

ファイルの 記憶空間管理 追加削除ができるために



今回は
ファイルの記憶空間管理
を考えます
要するにディスク上の
スペースの管理です



ファイル内容領域（スペース）管理

- 問題 1)

ディスクはブロック (セクタ) 単位でアクセスする
ファイルをブロックに分けて記憶させ、
管理しなければならない

2



ファイル内容領域（スペース）管理

- 問題 1)

ディスクはブロック (セクタ) 単位でアクセスする
ファイルをブロックに分けて記憶させ、
管理しなければならない

- 問題 2)

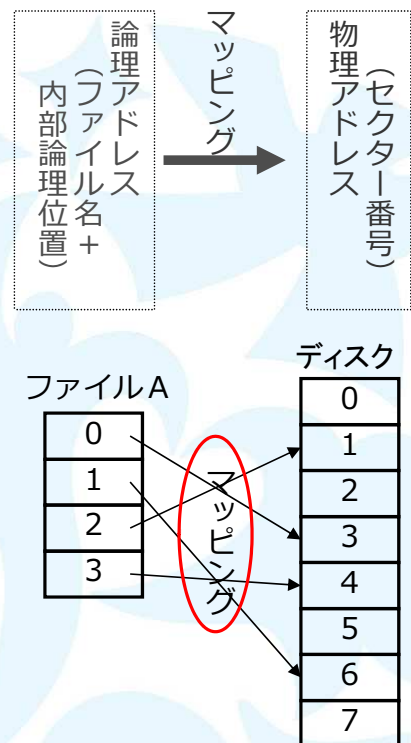
ファイルを編集すると長さが伸びることがある
編集後ファイルへ書き戻すときに、ファイルを
伸ばすのか？ 後ろが詰まっているときは？
逆に短くなると、小さい空き地ができる。使え
ない空き地が増える (フラグメンテーション)

3



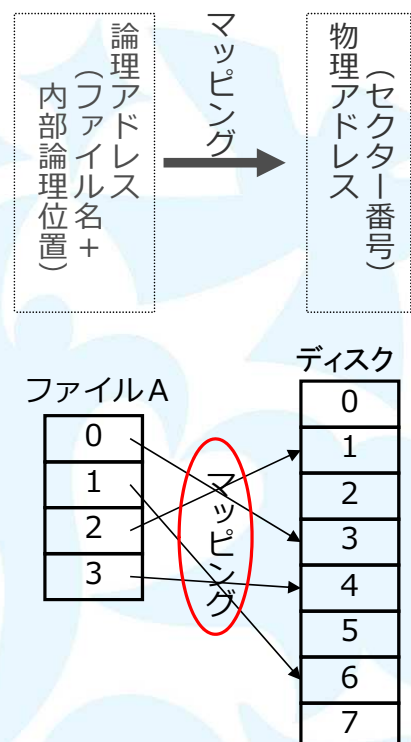
領域管理の考え方

- ブロックの「論理アドレス」と物理アドレス間をマッピング
- ブロックの順番が変わっても（途中追加など） OK
- 縮んでも隙間が（断片化）出来ない



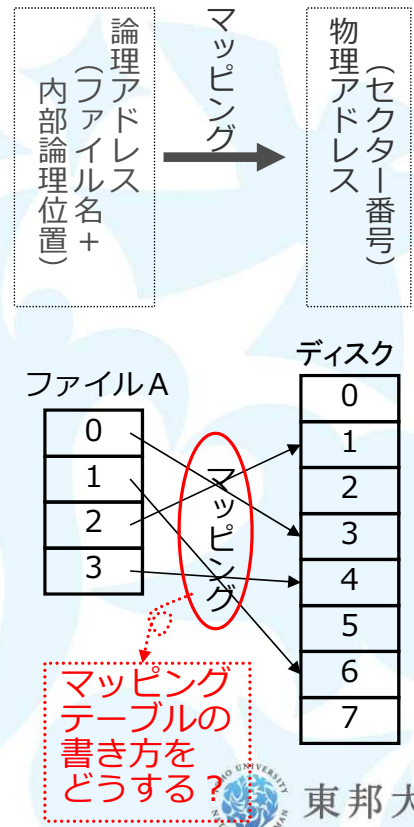
領域管理の考え方

- ブロックの「論理アドレス」と物理アドレス間をマッピング
- ブロックの順番が変わっても（途中追加など） OK
- 縮んでも隙間が（断片化）出来ない
- 仮想記憶と似たような発想
 - アドレスをブロック単位でマップ ⇒ 上記の問題解決



