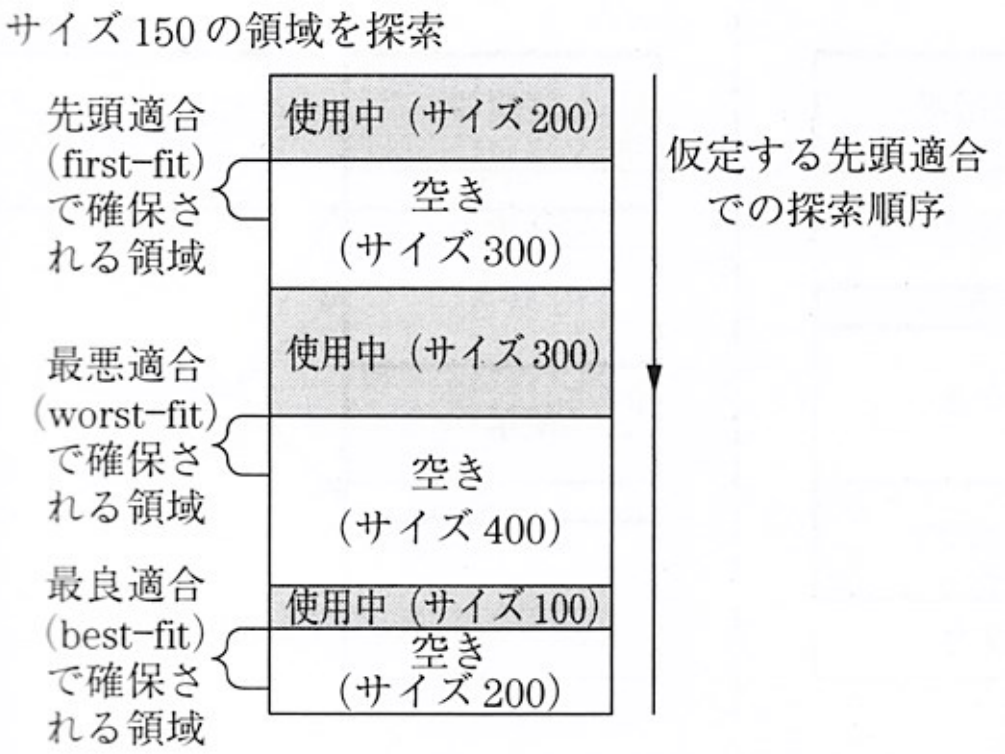
システムソフトウェア講義の概要

1. 計算機システムの復習：中央演算処理装置(CPU)，プログラムの実行，主記憶装置，補助記憶装置
2. 時分割処理：プロセス，スレッド，スケジューリング
3. スレッド間の排他制御：フラグ，セマフォ，モニタ，デッドロック
4. デバイス管理，HDDへのアクセス制御
5. 記憶管理：メモリ割り当て，ページング，セグメンテーション
6. 仮想記憶とファイルシステム
7. 演習問題
8. プログラミングシステムの概要，文法とそのクラス，字句解析と正規文法
9. 正規表現からの非決定性オートマトンの生成、決定性オートマトンへの変換
10. 字句解析用オートマトン生成ソフトウエアの実際
11. 構文解析と導出，文脈自由文法の構文解析法：LL構文解析
12. 文脈自由文法の構文解析法：LR構文解析
13. コンパイラ-コンパイラと構文解析の実際
14. 演習問題
15. 講義の総括と試験

記憶領域管理（復習）

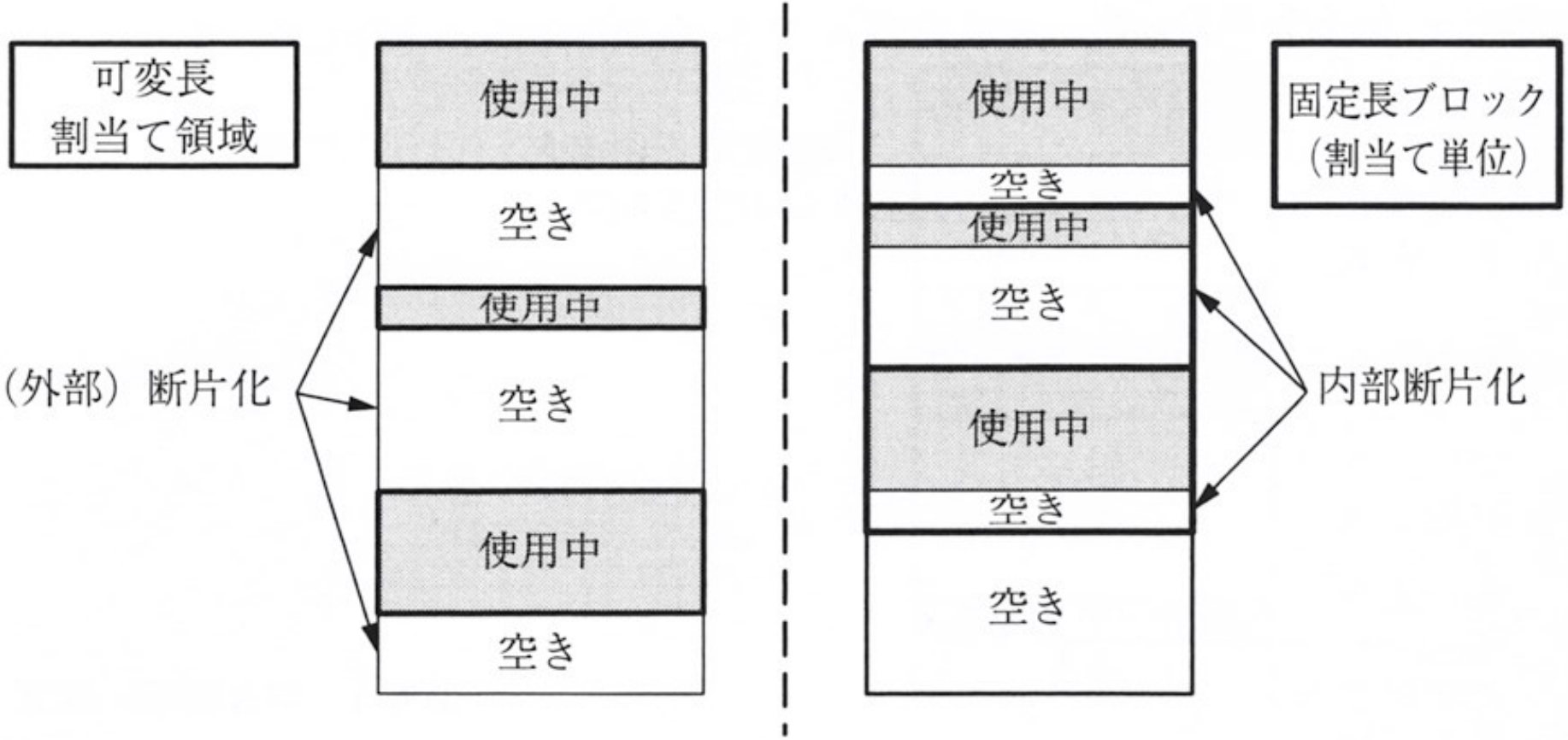
メモリの動的確保

- プログラム動作中に，メモリが必要になった場合，これを確保する事が出来る.
- メモリには番地がつけられており，連続する番地のメモリ領域が必要となる.
- 可変長管理の方法
  - 先頭適合
  - 最悪適合
  - 最良適合
- 不要になれば返却する.
- 最悪と最良では空き領域をソートしておく必要あり.



フラグメンテーション（断片化）

- メモリの確保と開放を繰り返すと，要求される大きさの連続したメモリ領域が確保できなくなる．



固定長管理

- 固定サイズのメモリ領域を単位としてメモリの割当を行う.
- 割り当て単位が空いている限り，割り当て単位以下の割当要求が成功する.
- 小さな断片が発生しにくい.

