- CPSC 275: Introduction to Computer Systems

CPSC 275: Introduction to Computer Systems

Fall 2025

- Syllabus
- Schedule
- Resources
- Upload
- Solution

Homework 14

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. Each of the following lines of code generates an error message when we invoke the assembler. Explain what is wrong with each line.

```
movb $0xF,(%bl)
movl %ax,(%esp)
movw (%eax),4(%esp)
movb %ah,%sh
movl %eax,$0x123
movl %eax,%dx
movb %si,8(%ebp)
```

   where the move instructions, `movb`, `movw`, and `movl`, operate on data of size 1, 2, and 4 bytes, respectively.

2. For each of the following lines of assembly language, determine the appropriate instruction suffix based on the operands. (For example, `mov` can be rewritten as `movb`, `movw`, or `movl`.)

```
 mov  %eax, (%esp)
 mov  (%eax), %dx
 mov  $0xFF, %bl
 mov  (%esp,%edx,4), %dh
 push $0xFF
 mov  %dx, (%eax)
 pop  %edi
```

3. 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 values for the indicated operands:

Operand	Value
%eax	_____
0x104	_____

```
$0x108                  _____
(%eax)                  _____
4(%eax)                 _____
9(%eax,%edx)            _____
260(%ecx,%edx)          _____
0xFC(,%ecx,4)           _____
(%eax,%edx,4)           _____
```

4. You are given the following information. A function with prototype

   ```
   void decode1(int *xp, int *yp, int *zp);
   ```

   is compiled into assembly code. The body of the code is as follows:

   ```
   xp at %ebp+8, yp at %ebp+12, zp at %ebp+16
   1 movl 8(%ebp), %edi
   2 movl 12(%ebp), %edx
   3 movl 16(%ebp), %ecx
   4 movl (%edx), %ebx
   5 movl (%ecx), %esi
   6 movl (%edi), %eax
   7 movl %eax, (%edx)
   8 movl %ebx, (%ecx)
   9 movl %esi, (%edi)
   ```

   Parameters `xp`, `yp`, and `zp` are stored at memory locations with offsets 8, 12, and 16, respectively, relative to the address in register `%ebp`. Write C code for `decode1` that will have an effect equivalent to the assembly code above.

- **Welcome: Sean**
  - LogOut