University of California Santa Cruz
Baskin School of Engineering
Computer Science and Engineering Department

CSE130 Winter 2020

Assignment 3

Marks Available: **50 (5% of final course mark)**

Submission:
Due: **23:59 Sunday February 16, 2018**
Format: **Single PDF Document**
Where**:** **Canvas**

(3 marks) *Question 1.* In the following piece of C code, how many processes are created when it is executed? Explain your answer and include diagrams as appropriate.

```c
int main() {
    fork();
    fork();
    exit(1);
}
```

(3 marks) *Question 2.* Describe how a web server might leverage multi-threading to improve performance. Include diagrams if you feel this will make your answer clearer.

(3 Marks) *Question 3.* Briefly outline the role of the Process Control Block (PCB), listing and describing three pieces of information an Operating System might choose to store in the PCB.

(3 Marks) *Question 4.* Identify the critical section and show how the following pseudo code could be modified to avoid deadlock. Explain your answer.

Shared Variables: lockA, lockB, resourceA, resourceB

```
Process1 {
  aquire(lockA);
  aquire(lockB);
  modify(resourceA);
  modify(resourceB);
  release(lockB);
  release(lockA);
}
```

```
Process2 {
  aquire(lockB);
  aquire(lockA);
  modify(resourceB);
  modify(resourceA);
  release(lockA);
  release(lockB);
}
```

(4 marks) *Question 5.* Define the terms “race condition”, “deadlock”, and “starvation” as they relate to Operating System design and outline the relationship between deadlock and starvation.

(4 marks) *Question 6.* (a) In a multiprogrammed environment with 16MB of memory where all processes require 1MB of unshared memory and spend 60% of their time in I/O wait, calculate how much memory will remain unused when approximately 99% CPU utilization is achieved. (b) In the same multiprogrammed environment, if each process now requires 3MB of unshared memory, calculate the maximum achievable CPU utilization. Show all your working.

(4 marks) *Question 7.* If we assume that when processes are interrupted they are placed in a queue containing all non-running processes not waiting for an I/O operation to complete, briefly describe two strategies the Operating System might adopt to service that queue. One-word answers will not suffice.

(4 marks) *Question 8.* Can the "priority inversion" problem outlined in section (3) of the background information to Lab 2 occur if user-level threads are used instead of kernel-level threads? Explain your answer.

(4 Marks) *Question 9.* List and describe the necessary conditions for deadlock.

(4 marks) *Question 10.* Explain how quantum value and the time taken to perform a context switch affect each other in a round robin process-scheduling algorithm.

(6 marks) *Question 11.* Describe a mechanism by which counting semaphores could be implemented using the minimal number of binary semaphores and ordinary machine instructions. Include C code snippets if you feel this will make your answer clearer and/or more concise.

(8 marks) *Question 12.* Five threads, A through E, arrive in alphabetic order at a scheduling queue one second apart from each other. Estimated running times are 10, 6, 2, 4, and 8 seconds, respectively. Their externally determined priorities are 3, 5, 2, 1, and 4, respectively, 5 being the highest priority. For each of the following scheduling algorithms, determine the mean turnaround time and mean waiting time. Assume thread switching is effectively instantaneous.

(a) First Come First Served
(b) Round Robin
(c) Preemptive Priority Scheduling
(d) Preemptive Shortest Job First

For (b), assume the system is multi-programmed with a quantum of 4 seconds. In all cases, show your work and include diagrams/charts/tables as appropriate.