Project04 — remaining details
    Project03 offer
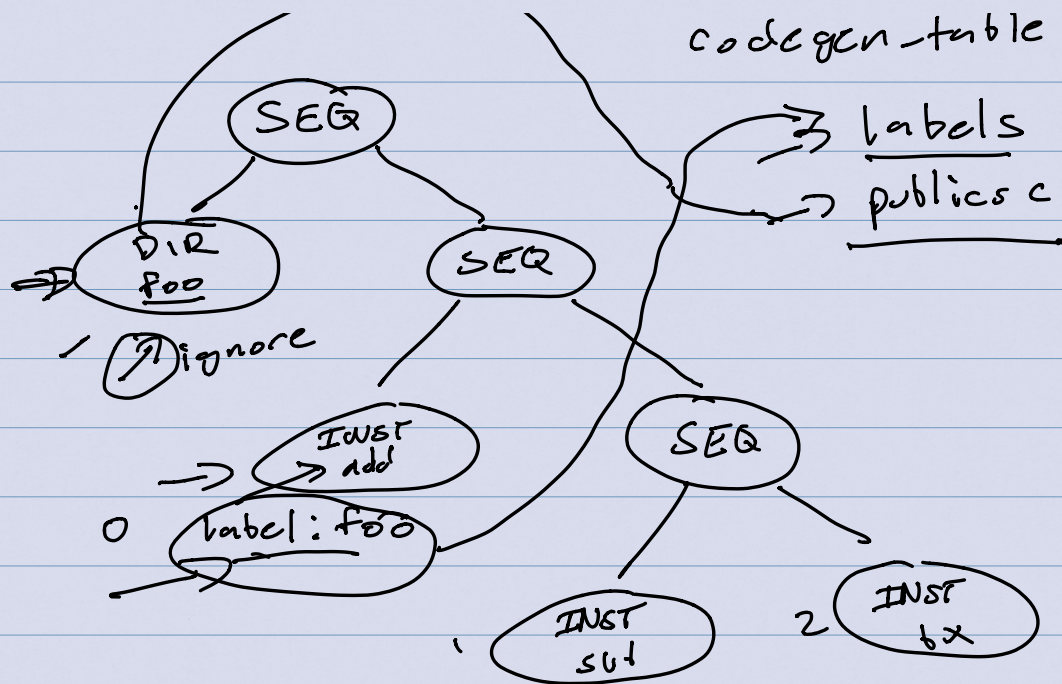
ELF

    object files
    executable file

$.count_main ⟶   MEM.
                          STACK
         ↑
        Kernel   ⟹ DATA
                            ⟹ CODE
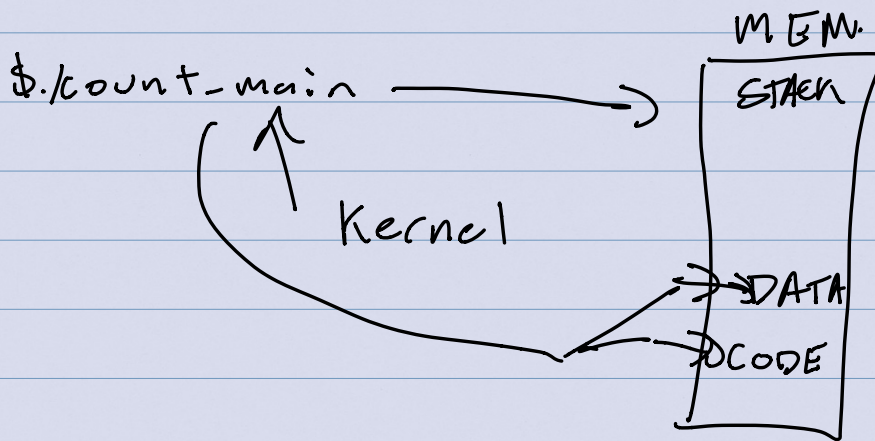
TK-DIR TK_IDENT

.global foo

foo:
0   add
1   sub
2   bx

codegen-table

SEG

DIR
foo

ignore

SEG

INST
add

label: foo

0

SEG

INST
sub

INST
bx

2

labels

publics c

offsets

when add global dir label to
publics → offset should be 0

when add label to labels array
we need offset / index

offsets → word offsets
instruction indexes
↓
ELF
↓ convert to
byte offsets
x4.

lsr , lsl   cmp , mvn

lsl   r2, r3, #2
      ↑   ↑
      rd   rm

E1A0 2103

1110 0001 1010 0000 0010 0001 0000 0011
cond dp I opcode S   Rn   r2   r3
 E   0   mov

00010, 00 0
      shamt   └→ lsl
        2

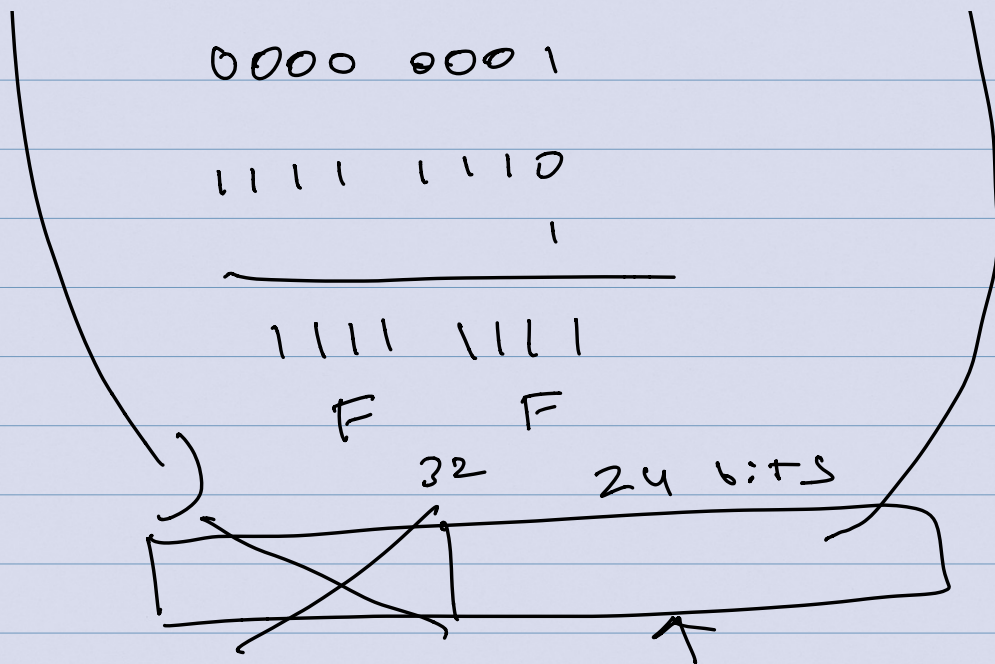target_index = codegen_get_index (free, label, 0)
         ↘   index   b label
                        ct→len

uint32_t
word_offset = target_index - (index + 2)

| cond | 101 | L | offset
                          ↑ nar

0000 0001

1111 1110
       1
_____

1111 1111

 F   F

        32   24 bits

cmp ⇒ does not use rd , S bit set to 1
                              CC

mov r0, #-1

→ mvn r0, #00
   opcode

FFF FFFF (-1)