

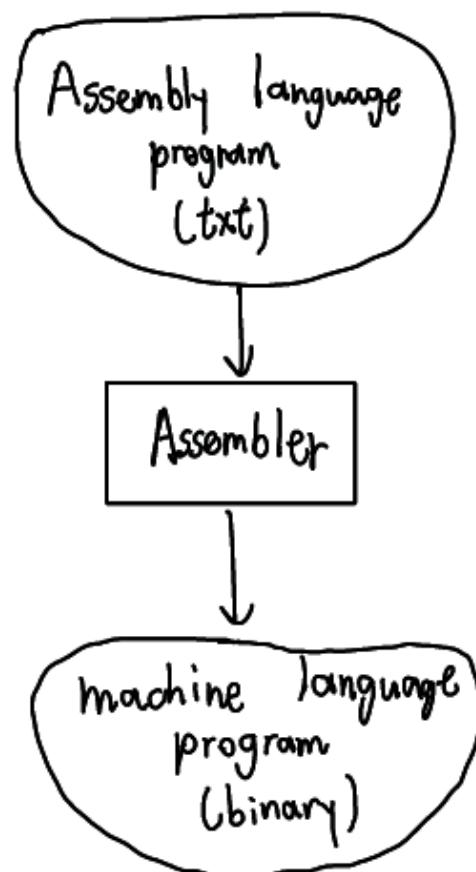
Lecture 5+6

Assemblers

CS 241: Foundations of Sequential Programs
Fall 2014

Troy Vasiga et al
University of Waterloo

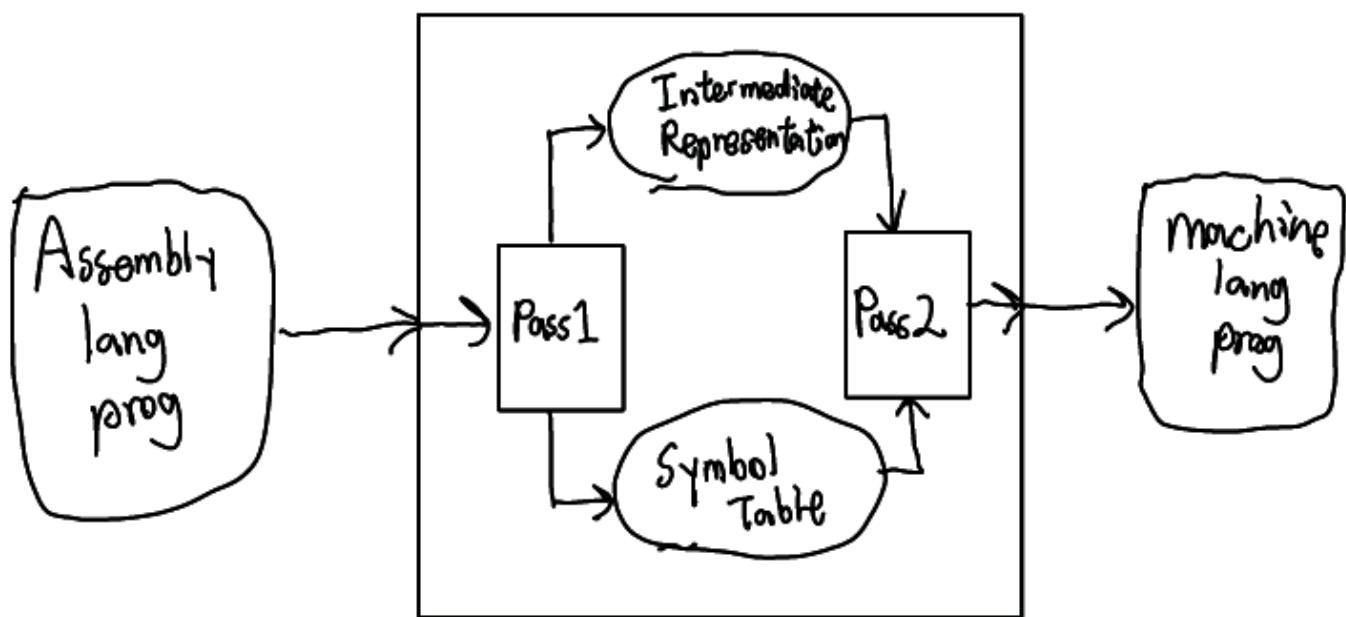
Overview



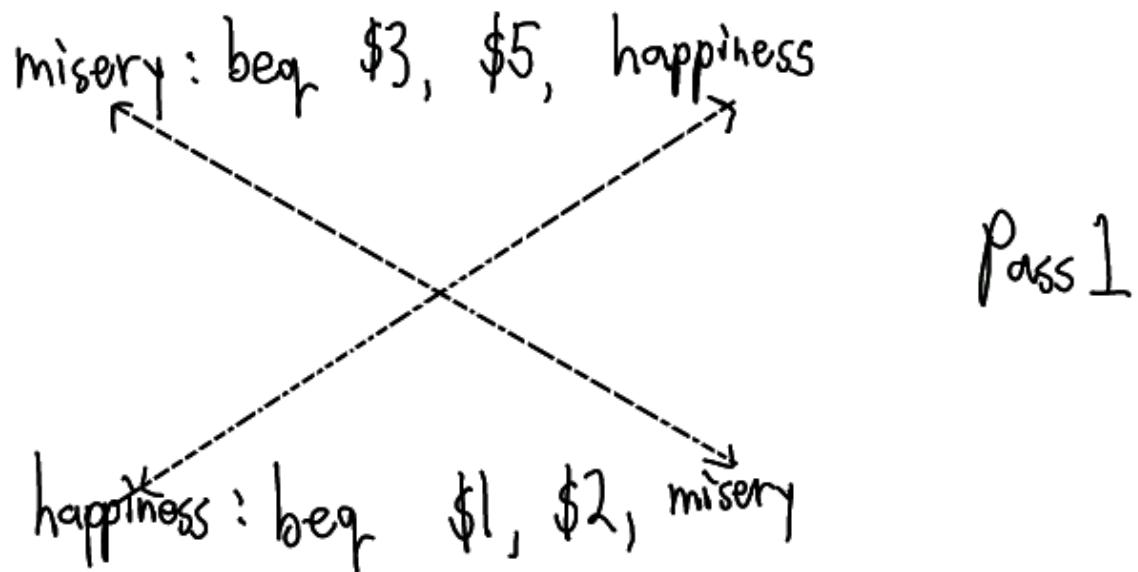
- ▶ Input to an assembler
 - ↳ sequence of characters
