

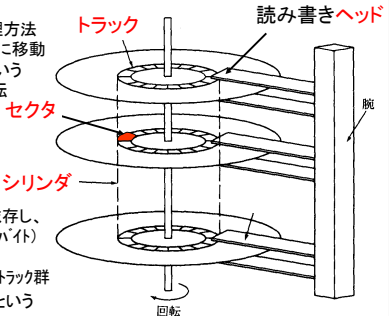
第10回 ディスクスケジューリング

重要: ディスク機構

基礎OSのスライド

セクタ: トラックを等分割した円弧部分(注1)。トラック(注2): ディスク面の円周部分
シリンダ: 同一円周上のトラックを一まとめにした円筒部分(注3)
ハードウェアはセクタ単位に処理(読み出し、書き込み)

- ハードウェアによるセクタの処理方法
- ①ヘッドをセクタがあるシリンダに移動
この動作をシーク(注4)という
 - ②セクタがヘッドの下まで回転
してくるまで待つ
 - ③セクタが通過する間に
該当トラックのヘッドから
データを読み書きする

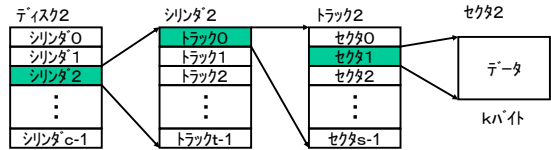


注1: セクタの大きさは装置に依存し、
32-4096バイト(通常512バイト)
注2: 面とも呼ばれる
注3: シリンダはヘッド位置が同じトラック群
注4: 日本語では位置決めという

シリンダ、トラック、セクタと容量の関係

基礎OSのスライド

ディスクのハードウェアアドレスは、シリンダ番号、トラック番号(注)、セクタ番号の3次元配列



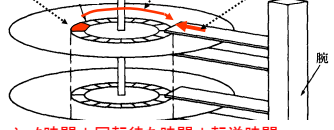
注: トラック番号は面番号とも呼ばれる

トラック当たりセクタ数: s 1トラックs * kバイト
シリンダ当たりトラック数: t 1シリンダt * s * kバイト (トラック数=ディスクの面数)
ディスク当たりシリンダ数: c 1ディスクc * t * s * kバイト

k=500
S=80
T=6
C=9000
2, 160, 000, 000 = 2GBバイト

ディスクのアクセス時間

- (1) シーク(位置決め)
ヘッドを要求されたシリンダに移動
- (2) 回転待ち(サーチ)
要求されたセクタがヘッドの位置まで回ってくるのを待つ
- (3) 転送
データの転送
(要求されたセクタがヘッドの下を通過)



アクセス時間=シーク時間+回転待ち時間+転送時間

シーク時間>回転待ち時間>転送時間(転送するデータが少ない場合)
ディスクのスケジューリングは、ヘッドの移動順序が重要

参考: 市販ディスクの例



記憶容量	300.0 GB
平均シーク時間	3.4 ms(読み出し)
平均回転待ち時間	2.0 ms
ディスク回転速度	15,000 rpm
データ転送速度	最大320MB/s
Ultra 320 SCSI	
データ転送速度	最大400MB/s
Fibre Channel	

1GBは10億バイトです。使用できる記憶容量は本表記の数値より少なくなります。

平均回転待ち時間は回転速度から計算できる
ディスクアクセス時間中、シーク時間が占める割合が最も大きい

回転待ち時間

基礎OSのスライド

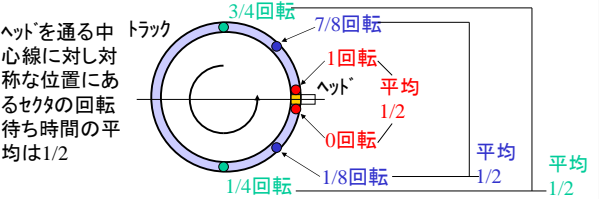
平均回転待ち時間 = $\frac{\text{ディスクの回転時間}}{2}$

10025rpmのハードディスクの平均回転待ち時間

10025[回転/分] ÷ 60[秒/分] = 167.0833[回転/秒]

