COSC 3360 – 6310 – Operating System Fundamentals
Assignment #2 for Fall 2018: The token ring simulator.
Due on Monday, October 29 at 11:59:59 PM

Objective

This assignment will introduce you to interprocess communications mechanisms in UNIX using pipes and sockets.

Specifications

You must write two programs to simulate the operation of a token ring network. A token ring network uses the token passing access method, which is based on a small frame called a token that circulates a ring-shaped media in a logical direction when devices are idle. To transmit information, the device must seize the token. This method guarantees that only one device at a time will have access to the network.

The server program: the user will execute this program using the following syntax:

./exec_filename port_no < input_filename

where exec_filename is the name of your executable file, port_no is the port number to create your socket, and input_filename is the name of your input file.

This program receives the information about the port number from the command line and reads the information about the size of the ring and the parameters for the simulation from a file using I/O redirection.

Input Format: the input file will look like:

```
3          // Number of processes in the ring
1          // Access request from P0 in the ring
0          // 0 = Network not needed
.          // 1 = Network needed.
.
-1         // End the simulation
```

- The size of the ring is a positive integer value greater than 1.
- The number of access requests in the simulation can be from 1 to *n*, where *n* can be greater than the size of the ring (the ring is a circular singly linked list of processes).
- An access request equal to -1 will end the execution of the simulation.

After reading the input file, the server program will create a socket to: a) receive the request from the client program (ring of processes) about the size of the ring; b) send the size of the ring to the client program; and c) answer the request of each process in the ring about the access request value. The server program will finish its execution after sending the access request value -1 (end of simulation) to a process in the ring.

The ring of processes program: this program creates a circular singly link list of processes using fork() and pipes. The parent process will: a) prompt the user for the hostname and port of the server program; b) send the request about the size of the ring to the server program; c) create the ring of processes (the id of the processes starts from 0); d) pass the token to the first process in the ring (can be the parent process); and e) Finish the simulation. The parent process must guarantee that all the processes of the ring end before finishing its execution.

Each process in the ring must: a) wait for the token (if the token value equals to end_of_simulation, skip steps b and c); b) connect to the server program to get the access request value (0 = Network not needed, 1 = Network needed, -1 = End simulation); c) print the following message “Process n is using the network”, where n is the process id, if the access request value is equal to 1; and d) pass the token to the next process in the ring. If the access request value is equal to -1 or the token value is equal to end_of_simulation, the process will finish its execution.

Implementation

Given the following input:

```
3
0
1
0
0
1
0
1
-1
```

The expected output is:

```
Process 1 is using the network
Process 1 is using the network
Process 0 is using the network
```

HINTS:

- Use a ***single-threaded*** server to avoid critical sections inside the server.
- Since your server will not fork any child processes, you should not worry about handling zombies and can safely ignore the fireman() call in the primer.
- You can safely assume that the input files will always be in the proper format and that the maximum size of the ring is ten processes.

These specifications were written on **Friday, October 5, 2018.** Please refer to the course web site for corrections and updates.