- ▶ Output from an assembler
 - ↳ sequence of bits

Stages to an assembler

A look inside the picture



Why two passes?



Defining assembly language

- ▶ informal description: see website
 - english
- ▶ formal description: later

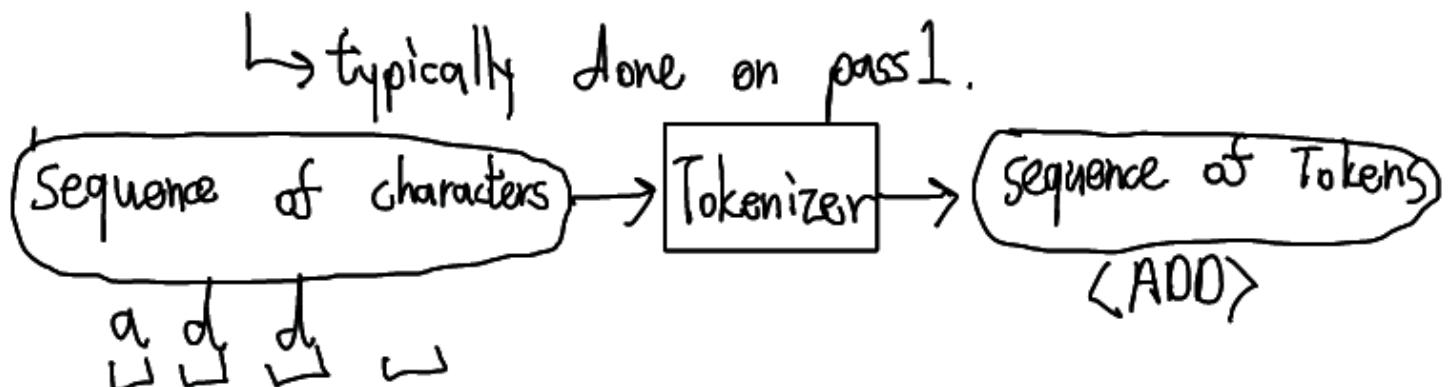
Basic assembly language syntax and semantics


