操作系统课程设计

任　务　书

一、 目的、要求

1. 课设目的

本次实验的时间为一周，目的是**使学生进一步加深对操作系统主要管理模块的理解和掌握，并使用高级程序设计语言进行操作系统中的典型算法的模拟实现**。通过本次课程设计对操作系统的相关重要概念进一步的理解，提高学生的实践动手能力。

2. 内容要求

(1) 每个同学完成一个设计题目，具体的题目和要求见附件一。

(2) 根据设计任务，用可视化编程工具编制程序，在机器上调试运行，并通过上机考核。

(3) 要求界面设计美观，功能完整，使用方便。

(4) 按照“课程设计报告规范”的要求，写出课程设计报告。

二、 计划进度

序号	设计(实验)内容	完成时间	备注
1	《操作系统课程设计》动员大会	课程设计开始前 1 天	
2	总体设计	第 1 工作日	
3	编码与调试	第 2、3 工作日	
4	撰写设计报告	第 4 工作日	
5	验收检查	第 5 工作日	

三、 设计成果要求

1．设计的软件代码应能实现题目要求的全部功能。

2．实验报告要按规范的格式撰写。具体要求见附件二。

3．验收时当面演示程序功能，并回答老师提问。

四、 考核方式

1．平时考核：考勤、学习态度、设计进度等。

2．检查验收：运行结果、讲解、口试等。

3．报告评定：内容与格式。

最后成绩=平时考核（30%）+检查验收(35%)+报告评定(35%)

指导教师：贾静平

2023 年 6 月 16 日

附件一　　操作系统课程设计题目

一、 设计要求

1．用可视化编程工具编制程序，在机器上调试运行，并通过上机考核。

2．在 3 个题目中完成其中的 1 个。其中题目 1 的难度较高，为一类题，最高成绩为优，题目 2、3 为二类题，最高成绩为良。

3．要求界面设计美观，功能完整，使用方便，能运行通过。

二、 设计题目

题目 1　SPOOLING 假脱机输入输出技术模拟

1．设计一个实现 SP001ING 技术的进程

要求设计一个 SP00LING 输出进程和两个请求输出的用户进程，以及一个 SP00LING 输出服务程序。当请求输出的用户进程希望输出一系列信息时，调用输出服务程序，由输出服务程序将该信息送入输出井。待遇到一个输出结束标志时，表示进程该次的输出文件输出结束。之后，申请一个输出请求块(用来记录请求输出的用户进程的名字、信息在输出井中的位置、要输出信息的长度等)，等待 SP00LING 进程进行输出。

SP00LING 输出进程工作时，根据请求块记录的各进程要输出的信息，将其实际输出到打印机或显示器。这里，SP00LING 输出进程与请求输出的用户进程可并发运行。

2．设计进程调度算法

进程调度采用随机算法，这与进程输出信息的随机性相一致。两个请求输出的用户进程的调度概率各为 45%，SP00LING 输出进程为 10%，这由随机数发生器产生的随机数来模拟决定。

3．进程状态

进程基本状态有 3 种，分别为可执行、等待和结束。可执行态就是进程正在运行或等待调度的状态；等待状态又分为等待状态 1、等待状态 2 和等待状态 3。

状态变化的条件为：

①进程执行完成时，置为“结束”态。

②服务程序在将输出信息送输出井时，如发现输出井已满，将调用进程置为“等待状态 1”。

③SP00LING 进程在进行输出时，若输出井空，则进入“等待状态 2”。

④SP00LING 进程输出一个信息块后，应立即释放该信息块所占的输出井空间，并将正在等待输出的进程置为“可执行状态”。

⑤服务程序在输出信息到输出井并形成输出请求信息块后，若 SP00LING 进程处于等待态，则将其置为“可执行状态”。

⑥当用户进程申请请求输出块时，若没有可用请求块时，调用进程进人“等待状态 3”。

4．数据结构

①进程控制块 PCB

```
struct    pcb
      {int id;                    /*进程标识数*/
      Int status;                 /*进程状态*/
      Int count:                  /*要输出的文件数*/
      Int x:                      /*进程输出时的临时变量*/
      }PCB[3];
       status＝
```

