# CS 162 Final Spring 01

**Problem 1**. (22 points total) Short Answer.

a. T/F and **Why** (9 points)

Circle the correct answer and provide a *short*, *one-sentence* reason for your answer.

i) Adding a cache never hurts performance.

ii) Unlike public key encryption, using private key encryption requires that the two communicating parties sharing a secret key ahead of time.

iii) The optimal place on the disk for directory blocks in the innermost cylinder of the disk.

b. (9 points) Assume the queue of disk blocks that the file system must write out contains blocks located on the following cylinders: **45, 18, 20, 10, 50**

Assume the disk head just wrote a block on cylinder 15 and is currently over cylinder 16. List the order will blocks be written out with:

i) FIFO:

ii) Shortest Seek Time First:

iii) SCAN / Elevator:

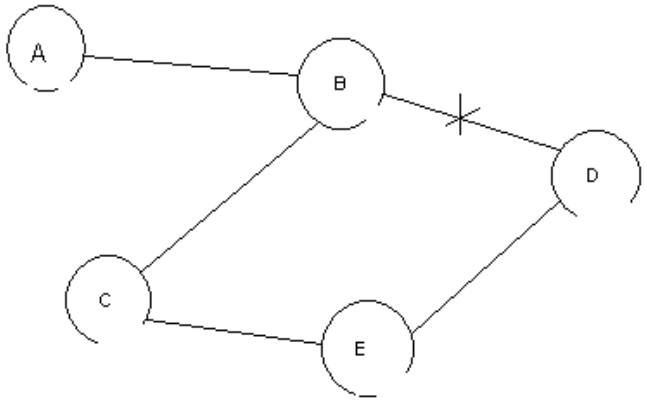
c. (4 points) network file systems use server-callback or client-poll validation schemes for cached data.

i) Which one does NFS use?

ii) Which one does AFS use?

**Problem 2**. (18 points total) Networking.

a. (10 points) Shown below is network configuration for each router's table. The network link marked with an X becomes unavailable. Fill in the final state for each router's table that results after multiple rounds of exchange between the routers.




| A Table | | A | B | C | D | E |
|---|---|---|---|---|---|---|
| Initial | Distance | 0 | 1 | 2 | 2 | 3 |
| | Out link | - | B | B | B | B |
| Final | Distance | | | | | |
| | Out link | | | | | |

| B Table | | A | B | C | D | E |
|---|---|---|---|---|---|---|
| Initial | Distance | 1 | 0 | 1 | 1 | 2 |
| | Out link | A | - | C | D | D |
| Final | Distance | | | | | |
| | Out link | | | | | |

| C Table | | A | B | C | D | E |
|---|---|---|---|---|---|---|
| Initial | Distance | 2 | 1 | 0 | 2 | 1 |
| | Out link | B | B | - | E | E |
| Final | Distance | | | | | |
| | Out link | | | | | |

| D Table | | A | B | C | D | E |
|---|---|---|---|---|---|---|
| Initial | Distance | 2 | 1 | 2 | 0 | 1 |
| | Out link | B | B | E | - | E |
| Final | Distance | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Out link | | | | | |

| E Table | | A | B | C | D | E |
|---|---|---|---|---|---|---|
| Initial | Distance | 3 | 2 | 1 | 1 | 0 |
| | Out link | C | C | C | D | - |
| Final | Distance | | | | | |
| | Out link | | | | | |

b. (8 points). Your new boss at Megasoft Corporation proposes a new sliding window-based reliable network transport protocol.

The protocol works as follows: instead of sending an acknowledgement for eery packet, it only sends an acknowledgement for every 4 packets (by sequence number). Thus ACK #1 is sent when packets with sequence numbers 0-3 have been received. ACK #2 is sent when packets with sequence 4-7 have been received.

Acknowledgements do not cover packets from earlier sequence numbers and can be sent out of order (i.e., ACK #2 can be sent when packets with sequence numbers 4-7 have been received successfully, even if packets with sequence numbers 0-3 have not been received). The protocol also uses a retransmission timer for each packet.

