OS 实验指南 -- Python版

1. 服务进程调度

1. 先来先服务

本实验是模拟进程调度中的先来先服务算法，每次CPU都是按照进入就绪队列的先后次序依次选中一个进程装入CPU运行，等结束时再选取下一个。

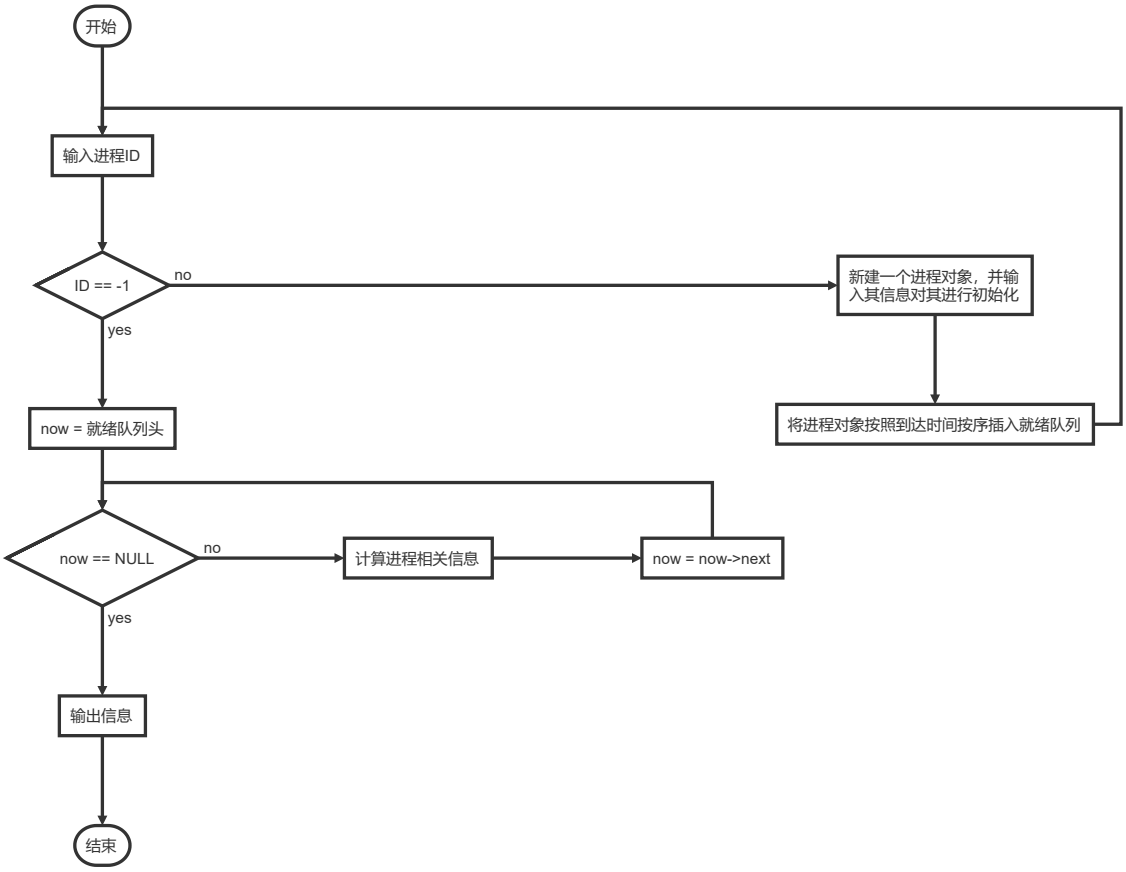
类成员变量：

```
self.id = id              # 编号
self.name = name          # 进程名
self.arrive = arrive      # 到达就绪队列时间
self.zx = zx              # 执行时间
self.start = None         # 开始时间
self.finish = None        # 完成时间
self.zz = None            # 周转时间 = 完成时间 - 到达时间
self.zzxs = None          # 带权周转系数 = 周转时间 / 执行时间
```

函数包括：

- `queue.buildQue()`：构建队列
- `queue.output()`：打印函数
- `fcfs()`：先来先服务算法

程序流程图：


开始
输入进程ID
ID == -1
no
yes
新建一个进程对象，并输入其信息对其进行初始化
将进程对象按照到达时间按序插入就绪队列
now = 就绪队列头
now == NULL
no
计算进程相关信息
now = now->next
yes
输出信息
结束

实验结果：

```
PS D:\学习资料\操作系统\python\2. 服务进程调度\modules> python3 fcfs.py
ID号 名字 到达时间 执行时间（分钟）：
1001  p1     9:40    20
1004  p4    10:10    10
1005  p5    10:05    30
1002  p2     9:55    15
1003  p3     9:45    25
-1
模拟进程FCFS调度过程输出结果：
ID号    名字   到达时间   执行时间（分钟）   开始时间   完成时间   周转时间（分钟）   带权周转系数：
1001    p1     9:40          20            9:40      10:00          20           1.00
1003    p3     9:45          25            10:00     10:25          40           1.60
1002    p2     9:55          15            10:25     10:40          45           3.00
1005    p5     10:05         30            10:40     11:10          65           2.17
1004    p4     10:10         10            11:10     11:20          70           7.00
系统平均周转周期时间为：                                          48.00
系统带权平均周转周期为：                                                       2.95
```

2. 优先级调度

本实验是模拟进程调度中的优先级调度算法，CPU先看当前有哪些进程进入了就绪队列，再从其中选去优先级最高的一个进程装入CPU运行，等结束之后重复上述过程。

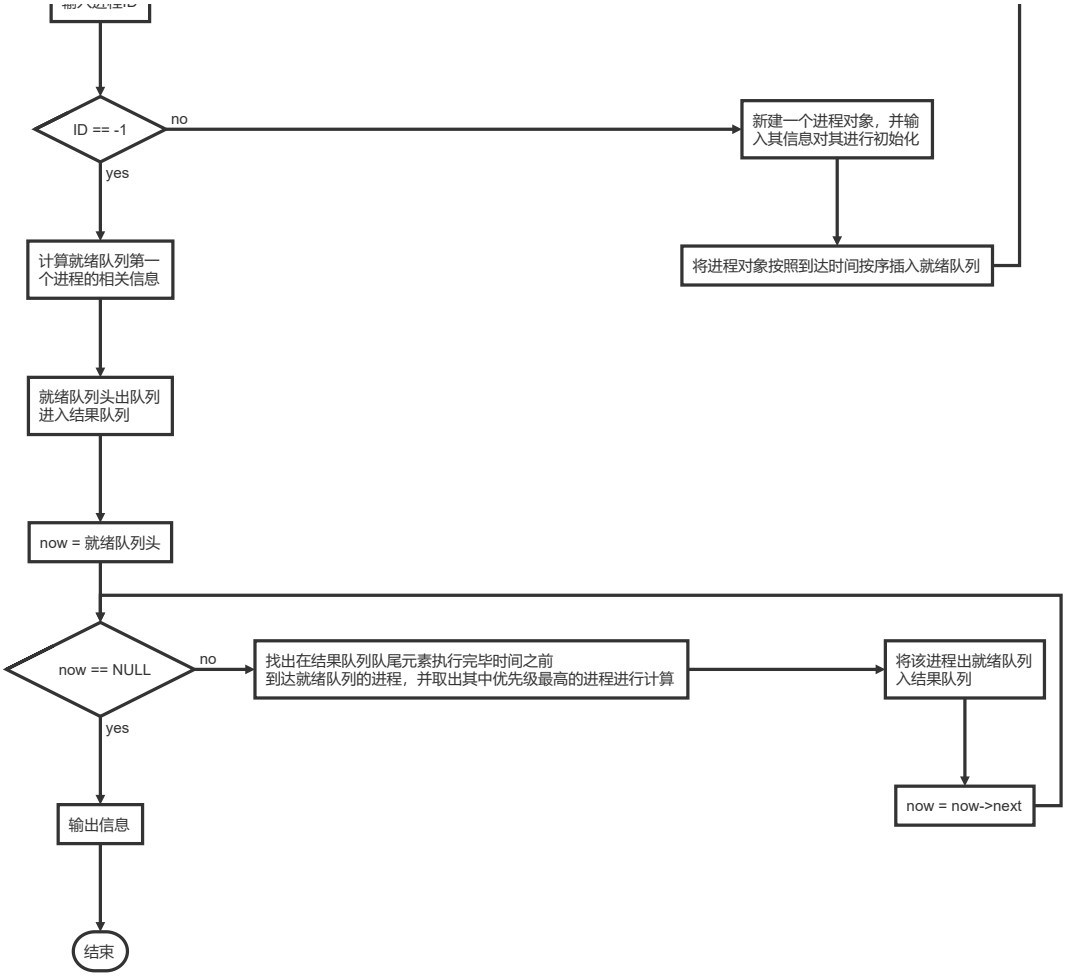
类成员变量：

```
self.id = id              # 编号
self.name = name          # 进程名
self.good = good          # 优先级
self.arrive = arrive      # 到达就绪队列时间
self.zx = zx              # 执行时间
self.start = None         # 开始时间
self.finish = None        # 完成时间
self.zz = None            # 周转时间 = 完成时间 - 到达时间
self.zzxs = None          # 带权周转系数 = 周转时间 / 执行时间
```

函数包括：

- `queue.buildQue()`：构建队列
- `queue.output()`：打印函数
- `ps()`：优先级调度算法

程序流程图：



实验结果：

```
PS D:\学习资料\操作系统\python\2. 服务进程调度\modules> python3 ps.py
ID号 名字 优先级 到达时间 执行时间（分钟）：
1001  p1    1      9:40      20
1004  p4    4     10:10      10
1005  p5    3     10:05      30
1002  p2    3      9:55      15
1003  p3    2      9:45      25
-1
模拟进程优先级调度过程输出结果：
ID号    名字   优先级   到达时间   执行时间（分钟）   开始时间   完成时间   周转时间（分钟）   带权周转系数：
1001    p1      1      9:40         20             9:40      10:00          20           1.00
1002    p2      3      9:55         15             10:00     10:15          20           1.33
1004    p4      4      10:10        10             10:15     10:25          15           1.50
1005    p5      3      10:05        30             10:25     10:55          50           1.67
1003    p3      2      9:45         25             10:55     11:20          95           3.80
系统平均周转周期时间为：                                                  40.00
系统带权平均周转周期为：                                                                1.86
```

3. 时间片轮转调度

本实验是模拟进程调度中的时间片轮转算法，首先对所有进程按到达时间排好序，然后逐个对就绪队列中的进程轮流进入CPU执行，每次开始的时间就是上个进程让出CPU的时间，在该进程本轮结束前(含结束时间)，所有入队的进程均按时间先后入队，结束时间到再将该进程排到队列的末尾，从而进入后续循环。

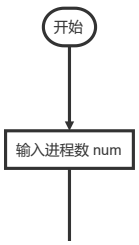
类成员变量：

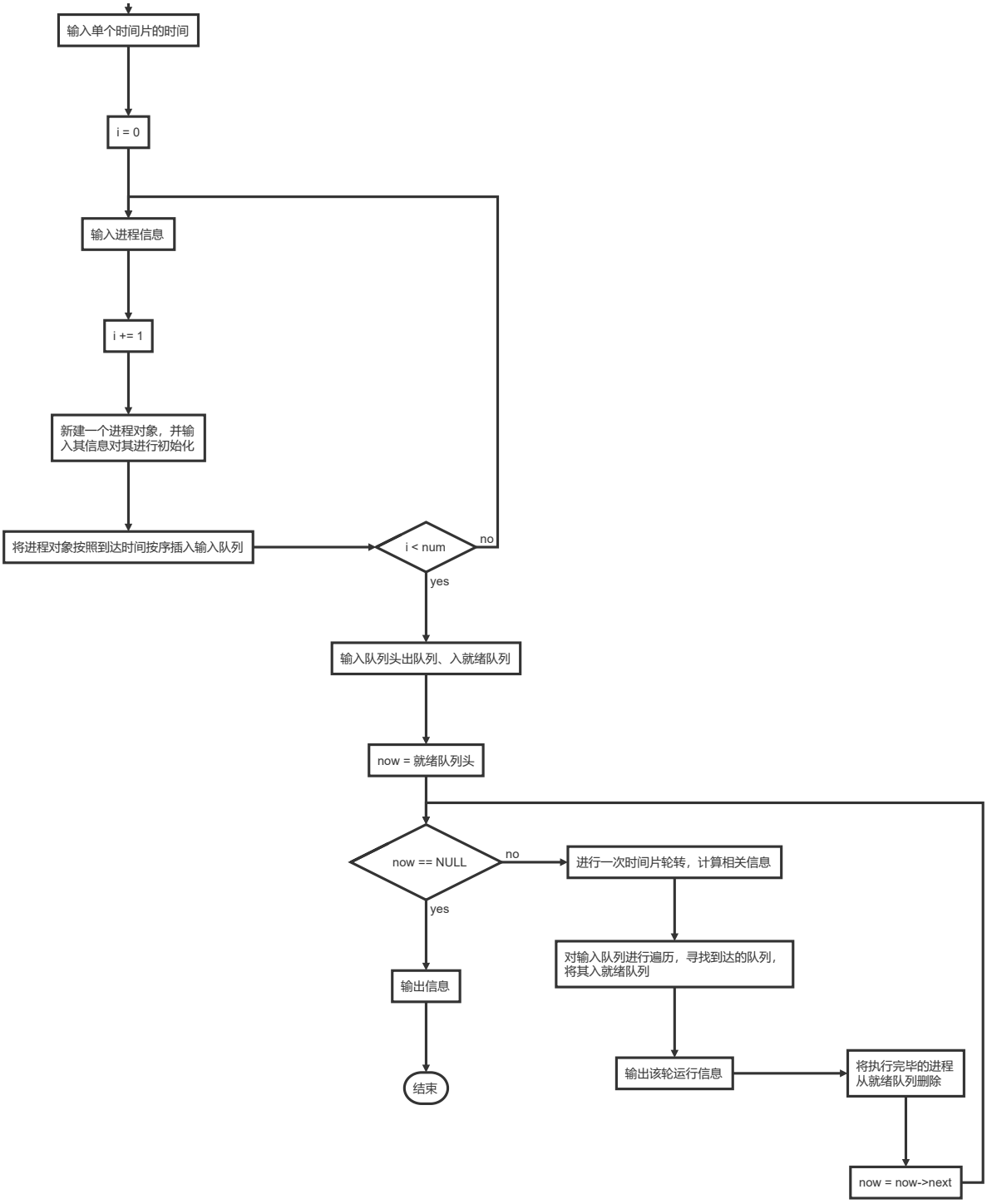
```
self.id = id              # 编号
self.name = name          # 进程名
self.arrive = arrive      # 到达就绪队列时间
self.zx = zx              # 执行时间
self.start = None         # 开始时间
self.finish = None        # 完成时间
self.zz = None            # 周转时间 = 完成时间 - 到达时间
self.zzxs = None          # 带权周转系数 = 周转时间 / 执行时间
self.nowstart = None      # 当前开始时间
self.donetime = 0         # 已完成时间
self.retime = zx          # 剩余完成时间
```

函数包括：

- `queue.buildQue()`：构建队列
- `queue.output()`：打印函数
- `rr()`：时间片轮转调度算法

程序流程图：




输入单个时间片的时间
i = 0
输入进程信息
i += 1
新建一个进程对象，并输入其信息对其进行初始化
将进程对象按照到达时间按序插入输入队列
i < num
no
yes
输入队列头出队列、入就绪队列
now = 就绪队列头
now == NULL
no
yes
进行一次时间片轮转，计算相关信息
对输入队列进行遍历，寻找到达的队列，将其入就绪队列
输出该轮运行信息
将执行完毕的进程从就绪队列删除
now = now->next
输出信息
结束

