# 操作系统内存管理项目一

## 说明文档

1452716

张尹嘉

## 一.项目分析:

1.题目要求：

假设初始态下，可用内存空间为640K，并有下列请求序列，请分别用首次适应算法和最佳适应算法进行内存块的分配和回收，并显示出每次分配和回收后的空闲分区链的情况来。

作业1申请130K

作业2申请60K

作业3申请100k

作业2释放60K

作业4申请200K

作业3释放100K

作业1释放130K

作业5申请140K

作业6申请60K

作业7申请50K

作业6释放60K。

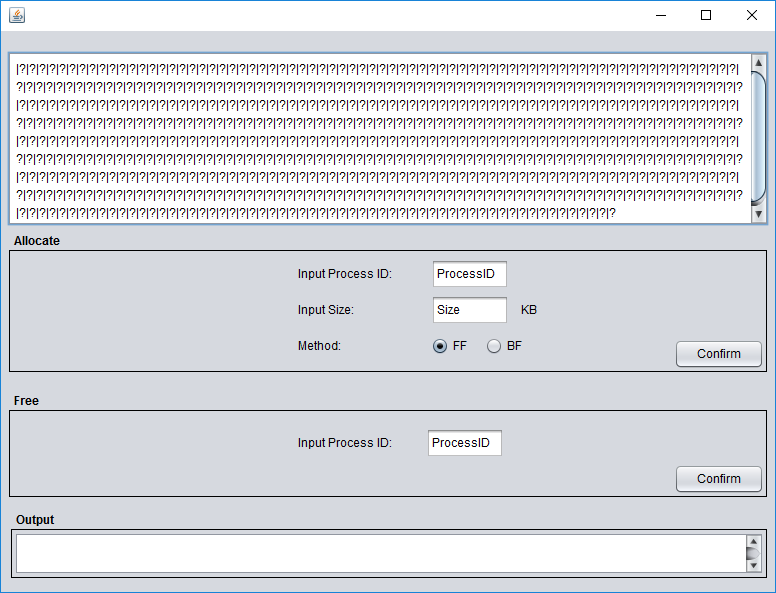
### 2.项目分析：

根据题目要求，程序应该提供FF和BF两种动态分配内存的方法。因为内存是不连续的块状空间，因此应该用链表进行维护，并需要在页面上显示内存的使用情况。同时也要提供释放内存的操作。

## 二.项目环境：

本程序使用java语言进行开发，并在NetBeans上进行界面的设计制作。开发时使用的JDK版本是JDK8，但在生成时同时生成了JDK8和JDK7两个版本。具体程序在VersionJDK7和VersionJDK8两个文件夹中。源代码保存在SourceCode文件夹中，以.java文件格式形成。

### 三.使用说明：



以上是程序运行的界面。上部文本框显示了内存的使用情况。本模拟程序以1KB的内存片为模拟单位。每1KB未分配内存由 |? 代表，当该KB内存被分配后，’？’ 会被替换成该内存被分配给的作业的编号。例如，某1KB内存被分配给作业1， 则该1KB内存的标识符变化为： |? ------> |1 。

1

分配前

### 2

操作

3

分配后

如果需要释放内存，在第二个控制面板输入要释放的作业编号，如果释放成功，文本框中内存标识符会对应改变。

3

释放前

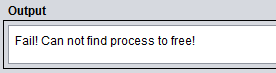


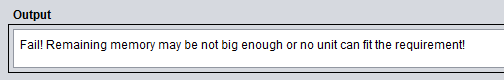
释放操作

1

释放后

界面最下面的输出面板用来显示运行结果和申请或释放内存错误时的可能错误原因。样例如下：





## 四.实现算法：

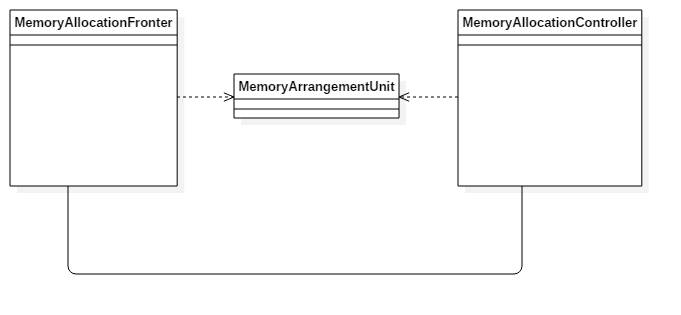
本程序的算法是维护两个链表，即繁忙链表和空闲链表。繁忙链表用来记录每个作业所占用的内存区间，空闲链表记录了内存中空闲的空间段。最先适应算法是在空闲链表中从头进行搜索查找第一个符合要求的内存段，进行内存分配后，把该作业的内存使用情况存入繁忙链表中，并更新空闲链表。最优适配算法是在链表中查找符合要求的并且空间最小的内存段并进行分配和更新。

