Week 2.b
CS3650
01/17 2024
https://naizhengtan.github.io/24spring/

1. A life cycle of a program
2. Why C?
3. C basics
4. C pointers
5. C arrays
6. C string

----

unsigned int

10100.. ⇒ decimel

$01_b \Rightarrow 1$

$11_b \Rightarrow 3$

signed int

positive
+0

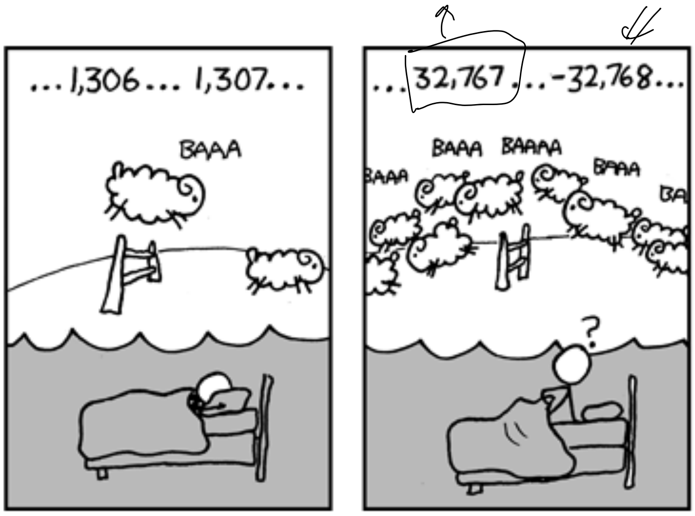
10.....0 ⇒ - largest

$11\cdots1_b \Rightarrow -1$

16 bits 0x7fff

01------1 ⇒ 10.....0

16bits

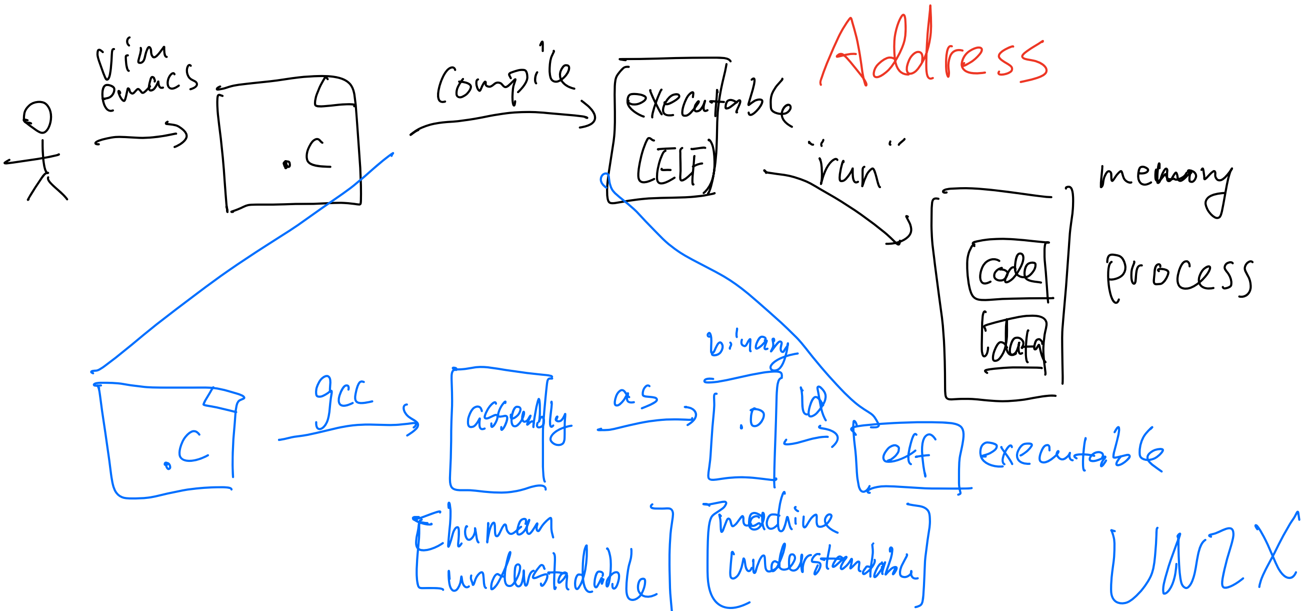


• CPU

• memory

• disk

array[10]



C basics

A. control flow

 - branches:

if, else, else if   goto

```
switch (var) {
   case 1:
   case 2:
   :
}
```

 - loops:

```
while ( ) { ◯ }
 do { ◯ } while ( );
  for(;;) {   }
```

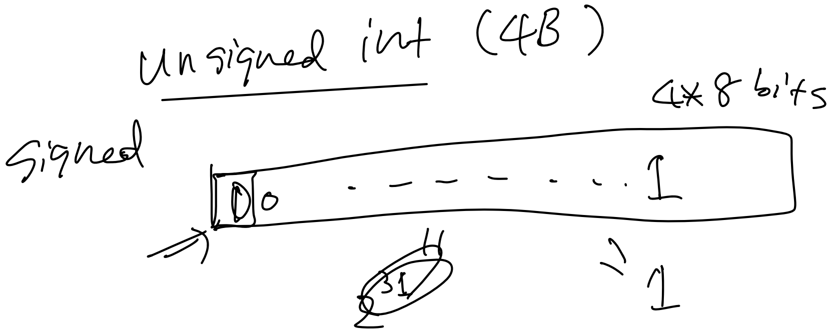
B. functions and scope

 - function definition:

int b = 100;

