CSE 332 Final

Olcay Taner YILDIZ

I. Question (16 points)

Consider the following snapshot of a system, which has four types of resources A, B, C and D and five processes P_1, P_2, P_3, P_4 and P_5.

	Allocation	Max	Available
	A B C D	A B C D	A B C D
P1	1 0 0 0	2 6 5 0	1 0 2 2
P2	1 3 4 4	2 3 7 6	
P3	0 2 3 2	0 4 5 2	
P4	0 1 1 4	0 4 5 6	
P5	0 0 1 1	1 0 1 2	

Answer the following questions using the Bankers algorithm:

- Is the system in a safe state? If so, provide a safe sequence. If not, justify your answer.
- If a request from process P1 arrives for (1,0,1,0), should the request be immediately granted?

II. Question (12 points)

Given two processes, (P_1, P_2) and four statements (S_1, S_2, S_3, S_4), where P_1 will execute S_1 and S_2, and P_2 will execute S_3 and S_4. Use semaphores to make the order of execution S_2, S_3, S_1, S_4. Use at most four semaphores. Show the initial semaphore values.

III. Question (15 points)

The first known correct software solution to the critical-section problem for two processes was developed by Dekker. The two processes, P_0 and P_1, share the following variables, boolean flag[2] and int turn. The structure of process P_i (i == 0 or 1) is shown below; the other process is P_j (j == 1 or 0). Prove that the algorithm satisfies all three requirements for the critical section problem.

```
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);
```

IV. Question (14 points)

Draw the resource allocation graph for the following situation. There are three processes P_1, P_2, and P_3. There are four resources R_1, R_2, R_3, and R_4. Resources R_2, R_1, R_4 and R_3 are allocated to processes P_1, P_2, P_2 and P_3 respectively. Processes P_1 and P_2 request resources R_1 and R_3 respectively. Is there a deadlock? If not, suggest a single change in the system, so that the resulting system is in deadlock state.

V. Question (12 points)

- A computer with a 32bit addess uses a twolevel page table. Virtual addresses are split into a 9bit toplevel page table field, and 11bit secondlevel page table field, and an offset. How large are the pages? How many pages are there in the virtual address space?
- A computer whose processes have 1024 pages in their address space keeps its page tables in memory. The overhead required for reading a word from the page table is 500 nsec. To reduce this overhead, the computer has an associative memory, which holds 32 (virtual page, physical page frame) pairs, and can do a look up in 100 nsec. We say that there is a cache hit if a page table entry needed to handle a reference to a virtual page is in the associative memory. What hit rate is needed to reduce the mean overhead to 200 nsec?

VI. Question (15 points)

Consider the following page reference string:

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

How many page faults occur for the replacement algorithms LRU, FIFO and Optimal, assuming there are 3 page frames? Assume that all the frames are initially empty.

VII. Question (12 points)

Consider the two-dimensional array A:

```
int A[][] = new int[100][100];
```

where A[0][0] is at location 200 in a paged memory system with pages of size 200. For three page frames, how many page faults are generated by the following array-initialization loops?

```
a. for (j = 0; j < 100; j++)
     for (i = 0; i < 100; i++)
       A[i][j] = 0;
b. for (i = 0; i < 100; i++)
     for (j = 0; j < 100; j++)
       A[i][j] = 0;
```