实验结果：

```
PS D:\学习资料\操作系统\python\2. 服务进程调度\modules> python3 rr.py
请输入进程数：5
请输入时间片的时间：8
请输入5个进程的：
ID号 名字 到达时间 执行时间（分钟）：
1001  p1     9:40    20
1004  p4    10:10    10
1005  p5    10:05    30
1002  p2     9:55    15
1003  p3     9:45    25

第1轮执行和就绪队列结果：
ID号    名字   到达时间   总执行时间（分钟）   当前开始时间   已完成时间   剩余完成时间：
1001    p1     9:40         20                 9:40          8            12
1003    p3     9:45         25                 0:00          0            25

第2轮执行和就绪队列结果：
ID号    名字   到达时间   总执行时间（分钟）   当前开始时间   已完成时间   剩余完成时间：
1003    p3     9:45         25                 9:48          8            17
1001    p1     9:40         20                 0:00          8            12
1002    p2     9:55         15                 0:00          0            15

第3轮执行和就绪队列结果：
ID号    名字   到达时间   总执行时间（分钟）   当前开始时间   已完成时间   剩余完成时间：
1001    p1     9:40         20                 9:56         16             4
1002    p2     9:55         15                 0:00          0            15
1003    p3     9:45         25                 0:00          8            17

第4轮执行和就绪队列结果：
ID号    名字   到达时间   总执行时间（分钟）   当前开始时间   已完成时间   剩余完成时间：

第4轮执行和就绪队列结果：
ID号    名字   到达时间   总执行时间（分钟）   当前开始时间   已完成时间   剩余完成时间：
1002    p2     9:55         15                 10:04          8             7
1003    p3     9:45         25                 0:00           8            17
1001    p1     9:40         20                 0:00          16             4
1005    p5     10:05        30                 0:00           0            30
1004    p4     10:10        10                 0:00           0            10

第5轮执行和就绪队列结果：
ID号    名字   到达时间   总执行时间（分钟）   当前开始时间   已完成时间   剩余完成时间：
1003    p3     9:45         25                 10:12         16             9
1001    p1     9:40         20                 0:00          16             4
1005    p5     10:05        30                 0:00           0            30
1004    p4     10:10        10                 0:00           0            10
1002    p2     9:55         15                 0:00           8             7

第6轮执行和就绪队列结果：
ID号    名字   到达时间   总执行时间（分钟）   当前开始时间   已完成时间   剩余完成时间：
1001    p1     9:40         20                 10:20         20             0
1005    p5     10:05        30                 0:00           0            30
1004    p4     10:10        10                 0:00           0            10
1002    p2     9:55         15                 0:00           8             7
1003    p3     9:45         25                 0:00          16             9

第7轮执行和就绪队列结果：
ID号    名字   到达时间   总执行时间（分钟）   当前开始时间   已完成时间   剩余完成时间：
1005    p5     10:05        30                 10:24          8            22
1004    p4     10:10        10                 0:00           0            10
1002    p2     9:55         15                 0:00           8             7
1003    p3     9:45         25                 0:00          16             9
```

```
第8轮执行和就绪队列结果：
ID号    名字    到达时间    总执行时间（分钟）    当前开始时间    已完成时间    剩余完成时间：
1004    p4      10:10         10                    10:32         8             2
1002    p2      9:55          15                    0:00          8             7
1003    p3      9:45          25                    0:00          16            9
1005    p5      10:05         30                    0:00          8             22

第9轮执行和就绪队列结果：
ID号    名字    到达时间    总执行时间（分钟）    当前开始时间    已完成时间    剩余完成时间：
1002    p2      9:55          15                    10:40         15            0
1003    p3      9:45          25                    0:00          16            9
1005    p5      10:05         30                    0:00          8             22
1004    p4      10:10         10                    0:00          8             2

第10轮执行和就绪队列结果：
ID号    名字    到达时间    总执行时间（分钟）    当前开始时间    已完成时间    剩余完成时间：
1003    p3      9:45          25                    10:47         24            1
1005    p5      10:05         30                    0:00          8             22
1004    p4      10:10         10                    0:00          8             2

第11轮执行和就绪队列结果：
ID号    名字    到达时间    总执行时间（分钟）    当前开始时间    已完成时间    剩余完成时间：
1005    p5      10:05         30                    10:55         16            14
1004    p4      10:10         10                    0:00          8             2
1003    p3      9:45          25                    0:00          24            1

第12轮执行和就绪队列结果：
ID号    名字    到达时间    总执行时间（分钟）    当前开始时间    已完成时间    剩余完成时间：
1004    p4      10:10         10                    11:03         10            0
1003    p3      9:45          25                    0:00          24            1
1005    p5      10:05         30                    0:00          16            14

第12轮执行和就绪队列结果：
ID号    名字    到达时间    总执行时间（分钟）    当前开始时间    已完成时间    剩余完成时间：
1004    p4      10:10         10                    11:03         10            0
1003    p3      9:45          25                    0:00          24            1
1005    p5      10:05         30                    0:00          16            14

第13轮执行和就绪队列结果：
ID号    名字    到达时间    总执行时间（分钟）    当前开始时间    已完成时间    剩余完成时间：
1003    p3      9:45          25                    11:05         25            0
1005    p5      10:05         30                    0:00          16            14

第14轮执行和就绪队列结果：
ID号    名字    到达时间    总执行时间（分钟）    当前开始时间    已完成时间    剩余完成时间：
1005    p5      10:05         30                    11:06         24            6

第15轮执行和就绪队列结果：
ID号    名字    到达时间    总执行时间（分钟）    当前开始时间    已完成时间    剩余完成时间：
1005    p5      10:05         30                    11:14         30            0

模拟进程时间片轮转调度算法过程输出结果：
ID号    名字    到达时间    执行时间（分钟）    首次开始时间    完成时间    周转时间（分钟）    带权周转系数：
1001    p1      9:40          20                  9:40            10:24       44                  2.20
1002    p2      9:55          15                  10:04           10:47       52                  3.47
1004    p4      10:10         10                  10:32           11:05       55                  5.50
1003    p3      9:45          25                  9:48            11:06       81                  3.24
1005    p5      10:05         30                  10:24           11:20       75                  2.50
系统平均周转周期时间为：                                                      61.40
系统带权平均周转周期为：                                                                          3.38
```

2. 死锁

1. 银行家算法

本实验首先判断每个进程对资源的最大需求量，若超出系统初始化的资源数则拒绝分配；然后逐次对每个进程的当前申请量进行判断：若申请量超过尚需求量则拒绝分配；若申请量超过系统可用资源数则推迟分配；否则进行安全性检查，即判断系统剩余资源数是否能确保系统产生一个安全执行序列，能则真正分配，否则推迟分配。

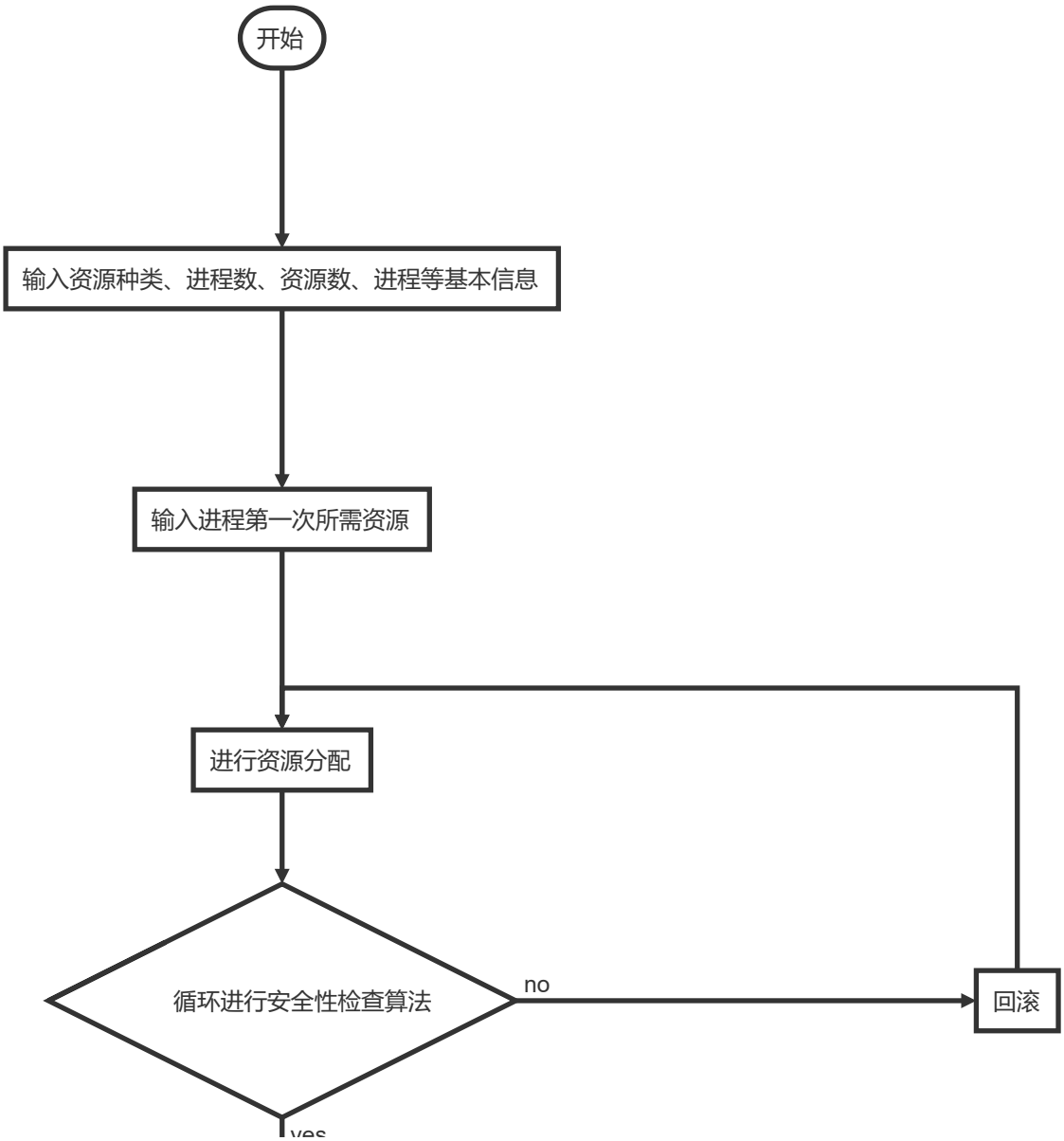
类成员变量：

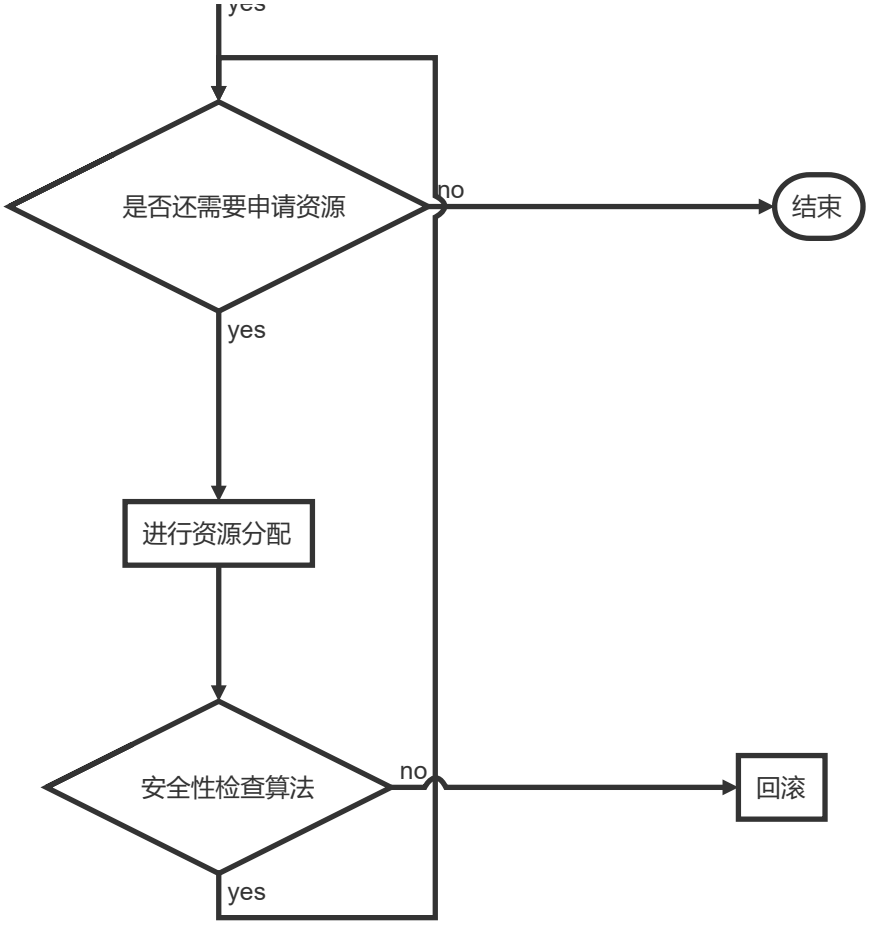
```
self.sourceTypeNum = None    # 资源种类
self.processNum = None       # 进程数
self.available = []          # 可用资源
self.max = []                # 资源最大量
self.allocation = []         # 已分配资源
self.need = []               # 尚需求量
self.flag = []               # 标识进程是否执行完毕
```

函数包括：

- `OS.build()`：入口处理函数
- `OS.output()`：数据输出
- `OS.security()`：安全性算法
- `OS.more()`：继续申请资源

程序流程图：





实验结果：

```
PS D:\学习资料\操作系统\python\3. 死锁> python3 ba.py
请输入资源种类： 3
请输入进程数： 5
请输入3类资源初始化的资源数： 10 5 7

请输入5个进程的：
进程名          最大需求量：
                A   B   C
进程p[1]        7   5   3
进程p[2]        3   2   2
进程p[3]        9   0   2
进程p[4]        2   2   2
进程p[5]        4   3   3

请输入5个进程的：
进程名          第一次申请量：
                A   B   C
进程p[1]        0   1   1
申请成功！安全序列为：p[1]-->p[2]-->p[3]-->p[4]-->p[5]
进程p[2]        2   0   0
申请成功！安全序列为：p[1]-->p[2]-->p[3]-->p[4]-->p[5]
进程p[3]        3   0   2
申请成功！安全序列为：p[2]-->p[1]-->p[3]-->p[4]-->p[5]
进程p[4]        2   1   1
申请成功！安全序列为：p[2]-->p[4]-->p[1]-->p[3]-->p[5]
进程p[5]        0   0   2
申请成功！安全序列为：p[2]-->p[4]-->p[1]-->p[3]-->p[5]

进程名          最大需求量              尚需求量              已分配量              执行结束否
进程p[1]        [7, 5, 3]               [7, 4, 2]             [0, 1, 1]             working
```

```
进程名          最大需求量              尚需求量              已分配量              执行结束否
进程p[1]        [7, 5, 3]              [7, 4, 2]             [0, 1, 1]             working
进程p[2]        [3, 2, 2]              [1, 2, 2]             [2, 0, 0]             working
进程p[3]        [9, 0, 2]              [6, 0, 0]             [3, 0, 2]             working
进程p[4]        [2, 2, 2]              [0, 1, 1]             [2, 1, 1]             working
进程p[5]        [4, 3, 3]              [4, 3, 1]             [0, 0, 2]             working
资源剩余数：[3, 3, 1]

是否需要再申请资源？（Y/N）y
请输入进程编号（1-5）：1
请输入进程p[0]对5类资源的申请：4 3 1
无安全序列，申请不成功！
是否需要再申请资源？（Y/N）y
请输入进程编号（1-5）：1
请输入进程p[0]对5类资源的申请：1 2 0
申请成功！安全序列为：p[2]-->p[4]-->p[1]-->p[3]-->p[5]

进程名          最大需求量              尚需求量              已分配量              执行结束否
进程p[1]        [7, 5, 3]              [6, 2, 2]             [1, 3, 1]             working
进程p[2]        [3, 2, 2]              [1, 2, 2]             [2, 0, 0]             working
进程p[3]        [9, 0, 2]              [6, 0, 0]             [3, 0, 2]             working
进程p[4]        [2, 2, 2]              [0, 1, 1]             [2, 1, 1]             working
进程p[5]        [4, 3, 3]              [4, 3, 1]             [0, 0, 2]             working
资源剩余数：[2, 1, 1]

是否需要再申请资源？（Y/N）y
请输入进程编号（1-5）：4
请输入进程p[3]对5类资源的申请：0 1 1
申请成功！安全序列为：p[2]-->p[1]-->p[3]-->p[5]

进程名          最大需求量              尚需求量              已分配量              执行结束否
进程p[1]        [7, 5, 3]              [6, 2, 2]             [1, 3, 1]             working
进程p[2]        [3, 2, 2]              [1, 2, 2]             [2, 0, 0]             working
进程p[3]        [9, 0, 2]              [6, 0, 0]             [3, 0, 2]             working
进程p[4]        [2, 2, 2]              [0, 0, 0]             [2, 2, 2]             finished
进程p[5]        [4, 3, 3]              [4, 3, 1]             [0, 0, 2]             working
资源剩余数：[4, 2, 2]

是否需要再申请资源？（Y/N）y
请输入进程编号（1-5）：2
请输入进程p[1]对5类资源的申请：1 2 2
申请成功！安全序列为：p[1]-->p[3]-->p[5]

进程名          最大需求量              尚需求量              已分配量              执行结束否
进程p[1]        [7, 5, 3]              [6, 2, 2]             [1, 3, 1]             working
进程p[2]        [3, 2, 2]              [0, 0, 0]             [3, 2, 2]             finished
进程p[3]        [9, 0, 2]              [6, 0, 0]             [3, 0, 2]             working
进程p[4]        [2, 2, 2]              [0, 0, 0]             [2, 2, 2]             finished
进程p[5]        [4, 3, 3]              [4, 3, 1]             [0, 0, 2]             working
资源剩余数：[6, 2, 2]

是否需要再申请资源？（Y/N）y
请输入进程编号（1-5）：3
请输入进程p[2]对5类资源的申请：6 0 0
申请成功！安全序列为：p[1]-->p[5]
```

```
进程名          最大需求量          尚需求量          已分配量          执行结束否
进程p[1]        [7, 5, 3]          [6, 2, 2]          [1, 3, 1]          working
进程p[2]        [3, 2, 2]          [0, 0, 0]          [3, 2, 2]          finished
进程p[3]        [9, 0, 2]          [0, 0, 0]          [9, 0, 2]          finished
进程p[4]        [2, 2, 2]          [0, 0, 0]          [2, 2, 2]          finished
进程p[5]        [4, 3, 3]          [4, 3, 1]          [0, 0, 2]          working
资源剩余数：[9, 2, 4]

是否需要再申请资源？（Y/N）y
请输入进程编号（1-5）：1
请输入进程p[0]对5类资源的申请：6 2 2
申请成功！安全序列为：p[5]

进程名          最大需求量          尚需求量          已分配量          执行结束否
进程p[1]        [7, 5, 3]          [0, 0, 0]          [7, 5, 3]          finished
进程p[2]        [3, 2, 2]          [0, 0, 0]          [3, 2, 2]          finished
进程p[3]        [9, 0, 2]          [0, 0, 0]          [9, 0, 2]          finished
进程p[4]        [2, 2, 2]          [0, 0, 0]          [2, 2, 2]          finished
进程p[5]        [4, 3, 3]          [4, 3, 1]          [0, 0, 2]          working
资源剩余数：[10, 5, 5]

是否需要再申请资源？（Y/N）y
请输入进程编号（1-5）：5
请输入进程p[4]对5类资源的申请：4 3 1
申请成功！安全序列为：

进程名          最大需求量          尚需求量          已分配量          执行结束否
进程p[1]        [7, 5, 3]          [0, 0, 0]          [7, 5, 3]          finished
进程p[2]        [3, 2, 2]          [0, 0, 0]          [3, 2, 2]          finished
进程p[3]        [9, 0, 2]          [0, 0, 0]          [9, 0, 2]          finished
进程p[4]        [2, 2, 2]          [0, 0, 0]          [2, 2, 2]          finished
进程p[5]        [4, 3, 3]          [0, 0, 0]          [4, 3, 3]          finished
资源剩余数：[10, 5, 7]
```