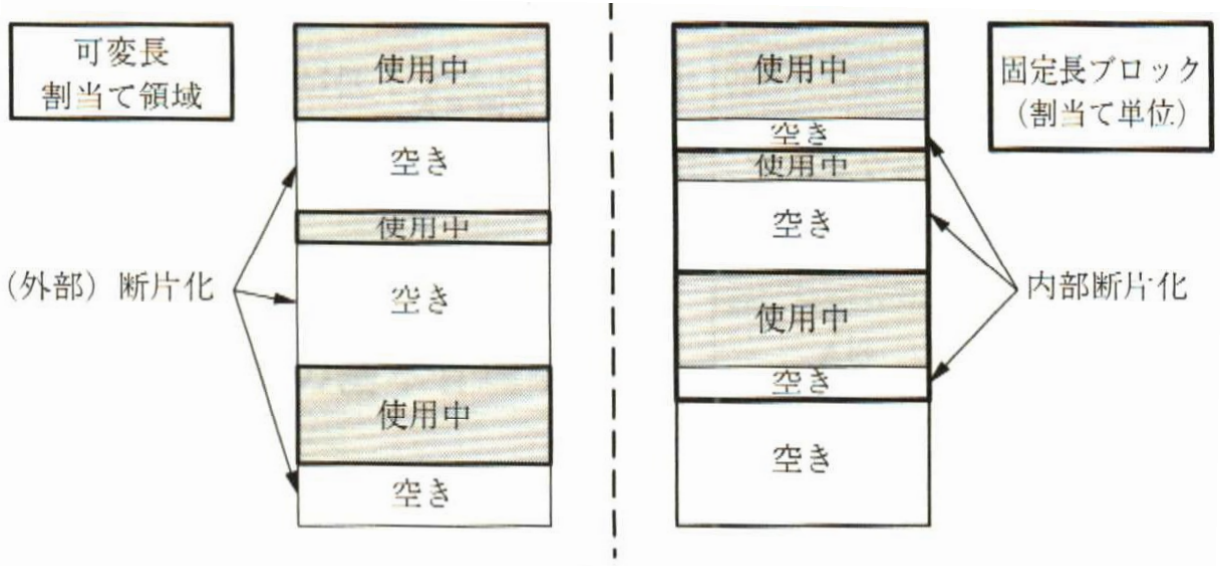
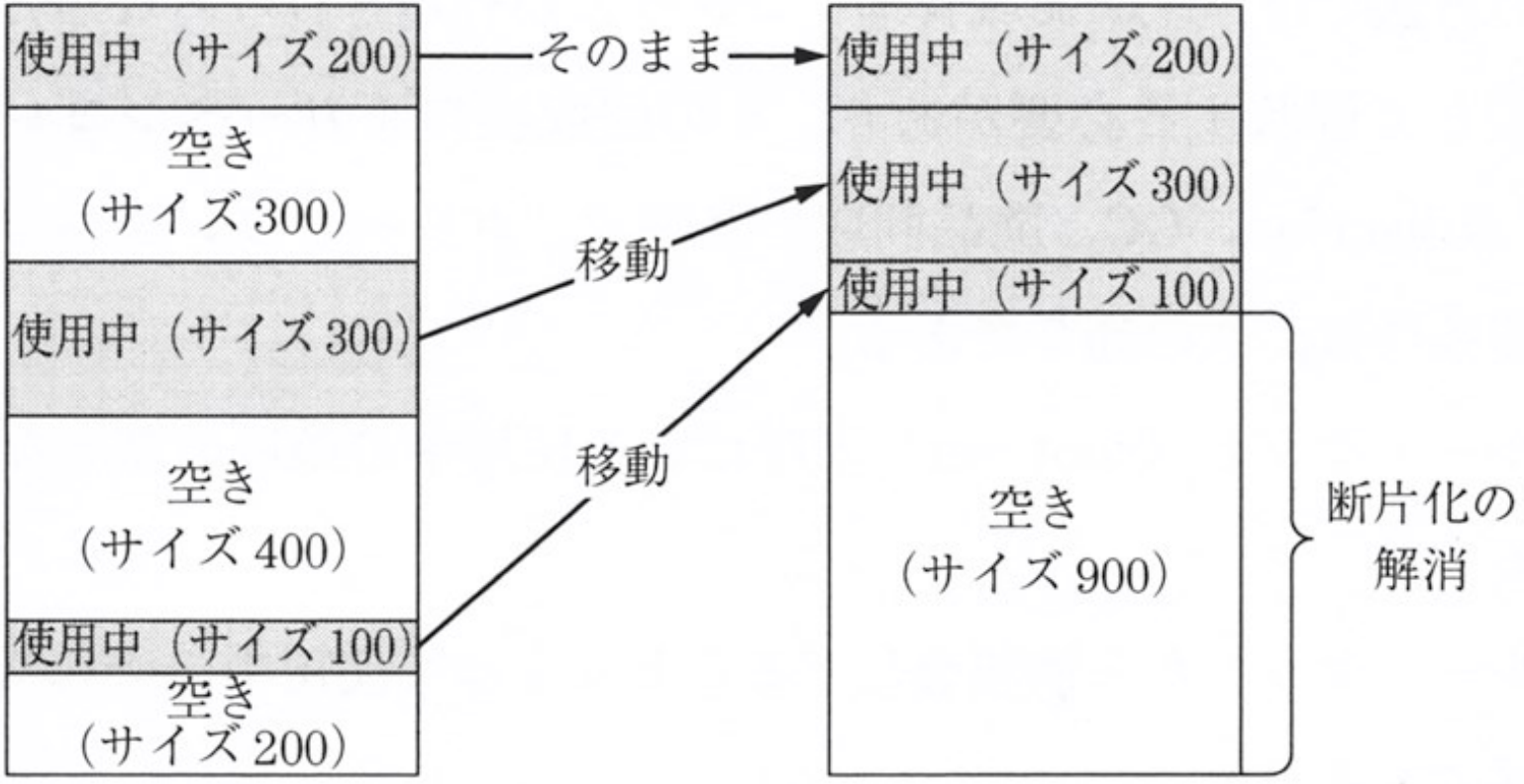


