CSC415 OPERATING SYSTEM PRINCIPLES

Homework 1

1. What is the difference between symmetric multiprocessing and asymmetric multiprocessing? (10 Points)
   Symmetric multiprocessing runs from the same OS, whereas asymmetric mutiprocessing can run heterogenous or homogenous kernels. Symmetric is easier to implement and work with, but asymmetric can offer better performance in some cases.

2. Why do user programs have to make system calls rather than just executing the code for the system calls themselves? (10 Points)
   Making system calls using the API is much easier than working with the underlying code directly, since the direct code is much more detailed and difficult to work with.

3. What are the four sections in the address space of a process? What is the Process Control Block? (10 Points)
   Text, data, heap, and stack.
   The Process Control Block (PCB) contains specific information about the process, such as its state, a program counter, CPU registers and scheduling info, memory tables, and I/O status info.

4. Assuming there are no errors, how many NEW processes will the following code create? Why? (10 Points)

   ```
   if (!fork()) <- creates 1 child, parent is !n = false but child is !0 = true
           fork(); <- create 1 child from child
   else {  <- parent jumps here from failing if(!fork())
           fork(); <- create 1 child from parent
           fork(); <- create 1 child from parent + 1 child from child
   }
   ```
   1 + 1 + 1 + 1 + 1 = 5

5. Describe the actions taken by a kernel to context-switch between processes? (10 Points)
   While one process is executing, an interrupt or system call occurs that prompts the PCB to save the current process' state, load another process' state from the PCB, and resume execution of the 2nd process. At some other point during execution of the 2nd process, another interrupt or system call occurs that prompts the PCB to save the 2nd process' state, load the 1st process state from the PCB, and resume execution of the 1st process again. When the PCB saves the current process' state, it stop executing and becomes idle until its state is loaded from the PCB and resumed again.

6. What are the two models for Inter-Process Communication? What are their differences? (10 Points)
The two models for Inter-Process Communication (IPC) are: Shared Memory and Message Passing. Shared Memory uses a region of memory and makes it available to multiple processes. Message Passing allows processes to communicate directly to each other via exchanging of messages between them.

7. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds: (40 Points)

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.
   a. Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling.
   b. What is the turnaround time of each process for each of the scheduling algorithms in Part a?
   c. What is the waiting time of each process for each of the scheduling algorithms Part a?
   d. What is the average waiting time for each of the scheduling algorithms in Part a?

a)

FCFS

P1	P2	P3	P4	P5
10	1	2	1	5

SJF

P2	P4	P3	P5	P1
1	1	2	5	10

Nonpreemptive Priority

P2	P5	P1	P3	P4
1	5	10	2	1

RR (quantum = 1)

P1	P2	P3	P4	P5	P1	P3	P5	P1	P5	P1	P5	P1	P5	P1	P1	P1	P1	P1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

b) turnaround time = start time - arrival time (0) + burst time = start time + burst time

FCFS:
P1 = 0 + 10 = **10**
P2 = 10 + 1 = **11**
P3 = 11 + 2 = **13**
P4 = 13 + 1 = **14**
P5 = 14 + 5 = **19**

SJF:
P1 = 9 + 10 = **19**
P2 = 0 + 1 = **1**
P3 = 2 + 2 = **4**
P4 = 1 + 1 = **2**
P5 = 4 + 5 = **9**

Nonpreemptive Priority:
P1 = 6 + 10 = **16**
P2 = 0 + 1 = **1**
P3 = 16 + 2 = **18**
P4 = 18 + 1 = **19**
P5 = 1 + 5 = **6**

RR, quantum = 1: burst + (wait)
P1 = 10 + (4 + 2 + 1 + 1 + 1) = **19**
P2 = 1 + (1) = **2**
P3 = 2 + (2 + 3) = **7**
P4 = 1 + (3) = **4**
P5 = 5 + (4 + 2 + 1 + 1 + 1) = **14**

c) wait time = start time - arrival time (0) = start time
FCFS:
P1 = **0**
P2 = **10**
P3 = **11**
P4 = **13**
P5 = **14**

SJF:
P1 = **9**
P2 = **0**
P3 = **2**
P4 = **1**
P5 = **4**

Nonpreemptive Priority:
P1 = **6**

P2 = **0**
P3 = **16**
P4 = **18**
P5 = **1**

RR, quantum = 1:
P1 = 4 + 2 + 1 + 1 + 1 = **9**
P2 = **1**
P3 = 2 + 3 = **5**
P4 = **3**
P5 = 4 + 2 + 1 + 1 + 1 = **9**

d) avg wait time = SUM_wait(P1:P5) / 5
FCFS: (0 + 10 + 11 + 13 + 14) / 5 = **9.6**
SJF: (9 + 0 + 2 + 1 + 4) / 5 = **3.2**
Nonpreemptive Priority: (6 + 0 + 16 + 18 + 1) / 5 = **8.2**
RR, quantum = 1: (9 + 1 + 5 + 3 + 9) / 5 = **5.4**