Exam Review!

15-213: Introduction to Computer Systems
6th Recitation, Oct. 16, 2011

Instructor: Adrian Trejo (atrejo)

Section H, 3:30p – 4:30p PH125C

Agenda

- Test tomorrow!
  - Cheat sheet: One 8.5 x 11, front and back
- Review
- Questions

[Subset of] What to Know

- Labs!
  - We try to reward people who did them well.

- Assembly
  - Basics (`cmp/test`, `mov/lea`, source vs. destination, operand order, etc.)
  - What registers are special? (Caller save vs. callee save, `esp/ebp` (x86), argument registers (x86-64), etc)
  - Switch statements and jump tables?
  - Loops?
  - You **should be able to** trace through assembly (like bomblab).
  - You **should be able to** write small amounts of assembly (like buflab).

[Subset of] What to Know

- Data Representation
  - Two’s complement
  - Floating point
  - Big Endian vs. Little Endian

- Stack
  - What’s different in x86 vs. x86-64?
  - You **should know** what goes where (e.g. parameters, old %ebp, return address, etc.).

- Structs (ignore unions)
  - Padding and alignment

[Subset of] What to Know

- Control
  - Loops in assembly? (for, while, do/while)
  - Recursion? (stack!)
- Memory
  - Heap vs. Stack
  - What is the L1 Cache?
- Arrays
  - Multi-dimensional access
  - Row-major order

Floating Point Review

- Basics
  - Sign, Exponent, Mantissa
    - $(-1)^s \times M \times 2^E$
  - Bias ($2^{k-1} - 1$)
  - Denormalized (exp = 000…000, M = 0.FFF…FFF, E = 1 – bias)
    - Small values close to zero.
  - Special Values (exp = 111…111)
    - +/-inf, NaN
  - Normalized (M = 1.FFF…FFF, E = exp – bias)
    - Everything else

1. Format A
   - There are $k = 3$ exponent bits. The exponent bias is 3.
   - There are $n = 5$ fraction bits.
2. Format B
   - There are $k = 5$ exponent bits. The exponent bias is 15.
   - There are $n = 3$ fraction bits.

Fill in the blanks in the table below by converting the given values in each format to the closest possible value in the other format. Express values as whole numbers (e.g., 17) or as fractions (e.g., 17/64). If necessary, you should apply the round-to-even rounding rule.

Format A		Format B	
Bits	Value	Bits	Value
`011 00000`	1	`01111 000`	1
			15
	$\frac{53}{16}$		
		`10100 110`	
`000 00001`			