3. 作业调度

1. 先来先服务

本实验在算法上与进程FCFS调度算法类似，细节参考进程FCFS调度算法即可。

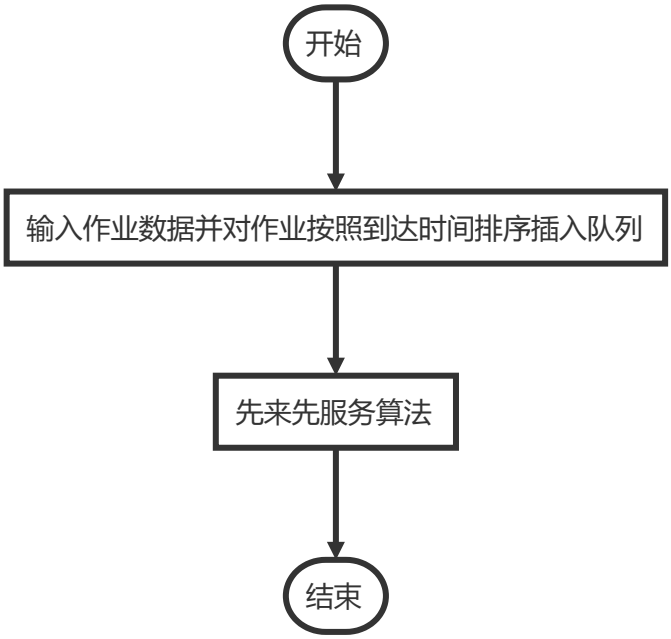
类成员变量：

```
self.name = name          # 作业名
self.arrive = arrive      # 作业到达时间
self.zx = zx              # 作业执行时间
self.start = None         # 作业开始时间
self.finish = None        # 作业完成时间
self.zz = None            # 作业周转时间
self.zzxs = None          # 作业带权周转系数
self.wait = None          # 作业调度等待时间
```

函数包括：

- `queue.buildQue()`：插入数据、新建队列
- `queue.output()`：队列输出
- `fcfs()`：先来先服务算法

程序流程图：



实验结果：

```
PS D:\学习资料\操作系统\python\4. 作业调度> python3 fcfs.py
请输入你需要创建的作业数：4
请依次输入：
作业名    入井时间    运行时间：
JOB1      8:00        120
JOB2      8:50        50
JOB3      9:00        10
JOB4      9:50        20

模拟作业FCFS调度过程输出结果：
名字  到达时间  运行时间  作业调度时间      作业调度等待时间  进程调度时间  进程调度等待时间  完成时间      周转时间（分钟）  带权周转系数：
JOB1   8:00       120       8:00                 0          8:00              0          10:00           120          1.0000
JOB2   8:50        50       10:00                70          10:00             0          10:50           120          2.4000
JOB3   9:00        10       10:50               110          10:50             0          11:00           120          12.0000
JOB4   9:50        20       11:00                70          11:00             0          11:20            90          4.5000
系统平均周转周期时间为：                                                                                  112.50
系统带权平均周转周期为：                                                                                              4.9750
```

4. 存储管理

1. 固定分区算法

本实验模拟存储管理实验中的固定分区算法，算法思想是预先将内存空间划分成若干个空闲分区，分配过程根据用户需求将某一个满足条件的分区直接分配（不进行分割），作业完成后回收对应内存，整个分配过程分区大小和个数不发生变化。

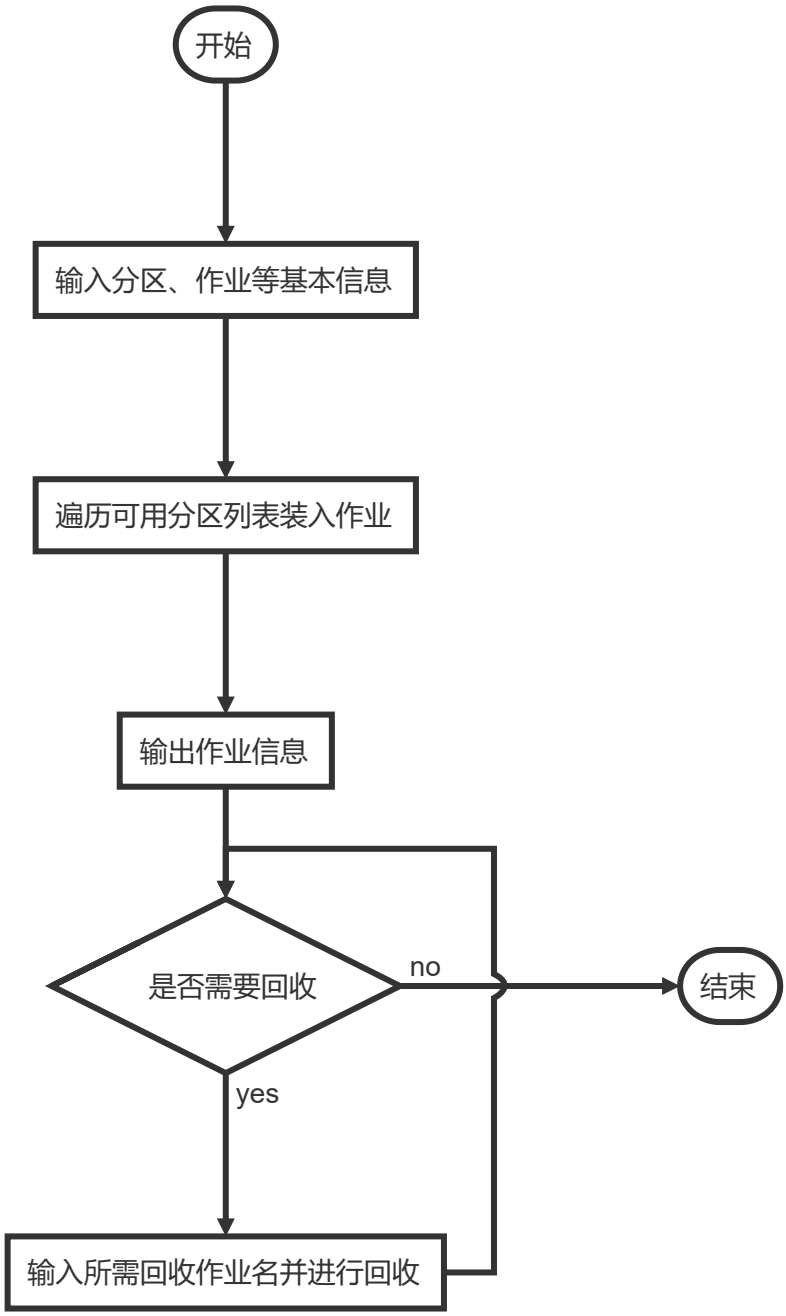
类成员变量：

```
self.num = None          # 分区块数
self.blocks = []         # 分区
self.worknum = None      # 作业个数
self.worksize = []       # 作业大小
```

函数包括：

- `Storage.build()`：程序入口
- `Storage.output()`：输出信息
- `Storage.more()`：是否还需要回收

程序流程图：


开始
输入分区、作业等基本信息
遍历可用分区列表装入作业
输出作业信息
是否需要回收
no
结束
yes
输入所需回收作业名并进行回收

实验结果：

```
PS D:\学习资料\操作系统\python\5. 存储管理\modules> python3 .\fpsm.py
请输入系统的分区块数: 5
请依次输入:
分区号   大小     起始
1        12       20
2        32       32
3        64       64
4        128      128
5        100      256

********************打印区块信息********************
分区号   大小(KB)         起始(KB)         状态
1        12               20               0
2        32               32               0
3        64               64               0
4        128              128              0
5        100              256              0
请输入作业的个数: 3
请输入这3个作业的信息:
请输入作业1的大小: 30
请输入作业2的大小: 60
请输入作业3的大小: 90
打印各作业信息:
作业名           作业大小
JOB1             30KB
JOB2             60KB
JOB3             90KB

********************打印区块信息********************
分区号   大小(KB)         起始(KB)         状态
1        12               20               0
2        32               32               JOB1
3        64               64               JOB2
4        128              128              JOB3
5        100              256              0
是否还需要回收? (y/n) n
```

2. 可变分区算法

本实验模拟存储管理实验中的可变分区算法，该算法主要思想是系统并不预先划分内存区间，而是在作业装入时根据作业的实际需要动态地划分内存空间。若无空闲的存储空间或无足够大的空闲存储空间供分配时，则令该作业等待。

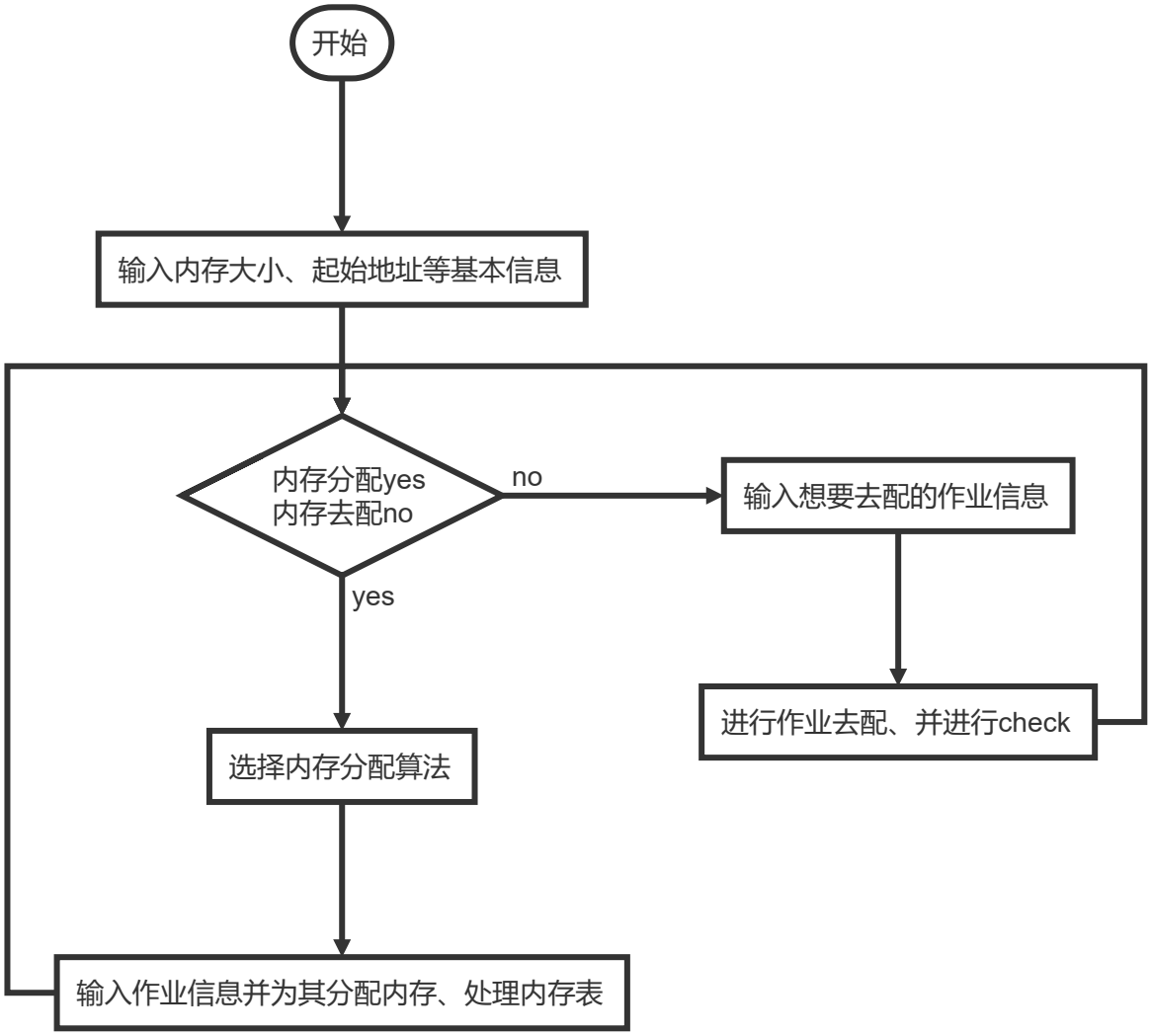
类成员变量:

```
self.size = None          # 内存大小
self.start = None         # 起始地址
self.use = []             # 已用分区
self.free = []            # 空闲分区
```

函数包括:

- `Strorge.build()`: 程序入口
- `Strorge.output()`: 输出信息
- `Strorge.check()`: 检查是否存在相连分区

程序流程图:


开始
输入内存大小、起始地址等基本信息
内存分配yes
内存去配no
no
yes
输入想要去配的作业信息
进行作业去配、并进行check
选择内存分配算法
输入作业信息并为其分配内存、处理内存表

