# CS 2200 Spring 2017 Final (B) 8 AM to 10 AM

Name:____________________GT Number: _______________

| Problem | Points/Minutes | Lost | Gained |
|---|---|---|---|
| 0 | 0 (+1) | | |
| 1 | 14 (15 mins) | | |
| 2 | 18 (20 mins) | | |
| 3 | 19 (20 mins) | | |
| 4 | 25 (25 mins) | | |
| 5 | 24 (25 mins) | | |
| Total | 100 (+1) (105 mins) | | |

**You may ask for clarification but you are ultimately responsible for the answer you write on the paper. If you make any assumptions state them.**

**Please look through the entire test before starting. WE MEAN IT!!!**

**NOTE: M = $10^6$ K = $10^3$ Mi = $2^{20}$ Ki = $2^{10}$**
**Illegible answers are wrong answers.**

**Show your work in the space provided to get any credit for problem-oriented questions.**

**Good luck!**

0. (1 bonus point, 1 min)

Given:
   5+3+2 = 151022
   9+2+4 = 183652
   8+6+3 = 482466
   5+4+5 = 202541

What is
   8+2+4 = ___________

**Cache**

# CS 2200 Spring 2017 Final (B) 8 AM to 10 AM

Name:____________________GT Number: ______________

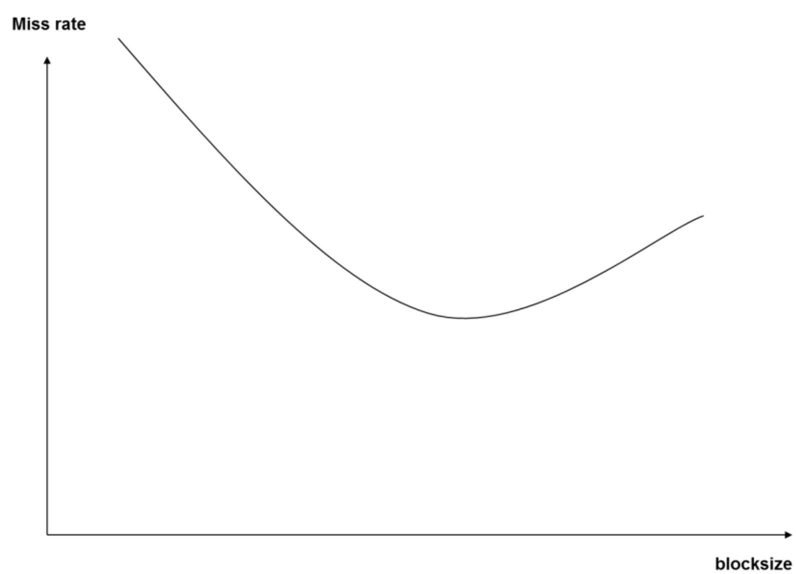
1. (14 points, 15 min)

a) (2 points) We have a 4-way set associative cache. How much meta-data is needed per cache line (*aka* set) to implement a TRUE LRU replacement policy? Show your work to get any credit.

b) (2 points) For a 32-way set associative cache, we want to implement an approximate LRU policy that protects the datablock that was most recently referenced. Suggest a scheme for doing this. How much meta-data is needed per cache line to implement your scheme? Show your work to get any credit. Your objective is to minimize the amount of meta-data needed per cache line.

c) As shown in the figure below, for a given cache size increasing the blocksize decreases the miss rate up to a certain point.

(i) (1 point) What is the principle that makes this possible?