図 5.2 断 片 化

コンパクション

- 時間はかかるが，使用中の領域をまとめることで，空き領域の大きさを拡げられる．Linuxではメモリ確保に失敗した後，これが行われる．



メモリ管理ユニット
Memory Management Unit

MMU
の話

MMUを用いたページング

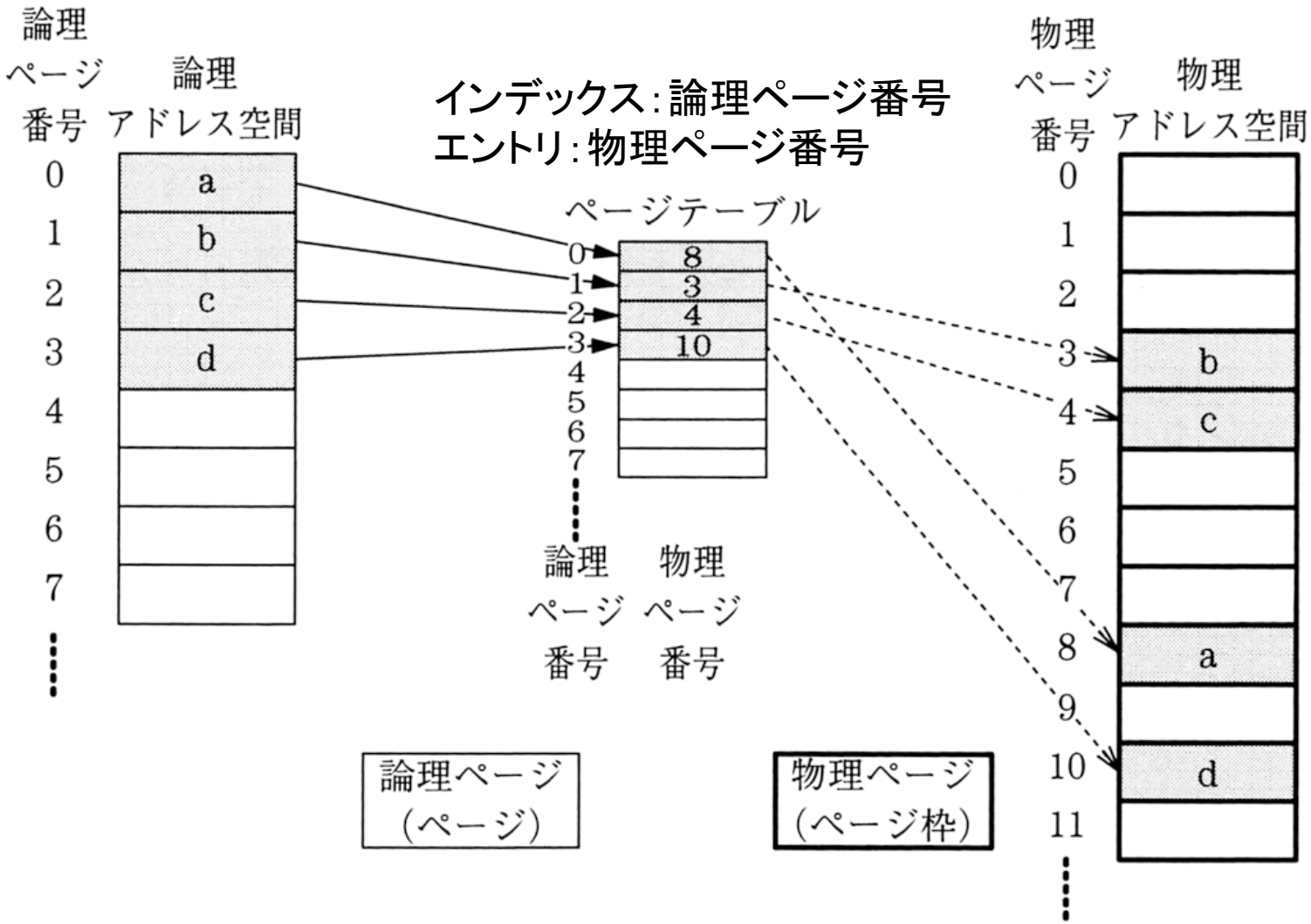
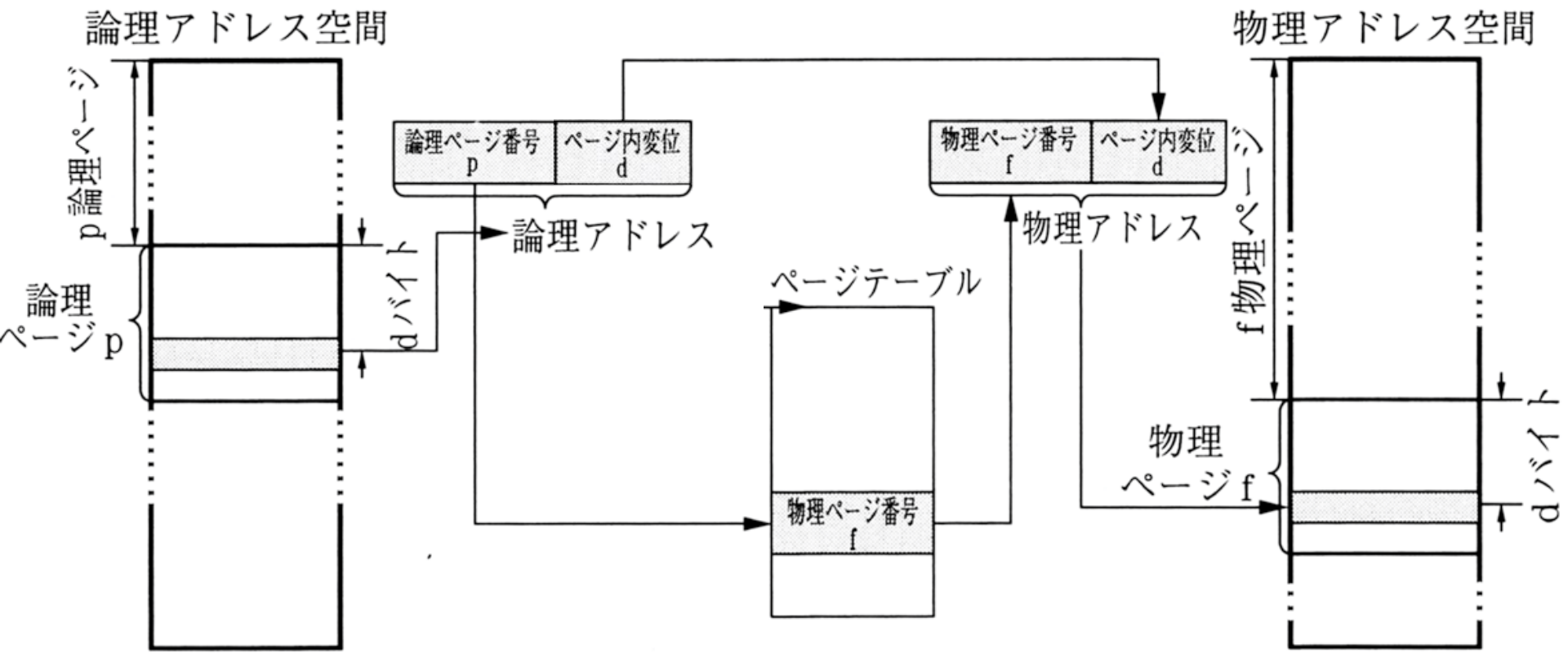


図 5.6　ページテーブルによるマッピング

ページングによる単純なアドレス変換

- ページ番号とページ内変位を与えて，メモリにアクセスするが，これは一つのアドレスと見える．



ページテーブルの個数

- アドレス空間が32bit の場合
  - 論理アドレス空間：2^{32}
  - ページサイズ：16KB=2^{14}
  - ページテーブル：$2^{32} \div 2^{14} = 2^{18}$=262144個
  - 1エントリ4byteなら，テーブルは2^{20}=1MB必要
- アドレス空間が64bit の場合
  - 論理アドレス空間：2^{64}
  - ページサイズ：16KB=2^{14}
  - ページテーブル：$2^{64} \div 2^{14} = 2^{50}$個
  - 1エントリ4byteなら，テーブルは2^{52}=4096TB=4PB必要

アドレス空間が広くても搭載されている物理メモリ量がそれだけなくても良い.

多段ページテーブル

- ページテーブルのサイズ縮小のため

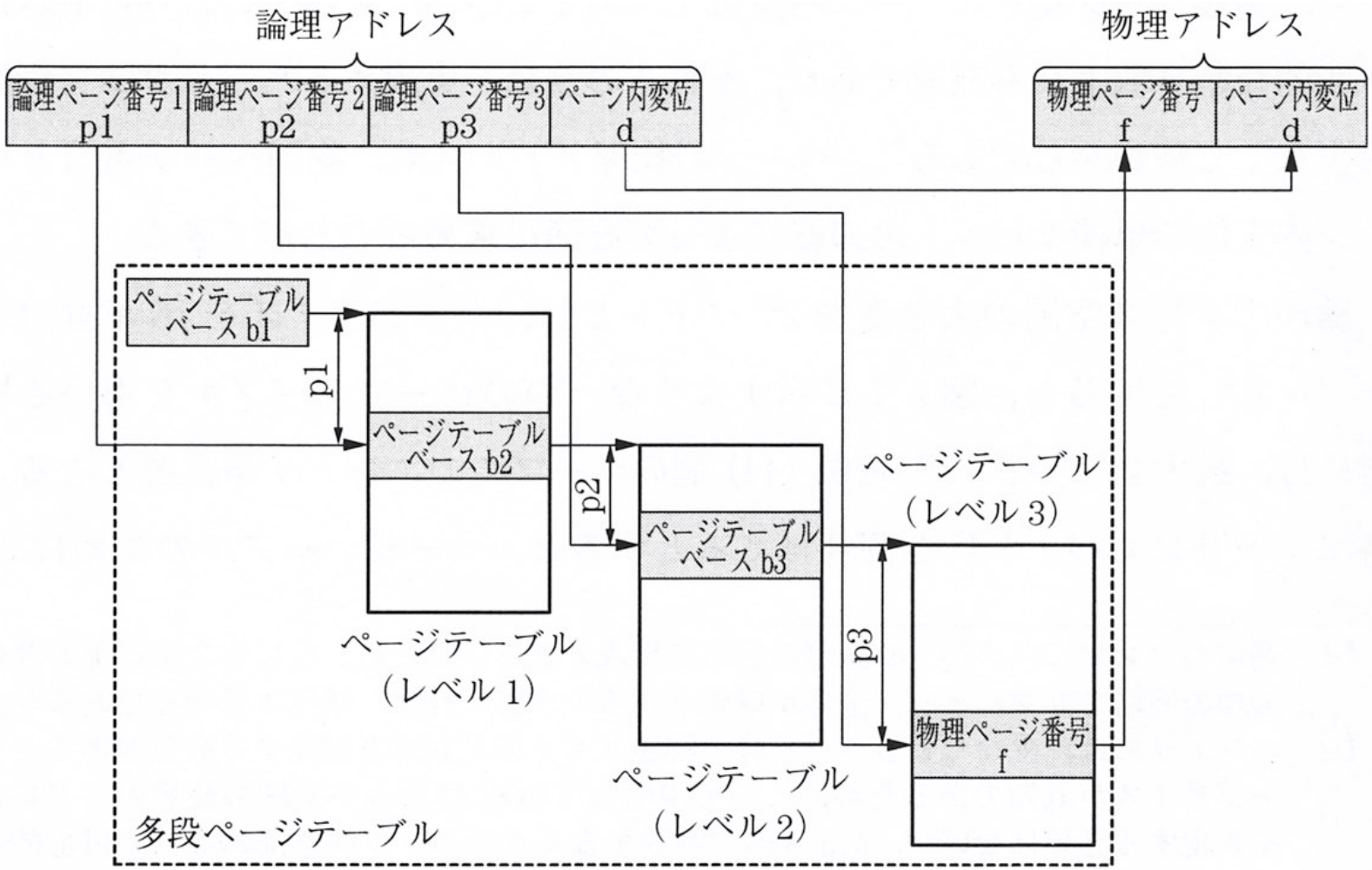
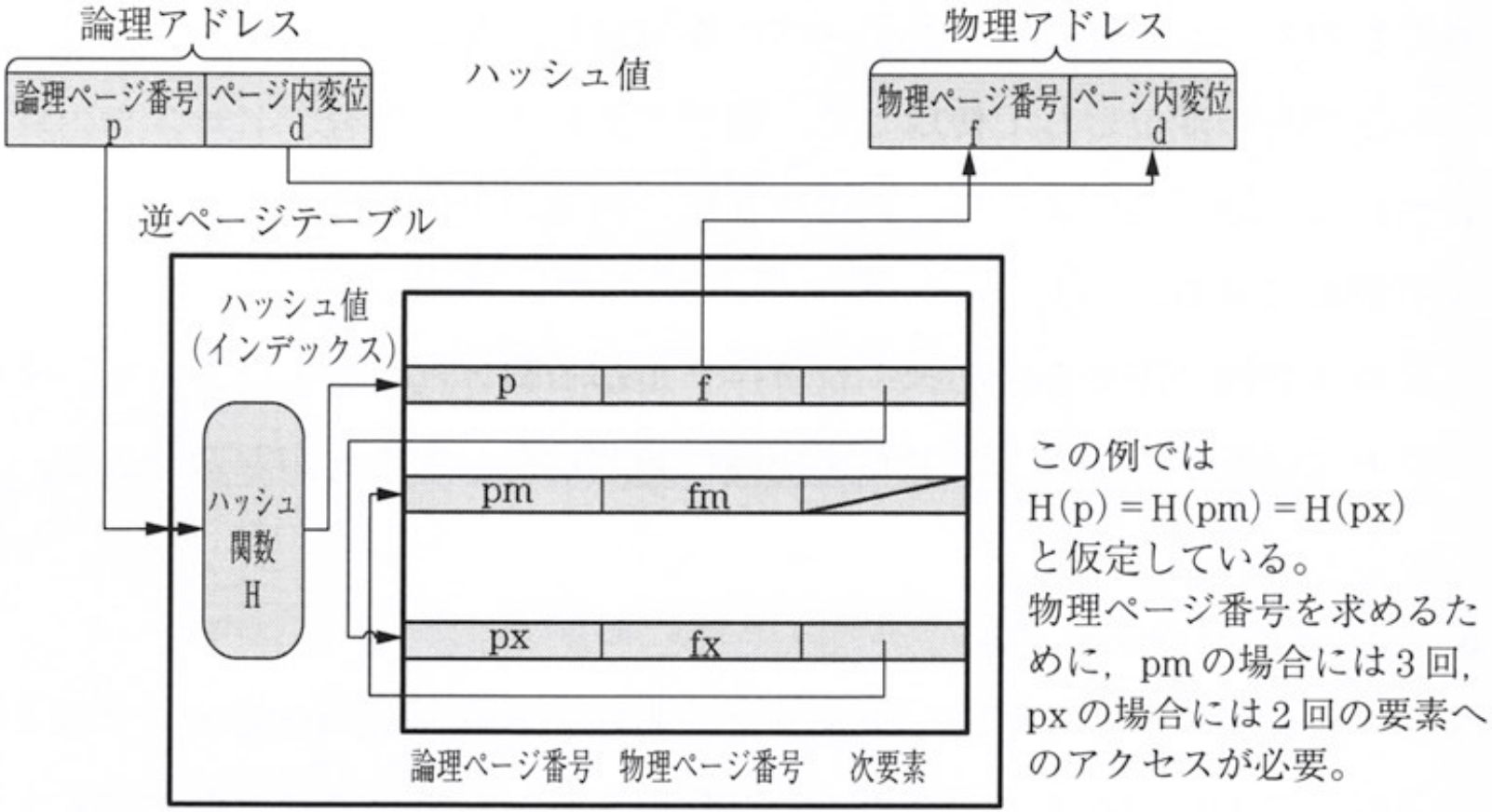


