University of Minnesota
Department of Computer Science
CSci 5103 - Fall 2016 (Instructor: Tripathi)
Final Exam — Date: December 19, 2016 (Time 4:00–6:00 am)
(Time: 120 minutes) Total Points – 100
CLOSED BOOK/CLOSED NOTES

STUDENT NAME:
STUDENT ID:

Prob. 1	Prob. 2	Prob. 3	Prob. 4	Prob. 5	Prob. 6	Prob. 7	Prob. 8	Total
32	10	10	10	12	6	10	10	100

Problem 1 (32 points): Eight questions, each of 4-points.

A (4 points) In many of the modern shared-memory multiprocessor systems, processes contending for a critical section are kept spinning (i.e. busy waiting). What is the justification for such an approach? Under what assumptions is this valid?

B (4 points) In most operating systems, the priority of a process may be temporarily boosted or lowered. For both these cases, identify the conditions under which this is done.

C (4 points) In FAT-16 file system using 16KB file-block size, how large can be the disk partition for this file system? How much memory is used for storing the *file allocation table* in memory?

D (4 points) Name one advantage of hard links over symbolic links and one advantage of symbolic links over hard links.

E (4 points) For the following page-reference string for a program, give the *distance string* when LRU replacement policy is used.

1 2 3 4 1 2 5 1 2 3 4 5

Using this distance string determine the number of page-faults this program would encounter when given (a) 3 frames, (b) 4 frames.

F (4 points) In Linux, what is the purpose of *vm_area_struct* structures? What information is typically contained in such a structure? Is there only one or several such structures associated with a process (task)?

G (4 points) The procedure for incremental backup dumps in a file system saves in the archival storage all files that have been modified since the last incremental dump. It also saves all directories which are on the path from the root to a modified file. What is the reason for saving all the directories, even though some of them are not modified since the last incremental dump?

H (4 points) What is the use of the SetUID mechanism in UNIX?

Problem 2 (10 points) Consider a system that contains two types of resources. There are three instances of type R_1 and two instances of type R_2. There are two processes P_1 and P_2 in the system. The maximum resource requirement of P_1 is 1 for R_1 and 1 for R_2, and that of P_2 is 3 for R_1 and 2 for R_2.

Identify all possible deadlock states for this system.

Problem 3 (10 points):

Part A: (6 points) Identify at least six items of information contained in the inode of a Unix file.

Part B: (4 points) Identify which fields of the inode of a given file would change in each of the following cases:

1. A new hard-link is created for the file
2. File access permissions are changed
3. File opened for reading
4. Data is appended to the file

Problem 4: (10 points) In UNIX, the file system maintains a cache of file-blocks. The file-blocks read from the disk are put in this cache. Also, the blocks to be written back to the disk are first buffered in this cache and then flushed out to the disk later.

[A] Discuss the pros and cons of this approach.

[B] It is possible to use perfect LRU based approach for managing the file-block cache. However, that raises several issues. What are these issues? Briefly outline the approach taken in UNIX to address these issues.

Problem 5 (12 points):

A (6 points) Briefly discuss the relative advantages and disadvantages of capability based and access list based protection schemes.

B (6 points) Briefly outline describe at least three methods that can be used for revoking privileges in a capability based protection system.

continue problem 6

Problem 6 (6 points): The access privileges of four processes P1 through P4 for files F1 through F5 are shown in the following access control matrix.

(a) (3 points) Can the information contained in file F1 leak to process P4? If yes, show one possible path for information flow, otherwise justify your answer.

	F1	F2	F3	F4	F5
P1			read		
P2		read	write	read	
P3	read	write			
P4					read

(b) (3 points) The above access matrix is now modified to one, shown below. Can the information contained in file F1 leak to process P4?

	F1	F2	F3	F4	F5
P1			read	write	
P2		read	write	read	write
P3	read	write			
P4					read

Problem 7 (10 points): Consider a 6000-rpm disk system with 160 sectors of 512 bytes per track, and 20 tracks per cylinder. Suppose that this disk has 7000 cylinders, the seek time to an adjacent cylinder takes 1 msec. Assume that the reading of sector data and transferring to main memory can occur in parallel.

A (4 points) How much cylinder skew is needed?

B (6 points) How much time would it take to read the entire disk sequentially, starting with sector 0 which is on the inner most cylinder? Assume skew as determined in the first part above. What would be the effective data transfer rate in this case?

Problem 8: (10 points) The virtual memory space used by a program has 5K words. This program is executed on two different machines, one with 1K word size pages and the other with 512 words size pages. Thus in the first system the program's address space has five pages and in the second system it has 10 pages. Each logical page in the first system splits into two logical pages in the second system. For example page number 1 in the first system splits into two pages numbered $1'$ and 1” in the second system, page 2 splits into $2'$ and 2”, and similarly for the other pages.

In both systems, the program is given 4K words of physical memory. The memory reference strings generated by the program in the two systems have the following form (these two are equivalent in terms of accessing locations in the virtual address space):

System I: (1 2 3 4 5 5 4 3 2 1)^N where N is a large integer

System II: (1’ 2’ 3’ 4’ 5’ 5” 4” 3” 2” 1”)^N

Which system will finish first? How do the execution times for this program on these two systems compare? Justify your answer in terms of number of page faults.

Continue problem 8 here...