NAME: ________________________________ (FIRST NAME FIRST) **SCORE**: ________

COSC 4330 FIRST MIDTERM SEPTEMBER 25, 2007

*This exam is **closed book**. You can have **one page** of notes. UH expels cheaters.*

1. ***Advantages and disadvantages:*** You will get no credit if you mention a disadvantage when an advantage is asked and vice versa. I will read the first three lines of each answer. **(**6×5 points)

a) What is the major advantage of ***multiprogramming***?

It allows the CPU to do something else while a process is waiting for an I/O operation. (also: It allows timesharing.)

b) What is the major disadvantage of the ***round-robin*** scheduling policy?

It cannot provide a good response time for interactive processes when the system is loaded without causing too many context switches.

c) What is the major advantage of ***microkernels***?

They reduce the size of the kernel making it more manageable.

d) What is the main disadvantage of ***modular kernels***?

They let users load external module in the kernel address space, thus making the kernel much less reliable.

e) What is the main advantage of the ***symmetric organization*** for multiprocessor operating systems?

They let the kernel run on several processors at the same time, thus eliminating a potential bottleneck.

f) What is the main advantage of ***delaying writes*** to the disk?

Delaying writes to disk ensures that many small sequential writes will normally result in a single disk access.

2. Which of the following statements apply to (a) kernel-supported threads, (b) user level threads and (c) all threads? (5 points per correct line)

Property	*Kernel-supported*	*User-level*	*Both types*
They do not require kernel modifications.	___	_X_	___
They share the address space of their parent.	___	___	_X_
They handle blocking system calls in a correct fashion.	_X_	____	___
They allow the kernel to allocate several processors to the threads sharing an address space.	_X_	___	___

3. ***Other Questions with Short Answers:*** **(**6×5 points)

a) Why would a process ***interrupt itself***?

To notify the kernel that the running process wants to make a system call.

b) Why is ***memory protection*** always implemented in hardware?

Because it must be very fast.

c) What is the difference between ***soft*** and ***hard*** real-time applications?

Failure to meet a deadline in a hard real-time system might have catastrophic consequences, including loss of human lives, which is not the case in soft-real time systems.

d) Which processes can be safely ***swapped out*** to make space in main memory?

We can safely swap out processes that have been in the waiting state for a long time.

e) What happens when a process executes a **`signal()`** system call?

A process executing a signal() system call sets up an interrupt handler that will catch .a specific interrupt.

f) In which ***state*** is a process waiting for the CPU?

A process waiting for the CPU is in the READY state (NOT in the waiting state!).

4. Unix is said to use a ***toolkit approach***. What do we mean by that? (5 points) Name one shell construct that facilitates this approach and explain how it does it? (5 points)

- Most UNIX programs were designed to perform a specific small task and to be combined with other programs to achieve the result wanted by the user.
- The pipe construct is used to let several programs perform different steps an application.
- The construct "a | b" ensures that the standard output of program "a" becomes the standard input of program "b."

5. In addition to preventing user processes from directly accessing data stored on disks, which other two precautions should be taken to prevent unauthorized access to these data and why? (2×5 points for a correct answer in two parts)

To be secure a system should also

- Have memory protection and
- Prevent users from rebooting the system with anything but the original OS.