



操作系统实践

实验06 内存管理 (1)

软件学院 基础组

本节知识点

- 1 实验目的
- 2 实验准备
- 3 实验内容
- 4 问题思考

本节知识点

- 1 实验目的
- 2 实验准备
- 3 实验内容
- 4 问题思考

1 实验目的

- 1、加深对固定分区分配内存管理方式的理解
- 2、掌握在分区大小相等和不等情况下，进行对两种存储管理方式利弊
- 3、理解内存分配和回收的原理

本节知识点

- 1 实验目的
- 2 实验准备
- 3 实验内容
- 4 问题思考

2 实验准备

2.1 固定分区分配原理

将内存用户空间划分为若干个固定大小的区域，在每个分区中只装入一道作业，这样把用户空间划分为几个分区，便允许有几道作业并发执行。当有一空闲分区时，便可以再从外存的后备作业队列中，选择一个适当大小的作业装入该分区，当该作业结束时，可再从后备作业队列中找出另一作业调入该分区。

2 实验准备

2.2 划分分区的方法

可用两种方法将内存的用户空间划分为若干个固定大小的分区：

- 分区大小相等：缺乏灵活性，用于一台计算机控制多个相同对象的场合；
- 分区大小不等：把内存区划分成含有多个较小的分区、适量的中等分区及少量的大分区，可根据程序的大小为之分配适当的分区。

2 实验准备

2.3 实现方式

为便于内存分配，通常将分区按大小进行排队，并为之建立一张分区使用表，其中各表项包括每个分区的起始地址、大小及状态(是否已分配)。

当有一用户程序要装入时，由内存分配程序检索该表，从中找出一个能满足要求的、尚未分配的分区，将之分配给该程序，然后将该表项中的状态置为“已分配”；若未找到大小足够的分区，则拒绝为该用户程序分配内存。

2 实验准备

2.3 实现方式

分区号	大小(K)	起址(K)	状态
1	12	20	已分配
2	32	32	已分配
3	64	64	已分配
4	128	128	未分配

分区说明表



存储空间分配情况

2 实验准备

2.4 存储形式

- 结构体，数组（或链表）
- PCB中要包含资源信息：分区号、分区大小等信息
- 存储形式

2 实验准备

➤ 存储形式

```
typedef struct
{
    char Name[NAME_MAXSIZE]; //进程名
    int PartiNum;             //分区号
    int MemorySize;           //内存大小
    DistributState DistbutSt; //分配状态
}PCB;
```

2 实验准备

```
typedef struct {    //分区号用数组下标代替
    int PartitionSize;        //分区大小
    int PartInitAddress; //分区起始地址
    PartitionSt PartitionSt; //分区状态
}PartitionInfo;
typedef struct{
    PartitionInfo elem[LIST_INITSIZE];
    int length;
}SqList;
```

//分区说明表

本节知识点

- 1 实验目的
- 2 实验准备
- 3 实验内容
- 4 问题思考

3 实验内容

3.1 实验思路

先根据进程申请的空间大小在分区说明表中进行寻找合适的空闲分区块，即查找分区说明表，找到第一个满足申请长度的空闲区，将其分配。否则，该进程无法分配；当进程结束时，修改对应的分区表，回收内存空间，从而完成内存的申请和释放。

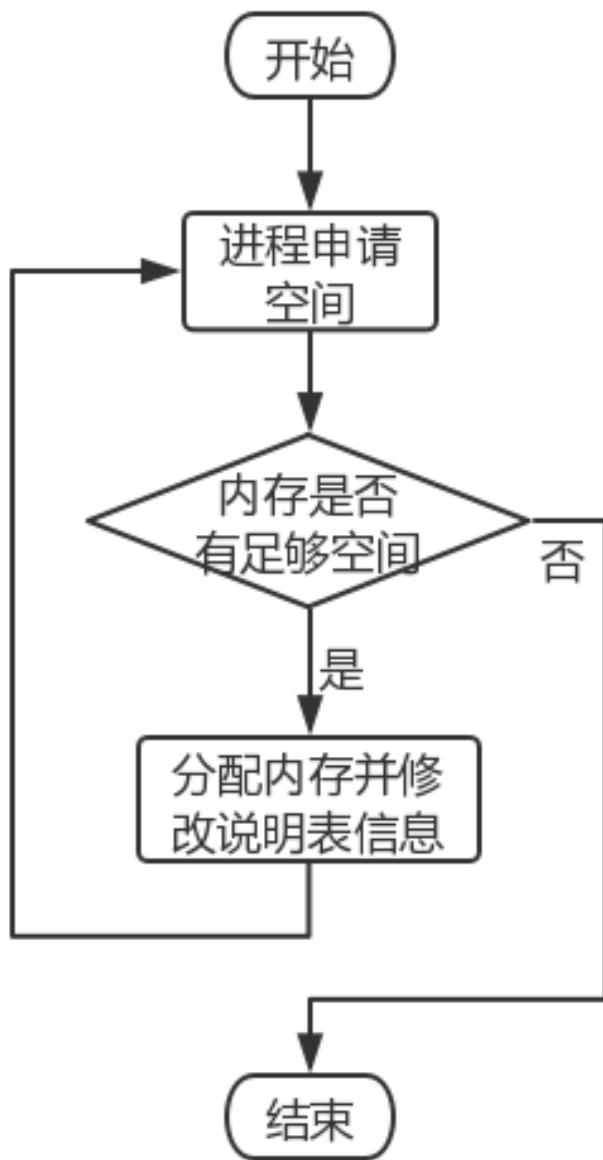
3 实验内容

3.2 固定分区实现步骤

- 1. 初始状态内存含有5块分区块（可以是大小相等，也可以大小不等），分区说明表记录内存使用信息；
- 2. 设置多个进程（包含申请的内存大小信息）放入后备队列，进程申请空间，操作系统根据申请空间大小，查找分区说明表，找是否有合适的空间，如果找到，将内存分给进程；否则，继续等待；

3 实验内容

- 3. 根据进程的退出次序，再将空间依次进行回收，修改分区说明表，当有一空闲分区时，便可以再从后备作业队列中，选择一个适当大小的作业装入该分区，并修改分区说明表；
- 4. 要求当每次修改分区说明表时，都要打印一次分区说明表的信息，以及进程信息。



3 实验内容

3.3 结果评价

对分区大小相等和不等的情况进行比较，分析其优缺点。

本节知识点

- 1 实验目的
- 2 实验准备
- 3 实验内容
- 4 问题思考

4 问题思考

- 1、采用固定分区方式，必然会造成存储空间的浪费。（作业）
- 2、分配算法过于单一，存在弊端。

谢谢观赏！