回転時間: 1[秒] ÷ 167.0833回転 = 0.00598秒=5.98ms

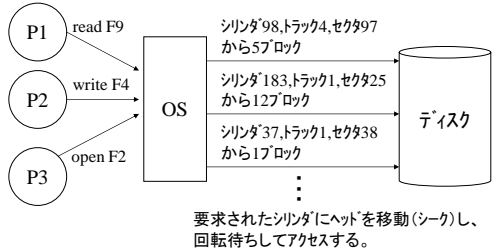
平均回転待ち時間: 5.98msec ÷ 2 = 2.99ms



ディスクへのアクセス要求

入出力要求
のシステムコール

ディスクへの入出力要求

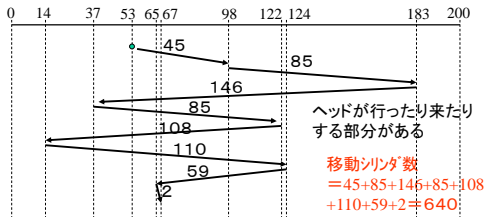


次スライド以降では、ヘッドがシリンダ番号53の位置にあり、以下のシリンダ番号への入出力要求が未処理である場合を例に説明する。
シリンダ番号(待ち列):98、183、37、122、14、124、65、67

ディスクスケジューリング(1)

先着順(FIFS: First Come First Served)

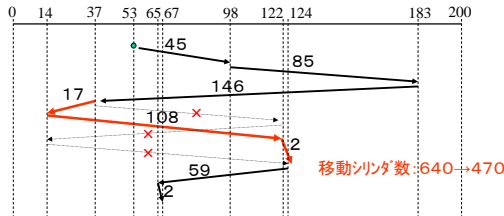
待ち列の順にヘッドを移動。処理は簡単だが、移動距離が大きいの。



待ち列:98、183、37、122、14、124、65、67
開始ヘッド位置53
処理順序:98、183、37、122、14、124、65、67

処理順序の変更

ヘッドが行き来する部分の処理順序を入れ替えると、移動シリンダ数が減少→アクセス時間が短縮される

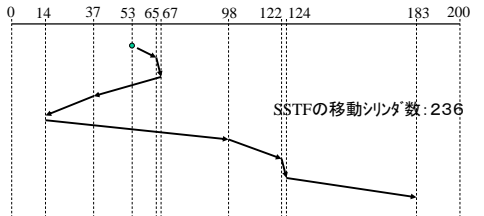


待ち列:98、183、37、122、14、124、65、67
開始ヘッド位置53
処理順序:98、183、37、14、122、124、65、67

ディスクスケジューリング(2)

最短位置決め時間優先(SSTF: Shortest Seek Time first)

現在のシリンダ位置から、最も近い要求にヘッドを移動(黒の実線の順序)
・移動距離が少ない。(但し、最短では無い。次スライド赤線の方が短い。)
・飢餓状態が起こり得る(他の要求に追い越され続けて、無限に待つ)



待ち列:98、183、37、122、14、124、65、67
開始ヘッド位置53
処理順序:65、67、37、14、98、122、124、183

ディスクスケジューリング(2: 参考)

ヘッドの移動距離が最短になるように、処理順序を最適化すれば、SSTFよりも効率的なスケジューリングが可能である。

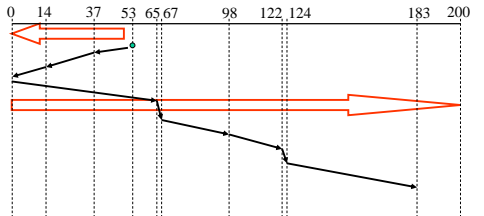


待ち列:98、183、37、122、14、124、65、67
開始ヘッド位置53
処理順序:37、14、65、67、98、122、124、183

ディスクスケジューリング(3)

SCAN

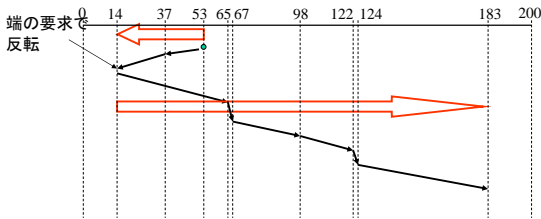
途中の要求を処理しながら、ヘッドが、端まで移動。
端(先頭、最後)に着いたら、反転し、再び要求を処理しながら移動
開始時、ヘッドがどちら方向に移動しているかで移動シリンダ数が異なる。



ディスクスケジューリング(4)

LOOK

SCANと同様。但し、ヘッド位置から進行方向に端まで、要求が無ければ、反転(要求を見ながら移動するため、LOOKという)