理论上最先适配算法的链表按照内存的地址进行排序，而最优适配算法按照内存片大小以升序进行排列。但考虑到链表长度是常数级，查找算法的时间复杂度比较小，使用按内存地址进行排列时，最优适配算法查找的开销可以忽略不计，同时也考虑到最先适配算法和最优适配算法的交替使用，因此两种算法都用按照内存地址排序的链表。

在释放内存时，要把处于繁忙队列中的内存段释放，同时把该段内存插入空闲内存段链表中。在插入空闲链表时，要检查该段内存是否可以与该段内存的前一片内存段和后一段内存段合并。如果可以和前一段或后一段合并时，只要改变前段或后段的范围即可。

## 五.代码实现：

本程序代码共分为三个公共类，分别作为界面类，控制类和用于在界面和控制类之间传输信息的传输类。使用者在界面上点击了对应的按钮后，界面类会对操作进行初步的判断。如果判断通过，界面类会收集对应输入框的内容并打包成传输类传送给控制类。控制类对传送的信息进行判断和操作，并把操作结果以传输类和boolean值得方式返回界面类，界面类把操作结果显示在界面上。



繁忙链表和空闲链表分别存储在控制类中，并且以控制类的内部类作为数据结构。界面类也有内部类，其用来表示每个1KB的内存并以数组的形式进行维护。虽然可以用String类显示内存标识符，但是当作业的编号不是个位数时，在释放内存修改String时会比较麻烦，用内部类对象的数组可以简化操作代码。

MemoryAllocationFronter类：

界面类，包括Swing界面组件的对象变量，对按钮点击操作的处理以及一些初步的判断。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 权限 | 返回值 | 函数名 | 参数 | 简介 |
| public | 无 | MemoryAllocationFronter | 无 | 构造函数，初始化文本框和内部类数组 |
| public | void | main | String args[] | 主类入口 |
| private | void | AllocateConfirmButtonActionPerformed | java.awt.event.ActionEvent evt | 分配内存按钮的响应方法，对操作进行初步判断，将信息传送给控制类进行内存分配。 |
| private | void | jRadioButton1ActionPerformed | java.awt.event.ActionEvent evt | FF按钮的响应程序，设置算法 |
| private | void | jRadioButton2ActionPerformed | java.awt.event.ActionEvent evt | BF按钮的响应方法，设置算法 |
| private | void | FreeConfirmButtonActionPerformed | java.awt.event.ActionEvent evt | 释放内存按钮的响应方法，进行初步判断，将信息传给控制类进行操作 |
| private | void | SetMemoryState | 无 | 分配和释放内存后对内存状态的显示进行改变 |
| private | boolean | CheckProcessID | Int ID | 检查当前要分配的作业的编号是否已经存在 |
| private | boolean | DeleteProcessID | Int ID | 释放后删除作业编号数组中对应的编号 |

MemoryAllocationController类：

控制类，主要进行内存状态链表的维护。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 权限 | 返回值 | 函数名 | 参数 | 简介 |
| public | 无 | MemoryAllocationController | 无 | 构造函数 |
| public | boolean | allocateMemory | int method, int size, MemoryArrangementUnit ref | 分配内存的公用接口，根据Method调用对应的private方法 |
| public | boolean | freeMemory | int begin, int end | 释放内存的公用接口，调用对应的private方法 |
| private | boolean | privateAllocateMemoryByFF | int size, MemoryArrangementUnit ref | 根据最先适配算法进行内存的分配 |
| private | boolean | privateAllocateMemoryByBF | int size, MemoryArrangementUnit ref | 根据最优适配算法进行内存的分配 |
| private | boolean | privateFreeMemory | int begin, int end | 释放内存的私有函数 |

MemoryArrangementUnit类：

信息传输的数据结构类，包含了作业ID，起始内存点和终止内存点三个共有变量。