**武汉大学计算机学院**

**本科生实验报告**

**主存空间的分配和回收**

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：操作系统设计

指 导 教 师 ：宋伟

学 生 学 号 ：

学 生 姓 名 ：

二○二一年七月

**郑 重 声 明**

本人呈交的实验报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本实验报告不包含他人享有著作权的内容。对本实验报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本实验报告的知识产权归属于培养单位。

本人签名： 日期： 2021.7.2

摘 要

主存空间的分配和回收实验的实验目的是帮助理解在不同的存储管理方式下应怎样进行存储空间的分配和回收。

实验设计主要遵循实验设计原理。

实验内容主要包括：主存储器空间的分配和回收。

实验结论为该实验让我重温了页表和位示图，加深了我对主存空间的分配和回收的理解，让我更清晰地了解每个作业在主存中的内存分配情况和页表的工作原理。

**关键词：**主存空间的分配和回收

**目 录**

**1** **主存储器空间的分配和回收**

1.1 实验内容及上机实验所用平台 ……………………………………………………… 1

1.2 数据结构及代码段分析 ……………………………………………………………… 1

1.3 调试过程 ………………………………………………………………………………… 3

1.4 实验总结 ………………………………………………………………………………… 5

**参考文献** …………………………………………………………………………………… 6

**1 主存空间的分配和回收**

**1.1 实验内容及上机实验所用平台**

通过本实验帮助理解在不同的存储管理方式下应怎样进行存储空间的分配和回收。

**1.1.1 实验内容**

在分页管理方式下采用位示图来表示主存分配情况，实现主存分配和回收。

**1.1.2 实验设计**

（1）假定系统的主存被分成大小相等的64个块，用0/1对应空闲/占用。

（2）当要装入一个作业时，根据作业对主存的需求量，先查空闲块数是否能满足作业要求，若能满足，则查位示图，修改位示图和空闲块数。位置与块号的对应关系为：块号=j\*8+i，其中i表示位，j表示字节。根据分配的块号建立页表。页表包括两项：页号和块号。

（3）回收时，修改位示图和空闲块数。

**1.1.3 上机实验所用平台**

开发软件采用VSCode，编程语言使用C++。

**1.2 数据结构及代码段分析**

**1.2.1 数据结构**

为PCB定义一个数据结构，其中包括：

·作业名

·作业位数

·页表

定义一个数组用来存放所有的作业信息。

定义一个位示图数组（8\*8）。

**1.2.2 代码段分析**

重点代码段如下：

int count = 0;

for(int i = 0; i < 8; i++){

bool flag = false;

for(int j = 0; j < 8; j++){

if(!bitMap[i][j]){

bits--;

bitMap[i][j] = 1;

P.pageTable.push\_back(make\_pair(count++, i \* 8 + j));

if(!bits){

flag = true;

break;

}

}

}

if(flag)

break;

}

这段代码实现了为作业分配页表，从位示图的第一项开始依次往后搜索，只要搜索到的块没有被占用，就占用它，然后将该块的块号写入作业的页表，直到为作业分配所需的块数后结束。

bool flag = true;

for(vector<PCB>::iterator it = PCBList.begin(); it != PCBList.end(); it++){

if(it->name == name){

for(int i = 0; i < it->pageTable.size(); i++)

bitMap[it->pageTable[i].second / 8][it->pageTable[i].second % 8] = 0;

empty += it->bits;

PCBList.erase(it);

flag = false;

break;

}

it++;

}

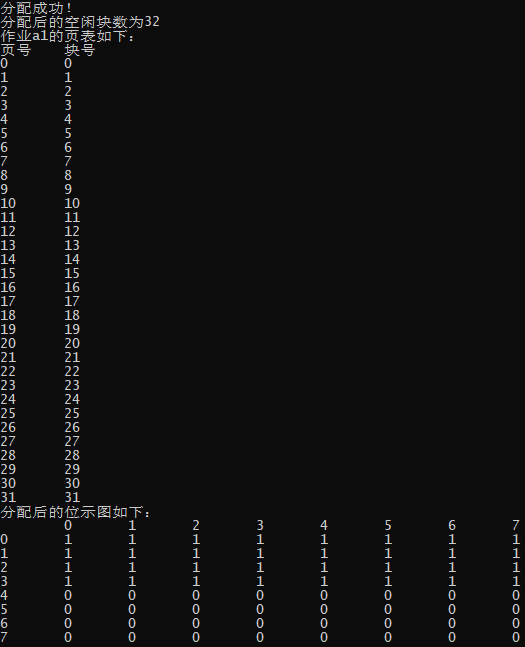
这段代码实现了释放作业占用的块，依次对作业的页表中的每一项进行块的释放。

**1.3 调试过程**

我们首先加入一个作业a1，位数为32；再加入一个作业a2，位数为10；再加入一个作业a3，位数为64，此时a3是无法加入的，程序会给出提示；然后释放作业a3，由于没有给a3分配物理块，所以a3无法在系统中找到，程序会给出提示；然后释放作业a1；再释放作业a2；结束程序。运行结果见以下图片。

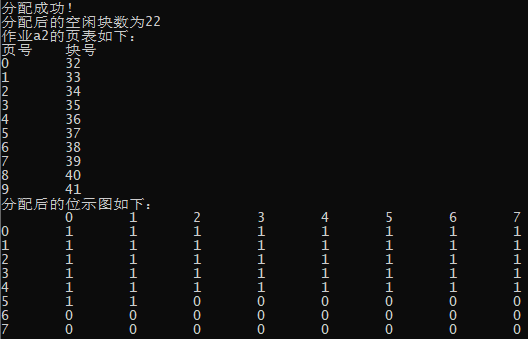


图**1.1** 加入作业a1



图**1.2** 加入作业a1的结果



图**1.3** 加入作业a2

图**1.4** 加入作业a2的结果



图**1.5** 加入作业a3



图**1.6** 加入作业a3的结果



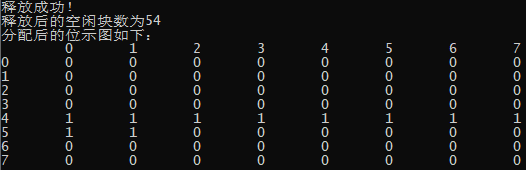
图**1.7** 释放作业a3



图**1.8** 释放作业a3的结果



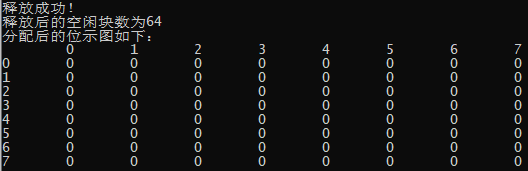
图**1.9** 释放作业a1



图**1.10** 释放作业a1的结果



图**1.11** 释放作业a2



图**1.12** 释放作业a2的结果

**1.4 实验总结**

该实验让我重温了页表和位示图，加深了我对主存空间的分配和回收的理解，让我更清晰地了解每个作业在主存中的内存分配情况和页表的工作原理。在编写代码的过程中我也遇到了一些问题，例如在分配物理块时忽略了位示图本身的性质；在使用iterator时忽略了一些可能出现的错误。但这些问题在经过我的仔细检查和更加谨慎的思考下都得到了有效的解决。总体来说我认为该实验原理并不是特别复杂，但能够加强我对主存空间的分配和回收的认知，让我有许多收获！

**参考文献**

[1] 郑鹏，曾平，金晶.计算机操作系统[M].武汉:武汉大学出版社，2014.7

教师评语评分

评语：

评分：

评阅人：

年 月 日

（备注：对该实验报告给予优点和不足的评价，并给出百分之评分。）