(ii) (1 point) In the same figure, beyond a certain point, the miss rate starts increasing beyond a certain blocksize. What is the reason for that phenomenon?

d) (3 points) Choose between **conflict miss**, **capacity miss**, and **cold/compulsory miss** for each of the situation described below.

i. There is a cache miss. The cache is full. The missing memory block is being brought for the first time into the cache. This kind of a miss is termed ______________________________
ii. There is a cache miss. The cache is NOT full. The missing memory block used to be in the cache before. To find a place to put this word in the cache, an existing datablock has to be evicted. This kind of a miss is termed ______________________________
iii. There is a cache miss. The cache is full. The missing memory block used to be in the cache before. This kind of a miss is termed ________________________

e) (2 points) Page coloring is a technique for (circle the right choice)

i. Reducing page faults
ii. Increasing the associativity of a virtually indexed physically tagged cache
iii. Increasing the blocksize of a virtually indexed physically tagged cache
iv. Increasing the number of cachelines in a virtually indexed physically tagged cache

f) (1 point) Consider a cache that has 4 memory words in a datablock, and maintains a dirty bit per word. Upon eviction of a datablock from the cache (circle the correct choice)

i. One word is written back to the memory
ii. No word is written back to the memory
iii. 4 words are written back to the memory
iv. The specific words whose dirty bits are set are written back to the memory

g) (2 points) Consider a cache that has 4 memory words in a datablock. The processor has a memory bus that can transfer only ONE word to/from the memory in one memory bus cycle. Upon a cache read miss (circle ALL the correct choices). (+1 for correct choice; -1 for incorrect choice)

i. Only the missing word is brought from the memory into the cache.
ii. The entire memory block containing the missing word is brought into the cache.
iii. The first word brought into the cache is the missing word.
iv. The first word brought into the cache is the first word of the missing memory block.

# CS 2200 Spring 2017 Final (B) 8 AM to 10 AM

Name:____________________GT Number: ________________

**Input/Output and Disk**

2. (18 points, 20 min)

a) (2 points)

In a computer system, the CPU is faster than any I/O device connected to it. Why don't we use programmed I/O to transfer to or from ALL I/O devices? (**Two concise bullets please**)

b) (6 points)

Given the following:

```
Total number of cylinders in the disk  =  200 (numbered 0 to 199)
        (0 is the outermost cylinder and 199 is the innermost)
Current head position                  =  cylinder 17
        (direction of head motion is towards cylinder 0)
Current requests in order of arrival   =  20, 18, 55, 15, 25, 78, 99
                                          R1, R2, R3, R4, R5, R6, R7
```

(i) If the scheduling algorithm is LOOK, what is the expected response time (expressed in number of cylinders traversed) for request R6? (show your work)

(ii) What is the total head traversal (in terms of number of cylinders traversed) if the scheduling algorithm is SSTF? (show your work)

c) (10 points total) Given the following specification for a disk drive:

```
Average seek time    =  4 ms
Rotational speed     =  7500 RPM
Platters             =  3
Surface per platter  =  2
Tracks per surface   =  8000
Sectors per track    =  1024
Recording density    =  256 bytes per sector
```

(i) (2 points)How much time (expressed in **ms**) is needed to get to a random sector on the disk?

(ii) (2 point) How much time (expressed in **ms**) is needed to read one random sector from the disk when the head is already positioned on the desired sector?

(iii) (2 points) If the disk gets a request to read 6 random sectors, how much total time (expressed in **ms**) will that request take to complete?

(iv) (2 points) If the disk gets a request to read 6 consecutive sectors, how much total time (expressed in **ms**) will that request take to complete?

(v) (2 points) What is the transfer rate of the disk?

# CS 2200 Spring 2017 Final (B) 8 AM to 10 AM

Name:____________________GT Number: _______________

**File Systems (see last page for a cheat sheet on Unix FS commands)**

3. (19 points, 20 min)

a) (1 point)

The current state of a file "foo" is as follows:

```
    ---xr-x---   3 rama            0 Apr 27 21:01 foo
```

After executing the following commands:

```
chmod w+w foo
chmod u+w foo
```

The new access rights are (circle the right choice):

(i) --wxrwx-w-
(ii) -rwxr-x---
(iii) --wxr-x-w-
(iv) ---xr-x-w-

b) (8 points) In the following table, assume **none of the files exist to start with** in the current directory. Fill in the table. The reference count in the table pertains to the i-node that is affected by the command in that row. If a new i-node is created, show the old reference count for that i-node as 0.

