CIS*3210 Computer Networks

Assignment 1

For this assignment you will implement a network simulation in C using named pipes. You will create seven C programs named **n1.c** to **n7.c** to simulate the nodes/computers in the network, and six named pipes named **link1** to **link7** to simulate the network links/connections. n1 reads a text file from the user followed by a space and the destination node. For example, "test.txt n4".

It then sends off the message towards the destination in packets containing one character + the destination. Use a (uniform) random number generator to introduce a **delay** of between 1 and 2000 milliseconds before sending each packet using **usleep**. These are forwarded until they reach their destination, which prints the text file to the screen and sends back a "received" message(s) with destination n1. When n1 receives confirmation that the message was received it asks the user if they want to continue, Y/N. If yes, a new message and destination is read from the user, and is sent off the same way as before. If no, it sends a terminate message to every other process, which send back an acknowledge message and exit. Once all other processes have exited, n1 also exits.

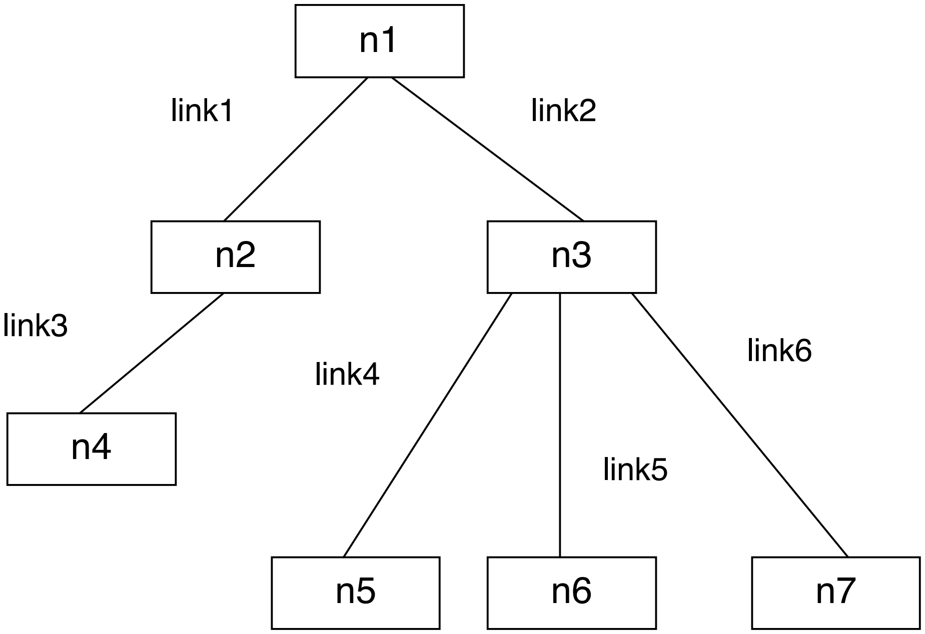

Figure 1. Network Connections

Although named pipes can be used by any process, for this assignment only the nodes connected to the named pipe can use it to send and receive messages as shown in Figure 1. Submit all of the programs with a Makefile and readme.txt in a tar file on Courselink. The Makefile should compile all of the nodes, and typing "**make run**" should start all of the programs.

For testing, you should be able to start each process in a separate terminal window, and watch it work.

You will have to decide on protocols and timing. You may hard code the network structure. (Each program knows which named file(s) it can access, and which programs it can communicate with.) **No forks or sockets** – just named pipes. **Processes can only open the named pipes in O_WRONLY or O_RDONLY, blocking mode.**

Marking scheme:

1 mark for documentation and formatting
1 mark for proper startup from "make run" (assuming program worked as required – non-functional code that starts is a 0)
6 marks for transmitting the message to every node as instructed
2 marks for clean exit on quit (assuming program worked as required – non-functional code that exits cleanly is a 0)
/10 marks total