CS 370: Operating Systems
Department of Computer Science
Colorado State University

FALL 2016
URL: http://www.cs.colostate.edu/~cs370
INSTRUCTOR: Yashwant K Malaiya

HW1

DUE DATE: Tuesday October 4 2016 5 PM. No late submissions accepted, since we want to share the solution. For problems 3 and 4, wait until they are covered in the class. Submit PDF file using Canvas.

Problem 1 (4+4+4+8 =20 points) Specify true or false, or fill in the blank

a. Multiprogramming is having multiple processors execute different programs at the same time. True __ **False** __
b. For a process, the following is a valid transition: Ready → Blocked True __ **False** __
c. Process creation using fork() results in a copy of the parent's memory image. Assume that a new program is not loaded into the child using the exec() command. After the fork( ) operation, when a variable is changed in the parent process, this change is reflected in the child process. True __ **False** __
d. Consider the code below. Specify the two answers.

```
main(int argc, char **argv) {
      int i;
      for (i=0; i < 3; i++) {
            fork();
printf("hello\n");
       }
}
```

a. What will be the total number of processes? Answer: **8**

(Hint complete the diagram below. Label each process with a number (just for counting. It is not necessarily the PID).

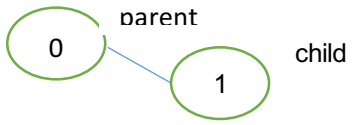

Diagram not graded.

b. How many times "hello" message will be printed? Answer: = **14**

Briefly mention why. *Not graded.*

Problem 2 (20 points) We had seen the exponential average expression used to predict the length of the next CPU burst. What are the implications of assigning the following values to the parameters used by the algorithm?

1. $\alpha = 0$ and $\tau 0 = 20$ milliseconds
2. $\alpha = 0.99$ and $\tau 0 = 10$ milliseconds

(Hint: Use the values in the equation, and comment on the implications for the value of $\boldsymbol{\tau_{n+1}}$)

Answer:

a) $\boldsymbol{\tau_{n+1} = \alpha * t_n + (1-\alpha) * \tau_n}$

$\tau_1 = 0 * t_0 + 1 * \tau_0 = 20$ milliseconds

$\tau_n = 0 * t_{n-1} + 1 * \tau_n = \tau_{n-1} = \tau_{n-1} \ldots = \tau_1 = \tau_0 = 20$ milliseconds

It means the prediction of burst time is always 20 milliseconds.

b) $\boldsymbol{\tau_{n+1} = \alpha * t_n + (1-\alpha) * \tau_n}$

$\tau_1 = 0.99 * t_0 + 0.01 * \tau_0 = = 0.99 * t_0 + 0.1$ milliseconds

$\tau_n = 0.99 * t_{n-1} + 0.01 * \tau_n = 0.99 * t_n + 0.01\ \tau_n$ milliseconds

It means the prediction of burst time mainly depend on the most recent process burst time.

Problem 3 (20 points) A system is running ten I/O-bound tasks and one CPU-bound task. The I/O-bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context switching overhead is 0.1 millisecond (which does not count as the CPU being utilized) and that all processes are long-running tasks. What is the **CPU utilization** for a round-robin scheduler when:

1. The time quantum is 1 millisecond
2. The time quantum is 10 milliseconds

Answer:

1.

I/O-bound tasks and CPU-bound task both execute 1 millisecond and then switch to each other. The only idle time is context switch time.

CPU utilization = 1 * 11 / (1 * 11 + 0.1 * 11) = 91%

2.

I/O-bound tasks execute 1 millisecond and then switch to each other. The CPU-bound task executes 10 milliseconds and then switches. The only idle time is context switch time.

CPU utilization = 1 * 10 + 10 / (1 * 10 + 10 + 0.1 * 11) = 94%

CS 370: Operating Systems
Department of Computer Science
Colorado State University

FALL 2016
URL: http://www.cs.colostate.edu/~cs370
INSTRUCTOR: Yashwant K Malaiya

Problem 4 (40 points) The processes in the table below are being scheduled using round-robin scheduling algorithm with preemption. Each process is has a numerical priority, with a higher number indicating a higher priority. In addition to the processes listed below, the system also has an idle task (which consumes no CPU resources and is identified as P_{idle}). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue. (A process is not preempted by a process with the same priority, unless the time slice expires. A process cannot be preempted by a lower priority process, even if the time slice expires.)

a. Show the scheduling order of the processes using a Gantt chart.
b. What is the turnaround time for each process?
c. What is the total waiting time for each process (including initial and in the Ready Queue)?
d. What is the CPU utilization rate?

Thread	Priority	Burst	Arrival
P_1	40	20	0
P_2	30	25	25
P_3	30	25	30
P_4	35	15	60
P_5	5	10	100
P_6	10	10	105

Answer: