Project #1

Overview

The given project work requires the development of a simple one ready queue and one I/O queue operating system simulation. The simulation requires:

- A process control block (PCB) structure including a Process ID (PID), a CPU usage term (CUT), an I/O request term (IRT), and a waiting term (WT).
- Use of the computer's internal clock to get the time (seconds) values and the system's random number generator.
- The ready queue and I/O queues to be arrays of structures. The array size is left to the student to determine.

Basic functions of the simulator

The program is to behave as follows:

1. Created a new process and assign it a PID (set the CUT, IRT, and WT to zero), and put it into the ready queue (since the ready queue is empty, this PCB will be at the front of the queue).
2. Move the PCB at the front of the ready queue into the CPU. Call the system clock to get the time the process started. To simulate the processing time, call the random number generator to select a value from 0 to 10,000, and executed the following loop:
   for (i = 0; i < value; i++) j=sin(i);
3. When the loop finishes call the system clock to get the time the process stopped (or was forced to stop). Using the start and stop time add the CPU usage time to the CUT value. Use those values to adjust the WT. Use the random number generator to select a value from 0 to 3 to decide what the process needs to do next.
   The options are:
   0 – process terminates and is removed from the system.
   1 – process returns to the ready queue to wait its turn.
   2 – process requires I/O and goes into the I/O queue.
4. If the process terminates – print out the PID, CUT, IRT, and WT values.
5. If the process returns to the ready queue – repeat from step 2 above.
6. If the process goes into an I/O queue call the random number generator to select a value from 0 to 100. Assume that the number returned is the number of seconds that the process had to wait for the I/O to complete. This value will affect the WT value. Then place the PCB back in the ready queue.

The operating system that is simulated is shown in Figure 1 below.

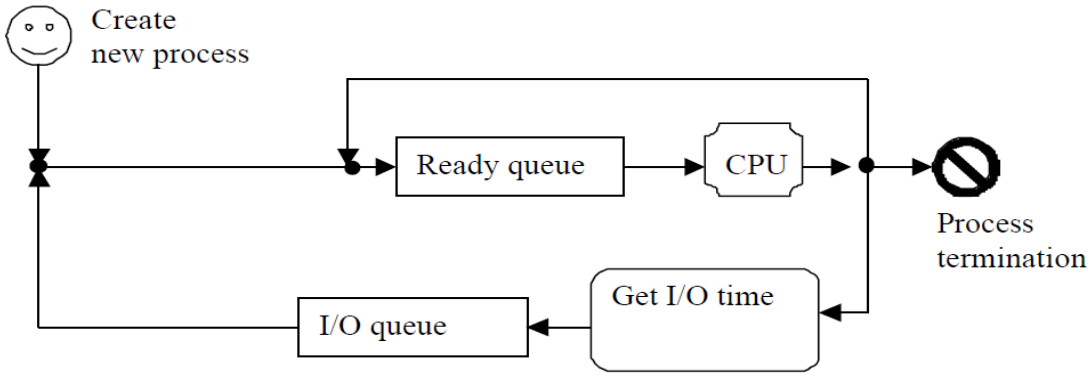

Figure 1: Operating system simulation

Operational details

Occasionally, create a new process and let the simulation run until 25 processes have successfully terminated. This assignment MUST be completed using C or C++ on Linux.