図5.8 多段ページテーブル

逆ページテーブル：ハッシュ関数の利用

- ページテーブルの大きさが物理アドレスの大きさに依存する．さらに，高速でもある．



さらなる高速化:アドレス変換キャッシュ Translation Look aside Buffer: TLB

- 一旦，アドレス変換した内容を，CPU内のメモリに保存しておく.

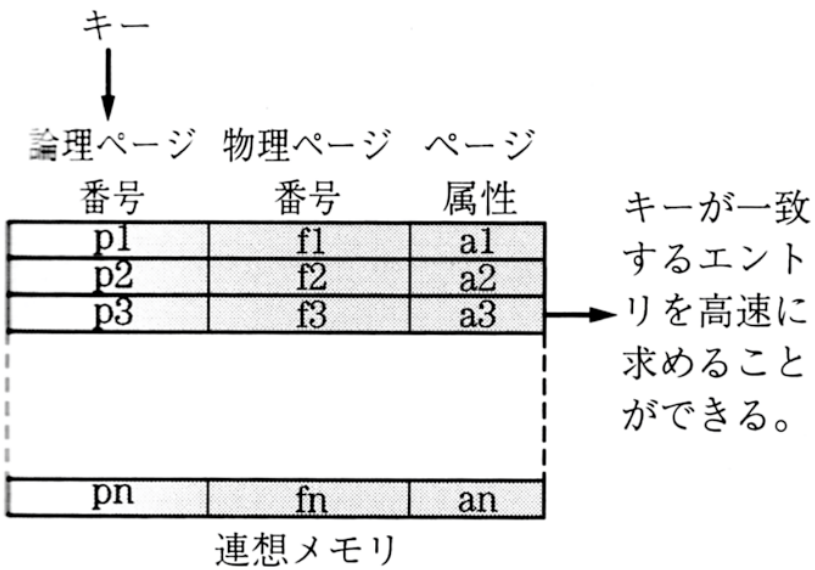


図 5.10 TLB（アドレス変換キャッシュ）

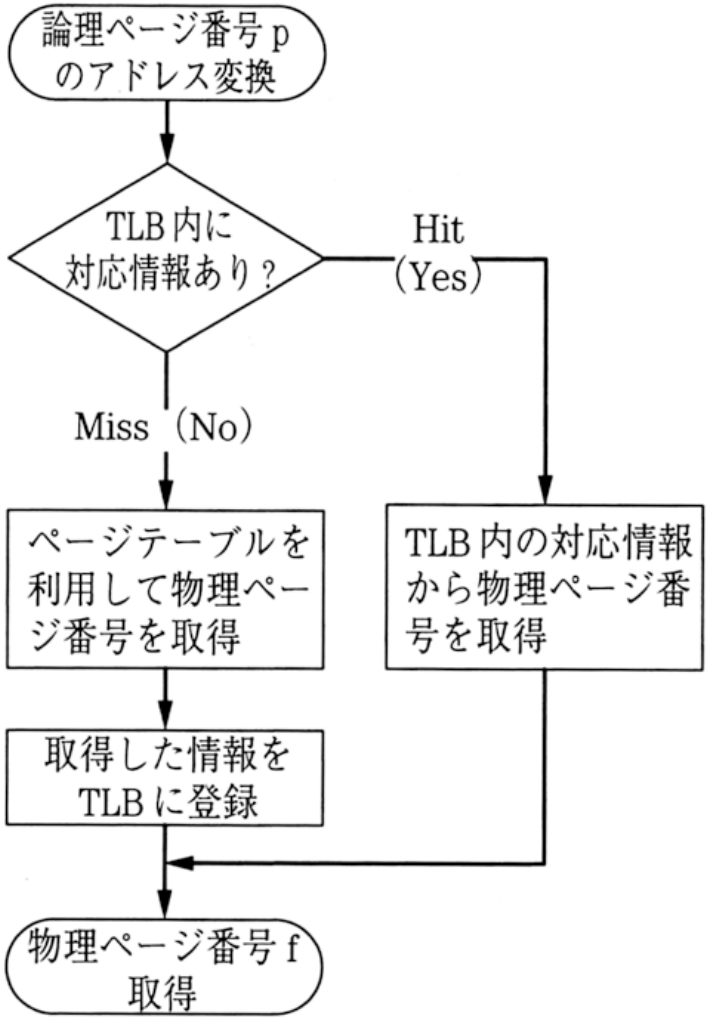


図 5.11 TLB を利用したアドレス変換の流れ

仮想記憶

物理メモリよりも広いメモリ空間の利用

プロセス・スワッピング1

- 優先度の低いプロセスのメモリをHDD上のバッキングストア領域に退避させ，優先させたいプロセスに空いたメモリ領域を割り当てる.

