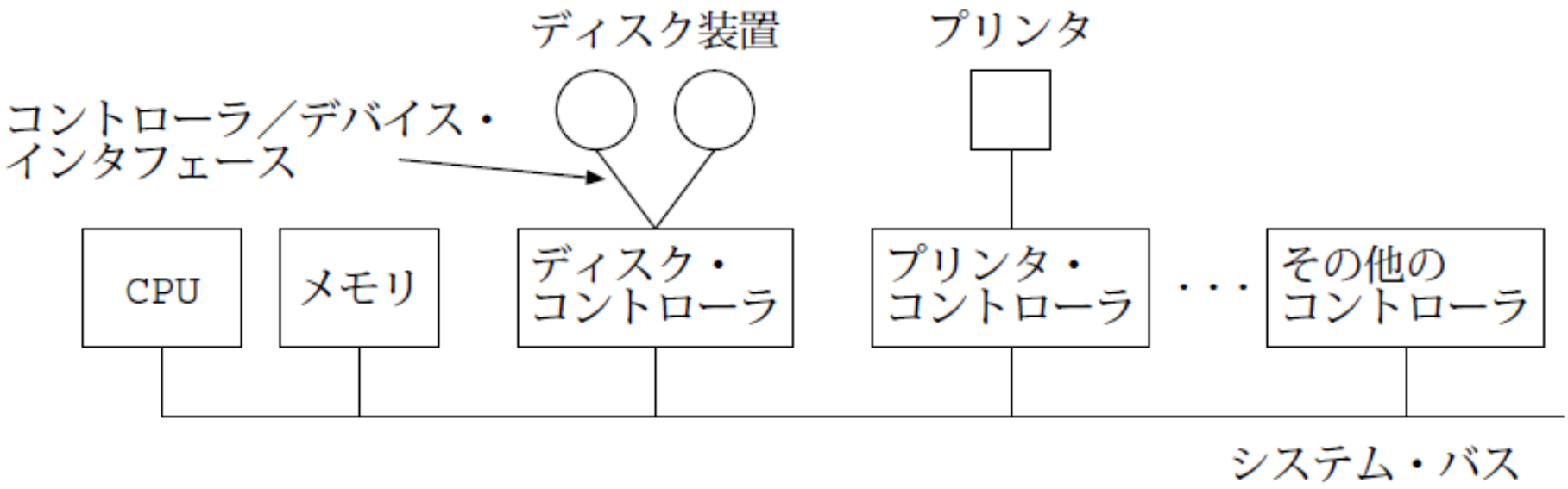
オペレーティングシステム論
(Theory of Operating Systems)
第十二回

ソフトウェア情報学部
基盤ソフトウェア講座
澤本　潤

入出力：ディスクアームスケジューリング

入出力装置(入出力デバイス)



