COSC 6360 **QUIZ #1** **JULY 19, 2010**

Name: _____ (First name first) **Score:** ____

Closed book. *You can have with you one single-sided 8½ by 11 sheet of notes. UH expels cheaters.*

1. Questions with very short answers: (4×10 pts)

a. Which UNIX system call is used to construct a ***single*** directory tree across disk partition boundaries?

Answer: The mount **system call.**

b. In Mach, what should be the ***inheritance attribute*** of the ***code segment*** of a process?

Answer: shared

c. Consider a 32-bit UNIX system running the Fast File System. Assuming that the file system block size is 8 KB, which files can be accessed with at most one level of indirection?

Answer: All files occupying 96KB + 8KxBK/4 = 96KB + 16MB ~~**bytes**~~ **or less.**

Explanation: If the block size is 8KB, we can access

- 12x8KB= 96KB directly from the i-node,
- The next 8K/4 = 2,048 blocks through one level of indirection,

d. In Mach, what is the cost of reclaiming a page from the global queue containing all pages waiting to be written back on disk?

Answer: two context switches

2. What is the main difference between ***complete subblocking*** and ***partial subblocking***? (20 pts)

Partial subblocking requires all the pages forming a subblock to occupy contiguous addresses in the process virtual memory as well as in physical memory. Complete subblocking requires all the pages forming a subblock to occupy contiguous addresses in the process virtual memory but lets tem occupy any location in physical memory.

3. What is the best way to implement the UNIX ***fork() system call***? (10 pts) Why? (10 pts)

The best way to implement the UNIX fork() system call is using copy-on-write. This technique lets the parent and the child process share the same address space until the child performs an exec(). UNIX fork() semantics are preserved by making a two copies of each page that one of the two processes attempts to modify and letting each process access its own copy of the shared page. Copy-on-write works well because most child processes modify very few pages of their address space before they perform an exec().

(Any good student will recall that copy-on-write is a lazy technique.)

4. Why does the Fast File System subdivide its disk partitions into ***cylinder groups***? (20 points)

FFS subdivides its disk partitions into cylinder groups and ensures that each cylinder group contains both i-nodes and data blocks. This eliminates the long seeks between the cylinders containing the partition i-node table and those containing its data blocks.