CS-736: Advanced Operating Systems
Sample Final Exam

A. Arpaci-Dusseau

1. For **each** of the following sentences concerning the VAX/VMS virtual memory system, circle the word in bold that makes the sentence true.

   - If there are two user processes in the system, there will be a total of **1 / 2 / 3 / 4 / 5 / 6** page table(s).
   - Each page table is defined by **1 / 2 / 3 / 4** hardware register(s).
   - The *user page table base register* contains a **virtual / physical** address.
   - The *system page table base register* contains a **virtual / physical** address.
   - **One / Two / Three** memory lookups are required to translate a user's virtual address to a physical address (assuming there is no TLB).

2. In Multics, "making a segment known" entails translating a symbolic path name to which type of address? Circle the **one** best answer.

   (a) Indirect address
   (b) Linkage address
   (c) Generalized address
   (d) Physical address

3. **[8 pts.]** Unix FFS introduced *cylinder groups* to improve locality and thus reduce seek time. Which of the following are placed together in the same cylinder group (assuming there is sufficient space)? Circle **all** that apply.

   (a) I-nodes for all files and directories
   (b) I-nodes for all files in the same directory
   (c) I-nodes for all sub-directories of the same directory
   (d) The i-node and all data blocks of a small file
   (e) The i-node and all data blocks of a small directory
   (f) The i-node and all data blocks of a very large file

4. **[9 pts.]** There are two different approaches for performing a small write in RAID-5 (note: this is **not** a layout issue). First, explain **two** different approaches for writing one data block out of the D data blocks in the group. Second, which approach do you expect to lead to better performance and why?

5. **[5 pts.]** When calling `read()` from an application to read a single data block from an open file in LFS, how many disk reads **could** occur? Circle **all** that apply.

   (a) Zero

   (b) One

   (c) Two

   (d) More than two

6. **[5 pts.]** The LFS cleaner uses information in the *segment summary block* to determine whether each data block in that segment is live. Which of the following information is kept for each data block in the segment summary block? Circle **all** that apply.

   (a) Superblock

   (b) Free block bitmap

   (c) File pathname

   (d) File number (*i.e.*, i-node number)

   (e) Indirect block

   (f) I-node map

   (g) Block number (*i.e.*, offset within file)

   (h) Version number of file

7. **[9 pts.]** What problems can occur during the *Emulation* phase of Coda? Describe the impact of these problems on what the user is and is not be able to do, as well as any options the user has for fixing the situation.

8. Given our knowledge of how file systems are used (at least according to the paper written in 1992), consider the following issues regarding file system design and implementation. (These questions are precisely those given in class and assume you have access to the paper; I would provide sample data if given a similar question on the exam – and would not have 5 questions!)

   (a) How would you perform layout of files (i.e., data blocks) on local, modern disks?

   (b) What prefetching strategy would you implement?

   (c) Should NFS write through each block or perform flush-on-close?

   (d) How long should write-back be delayed in Sprite?

   (e) Should Sprite add an interface to write back a few bytes from a block to the server?

9. **[9 pts.]** You are implementing public-key encryption and begin with the default protocol for exchanging keys.

$$A \rightarrow AS: \quad A, B \tag{1}$$
$$AS \rightarrow A: \quad \{PKB, B\}^{SKAS} \tag{2}$$
$$A \rightarrow B: \quad \{I_A, A\}^{PKB} \tag{3}$$
$$B \rightarrow AS: \quad B, A \tag{4}$$
$$AS \rightarrow B: \quad \{PKA, A\}^{SKAS} \tag{5}$$
$$B \rightarrow A: \quad \{I_A, I_B\}^{PKA} \tag{6}$$
$$A \rightarrow B: \quad \{I_B\}^{PKB} \tag{7}$$

   You now must decide a protocol for sending the message, M, in such a way that no one can snoop on the message, alter the message, or replay a message. For each of the following approaches for sending M from A to B, explain why or why not this approach is secure; if it is not secure, describe how an intruder could take advantage.

   (a) $\{M\}^{SKA}$

   (b) $\{M\}^{PKB}$

   (c) $\{\{M\}^{SKA}\}^{PKB}$

   (d) $\{M, I_A + 1\}^{PKB}$

   (e) $\{M, I_B + 1\}^{PKB}$

NFS

Time (sec)	Action/State of NFS Client X	Action/State of NFS Client Y	Action/State of NFS Server
0			
10			
20			
30			
31			
40			
50			
60			
70			
80			
81			
90			
100			
110			
120			

Time (sec)	AFS Action/State of AFS Client X	Action/State of AFS Client Y	Action/State of AFS Server
0			
10			
20			
30			
31			
40			
50			
60			
70			
80			
81			
90			
100			
110			
120			

10. **[18 pts.]** Given the same Unix program that reads and writes data from a file, NFS and AFS behave differently in terms of the messages sent between clients and servers and the data blocks that are kept in client caches. In this question, you will describe for first NFS and then AFS the general actions taken by Client X, Client Y, and the Server for the following file system operations written in high-level pseudo-code. Focus on the messages that are sent between machines and the state of the caches on the clients (*e.g.*, you should not describe how pathname lookups are performed). If the operation causes no traffic or change in file state, you can leave that line empty. You should assume that all operations occur instantaneously (*i.e.*, there is no delay to travel through the network or to write to disk) and that all file caches are initially empty.

Use the tables on the preceding two pages for your answer. You may remove this page from your exam for easier reference to the pseudo-code.

Time (sec)	**Client X**	**Client Y**
0	fd = open("fileA");	
10	read(fd, block 1);	
20	read(fd, block 2);	
30	read(fd, block 1);	
31	read(fd, block 2);	
40		fd = open("fileA");
50		write(fd, block 1);
60	read(fd, block 1);	
70		close(fd);
80	read(fd, block 1);	
81	read(fd, block 2);	
90	close(fd);	
100	fd = open("fileA");	
110	read(fd, block 1);	
120		close(fd);