General Information:
This is a closed book examination with two 2-side pages of handwritten notes. You have 170 minutes to answer as many questions as possible. The number in parentheses at the beginning of each question indicates the number of points given to the question; there are 100 points in all. You should read all of the questions before starting the exam, as some of the questions are substantially more time consuming.

Write all of your answers directly on this paper. Make your answers as concise as possible. If there is something in a question that you believe is open to interpretation, then please ask us about it!

Good Luck!!

<table>
<thead>
<tr>
<th>Problem</th>
<th>Max Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
1. (10 points) Multiple choices:
   a. (2 points) Which of the following best describes Belady’s anomaly?
      A. Use of a TLB can sometimes slow down page lookup.
      B. Page faults occur with the optimal page replacement algorithm.
      C. FIFO page replacement can perform as well as optimal page replacement.
      D. More page frames can result in more page faults.
      E. None of the above.

      Your Answer: ____ D

   b. (2 points) Which of the following statements is true about a packet switching network?
      A. It provides reliability.
      B. It provides low delay.
      C. Each packet of a flow can follow a different path in the network.
      D. Each packet of a flow must follow the same path in the network.
      E. None of the above.

      Your Answer: ____ C

   c. (2 points) Which of the properties below is true for 2 Phase Locking (2PL)?
      A. Guarantees the operations of transactions are not interleaved.
      B. Guarantees there is no cascading abort.
      C. Guarantees the transactions do not deadlock.
      D. Guarantees conflict serializability.
      E. None of the above.

      Your Answer: ____ D
d. (2 points) Which of the properties below is false for an SSD:
   A. SSD reads always outperform reads to a spinning media hard drive
   B. SSD writes always outperform writes to a spinning media hard drive
   C. Reads are faster than writes.
   D. SSDs are more expensive per byte than hard drives.
   E. None of the above.

   Your Answer: _____ B


e. (2 points) Which of the following are advantages of I/O using DMA over interrupt-driven I/O?
   I. Fewer interrupts.
   II. Frees the main CPU to do other work.
   III. DMA requires less hardware.
   A. II and III.
   B. I and II.
   C. I only.
   D. II only.
   E. I, II, and III.
   F. None of the above.

   Your Answer: _____ B
2. (15 points) Address Translation.
   a. (10 points) You’re hired by the HAL Corporation to develop their next generation machine, the HAL 2012. It will have a virtual memory architecture with the following parameters:
      • Virtual addresses are 56 bits.
      • The page size is 64K byte.
      • The architecture allows a maximum 128 Terabytes (TB) of real memory (RAM).
      • The first- and second-level page tables are stored in real memory.
      • All page tables can start only on a page boundary.
      • Each second-level page table fits exactly in a single page frame.
      • **There are only valid bits and no other extra permission, or dirty bits.**

   Your job is to draw and label a figure showing how a virtual address gets mapped into a real address. You should list how the various fields of each address are interpreted, including the size in bits of each field, the maximum possible number of entries each table holds, and the maximum possible size in bytes for each table (in bytes). Also, your answer should indicate where checks are made for faults (e.g., invalid addresses).

   **The offset is 16 bits.**

   *Each second-level page table has \(2^{14}\) (16K) entries. The page size is 64KB and the second-level PTE has a PPN of 31 bits plus a valid bit, which fits nicely in a 32 bit word or 4 bytes. A 64KB page can thus hold 16K 4-byte entries (4 bytes times 16K is 64KB).*

   *The first-level page table has \(2^{26}\) (64M) entries, each of which is 32 bits (31-bit PPN plus valid bit, or 4 bytes), so it has a maximum size of 256 MBytes.*

   *If you didn’t draw a figure, we deducted 3 points. If you didn’t check the valid bits, had the wrong page or offset field widths, your second level page table was the wrong size, you included extra bits in the page tables, or you did not specify the number of page table entries, we deducted 2 points (per error). If you used segments, we deducted five points. If you included extraneous checks, we deducted 1 point.*
b. (5 points) To improve the HAL 2012’s performance, you decide to add a TLB to the HAL 2012’s virtual memory architecture. The TLB will be 4-way set associative and have 4096 rows in each set. Draw and label a diagram of the TLB, showing the size of each field in the TLB. Indicate how bits of the virtual address are used as input to the TLB, and describe the outputs from the TLB. Include a read enable bit and write enable bit for each page in your TLB.