实验结果：

```
PS D:\学习资料\操作系统\python\5. 存储管理\modules> python3 vp.py
请输入内存大小为: 256
请输入起始地址大小为: 40
********************可变分区管理********************
        *            1.内存分配          *
        *            2.内存去配          *
        *            0.退出              *
                     请输入选项[1]
********************分配算法********************
        *            1.最先分配法        *
        *            2.最优分配法        *
        *            3.最坏分配法        *
                     请输入选项[1]

********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
未分配:
分区号  大小(KB)        起始(KB)        状态
1       256             40              空闲
请输入作业名及其所需分配的主存大小（单位: KB）: JOB_A 15
分配成功!

********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
未分配:
分区号  大小(KB)        起始(KB)        状态
2       241             55              空闲
********************可变分区管理********************
        *            1.内存分配          *
        *            2.内存去配          *
        *            0.退出              *
                     请输入选项[1]
********************分配算法********************
        *            1.最先分配法        *
        *            2.最优分配法        *
        *            3.最坏分配法        *
                     请输入选项[1]

********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
未分配:
分区号  大小(KB)        起始(KB)        状态
2       241             55              空闲
请输入作业名及其所需分配的主存大小（单位: KB）: JOB_B 50
分配成功!
```

```
********************主存分配情况********************
已分配:
分区号   大小(KB)          起始(KB)          状态
1        15                40                JOB_A
2        50                55                JOB_B
未分配:
分区号   大小(KB)          起始(KB)          状态
3        191               105               空闲
********************可变分区管理********************
         *              1.内存分配          *
         *              2.内存去配          *
         *              0.退出              *
                        请输入选项[1]
********************分配算法********************
         *              1.最先分配法        *
         *              2.最优分配法        *
         *              3.最坏分配法        *
                        请输入选项[1]

********************主存分配情况********************
已分配:
分区号   大小(KB)          起始(KB)          状态
1        15                40                JOB_A
2        50                55                JOB_B
未分配:
分区号   大小(KB)          起始(KB)          状态
3        191               105               空闲
请输入作业名及其所需分配的主存大小（单位：KB）：JOB_C 10
分配成功!

********************主存分配情况********************
已分配:
分区号   大小(KB)          起始(KB)          状态
1        15                40                JOB_A
2        50                55                JOB_B
3        10                105               JOB_C
未分配:
分区号   大小(KB)          起始(KB)          状态
4        181               115               空闲
********************可变分区管理********************
         *              1.内存分配          *
         *              2.内存去配          *
         *              0.退出              *
                        请输入选项[1]
********************分配算法********************
         *              1.最先分配法        *
         *              2.最优分配法        *
         *              3.最坏分配法        *
                        请输入选项[1]

********************主存分配情况********************
```

```
已分配：
分区号   大小(KB)        起始(KB)        状态
1        15              40              JOB_A
2        50              55              JOB_B
3        10              105             JOB_C
未分配：
分区号   大小(KB)        起始(KB)        状态
4        181             115             空闲
请输入作业名及其所需分配的主存大小（单位：KB）：JOB_D 25
分配成功！

********************主存分配情况********************
已分配：
分区号   大小(KB)        起始(KB)        状态
1        15              40              JOB_A
2        50              55              JOB_B
3        10              105             JOB_C
4        25              115             JOB_D
未分配：
分区号   大小(KB)        起始(KB)        状态
5        156             140             空闲
********************可变分区管理********************
         *               1.内存分配        *
         *               2.内存去配        *
         *               0.退出            *
                         请输入选项[1]
********************分配算法********************
         *               1.最先分配法      *
         *               2.最优分配法      *
         *               3.最坏分配法      *
                         请输入选项[1]

********************主存分配情况********************
已分配：
分区号   大小(KB)        起始(KB)        状态
1        15              40              JOB_A
2        50              55              JOB_B
3        10              105             JOB_C
4        25              115             JOB_D
未分配：
分区号   大小(KB)        起始(KB)        状态
5        156             140             空闲
请输入作业名及其所需分配的主存大小（单位：KB）：JOB_E 14
分配成功！

********************主存分配情况********************
已分配：
分区号   大小(KB)        起始(KB)        状态
1        15              40              JOB_A
2        50              55              JOB_B
3        10              105             JOB_C
```

```
4       25              115             JOB_D
5       14              140             JOB_E
未分配:
分区号  大小(KB)        起始(KB)        状态
6       142             154             空闲
********************可变分区管理********************
        *           1.内存分配          *
        *           2.内存去配          *
        *           0.退出              *
                    请输入选项[2]

********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       50              55              JOB_B
3       10              105             JOB_C
4       25              115             JOB_D
5       14              140             JOB_E
未分配:
分区号  大小(KB)        起始(KB)        状态
6       142             154             空闲
请输入你要回收的作业名: JOB_B
回收成功!

********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       10              105             JOB_C
3       25              115             JOB_D
4       14              140             JOB_E
未分配:
分区号  大小(KB)        起始(KB)        状态
5       50              55              空闲
6       142             154             空闲
********************可变分区管理********************
        *           1.内存分配          *
        *           2.内存去配          *
        *           0.退出              *
                    请输入选项[2]

********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       10              105             JOB_C
3       25              115             JOB_D
4       14              140             JOB_E
未分配:
分区号  大小(KB)        起始(KB)        状态
```

```
5       50              55              空闲
6       142             154             空闲
请输入你要回收的作业名：JOB_D
回收成功！

********************主存分配情况********************
已分配：
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       10              105             JOB_C
3       14              140             JOB_E
未分配：
分区号  大小(KB)        起始(KB)        状态
4       50              55              空闲
5       25              115             空闲
6       142             154             空闲
********************可变分区管理********************
        *           1.内存分配          *
        *           2.内存去配          *
        *           0.退出              *
                    请输入选项[1]
********************分配算法********************
        *           1.最先分配法        *
        *           2.最优分配法        *
        *           3.最坏分配法        *
                    请输入选项[2]

********************主存分配情况********************
已分配：
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       10              105             JOB_C
3       14              140             JOB_E
未分配：
分区号  大小(KB)        起始(KB)        状态
4       50              55              空闲
5       25              115             空闲
6       142             154             空闲
请输入作业名及其所需分配的主存大小（单位：KB）：JOB_X 15
分配成功！

********************主存分配情况********************
已分配：
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       10              105             JOB_C
3       15              115             JOB_X
4       14              140             JOB_E
未分配：
分区号  大小(KB)        起始(KB)        状态
5       50              55              空闲
```

```
6       10              130             空闲
7       142             154             空闲
********************可变分区管理********************
        *               1.内存分配        *
        *               2.内存去配        *
        *               0.退出            *
                        请输入选项[1]
********************分配算法********************
        *               1.最先分配法      *
        *               2.最优分配法      *
        *               3.最坏分配法      *
                        请输入选项[3]

********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       10              105             JOB_C
3       15              115             JOB_X
4       14              140             JOB_E
未分配:
分区号  大小(KB)        起始(KB)        状态
5       50              55              空闲
6       10              130             空闲
7       142             154             空闲
请输入作业名及其所需分配的主存大小（单位：KB）：JOB_G 5
分配成功!

********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       10              105             JOB_C
3       15              115             JOB_X
4       14              140             JOB_E
5       5               154             JOB_G
未分配:
分区号  大小(KB)        起始(KB)        状态
6       50              55              空闲
7       10              130             空闲
8       137             159             空闲
********************可变分区管理********************
        *               1.内存分配        *
        *               2.内存去配        *
        *               0.退出            *
                        请输入选项[1]
********************分配算法********************
        *               1.最先分配法      *
        *               2.最优分配法      *
        *               3.最坏分配法      *
                        请输入选项[1]
```

```
********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       10              105             JOB_C
3       15              115             JOB_X
4       14              140             JOB_E
5       5               154             JOB_G
未分配:
分区号  大小(KB)        起始(KB)        状态
6       50              55              空闲
7       10              130             空闲
8       137             159             空闲
请输入作业名及其所需分配的主存大小（单位：KB）：JOB_F 32
分配成功!

********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       32              55              JOB_F
3       10              105             JOB_C
4       15              115             JOB_X
5       14              140             JOB_E
6       5               154             JOB_G
未分配:
分区号  大小(KB)        起始(KB)        状态
7       18              87              空闲
8       10              130             空闲
9       137             159             空闲
********************可变分区管理********************
        *               1.内存分配      *
        *               2.内存去配      *
        *               0.退出          *
                        请输入选项[2]

********************主存分配情况********************
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              40              JOB_A
2       32              55              JOB_F
3       10              105             JOB_C
4       15              115             JOB_X
5       14              140             JOB_E
6       5               154             JOB_G
未分配:
分区号  大小(KB)        起始(KB)        状态
7       18              87              空闲
8       10              130             空闲
9       137             159             空闲
```

```
请输入你要回收的作业名：JOB_C
回收成功！

********************主存分配情况********************
已分配：
分区号   大小(KB)          起始(KB)          状态
1        15                40                JOB_A
2        32                55                JOB_F
3        15                115               JOB_X
4        14                140               JOB_E
5        5                 154               JOB_G
未分配：
分区号   大小(KB)          起始(KB)          状态
6        28                87                空闲
7        10                130               空闲
8        137               159               空闲
********************可变分区管理********************
         *              1.内存分配          *
         *              2.内存去配          *
         *              0.退出              *
                        请输入选项[2]

********************主存分配情况********************
已分配：
分区号   大小(KB)          起始(KB)          状态
1        15                40                JOB_A
2        32                55                JOB_F
3        15                115               JOB_X
4        14                140               JOB_E
5        5                 154               JOB_G
未分配：
分区号   大小(KB)          起始(KB)          状态
6        28                87                空闲
7        10                130               空闲
8        137               159               空闲
请输入你要回收的作业名：JOB_A
回收成功！

********************主存分配情况********************
已分配：
分区号   大小(KB)          起始(KB)          状态
1        32                55                JOB_F
2        15                115               JOB_X
3        14                140               JOB_E
4        5                 154               JOB_G
未分配：
分区号   大小(KB)          起始(KB)          状态
5        15                40                空闲
6        28                87                空闲
7        10                130               空闲
8        137               159               空闲
```

```
**********可变分区管理**********
        *           1.内存分配          *
        *           2.内存去配          *
        *           0.退出              *
                    请输入选项[2]

**********主存分配情况**********
已分配:
分区号  大小(KB)        起始(KB)        状态
1       32              55              JOB_F
2       15              115             JOB_X
3       14              140             JOB_E
4       5               154             JOB_G
未分配:
分区号  大小(KB)        起始(KB)        状态
5       15              40              空闲
6       28              87              空闲
7       10              130             空闲
8       137             159             空闲
请输入你要回收的作业名: JOB_F
回收成功!

**********主存分配情况**********
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              115             JOB_X
2       14              140             JOB_E
3       5               154             JOB_G
未分配:
分区号  大小(KB)        起始(KB)        状态
4       75              40              空闲
5       10              130             空闲
6       137             159             空闲
**********可变分区管理**********
        *           1.内存分配          *
        *           2.内存去配          *
        *           0.退出              *
                    请输入选项[2]

**********主存分配情况**********
已分配:
分区号  大小(KB)        起始(KB)        状态
1       15              115             JOB_X
2       14              140             JOB_E
3       5               154             JOB_G
未分配:
分区号  大小(KB)        起始(KB)        状态
4       75              40              空闲
5       10              130             空闲
6       137             159             空闲
请输入你要回收的作业名: JOB_E
```

```
回收成功!

********************主存分配情况********************
已分配:
分区号   大小(KB)         起始(KB)         状态
1        15               115              JOB_X
2        5                154              JOB_G
未分配:
分区号   大小(KB)         起始(KB)         状态
3        75               40               空闲
4        24               130              空闲
5        137              159              空闲
********************可变分区管理********************
         *              1.内存分配          *
         *              2.内存去配          *
         *              0.退出              *
                        请输入选项[2]

********************主存分配情况********************
已分配:
分区号   大小(KB)         起始(KB)         状态
1        15               115              JOB_X
2        5                154              JOB_G
未分配:
分区号   大小(KB)         起始(KB)         状态
3        75               40               空闲
4        24               130              空闲
5        137              159              空闲
请输入你要回收的作业名: JOB_X
回收成功!

********************主存分配情况********************
已分配:
分区号   大小(KB)         起始(KB)         状态
1        5                154              JOB_G
未分配:
分区号   大小(KB)         起始(KB)         状态
2        114              40               空闲
3        137              159              空闲
********************可变分区管理********************
         *              1.内存分配          *
         *              2.内存去配          *
         *              0.退出              *
                        请输入选项[2]

********************主存分配情况********************
已分配:
分区号   大小(KB)         起始(KB)         状态
1        5                154              JOB_G
未分配:
分区号   大小(KB)         起始(KB)         状态
```

```
已分配：
分区号   大小(KB)          起始(KB)          状态
1        5                 154               JOB_G
未分配：
分区号   大小(KB)          起始(KB)          状态
2        114               40                空闲
3        137               159               空闲
请输入你要回收的作业名：JOB_G
回收成功！

********************主存分配情况********************
已分配：
分区号   大小(KB)          起始(KB)          状态
未分配：
分区号   大小(KB)          起始(KB)          状态
1        256               40                空闲
********************可变分区管理********************
         *             1.内存分配          *
         *             2.内存去配          *
         *             0.退出              *
                       请输入选项[0]
```

---

3. 磁盘移臂调度算法

本实验模拟磁盘移臂调度算法中的FCFS、SSTF和电梯调度算法，FCFS算法的主要思想是根据访问请求的先后次序选择先提出访问请求的为之服务，SSTF算法的主要思想是以磁头移动距离的大小作为优先的因素，从当前磁头位置出发，选择离磁头最近的磁道为其服务，电梯调度算法的主要思想是选请求队列中沿磁头臂前进方向最接近于磁头所在柱面的访问请求作为下一个服务对象。

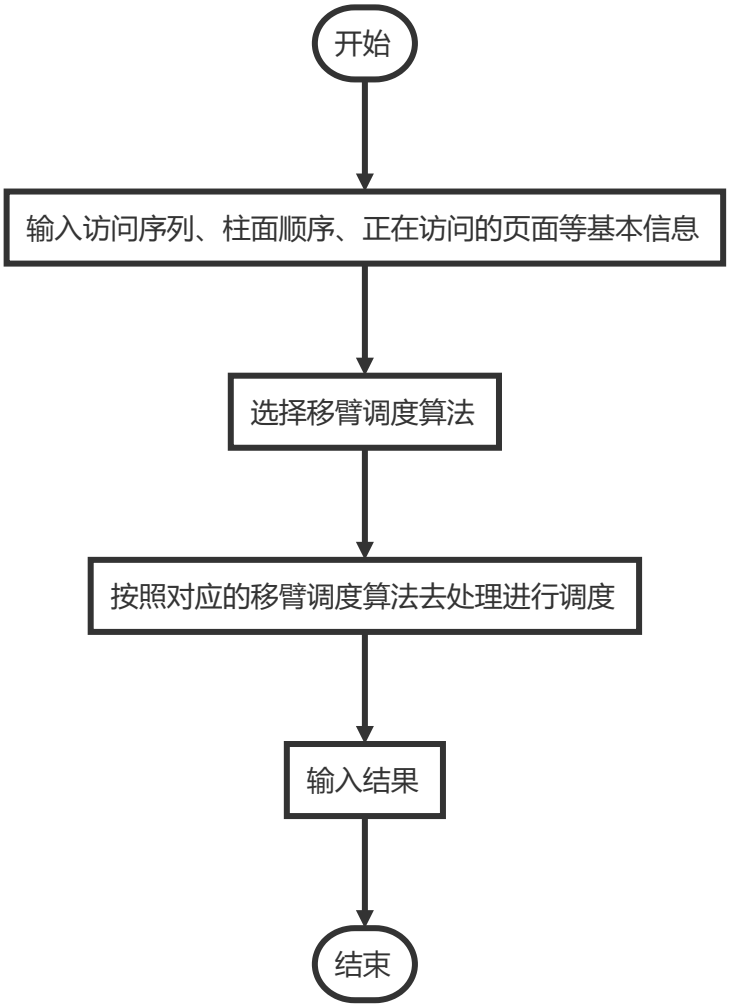
类成员变量：

```
self.size = None          # 访问序列长度
self.list = []            # 柱面顺序
self.now = None           # 正则访问的页面
```

函数包括：

- `Storage.build()`：输入基本信息
- `Storage.fcfs()`：FCFS算法
- `Storage.sstf()`：SSTF算法
- `Storage.elevator()`：电梯调度算法

程序流程图：



实验结果：

```
PS D:\学习资料\操作系统\python\5. 存储管理\modules> python3 .\ass.py
请输入访问序列的长度: 8
请输入访问的柱面顺序: 98 183 37 122 14 124 65 67
请输入正在访问的页面: 53
**********FCFS磁盘移臂调度过程**********
移动的顺序为:
[53, 98, 183, 37, 122, 14, 124, 65, 67]
移动柱面为: 640

**********SSTF磁盘移臂调度过程**********
移动的顺序为:
[53, 65, 67, 37, 14, 98, 122, 124, 183]
移动柱面为: 236

**********电梯磁盘移臂调度过程**********
由里向外移动的顺序为:
[53, 37, 14, 65, 67, 98, 122, 124, 183]
移动柱面为: 208
由外向里移动的顺序为:
[53, 65, 67, 98, 122, 124, 183, 37, 14]
移动柱面为: 299
```

4. 页面置换算法

本实验模拟的是页面置换算法，页面置换算法指的是当发生缺页中断时，如果操作系统内存中没有空闲页面，则操作系统必须在内存选择一个页面将其移出内存，以便为即将调入的页面让出空间，而用来选择淘汰哪一页的规则的算法。