領域管理の考えかた

- ブロックの「論理アドレス」と物理アドレス間をマッピング
- ブロックの順番が変わっても（途中追加など） OK
- 縮んでも隙間が（断片化）出来ない
- 仮想記憶と似たような発想
 - アドレスをブロック単位でマップ
⇒上記の問題解決



6

リンクでスペース管理

(スペース管理)

(名前管理・ディレクトリ)

ファイル名 test1.txt	先頭ブロックへのポインタ 23
--------------------	-----------------

ブロック23
次は44

ブロック44
次は71

ブロック71
終り

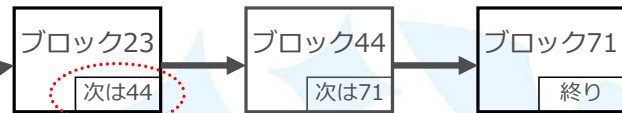
7

リンクでスペース管理

(スペース管理)

(名前管理・ディレクトリ)

ファイル名 test1.txt	先頭ブロックへのポインタ 23
--------------------	-----------------



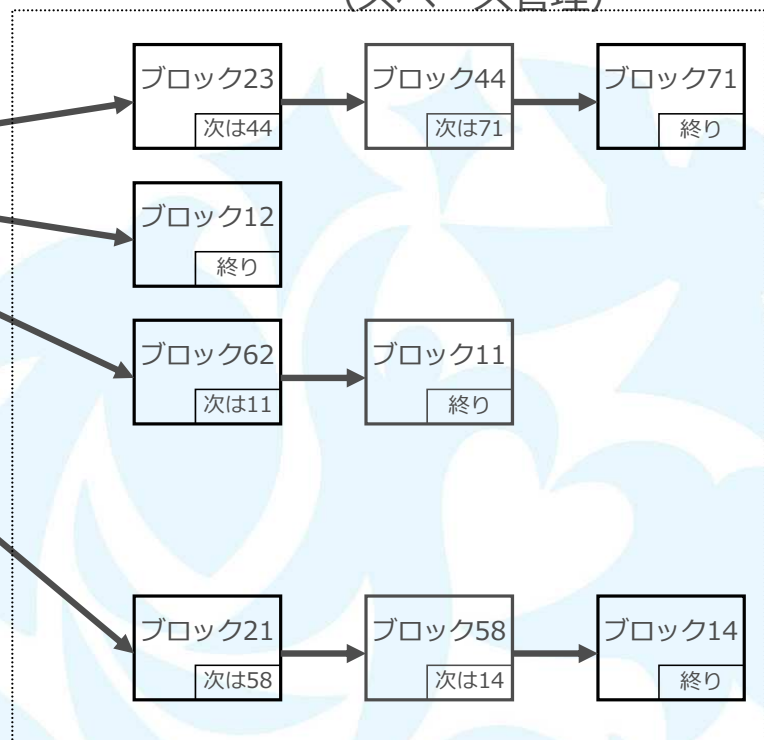
次を指すリンク

リンクでスペース管理

(スペース管理)

(名前管理・ディレクトリ)

ファイル名 test1.txt	先頭ブロックへのポインタ 23
ファイル名 kadai.doc	先頭ブロックへのポインタ 12
ファイル名 word.exe	先頭ブロックへのポインタ 62
ファイル名 work.java	先頭ブロックへのポインタ 21



リンクでスペース管理

(スペース管理)

(名前管理・ディレクトリ)