入出力装置は、

- 機械的なパーツ と
- 電子的パーツ（コントローラ）

で構成される。

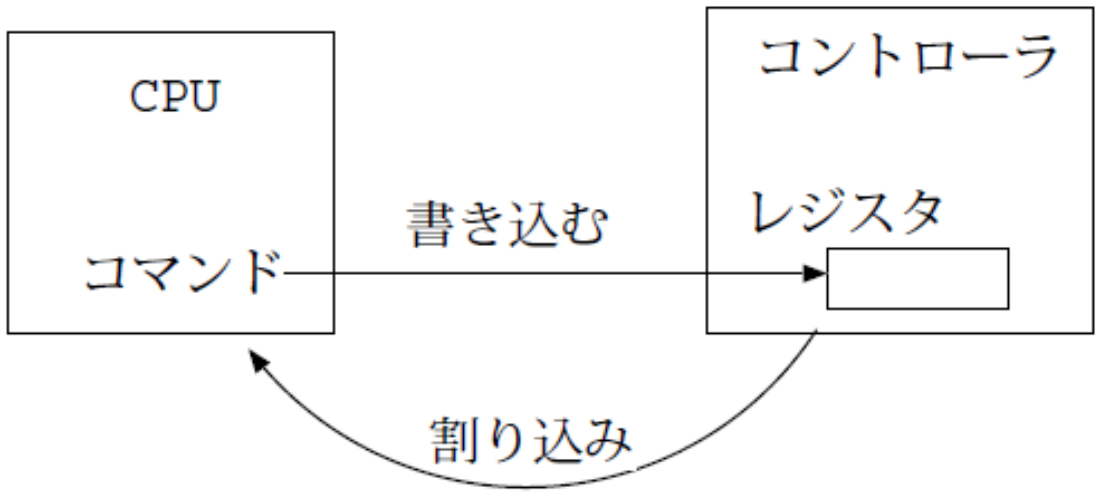
入出力装置の操作

コントローラは、CPUとの通信制御用レジスタを持つ。

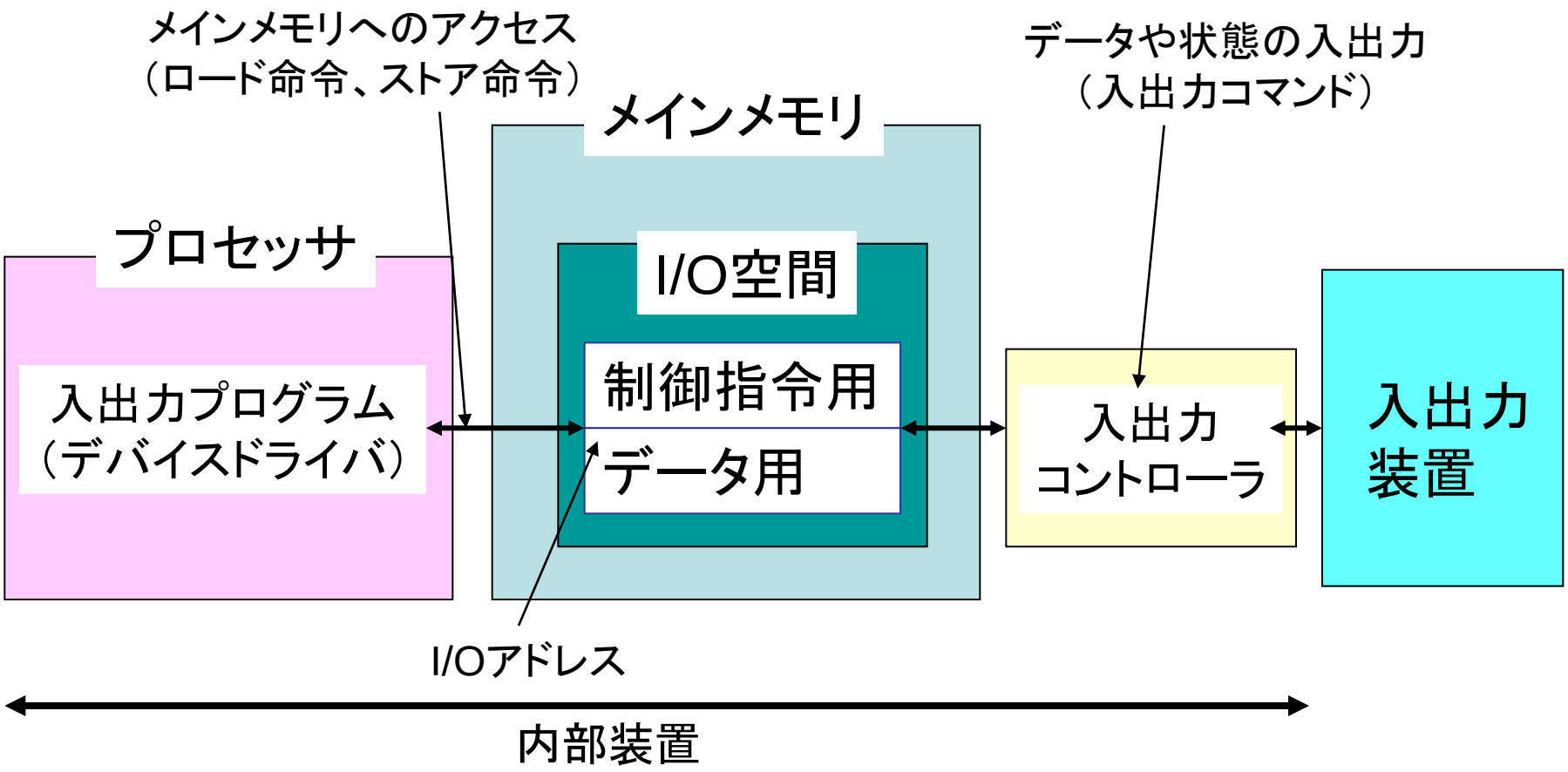
↑

CPUからメモリの一部として見える、システムもある。

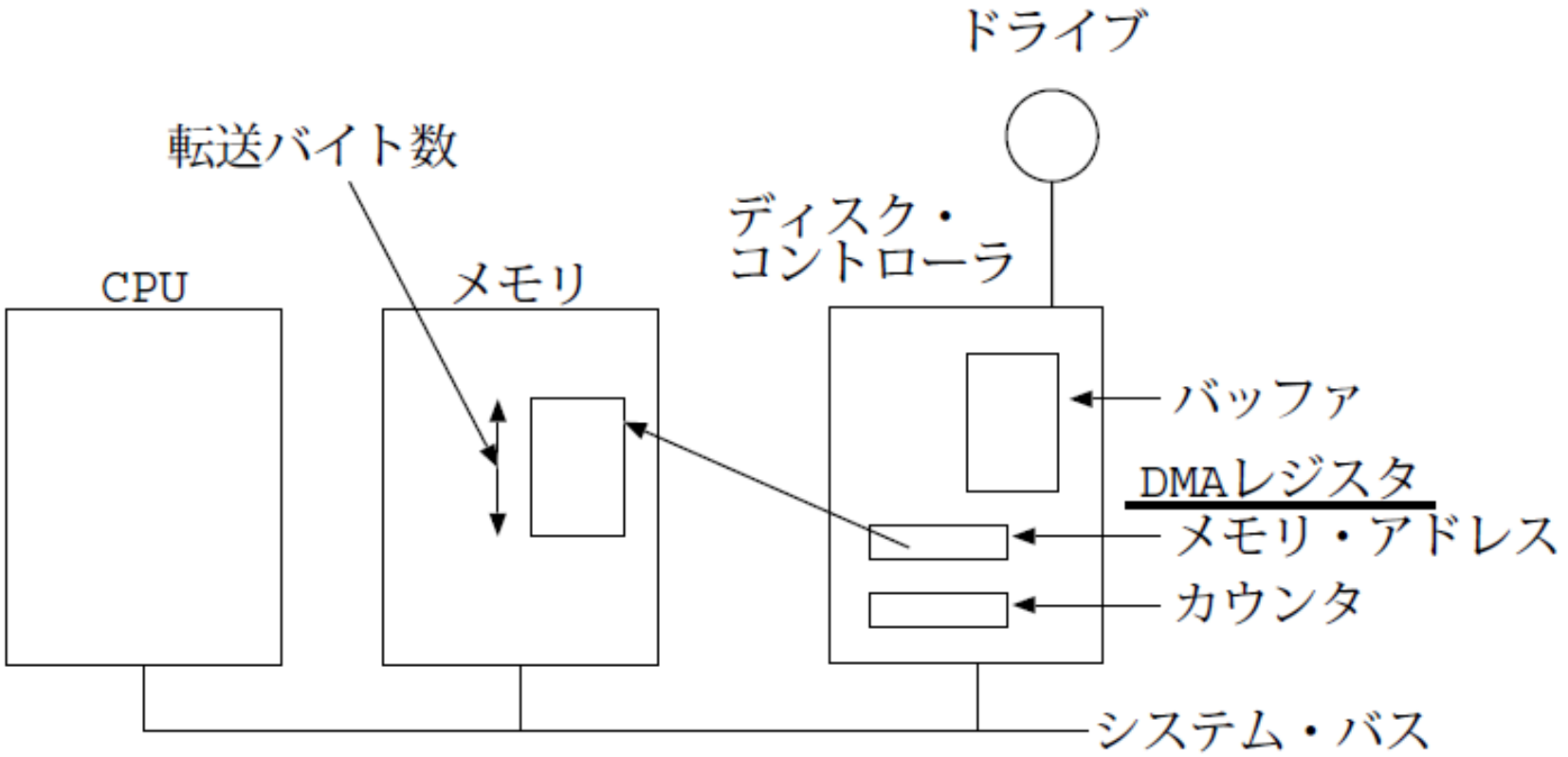
（メモリ・マップ型入出力 (memory-mapped I/O)）



メモリマップ入出力


メインメモリへのアクセス
（ロード命令、ストア命令）
データや状態の入出力
（入出力コマンド）
メインメモリ
プロセッサ
I/O空間
入出力プログラム
（デバイスドライバ）
制御指令用
データ用
入出力
コントローラ
入出力
装置
I/Oアドレス
内部装置

