同濟大學
TONGJI UNIVERSITY

项目说明文档

动态分区分配方式

指导教师：王冬青

1751984　王舸飞

1.综述

1.1 背景分析

在初始态下, 可用内存空间为640K

对于上述存储空间, 有一系列的请求序列, 要求使用首次适应算法和最佳适应算法进行内存块的分配

对于图形化界面, 要求显示出每次分配和回收后的空闲分区链的情况

1.2 功能分析

根据上述背景, 得知本次项目的主要难点如下:

1、调度算法: 需要使用两种方式对内存进行分配, 包括首次适应算法和最佳适应算法

2、图形化界面: 需要设计内存的显示和空闲分区块的显示

2.设计

2.1 解决方案设计初步

1、调度算法: 首次适应算法的思想是, 从低地址处开始对空白链进行查找, 一旦找到相应的分区立即进行分配. 显然, 该算法倾向于使用低地址部分进行分区, 而高地址的内存块很少被使用, 保留了高地址的大空闲区; 但该算法的缺点也是明显的: 低地址部分不断被划分, 留下了很多难以利用的小空闲区, 而每次查找都从低地址开始, 无疑增大了查找的开销

最佳适应算法的思想则是对空白链进行排序, 每次找到一个最小的块进行分配; 该算法被称为最佳的原因是:每次都能找到一个最适合大小的空闲块进行分配, 但是根据该算法的思想, 每次分配后的剩余空间一定是最小的, 会在存储器中留下很多难以利用的小空闲区

拟采用一个结构体数组记录内存的状况, 并根据不同算法的特点给出下一次插入的起始地址

2、图形化界面: 虽然本次项目给出的测试数据是一定的, 但为了后续的观察和学习方便, 添加了自定义下一次插入算法的选择控件和自定义输入大小的输入框; 逻辑中的每次更改、插入不仅会以图形的形式更新, 也会在能保存历史记录的滚动框中进行提示, 完成内部逻辑与外部界面的统一

点击已经生成的内存块上的删除选项, 可以删除该任务并在输出框中更新空白块的情况, 便于观察

3.实现

3.1 相关变量的实现

为了实现上述功能, 首先需要一些常变量, 增加代码可读性

变量名	值	功能
MEMO_SIZE	640	指示内存的大小,后续改变内存大小时会十分方便
FIND_QUICK	0	这两个变量代表了下一次插入使用的算法标记
FIND_BEST	1	

接下来是会用到的全局变量

变量名	初始值	变量作用	调用时间
running_block	[];	代表当前运行着的内存块,按照内存地址由低到高排序	两个关键算法的计算, ui添加和删除时都需要调用
current_id	1	记录当前块的编号	添加和删除时,尤其是删除需要根据id完成逻辑删除
vacant_block	640	按地址由低到高记录空块大小	输出空闲块状态时

3.2 主要函数及逻辑的实现

函数名	功能	调用时间	备注
init()	初始化函数	页面加载完毕后	仅初始化滚动提示条

函数名	功能	调用时间	备注
createBlock(start, end, id)	使用js的方式模拟创造一个结构体	输入的大小合法并且有位置插入时	在createMission()内调用
createMission(id, length, start)	在街面上创建块、调用逻辑创建块函数,更新提示信息	判断输入合法且有空间插入时	
$(“create”).click()	判断输入是否合法,是否超出内存,若不是,调用相应算法给出起始地址并给出提示信息	生成按钮被点按之后	
del()	先在逻辑上删除该块,之后在界面上将它移除	单个块的删除按钮被点击时	
findQuick(size)	根据一个合法的大小,利用全局变量数组,计算返回起始插入地址	判断输入合法且有空间插入时	首次适应算法
findBest(size)	根据一个合法的大小,利用全局变量数组,计算返回起始插入地址	判断输入合法且有空间插入时	最佳适应算法
updateVacant()	更新vacant数组	任何一个添加或者删除后	与滚动条的更新同步

3.4 总体流程说明

进入界面后, 初始化滚动条, 给出提示信息

初始化的算法为首次适应算法, 默认内存块大小为100kB,

由于javascript是响应式语言, 处理用户输入的时间十分方便, 故后续的内容均由浏览器的内容监听器完成: 在检测到相应的操作(根据算法和大小插入块、删除块)后, 进行界面的输出和滚动条提示信息的更新

4.测试(基于Chrome)

