Homework 4

Due Date: December 4, 2008 **Points:** 100

Questions

1. (*20 points*) Consider a system in which free space is kept in a free list.
   (a) Suppose that the pointer to the free list is lost. Can the system reconstruct the free list?
   (b) Suggest a scheme to ensure that the pointer is never lost as a result of memory failure.

2. (*20 points*) Consider a file currently consisting of 100 blocks of 512 words each. Ignoring the access to update the device directory, and assuming a disk block number fits into a single word, how many disk I/O operations are involved with contiguous, linked, and indexed allocation strategies, if one block:
   (a) is removed from the beginning?
   (b) is added in the middle?
   (c) is added at the end?

3. (*20 points*) In an electronic funds transfer system, there are hundreds of identical processes which work as follows. Each process reads an input line specifying an amount of money, the account to be credited, and the account to be debited. Then it locks both accounts and transfers the money, releasing the locks when done. With many processes running in parallel, there is a very real danger that, having locked account *X*, a process will be unable to lock account *Y* because *Y* has been locked by a process now waiting for *X*. Devise a scheme that avoids deadlocks and show that it does indeed avoid deadlocks. Do not release an account record until you have completed the transactions. (In other words, solutions that lock one account and then release it immediately if the other is locked are not allowed.) (*text*, problem 21, modified)

4. (*20 points*) Can the Trojan Horse attack work on a system protected by capabilities? (*text*, problem 5.29)

5. (*20 points*) After getting your degree, you apply for a job as director of a large university computer center that has just put its ancient operating system out to pasture and switched over to Linux. You get the job. Fifteen minutes after starting work, your assistant bursts into your office screaming: "Some students discovered the algorithm we use for encrypting passwords and posted it on the Internet." Does knowing the algorithm used for protecting passwords on the UNIX system pose a threat to the security of the system? If not, why not? If so, how serious is the threat? (*text*, problem 5.23, modified)

Extra Credit

1. (*10 points*) Prove that Belady's anomaly cannot occur in a stack algorithm.

2. (*10 points*) Some versions of Linux allow a user to connect a "watchdog" program to a file such that the watchdog is invoked whenever a program requets access to the file. The watchdog then either grants or denies access to the file. Discuss the pros and cons of using watchdogs for security.