Since the TLB has 4,096 rows in each set and the offset is 16 bits, we need 12 bits to index the appropriate row, and we have 28 bits as the tag field.
If your answer was missing bits in the virtual address (e.g., no tag field) or the size of the tag was wrong, we deducted one point. If your answer didn’t include a figure, your index and tag scheme was incorrect or missing, or your solution didn’t include four sets, we deducted three points for each error. If your answer didn’t include a reasonable index scheme or you incorrectly used a 4-way set associative cache, we deducted two points.
3. (10 points) Address Translation 2. Consider a multi-level memory management using 24-bit virtual address with the following format:

<table>
<thead>
<tr>
<th>Virtual seg #</th>
<th>Virtual Page #</th>
<th>Offset</th>
</tr>
</thead>
</table>

Each virtual address has 4 bits of virtual segment #, 8 bits of virtual page #, and 12 bits of offset. Page table entries are 8 bits. The segment table and physical memory contents are below. All values are in hexadecimal, and there are no permission bits associated with the page table entries.

### Segment Table

<table>
<thead>
<tr>
<th>Entry</th>
<th>Start</th>
<th>Size</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x2004</td>
<td>0x40</td>
<td>Valid, read only</td>
</tr>
<tr>
<td>1</td>
<td>0x0000</td>
<td>0x10</td>
<td>Valid, read/write</td>
</tr>
<tr>
<td>2</td>
<td>0x2040</td>
<td>0x40</td>
<td>Invalid</td>
</tr>
<tr>
<td>3</td>
<td>0x1010</td>
<td>0x10</td>
<td>Valid, read/write</td>
</tr>
</tbody>
</table>

### 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 | 7D | 7E | 7F | 80 | 81 |

Translate the following virtual addresses into physical addresses. If the translation causes any type of error, simply write Error!:

<table>
<thead>
<tr>
<th>Virtual Address</th>
<th>Physical Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x104CBF</td>
<td>0x12CBF</td>
</tr>
<tr>
<td>0x204ABC</td>
<td>Error!</td>
</tr>
<tr>
<td>0x3045AB</td>
<td>0x125AB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Virtual Address</th>
<th>Physical Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x232125</td>
<td>Error!</td>
</tr>
<tr>
<td>0x109933</td>
<td>0x17933</td>
</tr>
<tr>
<td>0x010DEF</td>
<td>0x16DEF</td>
</tr>
</tbody>
</table>
4. (15 points) Concurrency.

Consider the following function that swaps two elements in the same array. The array is indexed from 0. Assume that only the memory loads and stores are atomic, i.e., assume each instruction in `swap_array()` is atomic.

```c
void swap_array(int a[], int i, int j) {
    int tmp;
    tmp = a[j];
    a[j] = a[i];
    a[i] = tmp;
}
```

a) (5 points) Assume input array `A[3] = {1, 2, 3}`, and assume that `swap_array(A, 0, 1)`, and `swap_array(A, 1, 2)`, respectively, are called each in a different thread. What are the possible output values of `A[]` after the two threads finish the execution?

```
Thread 1
swap_array(A[], 0, 1) {
    tmp = A[1];
    A[0] = tmp;
}
Thread 2
swap_array(A[], 1, 2) {
    tmp = A[2];
}
```

Note that we have race conditions only on `A[1]` (i.e., both threads write/read `A[1]`), and that `tmp` is a local variable allocated on stack, and thus each thread will have its own copy.

```
Thread 1
tmp1 = A[1];
A[1] = 1;
A[0] = tmp1;
```
```
Thread 2
tmp2 = 3;
```

Which further can be simplified as:

```
Thread 1
tmp1 = A[1];
A[1] = 1;
A[0] = tmp1;
```
```
Thread 2
A[1] = 3;
```

So there are 6 possibilities:


b) (5 points) Assume \( \text{swap}_\text{array}(A, 0, 1, ) \) and \( \text{swap}_\text{array}(A, 1, 2, ) \), respectively, are called each in a different thread and the output is \( A[3] = \{1, 2, 3\} \). What are the possible input values of \( A[] \)?