其中，

0 为可执行态；

1 为等待状态 1，表示输出井满，请求输出的用户进程等待；

2 为等待状态 2，表示请求输出井空，SP00LING 输出进程等待；

3 为等待状态 3，表示请求输出井满，请求输出的用户进程等待；

4 为结束态，进程执行完成。

②请求输出块 reqblock

```
struct{
       int reqname;               /*请求进程名*/
       int length;                /*本次输出信息长度*/
       int addr;                  /*信息在输出井的首地址*/
     } reqblock：[10];
```

③输出井 BUFFER

SP00LING 系统为每个请求输出的进程在输出井中分别开辟一个区。本实验可设计一个二维数组(int buffer[2][100])作为输出井。每个进程在输出井最多可占用 100 个位置。

5．编程说明

为两个请求输出的用户进程设计两个输出井。每个井可存放 100 个信息，即 buffer[2][100]。为此，设计两个计数器，使用数组 C1[2]，分别表示两个用户进程可使用的输出井的空间。其初值 c1[0]，c1[1]都为 100。用 C2[2][2]二维指针数组表示输出井使用情况。C2[0][0]代表 buffer[1]的第一个可用空缓冲指针，C2[0][1]代表 buffer[0]的第一个满缓冲指针；C2[1][0]代表 buffer[1]的第一个可用空缓冲指针，C2[1][1]代表 buffer [1]的第一个满缓冲指针。

每个用户进程请求输出文件的个数由用户从键盘输入而定。当用户进程将其所有文件输出完时，终止运行。

为简单起见，用户进程简单地设计成：每运行一次，随机输出数字 0～9 之间的一个数，且用 0 作为文件结束标志。当输出值为零时，就形成一个请求信息块，填入请求输出信息块 reqblock 结构中。这个输出请求块结构也有一个计数器 C3，表示当前系统剩余的请求输出信息块个数，初值为 10。

另外，再设两个指针 Pt1 和 Ptr2 表示请求输出块使用情况。Ptr1 是要输出的第一个请求输出块指针，初值为 0；Ptr2 是空闲请求输出块指针，初值也为 0。两个指针按模 10 进行变化，即把请求输出块结构数组看成是一个环型数组。根据 Ptr1 和 Ptr2 的变化，确定请求输出块的使用情况。