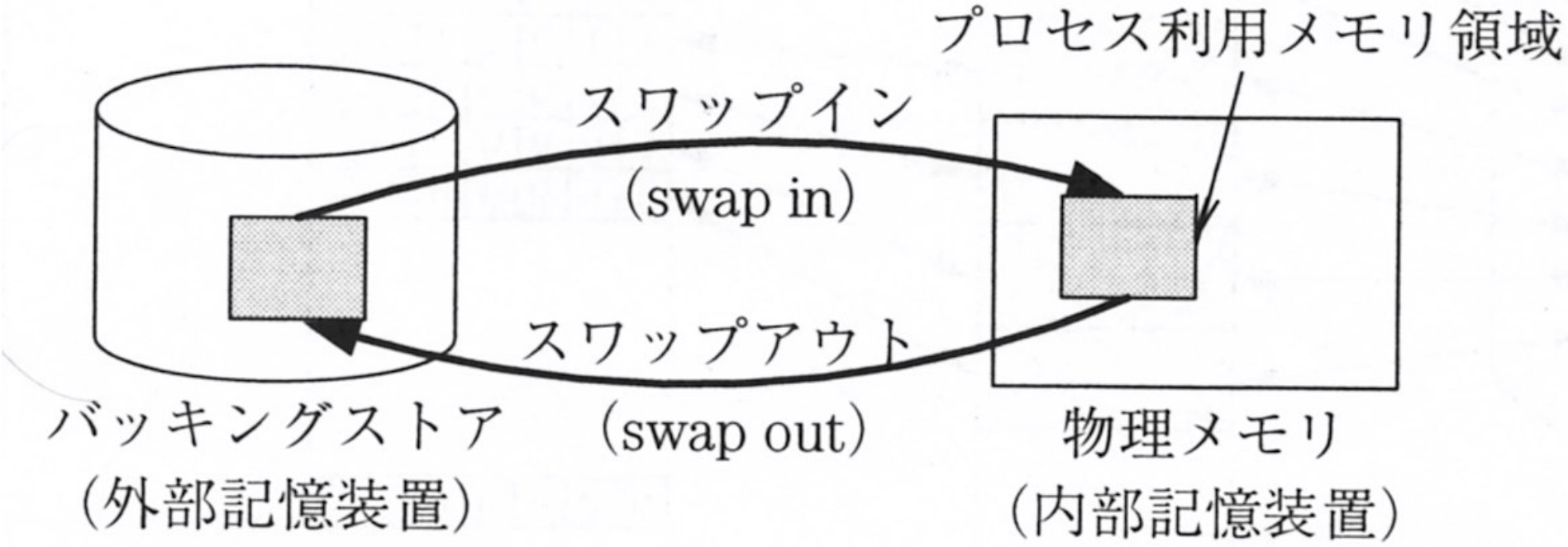


図 6.2　プロセス・スワッピング

デマンドページング

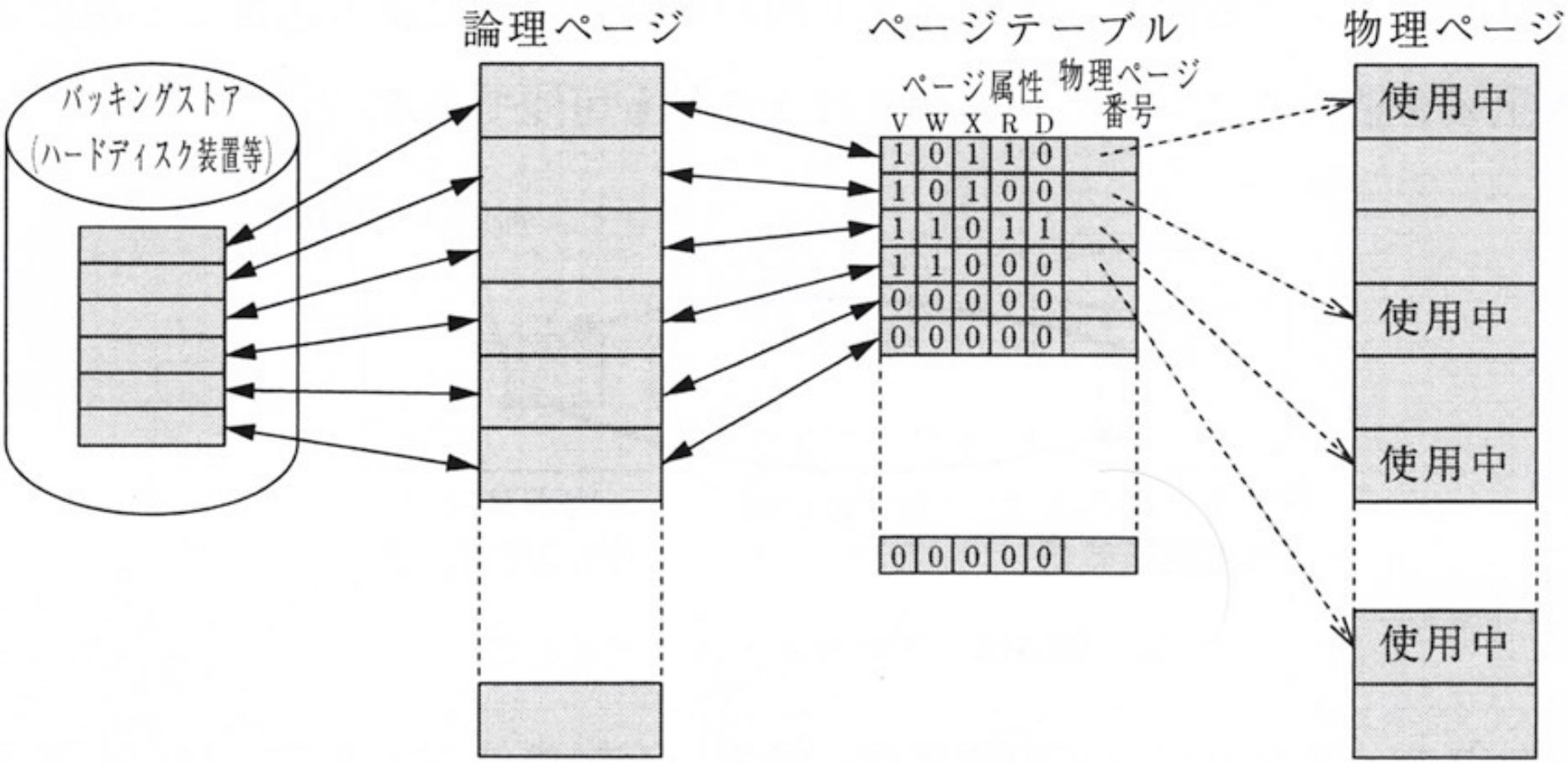


図 6.3　デマンドページングにおける記憶領域の関係

常に物理メモリに空きを作っておく

- ページアウト・デーモン
  - 不要なページをバッキングストアに退避させる
- 退避優先順位決定アルゴリズム
  - FIFO: 最も古いページの置き換え優先度を高くする.
  - OPT：最も将来利用されないページの置き換え優先度を高くする.（理想：未来はわからないので）
  - LRU：最も長い期間使われていないページの置き換え優先度を高くする.

FIFOとBeladyの異常

- 物理ページメモリが多いほうが，ページフォールトが多数発生することがある. (Beladyの異常)

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
ページフォールト	×	×	×	×	×	×	×	×	×				×	×	
物理ページ数＝4	0	0	0	0	1	2	3	4	0	0	0	0	1	2	2
		1	1	1	2	3	4	0	1	1	1	1	2	5	5
			2	2	3	4	0	1	2	2	2	2	5	3	3
				3	4	0	1	2	5	5	5	5	3	4	4

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
ページフォールト	×	×	×	×	×				×	×	×	×	×	×	×
物理ページ数＝5	0	0	0	0	0	0	0	0	1	2	3	4	5	0	1
		1	1	1	1	1	1	1	2	3	4	5	0	1	2
			2	2	2	2	2	2	3	4	5	0	1	2	3
				3	3	3	3	3	4	5	0	1	2	3	4
					4	4	4	4	5	0	1	2	3	4	5

図 6.8　FIFO の置換え例

OPTアルゴリズム

- 理想的であった場合の話. MINアルゴリズムとも呼ばれる.

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
ページフォールト	×	×	×	×	×				×				×	×	
物理ページ数＝4	0	0 1	0 1 2	0 1 2 3	0 1 2 4	0 1 2 4	0 1 2 4	0 1 2 4	0 1 2 5	0 1 2 5	0 1 2 5	0 1 2 5	3 1 2 5	3 4 2 5	3 4 2 5

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
ページフォールト	×	×	×	×	×				×					×	
物理ページ数＝5	0	0 1	0 1 2	0 1 2 3	0 1 2 3 4	0 1 2 3 4	0 1 2 3 4	0 1 2 3 4	0 1 2 3 5	0 1 2 3 5	0 1 2 3 5	0 1 2 3 5	0 1 2 3 5	4 1 2 3 5	4 1 2 3 5

図 6.9　OPT の置換え例

Least Recently Used: LRU(linux採用)

- 最も長い期間使われていないものを置き換える.