| **COMMAND** | NEW I-NODE CREATED (YES/NO) | REFERENCE COUNT OLD | NEW |
|---|---|---|---|
| **touch f1** | | | |
| **touch f6** | | | |
| **ln f1 f2** | | | |
| **ln –s f1 f3** | | | |
| **ln –s f3 f4** | | | |
| **ln f6 f5** | | | |
| **rm f6** | | | |
| **rm f2** | | | |

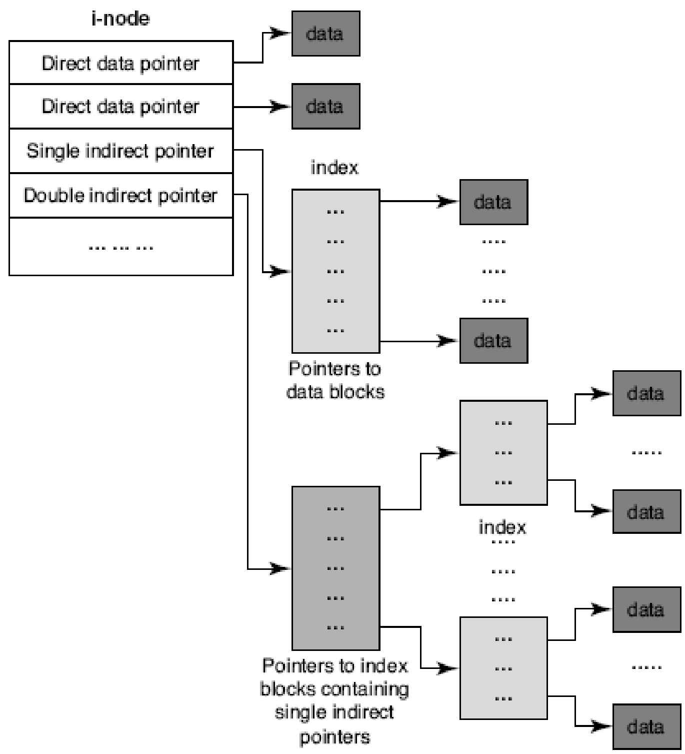
Use this area for rough work for this question

Name:____________________GT Number: ________________

c) (**NOTE Ki = $2^{10}$ Mi = $2^{20}$**)

Given the following:
  Size of index block = 256 bytes
  Size of Data block = 4096 bytes
  Size of pointer = 8 bytes (to index or data blocks)
  The i-node consists of
    2 direct data block pointers,
    1 single indirect pointer, and
    1 double indirect pointer.

Note that the index blocks and data blocks are allocated on a need basis. An index block is used for the top-level i-node as well as for the index blocks that store pointers to other index blocks and data blocks (see Figure).




i. (2 points) How many data blocks are used to store a 100 KiB file?

ii. (2 points) How many index blocks (including the i-node for the file) are needed to store a 1 MiB file?

iii. (2 points) What is the largest file size that can be supported in this file system?

d) (4 points) Strictly Contiguous allocation of disk blocks (using a best-fit allocation policy depending on expected file size) results in (circle ALL the TRUE statements in the following list) (+1 for correct choice; -1 for incorrect choice)

(i) External Fragmentation
(ii) Internal fragmentation
(iii) Ability to grow the file easily
(iv) Minimal time overhead for allocation
(v) Good sequential access to the files
(vi) Good random access to the files
(vii) Simple data structures for file maintenance
(viii) Very good space efficiency

# CS 2200 Spring 2017 Final (B) 8 AM to 10 AM

Name:____________________GT Number: ________________

**Parallel Systems**

4. (25 points, 25 mins)

a) (3 points)

(fill in selecting the appropriate phrases from these choices: **threads, process, processes, thread, memory footprint, VPN, PFN**