meaning

- ▶ syntax \Rightarrow form/structure
- ▶ location and labels what does this label mean?
 where I am

Analysis (First pass)

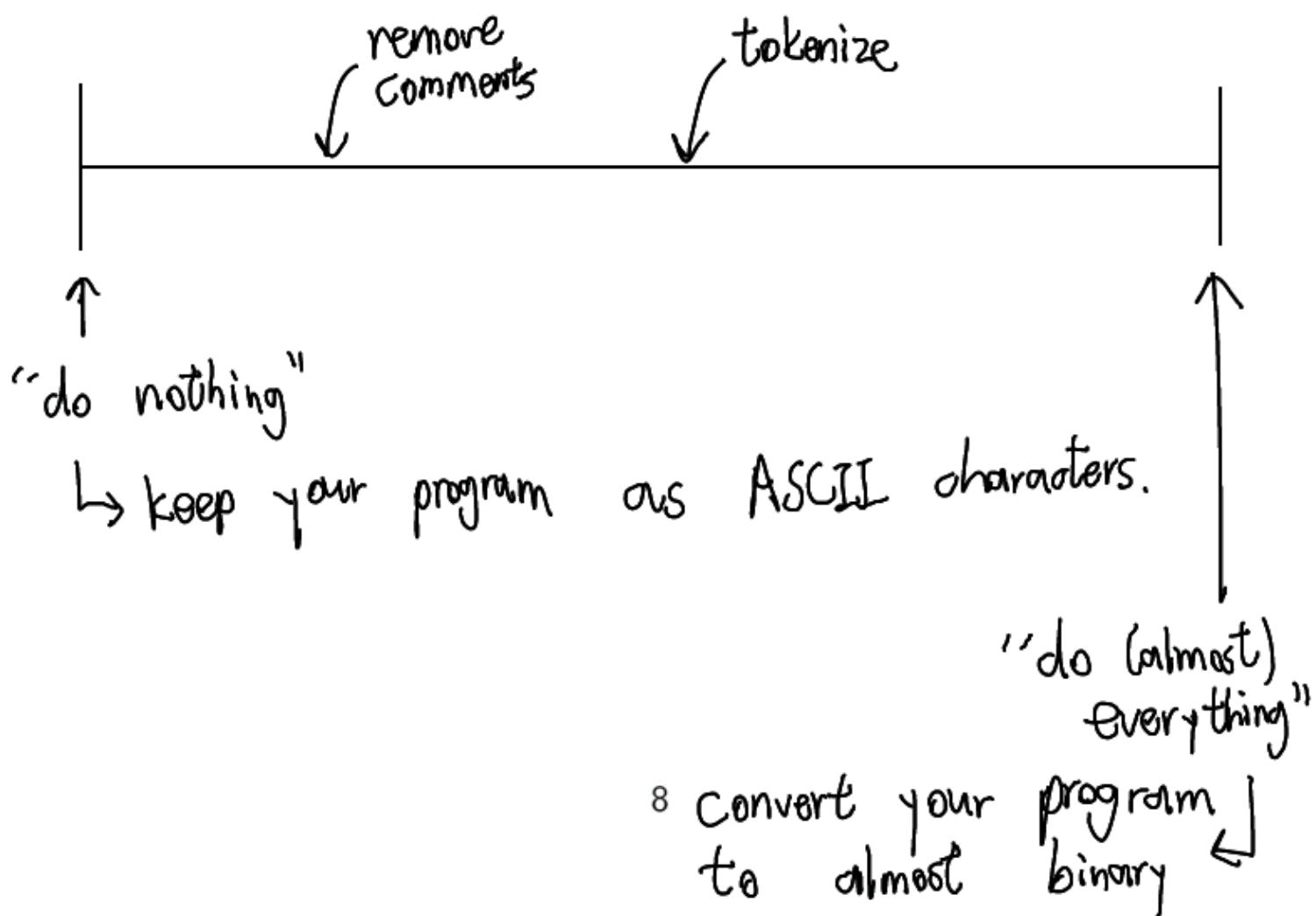
- ▶ must generate an intermediate representation
- ▶ must generate a symbol table
- ▶ tokenization



