

5.1 实验目的

- (1) 理解设备管理的概念和任务。
- (2) 掌握独占设备的分配、回收等主要算法的原理并编程实现。

5.2 实验内容

在 Windows 系统中,编写程序实现对独占设备的分配与回收的模拟,该程序中包括建立设备类表和设备表、分配设备和回收设备的函数。

5.3 实验准备

在多道程序环境下,独占设备应采用独享分配策略,即将一个设备分配给某进程后,便由该进程独占,直至该进程完成或释放该设备,然后系统才能再将该设备分配给其他进程使用。在实验中通过模拟方法实现对独占设备的分配和回收。

在操作系统中通常要通过表格记录相应设备状态等,以便进行设备分配。在进行设备分配时所需的数据结构(表格)有设备类表和设备控制表等。

设备类表记录系统中全部设备的情况,每个设备类占一个表目,包括设备类型、设备标识符、设备驱动程序入口、拥有设备数量、可分配设备数量、设备表起始地址等,如表 5-1 所示。

表 5-1 设备类表

设备类型	拥有设备数量	可分配设备数量	设备表起始地址
input	5	2	2
printer	4	3	5
...	...	...	...

系统为每一个设备都配置了一张设备控制表,用于记录本设备的情况。每个设备占一个表目,包括设备状态、是否分配、占用作业名等,如表 5-2 所示。

表 5-2 设备控制表

绝 对 号	状 态	是 否 分 配	占用作业名	相 对 号
11	好/坏	是/否	job	1
...	...	...	...	...

作业申请某设备时,先查“设备类表”,如果该类设备的拥有设备数量满足申请要求,则从设备类表中得到该类设备的设备表起始地址,然后找到“设备控制表”中该类设备的起始地址,依次查询该类设备的表项,找到状态是好且没有分配的设备分配给作业。分配设备的过程中要修改“设备类表”中的“可分配设备数量”,并且把“设备控制表”中设备“是否分配”项更改为“是”,并填写“占用作业名”和“相对号”。

设备回收时,系统首先查看“设备控制表”,找到需要释放的设备,将该设备“是否分配”项更改为“否”,然后在“设备类表”中将“可分配设备数量”增加 1。

5.4 程序示例

```
#include <stdio.h>
#include <stdio.h>
#include <string.h>
#define N 3 //假设系统有 3 类设备
#define M 5 //假设系统有 5 个设备
struct {
    char type[10]; //设备类型
    int count; //拥有设备数量
    int remain; //现存的可分配设备数量
    int address; //该类设备在设备表中的起始地址
}equip_type[N]; //设备类表定义,假设系统有 n 个设备类型
struct {
    int number; //设备绝对号
    bool status; //判断设备状态好坏
    bool IsRemain; //设备是否分配
    char jobname[10]; //占用设备的作业名称
    int lnumber; //设备相对号
}equipment[M]; //设备表定义,假设系统有 m 个设备
// ***** 函数说明 ***** //
//设备分配函数
// ***** //
bool allocate(char * job,char * type,int mm)
{
    int i=0,t;
    //查询该类设备
    while (i<N&&strcmp(equip_type[i].type,type)!= 0) i++;
    //没有找到该类设备
    if (i>= N)
```

```

    {
        printf("无该类设备,设备分配请求失败");
        return(false);
    }
    //所需设备现存的可用数量不足
    if(equip_type[i].remain<1)
    {
        printf("该类设备数量不足,设备分配请求失败");
        return(false);
    }
    //得到该类设备在设备表中的起始地址
    t = equip_type[i].address;
    while (!(equipment[t].status == true && equipment[t].IsRemain == false))
        t++;
    //填写作业名、设备相对号、状态
    equip_type[i].remain--;
    equipment[t].IsRemain = true;
    strcpy(equipment[t].jobname, job);
    equipment[t].lnumber = mm;
    return true;
}
// ***** 函数说明 ***** //
//设备回收函数
// ***** //
bool reclaim(char * job,char * type)
{
    int i = 0, t, j, k = 0, nn;
    while (i < N && strcmp(equip_type[i].type, type) != 0) i++;
    //没有找到该类设备
    if (i >= N)
    {
        printf("无该类设备,设备分配请求失败");
        return(false);
    }
    //得到该类设备在设备表中的起始地址
    t = equip_type[i].address;
    //得到该类设备的数量
    j = equip_type[i].count;
    nn = t + j;
    //修改设备为可使用状态和该类设备可用数量
    for( ; t < nn; t++)
    {
        if(strcmp(equipment[t].jobname, job) == 0 && equipment[t].IsRemain == true)
        {
            equipment[t].IsRemain = false;
            k++;
        }
    }
}

```

```
    equip_type[i].remain = equip_type[i].remain + k;
    if(k == 0) printf("作业没有使用该类设备");
    return true;
}

void main()
{
    char job[10];
    int i,mm,choose;
    char type[10];
    strcpy(equip_type[0].type,"input");//设备类型:输入设备
    equip_type[0].count = 2;
    equip_type[0].remain = 2;
    equip_type[0].address = 0;
    strcpy(equip_type[1].type,"printer");
    equip_type[1].count = 3;
    equip_type[1].remain = 3;
    equip_type[1].address = 2;
    strcpy(equip_type[2].type,"disk");
    equip_type[2].count = 4;
    equip_type[2].remain = 4;
    equip_type[2].address = 5;

    for (i = 0; i < 5; i++)
    {
        equipment[i].number = i;
        equipment[i].status = 1;
        equipment[i].IsRemain = false;
    }
    while(1)
    {
        printf("\n0 -- 退出,1 -- 分配,2 -- 回收,3 -- 显示");
        printf("\n 请选择功能项:");
        scanf("%d",&choose);
        switch(choose)
        {
            case 0:
                return;
            case 1:
                printf("请输入作业名、作业所需设备类和设备相对号");
                scanf("%s %s %d", job, type, &mm);
                allocate(job, type, mm);        //分配设备
                break;
            case 2:
                printf("请输入作业名和作业要归还的设备类型");
                scanf("%s %s", job, type);
                reclaim(job, type);            //回收设备
                break;
            case 3:
```

```

printf("\n 输出设备类表:\n");
printf("设备类型\t设备数量\t空闲设备数量\n");
for(i = 0; i < N; i++)

printf(" %8s %10d %18d\n", equip_type[i].type, equip_type[i].count, equip_type[i]
.remain);

printf("----- \n");
printf("输出设备控制表:\n");
printf("绝对号 状态 是否分配 占用作业名 相对号\n");
for(i = 0; i < M; i++)
{

printf(" %3d %7d %8d %10s %7d\n", equipment[i].number, equipment[i].status,
equipment[i].IsRemain, equipment[i].jobname, equipment[i].lnumber);
}
break;
default:
return;
}
}
}
}

```

5.5 实验结果

显示所有设备,如图 5-1 所示。



图 5-1 显示所有设备

分配设备后,如图 5-2 所示。

设备回收后,如图 5-3 所示。



图 5-2 分配设备



图 5-3 设备回收