



C语言高级编程-内存管理

主讲：小美老师

创客引领未来

扫微信二维码 获取更多信息





lemon

昵称：小姜老师
华清创客学院，嵌入式讲师

makerU

课程目录

- 存储模型(C基础补习中已讲)
- 内存管理

- C/C++定义了4个内存区间：
 - 代码区/全局变量与静态变量区/局部变量区即栈区/动态存储区，即堆区。
- 静态存储分配
 - 通常定义变量，编译器在编译时都可以根据该变量的类型知道所需内存空间的大小，从而系统在适当的时候为他们分配确定的存储空间。
- 在栈上创建。在执行函数时，函数内局部变量的存储单元都可以在栈上创建，函数执行结束时这些存储单元自动被释放。栈内存分配运算内置于处理器的指令集中，效率很高，但是分配的内存容量有限。

- 动态存储分配
 - 有些操作对象只有在程序运行时才能确定，这样编译器在编译时就无法为他们预定存储空间，只能在程序运行时，系统根据运行时的要求进行内存分配，这种方法称为。
 - 所有动态存储分配都在堆区中进行。
 - 从堆上分配，亦称动态内存分配。程序在运行的时候用malloc申请任意多少的内存，程序员自己负责在何时用free释放内存。动态内存的生存期由我们决定，使用非常灵活，但问题也最多。

- 堆内存的分配与释放
 - 当程序运行到需要一个动态分配的变量或对象时，必须向系统申请取得堆中的一块所需大小的存贮空间，用于存贮该变量或对象。当不再使用该变量或对象时，也就是它的生命结束时，要显式释放它所占用的存贮空间，这样系统就能对该堆空间进行再次分配，做到重复使用有限的资源。
 - 堆区是不会自动在分配时做初始化的（包括清零），所以必须用初始化式(initializer)来显式初始化。

- malloc/free

```
void * malloc(size_t num)
```

```
void free(void *p)
```

- malloc函数本身并不识别要申请的内存是什么类型，它只关心内存的总字节数。
- malloc申请到的是一块连续的内存，有时可能会比所申请的空间大。其有时会申请不到内存，返回NULL。
- malloc返回值的类型是void *，所以在调用malloc时要显式地进行类型转换，将void * 转换成所需要的指针类型。
- 如果free的参数是NULL的话，没有任何效果。
- 释放一块内存中的一部分是不被允许的。



扫微信二维码 获取更多信息