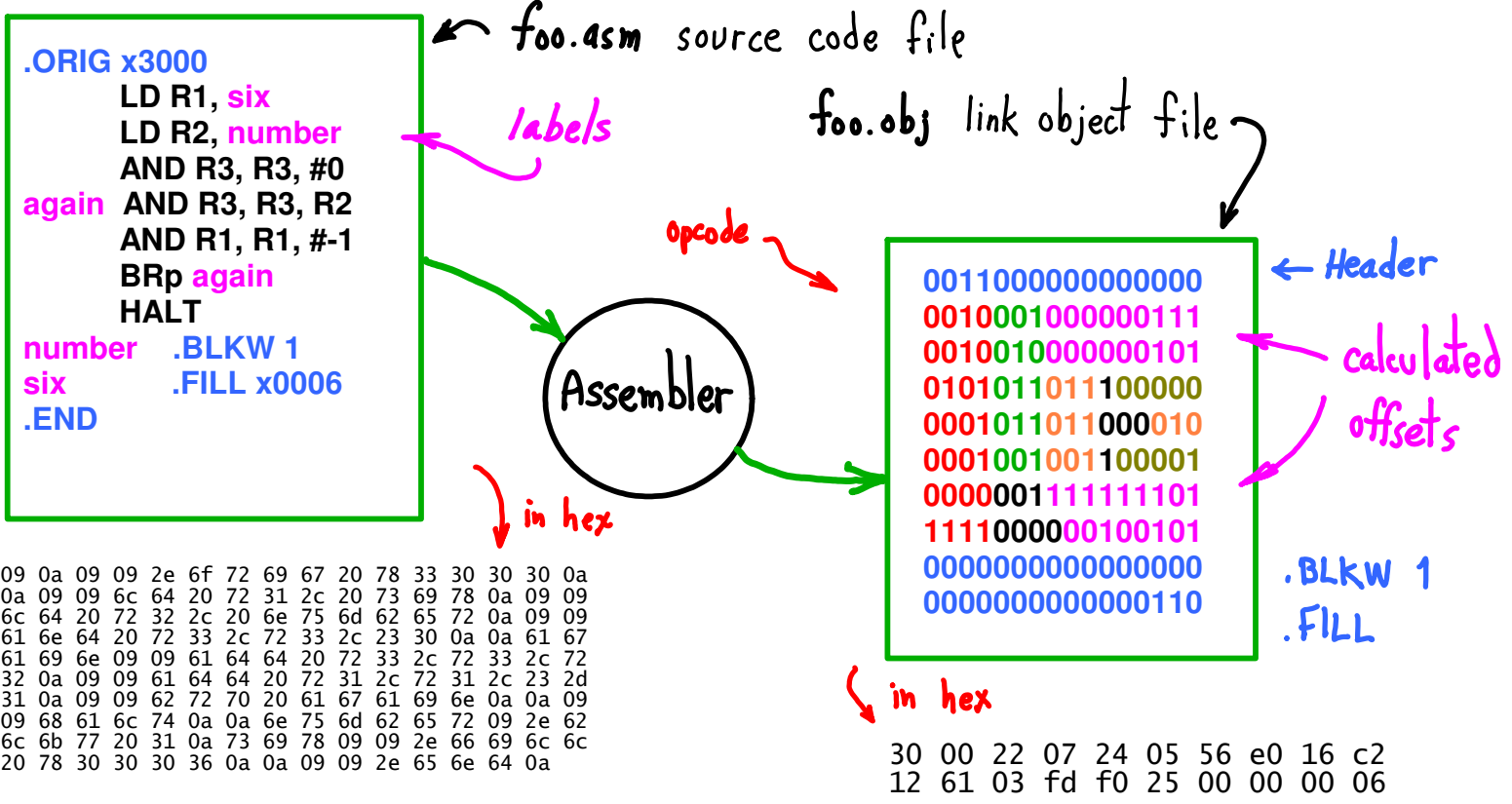


Assembly Language

P&P, Figure 7.1



Assembler Directives:

- .ORIG:** put load address into header.
- .END:** end of source code.
- .FILL:** put bits into a word.
- .BLKW:** create *n* blank words (all zeroes).

Assembler calculates **offsets** from **LABELS**.

LABEL == **word offset** into file.

