Solutions to Midterm I
Fall 2003 15-712

Please write your name on the following line (and on each page of the exam):

This midterm is open book and open notes.

You should write brief, precise, and legible answers. Rambling brain-dump answers may be punished twice — you run out of time writing them and we run out of patience reading them. :–)

If in doubt, list any assumptions you made about the question.

There are eight questions but we will only grade **six**. List the **six** questions we should grade:

Name: _____

1. LFS as a database

   (a) Parameterize LFS in terms of Haerder and Reuter's database taxonomy. For each paramter, circle one value and give a short (one or two sentence) justification.

   Propagation Strategy: ATOMIC / ¬ATOMIC

   ATOMIC: LFS doesn't overwrite data; it looks like shadow paging.

   Note that this atomic means "not update-in-place," as opposed to the A in ACID that guarantees correctness.

   Buffer Replacement: STEAL / ¬STEAL

   ¬STEAL: LFS writes an entire log segment at once.

   End of Transaction Processing: FORCE / ¬FORCE

   FORCE: Once a log segment is ready, it is written out. One could argue that because these writes are asynchronous, they aren't forced.

   Checkpoint Types: TOC / TCC / ACC / FUZZY

   TOC (if FORCE above): Using above policy, TOC comes for free. Even if one argues for ¬FORCE above, TCC comes for free because the log is kept in order so we know when each transaction has completed.

   Also, one could argue TCC because of checkpoints.

   (b) What does LFS need to undo and redo on a crash?

   Because all writes are "atomic" and dirty buffers aren't "stolen," any log in LFS marked completed is consitent. Hence, nothing needs to be undone.

   Because segments aren't marked completed until a checkpoint, any segments written since the last checkpoint must be redone.

   We must also fix invalid directory entries during redo.

Name: _____

2. The eager writing disk array was designed to support transaction processing workloads. Two designers are arguing about whether adding support for track-aligned extents would improve performance for environments in which the database is frequently scanned (for report generation) in addition to its normal transaction processing activity. Give an argument for one of the two positions.

   Con: Eager writing tends to scramble the disk layout, which would make apparent track-aligned extents (in terms of LBNs) not work properly. If there were not a mechanism for preventing (or defragmenting) such problems, it would fail to improve performance.

   Pro: Track-aligned extents are an ideal access pattern for sequential scanning applications that must share a storage system with other activity. It will result in almost streaming bandwidth for the portion of the time that the scanning application gets the disk head, IF the LBNs comprise an actual track.

Name: _____

3. A server crash can affect a clients' ability to read and write files. For each of AFS and Pangaea, explain which files can and cannot still be read and written, and why.

AFS

read: any cached files for which callbacks have not yet been revoked can still be read, because the server is not consulted. Anything not in cache cannot be read, because the server has the data and won't answer.

write: any open files on which writes are being done can still be worked on, but they cannot successfully be closed with new data, because the server is not there to accept it.

Pangaea

read: there should be no effect on ability to read files, because of the redundancy. Only if all machines with gold replicas crash is there a problem.

write: there should be no effect on ability to write files (same as above). Of course, consistency is weak, but that's the semantic model.

Everyone tried this problem.

Name: _____

4. (In)Secure systems

   (a) Consider a secure coprocessor such as Aegis or an IBM 4758. Suppose this hardware is running Linux, and that it has a kernel module that provides an encryption service. (Users send the encryption module a message and a key, and it returns an encrypted version.) Say this system is hacked into through a buffer overrun in the module. List three violations of Saltzer's security design principles that were exploited for this attack, and explain how.

   Economy of Mechanism. A full-size monolithic kernel like Linux is l ikely too big to examine line-by-line.

   Least privelege. What was the encryption module doing running in kernel mode?

   Least common mechanism. Why was there a shared kernel module, rather than a simple library procedure? (This is even the case Saltzer gives!)

   Work Factor. This was an easy penetration.

   There may be arguments for Complete mediation, Separation of Privilege, Fail-safe defaults, or Compromise Recording, but the explanation would be a bit different.

   Psychological acceptability and Open design are not acceptable answers.

   (b) Which class would Anderson assign to such an attack and why?

   Class I (clever outsiders). No sophisticated equipment needed, no proprietary knowledge needed; the attacker took advantage of an existing weakness.

Name: _____

5. The NASD model for file services lets clients directly access objects on storage devices, but uses a "file manager" for directories and other metadata management functions. Give and explain two reasons why it does so.

Trust: corrupted directory structures could mislead client machines into all kinds of badness, from just crashing to reading or writing the wrong object for a given name (but using the client's credentials).

Simplicity: centralizing metadata management functions makes them much easier. Implementing a consistent decentralized directory, for example, would significantly increase the size of the interface specification needed, not to mention the code itself.

Control: related to trust item, the file manager can implement controls like quotas to prevent space exhaustion denial-of-service attacks.

Portability: by separating the file manager, it is easier to create interfaces to different filesystem abstractions.

Embedded Performance: the file manager limits the intelligence required in the computationally limited ASIC on the disk.

Name: _____

6. Two implementors are arguing about whether direct memory access would be a good augmentation to a River-like system. Take a position (pro or con) and explain.

   This question should have said "remote direct memory access." Students were not penalized for not reading our minds, though most did.

   Pro: I think this would be a good match. River could flow data at the client system, as it is able to produce it, and stick it directly into memory very efficiently.

   Con: The queues already eliminate performance bottlenecks; RDMA is unneeded.

Name: _____

7. A friend argued that the Pangaea architecture would be a great way to improve the Harvest proxy cache system. Give one argument for each the pro and the con.

   Pro: Pangaea would allow updates to be propagated quickly, reducing the window of inconsistency for web pages that change.

   Con: the graph for popular web pages could be huge, leading to waste for those that don't change. Also, pages that change frequently could end up creating a lot of extra work for (perhaps) minimal benefit, given users' traditional expectations of WWW consistency.

Name: _____

8. Some systems researchers are considering extensions to Harvest to support direct memory access (aĺa DAFS) among nodes in the cache hierarchy. State whether you think this is good idea and explain why.

I think it's not a good idea, because the memory overhead associated with proxy server interactions is unlikely to be significant, given the speeds of network access points in most environments. This means that the overhead of data copies is unlikely to be high enough to merit the extra system complexity.

An argument could be made for it in corporate network environments with high-speed interconnects. An argument could also be made for it if the company has invested in a very high-speed network link. Such assumptions should be stated explicitly to support using this scheme.

We assumed most people would realize that RDMA in a high-latency network is nonsensical. However, my favorite answer came from an argument in favor: If Harvest is used as an HTTP accelerator, RDMA makes a lot of sense!