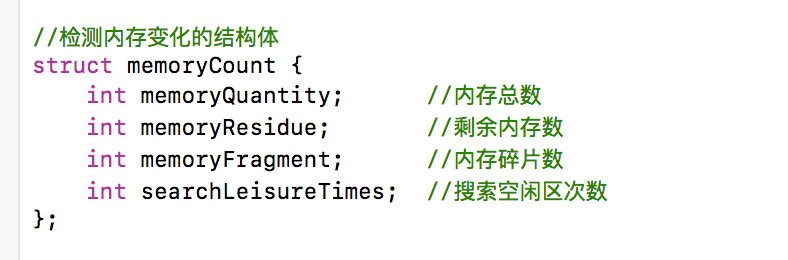
# 内存的分配和回收

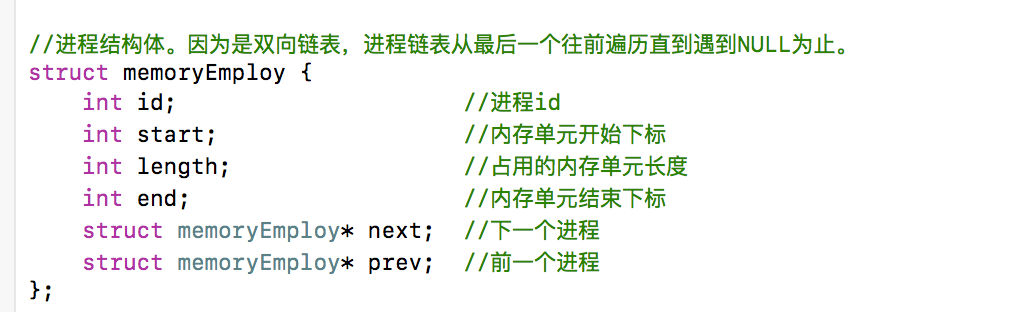
----------C语言实现

定义了三个结构体

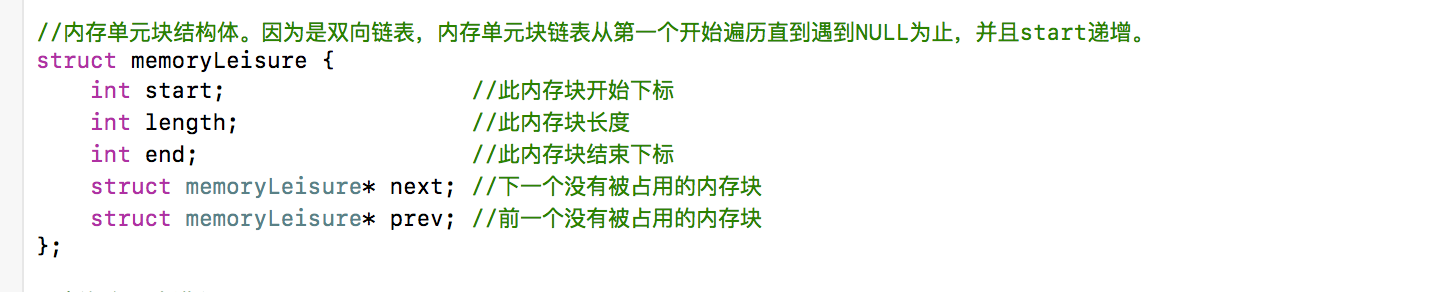
一个是检测内存变化的结构体



一个是进程结构体



一个是内存单元结构体



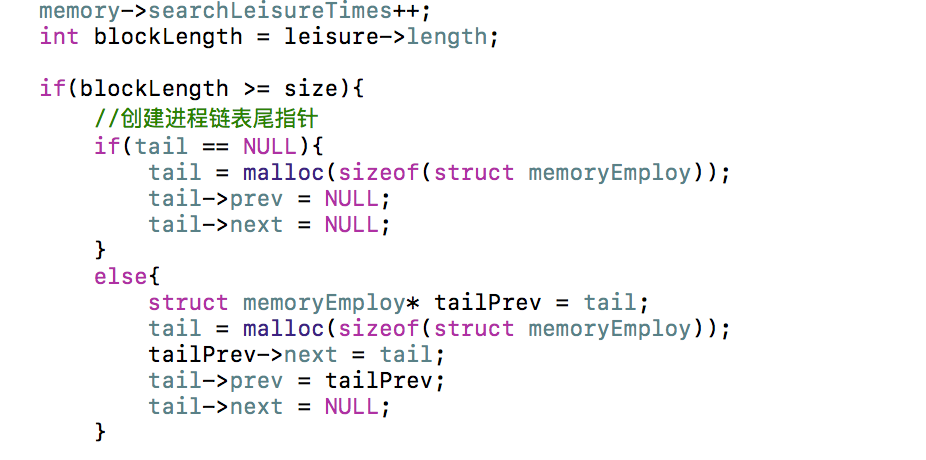
分别弄了三个结构体指针但是设置的是全局变量，这是向c语言函数传递指针出现难以预料的问题的妥协。