ファイル名 test1.txt	先頭ブロックへのポインタ 23
ファイル名 kadai.doc	先頭ブロックへのポインタ 12
ファイル名 word.exe	先頭ブロックへのポインタ 62
ファイル名 work.java	先頭ブロックへのポインタ 21



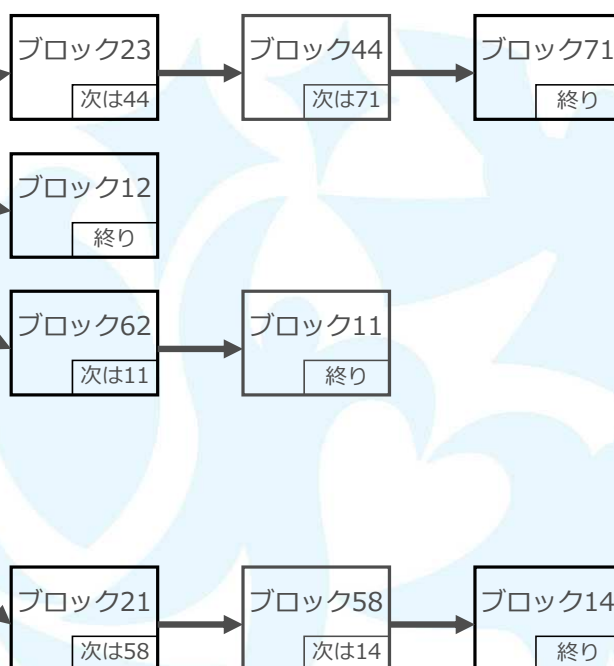
- 単純なアイデア

リンクでスペース管理

(スペース管理)

(名前管理・ディレクトリ)

ファイル名 test1.txt	先頭ブロックへのポインタ 23
ファイル名 kadai.doc	先頭ブロックへのポインタ 12
ファイル名 word.exe	先頭ブロックへのポインタ 62
ファイル名 work.java	先頭ブロックへのポインタ 21



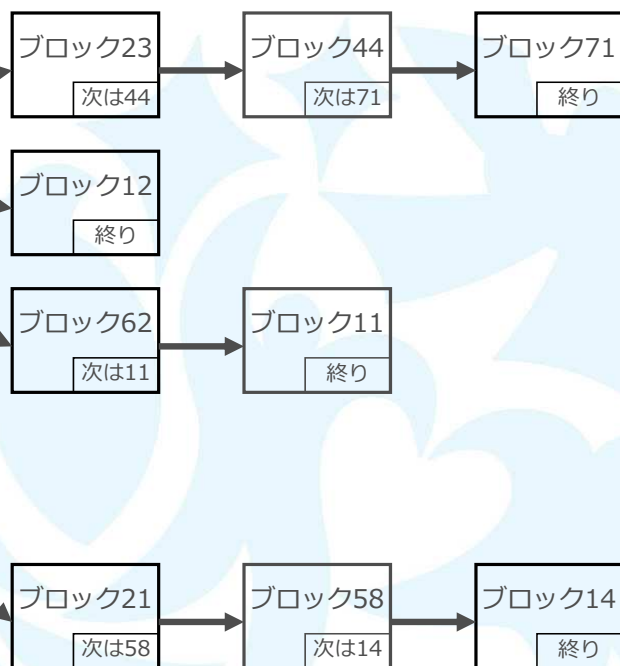
- 単純なアイデア
- 「次」はそのブロックを読まなければわからない

リンクでスペース管理

(スペース管理)

(名前管理・ディレクトリ)

ファイル名 test1.txt	先頭ブロックへのポインタ 23
ファイル名 kadai.doc	先頭ブロックへのポインタ 12
ファイル名 word.exe	先頭ブロックへのポインタ 62
ファイル名 work.java	先頭ブロックへのポインタ 21



