University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 120: Introduction to Computing

Assemblers

How Do Assemblers Work?

Just like we do (for this purpose)!

Step 1: Figure out the instruction sequence.
(Humans do this part.)

Step 2: Map instructions and data to memory addresses.
First pass of assembler:
find where labels fall in memory.

Step 3: Calculate and fill in relative offsets.
Second pass of assembler: write bits.

The Assembler Counts to Find Label Values (Addresses)

x3000
x3001
x3002
x3003

Start at x3000.

```
        .ORIG   x3000           ; starting address is x3000
        LEA R0,HIST             ; R0 to the start of the histogram
;
; fill the histogram with zeroes
        AND R6,R6,#0            ; put a zero into R6
        LD R1,NUM_BINS          ; initialize loop count to 27
        ADD R2,R0,#0            ; copy start of histogram into R2
;
; loop to fill histogram starts here
HFLOOP  STR R6,R2,#0            ; write a zero into histogram
        ADD R2,R2,#1            ; point to next histogram entry
        ADD R1,R1,#-1           ; decrement loop count
        BRp HFLOOP              ; continue until loop count reaches zero
;
;                               ; and R5 from memory
;                               ; R3 holds additive inverse of ASCII '@'
;                               ; R4 holds difference between ASCII '@' and 'Z'
        LD R5,AT_MIN_BQ         ; R5 holds difference between ASCII '@' and '`'
        LD R1,STR_START         ; point R1 to start of string
```

Let's count!

HFLOOP is x3004.

The Assembler Counts to Find Label Values (Addresses)

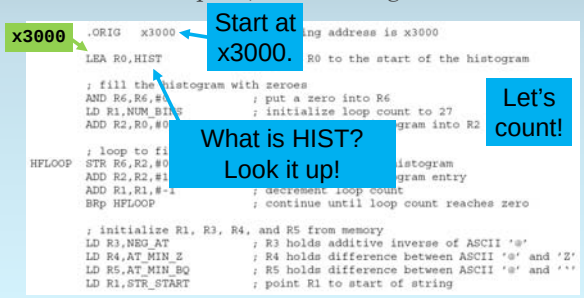
The first pass produces a **symbol table**.

Symbol	Address
HFLOOP	x3004
COUNTLOOP	x300C
NON_ALPHA	x3010
AT_LEAST_A	x3014
ALPHA	x3016
...	...
HIST	x3028
STRING	x3043

This table is generated for the assembly version of the letter frequency program.

The Assembler Uses the Symbol Table to Calculate Offsets

In the second pass, we count again!



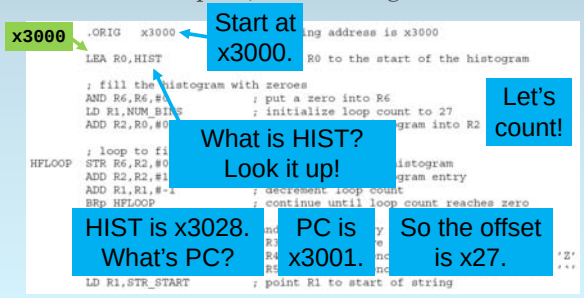
Find the Symbol to Calculate the LEA Offset

Symbol	Address
HFLOOP	x3004
COUNTLOOP	x300C
NON_ALPHA	x3010
AT_LEAST_A	x3014
ALPHA	x3016
...	...
HIST	x3028
STRING	x3043

Find "HIST" in the symbol table.

The Assembler Uses the Symbol Table to Calculate Offsets

In the second pass, we count again!



What Can Go Wrong?

What happens if the assembly file has a mistake?

~~The assembler fixes the bug.~~

Wishful thinking!

Computers are dumb.

The assembler says "ERROR!" …

…and waits for the human (you!) to fix the bug.

What Can Happen in the First Pass?

What kinds of errors can the assembler detect in the first pass?

Things like **bad mnemonics**…

```
MULT R1,R2,R3
```

and **bad operands**…

```
ADD R42,R0,#-3
ADD R1,R2,#1000
```


What Can Happen in the First Pass?

What about labels?

Is this code ok?

```
        BRp NO_LABEL_YET
        ADD R1,R2,R3
NO_LABEL_YET ; here it is!
```

The assembler **reads the file in order**.

A symbol in the file may not be found in the symbol table in the first pass, but that's ok.

What Can Happen in the First Pass?

What about this code…

```
        BRp NO_LABEL_YET
        ADD R1,R2,R3
NO_LABEL_YET  ; here it is!
        ADD R2,R1,R5
NO_LABEL_YET ; here, too!
```

The symbol **cannot be added twice**!

If a symbol is already in the table, the **label is multiply-defined** (first pass).

What Can Happen in the Second Pass?

What kinds of errors can the assembler find in the second pass?

We saw one already…

Find the Symbol to Calculate the An Offset

Symbol	Address
HFLOOP	x3004
COUNTLOOP	x300C
NON_ALPHA	x3010
AT_LEAST_A	x3014
ALPHA	x3016
...	...
HISTOGRAM	x3028
STRING	x3043

Find "HIST" in the symbol table.

Oops!

What Can Happen in the Second Pass?

What kinds of errors can the assembler find in the second pass?

We saw one already…

Label not defined

What else might go wrong in the second pass?

What's Wrong with This Code?

```
            LEA R0,MY_SPACE
            BRnzp STOP
            MY_SPACE .BLKW x4200
STOP        HALT
```

Data interleaved with code!

Bad style, but not an error.

What's the offset for **BRnzp**? More than 9 bits…

(Error: **Address/offset out of range**.)

Errors Found by the LC-3 Assembler

Found in the first pass
- bad opcode mnemonic
- bad operand (of any kind, such as the wrong number, wrong type, or out of range)
- multiply-defined label

Found in the second pass
- undefined label
- target address too far away

