

# CS 241 – Week 4 Tutorial

## Writing an Assembler Pt. I

Fall 2018

### 1 Symbol tables in MIPS

Construct the symbol table for the following MIPS assembly program.

```
begin:
label: beq $0, $0, after
jr $4

after:
sw $31, 16($0)
lis $4
abc0: abc1: .word after

loadStore:
lw $20, 4($0)
sw $20, 28($0)

end:
```

### 2 Error-checking MIPS programs

Identify the errors in the following assembly language program.

```
label: label: .word label
.word ; 0
.word aaaaa
.word 1 2 3
.word 2147483648 abcde:
.word ,
```

### 3 Bitwise operations in MIPS

Bitwise operations enable precise manipulation of bit patterns. The bitwise operations that will be useful in this course are the following

- Inclusive Or (`|`)
- And (`&`)
- Left shift (`<<`)
- Right shift (`>>`)

Assume 4-bit integers. What should each of the following computations produce, assuming we are using logical right shifting?

1. `3 & 5`
2. `3 | 5`
3. `3 << 2`
4. `3 >> 2`
5. `13 << 2`
6. `13 >> 2`

Briefly summarize how each operation works and its utility with respect to writing an assembler.

### 4 Binary output

Write pseudocode for a function called `output_word` that takes a 32-bit integer as input and outputs each of its four bytes to standard output.

You can assume a function called `output_byte` is available. This function takes an integer as input and first checks if the integer is small enough that it can be represented in 8 bits. If so, it outputs the corresponding byte; otherwise it produces an error.

How would you use this above function (in conjunction with the symbol table) to assemble the `.word foo` where `foo` is a label in an assembly program?