实现的功能有，添加一个进程，注销一个进程，和查看内存单元、进程运行和内存变化。

最主要的问题是添加一个进程和销毁一个进程时要到的问题。

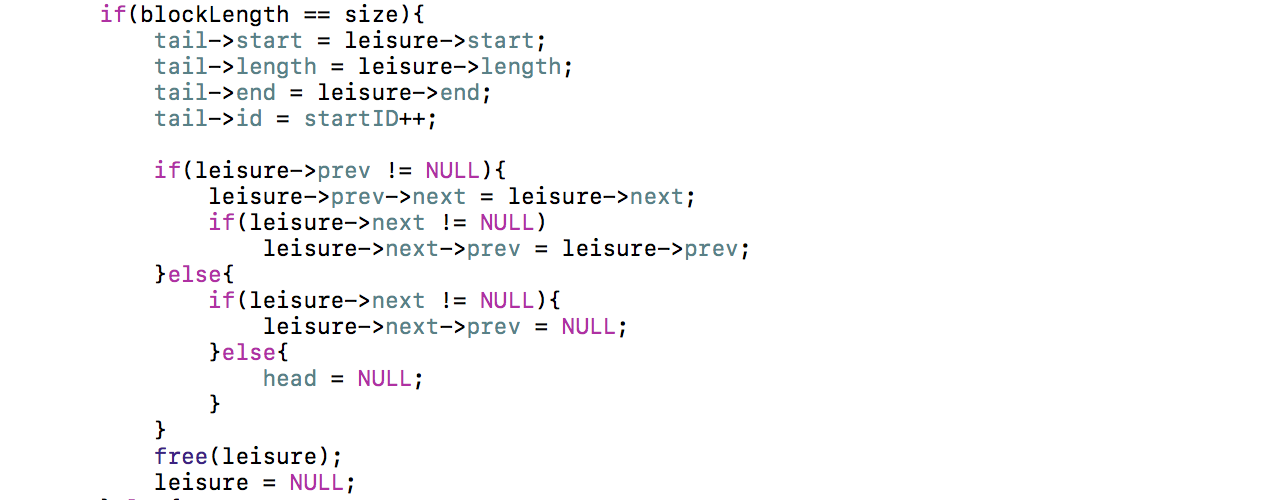
采用的策略是首次适配算法。

当需要创建一个进程时，先遍历空闲内存单元双向链表，找到一个空闲内存单元长度大于等于添加进来的进程所需的内存单元的节点，然后再次判断是等于还是大于。

