Your name:

CSE 30341 Operating Systems: Module 4 Exam

OPEN BOOK, OPEN NOTES, CLOSED ELECTRONIC SEARCHES

INDIVIDUAL EFFORT

Duration: 30 minutes

Keep It Simple. *Make no more assumptions than is needed*. You may write your justification for your answer in the space below. I look at the argument to give partial credit. You can also use these argument to convince me in person that your answer was correct (using only the assumptions and arguments written by you in the answer). Football penalty rules do not apply to his exam, there will be no penalty for you in trying to convince me

1) FACT: *Thrashing is caused when the process working set is larger than the available memory. Thrashing can be avoided by giving more memory to the process.*

For a given amount of memory, can we avoid thrashing by reducing the page size? Justify your answer.

2) Consider two coding approaches A and B: A uses explicit writes while B uses memory mapped files. For a system that uses FIFO page replacement algorithm, which approach causes the least number of disk writes? Assume that the caches are not unified and that the buffer cache is disabled. Justify your answer.

Approach A:	Approach B:

int fd = open(file, O_RDWR);
int mat[100];
 	
for (int i=0; i<100; i++) {
mat[i]=10;
write(fd, &mat[i], sizeof (int);
}
	int fd = open(file, O_RDWR);
	int *ptr;
	
	ptr = mmap(fd, ….); // Mmap a properly initialized file
	for (int i=0; i<100; row++)
	*ptr++=10;

3) FACT: The computer BIOS understands the format of boot control block. Master boot program is stored in the boot control block. The master boot program understands the format of volume control block. Inside each volume, we store a boot program that understands the way file systems are stored in the specific volume. The boot program uses this knowledge to locate the OS kernel and runs this OS kernel to boot up the machine.

Suppose you want to dual-boot your PC to Windows and Linux. Specify the role of the following:

a. BIOS:

b. Master boot program:

c. Boot program in Windows Volume:

d. Boot program in Linux Volume:

Your name:

4) *FACT: The UNIX inode structure uses direct blocks, single indirect block, double indirect blocks and triple indirect blocks.*

Consider the scenario where we operate on a very large file in UNIX. The disk overhead for reading the directory blocks is smaller for sequential access than it is for random access. Why?

5) Consider an operating system that uses a bit-vector to maintain free-space. Suppose your Operating System crashed, completely ruining (meaning that the bit-vector is completely random) the “on-disk” free-space list. Briefly explain the implications of operating with a corrupt bit-vector. Can you recreate the bit-vector by looking at the current state of the file system (without going to the backups).

6) Consider a file system built using flash memory (e.g., used in USB sticks). Flash memory supports the operations: read, write and erase. Updating a memory requires an explicit erase followed by a write. Flash technology has the limitation that any block can only be erased so many times before the block becomes unusable. In this system, to maintain the free-list, do you prefer a bit-vector based approach or the linked list based approach? Justify your answer.