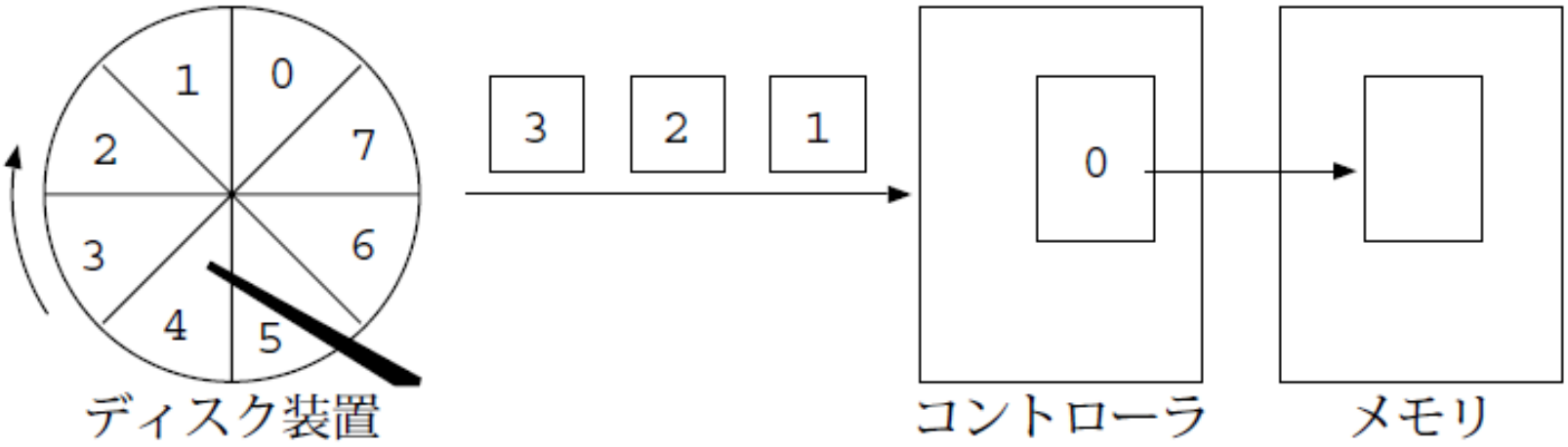
直接メモリ・アクセス(DMA; Direct Memory Access)



CPUからコントローラへ、{ディスク・アドレス、 メモリ・アドレス、 転送バイト数 } を送る。

直接メモリ・アクセス

ディスク装置は、コントローラの状態におかまいなしに、
一定の転送速度でデータ転送を行う。



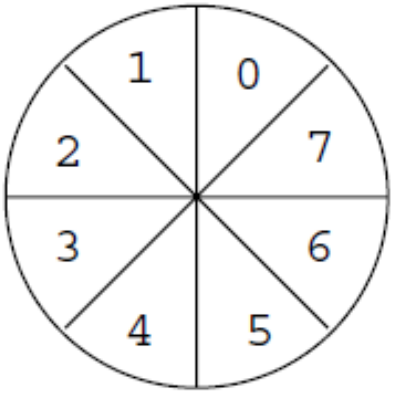
ブロック1が到着するが、ブロック0をメモリへ
転送中なので、受け取れない。← 失われる。

↑

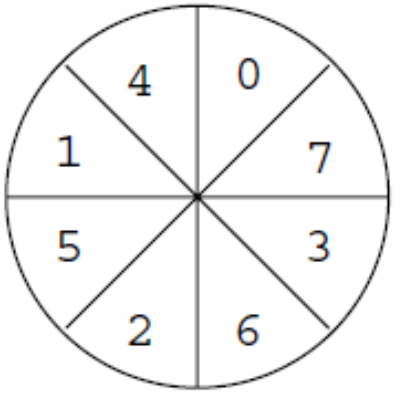
受け取るのに、次の回転まで待たねばならない。

インターリーブ (interleave)