Round-to-Even examples

- Assume s = 1, e = 4, f = 3, bias = $2^{(4-1)} - 1 = 7$

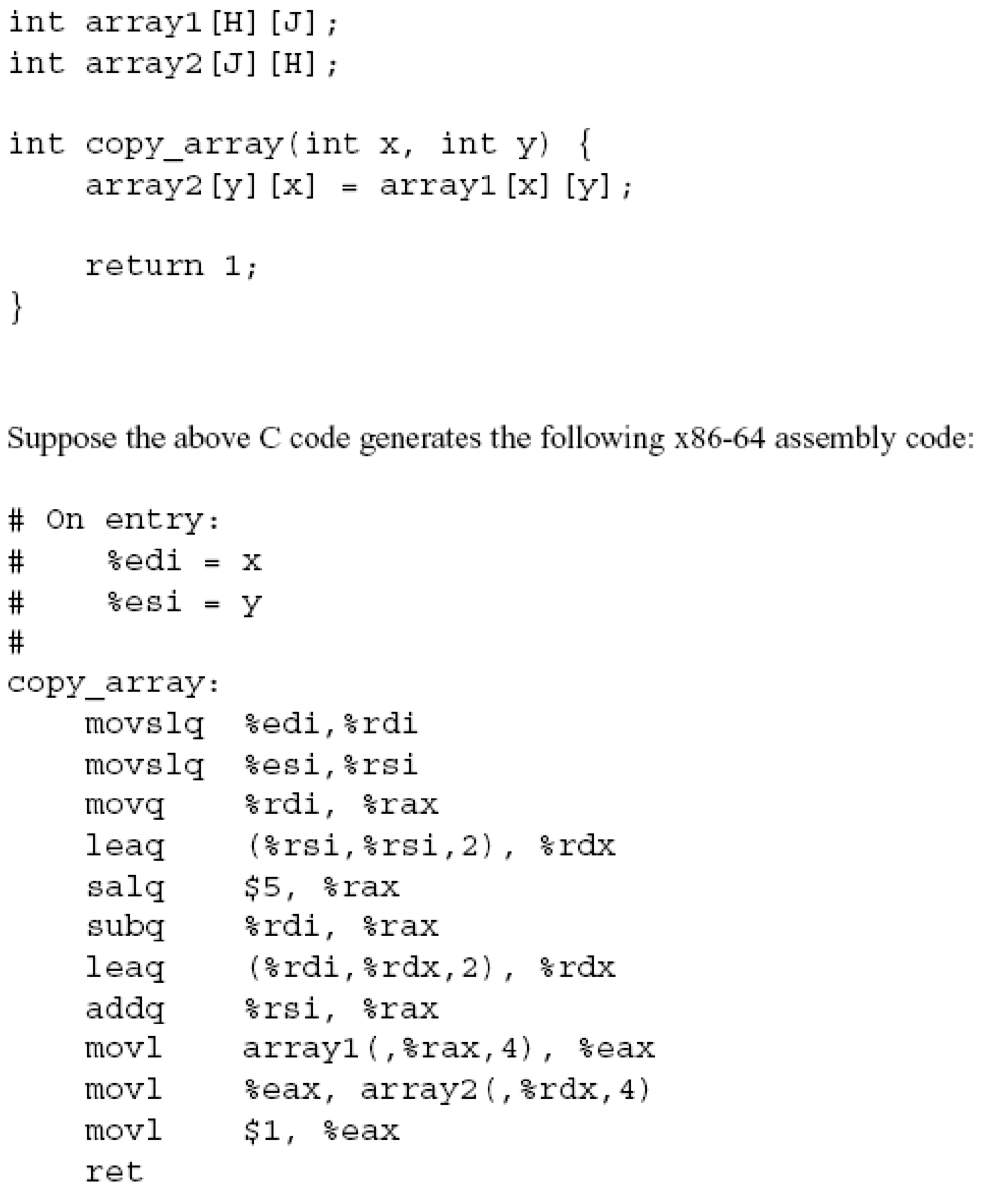
- $^{25}/_{64}$ -> $0.0110\ 010_2$
  - Round DOWN to $^{3}/_{8}$

- $^{27}/_{64}$ -> $0.0110\ 110_2$
  - Round UP to $^{7}/_{16}$

- $^{53}/_{128}$ -> $0.0110\ 101_2$
  - Round DOWN to $^{13}/_{32}$
  - Didn’t use round-to-even on this… it wasn’t a “tie”

Array Access

- Start with the C code
- Then look at the assembly
  - Work backwards!
- Easiest to just do an example



```
int array1[H][J];
int array2[J][H];

int copy_array(int x, int y) {
    array2[y][x] = array1[x][y];

    return 1;
}
```

Suppose the above C code generates the following x86-64 assembly code:

```
# On entry:
#    %edi = x
#    %esi = y
#
copy_array:
    movslq  %edi,%rdi
    movslq  %esi,%rsi
    movq    %rdi, %rax
    leaq    (%rsi,%rsi,2), %rdx
    salq    $5, %rax
    subq    %rdi, %rax
    leaq    (%rdi,%rdx,2), %rdx
    addq    %rsi, %rax
    movl    array1(,%rax,4), %eax
    movl    %eax, array2(,%rdx,4)
    movl    $1, %eax
    ret
```

Structs

- How big are things?
  - If you can’t remember, write it on your cheat sheet.
  - char, short, int, long, long long
  - float, double, long double
  - pointer
- Alignment rules (windows vs. linux)

```
struct {
    char *a;
    short b;
    double c;
    char d;
    float e;
    char f;
    long g;
    void *h;
} foo;
```

A. Show how the struct above would appear on a 32-bit Windows machine (primitives of size k are k-byte aligned). Label the bytes that belong to the various fields with their names and clearly mark the end of the struct. Use hatch marks to indicate bytes that are allocated in the struct but are not used.

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Questions/Advice

- Relax!
- Work past exams.
- Email us (15-213-staff@cs.cmu.edu)