Please read through the entire examination first!
- You have 110 minutes for this exam. Don’t spend too much time on any one problem!
- The last page is a reference sheet. Feel free to detach it from the rest of the exam.
- The exam is CLOSED book and CLOSED notes (no summary sheets, no calculators, no mobile phones).

There are 8 problems for a total of 65 points. The point value of each problem is indicated in the table below. Write your answer neatly in the spaces provided.

Please do not ask or provide anything to anyone else in the class during the exam. Make sure to ask clarification questions early so that both you and the others may benefit as much as possible from the answers.

POINTS WILL BE DEDUCTED if you are writing/erasing after the final bell has rung!

Good Luck!

Your Name:__________________________
UWNet ID:__________________________

Name of person to your left | Name of person to your right
________________________ | _________________________

<table>
<thead>
<tr>
<th>Problem</th>
<th>Topic</th>
<th>Max Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Caches</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>Processes</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Virtual Memory</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Memory Allocation</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Java</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>Compilation</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>Representation</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>Assembly to C</td>
<td>9</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td><strong>65</strong></td>
</tr>
</tbody>
</table>
1. Caches (11 points)

You are using a byte-addressed machine where physical addresses are 22-bits. You have a 4-way associative cache of total size 1 KiB with a cache block size of 32 bytes. It uses LRU replacement and write-back policies.

a) Give the number of bits needed for each of these:

Cache Block Offset: ___________  Cache Tag: _____________

b) How many sets will the cache have? ___________

c) Assume that everything except the array x is stored in registers, and that the array x starts at address 0x0. Give the hit rate (as a fraction or a %) for the following code, assuming that the cache starts out empty. Also give the total number of hits.

```
#define LEAP 1
#define SIZE 256
int x[SIZE][8];
... // Assume x has been initialized to contain values.
... // Assume the cache starts empty at this point.
for (int i = 0; i < SIZE; i += LEAP) {
    x[i][0] += x[i][4];
}
```

Hit Rate: ___________  Total Number of Hits: ___________

d) If we increase the cache block size to 64 bytes (and leave all other factors the same) what would the hit rate be?

Hit Rate: ___________  Total Number of Hits: ___________

e) For each of the changes proposed below, indicate how it would affect the hit rate of the code above in part c) assuming that all other factors remained the same as they were in the original cache:

<table>
<thead>
<tr>
<th>Change associativity from</th>
<th>4-way to 2-way:</th>
<th>increase</th>
<th>/</th>
<th>no change</th>
<th>/</th>
<th>decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change LEAP from</td>
<td>1 to 4:</td>
<td>increase</td>
<td>/</td>
<td>no change</td>
<td>/</td>
<td>decrease</td>
</tr>
<tr>
<td>Change cache size from</td>
<td>1 KiB to 2 KiB:</td>
<td>increase</td>
<td>/</td>
<td>no change</td>
<td>/</td>
<td>decrease</td>
</tr>
</tbody>
</table>
For the program above, list all of the possible outputs.

Hint: `execv(path, arg)` - replaces current process image with a new image. 
`/bin/echo` simply prints the 2nd argument (in this case "Hello") to the screen.

Answer:
3. Virtual Memory (9 points)

Assume we have a virtual memory detailed as follows:

- 256 MiB Physical Address Space
- 4 GiB Virtual Address Space
- 1 KiB page size
- A TLB with 4 sets that is 8-way associative with LRU replacement

For the following questions it is fine to leave your answers as powers of 2.

a) How many bits will be used for:

Page offset? __________
Virtual Page Number (VPN)? ________  Physical Page Number (PPN)? ________
TLB index? ________________  TLB tag? ________________

b) How many entries in this page table?

c) We run the following code with an empty TLB. Calculate the TLB miss rate for data (ignore instruction fetches). Assume \texttt{i} and \texttt{sum} are stored in registers and \texttt{cool} is page-aligned.

```c
#define LEAP 8
int cool[512];
... // Some code that assigns values into the array cool
... // Now flush the TLB. Start counting TLB miss rate from here.
int sum;
for (int i = 0; i < 512; i += LEAP) {
    sum += cool[i];
}
```

**TLB Miss Rate:** (fine to leave you answer as a fraction) ______________
4. Memory Allocation (8 points)

a) In Garbage Collection, describe what it means (in 1-2 sentences) for a block to be “reachable”. Be specific.

b) TRUE / FALSE: In a C program, freeing the same address multiple times will be detected by the memory allocator and ignored.

c) The following two C functions have errors:

```c
int* foo() {
    int val;
    return &val;
}
```