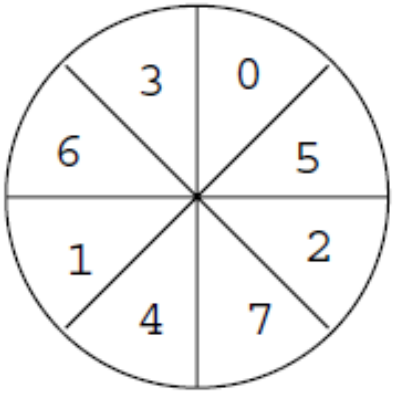
メモリ転送時間を「みこんで」、
ブロックをスキップして配置する。



インターリーブなし

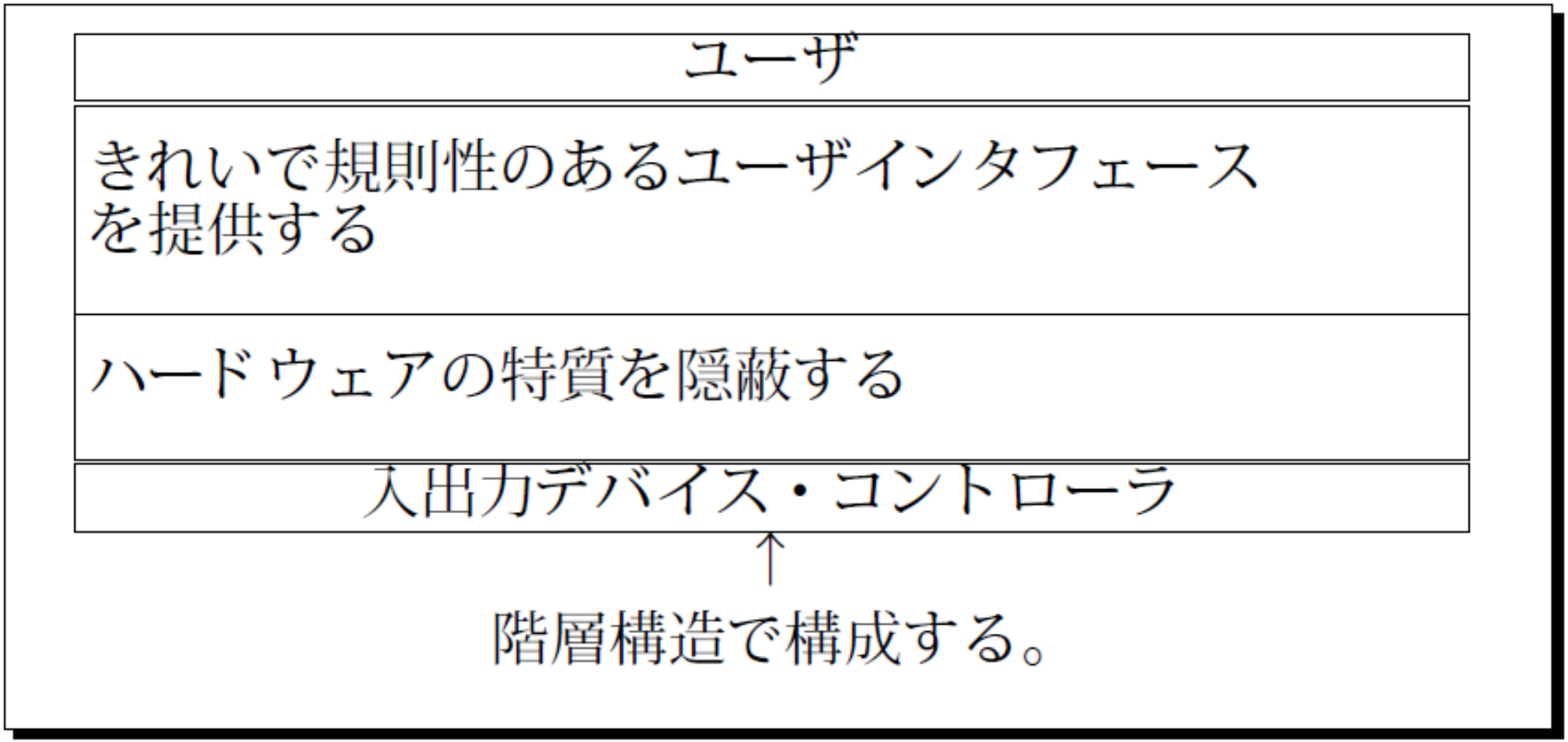


インターリーブ1



インターリーブ2

入出力ソフトウェア



階層構造で構成する。

入出力ソフトウェアの設計目標

装置非依存 (device independence)

フロッピーディスク上のファイルもハードディスク上のファイルも同じように扱いたい。

エラー処理

共有可能か占有か

複数ユーザが同じディスク上の別ファイルを同時にアクセスしても問題無い。
プリンタは、作業が終るまで1ユーザに占有される。

ユーザ・レベルのソフトウェア
装置非依存のOSソフトウェア
デバイス・ドライバ
割り込み処理ルーチン

1. 割り込み処理ルーチン
2. デバイス・ドライバ
3. 装置非依存のOSソフトウェア
4. ユーザ・レベルのソフトウェア

割り込み処理ルーチン

入出力操作を開始したプロセスを、
入出力完了の割り込みが発生するまで
ブロックさせておく。

例）プロセスは、セマフォに関してDOWNして、
自身をブロックする。

↓

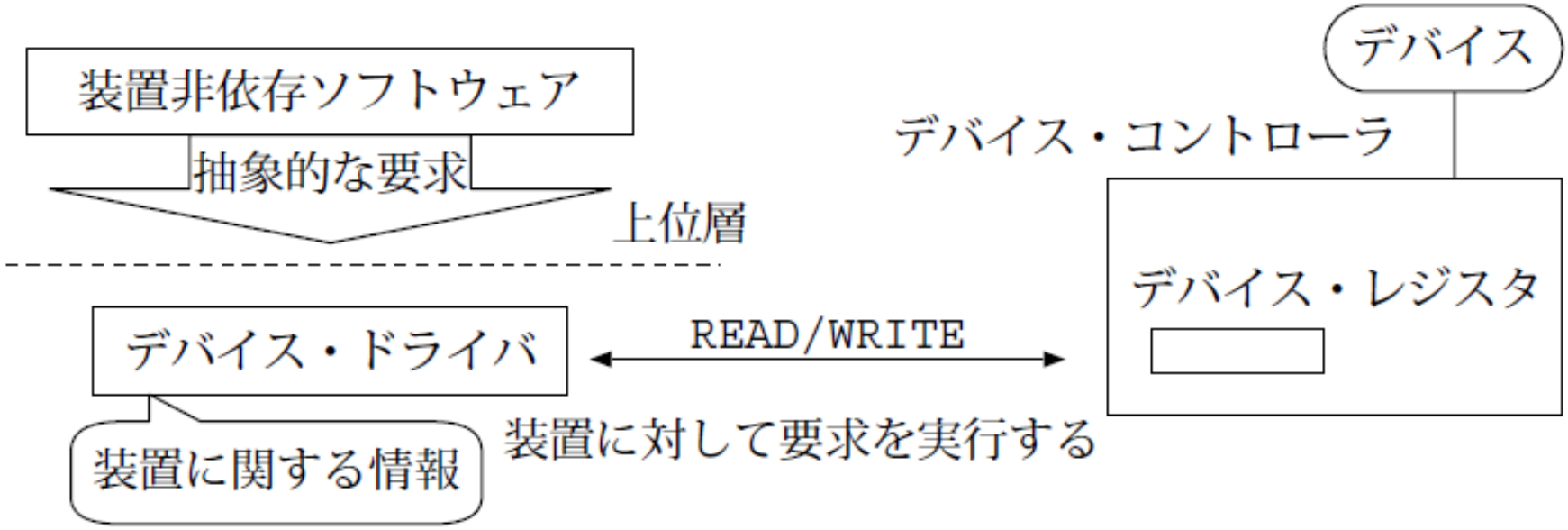
割り込みが発生

↓

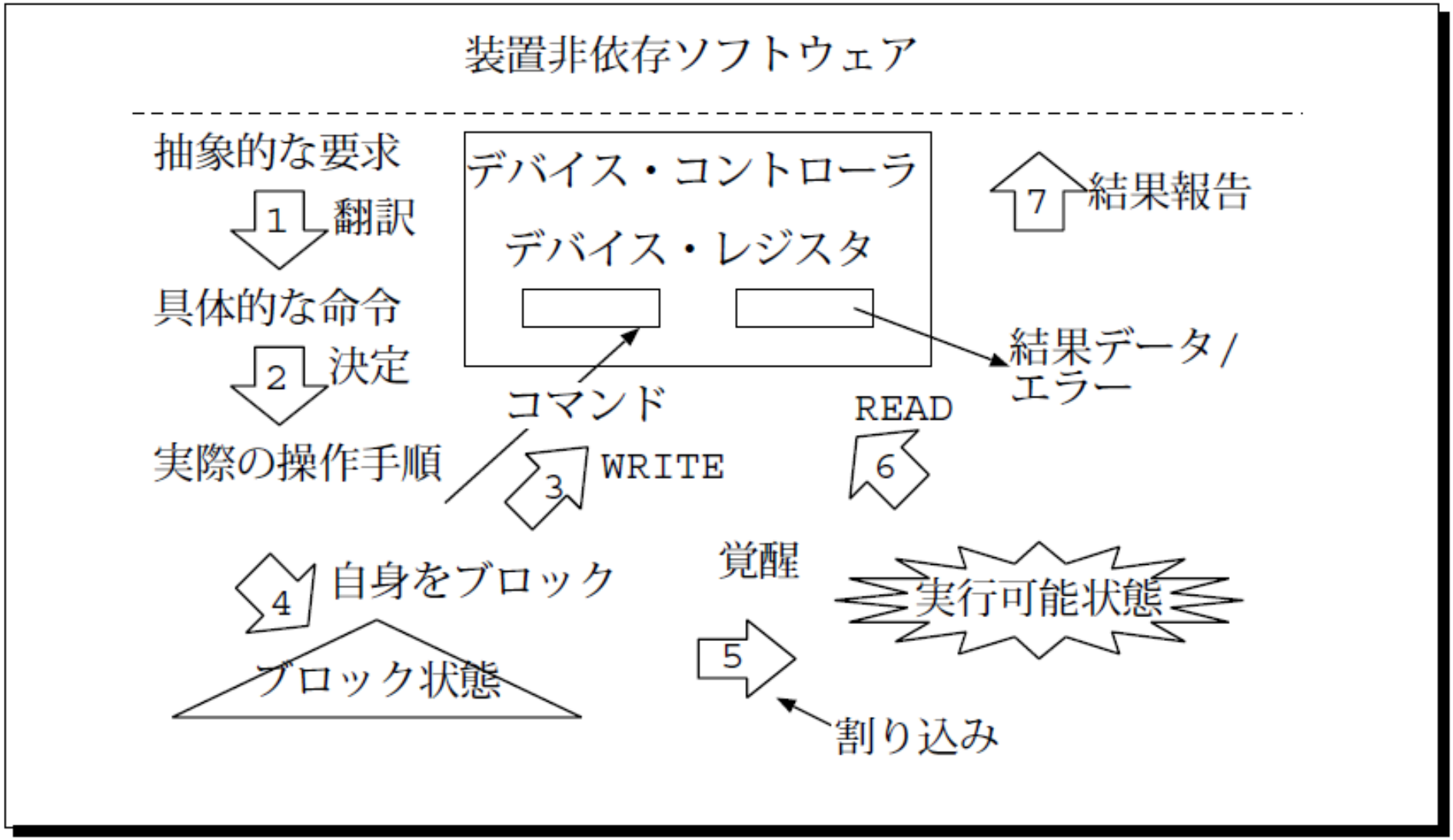
割り込み処理ルーチンは、セマフォをUPして
ブロックされたプロセスを実行可能にする。