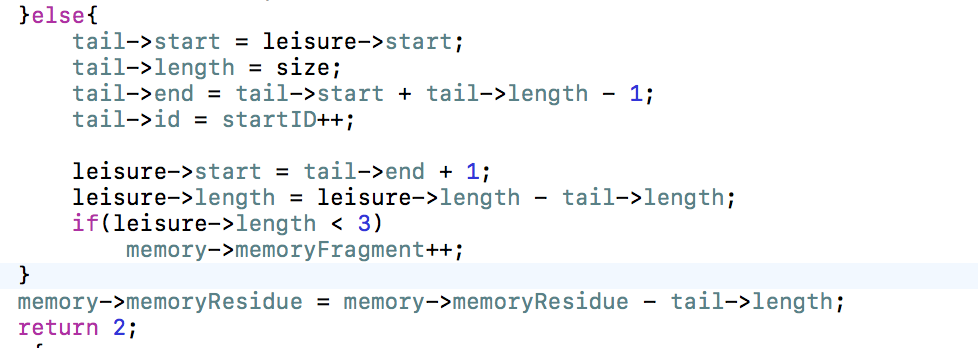


在这一步会创建一个节点来记录新添加进来的进程。

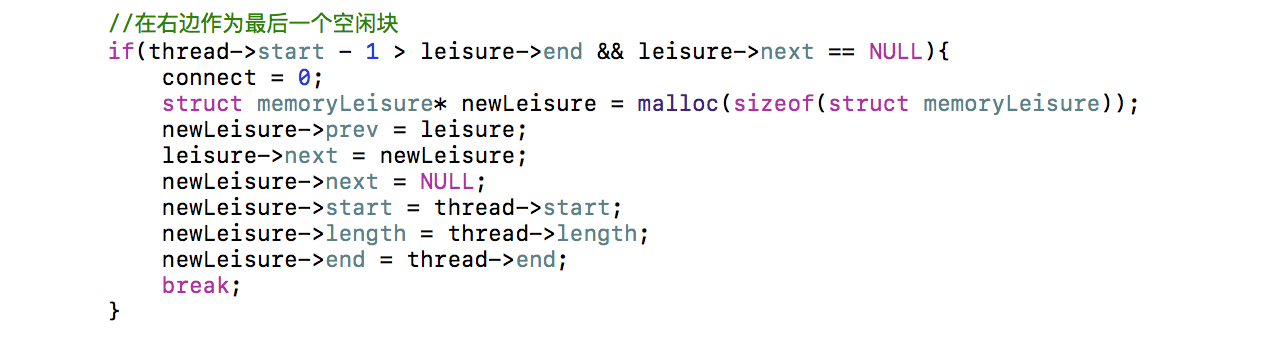
如果所需的内存单元和空闲内存单元的长度一样，则空闲内存单元就需要被替换掉，并且释放掉内存空间。

