CSE 2431 Homework 1

Instructor: Adam C. Champion, Ph.D.
Due: Thursday, February 1, 2018 (in class)   (40 points)

Please type your homework. Handwriting is not acceptable.

I. Processes (30 points)

(a) This C program creates a child process using the system call fork(). If the child process is created successfully:
   (i) What will be the output? (Assume the created new child process pid = 1000);
   (ii) How many lines will it output? (If your answer is one line, then skip part (iii). Otherwise, continue.);
   (iii) If we want to output the message only once, in the parent process, how should the program be modified?

II. Threads (30 points)

(a) What resources are shared among different threads in a multithreaded process?
(b) What resources are private to each thread in a multithreaded process?
(c) One problem for user-level thread packages is that the invocation of blocked system calls in one thread will block all the threads in the same process. How can we solve it? Describe two different ways.
III. CPU Scheduling (40 points)

<table>
<thead>
<tr>
<th>Process</th>
<th>Burst Time</th>
<th>Priority</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>12</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>P2</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>P3</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>P4</td>
<td>18</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 1: Process Information.

Consider the processes described in Table 1.

Questions: What is the average waiting time of those processes for each of the following scheduling algorithms? (Draw a Gantt chart for each algorithm.)

(a) First Come First Serve (FCFS)
(b) Non-preemptive Shortest Job First (NP-SJF)
(c) Preemptive Shortest Job First (P-SJF)
(d) Priority Scheduling
(e) Round Robin, with the following assumptions:

Assumption (1). The scheduling time quantum is 5 time units.
Assumption (2). If a new process arrives at the same time as the time slice of the executing process expires, the OS puts the executing process in the ready queue, followed by the new process.