デバイス・ドライバ

装置に依存するコードは、
すべてデバイス・ドライバ内にある。
1つのデバイス・ドライバは、1種類の装置あるいは
密接に関連する装置の1クラスを対象とする。



デバイス・ドライバ


装置非依存ソフトウェア
抽象的な要求
1
翻訳
具体的な命令
2
決定
実際の操作手順
デバイス・コントローラ
デバイス・レジスタ
コマンド
3
WRITE
4
自身をブロック
ブロック状態
5
覚醒
割り込み
READ
6
7
結果報告
結果データ/
エラー
実行可能状態

装置非依存入出力ソフトウェア

装置非依存ソフトとドライバとの境界は
システムに依存する。
↑
ドライバ内で行った方が効率が良い場合もあるため。

装置非依存ソフトウェアの基本的な機能とは、
すべての装置に共通する入出力機能を用意し、
ユーザ・レベルのソフトへ均質なインタフェースを
提供すること。

ユーザ空間の入出力ソフトウェア

入出力ソフトウェアのほとんどはOS内にあるが、
一部はライブラリとして提供される。

↑

ユーザ・プログラムとリンクされ、
カーネルの外で動作する。

例1) `count = write(fd, buffer, nbytes);`
ライブラリ内の手続きがリンクされ、そこからシステムコールが呼び出される。

例2) C言語の `printf`
書式指定に従って出力文字列を組み立て(ライブラリ内)、その後writeシステムコールで出力される。

ユーザ空間の入出力ソフトウェア

– スプーリング (spooling) –

マルチプログラミング・システムにおいて、
独占的な入出力デバイスを扱う一つの方法。

例） ラインプリンタ

↑

ユーザには特殊ファイルとして見える。

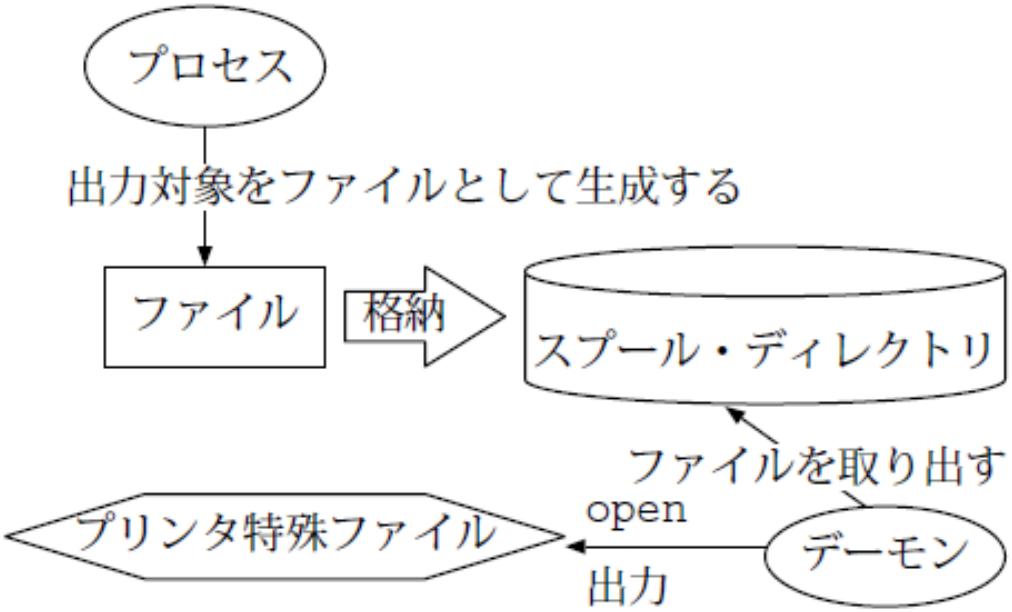
← 装置非依存ソフト

あるプロセスが、オープンしっぱなしにしたら ....