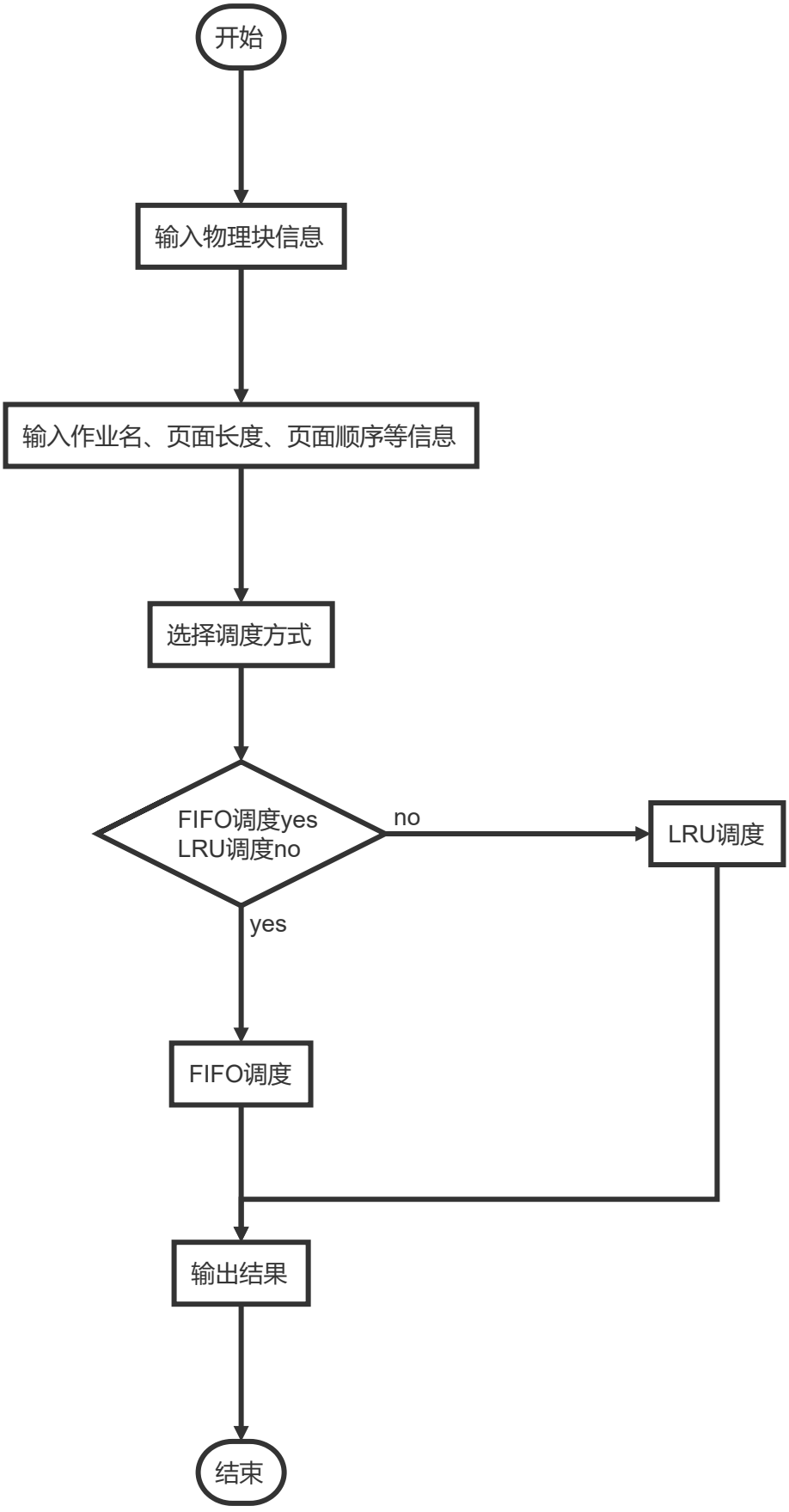
类成员变量：

```
self.block = None        # 物理块块数
self.job = None          # 作业名
self.len = None          # 作业页面长度
self.list = []           # 作业页面顺序
```

函数包括：

- `Storage.build()`：输入物理块块数
- `Storage.input()`：输入作业名、页面长度、页面顺序等信息
- `Storage.out()`：输出运行结果
- `Storage.fifo()`：FIFO调度
- `Storagelru()`：LRU调度

程序流程图：


开始
输入物理块信息
输入作业名、页面长度、页面顺序等信息
选择调度方式
FIFO调度yes
LRU调度no
no
LRU调度
yes
FIFO调度
输出结果
结束

实验结果：

```
PS D:\学习资料\操作系统\python\5. 存储管理\modules> python3 pr.py
请输入物理块的块数: 3
***************请求分页式存储管理***************
        *       1.FIFO分配      *
        *       2.LRU(LFU)分配  *
        *       0.退出          *
                请输入选项[1]
请输入作业名: job1
请输入作业页面的长度: 20
请输入作业页面的顺序: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
***************打印作业FIFO调度进入主存页的过程***************
作业名: job1
作业调度过程
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
     7  0  1  2  0  3  0  4  2  3  0  3  2  1  2  0  1  7  0  1
  0  7  0  1  2  2  3  0  4  2  3  0  0  0  1  2  2  2  7  0  1
  1     7  0  1  1  2  3  0  4  2  3  3  3  0  1  1  1  2  7  0
  2        7  0  0  1  2  3  0  4  2  2  2  3  0  0  0  1  2  7
     +  +  +  +     +  +  +  +  +  +        +  +        +  +  +
缺页中断率为: 75.0
***************请求分页式存储管理***************
        *       1.FIFO分配      *
        *       2.LRU(LFU)分配  *
        *       0.退出          *
                请输入选项[2]
请输入作业名: job2
请输入作业页面的长度: 20
请输入作业页面的顺序: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
***************打印作业FIFO调度进入主存页的过程***************
作业名: job2
作业调度过程
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
     7  0  1  2  0  3  0  4  2  3  0  3  2  1  2  0  1  7  0  1
  0  7  0  1  2  0  3  0  4  2  3  0  3  2  1  2  0  1  7  0  1
  1     7  0  1  2  0  3  0  4  2  3  0  3  2  1  2  0  1  7  0
  2        7  0  1  2  2  3  0  4  2  2  0  3  3  1  2  0  1  7
     +  +  +  +     +     +  +  +  +        +     +     +
缺页中断率为: 60.0
***************请求分页式存储管理***************
        *       1.FIFO分配      *
        *       2.LRU(LFU)分配  *
        *       0.退出          *
                请输入选项[0]
```

```
PS D:\学习资料\操作系统\python\5. 存储管理\modules> python3 pr.py
请输入物理块的块数: 4
***************请求分页式存储管理***************
        *       1.FIFO分配       *
        *       2.LRU(LFU)分配   *
        *       0.退出           *
                请输入选项[1]
请输入作业名: job3
请输入作业页面的长度: 20
请输入作业页面的顺序: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
***************打印作业FIFO调度进入主存页的过程***************
作业名: job3
作业调度过程
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
     7  0  1  2  0  3  0  4  2  3  0  3  2  1  2  0  1  7  0  1
  0  7  0  1  2  2  3  3  4  4  4  0  0  0  1  2  2  2  7  7  7
  1     7  0  1  1  2  2  3  3  3  4  4  4  0  1  1  1  2  2  2
  2        7  0  0  1  1  2  2  2  3  3  3  4  0  0  0  1  1  1
  3           7  7  0  0  1  1  1  2  2  2  3  4  4  4  0  0  0
     +  +  +  +     +     +        +        +  +        +
缺页中断率为: 50.0
***************请求分页式存储管理***************
        *       1.FIFO分配       *
        *       2.LRU(LFU)分配   *
        *       0.退出           *
                请输入选项[2]
请输入作业名: job4
请输入作业页面的长度: 20
请输入作业页面的顺序: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
***************打印作业FIFO调度进入主存页的过程***************
作业名: job4
作业调度过程
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
     7  0  1  2  0  3  0  4  2  3  0  3  2  1  2  0  1  7  0  1
  0  7  0  1  2  0  3  0  4  2  3  0  3  2  1  2  0  1  7  0  1
  1     7  0  1  2  0  3  0  4  2  3  0  3  2  1  2  0  1  7  0
  2        7  0  1  2  2  3  0  4  2  2  0  3  3  1  2  0  1  7
  3           7  7  1  1  2  3  0  4  4  4  0  0  3  3  2  2  2
     +  +  +  +     +     +                 +           +
缺页中断率为: 40.0
***************请求分页式存储管理***************
        *       1.FIFO分配       *
        *       2.LRU(LFU)分配   *
        *       0.退出           *
                请输入选项[0]
```

---

5. 分页、分段存储管理算法

本实验模拟存储管理中的分页、分段存储管理算法，分页存储管理算法的主要思想是内存被划分成大小固定相等的块，且块相对比较小，每个进程装入时被分成同样大小的页一页装入一帧，整个进程被离散装入到多个不连续的帧，分段存储管理的主要思想是把自己的作业按照逻辑关系划分为若干个段，一个进程的地址空间可以包含几个不同的段。

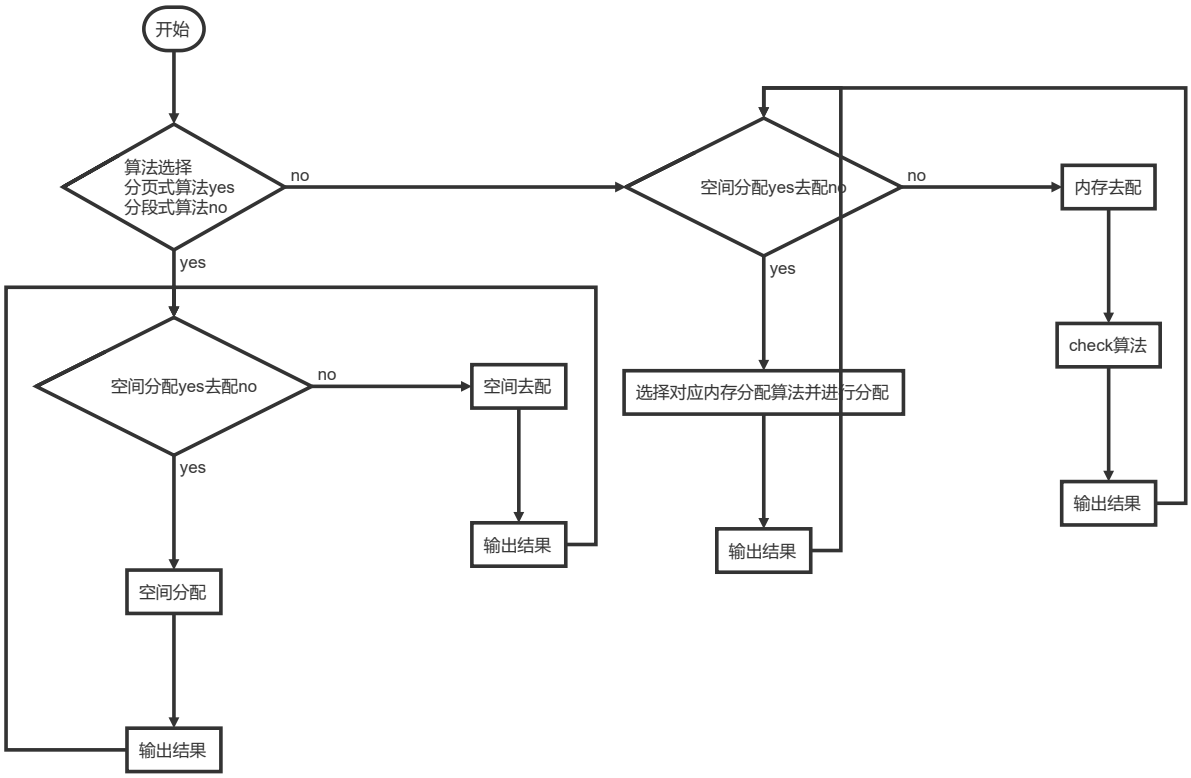
类成员变量:

```
# 段式
self.size = None              # 内存大小
self.start = None             # 起始地址大小
self.use = []                 # 已分配内存
self.free                     # 空闲内存
# 分页式
self.size = None              # 内存大小
self.wordlen = None           # 字长
self.blocklen = None          # 块长
self.a = []                   # 初始块信息
self.jobname = []             # 作业名
self.job = []                 # 已装入作业
```

函数包括：

- `Segment.main()`：分段存储管理算法入口
- `Segment.output()`：分段存储管理算法输出信息
- `Segment.check()`：分段存储管理回收check函数
- `Node.out()`：输出作业在辅存中的信息
- `Page.out()`：分页式算法输出信息
- `Page.distribute()`：分页式算法分配空间
- `Page.recyle()`：分页式算法回收作业
- `Page.main()`：分页式算法入口

程序流程图：


开始
算法选择
分页式算法yes
分段式算法no
no
yes
空间分配yes去配no
no
空间去配
输出结果
yes
空间分配
输出结果
空间分配yes去配no
no
内存去配
check算法
输出结果
yes
选择对应内存分配算法并进行分配
输出结果

