

武汉大学计算机学院

课程实验(设计)报告

专业(班)：____ 计算机科学与技术 三班 _____

学 号：____ 2013301500100 _____

姓 名：____ 秦贤康 _____

课程名称：____ 操作系统设计 _____

任课教师：____ 宋伟 _____

2015 年 11 月 26 日

说 明

1、本课程设计报告包括：实习题目、实习内容及设计思想（设计思路、主要数据结构、主要代码结构及代码段分析）、上机实习所用平台及相关软件、调试过程（测试数据设计、测试结果分析）、总结（实习中遇到的问题及解决方法、实习中产生的错误及原因分析、实习体会及收获）。

2、课程设计报告用 A4 纸打印。

标题 4 号宋体加粗，正文 5 号宋体，图表文字标注、程序等用 6 号。

页边距：左边 3.17cm，右边 3.17 cm，上 2.54 cm，下 2.54 cm。

行间距：1.5 倍行距，一页 30 行，每行 39 字。

页脚：1.5 cm，页脚(页码，居中)。

1	实习内容.....	4
2	实习题目及设计思想.....	4
	2.1 实习题目.....	4
	2.2 设计思路.....	4
	2.3 主要数据结构.....	4
	2.4 主要代码结构及代码段分析.....	4
4	调试过程.....	6
	4.1 测试数据设计.....	6
	4.2 测试结果分析.....	6
5	总结.....	9

磁盘存储空间的分配和回收

1 实习内容

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。

2 实习题目及设计思想

2.1 实习题目

第二题：用位示图管理磁盘存储空间

[提示]：

(1) 为了提高磁盘存储空间的利用率，可在磁盘上组织成链接文件、索引文件，这类文件可以把逻辑记录存放在不连续的存储空间。为了表示哪些磁盘空间已被占用，哪些磁盘空间是空闲的，可用位示图来指出。位示图由若干字节构成，每一位与磁盘上的一块对应，“1”状态表示相应块已占用，“0”状态表示该块为空闲。位示图的形式与实习二中的位示图一样，但要注意，对于主存储空间和磁盘存储空间应该用不同的位示图来管理，绝不可混用。

(2) 申请一块磁盘空间时，由分配程序查位示图，找出一个为“0”的位，计算出这一位对应块的磁盘物理地址，且把该位置成占用状态“1”。假设现在有一个盘组共8个柱面，每个柱面有2个磁道（盘面），每个磁道分成4个物理记录。那么，当在位示图中找到某一字节的某一位为“0”时，这个空闲块对应的磁盘物理地址为：

柱面号=字节号

磁道号= 位数 / 4

物理记录号= 位数 % 4

(3) 归还一块磁盘空间时，由回收程序根据归还的磁盘物理地址计算出归还块在位示图中的对应位，把该位置成“0”。按照（2）中假设的盘组，归还块在位示图中的位置计算如下：

字节号=柱面号

位数=磁道号(4+物理记录号

(4) 设计申请磁盘空间和归还磁盘空间的程序。

要求能接受来自键盘的空间申请及释放请求，要求能显示或打印程序运行前和运行后的位示图；分配时把分配到的磁盘空间的物理地址显示或打印出来，归还时把归还块对应于位示图的字节号和位数显示或打印出来。

2.2 设计思路

- 数组保存位示图
- 这题直接用 C 语言写更简单

2.3 主要数据结构

```
const int SIZE = 8;
int bitmap[SIZE][SIZE];           //位示图
```

2.4 主要代码结构及代码段分析

回收：

```
void call_back()
{
    int cylinder,track,sector;
    printf("柱面号:");
    scanf("%d",&cylinder);
    printf("磁道号:");
    scanf("%d",&track);
    printf("物理记录号:");
    scanf("%d",&sector);
    if(bitmap[cylinder][4*track+sector]==0)
    { printf("此块为未分配块！回收出错！\n");    }
    else {
        bitmap[cylinder][4*track+sector]=0;
        printf("回收块对应的字节号:%4d\t 位数:%4d\n",cylinder,4*track+sector);
    }
}
```

代入公式得到目标块，回收时要判断块是否已经被分配（或者越界）
申请

```
void call()
{
    int i,j;
    int flag=0;
    int cylinder,track,sector;
    for(i=0;i<8&& !flag;i++)
    {
        for(j=0;j<8&&!flag;j++)
        if(bitmap[i][j]==0)
        {      bitmap[i][j]=flag=1;}
    }
    if(flag==1)
    {
        cylinder=i;
        track=j/4;
        sector=j%4;
        printf("分配到的柱面号、磁道号、物理记录数");
        printf("%d\t%d\t%d\n",cylinder,track,sector);
    }
    else printf("空间不足，分配失败!\n");
}
```