Offsets are calculated by **subtraction**:

$$\text{LABEL} - (\text{current word count}) == \text{offset}$$

Assembly produces machine **words**:

- **ONE** for an instruction
- **ONE** for **.FILL**
- **n** for **.BLKW n**

--- **foo.obj** contains **BITS**, **not ASCII** codes for "1" and "0".
 --- "see" the bits:

```
>> od -t x1 foo.obj
```

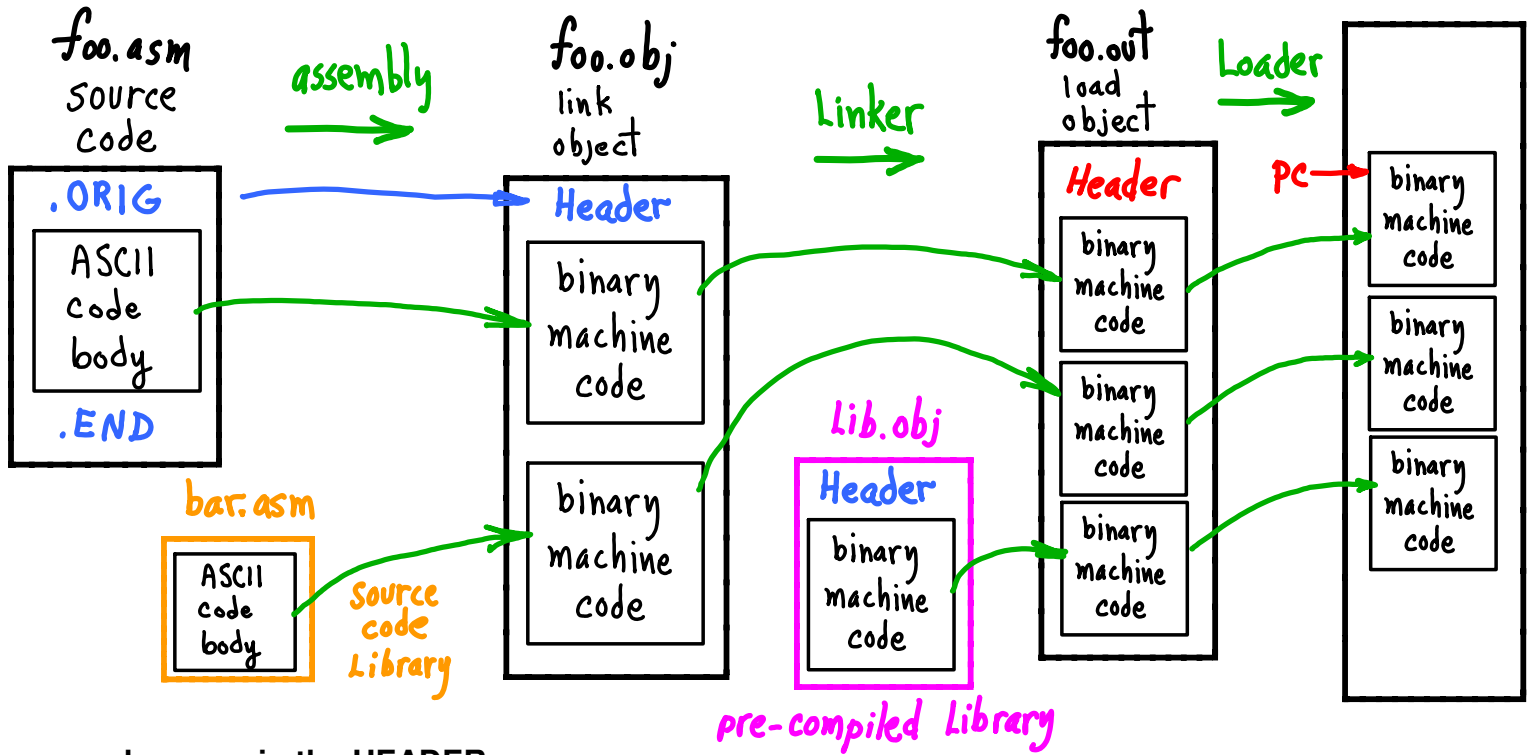
```

0000000 30 00 22 07 24 05 56 e0 16 c2 12 7f 03 fd f0 25
0000020 00 00 00 06
0000024
  
```

Format: **offset(hex)** 1-byte integers (hex), first to last.

Notice the **big-endian** order (x3000, 30 is at lower offset than 00). An artifact of the simulator's code. It does not reflect the LC3 micro-architecture because LC3's memory is **not byte-addressable**.

Assemble, link, load



There may be more in the HEADER.

Symbol Table:

LABEL offset

Labels can be names for external reference

- ====> external code
- ====> external data word

Linker:

- Combine modules
- Fix offsets (references)

Loader:

- Fix offsets
- Copy to memory
- Dynamically link modules

File Formats:

Standards define layout of link/load objects.

LC3 Assembly Language

See P&P APPEND. A

Source Code:

```
AND R3, R3, #0 ; This line of code needs explanation.
```

Operation name:
maps 1-1 to opcode

immediate value

- # --- decimal notation
- x --- hex notation

Comments:
everything after ";" is ignored.

NB--there are some special translations, eg., "Halt".

2-pass Assembly

PASS 1: Build Symbol Table.

Find all LABELS and their offsets

```

.ORIG x3000
ADD R1, R2, R3
BRnzp here
.FILL x5678
here:
ADD R1, R2, R3
BRz here
    
```

COUNT
3000

Symbol Table

STRING	offset
"here"	3003

count +=
1 (an instruction word),
1 (a data word),
0 (anything else).