Assume \( A = \{a, b, c\} \). Then from previous point, we have \( = \{b,c,a\}, \{c,a,b\},\{c,a,a\},\{b,a,b\},\{b,c,c\} \). The output has 3 distinct values, and thus only the output \( \{b, c, a\} \) and \( \{c,a,b\} \) are the valid one. Using substitution, we have the following inputs: \( A = \{3,1,2\} \), and \( A=\{2,3,1\} \)

c) (5 points) Assume an initial array \( A[3] = \{1, 2, 3\} \). Add synchronization primitives to the \( \text{swap}_\text{array}() \) code to guarantee that after the execution of \( \text{swap}_\text{array}(A, 0, 1) \) and \( \text{swap}_\text{array}(A, 1, 2) \), respectively, the output is either \( A = \{2, 3, 1\} \) or \( A = \{3, 1, 2\} \).

Just apply a lock on all instructions in \( \text{swap}_\text{array}() \), i.e.,

\[
\text{void swap}_\text{array}(\text{in } a[], \text{ int } i, \text{ int } j) \{ \\
\text{ int tmp; } \\
\text{ acquire(} &\text{lock); } \\
\text{ ... } \\
\text{ release(} &\text{lock); } \\
\}
\]
No Credit – Problem X (000000000000 points)

"Never, ever, think outside the box."

By Leo Cullum at The New Yorker, 1998
5. (12 points) Demand paging.

Consider the following page reference string:

A, B, A, C, D, A, D, E, C, D, A, B

We have 3 physical pages, and all are initially all empty. For each of the following page replacement algorithms list whether a page fault occurs on a particular page reference. For grading purposes, the ordering of pages within the 3 available frames does not matter. Mark pages whose reference bit is set using an asterisk (*).

Fill out the tables below and specify the number of page faults:

<table>
<thead>
<tr>
<th>Least Recently Used (6 pts)</th>
<th>Second Chance (6 pts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Refs</td>
<td>Fault? (Y/N)</td>
</tr>
<tr>
<td>A</td>
<td>Y</td>
</tr>
<tr>
<td>B</td>
<td>Y</td>
</tr>
<tr>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>C</td>
<td>Y</td>
</tr>
<tr>
<td>D</td>
<td>Y</td>
</tr>
<tr>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>D</td>
<td>N</td>
</tr>
<tr>
<td>E</td>
<td>Y</td>
</tr>
<tr>
<td>C</td>
<td>Y</td>
</tr>
<tr>
<td>D</td>
<td>N</td>
</tr>
<tr>
<td>A</td>
<td>Y</td>
</tr>
<tr>
<td>B</td>
<td>Y</td>
</tr>
</tbody>
</table>

-1 for each incorrect entry in either LRU or Second Chance (up to 6 point off for each).
6. (11 points) UNIX filesystems.
   a. (5 points) Consider a UNIX system with the following components:
      • Disk blocks are 1024 bytes. Sectors are 512 bytes long.
      • The inumbers are 4-bytes long.
      • An inode contains file attributes and a total of 14 block pointers, including
twelve direct pointers, one single indirect entry, and one double indirect
entry. The total inode size is 256 bytes.
      • Both indirect and double indirect blocks take up an entire disk block.

   How many disk reads are needed to read the 300th block of file A into memory,
assuming initially only A’s inode is in memory? Show your reasoning/steps for
partial credit.

   One index block contains 1KB/4B=256 entries. The direct entries and the single
indirect entry can support up to 10 + 256 blocks, which is less than 300 blocks.
So we need to use the double indirect entry. Thus, the number of disk reads is 3 –
one read for the double indirect block, one read for the single indirect block, and
one read for the data block.

b. (6 points) Using the components described above, list the set of disk blocks that
must be read into memory in order to read the 12,000-byte long UNIX file
/usr/cs162/final.doc in its entirety. Assume no inodes or directories are
currently in memory, and that the directories in question all fit into a single disk
block each. (Note: this is not always true in reality.)

