**武汉大学计算机学院**

**本科生实验报告**

**磁盘存储空间的分配和回收**

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： 2021.7.9

摘 要

磁盘存储空间的分配和回收实验的实验目的是使学生掌握磁盘存储空间的分配和回收算法。

实验设计主要遵循实验设计原理。

实验内容主要包括：模拟磁盘空闲空间的表示方法，以及模拟实现磁盘空间的分配和回收。

实验结论为该实验让我重温了UNIX系统的空闲块成组链接法，加深了我对磁盘存储空间的分配和回收的理解，让我更清晰地了解每个文件在磁盘中的存储空间分配情况和分配释放的方法。

**关键词：**磁盘存储空间的分配和回收

**目 录**

**1** **磁盘存储空间的分配和回收**

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 ……………………………………………………………… 2

1.3 调试过程 ………………………………………………………………………………… 5

1.4 实验总结 ………………………………………………………………………………… 8

**参考文献** …………………………………………………………………………………… 9

**1** **磁盘存储空间的分配和回收**

**1.1 实验内容及上机实验所用平台**

模拟磁盘空闲空间的表示方法，以及模拟实现磁盘空间的分配和回收。

**1.1.1 实验内容**

模拟UNIX系统的空闲块成组链接法，实现磁盘存储空间的管理。

**1.1.2 实验设计**

（1）假定磁盘存储空间已被划分成长度为n的等长块，共有M块可供使用。UNIX系统中采用空闲块成组链接的方法来管理磁盘存储空间，将磁盘中的每N个空闲块（N<M）分成一组，最后一组可以不足N块，每组的第一块中登记了下一组空闲块的块数和块号，第一组的块数和块号登记在专用块中，登记的格式如下：

|  |  |
| --- | --- |
| **0** | **空闲块数k** |
| **1** | **空闲块号1** |
| **2** | **空闲块号2** |
| **…** | **…** |
| **K** | **空闲块号k** |
|  |  |

图**1.1** 每组的登记格式

当第一项内容为“0”时，则第二项起指出的空闲块是最后一组。

（2）可用二维数组A [0…M-1] [0…n-1]来模拟管理磁盘空间，用A[i]表示第I块，第0块A[0]作为专用块。

（3）成组链接的分组情况记录在磁盘物理块中，为了查找链接情况，必须把它们读入主存，故当磁盘初始化后，系统先将专用块内容复制到主存中。定义一个数组MA存放专用块内容，即MA: =A[0]。申请一块磁盘空间时，查MA，从中找出空闲块号，当一组的空闲块只剩第一块时，则应把该块中指出的下一组的空闲块数和块号复制到专用块中，然后把该块分配给申请者。当一组的空闲块分配完后则把专用块内容（下一组链接情况）复制到主存，再为申请者分配。

（4）归还一块时给出归还的块号，若当前组不满规定块数时，将归还块登记入该组；若当前组已满，则另建一新组，这时归还块作为新一组的第一块，应把主存中登记的一组链接情况MA复制到归还块中，然后在MA重新登记一个新组。

（5）设计分配和归还磁盘空间的程序。

**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**

定义一个存储成组链接情况的二维数组，在代码中的定义如下：

int a[9][4] = {

{3, 1, 2, 3},

{3, 4, 5, 6},

{0, 0, 0, 0},

{0, 0, 0, 0},

{2, 0, 7, 8},

{0, 0, 0, 0},

{0, 0, 0, 0},

{0, 0, 0, 0},

{0, 0, 0, 0}

};

数组的第一维表示物理块号，其中a[0]实际上就是专用块的信息。在a[i]中，第一个数字表示该组中的空闲块数，后面三个数字若为非0，则表示空闲块号，若为0，则表示无空闲块。其中第二个数字若为非0，则表示了下一组的物理块号，若为0，表示当前组为最后一组。例如，由于a[1]为{3, 4, 5, 6},所以在该组中有3个空闲块，空闲块号分别为4、5、6，并且下一组的信息存储在空闲块4中，即a[4]。

定义一个数组ma，表示专用块，ma与a[0]实质上是同步的，所以这里ma为{3, 1, 2, 3}。

定义一个数组full，full[i]为1表示第i个物理块被占用，为0表示第i个物理块为空闲块。

**1.2.2 代码段分析**

重点代码段如下：

if(ma[0] > 1){

int id1 = 3;

while (!ma[id1])

id1--;

int id2 = ma[id1];

cout << "已分配物理块" << id2 << endl;

ma[id1] = 0;

a[now][id1] = 0;

ma[0]--;

a[now][0]--;

full[id2] = 1;

show();

}

在申请物理块时，如果第一组的物理块数大于1，则可直接选择其中一个物理块进行分配。

else if(ma[0] == 1){

int id1 = 3;

while (!ma[id1])

id1--;

int id2 = ma[id1];

cout << "已分配物理块" << id2 << endl;

a[now][id1] = 0;

a[now][0]--;

full[id2] = 1;

if(!ma[1])

for(int i = 0; i < 4; i++)

ma[i] = 0;

else{

now = ma[1];

memcpy(ma, a[ma[1]], sizeof(a[ma[1]]));

}

show();

}

在申请物理块时，如果第一组的物理块数为1，则将该物理块进行分配，并将下一组（如果存在）的信息存入专用块。

if(!ma[0]){

ma[0] = 1;

ma[1] = id;

a[0][0] = 1;

a[0][3] = id;

now = 0;