实验结果：

```
PS D:\学习资料\操作系统\python\5. 存储管理\modules> python3 .\psss.py
***************模拟存储管理***************
        *       1.分页式算法    *
        *       2.分段式算法    *
        *       3.段页式算法    *
        *       0.退出          *
                请输入选项[1]
***************分页式管理模拟***************
请输入系统内存空间的大小（单位：K）和字长（32 or 64）和块长（单位：K）：2000 64 2
********************打印内存空间情况********************
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
  0  0  0  0  0  0  1  0  1  0  1  1  0  0  1  1  1  0  0  0  0  0  1  0  1  0  1  0  1  1  1  0  1  0  0  1  1  1  1  0  0  1  1  0  0  1  0  0  1  0  0  1
  1  1  1  1  1  1  0  1  1  1  0  1  0  0  0  1  0  1  0  0  1  0  1  0  0  0  1  1  0  0  0  1  1  1  1  1  0  0  1  1  0  1  1  1  0  0  0  0  1  0  1  1
  2  0  0  0  1  1  1  1  1  0  1  1  0  0  1  0  1  1  1  1  1  1  1  1  0  1  0  1  1  0  0  1  1  0  1  0  1  0  0  1  1  1  0  0  0  0  0  1  1  0  0  1
  3  1  1  1  1  1  1  0  1  1  0  0  1  0  0  0  0  0  1  1  0  1  1  0  0  0  0  0  1  1  1  0  1  0  1  1  0  0  1  1  0  0  1  0  1  1  1  0  1  1  1  0
  4  1  1  0  1  1  0  1  1  0  0  0  1  1  1  0  1  0  0  1  1  0  1  1  0  1  0  0  0  1  0  1  1  1  1  0  0  0  1  0  1  1  0  1  1  1  0  0  0  0  1  0
  5  0  0  0  0  1  0  0  1  1  1  1  1  1  1  0  0  0  1  0  0  0  0  0  1  0  0  1  0  1  1  0  1  1  1  1  0  1  1  0  0  0  0  1  0  1  0  1  0  1  1  0
  6  0  0  0  1  0  1  0  0  1  1  0  1  0  1  0  0  0  1  1  1  1  1  0  0  1  1  0  0  1  0  0  0  1  1  0  0  1  1  1  0  1  0  1  0  0  1  0  1  1  1  1
  7  0  1  1  0  1  0  1  1  0  1  0  0  1  1  1  1  1  1  1  1  1  1  0  0  1  0  1  0  0  0  0  0  1  1  0  0  1  1  1  0  1  0  1  0  0  1  0  1  1  1  1
  8  1  0  0  0  0  0  1  0  1  0  0  0  1  0  1  0  1  1  1  0  1  1  1  0  1  1  0  1  0  0  0  0  1  0  0  0  1  1  1  0  1  0  1  0  0  0  0  0  1  0  1
  9  0  0  0  0  0  1  1  1  1  1  0  1  0  1  1  0  1  1  1  1  1  1  1  0  1  1  1  0  1  1  0  0  0  1  0  0  0  1  0  0  1  0  1  0  1  1  1  1  0  1  1
 10  0  0  0  0  1  1  0  1  1  0  0  0  1  0  0  1  1  0  0  1  0  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  1  1  1  0  0  1  1  0  1  1  1  0  1  1  1  0
 11  1  0  0  0  0  0  0  0  1  0  1  1  0  1  0  0  1  0  0  0  0  1  0  1  1  1  0  1  1  1  0  0  1  0  0  1  0  1  1  1  1  1  1  1  1  0  0  0  1  1  1
 12  1  0  1  0  1  0  0  1  0  1  1  1  0  1  1  0  1  0  0  0  1  0  1  1  0  1  0  0  0  1  0  0  1  1  1  1  0  1  1  1  0  0  0  1  1  1  1  1  1  1  0
 13  0  1  1  1  0  1  0  0  1  0  1  1  0  0  0  0  1  0  0  1  1  0  0  0  0  1  0  1  1  1  0  0  0  1  1  0  1  0  1  1  1  0  0  0  1  1  1  0  1  1  1
 14  0  1  0  1  0  0  1  0  1  1  0  0  1  1  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  1  1  0  0  0  1  1  0  1  1  1  1  0  0  0  1  0  1  0  1  1
 15  1  0  0  0  0  1  1  1  0  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  0  1  1  1  0  1  1  1  0  1  1  1  1  1  1  1
剩余空块数：499
***************分页式管理模拟***************
        *       1.空间分配      *
        *       2.空间去配      *
        *       0.退出          *
                请输入选项[1]
********************打印内存空间情况********************
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
  0  0  0  0  0  0  1  0  1  0  1  0  1  1  0  0  1  1  1  0  0  0  0  0  1  0  1  0  1  0  1  1  1  1  0  1  0  0  1  1  1  1  0  0  1  1  0  1  0  0  0  0  0  1  1  0  0  0  1  0  0  1  0  1
  1  1  1  1  1  1  1  0  1  1  0  0  1  0  1  1  1  1  1  1  1  1  0  1  0  1  1  0  0  1  1  1  1  1  0  0  1  0  1  1  1  1  0  0  0  0  0  1  0  1  1  0  1  1  0  1  1  1
  2  0  0  0  1  1  1  1  0  1  1  0  0  1  0  1  1  1  1  1  1  1  1  0  1  0  1  1  0  0  1  1  0  1  0  1  0  0  1  1  1  1  0  1  0  0  0  1  0  0  0  0  0  1  1  0  0  0  1  0  0  1  1  1  1  0
  3  1  1  1  1  1  1  0  1  1  0  0  1  0  0  0  0  0  1  1  0  1  1  0  0  0  0  0  1  1  1  0  1  0  1  1  0  0  1  1  0  0  1  0  1  1  1  0  1  1  1  0
  4  1  1  0  1  1  0  1  1  0  0  0  1  1  1  0  1  0  0  1  1  0  1  1  0  1  0  0  0  1  0  1  1  1  1  0  0  0  1  0  1  1  0  1  1  1  0  0  0  0  1  0
  5  0  0  0  0  1  0  0  1  1  1  1  1  1  1  0  0  0  1  0  0  0  0  0  1  0  0  1  0  1  1  0  1  1  1  1  0  1  1  0  0  0  0  1  0  1  0  1  0  1  1  0
  6  0  0  0  1  0  1  0  0  1  1  0  1  0  1  0  0  0  1  1  1  1  1  0  0  1  1  0  0  1  0  0  0  1  1  0  0  1  1  1  0  1  0  1  0  0  1  0  1  1  1  1
  7  0  1  1  0  1  0  1  1  0  1  0  0  1  1  1  1  1  1  1  1  1  1  0  1  0  1  1  0  0  1  0  0  0  1  1  0  0  1  1  1  0  1  0  1  0  0  1  0  1  1  1
  8  1  0  0  0  0  0  1  0  1  0  0  0  1  0  1  0  1  1  1  0  1  1  1  0  1  1  0  1  0  0  0  0  1  0  0  0  1  1  1  0  1  0  1  0  0  0  0  0  1  0  1
  9  0  0  0  0  0  1  1  1  1  1  0  1  0  1  1  0  1  1  1  1  1  1  1  0  1  1  1  0  1  1  0  0  0  1  0  0  0  1  0  0  1  0  1  0  1  1  1  1  0  1  1
 10  0  0  0  0  1  1  0  1  1  0  0  0  1  0  0  1  1  0  0  1  0  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  1  1  1  0  0  1  1  0  1  1  1  0  1  1  1  0
 11  1  0  0  0  0  0  0  0  1  0  1  1  0  1  0  0  1  0  0  0  0  1  0  1  1  1  0  1  1  1  0  0  1  0  0  1  0  1  1  1  1  1  1  1  1  0  0  0  1  1  1
 12  1  0  1  0  1  0  0  1  0  1  1  1  0  1  1  0  1  0  0  0  1  0  1  1  0  1  0  0  0  1  0  0  1  1  1  1  0  1  1  1  0  0  0  1  1  1  1  1  1  1  0
 13  0  1  1  1  0  1  0  0  1  0  1  1  0  0  0  0  1  0  0  1  1  0  0  0  0  1  0  1  1  1  0  0  0  1  1  0  1  0  1  1  1  0  0  0  1  1  1  0  1  1  1
 14  0  1  0  1  0  0  1  0  1  1  0  0  1  1  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  1  1  0  0  0  1  1  0  1  1  1  1  0  0  0  1  0  1  0  1  1
 15  1  0  0  0  0  1  1  1  0  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  0  1  1  1  0  1  1  1  0  1  1  1  1  1  1  1
剩余空块数：499
请输入申请空间的作业名字和需要分配辅存空间的大小：job1 50
内存分配成功！
********************打印内存空间情况********************
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  1  1  1  1  1  0  0  1  0  1  1  1  1  0  0  0  0  0  1  0  1  1  0  1  0  0  1  1  0  1  1  0  1  1
  2  0  0  0  1  1  1  1  0  1  1  0  0  1  0  1  1  1  1  1  1  1  1  0  1  1  0  0  1  1  0  1  0  1  0  0  0  1  1  1  1  0  1  0  0  0  0  0  1  1  0  0  0  1  0  0  0  1  1  1  1  0
  3  1  1  1  1  1  1  0  1  1  0  0  1  0  0  0  0  0  1  1  0  1  1  0  0  0  0  0  1  1  1  0  1  0  1  1  0  0  1  1  0  0  1  0  1  1  1  0  1  1  1  0
  4  1  1  0  1  1  0  1  1  0  0  0  1  1  1  0  1  0  0  1  1  0  1  1  0  1  0  0  0  1  0  1  1  1  1  0  0  0  1  0  1  1  0  1  1  1  0  0  0  0  1  0
  5  0  0  0  0  1  0  0  1  1  1  1  1  1  1  0  0  0  1  0  0  0  0  0  1  0  0  1  0  1  1  0  1  1  1  1  0  1  1  0  0  0  0  1  0  1  0  1  0  1  1  0
  6  0  0  0  1  0  1  0  0  1  1  0  1  0  1  0  0  0  1  1  1  1  1  0  0  1  1  0  0  1  0  0  0  1  1  0  0  1  1  1  0  1  0  1  0  0  1  0  1  1  1  1
  7  0  1  1  0  1  0  1  1  0  1  0  0  1  1  1  1  1  1  1  1  1  1  0  1  0  1  1  0  0  1  0  0  0  1  1  0  0  1  1  1  0  1  0  1  0  0  1  0  1  1  1
  8  1  0  0  0  0  0  1  0  1  0  0  0  1  0  1  0  1  1  1  0  1  1  1  0  1  1  0  1  0  0  0  0  1  0  0  0  1  1  1  0  1  0  1  0  0  0  0  0  1  0  1
  9  0  0  0  0  0  1  1  1  1  1  0  1  0  1  1  0  1  1  1  1  1  1  1  0  1  1  1  0  1  1  0  0  0  1  0  0  0  1  0  0  1  0  1  0  1  1  1  1  0  1  1
 10  0  0  0  0  1  1  0  1  1  0  0  0  1  0  0  1  1  0  0  1  0  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  1  1  1  0  0  1  1  0  1  1  1  0  1  1  1  0
 11  1  0  0  0  0  0  0  0  1  0  1  1  0  1  0  0  1  0  0  0  0  1  0  1  1  1  0  1  1  1  0  0  1  0  0  1  0  1  1  1  1  1  1  1  1  0  0  0  1  1  1
 12  1  0  1  0  1  0  0  1  0  1  1  1  0  1  1  0  1  0  0  0  1  0  1  1  0  1  0  0  0  1  0  0  1  1  1  1  0  1  1  1  0  0  0  1  1  1  1  1  1  1  0
 13  0  1  1  1  0  1  0  0  1  0  1  1  0  0  0  0  1  0  0  1  1  0  0  0  0  1  0  1  1  1  0  0  0  1  1  0  1  0  1  1  1  0  0  0  1  1  1  0  1  1  1
 14  0  1  0  1  0  0  1  0  1  1  0  0  1  1  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  1  1  0  0  0  1  1  0  1  1  1  1  0  0  0  1  0  1  0  1  1
 15  1  0  0  0  0  1  1  1  0  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  0  1  1  1  0  1  1  1  0  1  1  1  1  1  1  1
剩余空块数：499
请输入申请空间的作业名字和需要分配辅存空间的大小：job2 80
内存分配成功！
```

```
********************打印内存空间情况********************
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 4  1  1  1  1  1  1  1  1  1  1  0  0  1  1  1  0  1  0  0  1  1  0  1  1  0  1  0  0  0  1  0  1  1  1  1  0  0  0  1  0  1  1  0  1  1  0  0  0  0  1  0  1  0  1  0  1  1  1  1  1  0  0  1  0  0
 5  0  0  0  0  1  0  0  1  1  1  1  1  1  1  0  0  0  1  0  0  0  0  1  0  0  0  1  0  1  0  0  0  1  1  1  0  1  1  1  1  1  0  1  1  0  0  1  0  0  0  0  1  0  1  0  1  0  1  1  1  1  0  0
 6  0  0  0  1  0  1  0  0  1  1  0  1  0  1  0  0  1  1  1  1  1  0  0  1  1  0  0  1  0  0  0  1  1  0  0  1  1  1  0  1  0  1  0  0  1  0  1  1  1  1  0  1  0  0  1  0  0  0  0  0  1  0  1  0  1
 7  0  1  1  0  1  0  1  1  0  1  0  0  1  1  1  1  1  1  1  1  1  0  1  0  1  0  0  1  0  0  0  0  0  1  1  1  0  1  0  0  0  1  1  0  1  1  1  1  1  0  0  1  0  0  1  0  1  1  1  0  1  0  1  0  1
 8  1  0  0  0  0  0  1  0  1  0  0  0  1  0  1  0  1  1  0  1  1  0  1  1  1  1  0  1  1  0  0  1  1  1  1  0  0  1  1  0  1  0  1  1  1  1  0  0  0  1  1  1  0  0  0  0  0  1  0  0  0  1  0  1  1  1  1
 9  0  0  0  0  0  1  1  1  1  1  1  0  1  0  1  1  0  1  0  1  1  1  1  0  1  1  1  0  0  0  1  0  0  0  1  0  0  0  1  0  0  1  0  0  0  1  0  0  0  1  0  1  1  1  1  0  0  1  0  1  1  0  0  1
10  0  0  0  0  0  1  1  1  0  0  0  1  0  0  1  1  0  0  1  0  0  0  1  1  0  1  1  0  0  0  0  0  0  0  0  0  0  1  1  1  0  0  1  0  1  1  1  1  0  1  1  0  1  0  1  1  0  0  0  1  1  0
11  1  0  0  0  0  0  0  0  0  1  0  1  1  0  1  0  1  0  0  0  0  1  0  1  1  1  1  0  1  1  0  0  0  1  1  0  1  1  1  1  0  1  0  1  1  1  0  0  1  0  1  0  1  0  1  1  1  1  1  1  1  1  0  0  1  1  1  1
12  1  0  1  0  1  0  0  1  0  1  1  1  0  1  1  0  1  0  0  1  0  1  1  1  1  0  1  0  0  0  1  0  0  1  1  1  0  0  0  1  1  1  1  0  1  1  1  0  0  0  1  1  1  1  0  1  1  1  0  0  0  0  0  1  0
13  0  1  1  1  0  1  0  0  1  0  1  0  0  0  0  0  1  0  0  1  1  0  0  0  0  1  0  1  1  1  0  0  1  1  0  1  0  1  1  0  1  1  1  1  0  0  1  1  1  1  0  0  0  1  0  1  0  0  0  1  1  1  0  1
14  0  1  0  1  0  0  1  0  1  1  0  0  1  1  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  1  1  0  0  0  1  1  0  0  0  1  1  1  1  0  1  1  0  0  0  0  0  1  1  0  0  1  0  1  0  1  1
15  1  0  0  0  0  1  1  1  0  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  0  1  1  1  0  1  1  1  0  1  1  1  1  1  1  1
剩余空块数: 369
**************打印job2作业在辅存中的信息**************
  记录      块号
    1       92
    2       93
    3       99
    4      100
    5      102
    6      107
    7      108
    8      109
    9      110
   10      111
   11      113
   12      116
   13      118
   14      119
   15      122
   16      125
   17      128
   18      129
   19      130
   20      135
   21      138
   22      139
   23      141
   24      149
   25      151
   26      154
   27      155
   28      158
   29      160


   30      162
   31      163
   32      168
   33      170
   34      171
   35      172
   36      174
   37      175
   38      176
   39      177
   40      178
   41      181
   42      182
   43      183
   44      185
   45      186
   46      191
   47      197
   48      200
   49      201
   50      203
   51      204
   52      205
   53      206
   54      207
   55      210
   56      213
   57      214
   58      215
   59      216
   60      217
   61      221
   62      223
   63      226
   64      227
   65      230
   66      231
   67      233
   68      235
   69      237
   70      240
   71      241
   72      245
   73      246
   74      247
   75      248
   76      253
   77      255
   78      258
   79      261
   80      264
```

5. 磁盘管理

1. 位示图算法

本实验模拟的是磁盘管理中辅存空间位示图算法的实现过程，包括内存分配以及内存去配的模拟。

类成员变量：

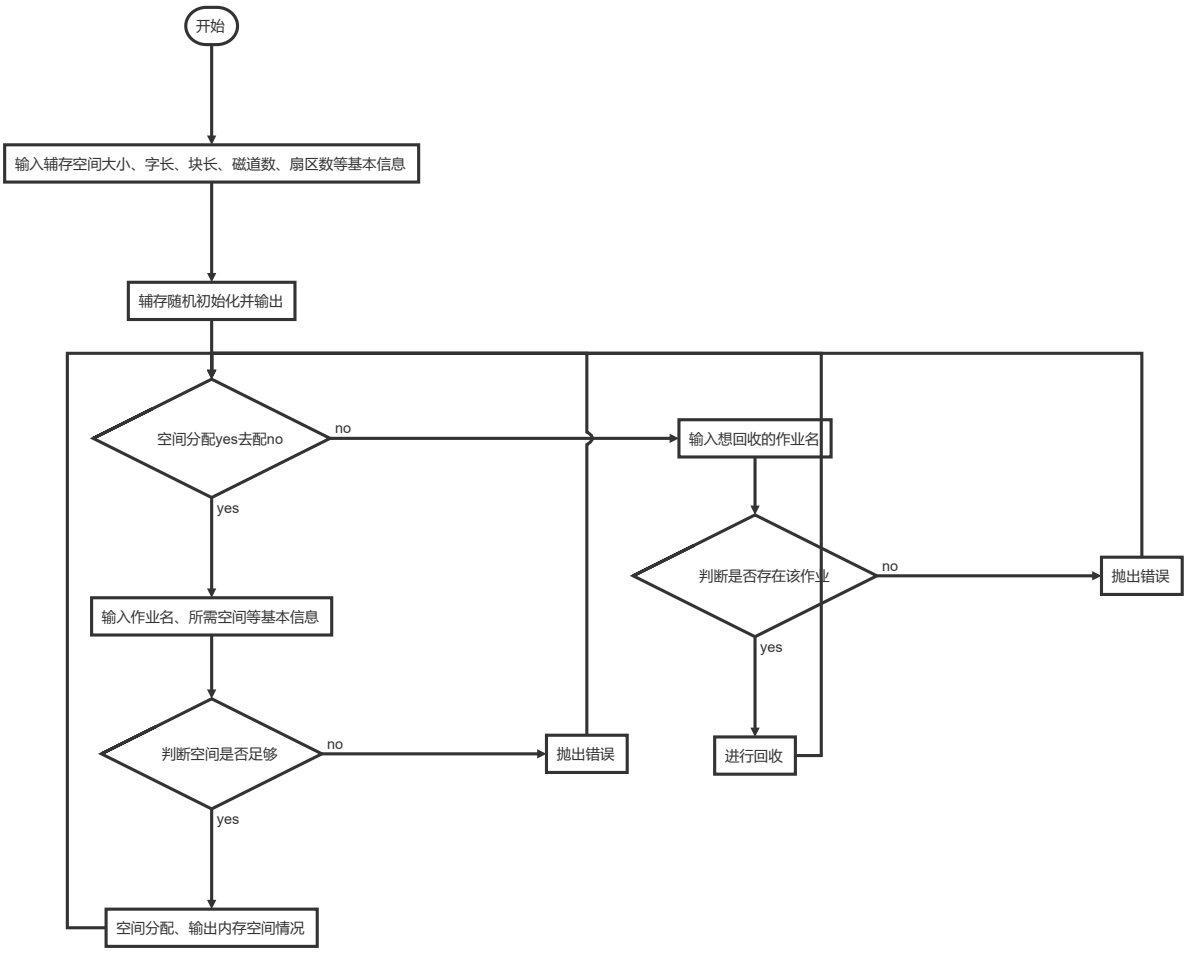
```
# Node
self.name = name            # 作业名
self.size = size            # 作业大小
self.a = []                 # 块号
self.zihao = []             # 字号
self.weihao = []            # 位号
self.zhu = []               # 柱面号
```

```
self.citou = []              # 磁头号
self.shanqu = []             # 扇区号
# Disk
self.size = size             # 磁盘大小
self.wordlength = wordlen    # 字长
self.blocklen = blocklen     # 块长
self.tracksum = tracksum     # 磁道数
self.sectorsum = sectorsum   # 扇区数
self.jobname = []            # 作业名列表
self.job = []                # 作业列表
self.a = []                  # 初始块信息
```

