****

**课程设计报告**

|  |  |
| --- | --- |
| **课程名称** | **操作系统课程设计** |
| **题 目** | **模拟设计请求页式存储管理的分配与回收（LRU置换算法）** |
| **学 院** | **计算机科学与技术** |
| **班 级** | **计算机1701班** |
| **学 号** |  |
| **姓 名** | **张超杰** |
| **指导教师** | **蔡菁** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2020 | 年 | 1 | 月 | 10 | 日 |

**课程设计任务书**

**学生姓名： 张超杰 专业班级： 计算机1701班**

**指导教师： 蔡菁 工作单位： 计算机科学与技术学院**

**题 目: 模拟设计请求页式存储管理的分配与回收（LRU置换算法）**

**初始条件：**

1. 预备内容：掌握操作系统的内存管理的相关内容，对内存的分配与回收有深入的理解。

2．实践准备：掌握一种计算机高级语言的使用。

**要求完成的主要任务:** （包括课程设计工作量及其技术要求，以及说明书撰写等具体要求）

1. 采用请求页式存储管理方案实施内存分配和回收（使用题目指定置换算法）。具体要求如下：
2. 能够输入给定的内存页面数及页面大小，进程的个数及每个进程的页数；
3. 能够选择分配 / 回收操作；
4. 模拟当某进程提出申请内存空间的大小后，能够判定是否能满足申请；
5. 显示进程在内存的存储情况；
6. 显示每次完成分配或回收后内存空间的使用情况。

2．设计报告内容应说明：

⑴ 课程设计的目的、功能与要求（明确该选题的作用并列出所选功能及要求）；

⑵ 问题的详细描述、需求分析；（分析说明相关方案原理及具体的实验内容）

⑶ 数据结构、功能设计（给出功能结构图、处理流程图）；

1. 开发工具及核心源程序的主要部分（对主要代码段附文字注释）；
2. 测试用例、运行结果与运行情况分析；

⑹ 自我评价与总结：

i）哪些地方做得比较好或比较出色；

ii）什么地方做得不太好，以后如何改正；

iii）本设计得到的收获（在编写，调试，执行过程中的经验和教训）；

iv）完成本设计是否有其他方法（简要说明）；

v）对实验题的评价和改进意见，请推荐设计题目。

**时间安排：**

设计安排一周：周1、周2：完成需求分析及设计；

周2、周3：完成程序调试及测试；

周4、周5：验收、撰写课程设计报告。

**（注意事项：严禁抄袭，一旦发现，一律按0分记）**

# 1课程设计的目的、功能与要求

## 1.1课程设计题目

模拟设计请求页式存储管理的分配与回收

## 1.2课程设计目的

掌握操作系统的内存管理的相关内容，对内存的分配与回收有深入的理解。掌握一种计算机高级语言的使用。

通过课程设计，进一步学习和掌握有关内存分配和回收的内容，以请求页式管理为主体，完成模拟设计请求页式存储管理的分配与回收的程序的设计。将书本上有关的理论知识用于实践中，让学生实际动手操作，提高学生的实践能力，让学生对操作系统的学习更加深入。同时，课程设计也是检验学生学习成果的最直观、公平的方式。

## 1.3课程设计功能与要求

### 1.3.1课程设计的功能

模拟请求页式存储管理的分配与回收程序可以正确地模拟请求页式存储管理的分配与回收，可以模拟缺页中断并进行中断处理，对于缺页中断采用LRU算法进行置换。可以打印进程在内存中情况、打印进程请求队列、打印内存的使用情况等。

### 1.3.2课程设计要求

采用请求页式存储管理方案实施内存分配和回收。具体要求如下：

①能够输入给定的内存页面数及页面大小，进程的个数及每个进程的页数；

②能够选择分配 / 回收操作；

③模拟当某进程提出申请内存空间的大小后，能够判定是否能满足申请；

④显示进程在内存的存储情况；

⑤显示每次完成分配或回收后内存空间的使用情况。

# 2问题的详细描述、需求分析

## 2.1问题描述

使用一种高级语言，更具题中地功能和要求，设计模拟请求页式存储管理的分配与回收程序。题中所给功能必须全部实现，可自选加入其他功能。

具体内容：因为是模拟程序，所以要用户手动输入一些信息。给出菜单供用户选择分配、回收、查看进程和内存情况等，并且手动输入内存大小以及页的大小和进程相关信息等，还可以输入请求页的队列，对页进行请求分配。