主程序中包括(或调用)调度程序。调度程序中包括一个随机数函数，以该函数值为依据，按照如图 1 所示框图调度 3 个进程。完成对各数据结构的初始化。

6．程序框图

（1）SP00LING 输出模拟系统主控流程图如图 1 所示。

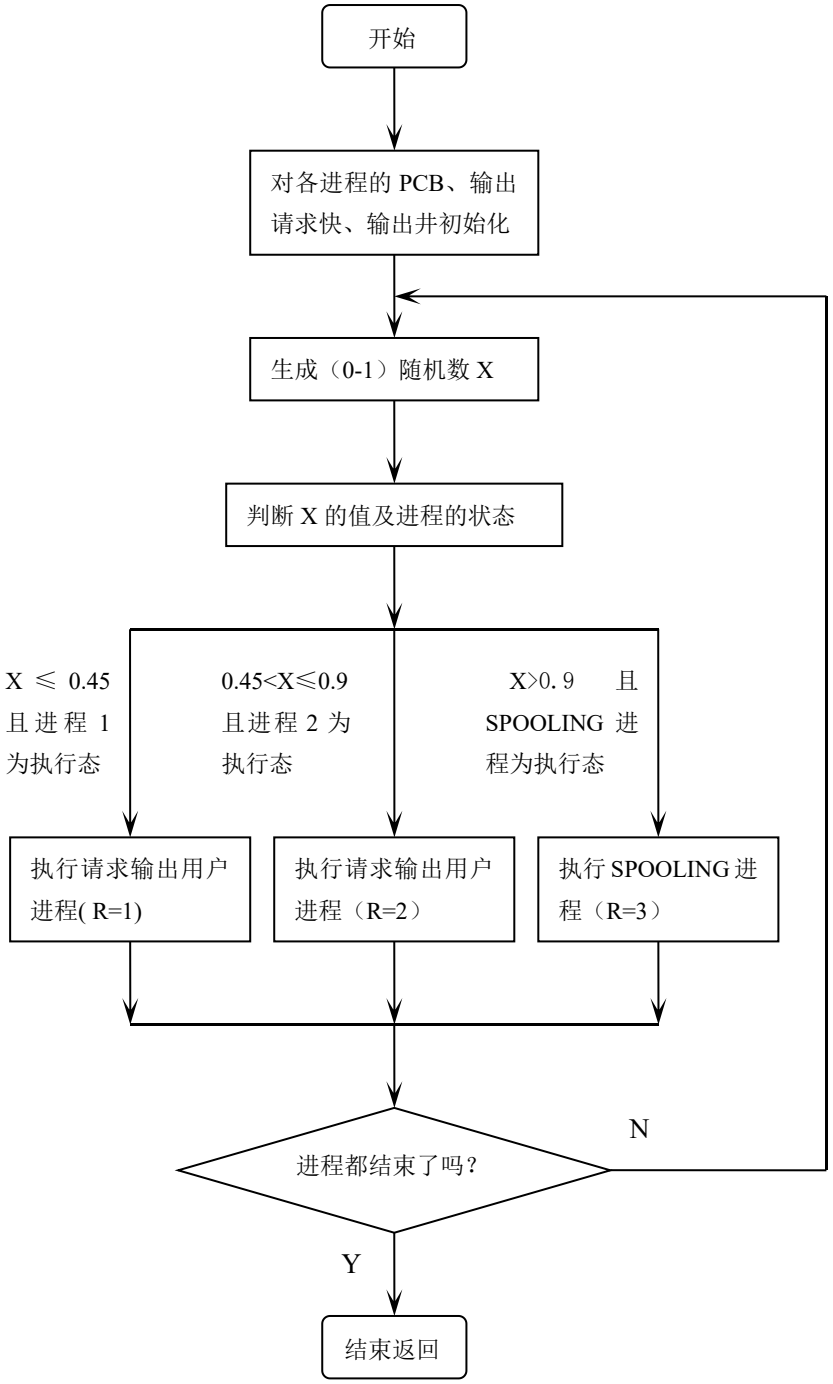

图 1　SPOOLING 输出模拟系统主控流程图

（2）SP00LING 输出服务程序由请求输出的两个用户进程调用，程序流程图如图 2 所示。

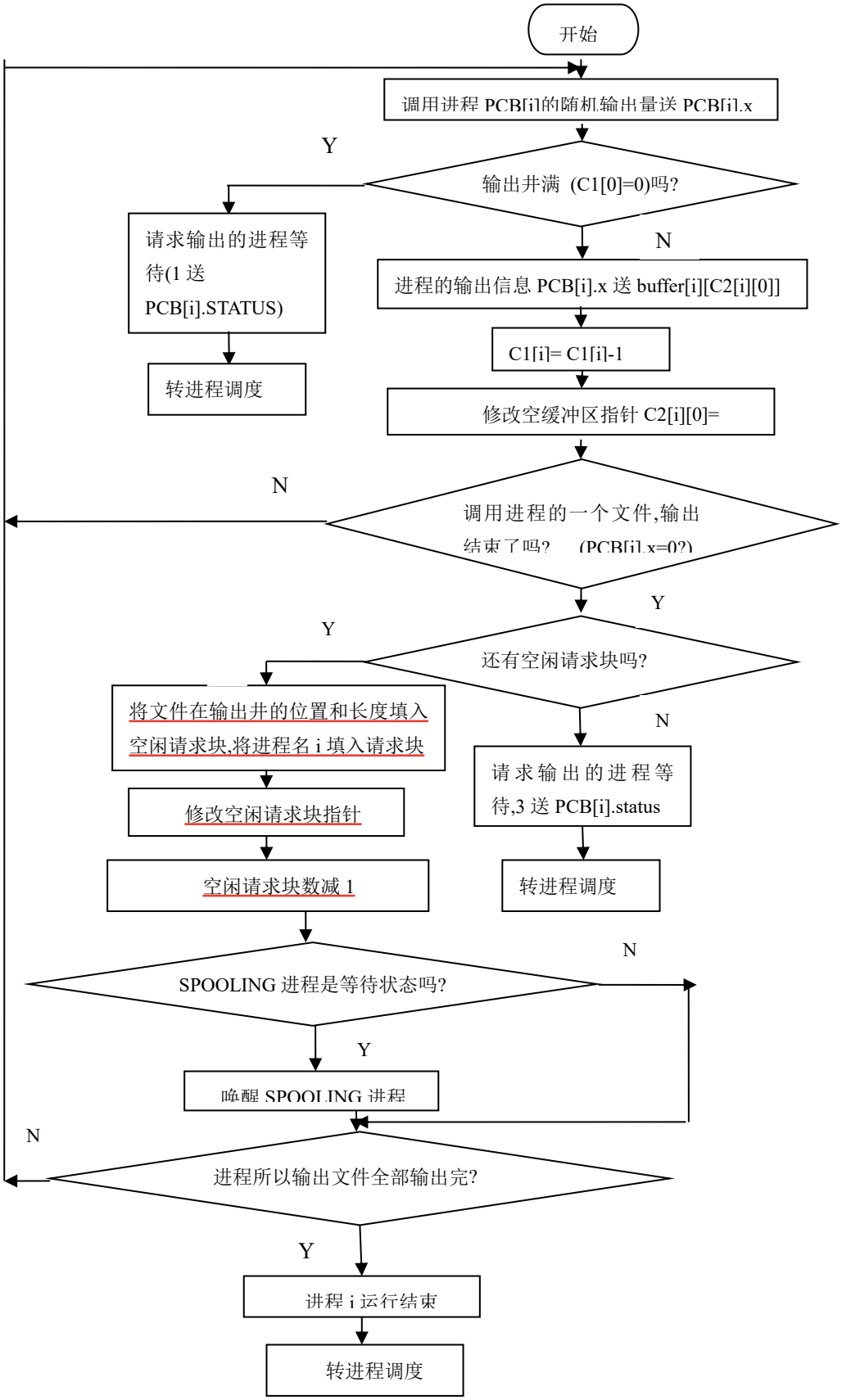

图 2　输出请求服务的程序框图

（3）SPOOLING 输出进程流程图如图 3 所示。

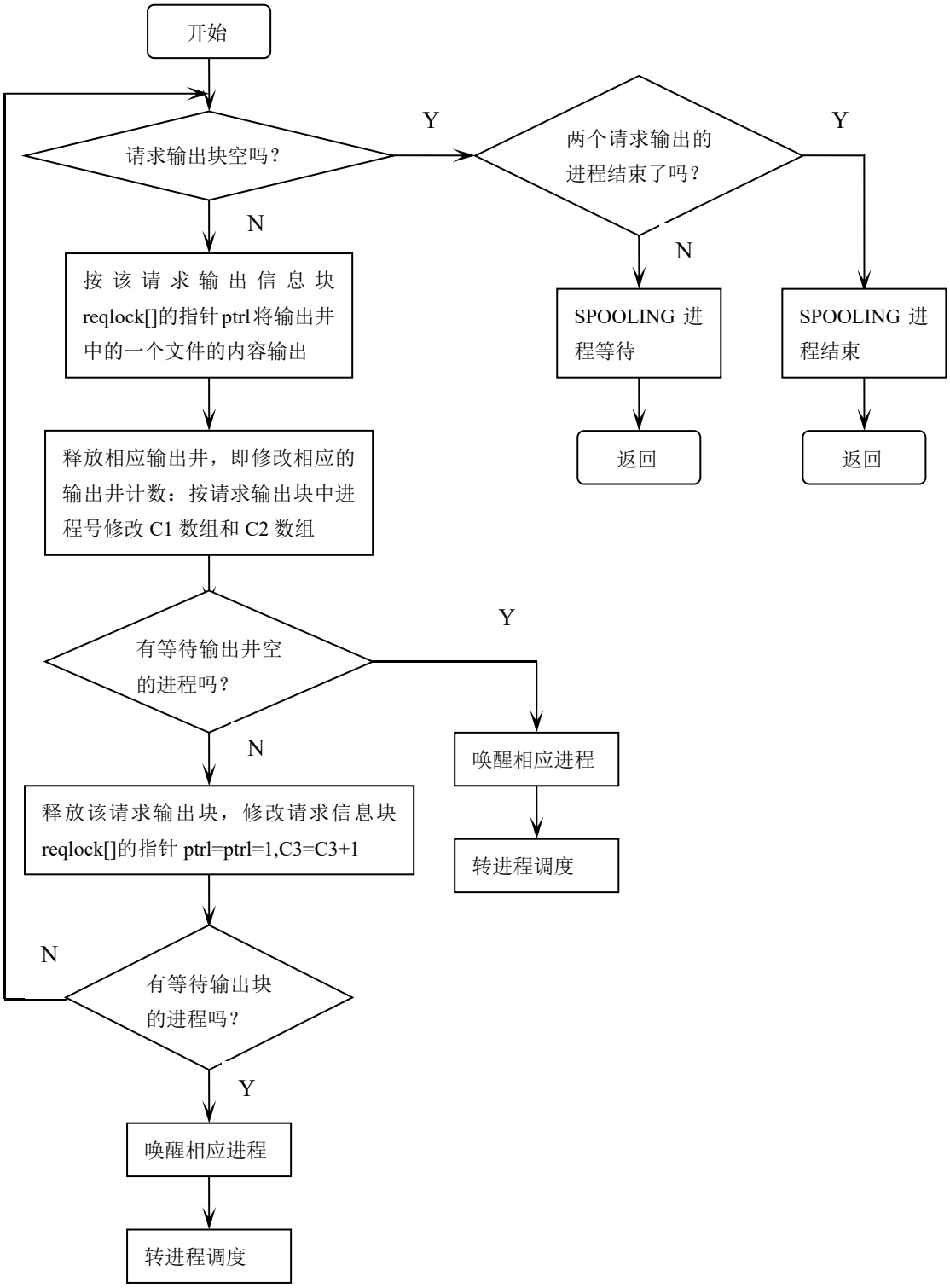

图 3 SPOOLING 输出进程流程图

题目 2　用位示图管理磁盘空间的分配与回收

