NORTHEASTERN UNIVERSITY
COLLEGE OF ENGINEERING
Department of Electrical & Computer Engineering

93

EECE7376: Operating Systems: Interface and Implementation — **Dr. Emad Aboelela**
Spring 2024 – Quiz 4 — **Time Allowed: 1.5 Hour**

Student Name: Student ID:

This is a **closed book** and **closed notes** quiz. The quiz has **6 questions**
You **are not allowed** to use any **electronic** device nor **scratch** papers
No answers outside these quiz pages will be graded (you can write on **both sides**)
Make sure to **write your full name** on the empty side of your cheat sheet
You cannot leave the room once the quiz starts (except in case of an emergency)

Q1. (16 Points)

16

By only using **semaphores**, modify the shown code to perform the following two tasks:

a) Avoid data race between threads p1 and p2.
b) Replace both pthread_join() functions.

Assume all needed header files are already included.

```
int counter = 100;

void* decrement(void* arg) {
      counter--;
      return NULL;
}

int main(int argc, char* argv[]) {
  pthread_t p1, p2;

  pthread_create(&p1, NULL, decrement, NULL);
  pthread_create(&p2, NULL, decrement, NULL);

'n  pthread_join(p1, NULL);
  pthread_join(p2, NULL);

  printf("%d\n", counter);

  return 0;
}
```

m_t mutex, don

//mutex
//do

}

Q2. (18 Points)

18

The following table represents the current state of a system in which four resources *A*, *B*, *C* and *D* are needed by four processes P_1, P_2, P_3, and P_4. The system contains the following total instances of each resource: 5 of *A*, 4 of *B*, 3 of *C*, 2 of *D*.

Processes	Allocation				Max				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P_1	2	0	1	1	3	2	1	1	1	2	0	0	1	2	1	0
P_2	1	1	0	0	1	2	0	2	0	1	0	2				
P_3	1	0	1	0	3	2	1	0	2	2	0	0				
P_4	0	1	0	1	2	1	0	1	2	0	0	0				

Answer the following questions using the Banker's algorithm:

✓a) Fill the missing entries in columns *Need* and *Available* in the above table.

✓b) Is the system in a safe state? Explain why by showing all numbers that support your answer.

✓ c) Can a request from P_3 of (1,2,0,0) be granted while ensuring that the system will not be in an unsafe state? Explain why by showing all numbers that support your answer.

Processes	Allocation				Max				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P_1	2	0	1	1	3	2	1	1	1	2	0	0	0	0	1	0
P_2	1	1	0	0	1	2	0	2	0	1	0	2				
P_3	~~1~~ 2	~~0~~ 2	1	0	3	2	1	0	1	0	0	0				
P_4	0	1	0	1	2	1	0	1	2	0	0	0				

Q3. (15 Points)

The following table contains incomplete directory data. After adding the needed **null characters** to each entry **name** and based on the file system we studied in class, answer the following questions:

13

a) Complete the directory data by filling the spaces with the missing values of the **rec_len** and **name_len** fields for each entry.

4By

inode	rec_len	name_len	file_type	name
6			2	.
2			2	..
19			1	abcd
21			1	uvwxyz
32			1	hello

b) What will be the changes in the above data when file **uvwxyz** is deleted?

Q4. (22 Points)

22 a) Assume an existing file **/DR1/FL1.txt** has been opened for **writing** and it has only one full data block associated with it. A system call **write()** is called on the **FL1.txt** file to write an extra **block** of data. X open()

Only during that write() call, indicate in the following table whether each of the listed disk data structure will be written to or not. Explain your answer in the third column of the table.

Disk Data Structure	**Will be Written to? Yes/No**	**If Yes, what will be written? If No, why not?**
data bitmap		
inode bitmap		
root inode		
DR1 inode		
FL1 inode		
root data		
DR1 data		[illegible]
FL1 data		

b) If later we need to open the file **`/DR1/FL1.txt`** for reading, show all the steps the file system (FS) follows until it reads the first data block of this text file.
Dedicate each step to one of the needed disk data structures by explaining what will be read/written from/to the data structure.

Q5. (15 Points)

14

Answer the following questions:

a) In the file allocation table (**FAT**) approach explained in the lectures, how does the file system locate the last block of a file? -1

(empty)

b) ✓ In the index nodes (**inodes**) approach explained in the lectures, how does the file system locate the last block of a file?

c) ✓ How many inodes are needed for two files that are **hard** linked together? Explain your answer.

d) ✓ How many inodes are needed for two files that are **symbolic** linked together? Explain your answer.

e) ✓ Why there is a **dangling symbolic link** problem while there is no such problem with hard link?

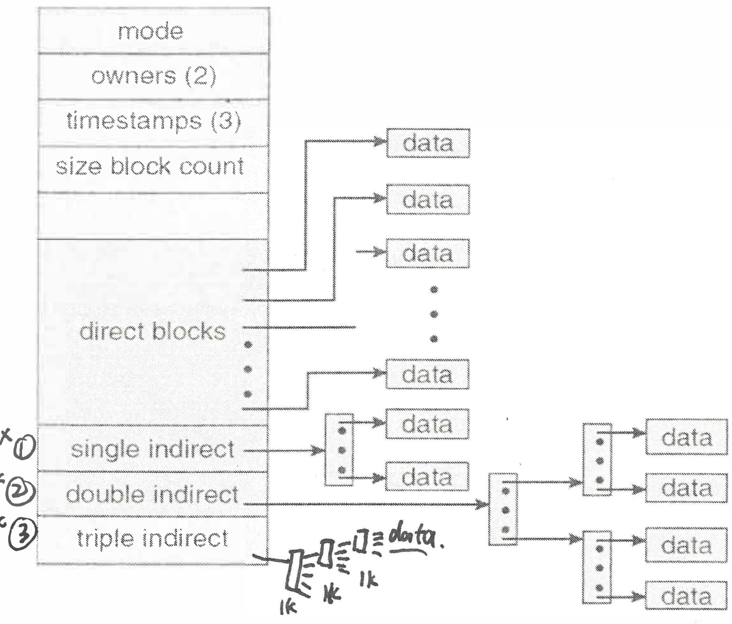
Q6. (14 Points)

10

The inode in the file system we studied has a total of 15 data block pointers. For each data block of size 4KB, the size of the block pointer (address) is 4 Bytes.

In the shown figure, one block pointer is used for **single indirect pointers**, another block is used for **double indirect pointers**, and a third block is used for **triple indirect pointers**.

What is the total size of the **data** that can be pointed to by the **triple indirect pointers** block? And what is the size of the total **overhead** space needed with this **triple indirect pointers** method? Show the details of your calculations. *Note:* 1KB = 1024 Bytes.



End of the Quiz Questions