January 18

- Hannah is sweet 16!
- You should have a book by now.
- A book is on reserve in the library.
- You should get the email I sent this AM.
- TA Office hours now in the POOP.
- Don’t email answers to ME
- Assignment 2

C versus ASM

```
Swap(int v[], int k) {
  int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

```
Swap:
        muli $t0, $a1, 4
        add $t0, $a0, $t0
        lw $t1, 0($t0)
        lw $t2, 4($t0)
        sw $t2, 0($t0)
        sw $t1, 4($t0)
        jr $ra
```

Form of the Instructions

- Opcode
- Register (usually result destination)
- Operand 1
- Operand 2

  e.g.

  ```
  add $t0, $a0, $t0
  ```

Naming Registers

This is all just software "convention"

- $a0 - $a3 arguments to functions
- $v0 - $v1 results from functions
- $ra return address
- $s0 - $s7 "saved" registers
- $t0 - $t9 "temporary" registers
- $sp stack pointer

What are the operands?

- Registers e.g. $a0
- With load and store this is logically enough
- But small constants are VERY common
- So, some instructions allow “immediate” operands. E.g. muli $t0, $a1, 4
- ***Where do we get big constants?***

C versus ASM

```
Swap(int v[], int k) {
  int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

```
Swap:
        muli $t0, $a1, 4
        add $t0, $a0, $t0
        lw $t1, 0($t0)
        lw $t2, 4($t0)
        sw $t2, 0($t0)
        sw $t1, 4($t0)
        jr $ra
```

Next: Performance

- Measure, Report, and Summarize
- Make intelligent choices
- See through the marketing hype
- Key to understanding underlying organizational motivation

Why is some hardware better than others for different programs?

What factors of system performance are hardware related?
(e.g., Do we need a new machine, or a new operating system?)

How does the machine's instruction set affect performance?