要求打印或显示程序运行前和运行后的位示图，以及分配和回收磁盘的物理地址过程。

提示

（1）假定现有一个磁盘组，共 40 个柱面。每个柱面 4 个磁道，每个磁道又划分成 4 个物理记录。磁盘的空间使用情况用位示图表示。位示图用若干个字构成，每一位对应一个磁盘块。1 表示占用，0 表示空闲。为了简单，假定字长为 16 位，其位示图如图 9—1 所示。系统设一个变量 S，记录磁盘的空闲块个数。

位	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
字 0	1	1	1	1	1	1	0	1	0	0	1	1	1	1	1	0
1																
2																
：																
39																

图 9—1　位示图

（2）申请一个磁盘块时，由磁盘块分配程序查位示图，找出一个为 0 的位，并计算磁盘的物理地址(即求出柱面号、磁道号(也即磁头号)和扇区号)。

由位示图计算磁盘的相对块号的公式如下：

相对块号一字号×16+位号

之后再将相对块号转换成磁盘的物理地址：

由于一个柱面包含的扇区数＝每柱面的磁道数×每磁道的扇区数＝4×4＝16，故柱面号＝相对块号／16 的商，即柱面号＝字号

磁道号＝(相对块号／16 的余数)／4 的商，即(位号／4)的商

物理块号＝(相对块号／16 的余数)／4 的余数，即(位号／4)的余数

（3）当释放一个相对物理块时，运行回收程序，计算该块在位示图中的位置，再把相应位置 0。计算公式如下：

先由磁盘地址计算相对块号：

相对块号＝柱面号×16+磁道号×4+物理块号

再计算字号和位号：

字号＝相对块号／16 的商，也即字号＝柱面号

位号＝磁道号×物理块数／每磁道+物理块号

（4）按照用户要求，申请分配一系列磁盘块，运行分配程序，完成分配。然后将分配的相对块号返回用户，并将相对块号转换成磁盘绝对地址，再显示系统各表和用户已分配的情况。

