United International University (UIU)

Dept. of Computer Science & Engineering

Final Examination: Trimester: Spring - 2017

Course: CSI 309

Course Title: Operating System Concepts

Total Marks: 40, Time: 2 hours

There are five questions. Answer **ANY FOUR.** Figures in the right-hand margin indicate full marks.

1. (a) What do you understand by context switch? Write one disadvantage of user level thread. [2+1=3]

(b) Suppose, a word processor has three threads: one for I/O, one for formatting and background saving, one for checking spellings. What would happen if it had only one thread for all these jobs? [2]

(c) What is meant by critical region? To protect critical region, lock variable is used in the following code:

```
int lock = 0;
while (lock);
lock = 1;
//EnterCriticalSection;
  access shared variable;
//LeaveCriticalSection;
lock = 0;
```

Does this code work? Explain. [1+2=3]

(d) What is Priority Inversion Problem? Explain. [2]

2. (a) A fast food restaurant has four kinds of employees: (i) order takers, who take customers' orders; (ii) cooks, who prepare the food; (iii) packaging specialists, who stuff the food into bags; and (iv) cashiers, who give the bags to customers and take their money. Each employee can be regarded as a process. Write code to maintain the synchronization among these processes. [3]

(b) Consider the dining philosophers' problem. A simple proposed solution is, "after taking left fork, check whether right fork is available. If not, then return left one, wait for some time and repeat again." What is the problem with this solution? Explain. [2]

(c) Consider an ATM cubicle with 4 ATM booths. Use semaphore to make sure that at most 4 people can access the 4 booths simultaneously. [3]

(d) What does the following instruction do? [2]

TSL REGISTER, LOCK

3. (a) Define deadlock. Write briefly the four conditions for deadlock. [1+2]

(b) Suppose there are 3 processes P1, P2, P3 and 3 resources R1, R2, R3. Now consider the following sequence of statements: [1.5+1.5=3]

i. P1 requests R1
ii. P1 requests R2
iii. P2 requests R2
iv. P3 requests R3

Now draw resource allocation graph after the above four statements have been executed.

After that, the following two statements have been executed. Now draw another resource allocation graph after these two statements have been executed after the above four.

v. P1 releases R2
vi. P2 requests R3

(c) Find if the state is safe using Banker's algorithm. [4]

Resource assigned	**Resource total needed**
[3 0 1 1	[4 1 1 1
0 1 0 0	0 2 1 2
1 1 1 0	4 2 1 0
1 1 0 1	1 1 1 1
0 0 0 0]	2 1 1 0]

Exist, E = (6 3 4 2)
Possessed, P = (5 3 2 2)

4. (a) Write down the principle of Ostrich algorithm. [1]

(b) Draw the file system layout. What is inode? [1+1=2]

(c) Suppose, you have a disk of size 10 MB where each block is of size 1 MB. Now you want to create 3 files: A (2.5 MB), then B (3.6 MB), then C (2 MB). Now show the disk block allocations using the following schemes: [1+2+1=4]

(i) Contiguous Allocation
(ii) Linked List Allocation
(iii) Linked list allocation using a table in memory

(d) Suppose there are two files named 'astronomy' and 'biology' in a directory. Draw the directory table showing the directory entries for these two files using: [3]

(i) IN-LINE file naming , (ii) IN-HEAP heap file naming

5. (a) What is symbolic linking? Suppose, “homework.txt” is a shared file which is owned by user A and A shares this file with user B via symbolic linking. Now after they have done with the homework, owner A removes the file. Is the file actually deleted from the file system? Will any attempt to read the file by user B succeed? [1+1+1=3]

(b) Consider a buffer pool where hashing is done by “mod 3”. The existing condition of the buffer pool is:

Queue 0: 3, 18, 12
Queue 1: 4, 16, 10
Queue 2: 17, 20, 11

Here free buffers are: 16 (delayed write), 10, 18.

Now describe what happens and draw the buffer pool after each of the following statements are executed (each time refer to the original given scenario): [2+2=4]

(i) getblk(18)
(ii) getblk(0)

(c) In getblk() function, when the kernel cannot find the buffer on the hash queue, and the free list is empty, the process goes to sleep. When it is awakened, why is it necessary to search for that block again in the hash queue? Explain with an example. [3]