e.g. add \$3,\$4, \$5 → ADD REG COMMA
REG COMMA REG

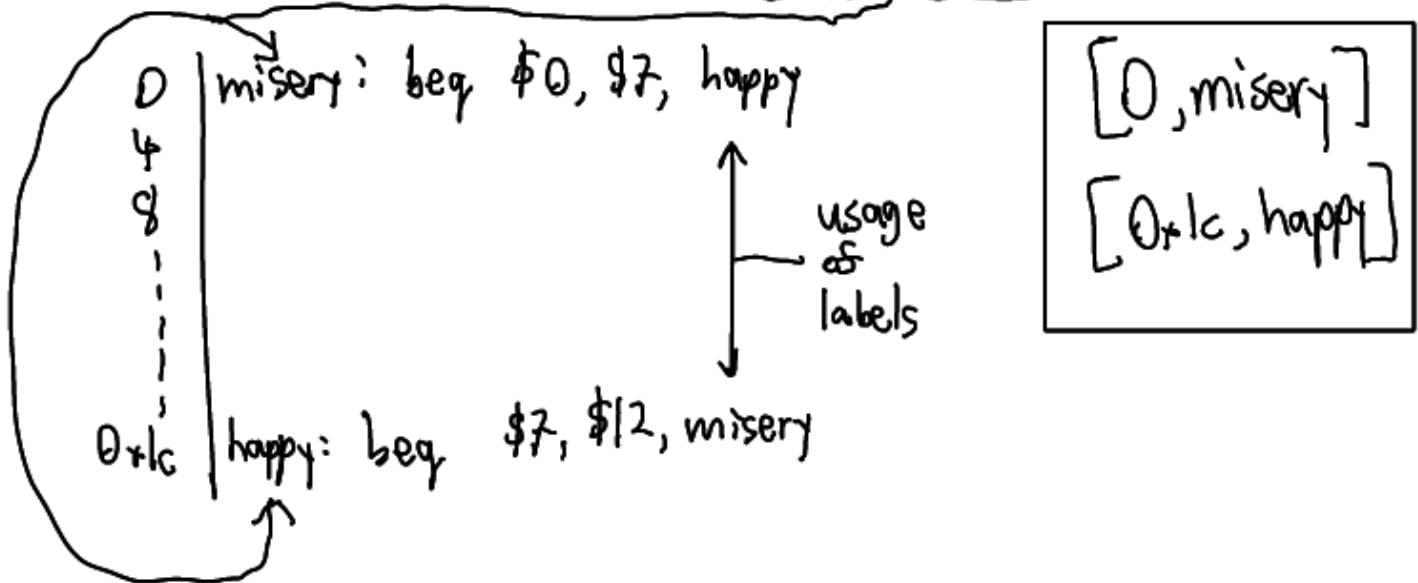
Intermediate Representation

A continuum

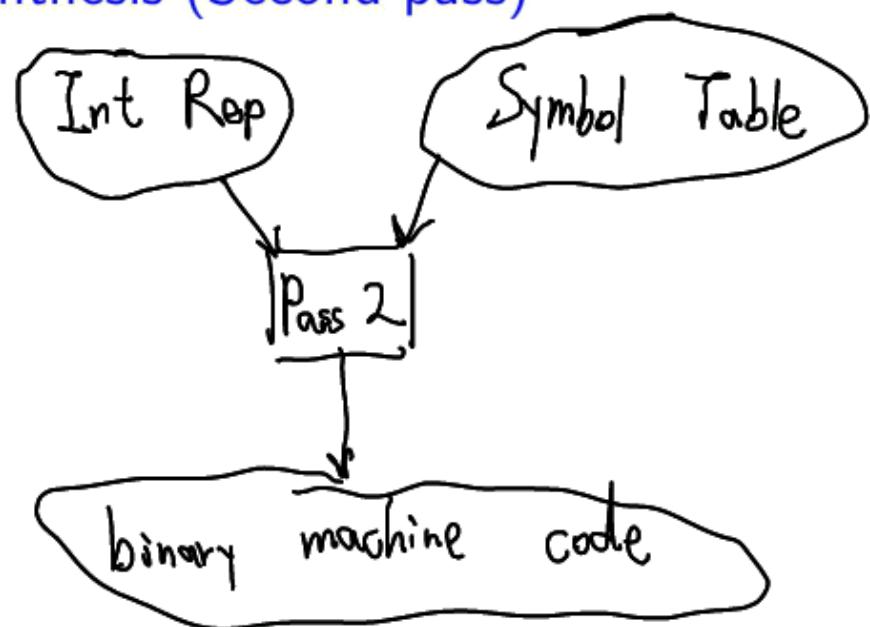


Symbol Table

- stores the address and name of each label definition



Synthesis (Second pass)



Encoding a beq instruction

0x1c happy: beq \$7, \$12, misery

$$\leftarrow \text{PC} = 0x20$$

Symbol Table:

0, misery
0x1c, happy