## 2.2原理描述

### 2.2.1请求页式存储管理原理描述

页式管理是一种内存空间存储管理的技术，页式管理分为静态页式管理和动态页式管理。基本原理是将各进程的虚拟空间划分成若干个长度相等的页(page)，页式管理把内存空间按页的大小划分成片或者页面（page frame），然后把页式虚拟地址与内存地址建立一一对应页表，并用相应的硬件地址变换机构，来解决离散地址变换问题。页式管理采用请求调页或预调页技术实现了内外存存储器的统一管理。

请求页式管理是动态页式管理的一种方式，是在静态页式管理（上述页式管理技术）的基础上发展起来的。其中主要有页面分配、地址变换和缺页中断等功能。

主要的数据结构：页表、请求表和存储页面表

1、分配页面与回收：采用局部分配策略，每个进程被分配不同的物理块数，并不是将全部的的页调入内存（不同于静态页式管理技术），而是根据访问的需要进行请求调入。如果请求的页面不在内存中，则进行缺页中断，将外存中相应的界面调入内存。

2、地址变换：同静态页式管理技术相同，由硬件自动完成地址变换，其中涉及到快表结构。在本次的课程设计中并不涉及到地址的变换，所以不进行过多阐述。

3、缺页中断：缺页中断主要涉及到页面置换的算法，随机淘汰法、轮转法、先进先出算法、最近最久未使用算法（LRU）和理想型算法。下文会进行详细介绍。

### 2.2.2最近最久未使用算法(LRU)算法原理描述

1、LRU算法的基本思想是：当需要淘汰某一页时，选择离当前时间最近的一段时间内最久没有使用过的页先淘汰。该页的出发点是，如果某页被访问了，则它可能马上还要被访问。或者反过来说，如果某页很长时间未被访问，则它在最近一段时间也不会被访问。

2、在编程中的具体实现：选取以进程分配的物理块数量大小的序列范围为最近一段时间，用一个动态数组存这个序列（数组大小为该进程分配的物理块数量）。如果该页已经在内存中，则将这一段时间该页之后的数组元素左移一位，该页位置移动到数组末尾；如果不在，则淘汰数组的最左边位置的页，并将数组左移一位，把该页添加到数组尾部。

