Operating Systems **CSI3131** *Winter 2012*

Assignment 1

Process/thread creation and inter-process communication

You must submit your assignment on-line with Virtual Campus. This is the only method by which we accept assignment submissions. Do not send any assignment by email. We will not accept them. We are not able to enter a mark if the assignment is not submitted on Virtual Campus! The deadline date is firm since you cannot submit an assignment passed the deadline. You are responsible for the proper submission of your assignments and you cannot appeal for having failed to do so. A mark of 0 will be assigned to any missing assignment.

Assignments must be done individually. Any team work, and any work copied from a source external to the student (including solutions of past year assignments) will be considered as an academic fraud and will be forwarded to the Faculty of Engineering for imposition of sanctions. Hence, if you are judged to be guilty of an academic fraud, you should expect, at the very least, to obtain an F for this course. Note that we will use sophisticated software to compare your assignments (with other student's and with other sources...). This implies that you must take all the appropriate measures to make sure that others cannot copy your assignment (hence, do not leave your workstation unattended).

Goals

Practise:

1. process creation in Linux using fork() and exec() system calls
2. thread creation in Linux with the pthread API
3. inter-process communication using pipes

Posted: January 28 2012
Due: Feb 12 2012 (midnight)

Description (Please read the complete assignment document before starting.)

The software used in this assignment creates a set of processes, threads and pipes that simulate an Ethernet/802.3 LAN network; it is described in the design document “Ethernet/802.3 Simulation Software Design”. Processes shall represent five devices in the network, that is, four stations and a network hub. Pipes represent twisted pair wires (two pipes per connection between a station and the hub). Threads are used within the hub process to monitor each of the “T-pair” pipes for retransmission across the “R-Pair” pipes. See the design document for details on the Ethernet/802.3 LAN and the simulation software.

The station program *stn*, in `stn.c`, is complete. Your task shall be to complete the *hub* program in `hub.c` such that the execution of the program `hub` creates the station processes and monitors “T-pair” pipes and transmits across the “R-pair” pipes.

To complete the assignment:

1. The file assign1.tar contains the source code for your assignment. To extract the files, transfer the file to the *siteDev* system and execute the command “*tar –xvf assign1.tar*”.
2. All your work will be completed in `hub.c`. Complete the documentation in the source file to indicate your name and student number. Take the time to document well your code. Consider taking the time to consider all information provided in this document, understand the system calls to be used, and design before you start coding.
3. Complete the following functions in `hub.c`: `createStation()` and `createHubThreads()`.
4. The programs should be compiled using the commands “`cc –o stn stn.c`” and “`cc –o hub hub.c –lpthread`” (note that for `hub` compilation, the pthread library must be explicitly specified). The file `Makefile` has been provided to compile both files – simply type in the command `make`.
5. To submit the assignment, upload the file *hub.c*. Do not forget to click on the submit button after (and only after) you have uploaded the file.
6. A word of caution, for debugging the programs, if you wish to write messages to the terminal from the station processes, you should write to the standard error using the following library call: `fprintf(stderr,”message string\n”)` as the standard input and standard output are to be modified.
7. See point 4 in “Background Information” in the design document for hints on how to observe processes/threads and pipes to help debug your programs.
8. When `hub` is run, the output similar to the following should appear on your screen (Note that PIDs enclosed in the parentheses shall be specific to your execution). The messages sent and received are enclosed in the characters ‘>’ and ‘<’.

```
Station A (1835): Sent to station C >Hello station C<
Station B (1836): Sent to station D >Hello station D, it's me station B<
Station C (1837): Sent to station A >Message 1<
Station D (1838): Sent to station A >Message a <
Station C (1837): Received from station A >Hello station C<
Station A (1835): Received from station C >Message 1<
Station A (1835): Received from station C an acknowledgement
Station A (1835): Sent to station C >Received your acknowledgement to first message<
Station A (1835): Received from station D >Message a <
Station D (1838): Received from station B >Hello station D, it's me station B<
Station C (1837): Received from station A an acknowledgement
Station C (1837): Sent to station A >Message 2<
Station C (1837): Received from station A >Received your acknowledgement to first message<
Station D (1838): Received from station A an acknowledgement
Station D (1838): Sent to station A >Message b<
Station B (1836): Received from station D an acknowledgement
Station B (1836): Sent to station D >Thanks for your acknowledgement<
Station D (1838): Received from station B >Thanks for your acknowledgement<
Station A (1835): Received from station C >Message 2<
Station A (1835): Received from station C an acknowledgement
Station A (1835): Sent to station C >Have a good day - see you next time<
Station A (1835): Received from station D >Message b<
Station B (1836): Received from station D an acknowledgement
Station B (1836): Sent to station D >Got to go - see you later<
Station C (1837): Received from station A an acknowledgement
Station C (1837): Sent to station A >Message 3<
Station C (1837): Received from station A >Have a good day - see you next time<
Station D (1838): Received from station A an acknowledgement
Station D (1838): Sent to station A >Message c<
Station A (1835): Received from station C >Message 3<
Station A (1835): Received from station C an acknowledgement
Station D (1838): Received from station B >Got to go - see you later<
Station B (1836): Received from station D an acknowledgement
Station C (1837): Received from station A an acknowledgement
Station C (1837): Sent to station A >Message 4<
Station A (1835): Received from station D >Message c<
Station A (1835): Received from station C >Message 4<
Station C (1837): Received from station A an acknowledgement
Station C (1837): Sent to station A >Message 5<
Station D (1838): Received from station A an acknowledgement
Station D (1838): Sent to station A >Message d<
Station A (1835): Received from station C >Message 5<
Station C (1837): Received from station A an acknowledgement
Station A (1835): Received from station D >Message d<
Station D (1838): Received from station A an acknowledgement
Station D (1838): Sent to station A >Message e<
Station A (1835): Received from station D >Message e<
Station D (1838): Received from station A an acknowledgement
```