List a one sentence advantage and one sentence disadvantage of using this new protocol.

i) Advantage:

ii) Disadvantage:

**Problem 3**. (18 points) Multi-level Virtual Memory Management.

Consider a multi-level memory management using the following virtual addresses:

| Virtual seg # | Virtual Page # | Offset |
|---|---|---|

Each virtual address has 2 bits of virtual segment #, 8 bits of virtual page #, and 12 bits of offset. Page table entries are 8 bits. All values are in *hexadecimal.*

Translate the following virtual addresses into physical addresses, or write INVALID if the virtual address is invalid:

| Virtual Address | Physical Address |
|---|---|
| 0x304FFF | |
| 0x20239A | |
| 0x139BCD | |

| Virtual Address | Physical Address |
|---|---|
| 0x032AEF | |

| | |
|---|---|
| 0x20CCBA | |
| 0x110123 | |

**Segment Table**

| Start | Number of pages | Flags |
|---|---|---|
| 0x2040 | 0x40 | Valid, read only |
| 0x1010 | 0x0F | Valid, read/write |
| 0x2004 | 0x40 | Valid, read/wrtie |
| 0x0000 | 0x10 | Invalid |

**Contents of Physical Memory**

| Address | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D |
| 0x0010 | 1E | 1F | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 2A | 2B | 2C | 2D |
| … | | | | | | | | | | | | | | | | |
| 0x1010 | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D |
| … | | | | | | | | | | | | | | | | |
| 0x2000 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 |
| 0x2010 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F | 20 | 21 |
| 0x2020 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 2A | 2B | 2C | 2D | 2E | 2F | 30 | 31 |
| 0x2030 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 3A | 3B | 3C | 3D | 3E | 3F | 40 | 41 |
| 0x2040 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 4A | 4B | 4C | 4D | 4E | 4F | 50 | 51 |
| 0x2050 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 5A | 5B | 5C | 5D | 5E | 5F | 60 | 61 |
| 0x2060 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 6A | 6B | 6C | 6D | 6E | 6F | 70 | 71 |
| 0x2070 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 7A | 7B | 7C | 7D | 7E | 7F | 80 | 81 |

**Problem 4**. (27 points total) Filesystems.

a. (6 points) Two machines, A and B, are sharing files over the network. Describe the contents of **B's cache**, *immediately* after the following sequence of file operations, all of which occur within a one second span:

```
B opens X; reads file; closes file

A opens X; writes new version; closes file

B opens Y; reads file
```

```
A opens y; writes new version; closes file
```

Define the old version of the files as X.OLD and Y.OLD, and the new versions as X.NEW and Y.NEW.

i) State the contents of B's cache using NFS and state why in one sentence:

ii) State the contents of B's cache using AFS and state why in one sentence:

b. (6 points) Assume that the number of disk blocks read to read the first block of a file is *R*. In terms of *R*, give a formula for the average time to read the first block of the file. Assume the following constants:

Ty - CPU cycle time in nanoseconds

Ts - Average disk seek time in seconds

F - File size in bytes

Tr - Average disk rotational latency in seconds

BWd - Disk bandwidth in bytes/second

B - Block size in bytes (equivalent to one disk sector)

To - CPU overhead to initiate a one block disk read in seconds

Explain your formula to ensure that you get partial credit.

c. (6 points) Remote user-level NFS server:

Using NFS, a client can send a single request to an NFS server to read the first data block from a remote file.

