OS 2022fall 11.10 hw10

20307140008 施君豪

运行环境：Windows 10 (pycharm)

1. **第三十七章**
2. **(1) disk.py -a 0 -G**

**Seek: 0 Rotate:165 Transfer: 30 Total: 195**

**（5.5\*30+30 = 195）**

**(2) disk.py -a 30 -G**

**Seek: 80 Rotate:265 Transfer: 30 Total: 375**

**(3) disk.py -a 7,30,8 -G**

**Block: 7 Seek: 0 Rotate: 15 Transfer: 30 Total: 45**

**Block: 30 Seek: 80 Rotate:220 Transfer: 30 Total: 330**

**Block: 8 Seek: 80 Rotate:310 Transfer: 30 Total: 420**

**TOTALS： Seek:160 Rotate:545 Transfer: 90 Total: 795**

**(4) disk.py -a 10,11,12,13 -G**

**Block: 10 Seek: 0 Rotate:105 Transfer: 30 Total: 135**

**Block: 11 Seek: 0 Rotate: 0 Transfer: 30 Total: 30**

**Block: 12 Seek: 40 Rotate:320 Transfer: 30 Total: 390**

**Block: 13 Seek: 0 Rotate: 0 Transfer: 30 Total: 30**

**TOTALS Seek: 40 Rotate:425 Transfer:120 Total: 585**

1. **(1) disk.py -a 0 -G -S 2 / -S 4 / -S 8 /-S 10 / -S 40 / -S 0.1**

**均为195**

**(2) disk.py -a 30 -G -S 2 / -S 4 / -S 8 /-S 10 / -S 40 / -S 0.1**

**-S 0.1为1095，其余均为375**

**(3) disk.py -a 7,30,8 -G -S 2 / -S 4 / -S 8 /-S 10 / -S 40 / -S 0.1**

**-S 0.1为2235，-S 2 为795，其余为435**

**(4) disk.py -a 10,11,12,13 -G -S 2 / -S 4 / -S 8 /-S 10 / -S 40 / -S 0.1**

**-S 0.1为945，其余均为585**

1. **(1) disk.py -a 0 -G -R 0.1 / -R 0.5 / -R 0.01**

**分别为：1950，390，19500**

**(2) disk.py -a 30 -G -R 0.1 / -R 0.5 / -R 0.01**

**分别为：3750，750，37501**

**(3) disk.py -a 7,30,8 -G -R 0.1 / -R 0.5 / -R 0.01**

**分别为：4349，1590，43500**

**(4) disk.py -a 10,11,12,13 -G -R 0.1 / -R 0.5 / -R 0.01**

**分别为：5850，1170，58501**

1. **默认的处理顺序为：7，30，8（FIFO）**

**该场景下更好的调度顺序：7，8，30（SSTF）**

**disk.py -a 7,30,8 -G**

**TOTALS： Seek:160 Rotate:545 Transfer: 90 Total: 795**

**总时间为795**

**disk.py -a 7,30,8 -G -p SSTF**

**TOTALS Seek: 0 Rotate:168 Transfer: 90 Total: 375**

**总时间为375**

1. **SATF的处理顺序为：7，8，30，与SSTF相同**

**TOTALS Seek: 0 Rotate:168 Transfer: 90 Total: 375**

**总时间为375，与SSTF相同**

**disk.py -a 12,31 -c -S 40 -R 3 -p SSTF**

**TOTALS Seek: 2 Rotate:113 Transfer: 20 Total: 135**

**disk.py -a 12,31 -c -S 40 -R 3 -p SATF**

**TOTALS Seek: 3 Rotate: 42 Transfer: 20 Total: 65**

**出现显著差距的原因是寻道速度和旋转速度的差距。**

1. **原因为寻道的时候从外圈旋转到中圈的时候越过了12的起点，需要重新旋转一圈后从头读起。可以引入磁道偏斜解决。**

**disk.py -a 10,11,12,13 -G -o 5**

**TOTALS Seek: 40 Rotate:215 Transfer:120 Total: 375**

**设寻道距离为D，旋转角速度为W，每区跨越角度为A，偏斜为A**

**公式：D/V < A\*O/W，所以O > D\*W/V\*A，满足条件的O都可**

1. **以下为运行结果：**