$$\begin{aligned} &\text{move back} \\ &-0x20 \text{ bytes} \\ &= (-32)_{10} \text{ bytes} \\ &\approx (-8)_{10} \text{ words} \end{aligned}$$

0001 0000 1110 1100 1111 1111 1111 1000
\$7 \$12 (dest. addr. - PC) / 4

11

0000 1000	8
$\Rightarrow 1111 0111$	\downarrow
$\Rightarrow 1111 1000$	-8

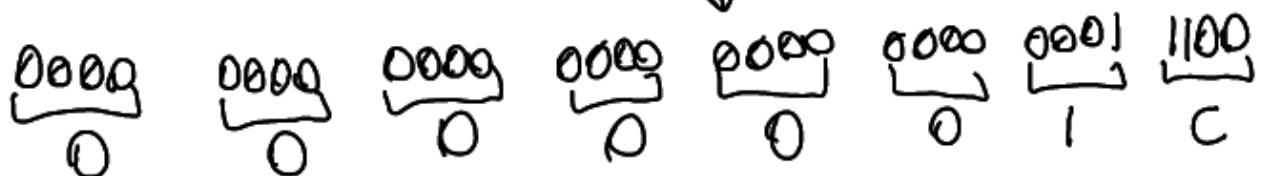
Symbol Table
(0, misery)
(0x1c, happy)

Encoding a .word <label> instruction

0x30 lis \$6 ✓

0x34 .word happy]

