

三、内存管理（1）

1 程序说明

1.1 实验要求

- 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

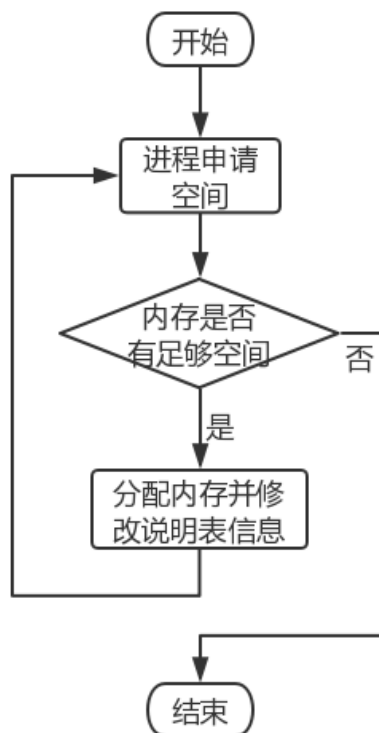
1.2 数据结构

程序中用到了哪些数据结构，分条描述。

1.3 函数功能说明

可以选择给出的现有函数接口去实现，也可以自己去写，但是要将自己写的函数一一列出，并说明函数的功能。

1.4 程序编写的思路及流程



实验 06 流程图

2 运行效果

2.1 结束进程

```
C:\Windows\system32\cmd.exe

=====
进程名 | 起始位置 | 申请大小 | 是否分配
P1      | 16        | 16        | 是
P2      | 32        | 32        | 是
P3      | 64        | 48        | 是
P4      | 128       | 96        | 是
P5      | 256       | 100       | 是
=====

分区号 | 起始位置 | 分区大小 | 是否分配
1       | 16        | 16        | P1
2       | 32        | 32        | P2
3       | 64        | 64        | P3
4       | 128       | 128       | P4
5       | 256       | 256       | P5
=====

=====
1. 结 束 进 程
2. 添 加 进 程
3. 退 出 系 统
=====

请选择:1
要结束的进程名:P4

=====
进程名 | 起始位置 | 申请大小 | 是否分配
P1      | 16        | 16        | 是
P2      | 32        | 32        | 是
P3      | 64        | 48        | 是
P5      | 256       | 100       | 是
=====

分区号 | 起始位置 | 分区大小 | 是否分配
1       | 16        | 16        | P1
2       | 32        | 32        | P2
3       | 64        | 64        | P3
4       | 128       | 128       | 否
5       | 256       | 256       | P5
=====

请按任意键继续. . . █
```

2.2 添加进程

```
C:\Windows\system32\cmd.exe

=====
| 进程名 | 起始位置 | 申请大小 | 是否分配 |
| P1     | 16       | 16       | 是       |
| P2     | 32       | 32       | 是       |
| P3     | 64       | 48       | 是       |
| P5     | 256      | 100      | 是       |
|=====|
| 分区号 | 起始位置 | 分区大小 | 是否分配 |
| 1      | 16       | 16       | P1       |
| 2      | 32       | 32       | P2       |
| 3      | 64       | 64       | P3       |
| 4      | 128      | 128      | 否       |
| 5      | 256      | 256      | P5       |
|=====|

=====
| 1. 结 束 进 程 |
| 2. 添 加 进 程 |
| 3. 退 出 系 统 |
|=====|

请选择:2
请输入添加的进程名，进程所占内存大小:P6 50

=====
| 进程名 | 起始位置 | 申请大小 | 是否分配 |
| P1     | 16       | 16       | 是       |
| P2     | 32       | 32       | 是       |
| P3     | 64       | 48       | 是       |
| P5     | 256      | 100      | 是       |
| P6     | 128      | 50       | 是       |
|=====|

=====
| 分区号 | 起始位置 | 分区大小 | 是否分配 |
| 1      | 16       | 16       | P1       |
| 2      | 32       | 32       | P2       |
| 3      | 64       | 64       | P3       |
| 4      | 128      | 128      | P6       |
| 5      | 256      | 256      | P5       |
|=====|

请按任意键继续. . . █
```

3 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。