Assume the network is reliable (no acknowledgements are used), B is smaller than the network's maximum packet size, and the following constants:

Ty - CPU cycle time in nanoseconds

Trblk - Average time to read a single disk block at the remote server

F - File size in bytes

R - Number of disk blocks accessed to read the first block of a file on the remote server

B - Filesystem block size in bytes

To - CPU overhead of reading or writing a packet in seconds

Tn - Latency of the network between the client and server in seconds

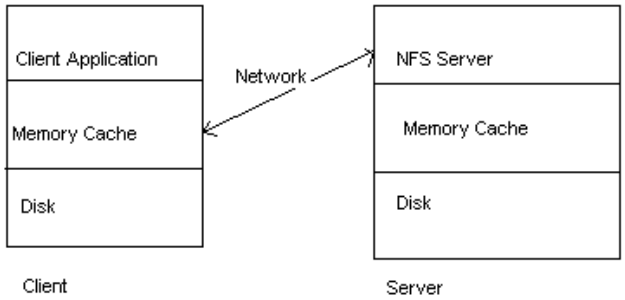
BWn - Network bandwidth between the client and server in bytes/second

Sr - The size of the block read request sent to the remote machine)

Using the above constants, specify the time to read the first data block from a remote user-level NFS filesystem?

d. (9 points) Local versus Remote Disk Access:

This problem explores and compares the *read-only* performance of local file access using a memory cache for the disk, with a remote fileserver with a memory cache.




Use the following costs for actions (assume there is no overhead to check the cache):

- Time to read a block from the client machine memory cache is 1 millisecond
- Time to read a block from the client machine disk is 20 ms
- Client machine hit rate for reading blocks from cache is 50%
- Time to read a block from the memory cache at the server machine is 1 ms
- Time to read a block from the disk at the server is 20 ms
- Time for either the client or the server to send a message on the network is 2.5 ms
- Time to receive am message at the client or server is 0 ms
- Server hit rate for reading blocks from the cache is Hr

i) Determine Tc the average time for the client application to read a single block using only the client machine memory cache and disk:

ii) Using the following access hierarchy, determine Tcs, the average time to read a single block:

- Check client machine memory cache
- If not present in cache, contact remote server
- At remote server, check memory cache
- If not present in cache, retrieve from disk and return block to client

iii) Compute the minimum value of Hr that makes the remote access faster than local filesystem accesses (ie determine Hr such that Tcs = Tc):

**Problem 5**. (15 points total) Security and Two-Phase Commit.

Suppose that CIA headquarters in Washington (W) communicates with branch offices in Moscow (M) and Beijing (B) using private-key-encrypted, reliable messages. Suppose furthermore that the

encryption key used is changed each day; every evening the new encryption key that is to be used the next day is sent from Washington to the two branch offices. Two-phased commit is used in order to ensure that, despite computer crashes, either (a) everyone eventually switches to the new encryption key, NEWKEY, or (b) no one switches to the new encryption key and everyone continues using OLDKEY. Assume that a machine takes *some* time to reboot and recover after a crash.

The steps involved in implementing two-phased committ are listed below, in time order:

1. W: write "begin transaction" to its log
2. W -> M: "New key is NEWKEY." (-> means W sends message to M)
3. M: write "new key is NEWKEY." to its log
4. M -> W: "Prepared to committ"
5. W -> B: "New key is NEWKEY."
6. B: write "New key is NEWKEY." to its log
7. B -> W: "Prepared to commit"
8. W: write "New key is NEWKEY." to its log
9. W: write "commit" to its log
10. W -> M: "commit"
11. M: write "got commit" to its log
12. M: Key = NEWKEY
13. M -> W: "ok"
14. W -> B: "commit"
15. B: write "got commit" to its log
16. B: Key = NEWKEY
17. B -> W: "ok"
18. W: Key = NEWKEY

a. (3 points) If W crashes after step 9 and no one else fails, what key will everyone end up using, once W reboots and recovers? *Give the reason why.*

b. (3 points) If M crashes after step 4 and no one else fails, what key will everyone end up using, once M reboots and recovers? *Give the reason why.*

c. (3 points) If B crashes after step 4 and no one else fails, what key will everyone end up using, once B reboots and recovers? *Give the reason why.*

d. (3 points) If M crashes after step 10, what recovery steps must it take after it reboots in order to achieve the correct global state with respect to which encryption key to use?

e. (3 points) If W crashes after step 11, what recovery steps must it take after it reboots in order to achieve the correct global state with respect to which encryption key to use?