

操作系统实验-移动头磁盘调度算法模拟实现与比较

1、实验目的

理解并掌握主要的移动头磁盘调度算法的基本设计思想和编程实现要旨。

2、实验内容

利用标准 C 语言，编程设计与实现关于移动头磁盘调度的先来先服务调度算法（FCFS）、最短寻道时间优先调度算法（SSTF）、电梯调度算法（SCAN）、循环式单向电梯调度算法（CSCAN）、双队列电梯调度算法（FSCAN），并随机发生一组磁盘访问事件（磁道号）序列，开展有关算法的测试及性能比较。

3、开发环境

Vs 2021+Windows操作系统

4、算法流程和实现方法

FCFS（先来先服务）

- **关键数据结构：**
 - 仅需要请求队列即可，按照请求顺序进行处理。
- **算法流程：**
 - 请求按照到达的顺序依次处理。
 - 磁头从当前位置移动到下一个请求所在的磁道。
 - 移动完成后，继续处理下一个请求，直至队列中的所有请求被满足。

SSTF（最短寻道时间优先）

- **关键数据结构：**
 - 请求队列，当前磁头位置。
- **算法流程：**
 - 每次选择离当前磁头位置最近的请求来处理。
 - 算法优先处理离当前磁头最近的磁道，最大限度地减少寻道时间。
 - 该算法可能导致某些请求长期等待，产生"饥饿"问题。

SCAN（扫描算法）

- **关键数据结构：**
 - 请求队列，当前磁头位置。
- **算法流程：**
 - 磁头按照一个方向（通常是向某一方向移动）扫描所有请求。
 - 当达到最边缘时，方向改变并继续扫描，而不返回到最初的位置。
 - 这个算法可能导致某些磁道长期无法被访问到（产生"电梯"效应）。

CSCAN（循环扫描算法）

- **关键数据结构：**
 - 请求队列，当前磁头位置。
- **算法流程：**

- 类似于SCAN，但是在到达最边缘后，直接回到另一端而不是返回最初的位置。
- 这样做有助于减少某些磁道长期无法访问的情况，减少"电梯"效应。

FSCAN（双重扫描算法）

- **关键数据结构：**
 - 两个请求队列，分别为当前进行的队列和等待的队列。
- **算法流程：**
 - 将所有请求分为两个队列，在处理当前队列的同时，后续请求进入等待队列。
 - 当前队列处理完毕后，切换到等待队列处理请求。
 - 这种方法减少了某些请求长时间等待的情况，减轻了"饥饿"和"电梯"效应。

5、技术难点及解决方案

算法效率与性能

- **难点：**算法的效率和性能是关键指标，但不同算法在不同场景下的表现可能不同。
- **解决方案：**进行实际测试和性能评估，比较不同算法在不同数据集和工作负载下的表现。根据实验结果优化算法选择，或者结合多种算法实现智能调度机制。

算法的正确性

- **难点：**确保算法的正确性和稳定性，避免出现错误和异常情况。
- **解决方案：**进行严格的测试和验证，针对各种边界条件和极端情况对算法进行全面测试，确保算法在各种情况下都能正确运行并且表现稳定。

6、运行截图

```
FCFS的平均寻道数为:79

SSTF的访问序列为:111->109->108->106->103->102->99->98->97->96->93->92->90->89->85->85->83->79->74->70->69->66->64->63->62->60->57->55->51->51->49->46->46->43->41->36->35->35->34->34->33->29->29->29->25->24->23->22->21->20->16->15->12->10->6->3->2->1->114->118->118->120->122->128->128->129->130->133->136->138->140->145->146->148->149->156->157->160->160->163->168->173->173->176->177->177->178->183->184->185->188->192->192->194->195->195->196->197->199
SSTF的平均寻道数为:3

SCAN的访问序列为:114->118->118->120->122->128->128->129->130->133->136->138->140->145->146->148->149->156->157->160->160->163->168->173->173->176->176->177->177->178->183->184->185->188->192->192->194->195->195->196->197->199->111->109->108->106->103->102->99->98->97->96->93->92->90->89->85->85->83->79->74->70->69->66->64->63->62->60->57->55->51->51->49->46->46->43->41->36->35->35->34->34->33->29->29->29->25->24->23->22->21->20->16->15->12->10->6->3->2->1
SCAN的平均寻道数为:2

CSCAN的访问序列为:114->118->118->120->122->128->128->129->130->133->136->138->140->145->146->148->149->156->157->160->160->163->168->173->173->176->176->177->177->178->183->184->185->188->192->192->194->195->195->196->197->199->1->2->3->6->10->12->15->16->20->21->22->23->24->25->29->29->29->33->34->34->35->35->36->41->43->46->46->49->51->51->55->57->60->62->63->64->66->69->70->74->79->83->85->85->89->90->92->93->96->97->98->99->102->103->106->108->109->111
CSCAN的平均寻道数为:3

FSCAN的访问序列为:114->118->129->140->146->149->156->160->160->168->173->177->177->185->188->195->196->197->108->106->103->102->99->97->96->93->90->89->79->74->69->66->51->51->46->41->36->35->33->29->25->24->21->16->15->12->10->6->3->1->118->120->122->128->128->130->133->136->138->145->148->157->163->173->176->176->178->183->184->192->192->194->195->199->111->109->98->92->85->85->83->70->64->63->62->60->57->55->49->46->43->35->34->34->29->29->23->22->20->2
FSCAN的平均寻道数为:5

D:\2023githubcode\cpp\模拟移动磁盘调度算法\x64\Debug\模拟移动磁盘调度算法.exe (进程 13976)已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。...
```

性能比较：

FCFS：

- **优点：**简单易实现。
- **缺点：**可能会导致平均寻道时间较长，因为它忽略了寻找最优路径的可能性。

SSTF：

- **优点：**最小化了磁头移动距离，平均寻道时间较短。
- **缺点：**可能产生"饥饿"问题，某些请求长时间等待。

SCAN：

- **优点：**减少了某些磁道长期无法访问的情况。

- **缺点：**可能导致某些请求长时间等待，产生"电梯"效应。

CSCAN：

- **优点：**减少了某些磁道长期无法访问的情况。
- **缺点：**与SCAN类似，可能产生长时间等待的情况。

FSCAN：

- **优点：**减轻了"饥饿"和"电梯"效应，提高了公平性。
- **缺点：**实现相对复杂。