把相应位置 1 即可

3 上机实习所用平台及相关软件

OS: mint Mint 17.2 rafaela
Kernel: x86_64 Linux 3.16.0-38-generic
CPU: AMD A8-4500M APU with Radeon HD Graphics @ 1.9GHz
GPU: Gallium 0.4 on AMD ARUBA
RAM: 1522MiB / 3337MiB
Vim:用 doxygen 自动生成注释
g++ 4.8.4

4 调试过程

Test.sh

```
#!/bin/sh
```

```
# 测试三条数据
```

```
cat test1|./mem >test1.out
```

运行 **test.sh** 自动进行输入输出，只要事先把测试数据写到 **test** 文件中就可以了

4.1 测试数据设计

Test1

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
2 0 0 0
```

```
2 0 0 2
```

说明：

1 申请空间

2 回收，后接柱面号，磁道号，物理记录号

4.2 测试结果分析

请输入：

分配到的柱面号、磁道号、物理记录数 2 1 2

```
1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 0 0
```

```
0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0
```

1:申请

2:回收

0:退出

请输入：

柱面号:磁道号:物理记录号:回收块对应的字节号: 0 位数: 0

```
0 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 0 0
```

```
0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

1:申请

2:回收

0:退出

请输入:

柱面号:磁道号:物理记录号:回收块对应的字节号: 0 位数: 2

```
0 1 0 1 1 1 1 1
1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

由结果可以看出:对应 test1 的输入数据,分配与回收功能都已经实现
另一测试分配失败的实时运行截图结果如下:



图 4-1

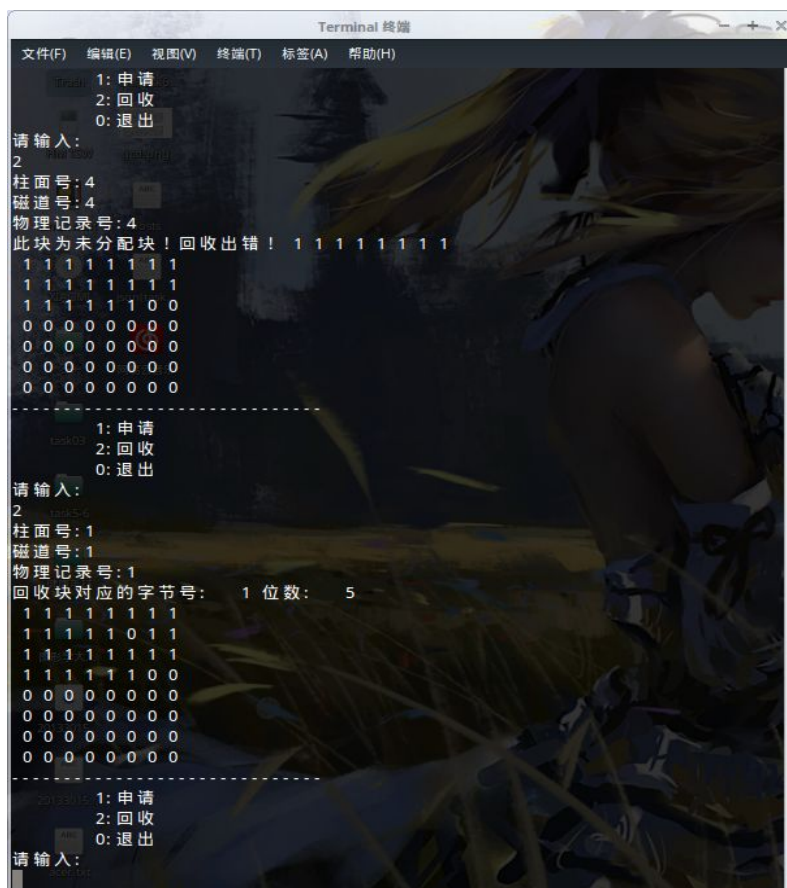


图 4-2



图 4-3

由上结果可以看出，此次实验顺利模拟了用位示图实现磁盘空间管理

5 总结

这个题目我看了很久，一开始没看懂，后来翻翻教材和查资料才弄懂。发现原理和分页那个挺相似的，磁盘上的所有盘块都有一个二进制位与之对应，只要带入公式算就可以了。

