- CPSC 275: Introduction to Computer Systems

CPSC 275: Introduction to Computer Systems

Fall 2025

- Syllabus
- Schedule
- Resources
- Upload
- Solution

Homework 15

NOTE: You are not required to hand in the following exercises, but you are strongly encouraged to complete them to strengthen your understanding of the concepts covered in class.

1. Suppose register `%eax` holds value `x` and `%ecx` holds value `y`. Fill in the table below with formulas indicating the value that will be stored in register `%edx` for each of the given assembly code instructions:

Instruction	Result
leal 6(%eax), %edx	__________________
leal (%eax,%ecx), %edx	__________________
leal (%eax,%ecx,4), %edx	__________________
leal 7(%eax,%eax,8), %edx	__________________
leal 0xA(,%ecx,4), %edx	__________________
leal 9(%eax,%ecx,2), %edx	__________________

2. Assume the following values are stored at the indicated memory addresses and registers:

Address	Value	Register	Value
0x100	0xFF	%eax	0x100
0x104	0xAB	%ecx	0x1
0x108	0x13	%edx	0x3
0x10C	0x11		

Fill in the following table showing the effects of the following instructions, both in terms of the register or memory location that will be updated and the resulting value:

Instruction	Destination	Value
addl %ecx,(%eax)	___________	___________
subl %edx,4(%eax)	___________	___________
imull $16,(%eax,%edx,4)	___________	___________
incl 8(%eax)	___________	___________
decl %ecx	___________	___________
subl %edx,%eax	___________	___________

3. Suppose we want to generate assembly code for the following C function:

```
int shift_left2_rightn(int x, int n) {
   x <<= 2;
   x >>= n;
   return x;
}
```

The code that follows is a portion of the assembly code that performs the actual shifts and leaves the final value in register `%eax`. Two key instructions have been omitted. Parameters `x` and `n` are stored at memory locations with offsets 8 and 12, respectively, relative to the address in register `%ebp`.

```
movl 8(%ebp), %eax    # Get x
__________________    # x <<= 2
movl 12(%ebp), %ecx   # Get n
__________________    # x >>= n
```

Fill in the missing instructions, following the annotations on the right. The right shift should be performed arithmetically.

4. In the following function, the expressions have been replaced by blanks:

```
int arith(int x, int y, int z) {
   int t1 = ________________;
   int t2 = ________________;
   int t3 = ________________;
   int t4 = ________________;
   return t4;
}
```

The portion of the generated assembly code implementing these expressions is as follows:

```
# x at %ebp+8, y at %ebp+12, z at %ebp+16
movl 12(%ebp), %eax
xorl 8(%ebp), %eax
sarl $3, %eax
notl %eax
subl 16(%ebp), %eax
```

Based on this assembly code, fill in the missing portions of the C code.

5. It is common to find assembly code lines of the form

```
xorl %edx,%edx
```

in code that was generated from C where no Exclusive-Or operations were present.

   A. Explain the effect of this particular Exclusive-Or instruction and what useful operation it implements.
   B. What would be the more straightforward way to express this operation in assembly code?
   C. Compare the number of bytes to encode these two different implementations of the same operation.

- **Welcome: Sean**
    - LogOut

Trinity College
HARTFORD CONNECTICUT