

Lab 2b

CSE141L

5/9

Announcements

- Lab 2b Due
 - 5/14
- 15 Min Lab Interviews
 - Sign up for your interview slot
 - 5/14
 - 3:30 pm – 6:00pm (total 10 slots)

Sample Code Stub

- How to handle
 - fib (10)

```
.text
  li $gp, DATA_ADDR    // set $gp register,
                        // DATA_ADDR is determined by what you pass
                        // to assembler for -data_addr

  li $sp, STACK_ADDR    // set $sp register
                        // determined by the physical memory size

  mov $a0, 10           // argument 0 = 10
  jal fib               // call function 'fib'
end:
  j end                 // while(1)
```

Sample Code Stub - SuperGarbage

```
#define SUB 0x0
#define RSFT 0x1
#define NOR 0x2
#define SWAP 0x3
#define IN 0x4
#define OUT 0x5
#define BLZ 0x6
#define HALT 0x7

struct inst test[] = {
    // supergarbage memory
    {0, 0, 0, 0},
    {0, 0, 0, 0},
    {0, 0, 0, 0},
    {0, 0, 0, 0},
    // supergarbage instruction start
    {SUB, 0, 0, 0}, // $0 <= 0
    {NOR, 0, 0, 1}, // $1 <= -1
    {SUB, 0, 1, 2}, .. // $2 <= 1
};
SuperGarbage(16, test);
```

Sample Code Stub - SuperGarbage

```
.text
li    $gp, DATA_ADDR           // set gp register
li    $sp, STACK_ADDR          // set sp register
la    $a0, SuperGarbage_pc
la    $a1, SuperGarbage_addr_space // set the second arg
sub   $a0,$a0,$a1              // set the first arg
jal   SuperGarbage             // call function `SuperGarbage`
end:
j    end // while(1)

.data
SuperGarbage_addr_space:
    .fill 16 0x0                // fill MEM[0] - MEM[15] with 0x0
SuperGarbage_pc:
    .word 0x0, 0x0, 0x0, 0x0    // {SUB, 0, 0, 0}
    .word 0x2, 0x0, 0x0, 0x1    // {NOR, 0, 0, 1}
    .....
```

.fill *n value* duplicates the same *value* *n* times.

.word directive can be followed by one or more words, separated by commas after the directive to describe data for multiple words.

Int34 Class

- 34 bit Arithmetic Operations
 - left shift (arithmetic)
 - right shift (arithmetic)
 - not
 - add
 - subtract
 - and
 - or
 - xor

Sample Code

```
public static void main(String[] args) {
    Int34 a = new Int34(1, 0x1111);
    Int34 b = new Int34(-1, 0xffffffff);
    Int34 c = new Int34(Long.MAX_VALUE);

    System.out.println("a = " + a.longValue());
    System.out.println("b = " + b.longValue());
    System.out.println("c = " + c.longValue());

    System.out.println("a: " + a + " " + a.shiftLeft(3) + " " + a.shiftRight(3) + " " + a.not());
    System.out.println("b: " + b + " " + b.shiftLeft(3) + " " + b.shiftRight(3) + " " + b.not());
    System.out.println("c: " + c + " " + c.shiftLeft(3) + " " + c.shiftRight(3) + " " + c.not());

    System.out.println("a + b = " + a.add(b));
    System.out.println("a - b = " + a.subtract(b));
    System.out.println("a | b = " + a.or(b));
    System.out.println("a xor b = " + a.xor(b));
}
```

Any Questions?