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
ページフォールト	×	×	×	×	×	×	×	×	×				×	×	×
物理ページ数＝4　最も古い	0	0	0	0	1	2	3	4	0	1	2	5	0	1	2
		1	1	1	2	3	4	0	1	2	5	0	1	2	3
			2	2	3	4	0	1	2	5	0	1	2	3	4
最も新しい				3	4	0	1	2	5	0	1	2	3	4	5

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
ページフォールト	×	×	×	×	×				×				×	×	×
物理ページ数＝5　最も古い	0	0	0	0	0	1	2	3	4	4	4	4	5	0	1
		1	1	1	1	2	3	4	0	1	2	5	0	1	2
			2	2	2	3	4	0	1	2	5	0	1	2	3
				3	3	4	0	1	2	5	0	1	2	3	4
最も新しい					4	0	1	2	5	0	1	2	3	4	5

図 6.10　LRU の置換え例

スラッシング(thrashing)

- ページフォールトが頻繁に発生し，処理が進まないこと.
  - ワーキングセット（プログラムが一定期間内にアクセスするアドレスの集合）を小さくするか，
  - 物理メモリをワーキングセット以上に大きくする.

問題

- Beladyの異常が発生する理由を説明しなさい

- FIFO, OPT, LRUアルゴリズムでの置き換え結果を物理ページ数が，3，4の場合につい

参照列1	0	3	2	1	0	3	4	0	3	2	1	4

参照列2	0	1	2	3	4	2	3	0	3	2	4	1

図 6.12

問題の答え

- FIFOアルゴリズムでは，物理ページサイズと等しい周期的なアクセスでページアウトが発生しやすい．しかし，位相がずれると，ページアウトが全く発生しない，または多発したりする．

参照列	0	3	2	1	0	3	4	0	3	2	1	4
ページフォールト	×	×	×	×	×	×	×			×	×	×
物理ページ数=3　最も古い	0	0	0	3	2	1	0	3	4	0	3	2
		3	3	2	1	0	3	4	0	3	2	1
最も新しい			2	1	0	3	4	0	3	2	1	4

参照列	0	3	2	1	0	3	4	0	3	2	1	4
ページフォールト	×	×	×	×			×			×	×	×
物理ページ数=4　最も古い	0	0	0	0	3	2	1	1	1	4	0	3
		3	3	3	2	1	0	3	4	0	3	2
			2	2	1	0	3	4	0	3	2	1
最も新しい				1	0	3	4	0	3	2	1	4

解図 6.1　LRU を利用した置換え

ファイルシステム

ファイル

- OSを終了してもデータが保持される永続的記録
  - 不揮発性の所以は装置の特性.
- データを格納する論理的な入れ物.
  - 実際は不連続なセクタ毎にデータが記録される.
- ファイルの名前
  - 同じ名前のファイルを同じ場所に置くことはできない
- アクセスの仕方
  - シーケンシャルアクセス（順アクセス）
  - ランダムアクセス（直接アクセス）

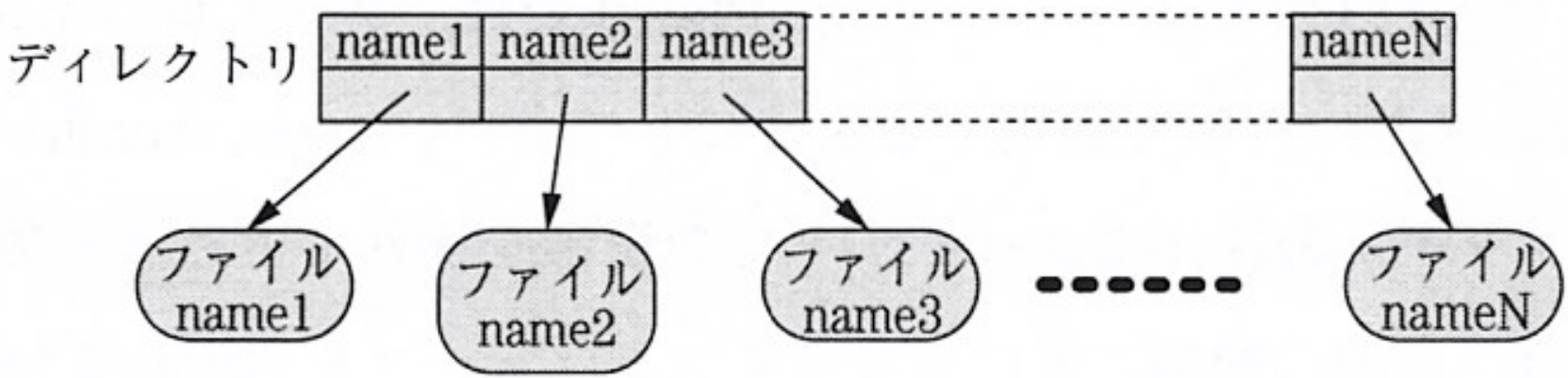
ファイルシステムの機能

- 利用者の為にファイルの名前空間を提供すること
  - 単層ディレクトリ構造：名前空間が狭い
  - 階層ディレクトリ構造：ディレクトリごとに名前空間が用意されている.

  名前をキーとしたファイルの検索にはハッシュが用いられる.
- ファイルと記憶装置の関係付けを管理すること
  - 属性管理：rwx等のパーミッション，所有者，時間，サイズなど
  - 内容管理：セクタ単位の情報を繋ぐ方法，
  - アクセス機能の提供

単層ディレクトリ

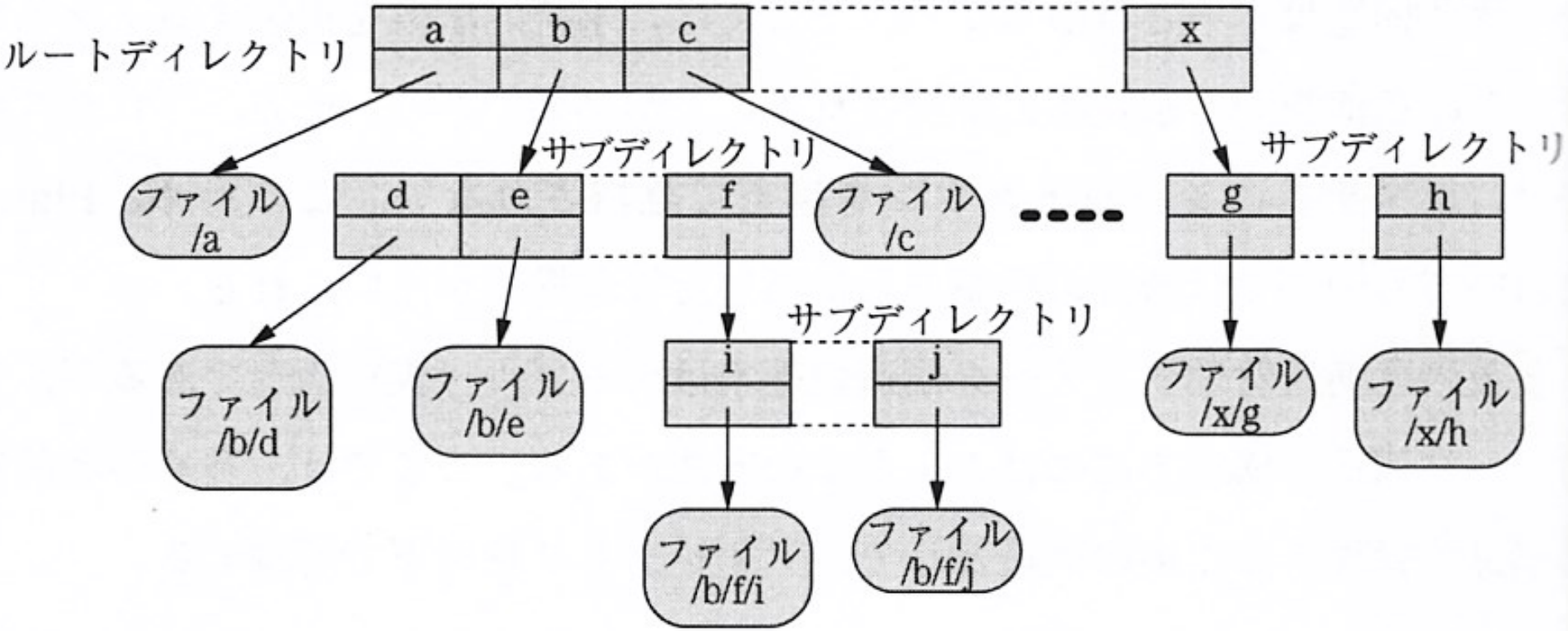
- 名前空間が狭く，人手でファイルに一意の名前をつけにくくなる.
- ファイル数が増え，人手で目的のファイルを探すのが困難になったりする.
- CP/Mなどの古いOSにしか見られない.



（a） 単層ディレクトリ

階層ディレクトリ

- 名前空間が広がる.
- ファイル名の参照法は,「絶対パス名」「相対パス名」の2種類が利用できる.