   (1) Read in file header for / (root). The root is always at fixed spot on disk.
   (2) Read in first data block for / (root).
   (3) Read in file header for usr.
   (4) Read in first data block for usr.
   (5) Read in file header for cs162.
   (6) Read in first data block for cs162.
   (7) Read in file header for final.doc.
   (8) Read in first data block for final.doc.
   (9 – 17) Read in second through 12th data blocks for final.doc.
   (Note: 12th data block will be only partially full.)
7. (12 points total) Security.

Itsy Bitsy Machines Corporation has just released a new e-mail program that allows people to “sign” their email. The program does this by including, at the bottom of an otherwise normal email message: the sender’s name and the date encrypted in the sender’s private key. The message itself is unencrypted, but a receiver can validate the signature by doing a “finger” of the sender to retrieve their public key. The finger server uses an unencrypted communication protocol.

Explain why this gives a false sense of security, by outlining two (2) different ways that you could make it appear that Larry Ellison signed mail saying “Bill Gates is a fink”. The definition of “different” is that each attack has a unique fix. For each of the three attacks you list, give a countermeasure that the sender/receiver could take to protect themselves against just that one attack, where the countermeasure would not help against any of the other attacks you list.

You may assume that the sender and receiver are on different machines, that both are running on “diskless” workstations whose files are provided by NFS, and that you have the ability to spy on and/or alter packets on any network at the sender or receiver’s site.

**Attack #1 and Countermeasure #1 (6 points):**

*Using a man-in-the-middle attack on a valid message from Ellison to Gates, the attacker can replace the text of a valid message from Ellison with the “Bill Gates is a fink.” message. The countermeasure is to generate a cryptographic hash of the entire message and include it in the signature of the message (i.e., encrypted with Ellison’s private key).*

**Attack #2 and Countermeasure #2 (6 points):**

*Using a man-in-the-middle attack on the request from Gates to the finger server, the attacker can replace the public key returned by the server with their own. The countermeasure is to use a certificate authority to distribute signed keys. This approach is vulnerable to compromise of the CA.*
8. (15 points) Transactions
   a. (5 points) Consider the following schedule of three transactions. Is this schedule conflict serializable or not? Briefly explain your answer (no more than two sentences). Recall that a schedule is conflict serializable if there is an equivalent schedule (i.e., a schedule generating the same output) in which transactions runs serially.

   T1: \( R(A), W(A), R(B), W(B) \)
   T2: \( R(A), W(A) \)
   T3: \( R(A), R(B), W(B), W(A) \)

   In the dependency graph:
   (a) There is an edge from T1 to T2, because \( W(A) \) of T1 is in conflict with \( R(A) \) of T2.
   (b) There is an edge from T2 to T3, because \( W(A) \) of T2 is in conflict with \( R(A) \) of T3.
   (c) There is an edge from T3 to T1, because \( W(B) \) of T3 is in conflict with \( R(B) \) of T1.

   Thus there is a cycle in the dependency graph which means that the schedule is not conflict serializable.

   b. (5 points) Assume T3 is aborted before committing (i.e., right after \( W(A) \)). Do any of the other transactions need to be aborted? If yes, which of these transactions need to be aborted? Briefly explain your answer (no more than two sentences)

   If T3 is aborted, then T1 needs to be aborted since it reads location B which was written by T3, and in turn T2 needs to be aborted since it reads location A, written by T2.
c. (5 points) Now, consider the same transactions with a slightly different schedule.

\[ T_1: \ \text{R}(A), \text{W}(A), \ \text{R}(B), \text{W}(B), \]  
\[ T_2: \ \text{R}(A), \text{W}(A), \]  
\[ T_3: \ \text{R}(A), \ \text{R}(B), \text{W}(B), \text{W}(A) \]

(i) (3 points) Is this schedule conflict serializable? Briefly explain your answers (no more than two sentences).

There is no cycle in the dependency graph, so this schedule is conflict-serializable.

(ii) (2 points) If T3 is aborted right before committing, does any of the other transactions need to be aborted? If yes, which? Briefly explain your answers (no more than two sentences).

If T3 is aborted, no other transaction needs to be aborted since none of the other transactions read values that have been changed by T3.