CSCI 331: Operating Systems I

Midterm # 1 -- open book section

The entire exam is to be turned in at 4:20PM. Work the closed book section first and turn it in before you consult your books and notes to work on the open book section. Write your answers to the open section on your own paper.

Problem 1: 5 points

Suppose a C program starting with the following lines:

```
main(argc, argv)
 int argc;
 char *argv[];
```

is compiled and the compiled code is stored in the file **easy**. If the command:

```
% easy argv question
```

is executed, what are the values of **argc** and the array elements of **argv**?

Problem 2: 10 points

Suppose **P[0]** is a file descriptor that refers to the read end of a pipe. Under what circumstances with the system call

```
read(P[0], buff, buff_size)
```

never return?

Problem 3: 5 points

How many times is the character **'X'** printed to standard output by the following four lines of C code:

```
fork();
fork();
fork();
write(1, "X", 1);
```

Problem 4: 15 points

In this question we explore the uses of pipes to obtain mutual exclusion by simulating semaphores. Suppose one ``master" process creates a pipe with the system call:

```
pipe(S);
```

initializes the pipe by writing a single character to it, and then creates several children with share both the read and write ends of the pipe.

Now show how the processes can use Minix system calls on the shared pipe to mimic the semaphore **wait** and **signal** operations.

Problem 5: 15 points

Many C compilers for MS/DOS systems provide a library routine called **spawnl**. The **spawnl** routine creates a child process to execute a command and then returns control to the parent when the child terminates. The parent and child processes do not run concurrently.

Outline how you would implement the function of the **spawnl** routine in Minix. State the system calls you'd use, but don't go into detail about the system call arguments.