Lecture 4

Assembly language and Procedures

Calling on in transit

CS 241: Foundations of Sequential Programs
Winter 2018

Troy Vasiga et al
University of Waterloo

Review

- Arrays
- Conditionals (if-statements)
- Loops

Loop example

```
;$2 <- 13
lis $2
.word 13

; clear $3
add $3, $0, $0

; add current value to total
add $3, $3, $2

; decrement $2
lis $1
.word -1
add $2, $2, $1

; if $2 != 0, loop
bne $2, $0, -5

; return to the OS
jr $31
```

Labels

Storing and Restoring Registers

- We wish to keep register values intact

Procedures (Example 6a and 6b)

Rules for stack usage

If I push, I pop

Recursion in MIPS (Example 7a and 7b)

A trace of the flow of control

Template for MIPS recursion

1. save registers (in particular, $31)
2. check base case(s)
3. recursive case
   - compute next values
   - call myself
   - compute my return value
4. restore registers

Binary trees in (linear) RAM

Other MIPS notes

- unsigned operators (e.g., `multu`, `sltu`, ...)
- local variables in MIPS?
- dynamic memory?