**disk.py -z 10,20,30 -a -1 -A 5,-1,0 -c -s 1**

**REQUESTS [7, 45, 41, 13, 26]**

**Block: 7 Seek: 0 Rotate:245 Transfer: 10 Total: 255**

**Block: 45 Seek: 40 Rotate: 55 Transfer: 20 Total: 115**

**Block: 41 Seek: 0 Rotate:260 Transfer: 20 Total: 280**

**Block: 13 Seek: 40 Rotate:335 Transfer: 10 Total: 385**

**Block: 26 Seek: 0 Rotate:120 Transfer: 10 Total: 130**

**TOTALS Seek: 80 Rotate:1015 Transfer: 70 Total:1165**

**外圈带宽为：3/(255+385+120) = 0.0039**

**中间带宽为：2/(115+280) = 0.00506**

1. **disk.py -A 1000,-1,0 -w 1 -p SATF -c**

**TOTALS Seek:20960 Rotate:169165 Transfer:30000 Total:220125**

**disk.py -A 1000,-1,0 -w 10 -p SATF -c**

**TOTALS Seek:8080 Rotate:26555 Transfer:30000 Total:64635**

**disk.py -A 1000,-1,0 -w 20 -p SATF -c**

**TOTALS Seek:3680 Rotate:14395 Transfer:30000 Total:48075**

**disk.py -A 1000,-1,0 -w 50 -p SATF -c**

**TOTALS Seek:1680 Rotate:7035 Transfer:30000 Total:38715**

**disk.py -A 1000,-1,0 -w 100 -p SATF -c**

**TOTALS Seek:1440 Rotate:5835 Transfer:30000 Total:37275**

**disk.py -A 1000,-1,0 -w 1000 -p SATF -c**

**TOTALS Seek:1520 Rotate:3955 Transfer:30000 Total:35475**

**经过计算，在-w值为172时达到最大性能，时间为35475.**

**当-w设置为1时，实际所花时间与使用方法无关。**

1. **举例：在一个连续的外圈访问序列之中插入一个内圈的访问，SATF的策略为在外圈访问完之后再去访问内圈那一个，导致饥饿产生。**

**disk.py -a 7,8,9,31,10,11 -w 10 -p SATF -G**

**REQUESTS ['7', '8', '9', '31', '10', '11']**

**Block: 7 Seek: 0 Rotate: 15 Transfer: 30 Total: 45**

**Block: 8 Seek: 0 Rotate: 0 Transfer: 30 Total: 30**

**Block: 9 Seek: 0 Rotate: 0 Transfer: 30 Total: 30**

**Block: 10 Seek: 0 Rotate: 0 Transfer: 30 Total: 30**

**Block: 11 Seek: 0 Rotate: 0 Transfer: 30 Total: 30**

**Block: 31 Seek: 27 Rotate:130 Transfer: 30 Total: 187**

**TOTALS Seek: 27 Rotate:145 Transfer:180 Total: 405**

**disk.py -a 7,8,9,31,10,11 -w 4 -p BSATF -G**

**REQUESTS ['7', '8', '9', '31', '10', '11']**

**Block: 7 Seek: 0 Rotate: 15 Transfer: 30 Total: 45**

**Block: 8 Seek: 0 Rotate: 0 Transfer: 30 Total: 30**

**Block: 9 Seek: 0 Rotate: 0 Transfer: 30 Total: 30**

**Block: 31 Seek: 80 Rotate:190 Transfer: 30 Total: 300**

**Block: 11 Seek: 80 Rotate: 10 Transfer: 30 Total: 120**

**Block: 10 Seek: 0 Rotate:300 Transfer: 30 Total: 330**

**TOTALS Seek:160 Rotate:515 Transfer:180 Total: 855**

**可见饥饿得到了缓解，但是相比于SATF降低了运行速度。**

**一般来说，磁盘应该尽可能加快运行速度，但也应该同时避免饥饿现象的出现。**

1. **见下列代码：**

**disk.py -a 9,20 -c**

**TOTALS Seek: 40 Rotate:335 Transfer: 60 Total: 435**

**disk.py -a 9,20 -c -p SATF**

**TOTALS Seek: 80 Rotate:325 Transfer: 60 Total: 465**