# 上一堂课内容回顾：

1. Sort 函数不懂
2. 关于代码笔记中的一个注释错误

北京-Jero提出来的： 二级指针 中的注释错误

1. 作业

# 题外话：

NDK 开发有什么用？

驱动，音视频，图片压缩，智能移动终端，AR，VR，都需要NDK作为基础

# 函数指针

第一步：什么是函数指针

就是指向函数的指针，

表达方式：带括号 int(\*p)(int, int);

表达方式： 不带括号 int\*p(int, int);

区别是什么？

区别就是一个是天一个是地。

第二步：函数为什么可以赋值给函数指针

打印一个函数的地址。说明函数名本身

第三步： 进行函数指针的运算举例

非void 类型

Void 类型

# 内存分配

1. 内存空间的分区是怎样的？

C 语音根据内存中数据存在的时间(生存期)不同，将内存空间分成了三个区。

1 程序区：用于存储程序的代码，就是程序的二进制代码

2 静态存储区： 用于存储全局变量和静态变量，这些变量的空间在程序编译时就已经分配好

3 动态存储区： 用于在程序执行时分配的内存， 分为：堆区（heap），和 堆栈区（stack）

堆区： 用于动态内存分配，就是程序运行时内存分配函数来分配的区域

栈区： 由编译器自动分配释放 ，存放函数的参数值，局部变量的值等，

这些内存区域会自动释放。2M空间

举例说明：

//int a[1024 \* 1024 /4]; 在栈区存放的

//malloc(1024 \* 1024 \* 10 \* sizeof(int)); 在堆区中存放

使用一个死循环来申请堆空间。

1. 内存分配函数的使用

void\* malloc (size\_t size)

malloc使用，（int \*）malloc（size）； 分配一块指定size 的内存。返回值是一个void \*

所以基本上使用的时候都需要给他们强制转换。

返回值 为什么是void \* ？

1. 没办法确定

因为C 语音没有多态，内存申请的块的单位只有申请人自己知道，如果确定了返回值，那么函数的数量就会暴涨，而且，对于自定义的类型更加没法确定；

1. 地址是4个字节

其次，返回的一定是一个地址，而地址就绝对是一个4字节的数据，所以，它可以任意的强制类型转换。所以，这个void\* 就是研究C语言的人想出来的一个 多态。

例子：int\* p = malloc(1024 \* 1024 \* 10 \* sizeof(int));

 void\* calloc (size\_t num, size\_t size);

动态分配内存空间并初始化为0。

表示分配 num 个长度为size 的连续空间。这个空间的连续是指 size大小的连续

而且会将他们初始化为0；

int\* p = calloc(len, sizeof(int));

malloc 和 calloc的区别，主要是calloc 可以初始化，而malloc 不会去初始化。

void\* realloc (void\* ptr, size\_t size);

作用： 使用往往是ptr 的内存不够，或者太多，这个时候要将内存扩容到size，于是就会重新去分配，如果ptr 后面有空的没有使用的空间，那么就会直接在后面分配，如果，后面没有足够的空间，就会另外去区一个新的足够大的空间，然后将ptr 的值赋值过去，再释放ptr 原来的位置上面的空间。

用的很少。

1. 内存回收机制

动态分配的内存空间，系统是不会去自动回收的，所以我们需要手动来回收。

C 语音是使用void free (void\* ptr);进行内存的释放

Free 的原则，

1. 是不能重复释放

例：

Free(p);

Free(p);

1. 是释放完之后，要给指针设置为NULL，

这个是一个标志，如果不设置会有预想不到的严重后果。

例如：

void main() {

int \*p = (int \*)malloc(1024 \* sizeof(int));

for (int i = 0; i < 1024; i++)

{

\*p = i;

i++;

}

free(p);

if (p != NULL) {

printf("p 所指的内存已经释放，但是p 还在这里赖着不走，会出大事。。。");

}

getchar();

}

第三，如果再给p 第二次赋值后，再对它进行释放，会有内存泄漏，第一次的内存不会释放掉，找不到这个地址对应的指针了。所以第一次申请的空间就泄漏了

字符串

1. 什么是C 语言的字符串

Java 中有String 类型，但是不是基本数据类型，而是一个引用类型

C 语音中根本就不存在String 这个说法，都是使用了char \*

如何定义字符串

1. 字符数组

char str[] = {'c','h','i','n','a'};

char str[] = {'c','h','i','n','a','\0'};

char str[10] = "china";

//可以修改

str[0] = 's';

说明，可以修改的原因:首先，str 定义的是一个字符数组，相当于定义了一片空间来存放“china”，又因为字符数组是相当于把字符一个一个的存进去，所以编译器把这个语句解析从了，将字符串中的字符一个一个的赋值到str 对应的地址中去了。

1. 字符指针

Char \* str = “china”；

说明：是否可以修改str[2] = b;

不能修改str[2] = ‘b’

把“abc”赋值给一个字符指针变量时，如： char\* ptr = “abc”；因为定义的是一个普通字符指针，并没有定义空间来存放“abc”，所以编译器得帮我们先找个地方存放“abc”，显然，把这里的“abc”当成常量并把它放到程序的常量区是编译器最合适的选择。所以的那个你去修改 char\* ptr = “abc” 中的值，如：ptr[0] = “g”的时候，会报错，因为这个地址里面存的是常量，常量是不能修改的。

1. Char \*str 和 char str[10] 区别？

北京-平福。

区别很大，第一个存储区域不同

第二个存储内容也不同

第三类型也不同

相同点： 都是用于存字符串而已

1. 字符“Hello world” 字符常量的基本说明

Char \*str = “Hello world”；

因为 “hello are you” 出现在一个表达式中时，“hello are you”使用的值就是这些字符所存储的地址（在常量区）。所以这个地址可以赋值给一个 char 类型的指针变量注意：“hello are you”作为字符数组初始化的时候就不是一个常量，其他情况下是一个常量。

举例说明：

成都-安安 同学使用情况。

5 字符串函数的相关使用

作业：1 熟练使用动态分配内存函数和释放内存的函数

2. char \*str = “hello world” 和 char str[20] = “hello world”; 区别总结？