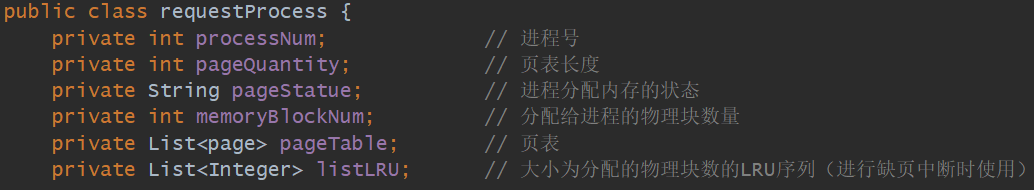
# 3 数据结构、功能设计

## 3.1数据结构设计

页表：与静态页式管理不同的是中断位判断该页是否在内存中



请求表：不同于静态页式管理，这里的分配状态指是否分配了物理块，这里页表加入到进程类里面，用于LRU算法的序列也加入到了进程类里面





存储页面表：采用位视图法进行存储。如果该页已被分配，则对应比特位为1；否则为0，具体使用动态数组存储



## 3.2功能结构图



## 3.3功能设计流程图

1、分配算法流程图（忽略地址转换）：



2、回收算法流程图：



3、分配页流程图：



# 4开发工具及核心源程序的主要部分

## 4.1开发工具

Windows 10简体中文专业版

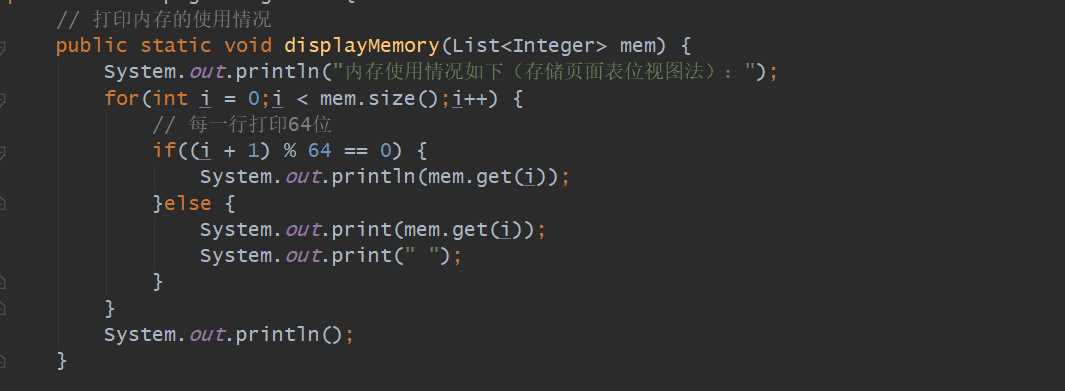
IntelliJ IDEA 2019.2.1 x64旗舰版

JDK1.8\_211

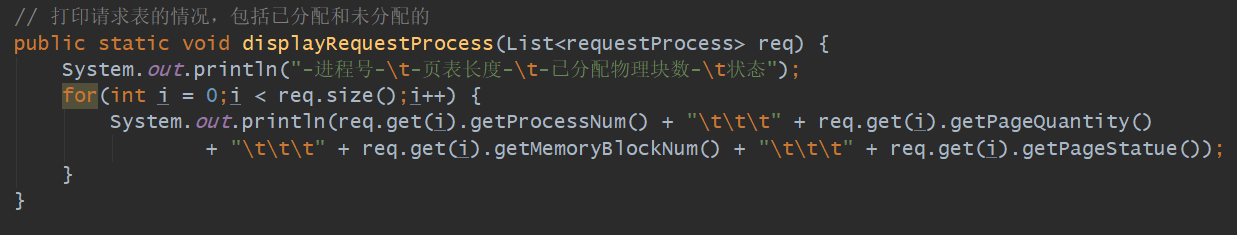