（5）磁盘空间分配框图如图 4 所示。

（6）设计一个回收算法，将上述已分配给用户的各盘块释放。并显示系统各表。回收算法框图如图 5 所示。

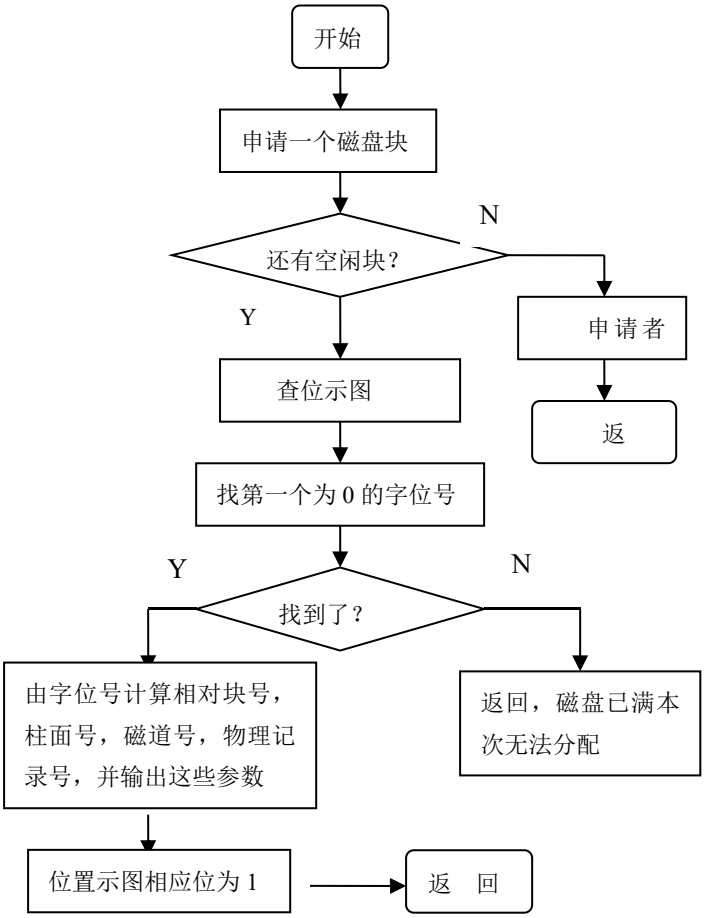

图 4　磁盘空间分配框图

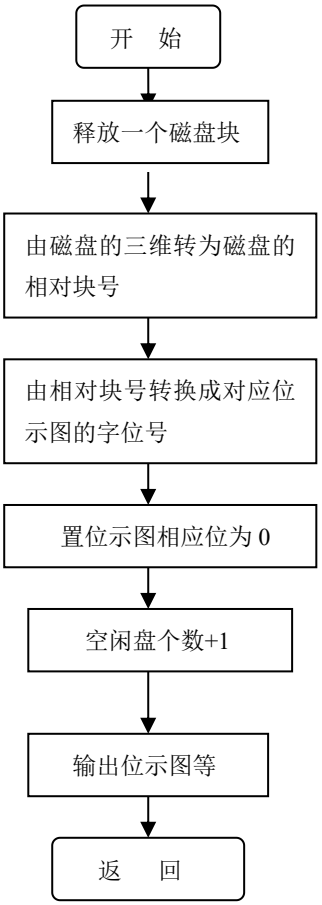

图 5　磁盘空间回收框图

题目 3　模拟 UNIX 系统的空闲块成组链接法实现磁盘空间管理

提示

（1）假定磁盘存储空间中，空闲块以 N 个盘块为一组，已划分成 M 组。假设 N＝3，M＝3，磁盘共有 7 块，则空闲块成组链接如图 6 所示。

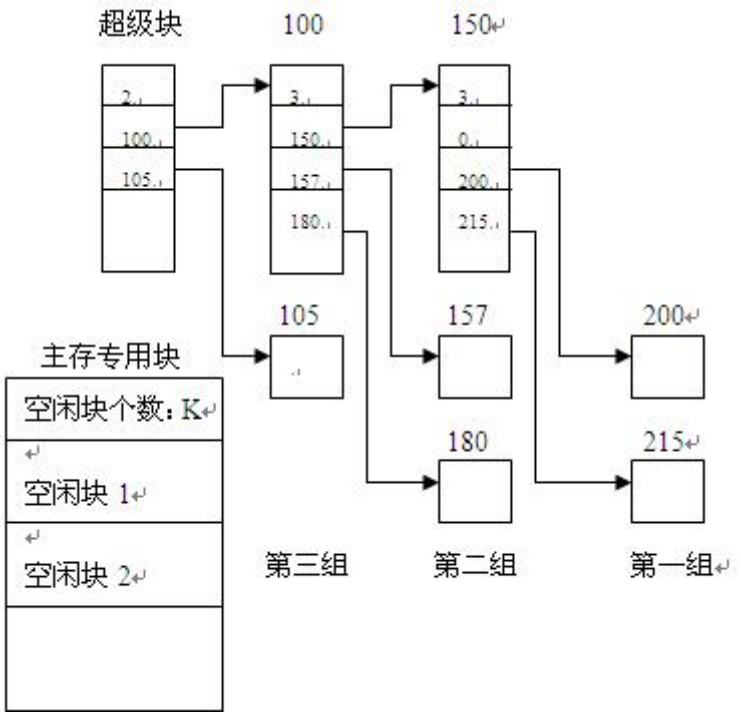

图 6　空闲块成组链接

N＝3，即每 3 块为一组。其中，第一组只有 2 块，以后的各组 3 块为一组，每组第一块登记前一组的空闲块号和空闲块数，最后一组由超级块管理。超级块用来登记最后一组的块数及块号。

