

Lab 1

Piazza

Survey on Lab 1 and Raspberry Pi HW

Labs →

Project 02 - ntlang

little language for working with decimal, hexadecimal values, binary values.

bit manipulation

$x = 0xFABC1234 \gg 24$

language

Project 03 - ARM Assembly

foo:

add r0,r0,r1
bx lr

Project 04 - ARM Assembler

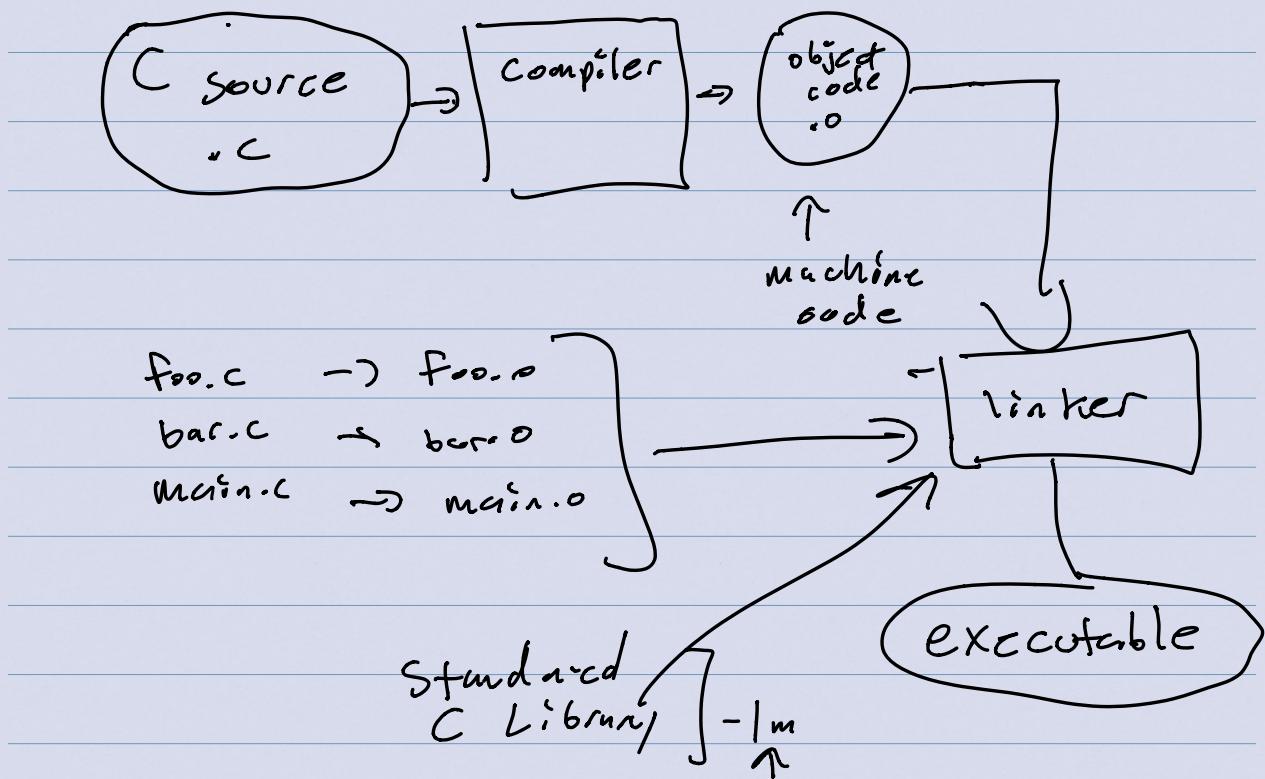
language

→ machine code

E080 0001

E12F FF1E

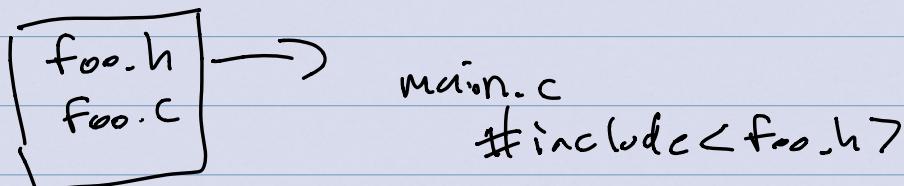
C programs



.h header files → datatypes structs prototypes func

```
struct node_st {  
    int x;  
};
```

```
int foo (int x; int y);
```



int foo (int x, int y) {
 return x+y; \Rightarrow

Machine code
E080 0001
E12F FF1E

3

1. FILE I/O - read a file into memory
 - * 2. Scanning - Scanner identify tokens
 3. Parsing - Parser create a tree
- \rightarrow 4a Compiler Code generator \rightarrow machine code
OR $\xrightarrow{\text{Linking}}$
 \rightarrow 4b Interpreter \rightarrow

ntcalc

x = 1

y = 2

z = x + y

printf("%d\n", z)

Scanning

(1) \downarrow (2) \downarrow
(x) \downarrow (=) \downarrow (3) \downarrow (*) \downarrow (4) \downarrow (0) \downarrow

enum
TK_INTLIT⁰, TK_PLUS¹, TK_EQ², TK_INTLIT³
TK_INTLIT⁰, TK_EQ², TK_INTLIT⁰
TK_MULT³, TK_INTLIT⁰

x = vclvvv \downarrow z;
 \downarrow \downarrow

1. + - * / =

2. ignore whitespace

3. integers 1, 37,

4. identifiers foo, x, ..., pos-x

EBNF - Extended Backus-Naur Form

symbol ::= '+' | '-' | '*' | '.' | '/' | '='

↗
production
rule

tokens ::= (symbol)* 0 or more

symbol ::= '+' | '-' | '*' | '.' | '/' | '='

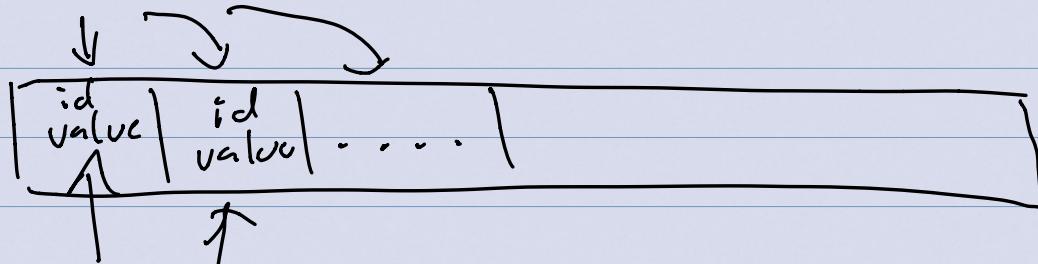
+

++

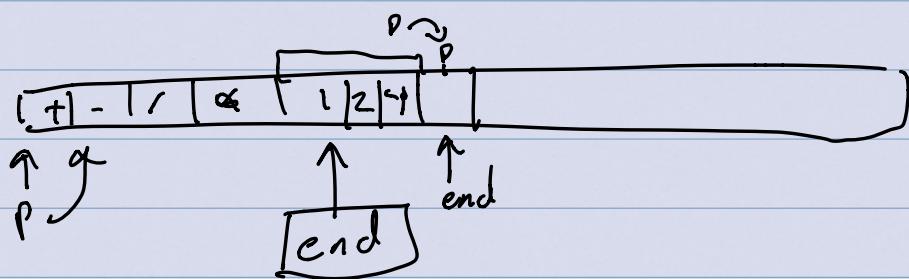
+-*+

[TK_PLUS, TK_PLUS, ...]

token table



scantokenst



[end]