COSC 4330

FIRST MIDTERM

JUNE 10, 2003

This exam is **closed book**. You can have **one page** of notes. Please answer every part of every question.

1. Match each of the following features with the *single sentence* that describes it best: (10×3 points) (*Hint: Several of the choices offered are plain wrong.*)

timer interrupts	_i	dual-mode CPU	_j
UNIX	_k	non-volatile RAM	_n
layered kernel organization	_m	batch systems	_a
swapping	_f	system call	_e
memory protection	_b	monolithic kernels	_h

- a) Allow their users to submit several jobs to be processed in sequence
- **b**) Always implemented in hardware.
- c) Implemented as a user-level process on Windows NT.
- d) Much slower than microkernels.
- e) Occasions at least *two* context switches.
- f) Provides a good way to make space in main memory.
- g) Sends a signal to another process.
- **h**) Still the fastest kernel organization.
- i) Used to prevent the running process from monopolizing the CPU.
- j) Used to prevent unauthorized access to users' files.
- **k**) Was the first OS that could run on different computer architectures
- I) Was the first OS with a graphical user interface.
- m) Would be a good idea if it was working.
- n) Would allow a much more intensive usage of delayed disk writes.
- 2. What are the respective advantages of microkernels and monolithic kernels? $(2 \times 5 \text{ points})$

Microkernels are smaller, more manageable and easier to debug than monolithic kernels. Unfortunately, they are much slower as system calls involving a server process require four context switches instead of two.

- **3.** For each of the statements below, indicate in one sentence whether the statement is true or false (2 points), **and why** (3 points).
 - A process waiting on a disk I/O is in the suspended state.
 False, it is in the waiting state.
 - b) It makes no sense to have memory protection on a single user system.

<u>False</u>, it prevents user processes from interfering with each other or damaging the kernel.

c) Processes never interrupt themselves.

False, processes interrupt themselves every time they do a system call.

- d) execve() system calls are often followed by a fork() system call.
 <u>False</u>, the fork(O normally precedes the execve().
- e) In a multiprogramming system, there can be many programs in the system but there is never more than a single process.

<u>False</u>, multiprogramming means having several processes in the system at the same time.

f) A UNIX process *shares* the address space of its parent.

False, each process has its own private address space.

4. What is the default action that a Unix process takes when it receives a signal from another process? (5 points) What can it do to prevent that from happening? (5 points)

By default, A UNIX process terminates when it receives a signal. To prevent that the process can catch it by using the signal() system call. The ninth signal, SIGKIL, cannot be caught.

5. How many lines of output will the following program print out? (5 points)?

```
main(){
    fork(); fork(); printf("Hi!\n");
}
```

The program will print __4____ lines.

6. What is happening to stdin, stdout, and stderr after the following lines are executed. $(3 \times 5 \text{ points})$