The state of a multi-threaded program is fully defined by its _____________________ and the state of the ___________________ executing within the _____________________.

b) (2 points)

You click on the icon of a multi-threaded program.
A program is currently in execution with 4 threads in it.
The OS creates a memory footprint of the program reading it from the disk.
Pictorially show this memory footprint.

c) (4 points)

Given the code fragment:

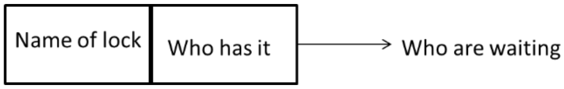
```
acquire_shared_resource()
{
  thread_mutex_lock(cs_mutex);
    if (res_state == BUSY)
      thread_cond_wait (res_not_busy, cs_mutex);
    res_state = BUSY;
  thread_mutex_unlock(cs_mutex);
}
```

A thread wakes up from the cond_wait shown in the code above.

(i) What are the invariants expected by this thread?

(ii) Who ensures these invariants are satisfied?

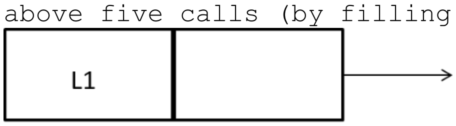
d) (5 points) The internal representation in the thread library for a LOCK is shown below:

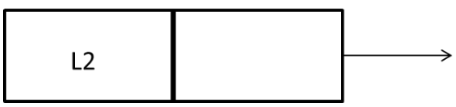



Assume that the following events happen in the order shown (T1-T5 are threads of the same process):

- T1 executes thread_mutex_lock(L2);
- T2 executes thread_mutex_lock(L2);
- T3 executes thread_mutex_lock(L1);
- T4 executes thread_mutex_lock(L2);
- T5 executes thread_mutex_lock(L1);

Assuming that there have been no other calls to the threads library prior to this, show the state of the internal data structures of L1 and L2 after the above five calls (by filling in the figures below).




e) (6 points)
A multithreaded process is executing on an SMP which supports hardware cache coherence and per processor TLB. Each thread is on a distinct processor.
(i) All threads have to execute within the same address space. How is this ensured?

(ii) All the threads have to see the same mapping from VPN to PFN regardless of which processor they are executing on. How is this ensured?

(iii) The OS implements mutual exclusion lock using a Test-and-Set instruction available in the processor architecture. The threads have to have synchronization atomicity for a given lock variable regardless of which processor they are executing on. How is this ensured?

(iv) All the threads have to see the same value in a given memory location. How is this ensured?

# CS 2200 Spring 2017 Final (B) 8 AM to 10 AM

Name:____________________GT Number: ________________

f) (5 points)

Given the following details about an SMP (symmetric multiprocessor):

Cache coherence protocol: write-invalidate

Cache to memory policy: write-back

Initially:

The caches are empty

Memory locations:

**A contains 20**

**B contains 18**

Consider the following timeline of memory accesses from processors P1, P2, and P3.

| Time (in increasing order) | Processor P1 | Processor P2 | Processor P3 |
|---|---|---|---|
| T1 | Load A | | |
| T2 | | Load B | |
| T3 | | | Load A |
| T4 | Store #80, A | | |
| T5 | | Store #30, B | |

Using the table below, summarize the activities and the values in the caches and memory.

| Time | Variables | Cache of P1 | Cache of P2 | Cache of P3 | Memory |
|---|---|---|---|---|---|
| T1 | A<br><br>B | | | | |
| T2 | A<br><br>B | | | | |
| T3 | A<br><br>B | | | | |
| T4 | A<br><br>B | | | | |
| T5 | A<br><br>B | | | | |

