CS 450 Exam 1

Mon. 10/3/2016

Name: ______________________________

Rules and Hints

- You may use one handwritten 8.5 × 11″ cheat sheet (front and back). This is the only additional resource you may consult during this exam. *No calculators.*
- You will lose points for extraneous incorrect information in an otherwise correct answer.

Grade

	Your Score	**Max Score**
Problem 1: Short answer		24
Problem 2: Interrupts		18
Problem 3: Virtualization		8
Problem 4: Tracing processes		25
Problem 5: Tracing pipes		25
Total		**100**

Problem 1: Short answer (24 points)

Answer each of the following questions. If you can't clearly describe the difference between the two terms, you can get partial credit by just defining either or both.

Part A

What is the difference between an *executable program* and a *process*?

Part B

What is the difference between a *syscall* and an *asynchronous notification*?

Part C

What is the difference between an *interrupt vector* and an *interrupt handler*?

Part D

What is the difference between *virtual memory* and *virtualization*?

Problem 2: Interrupts (18 points)

Part A

When an interrupt happens, explain how the stack pointer and/or the program counter/instruction pointer need to change.

Part B

Explain how the old values of these two registers are saved and then restored, if necessary.

Part C

You have a system that is constantly being barraged with network traffic. What is interrupt coalescing, and would you expect it to be good, bad, or neutral for this system?

Problem 3: Virtualization (8 points)

Part A

Define *hypervisor*.

Part B

An ISA has a machine-code instruction that disables interrupts. Executing this instruction in user mode causes an illegal instruction exception. Is it possible to virtualize this instruction, and if so, how?

Problem 4: Tracing processes (25 points)

Consider the following program. Assume that its original process ID is 3407 and that the shell's PID is 1502. Assume that subsequent processes created by this process will get the PIDs 3408, 3409, etc. in order.

```
int main(void) {
    int a = 5;
    int b = 2;
    pid_t x = fork();
    b += 1;
    pid_t y = fork();
    if (x + y == 0) {
        a += 1;
    }
    printf("Process %d (parent %d): (%d, %d)\n", getpid(), getppid(), a, b);

    return 0;
}
```

Part A

Show a possible output for the program (assuming that all necessary headers have been included). Answers may vary – you will get full credit if your answer is internally consistent.

Part B

How would your answer to Part A change if the line

```
execlp("pwd", "pwd", (char *)NULL);
```

were inserted just above the increment to *a* inside the if-statement?

Problem 5: Tracing pipes (25 points)

The code below is a seriously flawed version of the pipe tracing code you saw in class. On the next page, draw each process's file descriptor table just before it exits, and explain what is wrong with each of these tables (if anything). You can assume that *wc* and *who* don't make further changes to the file descriptor table once executed.

```
int main(void)
{
    int pfd[2];
    pid_t pid;

    pipe(pfd);

    pid = fork();
    if (pid == 0) {
            dup(pfd[0]);
            execlp("wc", "wc", NULL);
            close(0);
            close(1);
    }

    pid = fork();
    if (pid == 0) {
        dup(pfd[1]);
        execlp("who", "who", NULL);
        close(0);
        close(1);
    }
    close(0);
    close(1);

    while ( wait(NULL) != -1) ;
}
```

[Drawings for Problem 5 go here]