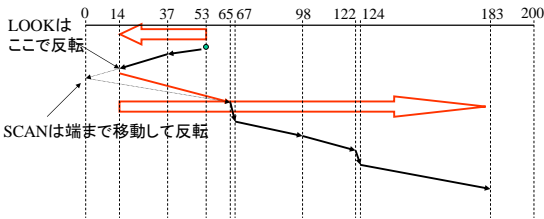


待ち列: 98, 183, 37, 122, 14, 124, 65, 67
開始ヘッド位置 53
処理順序: 37, 14, 65, 67, 98, 122, 124, 183

参考: ディスクスケジューリング(3, 4)

LOOKは、最後(一番端)の要求かどうかを見て反転

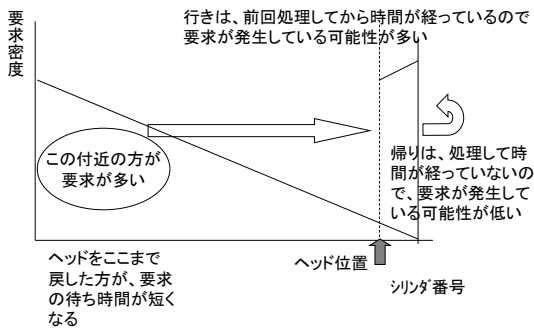
SCANでは、最後の要求かどうかを見ずに、端までヘッドが動いてから反転



待ち列: 98, 183, 37, 122, 14, 124, 65, 67
開始ヘッド位置 53

参考: シリンダに対する要求の発生

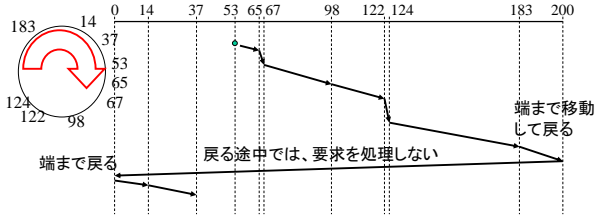
SCAN/LOOKにおいて、シリンダへのアクセス要求が、一様に発生する場合



ディスクスケジューリング(5)

C-SCAN: Circular SCAN

端(0番: 先頭のシリンダ)から要求を処理しながら移動。反対側の端(最後のシリンダ)まで来ると、端(0番: 先頭のシリンダ)に戻る。(SCANとは違い、戻るときには要求を処理しない)

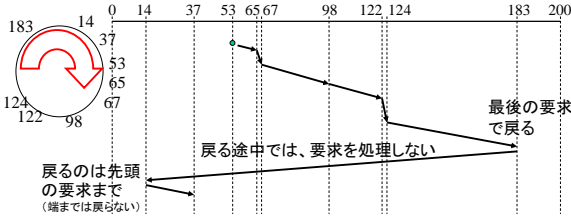


待ち列: 98, 183, 37, 122, 14, 124, 65, 67
開始ヘッド位置 53
処理順序: 65, 67, 98, 122, 124, 183, (200), (0), 14, 37

ディスクスケジューリング(6)

C-LOOK: Circular LOOK (Windowsは、この方式)

C-SCANと同様。但し、最後の要求まで来ると、先頭の要求に戻る



待ち列: 98, 183, 37, 122, 14, 124, 65, 67
開始ヘッド位置 53
処理順序: 65, 67, 98, 122, 124, 183, 14, 37

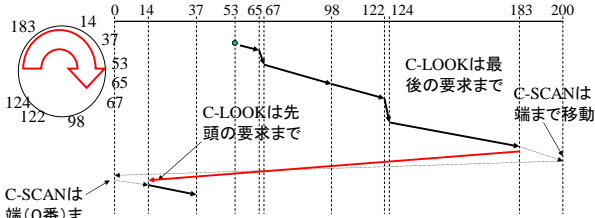
参考: ディスクスケジューリング(5, 6)

C-SCAN

端(最後のシリンダ)まで移動し、反対の端(0番: 先頭のシリンダ)に戻る。

C-LOOK

最後の要求まで来ると、先頭の要求に戻る



待ち列: 98, 183, 37, 122, 14, 124, 65, 67
開始ヘッド位置 53

ヘッド移動距離の計算 (SSTFの例)