PASS 2: Generate code

COUNT
3000

```

ADD R1, R2, R3
BRnzp here
.FILL x5678
here:
ADD R1, R2, R3
BRz here
    
```

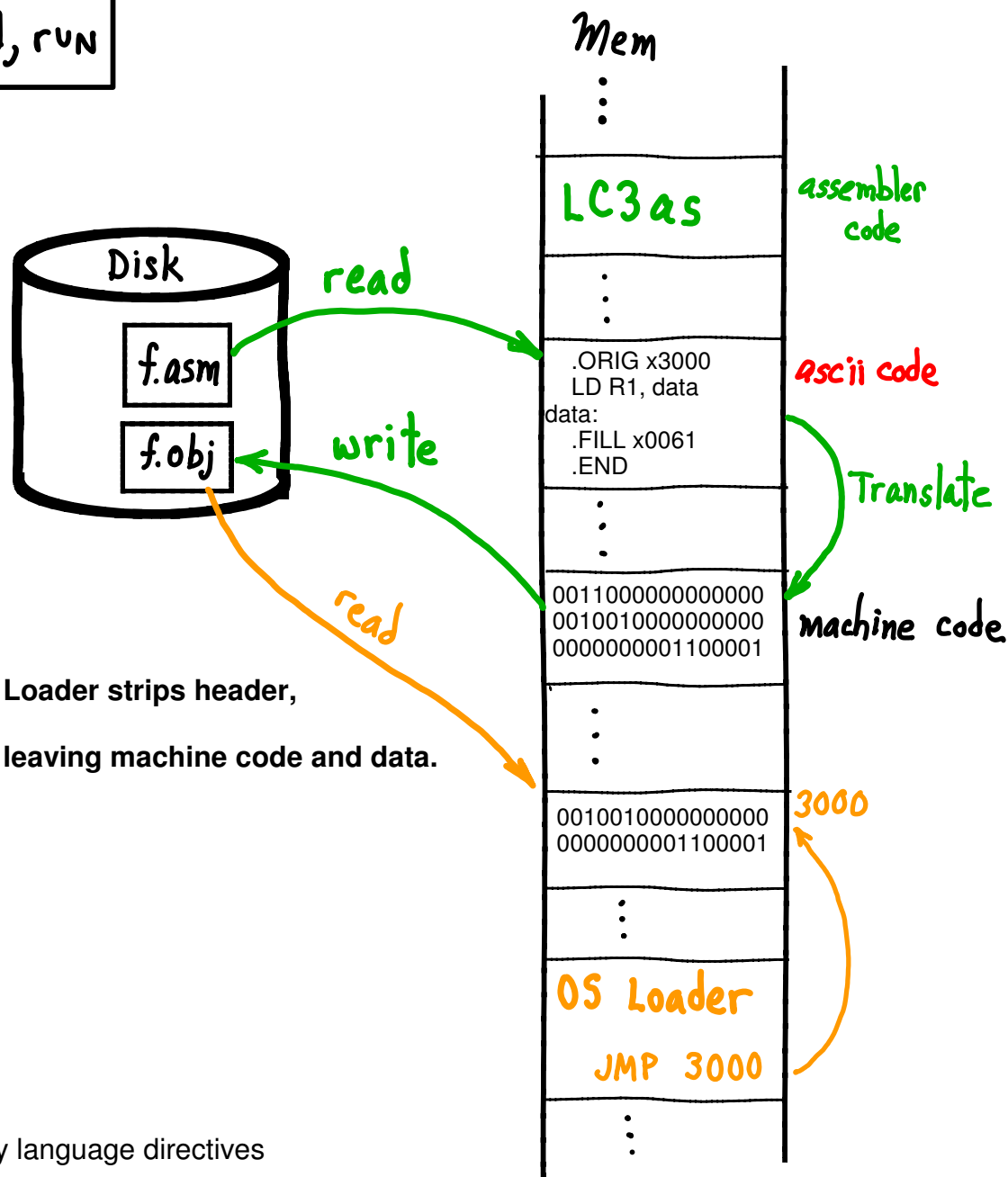
Symbol Table

STRING	offset
"here"	3003

```

0011 0000 0000 0000
0001 001 010 000 011
0000 111 000000001
+1 0101 0110 0111 1000
-2 0001 001 010 000 011
0000 010 111111110 here
    
```

Translate, load, run



More on assembly language directives

```
.STRINGZ "abcd"
.FILL x0061
.FILL x0062
.FILL x0063
.FILL x0064
.FILL x0000
```

Same as { 'a', 'b', 'c', 'd', NUL }

```
.BLKW 3
.FILL x0000
.FILL x0000
.FILL x0000
```

Same as { }

NB--Loader could do part of job of assembler: leave ".BLKW 3" in header, fill memory at load time.

