CS 111 Midterm Examination #2
Spring Quarter, 2021

You have 90 minutes for this examination; the number of points for each question indicates roughly how many minutes you should spend on that question. While taking this exam, you may consult any materials available on your personal computer at the time of the exam, as well as materials on the class Web site; you may not search the Internet during the exam. If there is a trivial detail that you need for one of your answers but cannot recall, you may ask the course staff. Fill in the fields of this PDF file, then submit it at www.gradescope.com.

By filing this examination electronically, you acknowledge and accept the Stanford University Honor Code, and you affirm that you have neither given nor received aid in answering the questions on this examination.

____
(Name)

____
(SUNet email id)

Problem	#1	#2	#3	#4	#5	#6	Total
Score							
Max	10	3	10	10	10	35	78

Problem 1 (10 points) Friendly advice

(a) (3 points) A friend comes to you with a suggestion for disk management. Your friend points out that modern operating systems don't allow disk space to become totally full, and suggests that the simplest way to implement this is to permanently block off a range of blocks at one end of the disks, so those blocks are never allocated. Is this a good idea or a bad idea, and why?

(b) (3 points) Your friend also suggests a performance improvement for the clock algorithm. Rather than skipping over pages whose reference bit is set, your friend suggests just evicting the first page the algorithm encounters. Is this a good idea or a bad idea, and why?

(c) (4 points) Your friend then suggests that it would be better for a file system to eliminate inodes as a separate structure and simply store all of the inode information for a file directly in the directory entry for the file. Give one advantage and one disadvantage of this new approach.

Problem 2 (3 points)

Internal fragmentation occurs both in virtual memory systems based on paging and in file systems. In which of these two subsystems is internal fragmentation a more significant issue, and why?

Problem 3 (10 points)
Consider the segmented approach to memory management discussed in class, and assume that the segment number is taken from the top few bits of the virtual address. That mechanism allows segments to grow upwards (to larger virtual addresses), but not downwards. For example, the mechanism would not allow a stack segment to initially occupy the top virtual addresses of a segment's region and grow downward as needed. Describe how the mechanism could be modified to support segments that grow downwards as well as upwards, while retaining a segmented approach. Note: in order for your mechanism to be implemented efficiently in hardware, it needs to use only simple operations such as addition, subtraction, comparison, and bit concatenation.

Problem 4 (10 points)

This question concerns the 4.3 BSD multi-level file structure which uses zero, one, or two levels of indirection, depending on the file size. Assume that blocks are 4 Kbytes, block pointers are 32 bits, and the file size stored in inodes is 64 bits.

(a) (4 points) What is the largest file that can be supported by this structure? You may round your answer to the nearest power of 2. You may use a calculator to compute the answer, but show the formula(s) used for any calculations.

(b) (6 points) The maximum file size could be increased by adding more block pointers to the inode, for 3-level indirection, 4-level indirection, and so on. However, there will be a point where it doesn't help to add more levels of indirection. What is the largest number of levels of indirection that makes sense, and why?

Problem 5 (10 points)

Your friend from Problem 1 has built a file system that uses write-ahead logging for crash recovery; it logs both metadata and file data. However, you notice that the system does not seem to recover properly after some crashes. In looking over the file system code, you discover that in some situations the system creates a log record of the form “append data D to the file with i-number I” (the log record contains an opcode indicating an append operation, plus the new data and the file’s i-number).

(a) (5 points) How can this log record cause incorrect behavior after a crash?

(b) (5 points) Describe how the log record could be restructured to eliminate this problem. Be precise about exactly what information should be present in the log record, and why that fixes the problem.

Problem 6 (35 points)

Modify your solution to Project 7 ("Unix V6 File System") to support symbolic links when opening files.

(a) (5 points) What changes would need to be made to existing structure declarations in the project code to support symbolic links? You don't need to worry about additional issues that might arise if structures grow.

(b) (30 points) Modify the `pathname_lookup` function you wrote for Project 7 to properly handle symbolic links. You do not need to write the code to create symbolic links, but your answer to (a) must make it clear how those links will appear in the file system data structures. You may assume that the contents of a symbolic link are never longer than 512 bytes, and that the target of a symbolic link is always in the same `struct unixfilesystem` as the source.

Include here or on the following pages any new functions or structure declarations that you write, as well as any existing ones that you modify.

Additional working space for Problem 6:

Additional working space for any problem, if needed