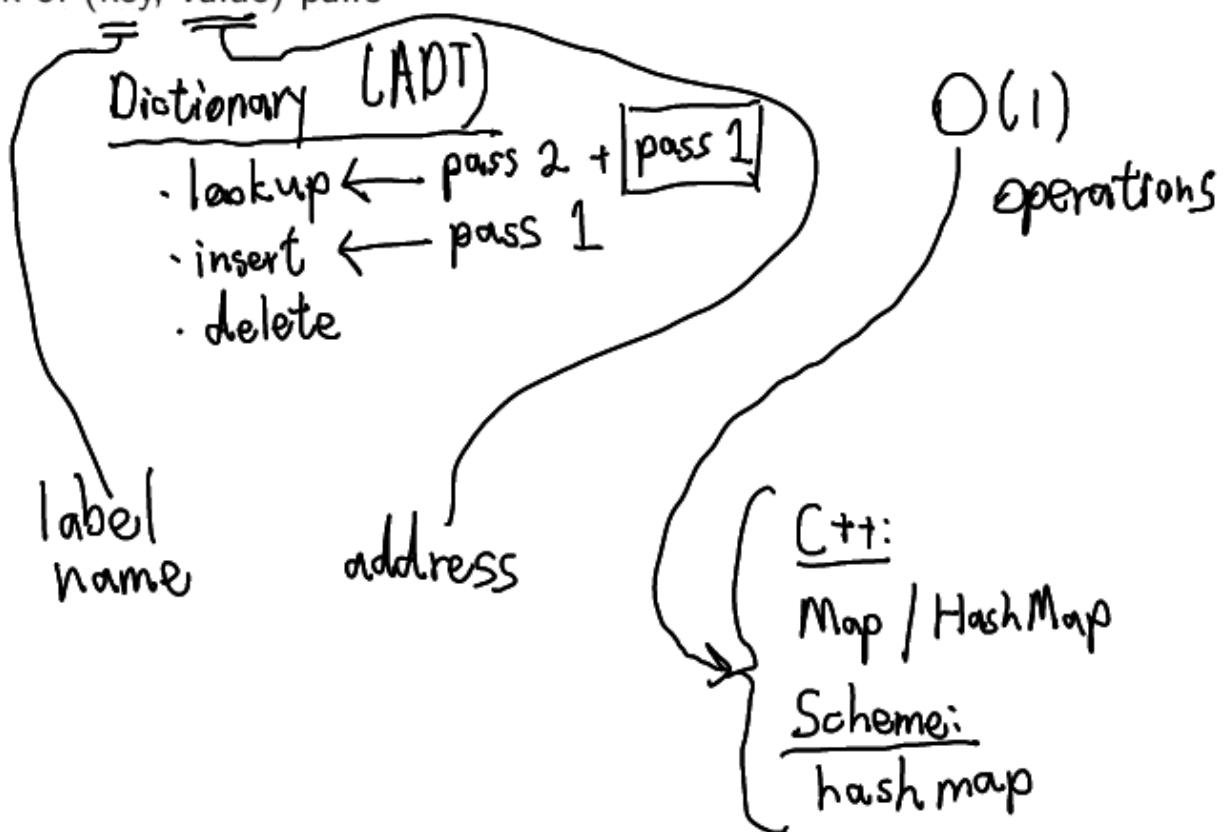
0x38 jalr \$6 ✓



Encoding lis or jalr or ... Assignment 1

Efficiency of the symbol table

Think of (key, value) pairs



Error checking: Pass 1

Syntax:

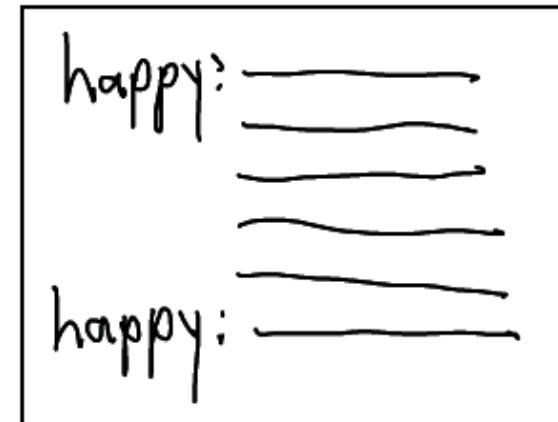
\$7:

7bob!!:

bob!!:

Semantic errors:

duplicate
label
errors



Error checking: Pass 2 (or earlier)

