Name_____SAMPLE_____
Last 4 digits of Student ID_____

CSE 3320 GRADE:_____/100
Operating Systems
Exam 1, Fall 2012

Please read this:
This is a closed book, closed notes exam. You may, however, use one sheet of notes. Use a dark ink (or pencil) and print answers on the test paper following the question or on separate paper. Please **do not write on the back of sheets** of paper **or** on the **back of the exam**. Please **keep answers to questions together** and try to avoid continuing answers off the page. Please answer the questions in a **few keywords**, complete sentences are not necessary, **be brief**! Please put your name on each page. Turn in all pages of the test. **Write** your answers **legibly**. Unreadable answers will be counted wrong. Make sure you have all pages of the test.
Read each question **carefully** (twice) and be sure your answer addresses the question. Overly general (non-specific) answers will be counted wrong. If any part of an answer is incorrect (even if other parts are correct) points will be deducted.
Point values are given for each question. The exam has a total of 100 points.
Please turn in the exam promptly when called for. **Late exams will have points deducted**.

"Explain" - means briefly describe why - just a few words.

You do not need to calculate the final result - just show the calculation, and your reasoning.

1. [30 pts] Short Answer:
   (a) On most PCs, disks have a disk controller.
       (1) Is this controller hardware or software (or something else)?
       (2) Is this controller part of the OS? Why?
       (3) What is the purpose of this disk controller?
       (4) Does the disk controller replace a disk device driver (why)?
   (b) When does a process change from the "wait" state directly to the "run" state (or why doesn't it)?
   (c) (1) How would an OS use a timer interrupt?
       (2) Briefly, what happens during a timer interrupt?
   (d) What is "secondary" storage? What is it used for?
   (e) Can a single core, single cpu system do multitasking? How can it, or why can it not?

2. [20 pts] Short Answer:
   (a) For the following (partial) code, describe what happens, in what order, and what will get printed (in what order):

```
if (!(child1 = fork())) {
   printf("A, I am = %d\n", child1 );
   if (!(child2 = fork())){
      printf("B, I am = %d\n", child2 );
   }
      else
        printf("C, I am = %d\n", child2 );
   exit( 0 );   }
printf( "D, I am = %d \n", child1 );
exit( 0 );
```

(b) For the following (partial) code, describe what happens, in what order, and what will get printed (in what order):

```
execl( "/bin/cat", "/bin/cat", "f.c", NULL );
printf("A \n" );
child = fork();
if !(child) {
   printf("B \n" ); }
else
   printf("C \n" );
exit( 0 );
```

3. [20 pts] Short Answer:
   (a) When a program written in C needs to find out the time it can call the C function "time()". However, that function time() is not built in to the Operating System. Please explain how the C program can use the time() function to get the time from the OS (step by step, what happens)?
   (b) What can a program do, that will create a new process control block (PCB)? Where is the PCB stored (on the disk, in registers, on the graphics card, or somewhere else?)
   (c) Almost all OSs have different execution modes for processes: privileged mode (also known as supervisor mode), and user mode. Why have these different modes for processes, and what type of process runs in privileged mode and what runs in user mode?
   (d) Is a microkernel the same thing as a hardware virtual machine (such as VirtualBox)? If not, how are they different?

4. [30 pts]
   For the CP/M operating system:
   (a) How is RAM memory used, from location 0 to the maximum address, in order, what is placed in memory?
   (b) Does a single tasking OS such as CP/M need a "shell" (command processor)? Why or why not?
   (c) For a custom designed CP/M disk, the disk had 40 tracks of 32 sectors each was 512 Bytes long and the disk had two sides. The disk had the "standard" CP/M disk directory.
      (1) What is the raw disk capacity.
      (2) Not all of this space could be used for files, what occupied the non-file space?
      (3) For this disk, how many 1 byte files could be stored?
      (4) (4a) What is the largest single file possible?
          (4b) For the large file in (4a), called "big.c", describe the contents of the first two directory entries.
   (d) Even though early CP/M systems had very small memory they often ran very large programs. How could they (please briefly explain)?

Bonus [5 pts]:
After CP/M was used for a few years, many users wanted larger file sizes, file (creation or modification) time and date as well as larger file names. But they didn't want the overhead to increase by very much (so one can not just double or triple the size of something).
Starting with the old CP/M structure, what would need to be added or changed to allow these enhancements? You do not need to give code or describe byte-by-byte changes, but be detailed.