（2）可用一个二维数组 A 来管理成组链接：A[M][N]。其中 M 是空闲块分成的组数，N 为每组的磁盘块数。

用 A[0]表示超级块，管理最后一组的空闲块。这样 A[0][0]表示超级块中管理的空闲块数，A[0][1]表示超级块中管理的第一个空闲块号，A[0][2]表示超级块中管理的第二个空闲块号，A[0][3]表示超级块中管理的第三个空闲块号，实际管理的块数由最后组的块数决定。A[1]管理第 2 组的空闲块。A[2]管理的是第一组的空闲块。采用栈式存取方式对数组进行操作。

为模拟实际系统，可再定义一个一维数组，作为主存专用块。系统初启时，先将超级块中的信息复制到主存专用块中。

当用户申请一个磁盘块时，查主存专用块这个一维数组，找出一空闲块号。当找到的这块为这组最后一块时，在将空闲块分给用户之前，先将这块存储的一组空闲块信息复制到主存专用块中。之后再分配给用户。分配算法如图 7 所示。

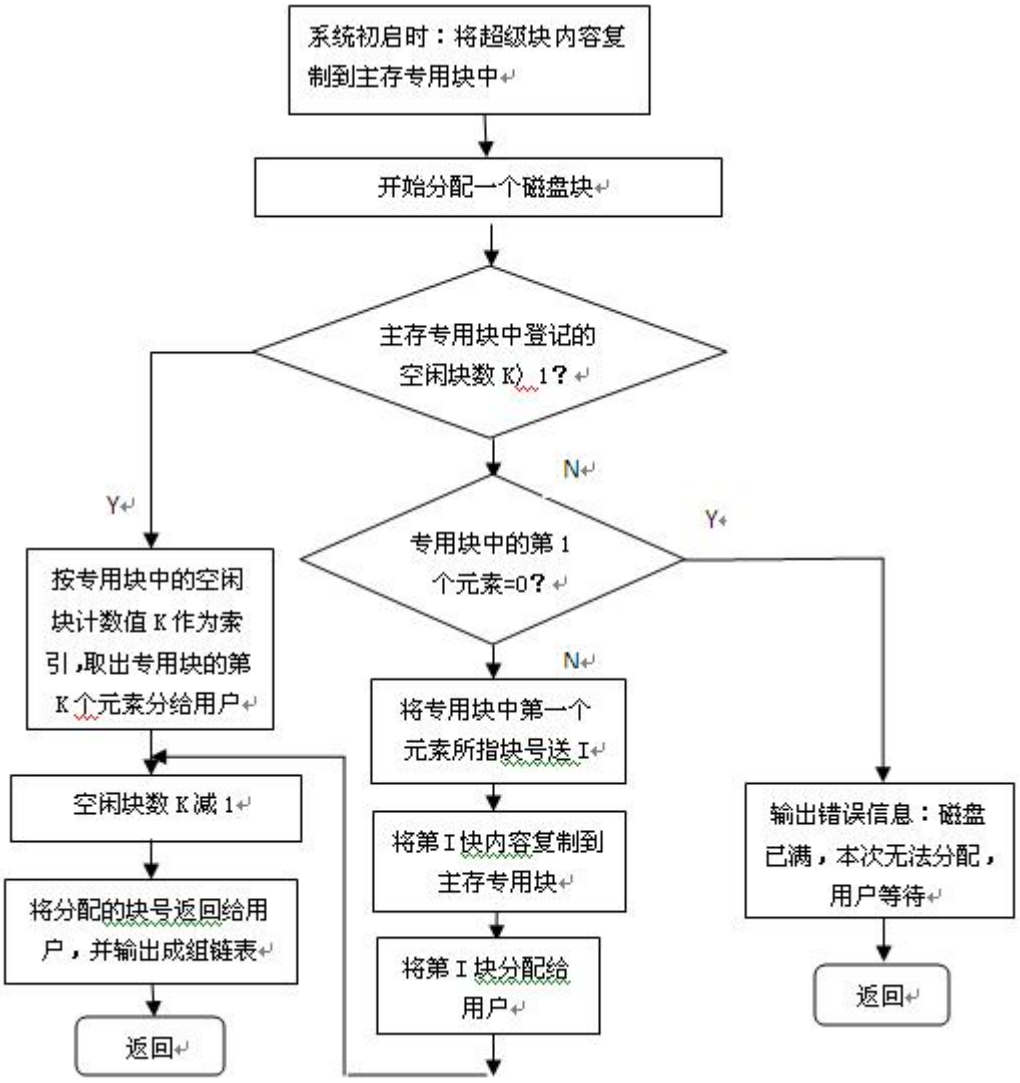

图 7 空闲块成组链接分配算法框图

（3）释放一磁盘块时，若当前主存专用块还不满，则将释放块直接登记到专用块中；若已满，则另建一组，并将该释放块作为新组的第一块，然后再把主存专用块的内容复制到该释放块中。之后将新建组的这个块的块号放入主存专用块中，并将块计数置为 1。算法结束。回收算法如图 8 所示。