# CS 2200 Spring 2017 Final (B) 8 AM to 10 AM

Name:____________________GT Number: ________________

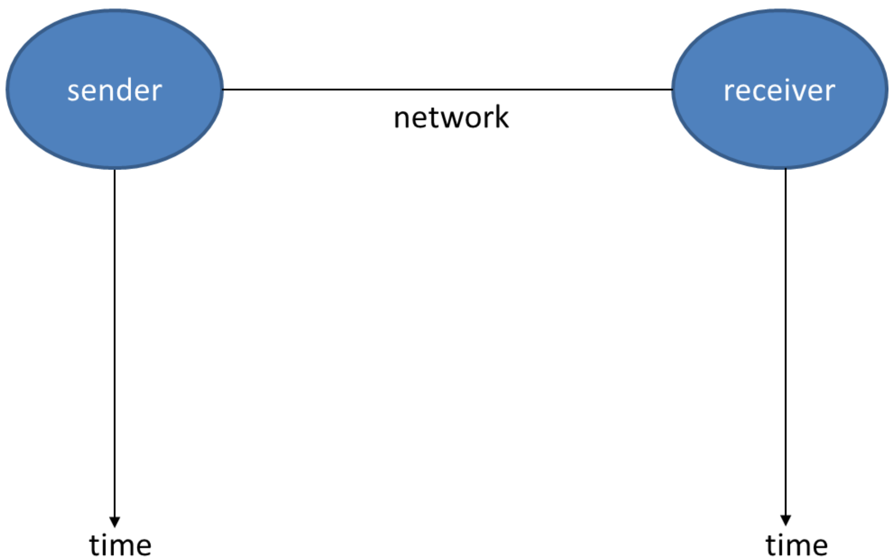
**Networking**

5. (24 points, 25 min)

a) (3 points) The one-bit sequence number requirement for the "Stop-and-wait protocol (aka alternating bit protocol)" is predicated on the following assumption:

- packets do not get reordered in the network OR arbitrarily delayed

Show using the space-time diagram below what can go wrong if this assumption does not hold.




b) (2 points) What determines the timeout for packet retransmission in the transport protocol? (**one concise bullet please**)

c) (5 points) Match the left side to the right side with respect to transport protocols (**draw an arrow from the concern on the left side to the correct solution on the right side**)

| **Concerns or Issues** | **Solution** |
|---|---|
| Arbitrary message size | Sequence numbers |
| Out of order delivery | Checksum |
| Packet loss | Windowing mechanism |
| Bit errors in transmission | Multiple packets |
| Queueing delays | Positive ACKs and Buffering |

d) (2 points) (**Answer True/False with justification**)
With reference to the packet header generated by the transport layer shown below

```
struct header_t {
  int destination_address;
  int source_address;
  int num_packets;
  int sequence_number;
  int packet_size;
  int checksum;
};
```

It is necessary to have the field "num_packets" in the header only in the first packet.

e) (2 points) Why do we need MAC address when each node on the Internet has an IP address?

f) (2 points) (**Answer True/False with justification**) The load on a LAN connecting computers is very light. Which link layer protocol (Ethernet or Token ring) would you recommend and why?

g) (1 point)With respect to this figure

The number of collision domains (circle the right choice):

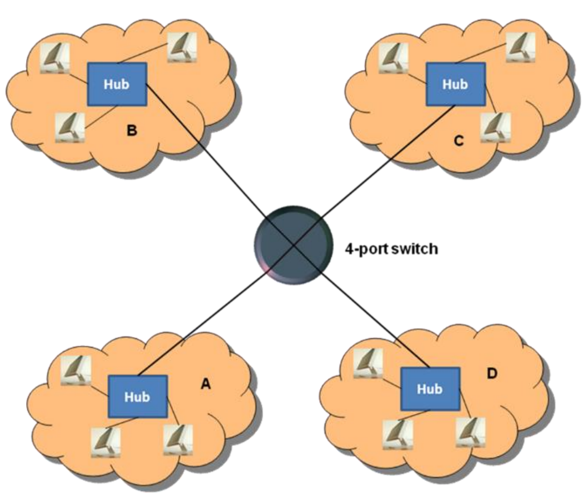
(i) One

(ii) Two

(iii) Three

(iv) Four

(v) Twelve



# CS 2200 Spring 2017 Final (B) 8 AM to 10 AM

Name:____________________GT Number: _______________

h) (2 points) (**Answer True/False with justification**) Assume computers are connected by Ethernet as the link layer protocol. TCP/IP is used for them to communicate on the Internet. Since Ethernet does collision detection there is no need to check for any packet corruption in the Transport layer.