## 4.2核心源程序

### 4.2.1打印程序

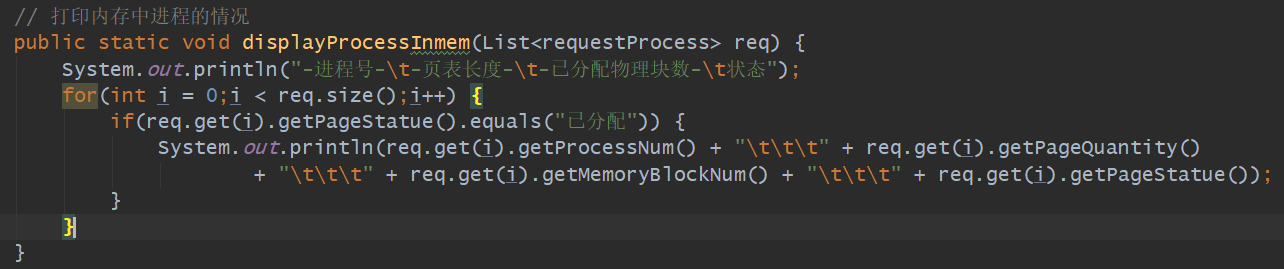
1、打印内存使用情况



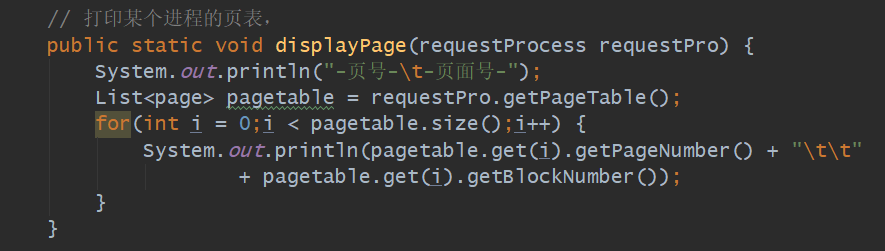
2、打印请求表



3、打印内存中进程的情况

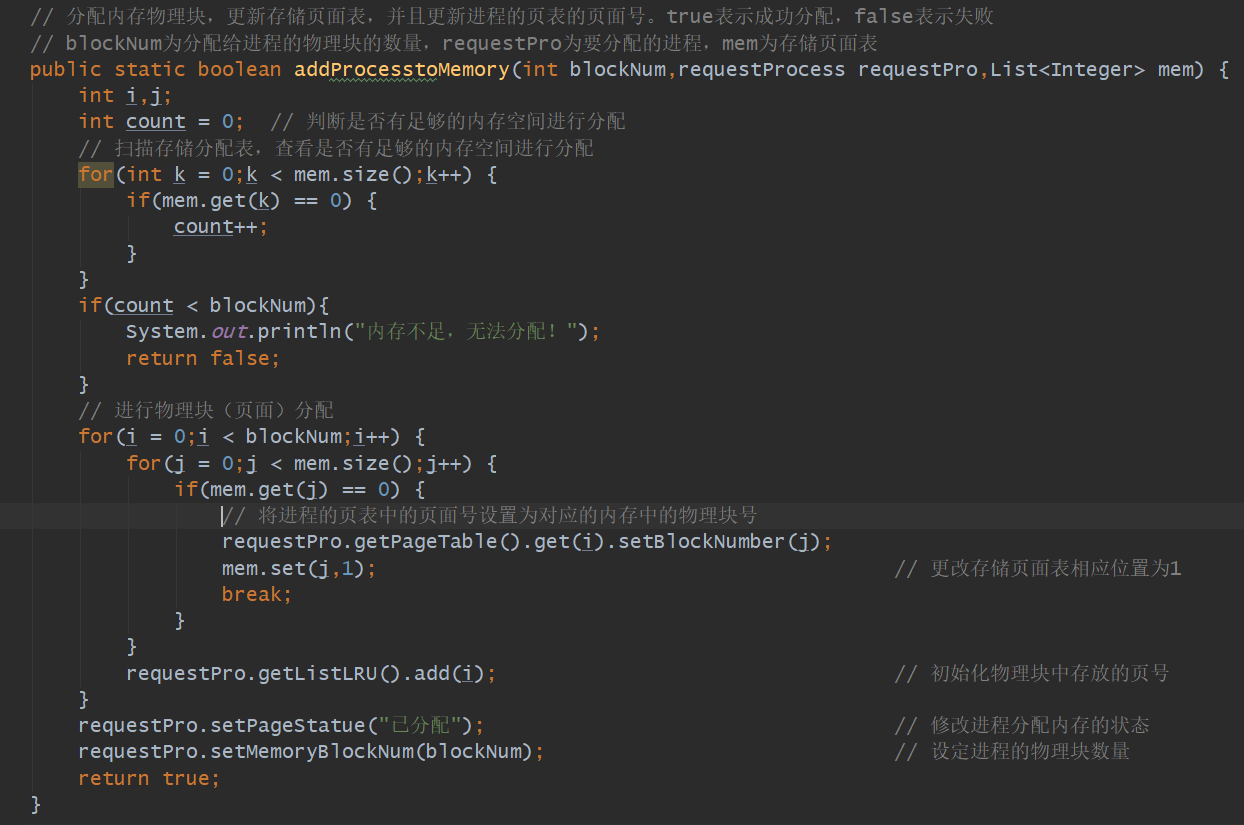


4.打印进程的页表



### 4.2.2分配程序

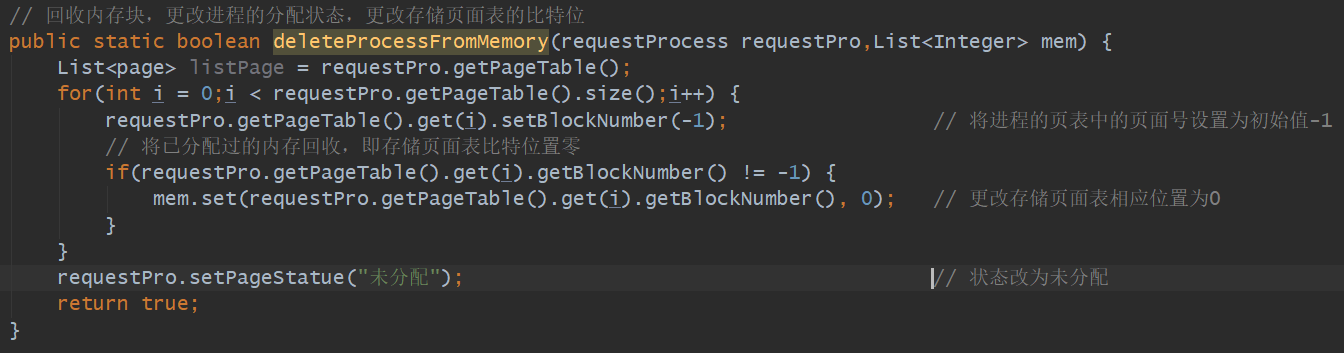
1、分配物理块