Syntax errors

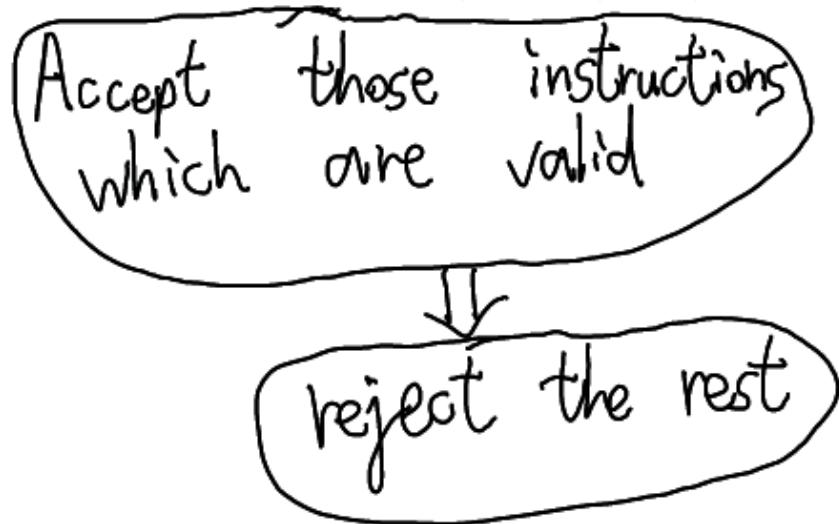
semantic error →

lw \$3 0(\$4)
lw \$3, 0(\$4)
lv \$3, 0(\$4) ← Ambig: semantic error.
lw lw \$3, 0(\$4)
lw \$3, 4444444444444444 (\$4)
lw \$3, 5(\$30) ← maybe??

CS241. binasm

Error checking: Philosophy

- ▶ this will be true for the rest of this course



A little bit about bits

You have to output bits, not ASCII 0 and 1.

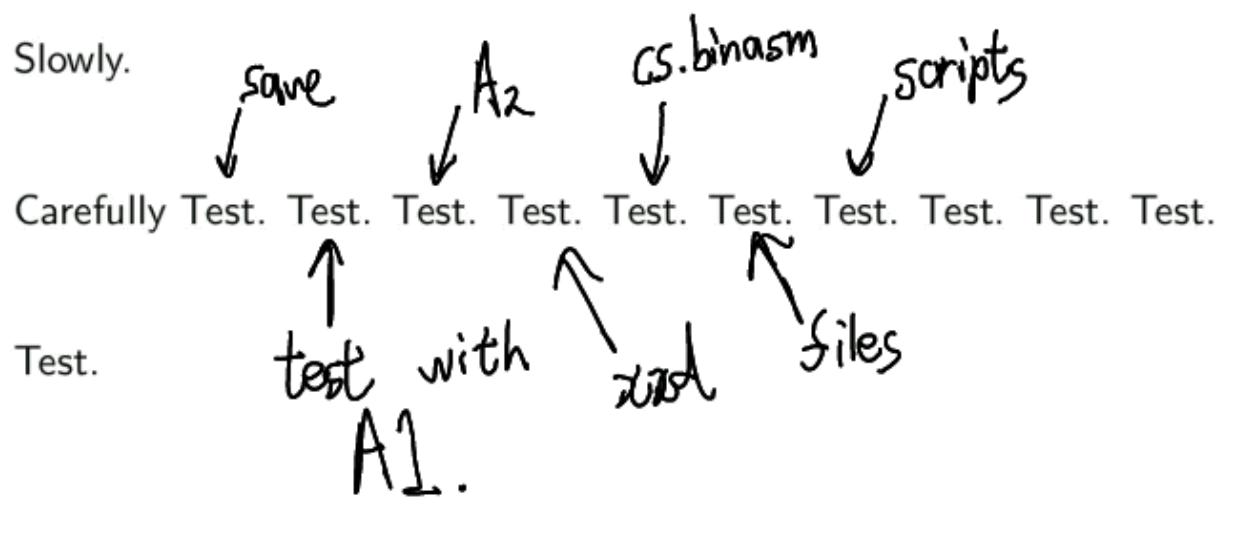


check BEQ example!

A gift

~~asm.h~~ or asm.ss or asm.c or asm.cc
tokenizer! ↴ A5/A6

How to write an assembler



./a.out < test1.asm > test1.mips
mips.twinkl test1.mips < ih > ~~~~