チエーン型は途中のブロックを全部読まないと
後ろのブロックに行き着かない

- 単純なアイデア
- 「次」はそのブロックを読まなければわからない



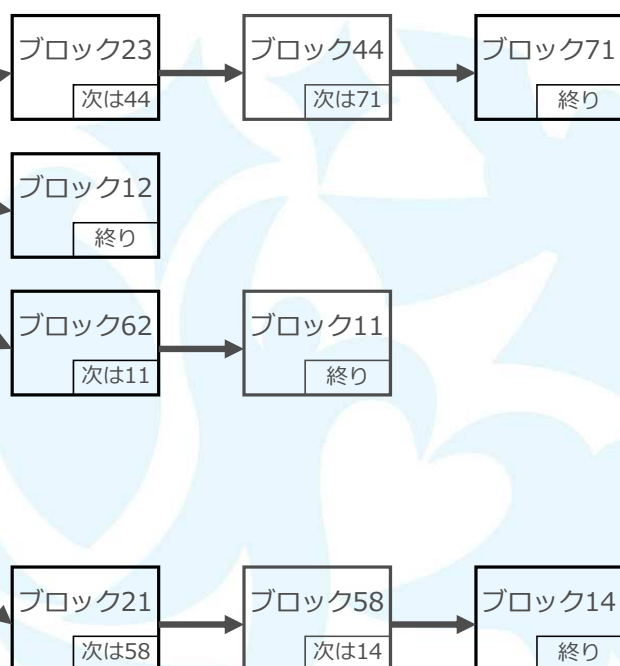
東邦大学

リンクでスペース管理

(スペース管理)

(名前管理・ディレクトリ)

ファイル名 test1.txt	先頭ブロックへのポインタ 23
ファイル名 kadai.doc	先頭ブロックへのポインタ 12
ファイル名 word.exe	先頭ブロックへのポインタ 62
ファイル名 work.java	先頭ブロックへのポインタ 21