(1) 要求(待ち列)を処理順序に並べ替える	(2) 移動距離(差)
- 98, 183, 37, 122, 14, 124, 65, 67	
• 開始ヘッド位置53に一番近い要求: 65	65 - 53 = 12
- 98, 183, 37, 122, 14, 124, 65, 67	
• 移動先ヘッド位置65に一番近い要求: 67	67 - 65 = 2
- 98, 183, 37, 122, 14, 124, 65, 67	
• 移動先ヘッド位置67に一番近い要求: 37	67 - 37 = 30
- 98, 183, 37, 122, 14, 124, 65, 67	
• 移動先ヘッド位置37に一番近い要求: 14	37 - 14 = 23
- 98, 183, 37, 122, 14, 124, 65, 67	
• 移動先ヘッド位置14に一番近い要求: 98	98 - 14 = 84
- 98, 183, 37, 122, 14, 124, 65, 67	
• 移動先ヘッド位置98に一番近い要求: 122	122 - 98 = 24
- 98, 183, 37, 122, 14, 124, 65, 67	
• 移動先ヘッド位置122に一番近い要求: 124	124 - 122 = 2
- 98, 183, 37, 122, 14, 124, 65, 67	
• 移動先ヘッド位置124に一番近い要求: 183	183 - 124 = 59
	236

各方式の名称とスケジューリング方法、移動距離を計算できるようになっておくこと

重要: ヘッドの移動距離

	処理の順序と移動距離											
FCFS	(53)	98	183	37	122	14	124	65	67	計		
	差	45	85	146	85	108	110	59	2	640		
SSTF	(53)	65	67	37	14	98	122	124	183	計		
	差	12	2	30	23	84	24	2	59	236		
SCAN(注)	(53)	37	14	(0)	65	67	98	122	124	183	計	
	差	16	23	14	65	2	31	24	2	59	236	
LOOK(注)	(53)	37	14	65	67	98	122	124	183	計		
	差	16	23	51	2	31	24	2	59	208		
C-SCAN	(53)	65	67	98	122	124	183	(200)	(0)	14	37	計
	差	12	2	31	24	2	59	17	200	14	23	384
C-LOOK	(53)	65	67	98	122	124	183	14	37	計		
	差	12	2	31	24	2	59	169	23	322		

注: SCANとLOOKは、開始位置からヘッドがどちら方向に移動していたかによって、移動距離が異なる。表は、先頭(0番)の方向に移動中の場合を示す

SCANとLOOKのヘッド移動距離

初期状態の前に、どちらに移動中であったかで、以後の移動距離が異なる。

	処理の順序と移動距離(末尾方向に移動中)											
SCAN(注)	(53)	65	67	98	122	124	183	(200)	37	14	計	
	差							147		185	332	
LOOK(注)	(53)	65	67	98	122	124	183	37	14	計		
	差						130		169	299		
	処理の順序と移動距離(先頭方向に移動中: 前のスライド)											
SCAN(注)	(53)	37	14	(0)	65	67	98	122	124	183	計	
	差	→	→	53	→	→	→	→	→	183	236	
LOOK(注)	(53)	37	14	65	67	98	122	124	183	計		
	差	→	39	→	→	→	→	→	→	169	208	

(他のアルゴリズムは、初期状態前の移動方向と、以後の移動とは無関係)

ヘッドの移動距離

	処理の順序と移動距離											
FCFS	(53)	98	183	37	122	14	124	65	67	計		
	差	→	130	146	85	108	110	59	2	640		
SSTF	(53)	65	67	37	14	98	122	124	183	計		
	差	→	14	→	53	→	→	→	169	236		
SCAN(注)	(53)	37	14	(0)	65	67	98	122	124	183	計	
	差	→	→	53	→	→	→	→	→	183	236	
LOOK(注)	(53)	37	14	65	67	98	122	124	183	計		
	差	→	39	→	→	→	→	→	→	169	208	
C-SCAN	(53)	65	67	98	122	124	183	(200)	(0)	14	37	計
	差	→	→	→	→	→	→	147	200	→	37	384
C-LOOK	(53)	65	67	98	122	124	183	14	37	計		
	差	→	→	→	→	→	130	169	23	322		

ヘッドの移動方向が変わらない間は、まとめて差を計算しても同じ答えになる

スケジューリング方式の選択

- ・ヘッド移動量: SSTF、更に最適なアルゴリズム
- ・ディスク負荷: SCAN、C-SCANを推奨した研究有
- ・要求の数: 待ち列が1個⇒SSTF、SCAN、LOOKはFCFSと同じ
例: 要求間隔とアクセス時間が指数分布で利用率50%⇒平均待ち列: 0. 5
- ・ファイル割当て技法: 連続的割当てvs鎖状ファイル、検索ファイル
- ・デレトリ、検索ブロック: 頻繁にアクセス⇒ディスクの中間に配置

独立したモジュールとし、必要に応じて取り替えて使用可能
(デフォルトで、SSTFまたはLOOKとするのが合理的)