2、进程请求页

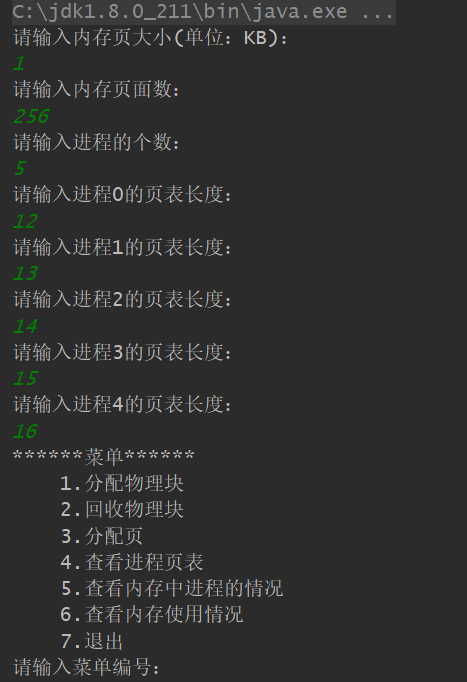


### 4.2.3回收程序



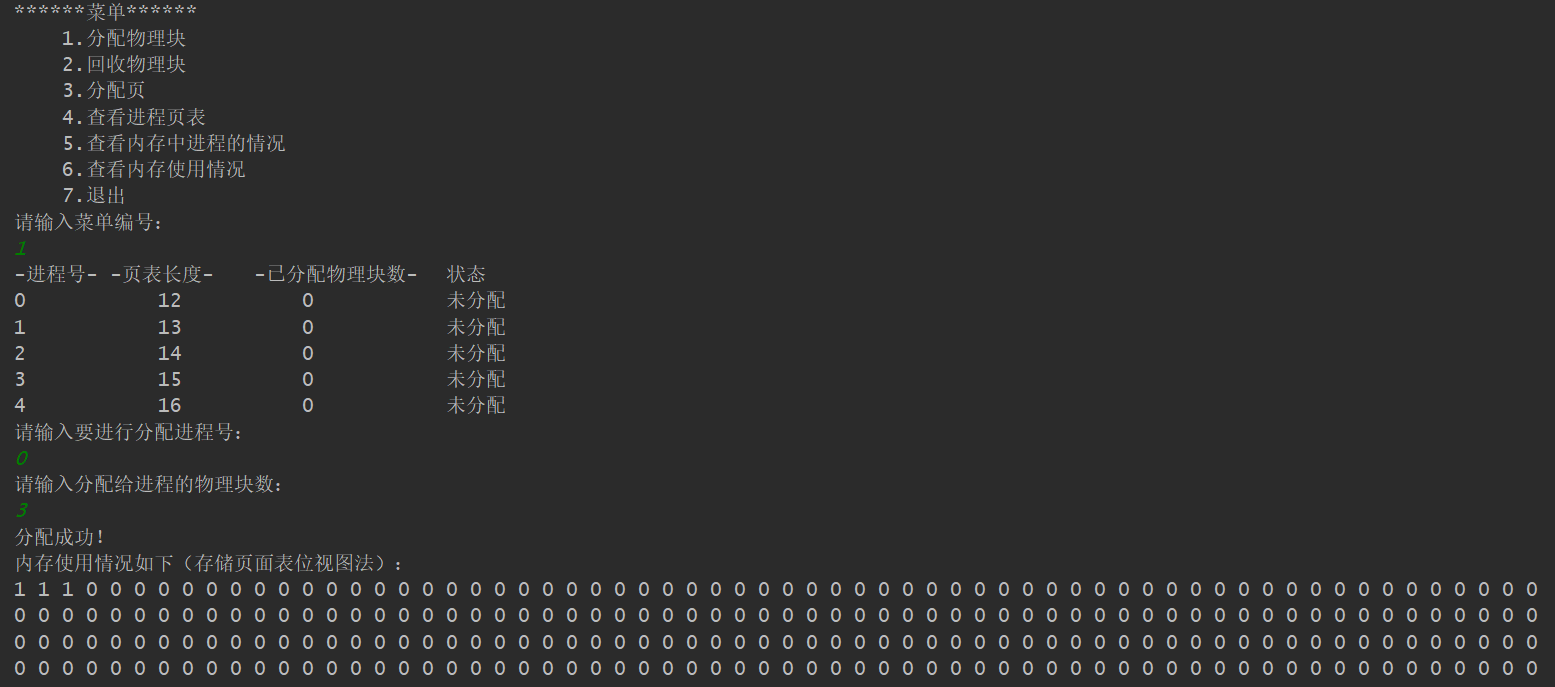
# 5测试用例、运行结果与运行情况分析

## 5.1界面测试及结果



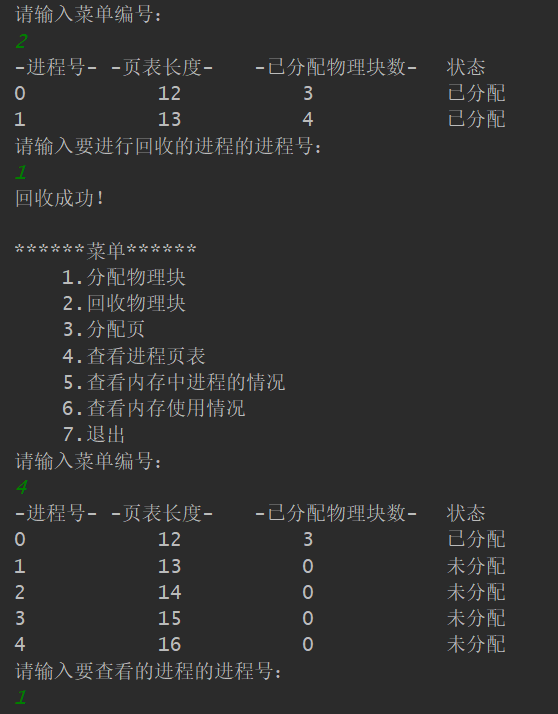
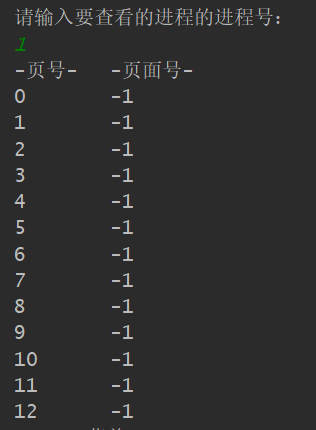
## 5.2分配物理块测试及结果分析