（4）设计该程序，要求能显示或打印分配的磁盘块号，并显示分配或释放后空闲块链的情况。

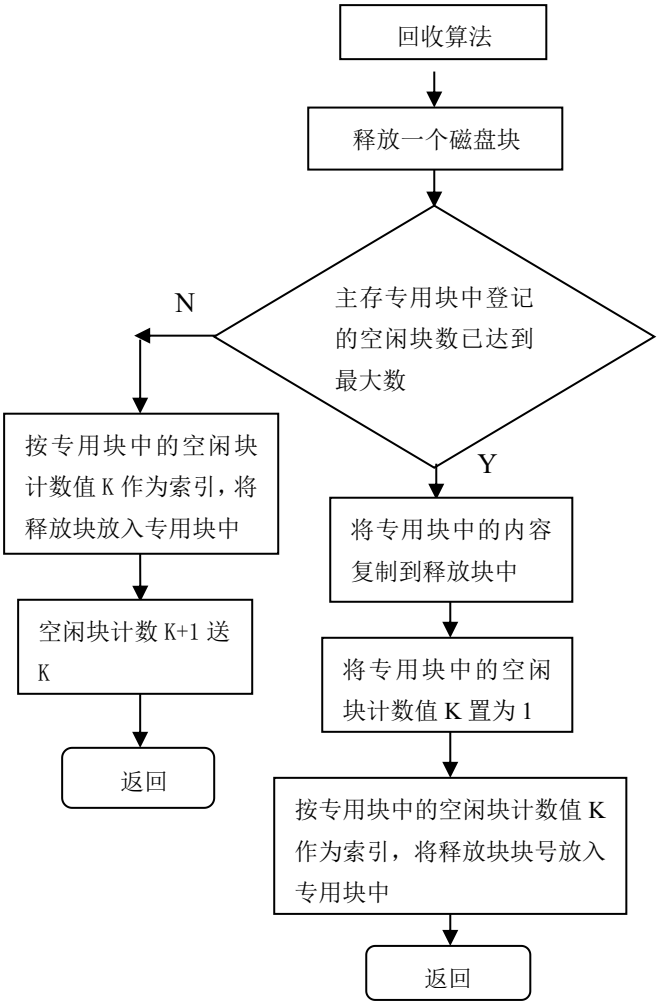

图 8　空闲块成组链接分配算法框图

附件二　　设计报告要按规范的格式

华北电力大学

课 程 设 计 报 告

(2022--2023 年度第二学期)

课程名称: 操作系统课程设计

课设题目:

院　　系: 控制与计算机工程学院

班　　级:

姓　　名:

指导教师: 贾静平

设计周数: 一周

成　　绩:

2023 年 6 月 25 日

设计报告内容

一、需求分析
二、整体功能及设计（功能划分及流程图）
三、编程实现（要求有注释）
四、使用说明
五、结果分析

排版要求

一、首页按老师给定模版填写，注意线条长度不要改变。
二、标题一、（一级标题） 设黑体四号字，左对齐，单倍行距、段前段后 0.5 行

标题 1. （二级标题）设黑体小四号字，首行缩进 2 字符，单倍行距、段前段后 0.5 行

正文　　设宋体小四号字，首行缩进 2 字符，单倍行距，两端对齐。