Scheduling (25 pts)

Answer the following questions from the Operating Systems Concepts (OSC) 9/e, textbook. You only need a few sentences to answer each question.

OSC 6.2: Explain the difference between preemptive and nonpreemptive scheduling.

OSC 6.6: Suppose that a scheduling algorithm (at the level of short-term CPU scheduling) favors those processes that have used the least processor time in the recent past. Why will this algorithm favor I/O-bound programs and yet not permanently starve CPU-bound programs?

OSC 6.11: Discuss how the following pairs of scheduling criteria conflict in certain settings:

a. CPU utilization and response time
b. Average turnaround time and maximum waiting time
c. I/O device utilization and CPU utilization

Synchronization and Deadlocks (50 points)

Answer the following questions from the Operating Systems Concepts (OSC) 9/e, textbook. You only need a few sentences to answer each question.

OSC 5.5: Show that, if the wait() and signal() semaphore operations are not executed atomically, then mutual exclusion may be violated.

OSC 5.8: The first known correct software solution to the critical-section problem for two processes was developed by Dekker. The two processes, P0 and P1, share the following variables:
boolean flag[2]; /* initially false */
int turn;

The structure of process Pi (i == 0 or 1) is shown below:

```java
do {
    flag[i] = true;
    while (flag[j]) {
        if (turn == j) {
            flag[i] = false;
            while (turn == j)
                ; /* do nothing */
            flag[i] = true;
        }
    }
    /* critical section */
    turn = j;
    flag[i] = false;
    /* remainder section */
} while (true);
```

The other process is Pj (j == 1 or 0). Prove that the algorithm satisfies all three requirements for the critical-section problem.

**OSC 7.16:** In a real computer system, neither the resources available nor the demands of processes for resources are consistent over long periods (months). Resources break or are replaced, new processes come and go, and new resources are bought and added to the system. If deadlock is controlled by the banker’s algorithm, which of the following changes can be made safely (without introducing the possibility of deadlock), and under what circumstances?

a. Increase *Available* (new resources added).

b. Decrease *Available* (resource permanently removed from system).

c. Increase *Max* for one process (the process needs or wants more resources than allowed).

d. Decrease *Max* for one process (the process decides it does not need that many resources).

e. Increase the number of processes.

f. Decrease the number of processes.

**OSC 7.19:** Consider the version of the dining-philosophers problem in which the chopsticks are placed at the center of the table and any two of them can be used by a philosopher. Assume that requests for chopsticks are made one at a time. Describe a simple rule for determining whether a particular request can be satisfied without causing deadlock given the current allocation of chopsticks to philosophers.
Memory Management (25pts)

Answer the following questions from the *Operating Systems Concepts (OSC)* 9/e, textbook. You only need a few sentences to answer each question.

**OSC 8.11:** Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)? Rank the algorithms in terms of how efficiently they use memory.

**OSC 8.13:** Compare the memory organization schemes of contiguous memory allocation, pure segmentation, and pure paging with respect to the following issues:

a. External fragmentation

b. Internal fragmentation

c. Ability to share code across processes.

**9.21:** Consider the following page reference string:

```
7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1
```

Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms?

- LRU replacement
- FIFO replacement
- optimal replacement

**Note:** Like all assignments in this class you are prohibited from copying any content from the Internet or sharing ideas, code, configuration, text or anything else or getting help from anyone in or outside of the class, except where noted. Consulting online sources is acceptable, but under no circumstances should anything be copied. Failure to abide by this requirement will result in sanctions ranging from zero on the assignment to dismissal from the class.