COSC 3360 – 6310 – Operating System Fundamentals Assignment #1 for Fall 2018: The offline scheduler.

Due on Monday, September 24 at 11:59:59 PM

Objective

This assignment will introduce you to the process creation mechanism in UNIX using the fork function.

Specifications

You must write a program that generates an offline scheduling solution based on a priority queue mechanism. After your parent process determines the order of execution of the processes, it will create and execute them based on this order.

Input Format: Your program should read its input from stdin (C++ cin) and use input redirection as in:

assignment1 < input1.txt

This input will look like:

3 // Quantum4 9 // Execution time and Priority3 92 15

- The quantum represents the maximum amount of time that a process can use the processor.
- The execution time is the amount of time (in seconds) needed by the process to complete its execution. This parameter will have a value between 1 and quantum * 2.
- The priority will be used to sort the processes in the ready queue (the higher the value, the lower the priority).
- The parent process will assign a process identifier (PID) to each child process. The PID is a unique integer value that starts in 0 and increments by one per child process in the input file.

After reading the input file and assigning the PID to the child processes, the parent process will generate the scheduling queue that defines the order of execution of the child processes.

Note: If a child process has an execution time greater than the quantum, it will be divided into two processes. The first part of the process will take its normal place in the priority queue, while the second part of the process will be considered as the last element of the processes with the same priority.

Using the example input file presented before, the scheduling queue is:

PID	Execution Time	Priority
0	3	9
1	3	9
0	1	9
2	2	15

Implementation

You will have two types of processes:

- Parent process: is the process that reads the input file, generates the scheduling queue, prints the priority queue, creates the child processes (one at a time), and waits for all the child processes to complete before ending its execution.
- Child processes: these are the processes created by the parent process. Each child process will perform the following operations: prints its information (PID, execution time, priority), sleeps for the number of seconds based on the execution time, and prints that it has completed its execution.

Based on the previous example, the corresponding output is:

Scheduling queue: (0,3,9), (1,3,9), (0,1,9), (2,2,15)

Process 0: exec time = 3, priority = 9 Process 1: exec time = 3, priority = 9

Process 1 ends.

Process 0: exec time = 1, priority = 9

Process 0 ends.

Process 2: exec time = 2, priority = 15

Process 2 ends.

HINTS:

 You can safely assume that the input files will always be in the proper format and that the maximum number of processes is 10.

These specifications were written on **Friday**, **Aug 31**, **2018**. Please refer to the course web site for corrections and updates.