（b） 階層ディレクトリ

ハッシュを用いたファイル名の管理

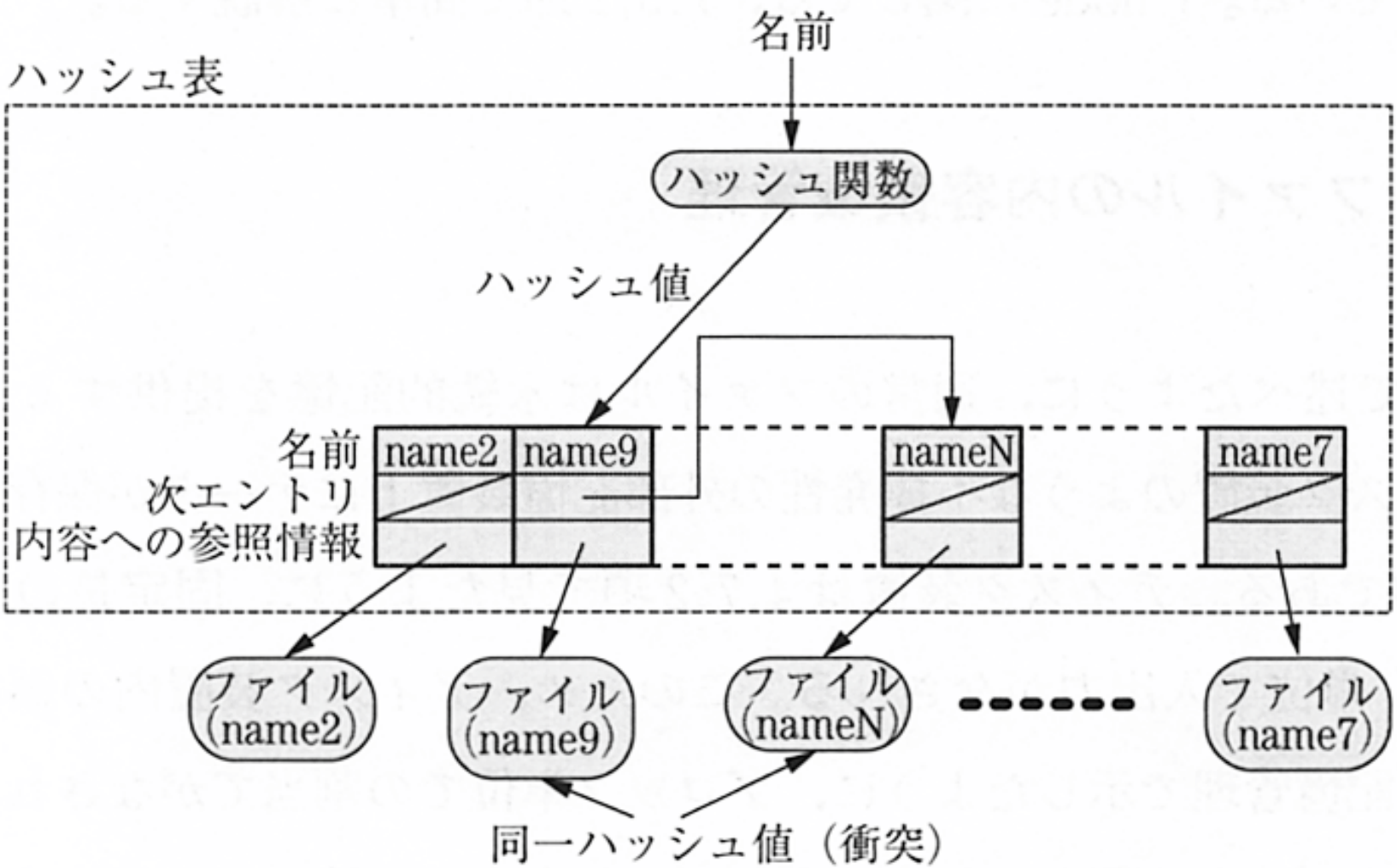


図 7.3　ハッシュ表を利用した名前管理

ファイルの属性：メタデータ

サイズ，種類，アクセス権(rwx)，所有者，グループ，ファイル作成年月日，時刻，修正年月日，時刻，アクセス年月日，時刻，i-node番号，リンク数，

```
% ls -l
-rw-r--r--   1 twada  staff     923  5  5  2011 x
-rw-r--r--   1 twada  staff     625  5  6  2011 y
-rw-r--r--   1 twada  staff     490  5  6  2011 z.bz2
```

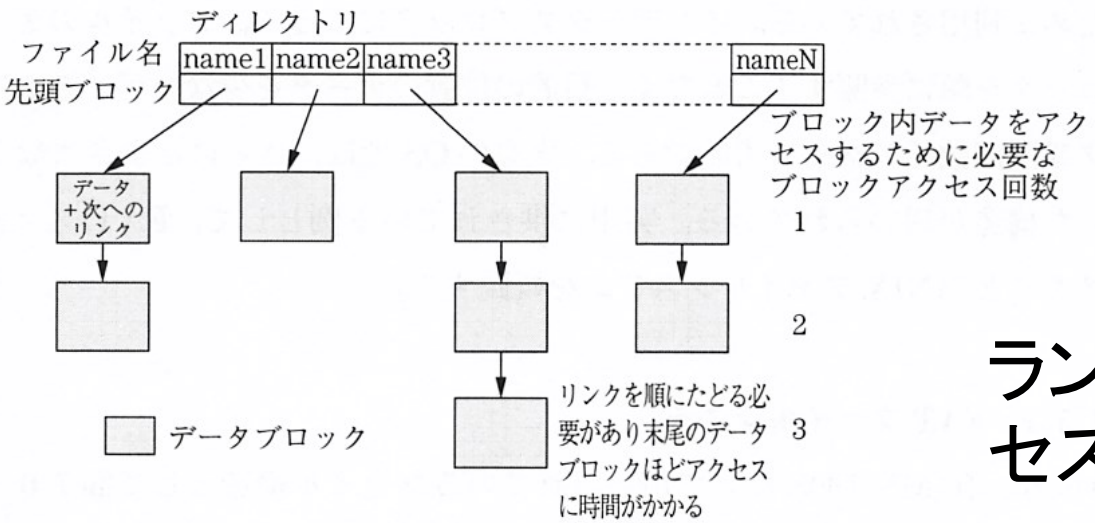
所有者 rw
グループ r
他人 r
リンク数
所有者
グループ
サイズ
月
日
年
ファイル名

```
%ls –l /bin/ls
-r-xr-xr-x  1 root  wheel  80688 12  8  2011 /bin/ls

%ls -ld /var/log
drwxr-xr-x  46 root  wheel  1564 10 21 06:33 /var/log

%ls -l /dev/tty
crw-rw-rw-  1 root  wheel    2,   0 10 19 17:54 /dev/tty
```

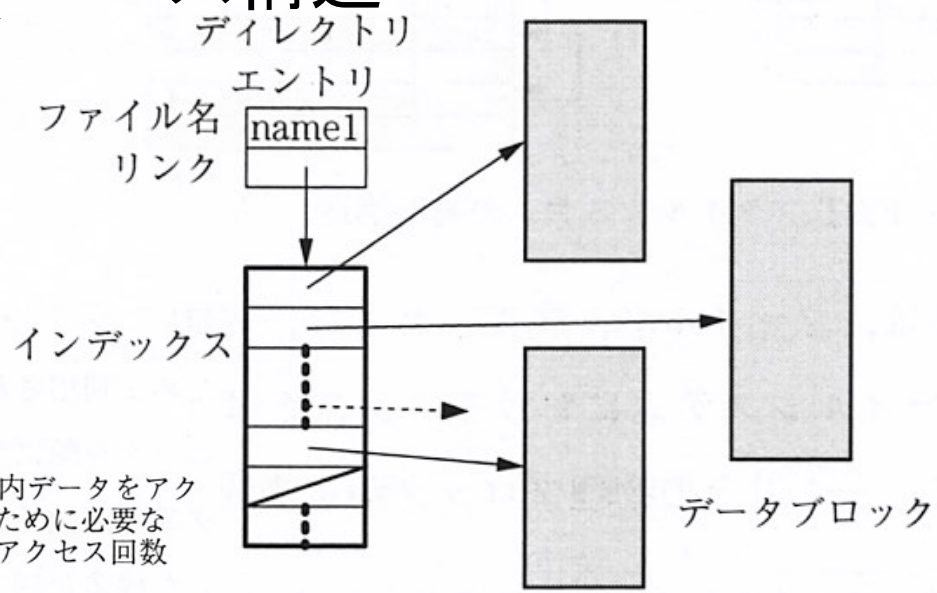
ファイルはセクタの集まり：つなげる必要がある.

- データブロックのリンク構造

シーケンシャルアクセスのみ



- データブロックのインデックス構造



ランダムアクセス（直接アクセス）可能

File Allocation Table: FAT(リンク構造)

クラスタサイズ16KBで，FAT16なら，最大ファイルサイズは2^16*16*1024=1GB

このビット数に応じて，FAT12, FAT16, FAT32がある.