What is the error? ____________________________________________

Why is this bad? ____________________________________________

```c
void bar() {
    int *x = (int *) malloc( 10 * sizeof(int) );
    return;
}
```

What is the error? ____________________________________________

Why is this bad? ____________________________________________

For bar, which of the following is most true (circle ONLY one):

i. This error will always be detected by the compiler.

ii. If this code runs, the error will always (eventually) cause the program to stop running unexpectedly.

iii. If this code runs, the error could potentially go undetected.
Given the class hierarchy above and the following additional code:

```java
class FinalExam {
    public static void main(String[] args) {
        Boat b1 = new Boat();
        PirateShip ps1 = new Boat();
        Boat b2 = new PirateShip();
        Vehicle v = new PirateShip();
        PirateShip ps2 = (PirateShip) b1;
        PirateShip ps3 = (PirateShip) v;
    }
}
```

a) Mark the appropriate column(s) of the table above to indicate if the line will cause a compiler and/or runtime error or no error.

b) Given our discussion in class, circle whether you would expect the following to be True or False:

i. TRUE / FALSE: A Car object will be the same size as a Boat object.

ii. TRUE / FALSE: A PirateShip object will be the same size as a Boat object.

iii. TRUE / FALSE: The vtable for a Car will be the same size as the vtable for a Boat.

iv. TRUE / FALSE: The vtable for a PirateShip will be the same size as the vtable for a Car.

v. TRUE / FALSE: The code for doWheelie will be on the heap.

c) Given: Vehicle v2 = new PirateShip();

v2.makeNoise(); will print ____________________________
6. Compiling and Running Programs (7 points)

a) Assume you were given a file `fact.c` identical to the one used in Homework 3, containing two functions `factorial` and `main`. Fill in the missing parts of the table below:

<table>
<thead>
<tr>
<th>Tool Name (gcc command)</th>
<th>Type of file Produced (Give a description, not just file name or extension)</th>
<th>Can you run this file directly (yes/no)?</th>
<th>Can you easily edit this file in a text editor (yes/no)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linker (<code>gcc fact.o</code>)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compiler (<code>gcc -S fact.c</code>)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assembler (<code>gcc -c fact.s</code>)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b) In C, who determines whether an array is allocated on the stack or the heap?

  - Programmer
  - Compiler
  - Language (Java) Runtime
  - Operating System

c) In C, who determines whether local variables are allocated on the stack or stored in registers?

  - Programmer
  - Compiler
  - Language (C) Runtime
  - Operating System
d) Who/what assigns process IDs to individual processes?

  - Programmer
  - Compiler
  - Language (C, Java) Runtime
  - Operating System
e) Who/what finds data in the L1 cache and brings it into a register?

  - Hardware
  - Compiler
  - Language (C, Java) Runtime
  - Operating System
7. Representation (6 points)

a) Given the following declaration:

```c
int x = ...; // x < 0
```

For each of the following, indicate if it is TRUE for all possible values of \( x < 0 \). **If not, select FALSE and give a BRIEF one sentence justification for your answer** – BE SPECIFIC. You do not need to give a justification for true answers.

i) \( x == (int)(float) x \)  
   **TRUE**  
   **FALSE**

ii) \( x == (int)(double) x \)  
    **TRUE**  
    **FALSE**

b) On a 64-bit word machine, you are given the following array declaration in C: `int a[6][3]`. If \( a \) starts at address 0, what will the expression `& (a[2][5])` evaluate to? (If “unknown” or “cannot be guaranteed”, state that. Otherwise give your answer in **decimal**.)

c) Given the following struct in x86-64:

```c
struct student {
    char name[10];
    int id;
    char color[7];
    double weight;
};
```

What is the total size of this struct in bytes?

As a programmer, could you have declared this struct differently so that it uses less memory? If no, **explain why not**. If yes, **show how you would declare it and give the new total size in bytes.**
8. Assembly to C (9 points)
Fill in the rest of the C code for the assembly code given below:

```c
sunny(int*, int):
    cmpl $1, %esi
    jne .L2
    movl (%rdi), %eax
    ret
.L2:
    cmpl $4, %esi
    jg .L4
    leal -1(%rsi,4), %esi
    addq $4, %rdi
    call sunny(int*, int)
    ret
.L4:
    testq %rdi, %rdi
    jne .L5
    leal 0(%rsi,8), %eax
    subl %esi, %eax
    ret
.L5:
    movl (%rdi), %eax
    shll $4, %eax
    ret

int sunny (int* n, int k) {
    if (______________) {
        ______________________
    } else if (______________) {
        ______________________
    } else if (______________) {
        ______________________
    } else {
        ______________________
    }
}
```