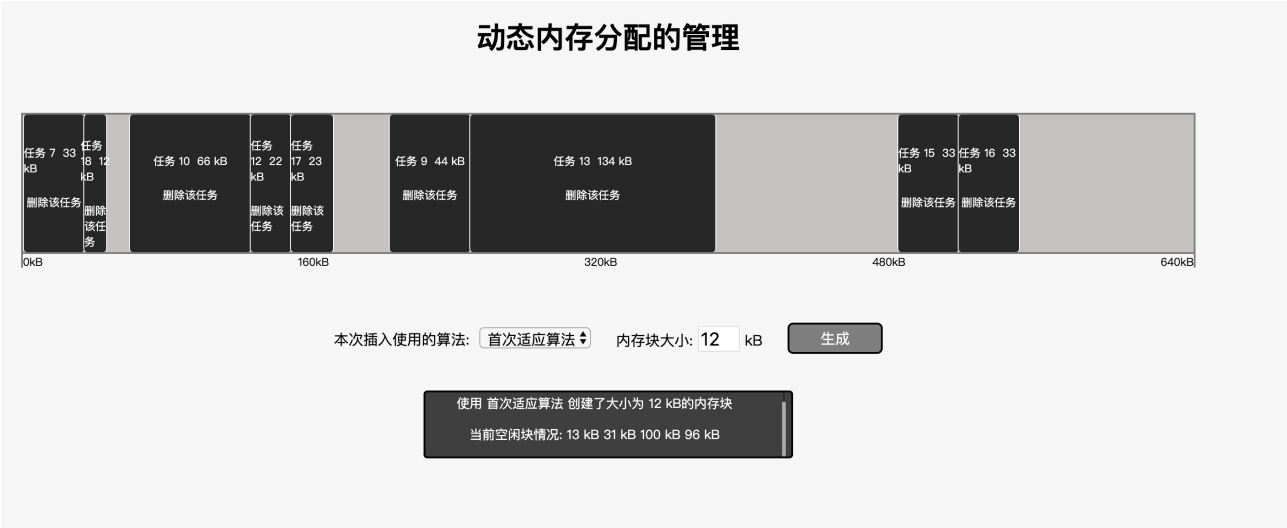
4.1 初始界面

(程序的初始界面)

4.2 运行状态测试

4.2.1 首次适应算法测试

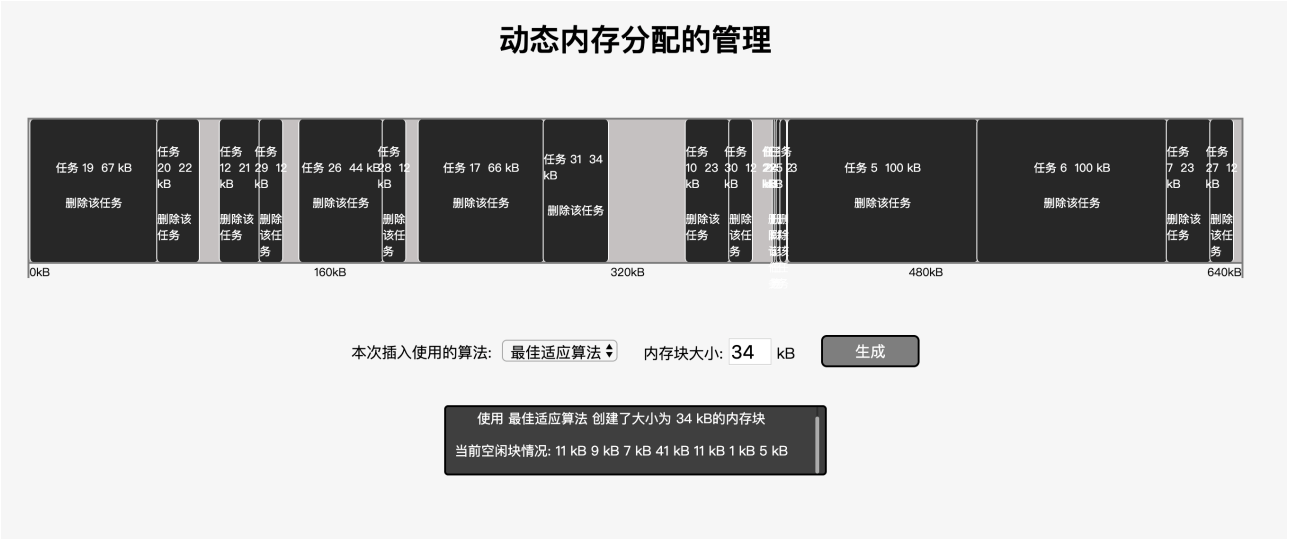
(规定测试数据的首次适应算法分配结果)



(随机测试数据的测试结果, 内存块集中在低地址)

4.2.2 最佳适应算法测试

(规定测试数据的运行结果)



(随机测试数据的运行结果,内存均匀但有较多小内存块)

4.2.3 其他输入情况测试

5.总结

5.1 开发环境总结

采用了Atom 与 Chrome 的开发者工具

5.2 开发过程总结

在设计之初,首先利用良好的前端布局和命名习惯对html和css文件进行了部署

之后, 分析了主要的设计难点, 并初步定义了大量变量和用于处理问题的函数

在具体实现的过程中, 对现存的变量和函数的必要性进行了反思, 并将重复的变量进行了删除

初步的原型完成后, 进行了大量边界测试, 进一步完善了代码的逻辑

全部逻辑测试通过后, 开始进一步思考界面的布局, 调整部分美工后, 运用mScrollBar对每一次更新状态进行了指示

由于js文件中写好了详细的注释,故在此不再将代码复制粘贴