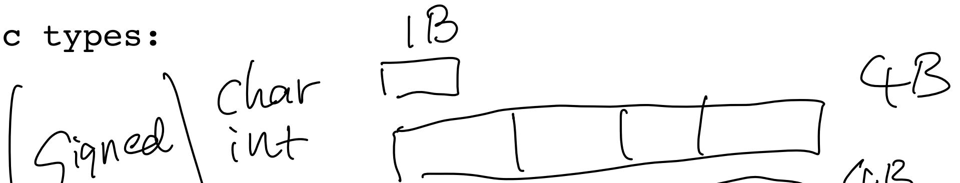
```
int foo (int x, int y) {
        :          int a = 1;
    return XYZ;
}
```

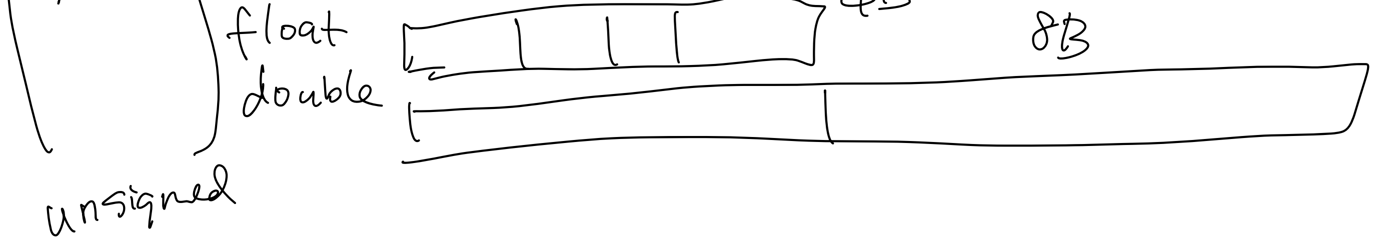
 - variable scopes, local and global variables:



C. types & operators

 - basic types:





- assignment: =

int a = 1;

- arithmetic operators: +, -, *, /, %

10 % 4 = 2

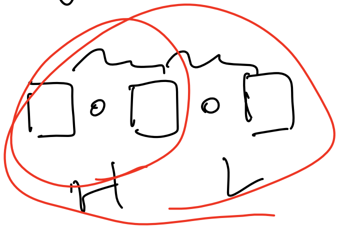
- relational operators: <, <=, >, >=, ==, !=, &&, ||

=

AND OR

- precedence and associativity (tricky)

$\boxed{a} = \boxed{b} == \boxed{c}$



$A + \underbrace{B * C * D}$

① tmp1 = B*C

② tmp2 = tmp1*D

③ result = A + tmp2

- A[B]->C

① tmp1 = A[B]

② result = tmp1->C

- (int) A[B]

① tmp1 = A[B]

② result = (int) tmp1

- A = B == C

  ① B == C

  ② A = tmp1

0 ⟺ False
others ⟺ True

- C pointers

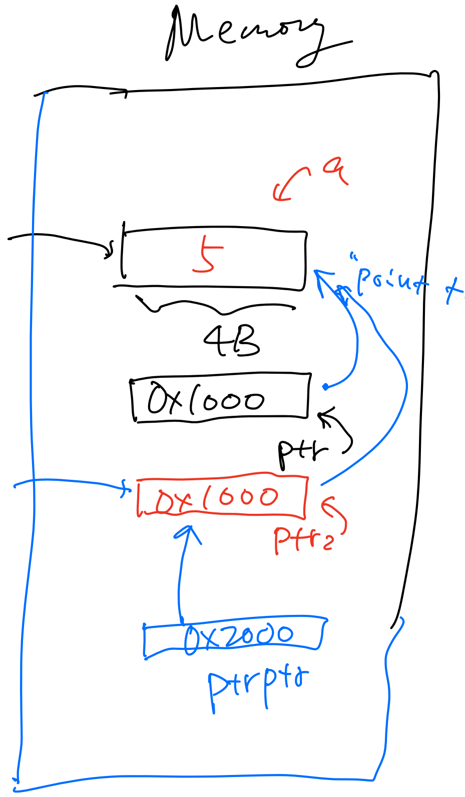
pointer ⟺ address

$\underbrace{\text{int} *}\ \underline{\text{ptr}} = \underbrace{0x1000}_{\text{Address}};$

int $\underline{a}$ = 5;

int * ptr2 = $\underline{\&a}$;

int ** ptrPtr = &ptr2;



ptrptr

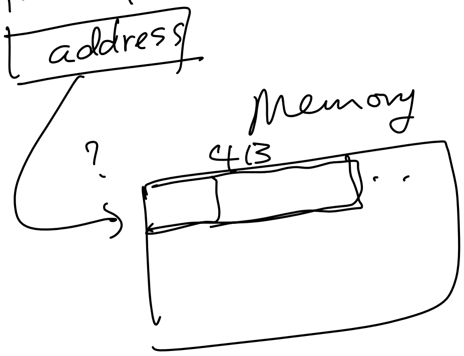
foo( ) {

ptr2 ← address ..

}

char * cptr*

int * ptr;



- pointer arithmetic ⟸
  - "int *" "char*"
  - address size ⟸ CPU
  -

ptr + 1 ⇒ +4

int * ptr → 4B

char * cptr → 1B

Cptr+1 ⇒ +1