华清创客学院
makeru.com.cn

C语言高级编程-内存管理二

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：小美老师

华清创客学院，嵌入式讲师

maker

课程目录

- 内存管理（上一讲）
- 动态内存

动态内存

- 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。

动态内存

- malloc/free
  - 注意事项：

    删除一个指针p（free(p);）,实际意思是删除了p所指的目标（变量或对象等），释放了它所占的堆空间，而不是删除 p 本身，释放堆空间后， p 成了空悬指针

    动态分配失败。返回一个空指针（NULL），表示发生了异常，堆资源不足，分配失败。

    malloc与free是配对使用的， free只能释放堆空间。如果malloc返回的指针值丢失，则所分配的堆空间无法回收，称内存泄漏，同一空间重复释放也是危险的，因为该空间可能已另分配，所以必须妥善保存malloc返回的指针，以保证不发生内存泄漏，也必须保证不会重复释放堆内存空间。

动态内存

- malloc/free
  - 注意事项：

    动态分配的变量或对象的生命期。无名对象的生命期并不依赖于建立它的作用域，比如在函数中建立的动态对象在函数返回后仍可使用。我们也称堆空间为自由空间（free store）就是这个原因。但必须记住释放该对象所占堆空间，并只能释放一次，在函数内建立，而在函数外释放是一件很容易失控的事，往往会出错。

动态内存

- malloc/free
  - 野指针：

    不是NULL指针，是指向“垃圾”内存的指针。“野指针”是很危险的。

    “野指针”的成因主要有两种：
    - 指针变量没有被初始化。
    - 指针p被free之后，没有置为NULL，让人误以为p是个合法的指针。
    - 指针操作超越了变量的作用范围。这种情况让人防不胜防。

扫微信二维码 获取更多信息