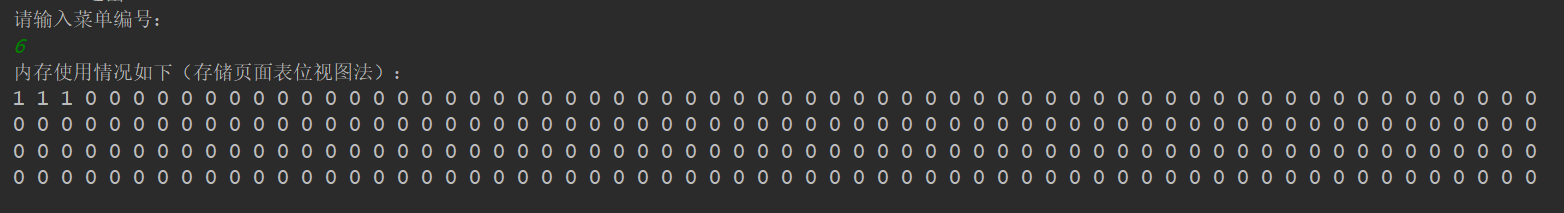
根据选择的进程，对其分配物理块，修改进程分配状态，并修改存储页面表。实验结果符合预期效果，如下图所示：



## 5.3回收物理块测试及结果分析

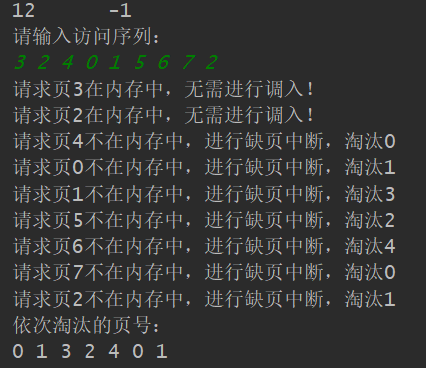
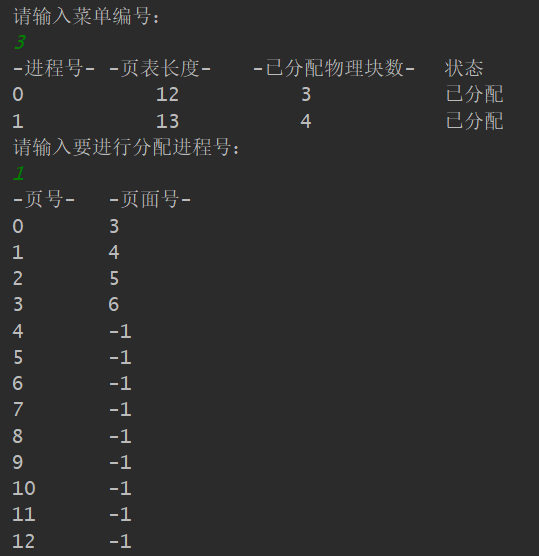
选择在内存中的进程进行回收，并修改进程的页表和存储页面表到初始状态，符合预期效果。结果如图所示：