スプーリング・システム

デーモン (daemon) という特殊プロセス と

スプール・ディレクトリ (spooling directory) という特殊ディレクトリを用意する。

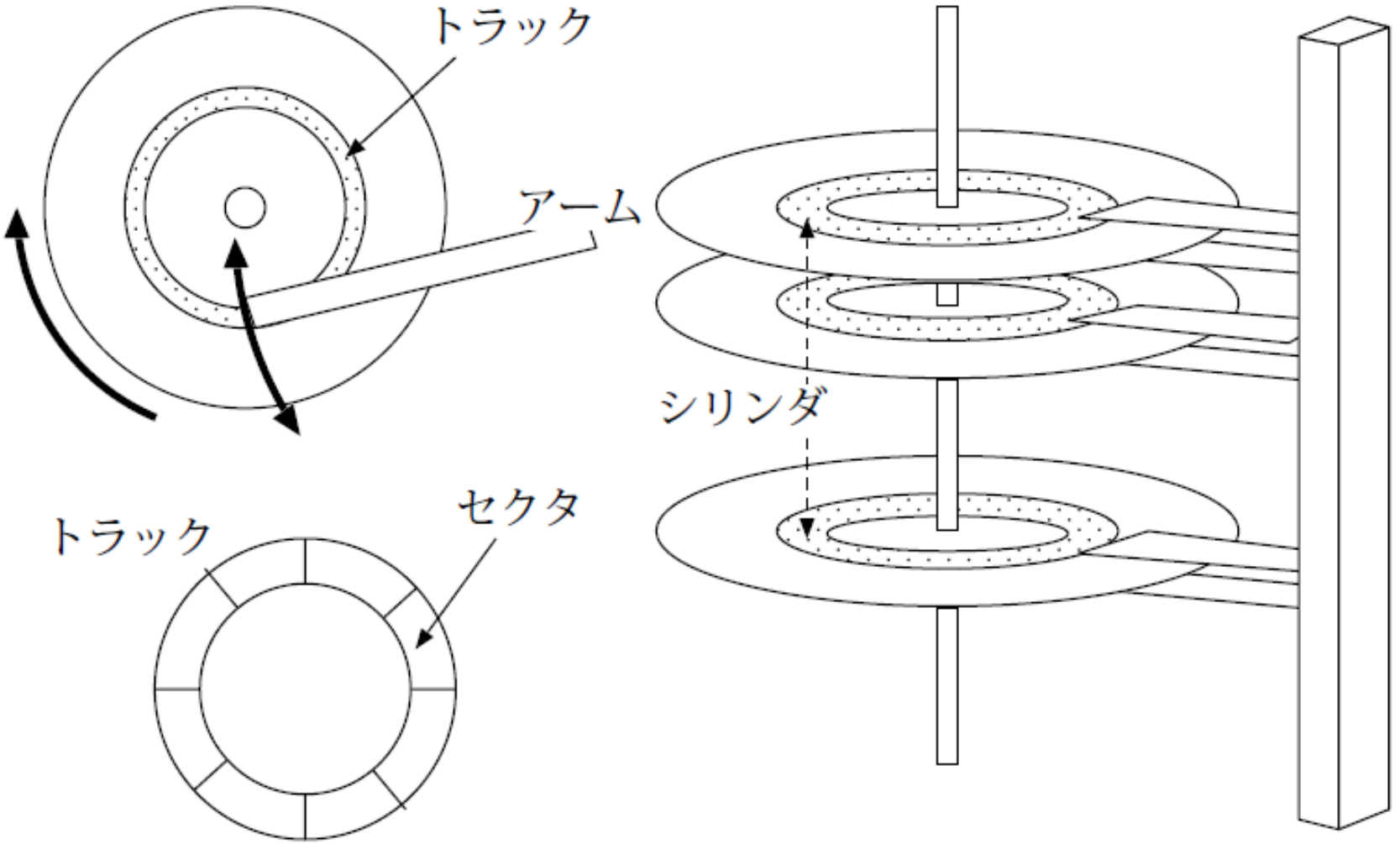


プリンタ特殊ファイルを扱えるのを、デーモンだけにする。

↑

ユーザが open しっぱなしにするのを防ぐ。

ディスク・ハードウェア



ディスクの性能

シーク時間：アームを適切なシリンダへ動かす時間

回転遅延：
　該当するセクタが回転してヘッドの下へ来るまでの時間

転送時間：ヘッドでの読み書きとコントローラへのデータ転送

ディスクアクセス時間の大半が、シーク時間
↓
平均シーク時間を減らすことで、
性能を大いに改善できる。

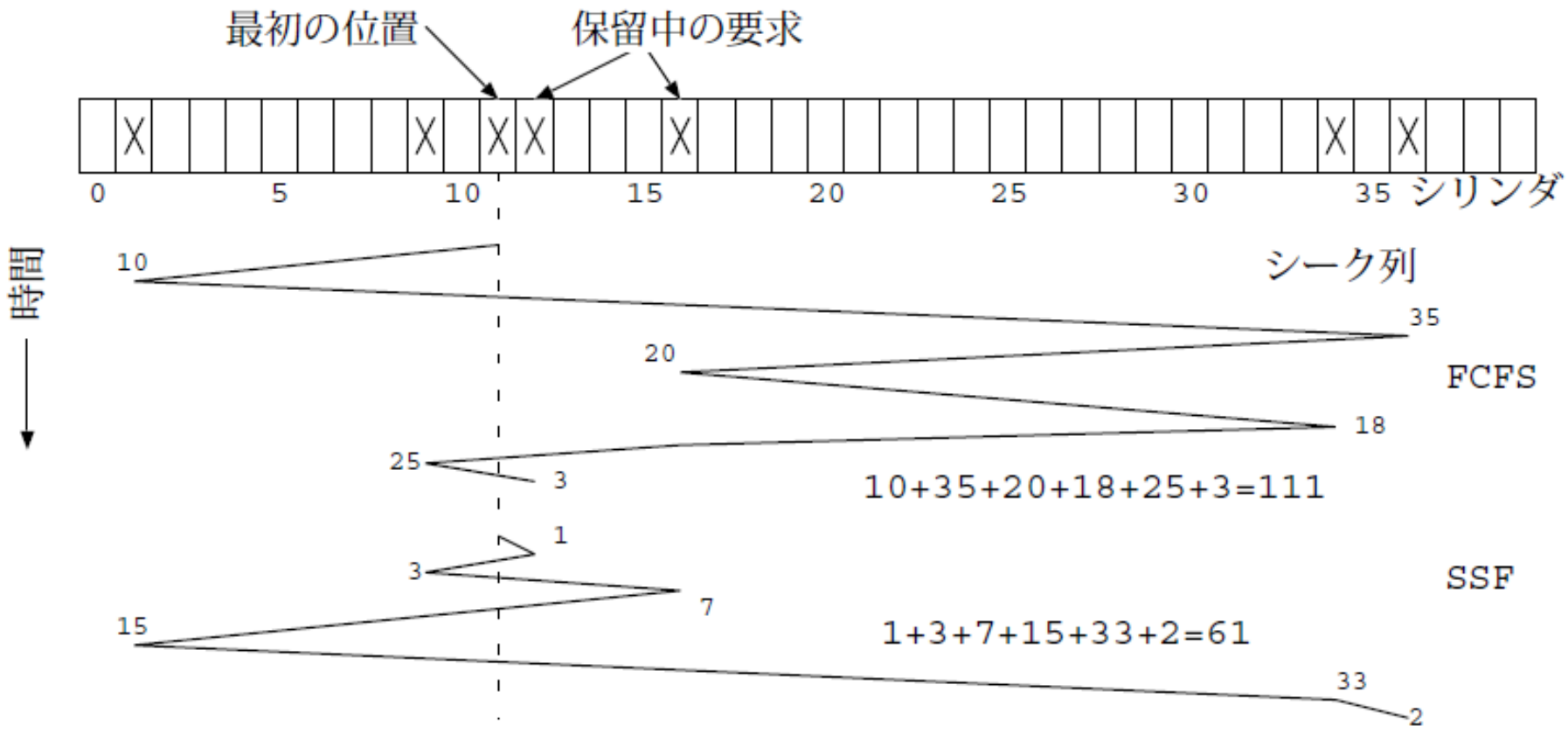
ディスク・アームのスケジューリング

到着順サービス (FCFS, First-Come First-Served)
