

CS110 Review: C

C Compilation:

(Abstract Syntax Tree)

main.c \rightarrow pre-process: **macro**, **#include** \rightarrow check errors/生成AST
 \rightarrow AST \rightarrow LLVMIR, then 生成汇编 (.s file) \rightarrow 汇编 \rightarrow 机器码
 \rightarrow machine code object file (.o file) (main.o) \rightarrow Linker* \rightarrow main.out

*: $\begin{matrix} \text{lib.o} \\ \downarrow \\ \text{main.o} \end{matrix} \rightarrow \text{Linker} \rightarrow \text{main.out}$

*: macro convention: **在哪儿都加括号!!**

Eg. $\# \text{define MAGO}(x,y) \quad \text{sqrt}(x*x+y*y)$
 $\# \text{define MAGI}(x,y) \quad (\text{sqrt}(x*x)+y*y)$

则 $\text{MAGO}(i+1, j+1)$ output: $\text{sqrt}(i+1*i+1+j+1*j+1)$

$\text{MAGI}(i+1, j+1)$ output: $\text{sqrt}(i+1)*i+1+j+1*j+1$

因此, 尽可能少用宏, 几乎只有很小的 speed up

Size of type:

32 bit 蓝	char	short	short int	int	long int	unsigned int
64 bit 红	1 1	2 2	2 2	4 4	4 8	4 4
	void*	size_t	float	double		
	4 8	4 8	4 4	8 8		

Diff: long int = 电脑多少位 = void* = size_t

一个地址在几位 PC 便几位

\downarrow 设计目的便是对齐 32/64 bit

"True or False" Boolean in C:

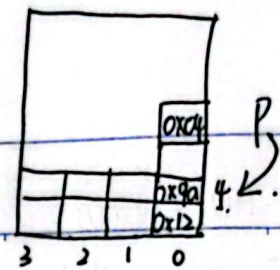
$\text{FALSE} \Leftrightarrow 0(\text{int})$. $\text{NULL}(\text{pointer})$; $\text{TRUE} \Leftrightarrow \neg \text{FALSE}$

Address vs. Value

Memory 其实是就是 byte array!



△: malloc △: dangling pointer: function
 realloc (初始化) 内开局部 var, return 其地址;
 realloc 但它在 heap 上, 故会回收



- 一个 cell, i.e., 一个 byte; int: 4 bytes; char: 1 byte.

因此存放不同数据的结构体有 alignment 现象!

- 一个地址多少 byte: up to PC. CS110 默认 32 位, 4 byte.

- 一个地址 \Rightarrow particular memory location; Pointer 就是存放地址的变量.

Syntax of Pointer:

&: 取 address. *p: 解引用, 返回 p 地址对应 value

注意函数传参: 是传地址! (引用传递; 值传递 \Leftrightarrow copy init)
 避免指针这个 variable 声明时不告诉它地址 (Undefined Behavior)

Array:

string: 最后有 '\0' (NULL) Byte (NULL Terminator).

Eg: char s1[] = "abc" \Rightarrow {a, b, c, '\0'}

char s2[] = "abc" \Rightarrow {a, b, c}

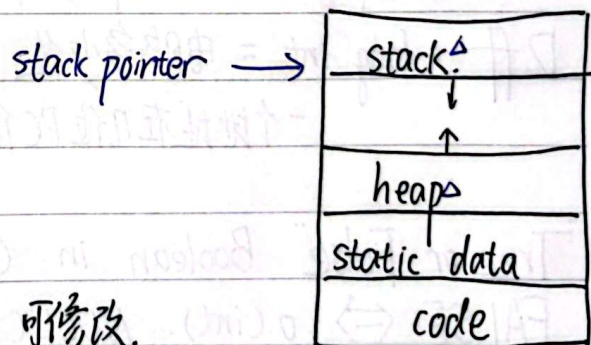
*: Array variable is a "pointer" to the first element

arr[2] \sim *(arr+2). *: type* p: p 指向地址: p * sizeof(type)

Key Concept: 传参时欲对传入变量就地操作, 传其指针!

我有 int array 的指针 int *q, 欲指向下一个地址 (令 q).

则函数应传入 int **p



C Memory Management:

stack: function 中局部变量

heap: malloc() 开辟动态内存

static data: function 外声明的变量, 可修改.

code: (a.k.a. text) 不可修改

*: 只能 Array 隐式转至 pointer! 反之不行!

Campus △: 传参只能传 pointer! 传 array 也行, 但严格转化为了 pointer!