}

在释放物理块时，如果第一组的物理块数为0（即无空闲块），则将回收的块放入第一组，将第一组信息存入专用块。

else if(ma[0] < 3){

int id1 = 3;

while(ma[id1])

id1--;

ma[id1] = id;

ma[0]++;

a[now][id1] = id;

a[now][0]++;

}

在释放物理块时，如果第一组的物理块数不为0且没有到达一组的最大容量，则将回收的块放入该组即可。

else{

for(int i = 0; i < 4; i++)

a[id][i] = ma[i];

ma[0] = 1;

ma[1] = id;

ma[2] = 0;

ma[3] = 0;

a[0][0] = 1;

a[0][1] = id;

a[0][2] = 0;

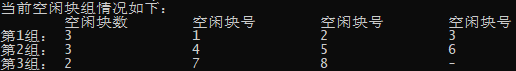
a[0][3] = 0;

}

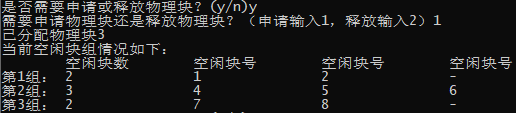
在释放物理块时，如果第一组的物理块数达到了一组的最大容量，则将回收的块单独放入一组，将专用块的信息存入该物理块，再将新建的组的信息存入专用块。

**1.3 调试过程**

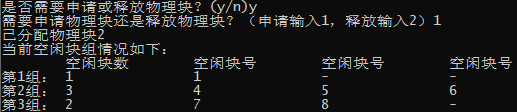
在代码中的初始状态有8个空闲块。我们依次申请8个物理块，然后按物理块号为6、4、8、1、3、2、5、7的顺序释放物理块。



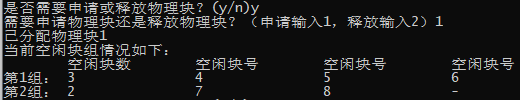
图**1.2** 初始状态



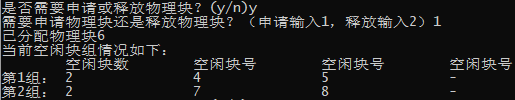
图**1.3** 第一次申请物理块

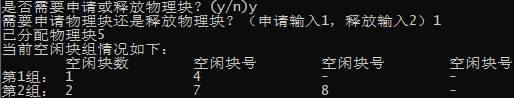


图**1.4** 第二次申请物理块

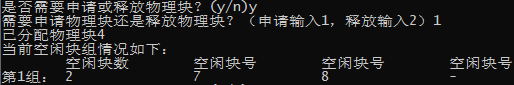


图**1.5** 第三次申请物理块

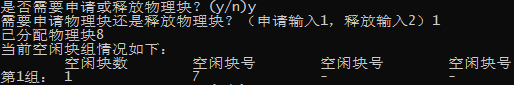


图**1.6** 第四次申请物理块

图**1.7** 第五次申请物理块



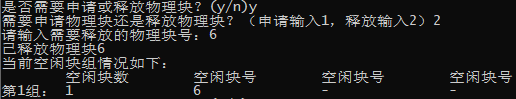
图**1.8** 第六次申请物理块



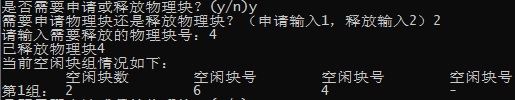
图**1.9** 第七次申请物理块



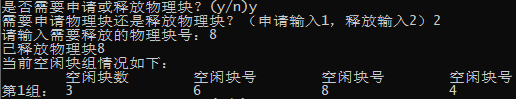
图**1.10** 第八次申请物理块



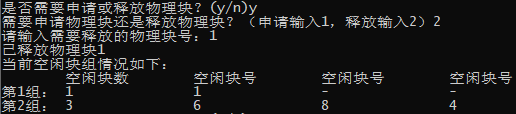
图**1.11** 释放物理块6



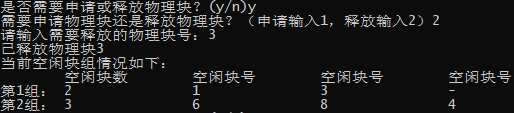
图**1.12** 释放物理块4



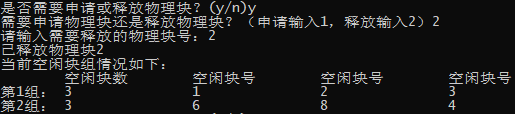
图**1.13** 释放物理块8



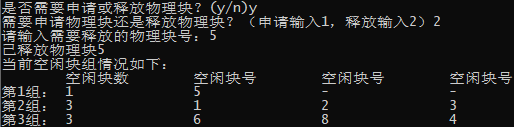
图**1.14** 释放物理块1



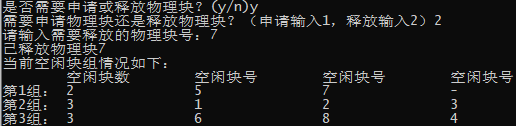
图**1.15** 释放物理块3



图**1.16** 释放物理块2



图**1.17** 释放物理块5



图**1.18** 释放物理块7

**1.4 实验总结**

该实验让我重温了UNIX系统的空闲块成组链接法，加深了我对磁盘存储空间的分配和回收的理解，让我更清晰地了解每个文件在磁盘中的存储空间分配情况和分配释放的方法。在编写代码的过程中我也遇到了一些问题，例如在分配和释放物理块时忽略了专用块信息的变化。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稍显复杂，但能够加强我对UNIX系统的空闲块成组链接法的认知，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。）