一つずつ要求を受け、順番に処理する。

最短シーク順 (shortest seek first, SSF) アルゴリズム
シーク時間を最小にするために、距離の短い要求を処理する。

最短シーク順（SSF）アルゴリズム

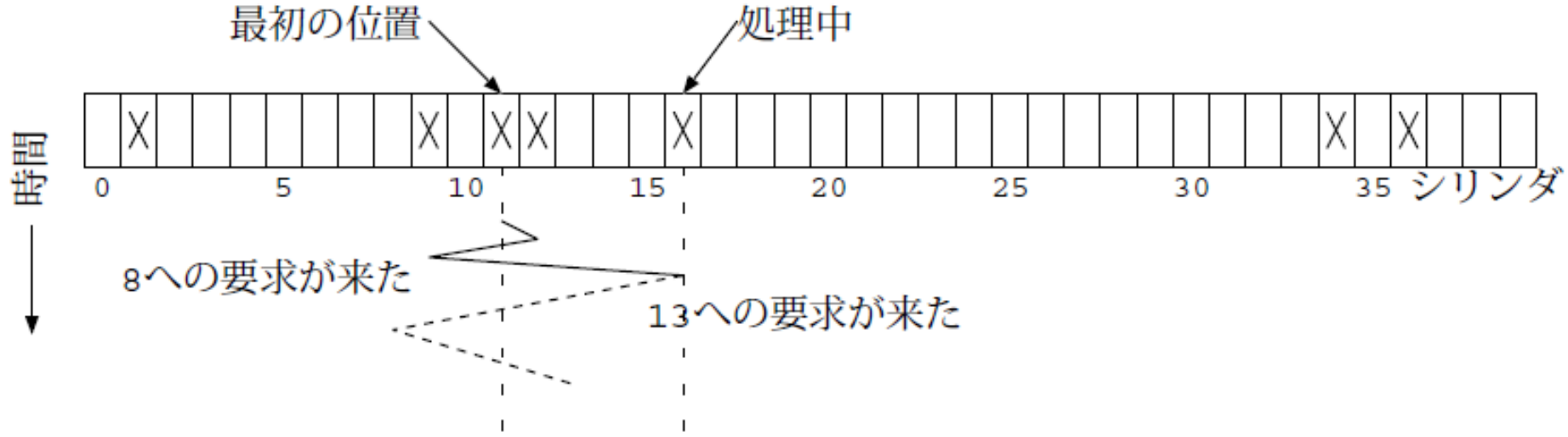
シリンダ 1, 36, 16, 34, 9, 12 の順に要求が到達する。



上のような要求の処理中に、
さらに多くの要求が到着し続けると ...

SSFアルゴリズムの問題点

シリンダ 16の処理中に、シリンダ8の要求が入り、...



ディスクが過負荷だと、
アームはディスクの中間地点ばかり指す傾向がある。
→ 中間付近の要求が無くなるまで、
両端への移動は待たされる。

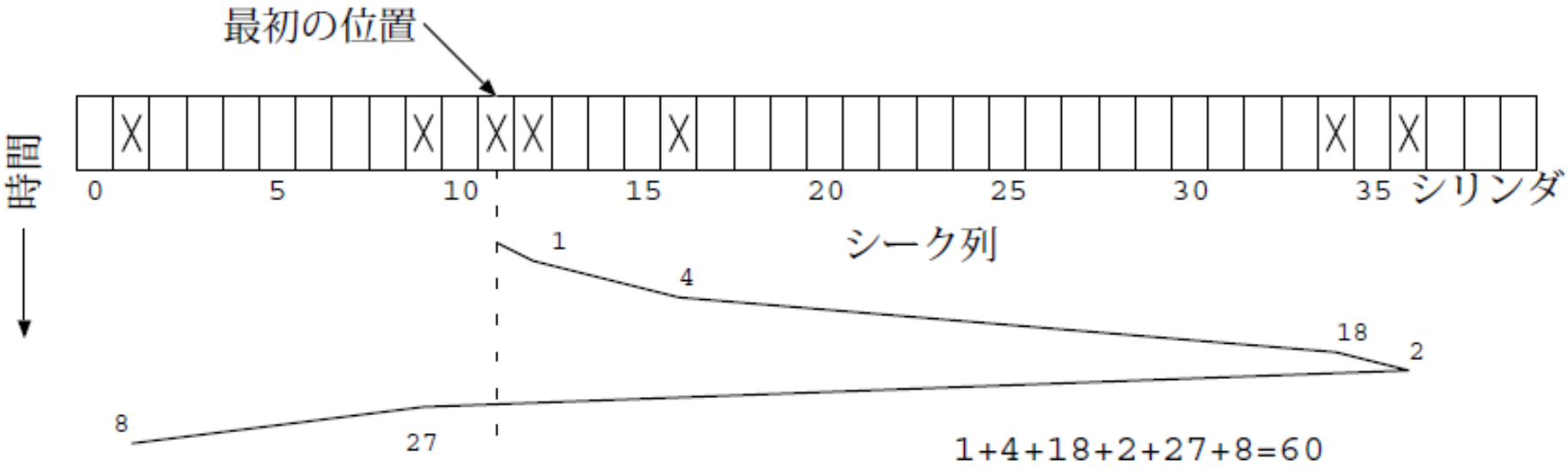
最小の応答時間 ⇔ 公正さ

高層ビルのエレベータも、同じトレードオフを扱う。

エレベーター・アルゴリズム(elevator algorithm)

(SCAN またはLOOK)

