COMP3230 Principles of Operating Systems

Problem Set #1

Due date: 23:59 pm, October 8, 2023

Total 5 points

(version: 1.0)

Note: This assignment weights a total of 5 points in the final marks of this course. However, just like each process has its own private virtual address space, this assignment also has its own virtual score space of 100 points, which is mapped to 5 points of physical marks. The below points for each question are given in the said virtual score space.

Question 1 – Process (25 points)

(This question is related to ILO2a – “explain how OS manages processes/threads”.)

Consider an operating system that runs on a single-core CPU and supports four process states: Ready, Running, Blocked, and Terminated.

(1) A process may transit from one state to another state. For each valid state transition, describe two different system/process event(s)/condition(s) that trigger this state transition. Organize your answer by listing all valid state transitions and provide description of the event(s) for each state transition.

(2) Consider a parent process that forks a child process during running. Since the child process inherits most properties from the parent process, which is currently running, the child process will also be in the Running state. Is the statement true? Why or why not?

(3) Now suppose we consider an operating system that does not support time-sharing, i.e., it runs each process without any interruption until it terminates. What process states do we still need? And how does your answer to (1) change?

Question 2 – Context Switch and Mode Switch (30 points)

(This question is related to ILO2 – "explain the principles behind the core function: Process Management", and ***ILO 4*** – "demonstrate knowledge in applying system software and tools".)

Consider an operating system running on a single-core CPU. The CPU has a clock speed of 2.5 GHz (i.e., 2.5x10^9 cycles per second). Each context switch takes 2500 cycles to complete, and each mode switch (transition between user mode and kernel mode) takes 500 cycles to complete. Assume an FIFO scheduling policy, and consider the following processes (ordered by their arrival time) with their respective number of system calls they make. Also assume the system calls do not include fork() and the processes do not involve I/O events, unless otherwise specified.

Process A: Arrival Time = 0 ms, # of System Calls = 10, including one I/O event with Wait Time = 20 ms

Process B: Arrival Time = 10 ms, # of System Calls = 5 (including the fork() system call for creating a child process D, and the wait() call by process B to wait for the child)

Process C: Arrival Time = 10 ms, # of System Calls = 20

(1) Calculate the total time spent on context switches for the above scenario and that on mode switches for each process (except Process D). Ignore the time to load Process A.
(2) How will your answer change if the I/O Wait Time for Process A becomes 5 ms? Discuss all possibilities.

Question 3 - Process Scheduling (45 points)

(This question is related to ***ILO 2a*** - "discuss the mechanisms and policies in efficiently sharing of CPU resources" and ***ILO 3*** – "analyze and evaluate the algorithm and explain the performance issues".)

Consider a preemptive scheduling scheme which selects a process that has consumed ***the least amount of CPU time since its arrival*** as the selection policy. Like the RR policy, a process is preempted when its time quantum expires and is placed back to the ready queue. Processes in the ready queue are arranged in ascending order of elapsed time. Such that the process that consumes the least amount of CPU time is at the head of the queue and will be dispatched next. If two processes have the same amount of elapsed time, they are ordered by their arrival time.
Below is an example system workload.

Process	**Arrival Time**	**CPU Time**
A	0	85
B	10	50
C	25	45
D	40	65
E	65	70
F	80	40

a) Apply the above scheduling scheme to these processes and draw the timeline (Gantt chart) diagram. Suppose the system uses a time quantum of 20-time units and has negligible context switch overhead. Assume that new processes always be inserted to the queue just before the arrival time, e.g., at t=25, C has been added to the end of the ready queue. Find the average turnaround time, waiting time, and response time of this scheme.

b) Apply the RR scheme with a quantum of 20-time units and the SJF scheme to the same workload. Again, assume negligible context switch overhead. Find the average turnaround time, waiting time, and response time of the RR and SJF schemes.

c) Comment on the performance of this new preemptive scheme.