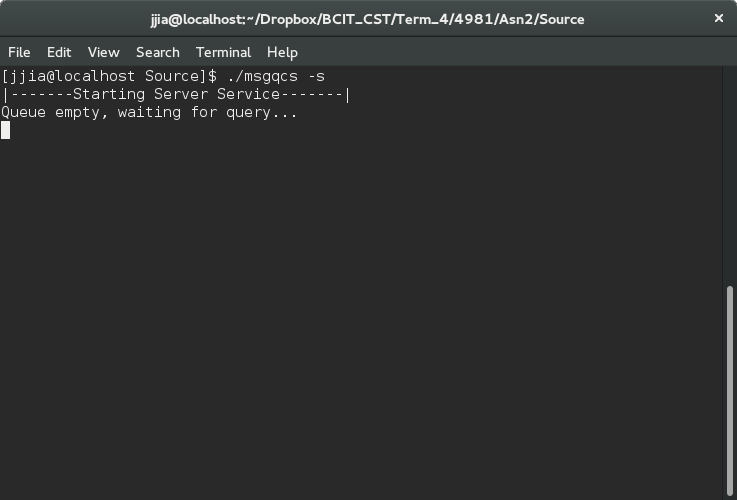


To specify which program to run enter:

* + *./msgqcs -c [filename] [priority]* for **Client**
    - * *[filename]* is the name of the file the server will open and read from
      * *[priority]* is the priority of the connection, ranging from 1 to 100 (1 being the smallest)



