University of Minnesota
Department of Computer Science
CSci 5103 - Spring 2018 (Instructor: Tripathi)
Final Exam — Date: May 8, 2018 (Time 10:30 am–12:30 pm)
(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	Total
16	10	20	30	12	12	100

Problem 1 (16 points):

Part A (3 points): The *distance string* for a program's page references is defined as follows. Suppose that we maintain a stack containing all pages referenced by a process during its execution. Every time a page is referenced, it is put at the top of the stack. Thus the most recently accessed page is at the top. Now we transform a reference string to the corresponding distance string as follows. For a page number in the reference string, in the distance string put the distance of that page from the top of the stack, counting the topmost page at distance of 1. If a page is not on the stack, that means it is being referenced for the first time, put distance as ∞.

Example of a Reference String and the corresponding Distance String:

Reference String:	0	2	1	3	4	3	5	4	5	3	2
Distance String:	∞	∞	∞	∞	∞	2	∞	3	2	3	5

Question: Based on the distance string for a program execution, give the rule for computing the number of page-faults this program would encounter on a system with M page frames and using the LRU replacement policy.

Part B (5 points): Suppose that in the distance string of a program the probability density function for the values in the distance is given by the function $P[distance = k] = f(k) = (1/2)^k$ for k = 1, 2, .., ∞. This probability density gives the relative frequency with which the value k occurs in the distance string. Assume that the LRU replacement policy is used in a system executing this program. How many page frames should be given to a process executing this program so that the probability of page-fault is less than 10^{-3}?

Part C (4 points) In managing shared pages among processes in a virtual memory system, what issues need to be considered when a process is terminated, or when swapping a process in or out?

Part D (4 points) What the pros and cons of using large-page sizes in virtual memory management? (Identify two advantages and two disadvantages of using large page size.)

Problem 2 (10 points):

1. **(5 points)** Identify the steps involved and the corresponding disk I/O operations that are performed by the file system to open the following file:

   /usr/5103/assignments/solutions/problem1

   Assume that the inode for the root directory is in memory, but nothing else along the path is in memory. Assume that all directory entries fit in one disk block.

2. **(5 points)** In a Unix environment explain what happens in the following three cases involving a file with hardlink count equal to 1. Explain the behavior stated in your answer in terms of the Unix file system implementation.

   (a) (2 points) A user's program opens in the write mode a file which is owned by another user. Just a few seconds later the owner changes the protection mode to revoke the write permission for the user, and makes it a read-only file. Will the user be able to continue modifying the file?
   (b) (2 points) The user program now forks a child. Will the child process be able to modify the file?
   (c) (1 points) A few minutes later, the owner deletes the file. Will the user program be able to continue accessing the file? What happens when the user program and its child terminate?

Problem 3 (20 points): This question has two parts.

Part A (15 points) Consider a system with two producer processes, say P_1 and P_2, and two consumer processes, say C_1 and C_2. The producers and consumers communicate items through two buffers, Buffer-1 of size N and Buffer-2 of size M. All these are non-terminating processes, i.e. they endlessly produce or consume items.

Producer P_1 deposits items in Buffer-1, and producer P_2 deposits items in Buffer 2. The consumer processes extract items from either of the two buffers, whichever is non-empty.

Using semaphores write code for the producer and consumer processes satisfying the following requirements:

(1) Your solution should not contain any busy waiting.

(2) A consumer process should not get blocked waiting for an item in a buffer while the other buffer is non-empty.

continue answer for Problem 3

Part B: (5 points) The following solution was proposed by a student for solving the critical section problem for two processes. Determine its correctness. If it is incorrect, show an example which violates the requirement of mutual exclusion. Otherwise, present a justification for its correctness. The two processes are numbered 0 and 1. The shared variables are:

```
  var flag: array[0..1] of boolean (* initially both flags are false *)
      turn: 0..1;  (* initial value = 0 *)
Process (i)
    /* i can either 0 or 1, j is (i+1) mod 2  */
    repeat
         flag[i] = true;
         while ( turn != i )  do   {
            while ( flag[j] ) do   /* do nothing */ ;
            turn = i;
         }
         CRITICAL SECTION
         flag[i] = false;

         REMAINDER SECTION
   until  false;
```

Problem 4: (30 points) This problem as 5 questions, each of 6 points.

1. In Linux, the data and the stack segments are paged and swapped to the swap partition on the disk, but the text segment is treated differently. Explain how and why?

2. Linux kernel provides *interruptible* and *non-interruptible* mechanisms for a process to sleep/block in the kernel mode execution. Explain the differences between these two and under what condition you would use one or the other.

3. There are two approaches for managing the swap space. One is to allocate a fixed size disk partition, and the other is to use files for swap space. What are pros and cons of these two approaches?

4. For recovery, most files-systems take incremental backup dumps regularly, typically every day. It 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 such directories, even though some of them are not modified since the last incremental dump?

5. A user's home-directory contains and executable file with the set SetUID bit on. The file permissions are `rwxrwx-wx`. Does this situation pose a security vulnerability? If so, explain how an attacker can exploit this situation.

Problem 5 (12 points):

Consider a system with a single CPU server in which jobs arrive at rate λ per second. Jobs are processed using the FCFS scheduling policy. Each job requires average S seconds of service time.

1. (2 points) What is the server (CPU) utilization (denoted by ρ) in this system? Express your answer in terms of λ and S.

2. (4 points) Let N denote the average number of jobs in this system and let T be the average amount of time a job spends in the system before departing. Given that jobs arrive at random points in time, a newly arriving job would find the average number of jobs in the system equal to N.

   With FCFS discipline, an arriving job has to wait for the completion of all of the jobs before it in the system. Ignoring the amount of service a job in the system may have already received and assuming that each of these jobs would require S seconds of average service time, give an expression for the average turnaround time T in terms of S and N.

3. (4 points) By Little's Law, $N = \lambda.T$. Using this law, find the expression for N in terms of the server utilization ρ.

4. (2 points) What is the value of N when ρ is equal to 1? What is its value when ρ is 0.5?

Continue answer for Problem 5

Problem 6: (12 points)

1. (4 points) Briefly describe the buffer overflow problem and how is it utilized by an attacker to gain access to a system.
2. (4 points) Briefly describe at least one software-based approach to prevent such attacks.
3. (4 points) Briefly explain how can one utilize a segment-based memory management architecture to prevent such attacks

Continue answer for Problem 6