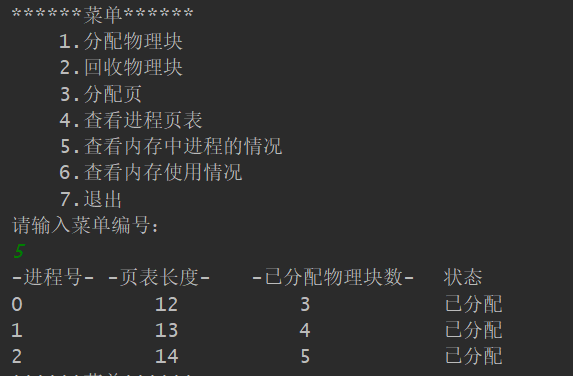
## 5.4分配页测试及结果分析

选择在内存中的进程进行分配页，如果访问的页在内存中，则直接访问；若不在，则要进行缺页中断，从外存中调入。修改进程的页表，符合预期效果。结果如图所示：



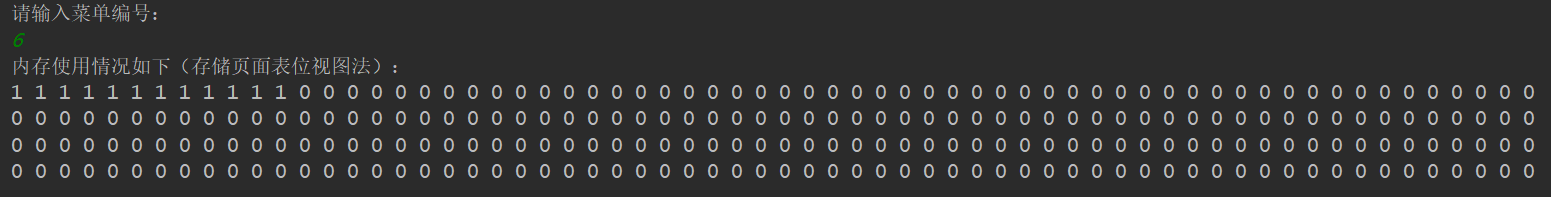
## 5.5查看内存中进程的情况测试及结果分析

查看已经分配物理块的进程在内存中的情况，符合预期效果。结果如图所示：



## 5.6查看内存的使用情况测试及结果分析

查看内存的使用情况，符合预期效果。结果如图所示：



# 6自我评价与总结

此次试验是完全在自己独立完成的，首先在分析问题并把问题转化为编程问题，我觉得个人把握的很好，对请求页式存储管理的基本原理理解的比较透彻；另外对于进程请求表使用动态操作，结构体保存项目的设计也独具匠心，输出请求表的时候比较清晰明了；同时，很好地实现了题目中的要求，能够清晰地表示出相应结果；将功能函数单独放在一个类中，注释清晰，使代码更加规范、可复用且易读性好；对于实现LRU算法的具体方法简单易懂且便于实现。

但是程序还有不足之处，程序仅仅只能从直观数值上来体请求现页式存储管理的操作，因为是人工进行输入模拟，所以有很强的主观性，不能真正体现请求页式存储管理随机分配的特性。同时，也无法像Linux之类操作系统一般真正的进行内存分配与回收，这是课后深入研究的一个方面。还有就是由于时间原因，没有采用图形化界面的形式使模拟结果更加直观的展示出来，在后续时间里，我会进一步完善代码，并加入图形化界面。

这次自己的收获还是不小，首先使我提高了分析问题，并根据需求转化成相应的程序结构的能力；其次也丰富了自己编写程序，调试程序的经验，这使得我编程时可能出现的错误的认识，并如何去避免产生了新的认识。在编写程序的过程中翻阅了大量相关书籍,保证在请求页式存储管理基本原理透彻了解情况下进行设计，事半功倍，让我认识到磨刀不误砍柴工的道理。总的来说这次试验比较成功，加深我对页式管理的理解，同时也提高了自己的编程的能力。编程是个长久的过程，平时要多去动手实践，去提高自己分析问题、发现问题、解决问题的能力。路漫漫其修远兮，吾将上下而求索。

我本次设计使用的时Java语言进行开发设计，但不局限于语言的形式，主要在于对请求页式存储管理算法以及缺页置换算法的理解。只要掌握了相应的算法，使用何种语言进行设计，都不是难题。正如此次实验，我同组成员便使用的 C++进行实现，并无不可。对于缺页中断也可选择不同的置换算法进行实现，重点还是在于原理和算法，万变不离其宗。

**本科生实践课程成绩评定表**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名** | **张超杰** | **班 级** | **计算机1701班** |
| **课程设计题目：模拟设计请求页式存储管理的分配与回收（LRU置换算法）** | | | |
| **课程设计答辩或质疑记录：** | | | |
| **成绩评定依据：** | | | |
| **最终评定成绩（以优、良、中、及格、不及格评定）** | | | |

**指导教师签字：**

**年**   **月 日**