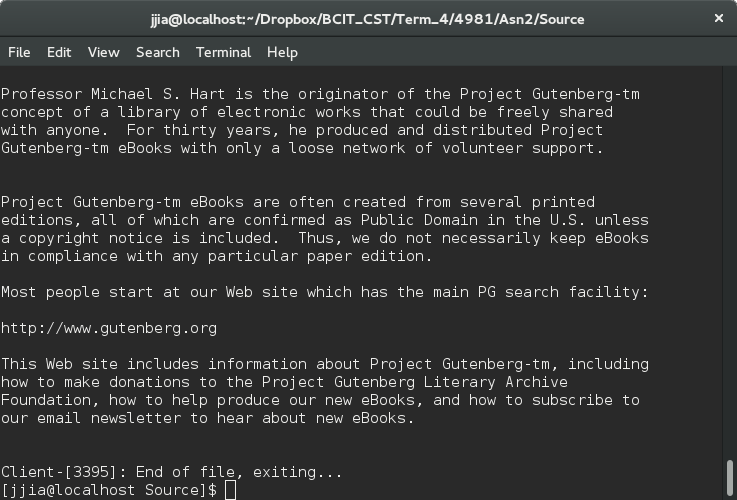
* + *./msgqcs -s*for **Server**



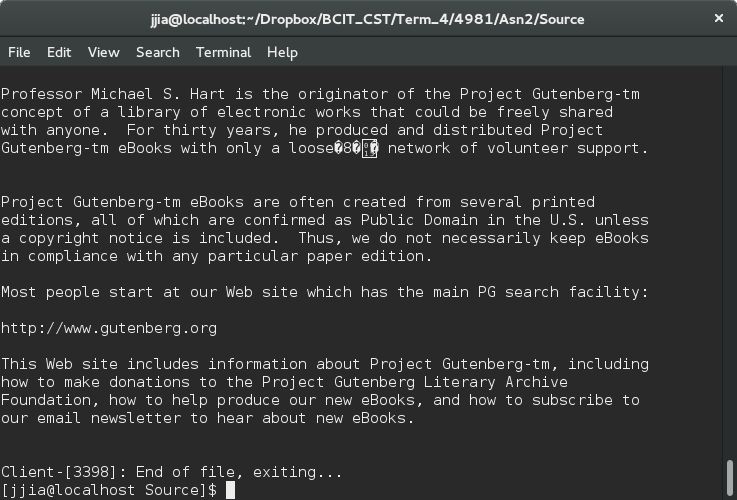
# Sending and receiving files

Multiple file communications between **Clients** and **Server**

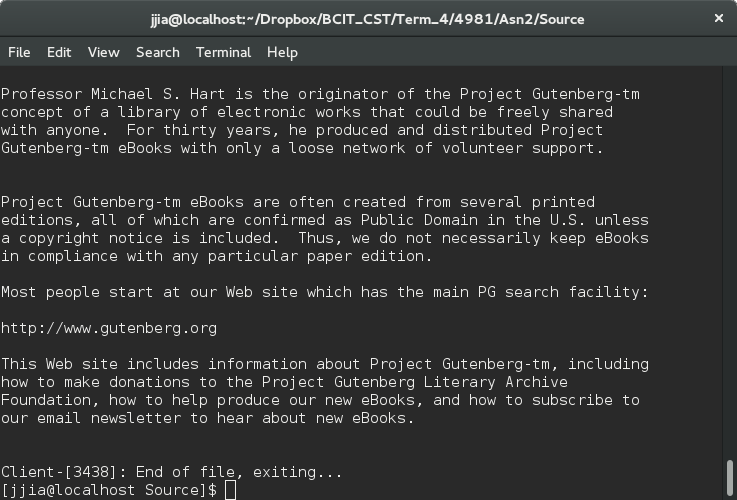
**Client 1 (Filename: WarAndPeace Priority: 1)**

****

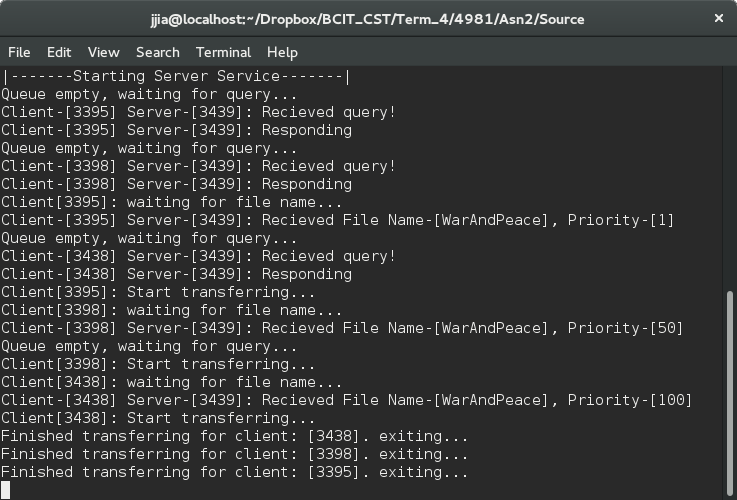
**Client 2 (Filename: WarAndPeace Priority: 50)**

****

**Client 3 (Filename: WarAndPeace Prirority:100)**

****

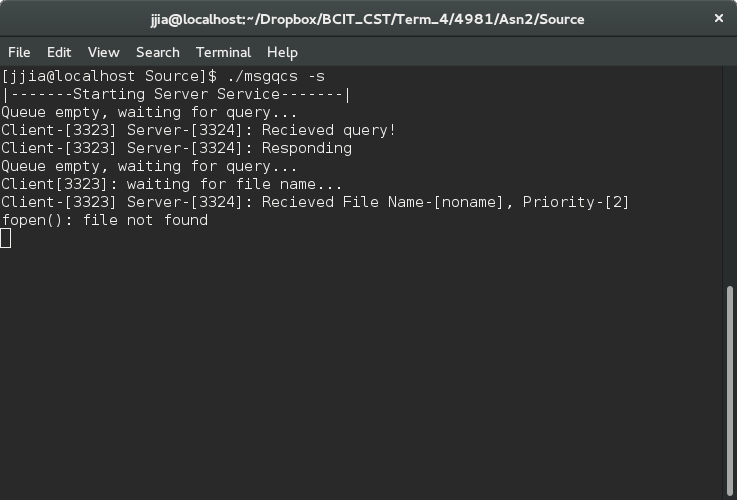
**Server (Note: Client 3 finished first because it had the highest priority)**

****

# Notes

*[filename]* has to be a valid file on the server side. If not a message will be sent to the client indicating a **FILE\_NOT\_FOUND** error

Invalid file on the server



Client receives an invalid file from server

