学号 E21514033 专业 软件工程 姓名 何铭春

实验日期 **12.19**  教师签字 成绩

实验报告

【实验名称】 磁盘调度算法SSTF和SCAN比较

【实验目的】

比较不同磁盘调度算法的运行方式，比较他们的寻道长度。

【实验原理】

SSTF:

要求访问的磁道与当前磁头所在的磁道距离最近，即每次选取与上次磁道最近的磁道访问。

SCAN:

扫描算法不仅考虑到欲访问的磁道与当前磁道间的距离，更优先考虑的是磁头当前的移动方向。即在移动方向上选取距离最近的磁道。

【实验内容】

SSTF函数：

先假设第一个是要选取的赋值到choose，计算距离赋值给distance,然后不断循环查找是否有更小的距离。

def SSTF(tempList):

orderList=[]

orderList.append(tempList[0])

tempList.remove(tempList[0])

while True:

if len(tempList)==0:

break

choose=0

distance=abs(orderList[-1]-tempList[choose])

for i in range(1,len(tempList)):

newDistance=abs(orderList[-1]-tempList[i])

if newDistance<distance:

choose=i

distance=newDistance

orderList.append(tempList[choose])

tempList.remove(tempList[choose])

return orderList

SCAN函数：

预设flag=1，表示从左向右访问。

1. 循环，如果所有磁道都被访问那就结束循环。
2. choose预设为none，如果循环一圈没有赋值choose，那就反向访问，flag=flag\*-1
3. 其余按照最小距离计算方法，求choose

def SCAN(tempList):

orderList=[]

orderList.append(tempList[0])

tempList.remove(tempList[0])

tempList.sort()

flag=1

while True:

if len(tempList)==0:

break

choose=-1

distance="none"

for i in range(0,len(tempList)):

newdistance=tempList[i]-orderList[-1]

newdistance=newdistance\*flag

if newdistance<0:

pass

else:

if distance=="none" or newdistance<distance:

distance=newdistance

choose=i

if choose=="none":

flag=flag\*(-1)

else:

orderList.append(tempList[choose])

tempList.remove(tempList[choose])

return orderList



【小结或讨论】

1. 使用flag来控制扫描方向，flag=1即从左向右，flag=-1即从右到左。做法是直接相减的距离乘flag，当flag=1时，正数的距离被选中。反之负数距离被选中。
2. SSTF只需要找到目前距离最短的即可。