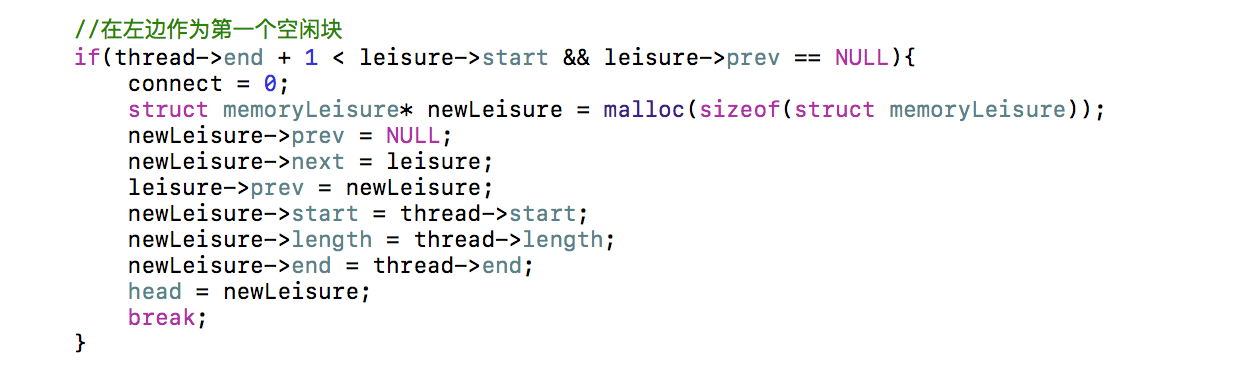


如果要是大于所需空间的话只需创建一个进程节点并且让空闲内存单元块的长度变小

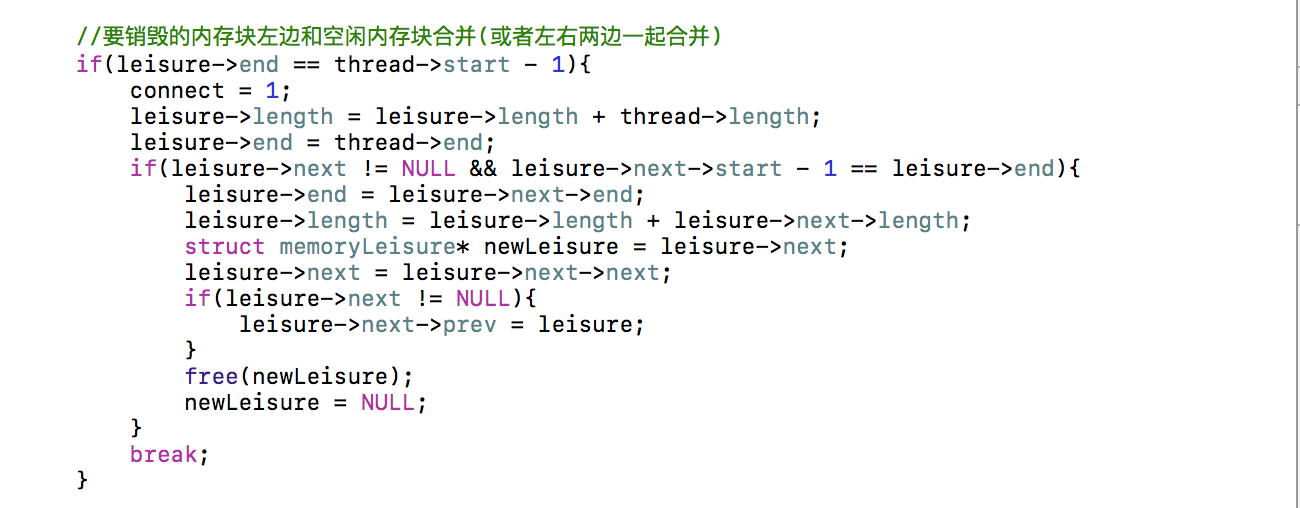


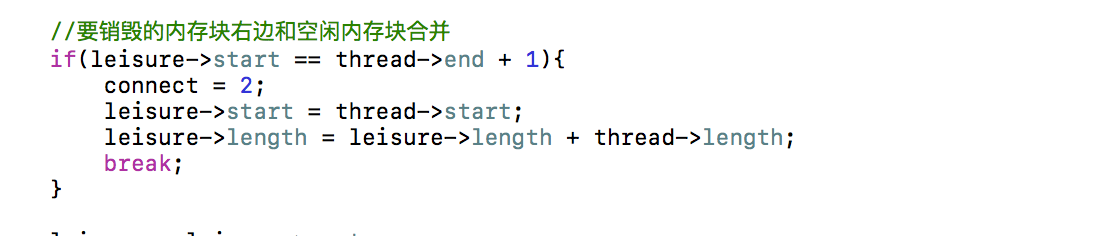
然后是销毁进程，这里情况很多。我的思路是先找到id一样的进程节点，然后进程节点的开始内存单元下标及结束内存单元下标和空闲内存单元块的对比来实现定位然后再来结合或者单独作为节点。这里有一点就是我的空闲内存单元块节点是按照下标递增的格式来组成链条的，这个是销毁进程函数的关键条件。













最后截一个我完成的运行实例。

