COSC 6360 FIRST MIDTERM JULY 20, 2006

This exam is **closed book**. You can have **one sheet** (i.e., **two pages**) of notes.
Please answer every part of every question.

1. Explain the following terms: (5×5 points)

(a) ***Munin migratory variable:***

A shared variable that is never replicated in different processes: every process accessing it will always get full read and write access even if it only requested read access.

(b) ***Unix pipe:***

A mechanism letting the standard output of a process to forwarded to the standard input of another process as in "ls –l | more".

(c) ***Inheritance attribute:***

A parameter in the Mach address map that specifies whether pages in a given range of addresses are to be copied, shared or ignored at fork() time.

(d) ***UNIX special files:***

File names associated with specific devices such as "/dev/hd0".

(e) ***Thre*ad:**

A process executing in the same address space as its parent.

2. Consider a virtual memory system with 64-bit addresses and a clustered page table with a clustering factor of 2. Given that each address occupies 8 bytes, what would be the length of a page table entry assuming that we are implementing:

(a) ***Complete subblockin***g (5 points)? ____4×8=32_ **bytes**

(You may detail here your computation for possible partial credit)

(b) ***Partial subblocking*** (5 points)? ___3×8=24__ **bytes**

(You may detail here your computation for possible partial credit)

3. Describe the contents of a Unix *directory entry*. (2×5 points)

A UNIX directory entry contains a name and an i-node number.

4. How does the current Berkeley UNIX *page replacement policy* differ from that described by Babaoglu and Joy? (5 points)

A second hand, following the original clock hand at a fixed angle, was added to reclaim that have been marked not referenced by the first hand and have been accessed since then.

5. What is the *main advantage* of *mapped file systems* over *conventional systems*? (5 points)

File blocks that are brought into main memory are mapped in the address spaces of the processes that access them. Hence the contents of these blocks can be directly accessed by each process without any additional context switches.

6. Assume you are working as system administrator for an engineering company. A fellow employee tells you that he believes an intruder has had access to his Unix account. He adds that he was told he should change his password and forget about the incident. Is that sound advice? (5 points) Why? (5 points) Is there anything that you could do to ensure that the intruder cannot access again the account of your fellow employee? (10 points) (***Hint:*** *You have just compiled a list of all public directories where every user can store files.*)

No, this is not sound advice because the intruder could have left somewhere an executable file owned by the fellow employee, writable by all and having its set user-ID bit on. If this is the case, the intruder could use this executable to regain access to the Unix account of the employee.

You should immediately look for executable files owned by the fellow employee, writable by all and having its set user-ID bit on in all public directories.

7. What would happen if someone writing a Munin program forgets to insert

   (a) A *reques*t() call *before accessing* a shared data variable? (5 points)

   The process accessing the shared variable will not be guaranteed to have the most recent value of the shared variable.

   (b) A *release*() call *after accessing* that shared data variable? (5 points)

   The new value of the shared variable will not be propagated to all the other processes.

8. How does Unix implement the *access control lis*t of a file. (5 points) Give one *advantage* and one *disadvantage* of this approach. (2×5 points)

Unix specifies three classes of users (owner, all members of one of the groups specified in /etc/group, all users) and three access rights (read, write and execute).

This implementation keeps the access control list very short and allows it to be stored in the i-node. It lacks flexibility as users cannot define arbitrary subsets of users who have specific rights.