函数包括：

- `Node.out()`：输出作业在辅存中的信息
- `Disk.out()`：输出磁盘信息
- `Disk.distribute()`：空间分配
- `Disk.recycle()`：空间去配

程序流程图：


开始
输入辅存空间大小、字长、块长、磁道数、扇区数等基本信息
辅存随机初始化并输出
空间分配yes去配no
no
yes
输入作业名、所需空间等基本信息
判断空间是否足够
no
yes
抛出错误
空间分配、输出内存空间情况
输入想回收的作业名
判断是否存在该作业
no
yes
抛出错误
进行回收

实验结果：

```
PS D:\学习资料\操作系统\python\6. 磁盘管理> python3 .\bitmap.py
请输入辅存空间的大小（单位：K）和字长（32 or 64）和块长（单位：K）：1000 64 1
请输入该辅存硬盘的磁道数（磁头数）、每磁道的扇区数：8 96
********************辅存初始位示图如下********************
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
  0  0  0  1  1  1  0  1  0  1  1  1  1  1  0  0  0  1  0  1  1  1  1  0  1  0  0  1  0  1  1  0  0  0  1  0  0  0  1  0  1  0  0  0  0  0  1  1  1  0  0  1  0  0  0  1  1  1  0  1  1  1  1  1  1
  1  1  1  1  0  0  0  0  1  1  0  1  1  1  1  1  0  1  1  0  1  1  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  1  1  1  1  1  1  1  0  1  1  0  1  0  0  1  0  0  1  1  1  0  0  1  0  0  1  1  1
  2  1  1  1  1  1  0  1  0  0  0  1  1  1  0  1  0  1  1  0  1  0  1  0  0  0  1  1  0  0  1  0  0  0  1  1  1  0  1  0  1  1  1  0  1  1  0  0  0  1  0  0  0  0  1  1  0  0  0  0  0  0  1  1  0
  3  0  1  0  0  1  0  0  0  0  1  1  1  1  1  0  0  1  1  0  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  0  1  1  1  0  0  0  0  1  0  1  0  0  0  1  1  1  0  0  1  1  0  1  0
  4  1  0  1  0  0  1  0  0  1  1  1  0  0  1  1  0  0  1  1  1  0  1  1  0  1  1  1  0  1  1  0  0  0  1  1  0  1  0  0  1  1  0  0  1  0  1  0  1  1  0  1  1  1  1  0  1  0  1  1  1  1  1  1  1
  5  0  1  0  0  0  0  0  0  1  0  1  1  0  0  1  0  1  1  1  1  1  0  1  0  0  1  1  1  1  1  1  0  1  1  0  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  0  1  1  0
  6  1  1  0  1  1  1  1  0  0  0  0  1  0  1  0  1  1  0  1  0  0  0  0  1  1  0  1  1  1  1  1  0  0  1  0  1  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  1  0  0  0  0  0  0  1  1  1  1  1  0
  7  0  0  1  0  1  0  0  0  1  1  0  0  0  1  0  0  1  0  0  0  1  0  1  0  1  1  0  0  1  0  1  1  0  0  1  1  0  0  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  0  1  0  0  1  1  1  1  0  1  0
  8  0  0  0  0  1  1  1  0  1  1  1  0  0  0  1  0  0  1  0  1  0  1  1  1  0  0  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  0  1  0  0  1  1  1  1  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0
  9  0  1  0  1  0  1  1  0  0  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  1  0  1  1  1  0  1  0  0  1  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  1  1  0  1  0  1  0  0  1  0  0  0  0
 10  1  0  1  1  0  0  1  0  0  1  1  1  0  1  0  0  0  1  0  1  1  1  0  1  1  1  1  1  1  1  1  1  1  0  1  1  1  0  1  0  0  0  0  1  1  0  0  1  0  0  0  0  0  1  0  0  0  1  0  1  0  1  1  1
 11  0  0  0  1  1  0  0  0  1  1  1  1  0  0  1  1  1  0  1  1  0  1  0  0  1  1  0  1  0  0  0  1  1  0  1  0  0  0  1  1  0  1  0  1  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  1  1  1  0
 12  0  0  0  0  1  1  1  1  0  0  1  1  0  0  1  1  0  0  0  1  0  1  1  1  1  1  0  0  0  0  1  1  1  1  1  0  1  1  0  1  1  0  0  1  1  1  1  0  1  1  0  0  1  1  0  0  1  1  0  0  1  1  0  0
 13  1  0  1  1  0  1  0  0  1  0  1  1  1  0  1  0  1  0  1  0  1  1  1  1  0  1  0  0  0  1  1  0  1  0  0  1  0  0  0  1  1  1  0  1  1  0  1  0  1  0  0  0  0  0  1  0  0  1  0  0  1  1  1  0
 14  0  1  1  1  1  0  0  0  1  0  1  1  0  0  1  0  1  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  0  0  1  1  0  1  1  0  0  1  1  1  0  1  0  1  1  0  1  1  1  1  1  0  0  1  0  1  1  0
 15  0  0  0  0  0  1  0  0  1  1  1  1  1  0  1  0  1  0  0  0  0  0  0  1  1  0  0  1  0  0  1  0  0  0  1  1  0  1  0  0
辅存剩余空块数：496
***************辅存管理***************
        *       1.空间分配       *
        *       2.空间去配       *
        *       0.退出           *
                请输入选项[1]
********************打印内存空间情况********************
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
  0  0  0  1  1  1  0  1  0  1  1  1  1  1  0  0  0  1  0  1  1  1  1  0  1  0  0  1  0  1  1  0  0  0  1  0  0  0  1  0  1  0  0  0  0  0  1  1  1  0  0  1  0  0  0  1  1  1  0  1  1  1  1  1  1
  1  1  1  1  0  0  0  0  1  1  0  1  1  1  1  1  0  1  1  0  1  1  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  1  1  1  1  1  1  1  0  1  1  0  1  0  0  1  0  0  1  1  1  0  0  1  0  0  1  1  1
  2  1  1  1  1  1  0  1  0  0  0  1  1  1  0  1  0  1  1  0  1  0  1  0  0  0  1  1  0  0  1  0  0  0  1  1  1  0  1  0  1  1  1  0  1  1  0  0  0  1  0  0  0  0  1  1  0  0  0  0  0  0  1  1  0
  3  0  1  0  0  1  0  0  0  0  1  1  1  1  1  0  0  1  1  0  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  0  1  1  1  0  0  0  0  1  0  1  0  0  0  1  1  1  0  0  1  1  0  1  0
  4  1  0  1  0  0  1  0  0  1  1  1  0  0  1  1  0  0  1  1  1  0  1  1  0  1  1  1  0  1  1  0  0  0  1  1  0  1  0  0  1  1  0  0  1  0  1  0  1  1  0  1  1  1  1  0  1  0  1  1  1  1  1  1  1
  5  0  1  0  0  0  0  0  0  1  0  1  1  0  0  1  0  1  1  1  1  1  0  1  0  0  1  1  1  1  1  1  0  1  1  0  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  0  1  1  0
  6  1  1  0  1  1  1  1  0  0  0  0  1  0  1  0  1  1  0  1  0  0  0  0  1  1  0  1  1  1  1  1  0  0  1  0  1  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  1  0  0  0  0  0  0  1  1  1  1  1  0
  7  0  0  1  0  1  0  0  0  1  1  0  0  0  1  0  0  1  0  0  0  1  0  1  0  1  1  0  0  1  0  1  1  0  0  1  1  0  0  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  0  1  0  0  1  1  1  1  0  1  0
  8  0  0  0  0  1  1  1  0  1  1  1  0  0  0  1  0  0  1  0  1  0  1  1  1  0  0  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  0  1  0  0  1  1  1  1  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0
  9  0  1  0  1  0  1  1  0  0  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  1  0  1  1  1  0  1  0  0  1  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  1  1  0  1  0  1  0  0  1  0  0  0  0
 10  1  0  1  1  0  0  1  0  0  1  1  1  0  1  0  0  0  1  0  1  1  1  0  1  1  1  1  1  1  1  1  1  1  0  1  1  1  0  1  0  0  0  0  1  1  0  0  1  0  0  0  0  0  1  0  0  0  1  0  1  0  1  1  1
 11  0  0  0  1  1  0  0  0  1  1  1  1  0  0  1  1  1  0  1  1  0  1  0  0  1  1  0  1  0  0  0  1  1  0  1  0  0  0  1  1  0  1  0  1  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  1  1  1  0
 12  0  0  0  0  1  1  1  1  0  0  1  1  0  0  1  1  0  0  0  1  0  1  1  1  1  1  0  0  0  0  1  1  1  1  1  0  1  1  0  1  1  0  0  1  1  1  1  0  1  1  0  0  1  1  0  0  1  1  0  0  1  1  0  0
 13  1  0  1  1  0  1  0  0  1  0  1  1  1  0  1  0  1  0  1  0  1  1  1  1  0  1  0  0  0  1  1  0  1  0  0  1  0  0  0  1  1  1  0  1  1  0  1  0  1  0  0  0  0  0  1  0  0  1  0  0  1  1  1  0
 14  0  1  1  1  1  0  0  0  1  0  1  1  0  0  1  0  1  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  0  0  1  1  0  1  1  0  0  1  1  1  0  1  0  1  1  0  1  1  1  1  1  0  0  1  0  1  1  0
 15  0  0  0  0  0  1  0  0  1  1  1  1  1  0  1  0  1  0  0  0  0  0  0  1  1  0  0  1  0  0  1  0  0  0  1  1  0  1  0  0
辅存剩余空块数：496
请输入申请空间的作业名字和需要分配辅存空间的大小：job1 50
内存分配成功！


********************打印内存空间情况********************
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  1  0  0  1  1  1
  2  1  1  1  1  1  0  1  0  0  0  1  1  1  0  1  0  1  1  0  1  0  1  0  0  0  1  1  0  0  1  0  0  0  1  1  1  0  1  0  1  1  1  0  1  1  0  0  0  1  0  0  0  0  1  1  0  0  0  0  0  0  1  1  0
  3  0  1  0  0  1  0  0  0  0  1  1  1  1  1  0  0  1  1  0  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  0  1  1  1  0  0  0  0  1  0  1  0  0  0  1  1  1  0  0  1  1  0  1  0
  4  1  0  1  0  0  1  0  0  1  1  1  0  0  1  1  0  0  1  1  1  0  1  1  0  1  1  1  0  1  1  0  0  0  1  1  0  1  0  0  1  1  0  0  1  0  1  0  1  1  0  1  1  1  1  0  1  0  1  1  1  1  1  1  1
  5  0  1  0  0  0  0  0  0  1  0  1  1  0  0  1  0  1  1  1  1  1  0  1  0  0  1  1  1  1  1  1  0  1  1  0  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  0  1  1  0
  6  1  1  0  1  1  1  1  0  0  0  0  1  0  1  0  1  1  0  1  0  0  0  0  1  1  0  1  1  1  1  1  0  0  1  0  1  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  1  0  0  0  0  0  0  1  1  1  1  1  0
  7  0  0  1  0  1  0  0  0  1  1  0  0  0  1  0  0  1  0  0  0  1  0  1  0  1  1  0  0  1  0  1  1  0  0  1  1  0  0  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  0  1  0  0  1  1  1  1  0  1  0
  8  0  0  0  0  1  1  1  0  1  1  1  0  0  0  1  0  0  1  0  1  0  1  1  1  0  0  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  0  1  0  0  1  1  1  1  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0
  9  0  1  0  1  0  1  1  0  0  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  1  0  1  1  1  0  1  0  0  1  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  1  1  0  1  0  1  0  0  1  0  0  0  0
 10  1  0  1  1  0  0  1  0  0  1  1  1  0  1  0  0  0  1  0  1  1  1  0  1  1  1  1  1  1  1  1  1  1  0  1  1  1  0  1  0  0  0  0  1  1  0  0  1  0  0  0  0  0  1  0  0  0  1  0  1  0  1  1  1
 11  0  0  0  1  1  0  0  0  1  1  1  1  0  0  1  1  1  0  1  1  0  1  0  0  1  1  0  1  0  0  0  1  1  0  1  0  0  0  1  1  0  1  0  1  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  1  1  1  0
 12  0  0  0  0  1  1  1  1  0  0  1  1  0  0  1  1  0  0  0  1  0  1  1  1  1  1  0  0  0  0  1  1  1  1  1  0  1  1  0  1  1  0  0  1  1  1  1  0  1  1  0  0  1  1  0  0  1  1  0  0  1  1  0  0
 13  1  0  1  1  0  1  0  0  1  0  1  1  1  0  1  0  1  0  1  0  1  1  1  1  0  1  0  0  0  1  1  0  1  0  0  1  0  0  0  1  1  1  0  1  1  0  1  0  1  0  0  0  0  0  1  0  0  1  0  0  1  1  1  0
 14  0  1  1  1  1  0  0  0  1  0  1  1  0  0  1  0  1  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  0  0  1  1  0  1  1  0  0  1  1  1  0  1  0  1  1  0  1  1  1  1  1  0  0  1  0  1  1  0
 15  0  0  0  0  0  1  0  0  1  1  1  1  1  0  1  0  1  0  0  0  0  0  0  1  1  0  0  1  0  0  1  0  0  0  1  1  0  1  0  0
辅存剩余空块数：446
***************打印job1作业在辅存中的信息***************
  记录      块号      柱面号      磁头号      扇区号
   1         0         0          0          0
   2         1         0          0          1
   3         5         0          0          5
   4         7         0          0          7
   5        13         0          0         13
   6        14         0          0         14
   7        15         0          0         15
   8        17         0          0         17
   9        22         0          0         22
  10        24         0          0         24
  11        25         0          0         25
  12        27         0          0         27
  13        30         0          0         30
  14        31         0          0         31
  15        32         0          0         32
  16        34         0          0         34
  17        35         0          0         35
  18        36         0          0         36
  19        38         0          0         38
  20        40         0          0         40
  21        41         0          0         41
  22        42         0          0         42
  23        43         0          0         43
  24        44         0          0         44
  25        48         0          0         48
  26        49         0          0         49
  27        51         0          0         51
  28        52         0          0         52
  29        53         0          0         53
  30        57         0          0         57


  31        67         0          0         67
  32        68         0          0         68
  33        69         0          0         69
  34        70         0          0         70
  35        73         0          0         73
  36        79         0          0         79
  37        82         0          0         82
  38        86         0          0         86
  39        89         0          0         89
  40        91         0          0         91
  41        93         0          0         93
  42        98         0          1          2
  43        99         0          1          3
  44       107         0          1         11
  45       110         0          1         14
  46       112         0          1         16
  47       113         0          1         17
  48       115         0          1         19
  49       116         0          1         20
  50       120         0          1         24
***************辅存管理***************
        *       1.空间分配       *
        *       2.空间去配       *
        *       0.退出           *
                请输入选项[1]
********************打印内存空间情况********************
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  1  0  0  1  1  1
  2  1  1  1  1  1  0  1  0  0  0  1  1  1  0  1  0  1  1  0  1  0  1  0  0  0  1  1  0  0  1  0  0  0  1  1  1  0  1  0  1  1  1  0  1  1  0  0  0  1  0  0  0  0  1  1  0  0  0  0  0  0  1  1  0
  3  0  1  0  0  1  0  0  0  0  1  1  1  1  1  0  0  1  1  0  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  0  1  1  1  0  0  0  0  1  0  1  0  0  0  1  1  1  0  0  1  1  0  1  0
  4  1  0  1  0  0  1  0  0  1  1  1  0  0  1  1  0  0  1  1  1  0  1  1  0  1  1  1  0  1  1  0  0  0  1  1  0  1  0  0  1  1  0  0  1  0  1  0  1  1  0  1  1  1  1  0  1  0  1  1  1  1  1  1  1
  5  0  1  0  0  0  0  0  0  1  0  1  1  0  0  1  0  1  1  1  1  1  0  1  0  0  1  1  1  1  1  1  0  1  1  0  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  0  1  1  0
  6  1  1  0  1  1  1  1  0  0  0  0  1  0  1  0  1  1  0  1  0  0  0  0  1  1  0  1  1  1  1  1  0  0  1  0  1  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  1  0  0  0  0  0  0  1  1  1  1  1  0
  7  0  0  1  0  1  0  0  0  1  1  0  0  0  1  0  0  1  0  0  0  1  0  1  0  1  1  0  0  1  0  1  1  0  0  1  1  0  0  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  0  1  0  0  1  1  1  1  0  1  0
  8  0  0  0  0  1  1  1  0  1  1  1  0  0  0  1  0  0  1  0  1  0  1  1  1  0  0  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  0  1  0  0  1  1  1  1  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0
  9  0  1  0  1  0  1  1  0  0  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  1  0  1  1  1  0  1  0  0  1  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  1  1  0  1  0  1  0  0  1  0  0  0  0
