

**Subject:** Lecture on Operating System Lecture Exercise 2.2

**From:** IKENOYA Katsutoshi <j05002@ie.u-ryukyu.ac.jp>

**Date:** Fri, 03 Nov 2006 20:59:15 +0900

**To:** kono@ie.u-ryukyu.ac.jp

学籍番号 : 055702B ( j05002 )

### 問題2.2

Track 間の移動速度 3 msec/track のハードディスクで、Track 5, 20, 100 のアクセスの100msec後に、Track 7, 120, 70, 2 の順でアクセスが来た。これを、もっとも効率よくアクセスするためには、どのような順番でTrackにアクセスすれば良いか? (回転の待ち時間は無視するものとする)  
一般的には、データの要求を前もって知らなければ最適なスケジューリングをすることはできない。公平性を考慮したもには、SCAN algorithmというものがある。これは内向きに移動したら、内側のリクエストがなくなるまで内側のリクエストを処理し、それから外向きに順番に処理するというものである。  
現状ではハードディスクのデータ転送速度は回転数で決まる。市販されているハードディスクの回転数と転送速度を調べてみよう。

・以下のスケジューリングアルゴリズムを用いた時のアクセス時間を調べてみた

(1)FCFS (First-come, First-Serve algorithm)

命令が来た順番にアクセスするアルゴリズム。

(2)SSTF (Shortest-seek-time-first algorithm)

移動距離が最短のトラックからアクセスしていくアルゴリズム。

途中で新しいアクセス要求があった場合その処理を優先する。

(3)SCAN algorithm

内向きに移動したら、内側のリクエストがなくなるまで内側のリクエストを処理し(一番内側まで移動する)、それから外向きに順番に処理するアルゴリズム。

(4)C-SCAN (Circle SCAN) Scheduling

内向きに移動したら、内側のリクエストがなくなるまで内側のリクエストを処理し(一番内側まで移動する)、その後一番外側に移動し、リクエストを処理するアルゴリズム。

(5)C-LOOK scheduling

内向きに移動したら、内側のリクエストがなくなるまで内側のリクエストを処理し(一番内側まで移動はしない)、それから外向きに順番に処理するアルゴリズム。

スタート位置を50とし、Trackの一番外側を199とする。

(1)FCFS

アクセスの順番

50 → 5 → 20 → 100 → 7 → 120 → 70 → 2

移動時間

135 + 45 + 240 + 279 + 339 + 150 + 204 = 1392msec

(2)SSTF

アクセスの順番

50 → 20 → 7 → 5 → 2 → 70 → 100 → 120

移動時間

90 + 39 + 6 + 9 + 204 + 90 + 60 = 498msec

(3)SCAN algorithm

アクセスの順番

50 → 20 → 7 → 5 → 2 → 0 → 70 → 100 → 120

移動時間

90 + 39 + 6 + 9 + 6 + 210 + 90 + 60 = 510msec

(4)C-SCAN

アクセスの順番

50 → 20 → 7 → 5 → 2 → 0 → 199 → 120 → 100 → 70

移動時間

90 + 39 + 6 + 9 + 6 + 597 + 237 + 60 + 90 = 1134msec

(5)C-LOOK

50 → 20 → 7 → 5 → 2 → 70 → 100 → 120

移動時間

$90 + 39 + 6 + 9 + 204 + 90 + 60 = 498\text{msec}$

以上の結果より今回の場合はSSTFとC-LOOKが最も効率が良いアクセス方法である。