CS315-02 RISC-V Assembly 2
Functions, Arrays, Control

Project01 explaining
code quality

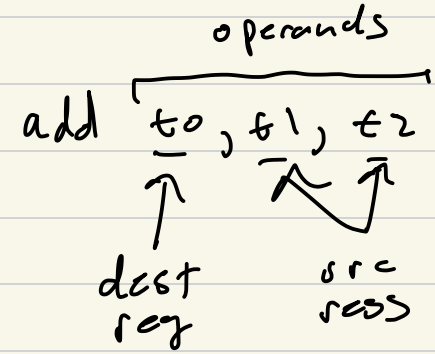
Lab02 today due Tue 10th
Exam problems due Wed 11th
problems.pdf
Project02

RISC-V Assembly

review concepts
more instructions
array access
if/then/else
loops
gdb

Instructions and Registers (32)

64 bits wide




```
addi  a0, a0, 1
```
↑ immediate

registers : x0, x1, ... , x31

arguments a0, a1, a2, a3, ...

return value a0

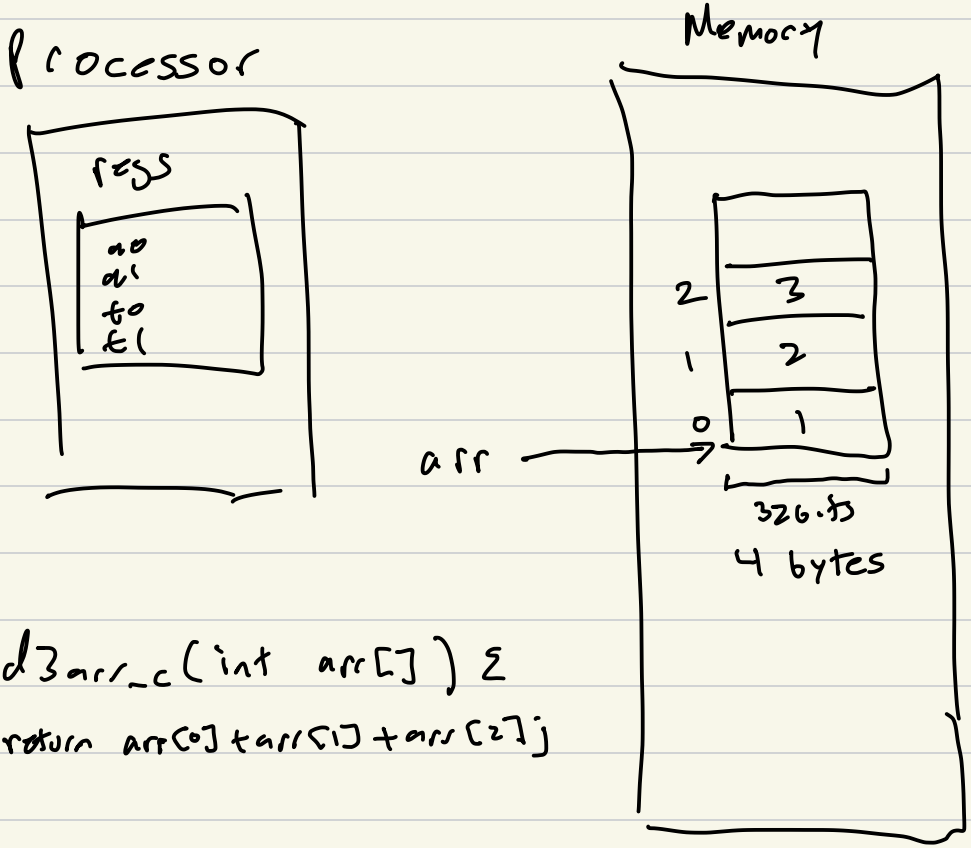
zero (x0) is always zero

```
add t1, zero, t2
add zero, t1, t2
```

3 categories of instructions ① data processing ② control ③ memory

Arrays

`int arr[3] = {1, 2, 3};`




```
add3arr_c(int arr[]) {
  return arr[0]+arr[1]+arr[2];
}
```

Memory instructions

load / store

```
lw t0, (a0)
```

`#` t0 = *a0

- lw: load word (32 bit value)
- t0: dst reg
- (a0): address

Control Statements

if / then / else

C

```
int val;
int r;

if (val > 0) {
    r = 1;
} else {
    r = 0;
}
```

ASM

```
# t0 - int r

    ble a0, zero, else
    li t0, 1
    j done
else:
    li t0, 0
done:
```

ble is a conditional branch

j is an unconditional jump

loops

C

```
loopsum (int n)

int i;
int sum = 0;

for (i=0; i<n; i++) {
    sum = sum+1;
}
```

Asm

```
# t0 - int i
# t1 - int sum

loopsum_s:
    li t0, 0
    li t1, 0

loop:
    bge t0, a0, done
    add t1, t1, t0
    addi t0, t0, 1
    j loop
done:
    mv a0, t1
    ret
```