 10  1  0  1  1  0  0  1  0  0  1  1  1  0  1  0  0  0  1  0  1  1  1  0  1  1  1  1  1  1  1  1  1  1  0  1  1  1  0  1  0  0  0  0  1  1  0  0  1  0  0  0  0  0  1  0  0  0  1  0  1  0  1  1  1
 11  0  0  0  1  1  0  0  0  1  1  1  1  0  0  1  1  1  0  1  1  0  1  0  0  1  1  0  1  0  0  0  1  1  0  1  0  0  0  1  1  0  1  0  1  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  1  1  1  0
 12  0  0  0  0  1  1  1  1  0  0  1  1  0  0  1  1  0  0  0  1  0  1  1  1  1  1  0  0  0  0  1  1  1  1  1  0  1  1  0  1  1  0  0  1  1  1  1  0  1  1  0  0  1  1  0  0  1  1  0  0  1  1  0  0
 13  1  0  1  1  0  1  0  0  1  0  1  1  1  0  1  0  1  0  1  0  1  1  1  1  0  1  0  0  0  1  1  0  1  0  0  1  0  0  0  1  1  1  0  1  1  0  1  0  1  0  0  0  0  0  1  0  0  1  0  0  1  1  1  0
 14  0  1  1  1  1  0  0  0  1  0  1  1  0  0  1  0  1  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  0  0  1  1  0  1  1  0  0  1  1  1  0  1  0  1  1  0  1  1  1  1  1  0  0  1  0  1  1  0
 15  0  0  0  0  0  1  0  0  1  1  1  1  1  0  1  0  1  0  0  0  0  0  0  1  1  0  0  1  0  0  1  0  0  0  1  1  0  1  0  0
辅存剩余空块数：446
请输入申请空间的作业名字和需要分配辅存空间的大小：job2 80
内存分配成功！
```

```
********************打印内存空间情况********************
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 4  1  1  1  1  1  1  1  1  1  1  1  1  0  1  1  0  0  1  1  1  0  1  1  0  1  1  1  0  1  1  0  0  0  1  1  0  1  0  0  1  1  0  0  1  0  1  0  1  1  0  1  1  1  1  0  1  0  1  1  1  1  1  1  1
 5  0  1  0  0  0  0  0  0  1  0  1  1  0  0  1  0  1  1  1  1  1  0  1  0  0  1  1  1  1  1  1  0  1  1  0  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  0  1  1  0
 6  1  1  0  1  1  1  1  0  0  0  0  1  0  1  0  1  1  0  1  0  0  0  0  1  1  0  1  1  1  1  1  0  0  1  0  1  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  1  0  0  0  0  0  0  1  1  1  1  1  0
 7  0  0  1  0  1  0  0  0  1  1  0  0  0  1  0  0  1  0  0  0  1  0  1  0  1  1  0  0  1  0  1  1  0  0  1  1  0  0  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  0  1  0  0  1  1  1  1  0  1  0
 8  0  0  0  0  1  1  1  0  1  1  1  0  0  0  1  0  0  1  0  1  0  1  1  1  0  0  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  0  1  0  0  1  1  1  1  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0
 9  0  1  0  1  0  1  1  0  0  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  1  0  1  1  1  0  1  0  0  1  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  1  1  0  1  0  1  0  0  1  0  0  0  0
10  1  0  1  1  0  0  1  0  0  1  1  1  0  1  0  0  0  1  0  1  1  1  0  1  1  1  1  1  1  1  1  1  1  0  1  1  1  0  1  0  0  0  0  1  1  0  0  1  0  0  0  0  0  1  0  0  0  1  0  1  0  1  1  1
11  0  0  0  1  1  0  0  0  1  1  1  1  0  0  1  1  1  0  1  1  0  1  0  0  1  1  0  1  0  0  0  1  1  0  1  0  0  0  1  1  0  1  0  1  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  1  1  1  0
12  0  0  0  0  1  1  1  1  0  0  1  1  0  0  1  1  0  0  0  1  0  1  1  1  1  1  0  0  0  0  1  1  1  1  1  0  1  1  0  1  1  0  0  1  1  1  1  0  1  1  0  0  1  1  0  0  1  1  0  0  1  1  0  0
13  1  0  1  1  0  1  0  0  1  0  1  1  1  0  1  0  1  0  1  0  1  1  1  1  0  1  0  0  0  1  1  0  1  0  0  1  0  0  0  1  1  1  0  1  1  0  1  0  1  0  0  0  0  0  1  0  0  1  0  0  1  1  1  0
14  0  1  1  1  1  0  0  0  1  0  1  1  0  0  1  0  1  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  0  0  1  1  0  1  1  0  0  1  1  1  0  1  0  1  1  0  1  1  1  1  1  0  0  1  0  1  1  0
15  0  0  0  0  0  1  0  0  1  1  1  1  1  0  1  0  1  0  0  0  0  0  0  1  1  0  0  1  0  0  1  0  0  0  1  1  0  1  0  0
辅存剩余空块数: 366
***************打印job2作业在辅存中的信息***************
  记录      块号      柱面号      磁头号      扇区号
   1        121        0          1          25
   2        123        0          1          27
   3        124        0          1          28
   4        133        0          1          37
   5        135        0          1          39
   6        136        0          1          40
   7        137        0          1          41
   8        141        0          1          45
   9        143        0          1          47
  10        146        0          1          50
  11        148        0          1          52
  12        150        0          1          54
  13        151        0          1          55
  14        152        0          1          56
  15        155        0          1          59
  16        156        0          1          60
  17        158        0          1          62
  18        159        0          1          63
  19        160        0          1          64
  20        164        0          1          68
  21        166        0          1          70
  22        170        0          1          74
  23        173        0          1          77
  24        174        0          1          78
  25        175        0          1          79
  26        177        0          1          81
  27        178        0          1          82
  28        179        0          1          83
  29        180        0          1          84

  30        183        0          1          87
  31        184        0          1          88
  32        185        0          1          89
  33        186        0          1          90
  34        187        0          1          91
  35        188        0          1          92
  36        191        0          1          95
  37        192        0          2           0
  38        194        0          2           2
  39        195        0          2           3
  40        197        0          2           5
  41        198        0          2           6
  42        199        0          2           7
  43        200        0          2           8
  44        206        0          2          14
  45        207        0          2          15
  46        210        0          2          18
  47        211        0          2          19
  48        212        0          2          20
  49        214        0          2          22
  50        217        0          2          25
  51        218        0          2          26
  52        219        0          2          27
  53        220        0          2          28
  54        221        0          2          29
  55        222        0          2          30
  56        223        0          2          31
  57        224        0          2          32
  58        225        0          2          33
  59        226        0          2          34
  60        227        0          2          35
  61        228        0          2          36
  62        232        0          2          40
  63        236        0          2          44
  64        237        0          2          45
  65        238        0          2          46
  66        239        0          2          47
  67        241        0          2          49
  68        243        0          2          51
  69        244        0          2          52
  70        245        0          2          53
  71        249        0          2          57
  72        250        0          2          58
  73        253        0          2          61
  74        255        0          2          63
  75        257        0          2          65
  76        259        0          2          67
  77        260        0          2          68
  78        262        0          2          70
  79        263        0          2          71
  80        267        0          2          75


***************辅存管理***************
        *       1.空间分配       *
        *       2.空间去配       *
        *       0.退出           *
                请输入选项[2]
当前分配的作业: job1->job2
请输入你当前要回收的作业名: job1
***************打印job1作业在辅存中的信息***************
  记录      块号      柱面号      磁头号      扇区号
该作业无存储信息! 回收成功!
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 0  0  0  1  1  1  0  1  0  1  1  1  1  1  0  0  0  1  0  1  1  1  1  0  1  0  0  1  0  1  1  0  0  0  1  0  0  0  1  0  1  0  0  0  0  0  1  1  1  0  0  1  0  0  0  1  1  1  0  1  1  1  1  1  1
 1  1  1  1  0  0  0  0  1  1  0  1  1  1  1  1  0  1  1  0  1  1  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  1  1  1  1  1  1  1  0  1  1  0  1  0  0  1  0  0  1  1  1  0  1  1  1  1  1  1  1
 2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 4  1  1  1  1  1  1  1  1  1  1  1  1  0  1  1  0  0  1  1  1  0  1  1  0  1  1  1  0  1  1  0  0  0  1  1  0  1  0  0  1  1  0  0  1  0  1  0  1  1  0  1  1  1  1  0  1  0  1  1  1  1  1  1  1
 5  0  1  0  0  0  0  0  0  1  0  1  1  0  0  1  0  1  1  1  1  1  0  1  0  0  1  1  1  1  1  1  0  1  1  0  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  0  1  1  0
 6  1  1  0  1  1  1  1  0  0  0  0  1  0  1  0  1  1  0  1  0  0  0  0  1  1  0  1  1  1  1  1  0  0  1  0  1  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  1  0  0  0  0  0  0  1  1  1  1  1  0
 7  0  0  1  0  1  0  0  0  1  1  0  0  0  1  0  0  1  0  0  0  1  0  1  0  1  1  0  0  1  0  1  1  0  0  1  1  0  0  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  0  1  0  0  1  1  1  1  0  1  0
 8  0  0  0  0  1  1  1  0  1  1  1  0  0  0  1  0  0  1  0  1  0  1  1  1  0  0  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  0  1  0  0  1  1  1  1  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0
 9  0  1  0  1  0  1  1  0  0  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  1  0  1  1  1  0  1  0  0  1  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  1  1  0  1  0  1  0  0  1  0  0  0  0
10  1  0  1  1  0  0  1  0  0  1  1  1  0  1  0  0  0  1  0  1  1  1  0  1  1  1  1  1  1  1  1  1  1  0  1  1  1  0  1  0  0  0  0  1  1  0  0  1  0  0  0  0  0  1  0  0  0  1  0  1  0  1  1  1
11  0  0  0  1  1  0  0  0  1  1  1  1  0  0  1  1  1  0  1  1  0  1  0  0  1  1  0  1  0  0  0  1  1  0  1  0  0  0  1  1  0  1  0  1  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  1  1  1  0
12  0  0  0  0  1  1  1  1  0  0  1  1  0  0  1  1  0  0  0  1  0  1  1  1  1  1  0  0  0  0  1  1  1  1  1  0  1  1  0  1  1  0  0  1  1  1  1  0  1  1  0  0  1  1  0  0  1  1  0  0  1  1  0  0
13  1  0  1  1  0  1  0  0  1  0  1  1  1  0  1  0  1  0  1  0  1  1  1  1  0  1  0  0  0  1  1  0  1  0  0  1  0  0  0  1  1  1  0  1  1  0  1  0  1  0  0  0  0  0  1  0  0  1  0  0  1  1  1  0
14  0  1  1  1  1  0  0  0  1  0  1  1  0  0  1  0  1  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  0  0  1  1  0  1  1  0  0  1  1  1  0  1  0  1  1  0  1  1  1  1  1  0  0  1  0  1  1  0
15  0  0  0  0  0  1  0  0  1  1  1  1  1  0  1  0  1  0  0  0  0  0  0  1  1  0  0  1  0  0  1  0  0  0  1  1  0  1  0  0
辅存剩余空块数: 416
***************辅存管理***************
        *       1.空间分配       *
        *       2.空间去配       *
        *       0.退出           *
                请输入选项[2]
当前分配的作业: job2
请输入你当前要回收的作业名: job2
***************打印job2作业在辅存中的信息***************
  记录      块号      柱面号      磁头号      扇区号
该作业无存储信息! 回收成功!
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 0  0  0  1  1  1  0  1  0  1  1  1  1  1  0  0  0  1  0  1  1  1  1  0  1  0  0  1  0  1  1  0  0  0  1  0  0  0  1  0  1  0  0  0  0  0  1  1  1  0  0  1  0  0  0  1  1  1  0  1  1  1  1  1  1
 1  1  1  1  0  0  0  0  1  1  0  1  1  1  1  1  0  1  1  0  1  1  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  1  1  1  1  1  1  1  0  1  1  0  1  0  0  1  0  0  1  1  1  0  0  1  0  0  1  1  1
 2  1  1  1  1  1  0  1  0  0  0  1  1  1  0  1  0  1  1  0  1  0  1  0  0  0  1  1  0  0  1  0  0  0  1  1  1  0  1  0  1  1  1  0  1  1  0  0  0  1  0  0  0  0  1  1  0  0  0  0  0  0  1  1  0
 3  0  1  0  0  1  0  0  0  0  1  1  1  1  1  0  0  1  1  0  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  0  1  1  1  0  0  0  0  1  0  1  0  0  0  1  1  1  0  0  1  1  0  1  0
 4  1  0  1  0  0  1  0  0  1  1  1  0  0  1  1  0  0  1  1  1  0  1  1  0  1  1  1  0  1  1  0  0  0  1  1  0  1  0  0  1  1  0  0  1  0  1  0  1  1  0  1  1  1  1  0  1  0  1  1  1  1  1  1  1
 5  0  1  0  0  0  0  0  0  1  0  1  1  0  0  1  0  1  1  1  1  1  0  1  0  0  1  1  1  1  1  1  0  1  1  0  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  0  1  1  0
 6  1  1  0  1  1  1  1  0  0  0  0  1  0  1  0  1  1  0  1  0  0  0  0  1  1  0  1  1  1  1  1  0  0  1  0  1  0  0  1  1  1  0  0  0  0  1  1  1  0  0  1  1  0  0  0  0  0  0  1  1  1  1  1  0
 7  0  0  1  0  1  0  0  0  1  1  0  0  0  1  0  0  1  0  0  0  1  0  1  0  1  1  0  0  1  0  1  1  0  0  1  1  0  0  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  0  1  0  0  1  1  1  1  0  1  0
 8  0  0  0  0  1  1  1  0  1  1  1  0  0  0  1  0  0  1  0  1  0  1  1  1  0  0  1  0  1  1  0  1  0  1  0  1  1  1  1  0  0  0  1  0  0  1  1  1  1  0  1  0  0  0  1  0  1  0  1  0  1  0  0  0
 9  0  1  0  1  0  1  1  0  0  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  1  0  1  1  1  0  1  0  0  1  1  0  0  1  0  0  0  0  0  1  1  1  1  1  0  1  1  0  1  0  1  0  0  1  0  0  0  0
10  1  0  1  1  0  0  1  0  0  1  1  1  0  1  0  0  0  1  0  1  1  1  0  1  1  1  1  1  1  1  1  1  1  0  1  1  1  0  1  0  0  0  0  1  1  0  0  1  0  0  0  0  0  1  0  0  0  1  0  1  0  1  1  1
11  0  0  0  1  1  0  0  0  1  1  1  1  0  0  1  1  1  0  1  1  0  1  0  0  1  1  0  1  0  0  0  1  1  0  1  0  0  0  1  1  0  1  0  1  1  0  1  0  0  0  0  0  0  0  1  1  1  1  0  0  1  1  1  0
```

```
 12  0  0  0  0  1  1  1  1  0  0  1  1  0  0  1  1  0  0  0  1  0  1  1  1  1  1  0  0  0  0  1  1  1  1  1  1  0  1  1  0  1  1  0  1  0  0  1  1  0  0  1  1  0  0  1  1  0  0
 13  1  0  1  1  0  1  0  0  1  0  1  1  1  0  1  0  1  0  1  0  1  1  1  1  1  0  1  0  0  0  1  1  0  1  0  0  1  0  0  1  1  0  0  0  1  1  0  0  1  1  0  0  0  0  0  1  0  0  1  0  0  1  1  1  0
 14  0  1  1  1  1  0  0  0  1  0  1  1  0  0  1  0  1  1  1  0  0  0  0  1  0  0  1  1  1  0  0  0  0  1  0  0  1  1  0  1  1  0  0  1  1  1  0  1  0  1  1  0  1  1  1  1  1  0  0  1  0  1  1  0
 15  0  0  0  0  0  1  0  0  1  1  1  1  1  0  1  0  1  0  0  0  0  0  0  1  1  0  0  1  0  0  1  0  0  0  1  1  0  1  0  0
辅存剩余空块数：496
***************辅存管理***************
        *       1.空间分配      *
        *       2.空间去配      *
        *       0.退出          *
                请输入选项[0]
```

6. 注意事项

- 本项目源码地址：https://github.com/Yang-Zhongshan/OS-labsource
- 由于时间、技术水平有限，代码难免出现纰漏，如果您发现bug，在该项目地址提交Issue即可