MCS-378 Intraterm Exam 2

Serial #:

This exam is closed-book and mostly closed-notes. You may, however, use a single 8 1/2 by 11 sheet of paper with *hand-written* notes for reference. (Both sides of the sheet are OK.)

Please write your name only on this page. Do not turn the page until instructed, in order that everyone may have the same time. Then, be sure to look at all problems before deciding which one to do first. Some problems are easier than others, so plan your time accordingly. **You have 50 minutes to work. Choose only four of the five problems. If you do any work on all five, mark one "do not grade." Only four will be graded.**

Write the answer to each problem on the page on which that problem appears. You may also request additional paper, which should be labeled with your test number and the problem number.

Printed name: _____

On my honor, I pledge that I have not given, received, nor tolerated others' use of unauthorized aid in completing this work.

Signature for above honor pledge: _____

Problem	Page	Possible	Score
1	2	25	
2	3	25	
3	4	25	
4	5	25	
5	6	25	
Total		100	

1. [**25 Points**]

   (a) Suppose a system uses 4-KB pages, that is, pages of 4096 bytes each. Suppose that the page table shows that page 0 is mapped into page frame 10, page 1 is mapped into page frame 100, and page 2 is mapped into page frame 1. What physical address corresponds to the virtual address 6104? (All numbers in this problem are given in decimal.)

   (b) Briefly explain the difference between demand paging and prepaging.

   (c) Under what circumstances would demand paging likely outperform prepaging?

2. [**25 Points**] Suppose a system has three page frames, each initially empty. If the first four page references are 1, 2, 3, and 1 again, these will constitute three page faults and then a hit, independent of whether LRU or FIFO replacement is used. The contents of the three page frames will be pages 1, 2, and 3. Now suppose the fifth page reference is for page 4. This is a page fault.

   (a) Which page is replaced using LRU replacement?

   (b) Which page is replaced using FIFO replacement?

   (c) For the particular reference sequence considered here (1, 2, 3, 1, 4), reducing the number of page frames from three to two would increase the number of page faults, independent of whether LRU or FIFO is used. But there are some other sequences of page references where reducing the number of page frames from three to two could actually decrease the number of page faults, at least for some replacement policies. What is the name of this strange phenomenon, in which decreasing the number of page frames available decreases the number of page faults?

   (d) The phenomenon described in part (c) can affect which of the following? (I) LRU only, (II) FIFO only, (III) both LRU and FIFO, (IV) neither LRU nor FIFO

3. [**25 Points**] The following C++ program makes use of the `fork` system call:

```
#include <unistd.h>

int main(){
  fork();    // this is line a
  fork();    // this is line b
  sleep(5);  // this is line c
  return 0;
}
```

Assume that when this program is run, the `fork` procedure does not encounter any error condiion.

(a) How many processes execute the `fork` on line a?

(b) How many processes execute the `fork` on line b?

(c) How many processes execute the `sleep` on line c?

(d) Of the processes mentioned in part (c), how many were directly forked off by the original process as children?

(e) Of the processes mentioned in part (c), how many are grandchildren of the original process?

(f) Draw lines to match the following system calls with their descriptions:

`fork`	can turn an ordinary process into a zombie
`execve`	can increase the number of processes
`exit`	can make a zombie go away
`waitpid`	always leaves the number of processes and of zombies unchanged

4. [**25 Points**] On a POSIX system, a file and a directory are both owned by user 37 and group 53, and both have permissions `rwx--xr--`; that is, `rwx` for the owner, `--x` for the group, and `r--` for others. The members of group 53 are users 37, 42, and 71. The only other user on the system has user ID 85. (You may disregard the superuser, `root`, who has user ID 0.)

   (a) Which user(s) may read the file?

   (b) Which user(s) may write the file?

   (c) Which user(s) may execute the file?

   (d) When the file is executed by user 42, what are the two possible numeric values for the effective user ID?

   (e) What determines which of these two possible user IDs is used?

   (f) Which of the following are true?

      i. User 37 may list the contents of the directory.

      ii. User 37 may use the directory in a pathname to access files under it, subject to those files' permissions.

      iii. User 42 may list the contents of the directory.

      iv. User 42 may use the directory in a pathname to access files under it, subject to those files' permissions.

      v. User 85 may list the contents of the directory.

      vi. User 85 may use the directory in a pathname to access files under it, subject to those files' permissions.

   (g) Suppose we wished to additionally allow users 42 and 71 to create and remove files in the directory. What one permission bit would we add?

   (h) Supposed that we wished to allow users 42 and 71 to create files in the directory but not to remove each other's files. Beyond the permission bit mentioned in part (g), what additional bit would you need to set?

5. [**25 Points**]

(a) Assume an inode contains 15 direct block numbers, as well as single, double, and triple indirect block numbers. Further, assume that each block is 8 KB, and that each block number is 4 bytes. What is the largest a file can be without needing to use the triple indirect block? (You may give your answer as a formula that could be used to calculate the answer; you don't need to do the arithmetic. Keep in mind that K means 1024 in this context, not 1000.)

(b) Modify the B^+-tree extent map shown here to reflect what would happen if the file grew by 300 blocks, allocated as three 100-block extents starting at disk blocks 4000, 5000, and 6000 respectively. Assume the leaf nodes can hold at most four extent descriptors and nonleaf nodes have room for up to four keys and the associated pointers. Any plausible correct answer is OK; there is more that one.