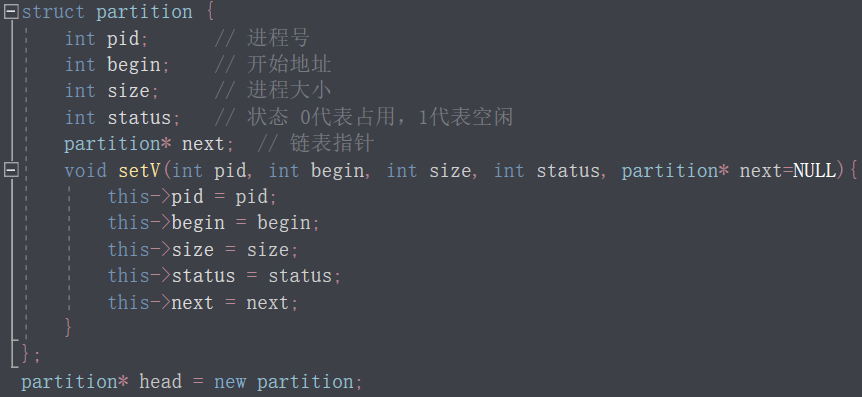
动态分区

2054305 陈敬麒

## 一、项目简介

采用动态分区分配方式，对内存进行分配。用户可以选择首次适应算法和最优适应算法。首次适应算法：为申请内存空间的任务分配首个足够大的孔。最优适应算法：为申请内存空间的任务分配最小的足够大的孔。初始态下，可用的内存空间为640k，用户可选择是手动输入任务申请及释放请求，还是直接参考系统设定的默认任务。输入后系统会显示每次分配或回收后空闲和非空闲分区链的具体情况。

## 二、数据结构的设计



采用链表的模式——

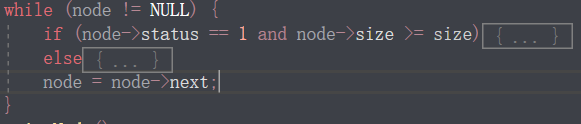
初始为单一分区节点，size为640

进程申请会拆分出新的分区，用链表链接

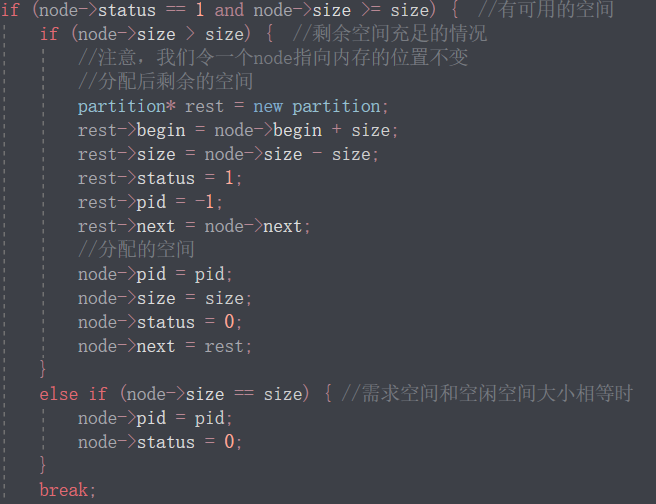
进程释放会释放该分区节点，如果周围有其他空闲分区，将进行合并

## 算法一：首先适配算法

While循环从头遍历链表，寻找地址最小的可适配分区



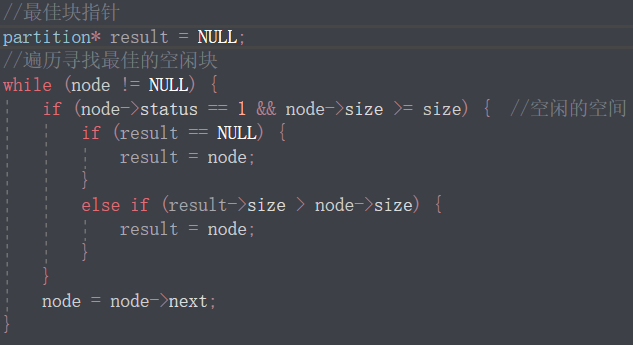
寻找到可是配分区后，拆分出新的分区（在链表相应的位置new一个分区）



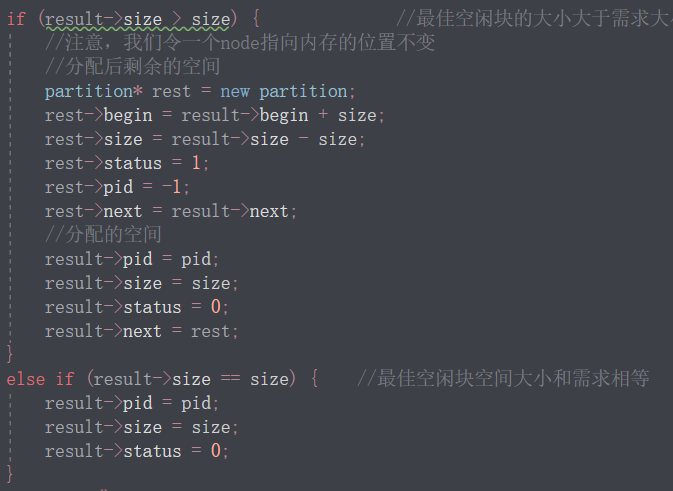
## 算法二：最佳适配算法

遍历寻找最佳的空闲块——

从head开始遍历链表：谁的分区size更小且可以大于等于需求，就分配谁



特殊情况分类处理——是否需要生成新分区



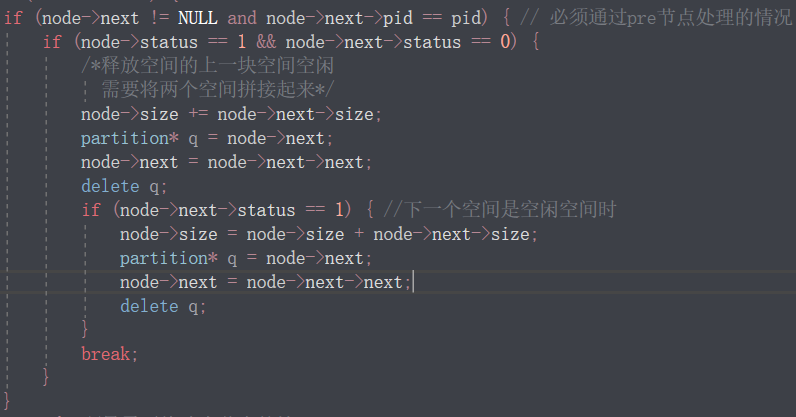
## 释放算法

* 注意——分区在释放后，需要和前面、后面的空闲分区合并。

从head开始，遍历链表，寻找该进程

情况一：必须通过pre节点处理的情况。

释放空间的上一块空间空闲，需要将两个空间拼接起来



情况二：下一个空间是空闲空间时