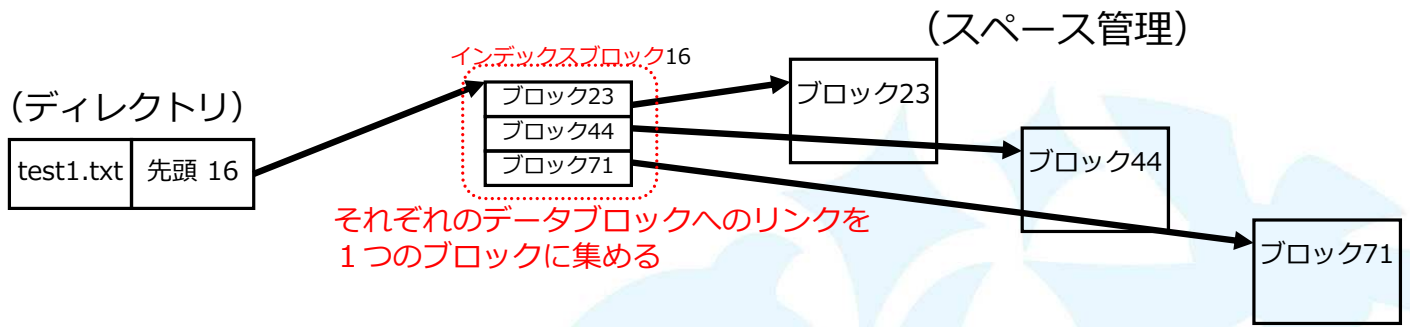
- 単純なアイデア
- 「次」はそのブロックを読まなければわからない

● 「空き」の管理もしなければならぬ

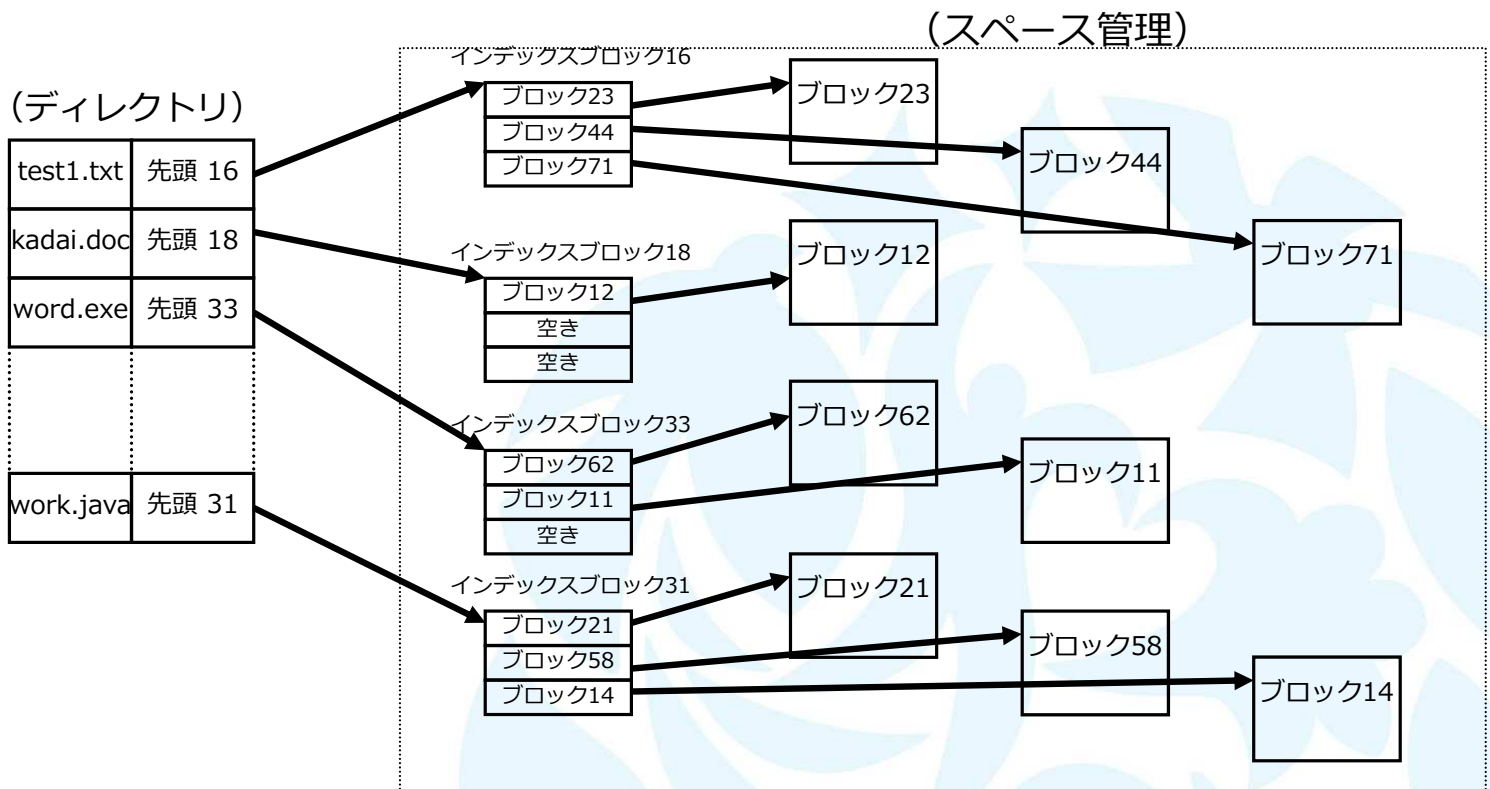


東邦大学

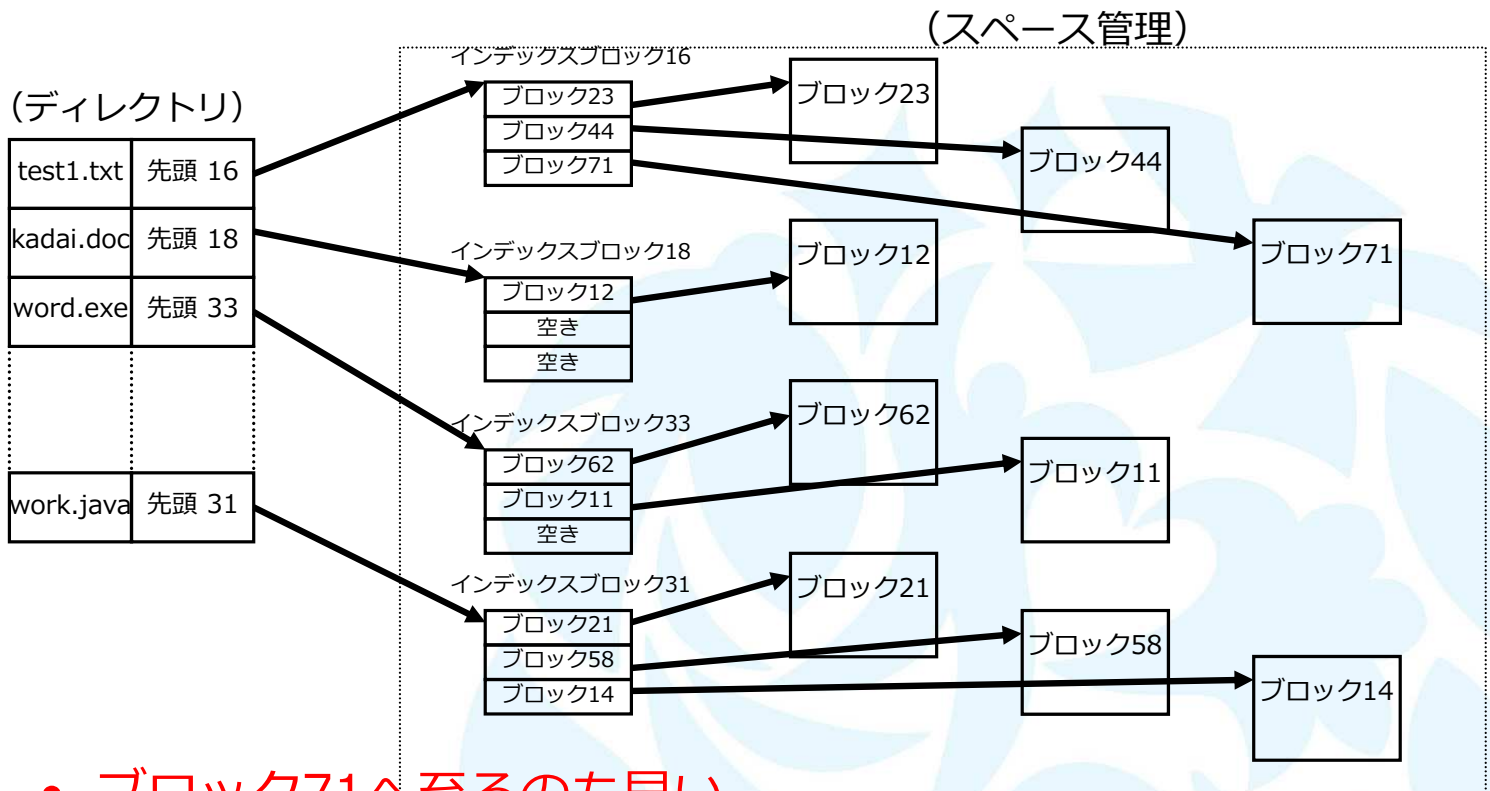
スペース管理用の情報を集める



スペース管理用の情報を集める

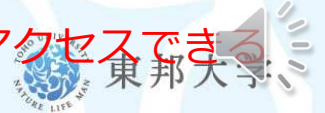


スペース管理用の情報を集める

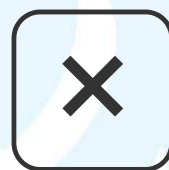


- **ブロック71へ至るのも早い**
インデックスブロック16を1つ読めば23も44も71もアクセスできる

16



リンクリストや
インデックスノードによる
記憶空間管理の方法が
理解できましたか？



次へ

17