i) (1 point) In IPv4, an IP address is given by the notation:

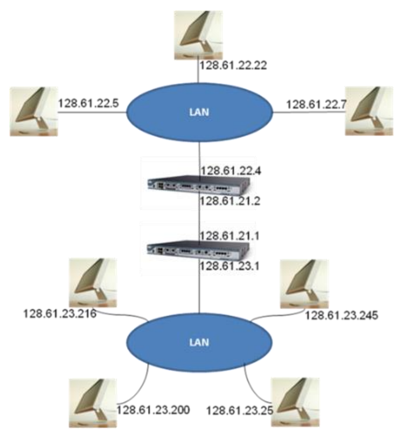
a.b.c.d

The number of bits to represent this IP address is (**circle the right choice**):

(i) 64 bits
(ii) 32 bits
(iii) 128 bits
(iv) 256 bits

j) (2 points) Given the figure:

Identify the distinct IP networks in the figure by **drawing an oval** around each




k) (2 points)
Given the following:

| | | |
|---|---|---|
| Message size | = | 20,000 bits |
| Bandwidth on the wire | = | 200,000 bits/sec |
| Time of flight | = | 10 msecs |
| Sender overhead | = | 1 ms |
| Receiver overhead | = | 1 ms |

Assuming no packet loss, compute the throughput.

# CS 2200 Spring 2017 Final (B) 8 AM to 10 AM

Name:____________________GT Number: ________________

| Unix command | Semantics | Elaboration |
|---|---|---|
| **touch <name>** | **Create a file with the name <name>** | Creates a zero byte file with the name <name> and a creation time equal to the current wall clock time |
| **mkdir <sub-dir>** | **Create a sub-directory <sub-dir>** | The user must have write privilege to the current working directory (if <sub-dir> is a relative name) to be able to successfully execute this command |
| **rm <name>** | **Remove (or delete) the file named <name>** | Only the owner of the file (and/or superuser) can delete a file |
| **rmdir <sub-dir>** | **Remove (or delete) the sub-directory named <sub-dir>** | Only the owner of the <sub-dir> (and/or the superuse) can remove the named sub-directory |
| **ln –s <orig> <new>** | **Create a name <new> and make it symbolically equivalent to the file <orig>** | This is name equivalence only; so if the file <orig> is deleted, the storage associated with <orig> is reclaimed, and hence <new> will be a dangling reference to a non-existent file |
| **ln <orig> <new>** | **Create a name <new> and make it physically equivalent to the file <orig>** | Even if the file <orig> is deleted, the physical file remains accessible via the name <new> |
| **chmod <rights> <name>** | **Change the access rights for the file <name> as specified in the mask <rights>** | Only the owner of the file (and/or the superuser) can change the access rights |
| **chown <user> <name>** | **Change the owner of the file <name> to be <user>** | Only superuser can change the ownership of a file |
| **chgrp <group> <name>** | **Change the group associated with the file <name> to be <group>** | Only the owner of the file (and/or the superuser) can change the group associated with a file |
| **cp <orig> <new>** | **Create a new file <new> that is a copy of the file <orig>** | The copy is created in the same directory if <new> is a file name; if <new> is a directory name, then a copy with the same name <orig> is created in the directory <new> |
| **mv <orig> <new>** | **Renames the file <orig> with the name <new>** | Renaming happens in the same directory if <new> is a file name; if <new> is a directory name, then the file <orig> is moved into the directory <new> preserving its name <orig> |
| **cat/more/less <name>** | **View the file contents** | |