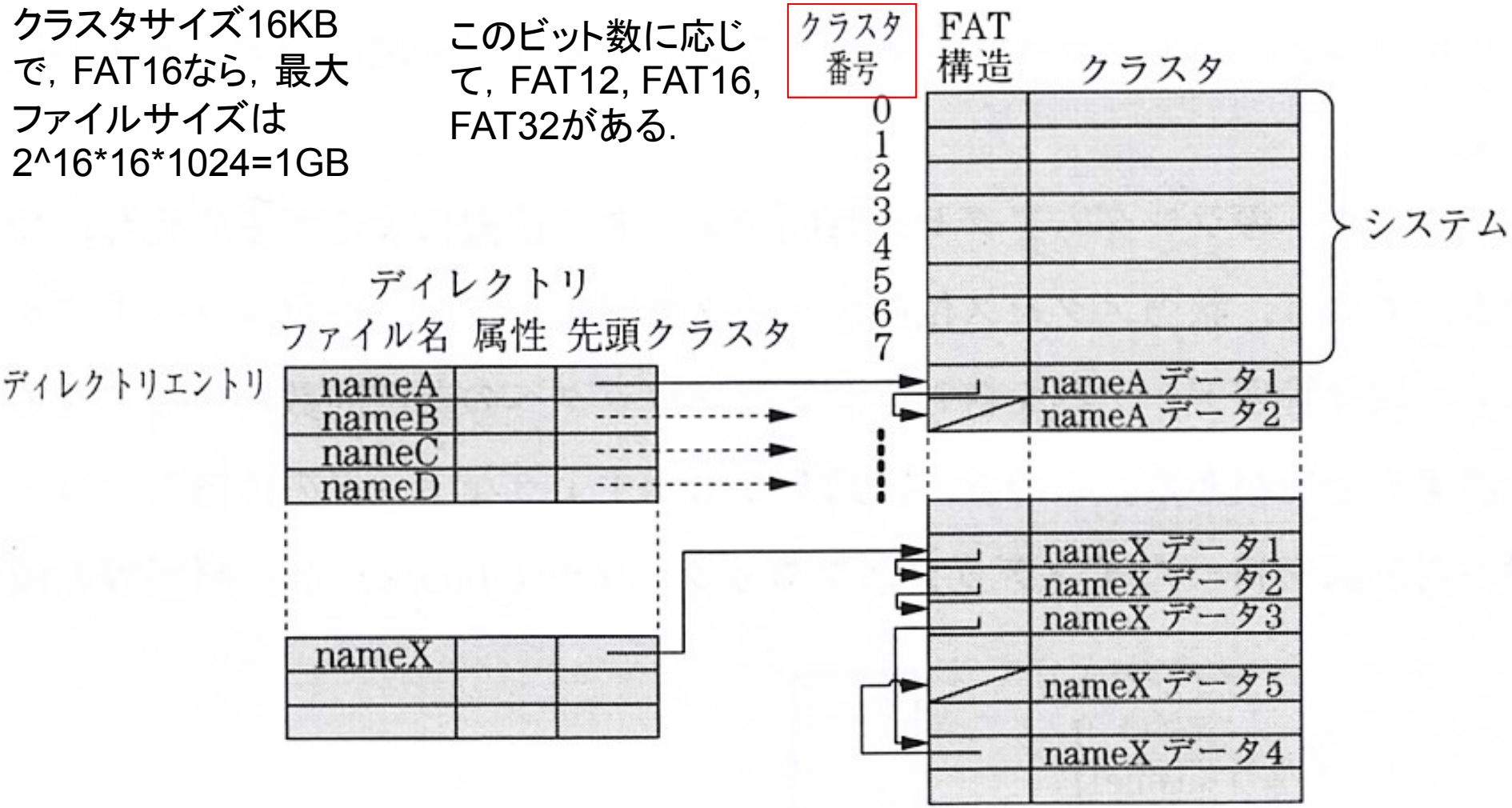


図 7.6 FAT ファイルシステムの論理構造

UNIXのファイルシステム

- インデックス＋リンク構造を採用
- i-node構造体の末尾3つが
  - 間接ブロック
  - ２重間接ブロック
  - 3重間接ブロック

  になっている.
- ランダムアクセス（直接アクセス）可能

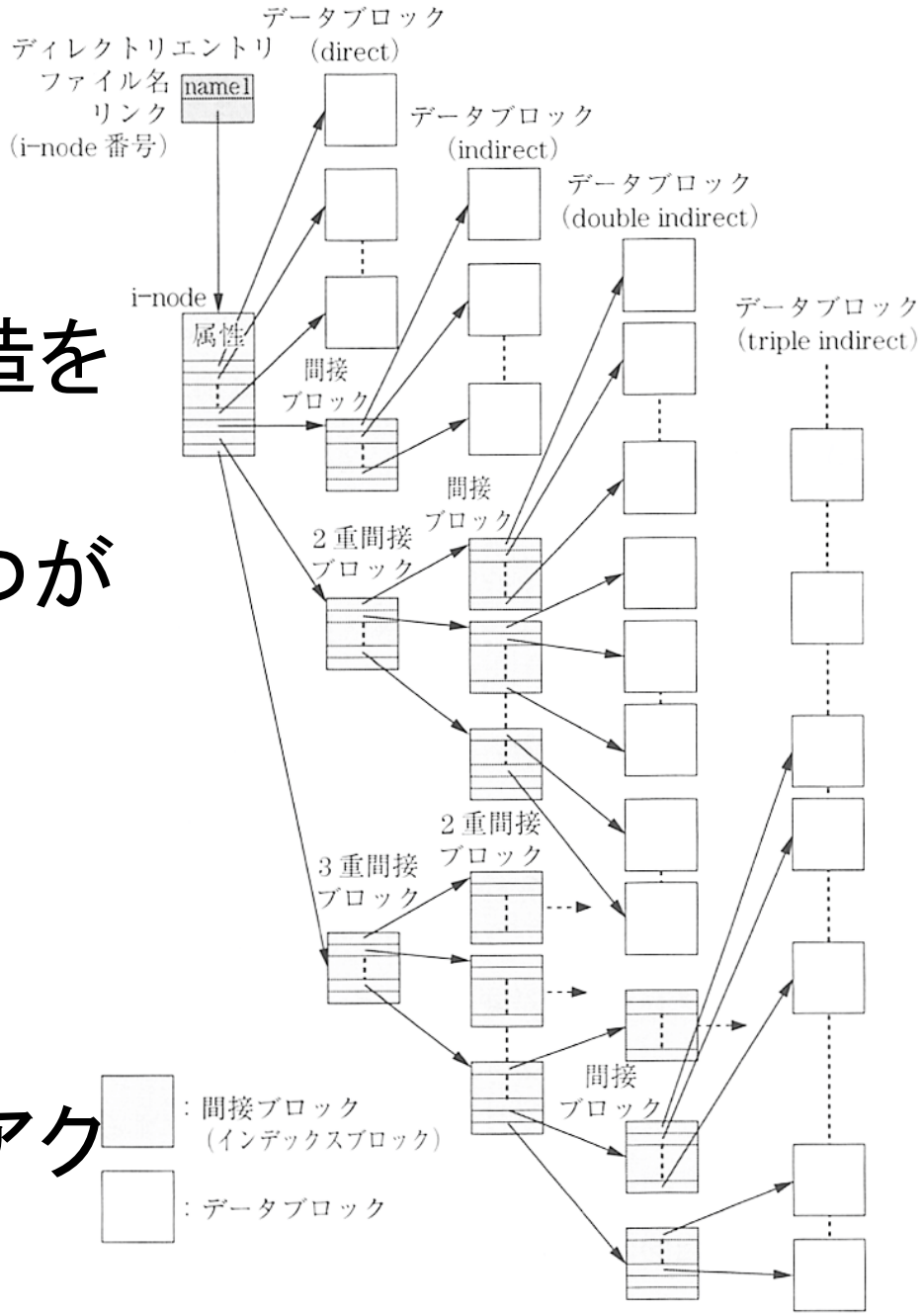


図 7.7　i-node と間接ブロックを利用したファイルの論理構造

ログ構造ファイルシステム

- ファイルへの書き出しを，その変更部分の追記という形で実現するものが，ログ構造ファイルシステムである.
- CD-RWやDVD-RW，フラッシュメモリなど書き換え回数に制限のあるデバイスでしばしば用いられる.

ジャーナリング・ファイルシステム

- データ操作と，その操作による変更を，「ログ領域」（ジャーナル：日誌）に記録する.
- ファイルシステム操作中に障害が発生しても，回復できるようにする.
- XFS, JFS, RaiserFS, など

RAID: Redundant Array of Independent Disks

- 複数のHDDを束ねて使う方式
- RAID0: ストライピング （耐故障性なし，高速）
- RAID1：ミラーリング（二重化）
- RAID2：ビット単位での誤り訂正専用ドライブ
- RAID3：ビット／バイト単位のパリティドライブ
- RAID4：ブロック単位での専用パリティドライブ
- RAID5：ブロック単位でのパリティ分散記録

問題

- FAT32で，クラスタサイズ8KBの場合のファイルの最大サイズはいくらか？