方向を決めて、
その方向への目立った要求がなくなるまで移動し、
その後方向を変える。



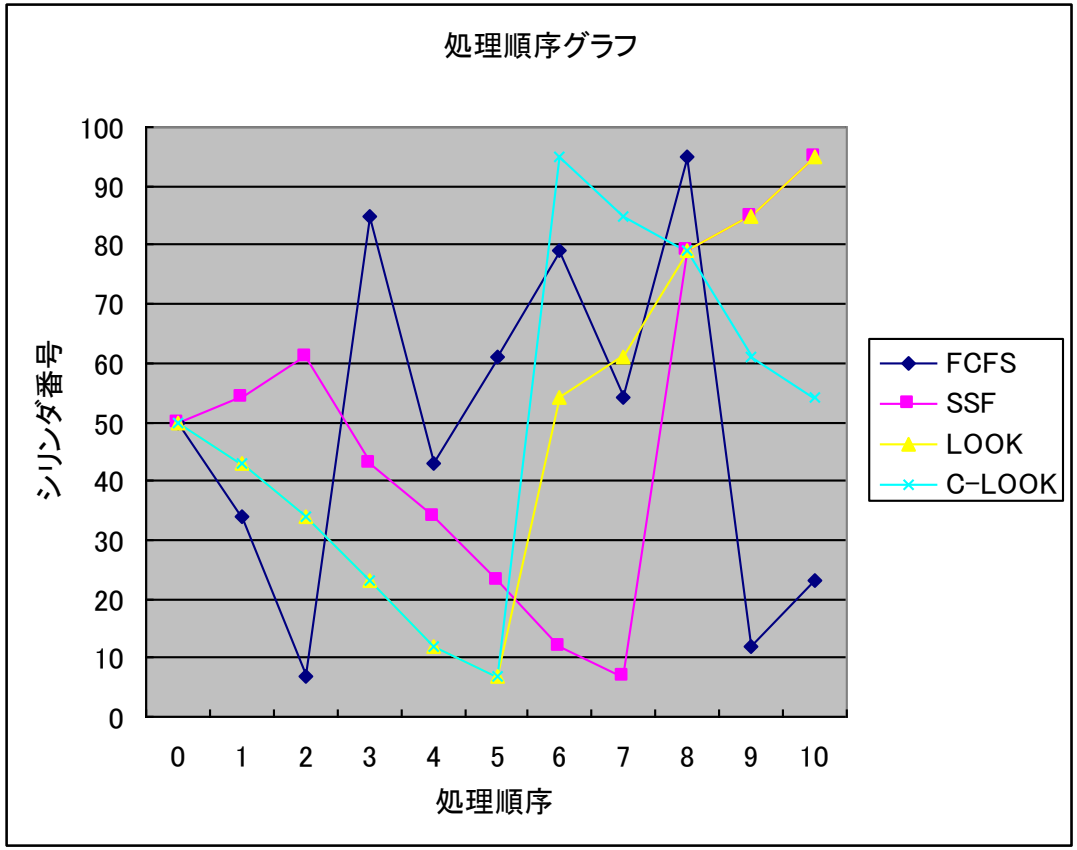
エレベーター・アルゴリズムの優れた特質

どんな要求列に対しても、移動距離の合計の上限は決まっている。

初期位置　50

待ち行列	34	7	85	43	61	79	54	95	12	23

	0	1	2	3	4	5	6	7	8	9	10
FCFS	50	34	7	85	43	61	79	54	95	12	23
SSF	50	54	61	43	34	23	12	7	79	85	95
LOOK	50	43	34	23	12	7	54	61	79	85	95
C-LOOK	50	43	34	23	12	7	95	85	79	61	54

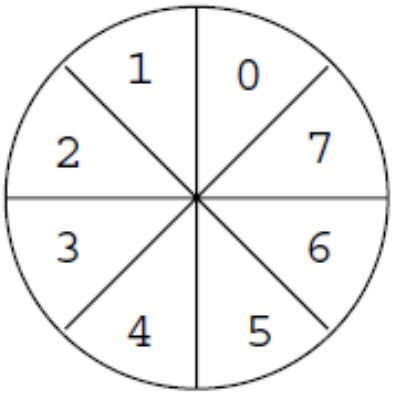


演習クイズ（第１２回）

- ディスクは図(c)のように係数２でインタリーブされている。1トラック当たり512バイト、セクタ数が８、回転速度３００rpmである。順番にトラック内のセクタを読み取る場合の所要時間を計算せよ。ただし、アームは正しい位置にあるとし、ヘッドの下にセクタ0が回転してくるのは1/2回転後であるとする。そのときの転送速度はいくつになるか。同じ条件で、インタリーブを行わない場合はどうなるか。インタリーブを行ったために速度はどの程度の割合で改善するか。

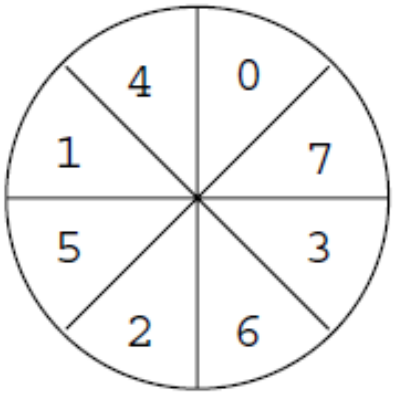
インターリーブ (interleave)

メモリ転送時間を「みこんで」、
ブロックをスキップして配置する。



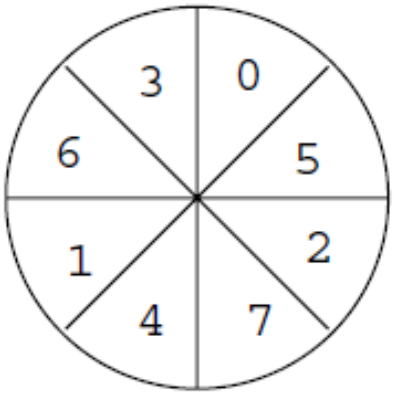
インターリーブなし

(a)



インターリーブ1

(b)



インターリーブ2

(c)

演習クイズ（第１２回）

- インタリーブ2を行う場合
  - 300 rpmより、５回転／秒、　　m sec ／１回転
  - 1セクタを読み取る時間　=　　msec / 8 =　　msec
  - 1セクタ読み次のセクタに移動する時間　= 25+50 = 75 msec
    (2セクター読み飛ばす50 msecの間にメモリへの転送を行っている）
  - 8セクタ読み取る時間　=　　×８ msec =　　msec
  - セクタ待ち時間（1/2回転）　　msec を足して　読み取り時間=　　msec
  - 転送速度　=　　バイト/　　msec =　　バイト／秒
- インタリーブをおこなわない場合
  - 結局　7回転＋αすることになるので　　msec ×７＋75 msec =　　msec
  - セクタ待ち時間（1/2回転）　　msec を足して　読み取り時間=　　msec
  - 転送速度　=　　バイト/　　msec =　　バイト／秒
- 速度の改善割合
  - 　　／　　=　　倍となる

演習クイズ（第12回）

- 10, 22, 20, 2, 40, 6, 38のシリンダ順でディスクへの要求あり。シリンダを一つ移動するのに必要なシーク時間は、6ミリ秒である。次に示す処理のシーク時間を求めよ。
  (a) 先着順サービス（FCFS）
  (b) 最短シーク順（SSF）
  (c) エレベータ・アルゴリズム（LOOK）（シリンダ番号の大きい方向から）

いずれの場合も最初のアームに位置は20シリンダとする

演習クイズ（第12回）

(a) 先着順サービス（FCFS）

(b) 最短シーク順（SSF）

(c) エレベータ・アルゴリズム