

# 情報理論及演習

## 2003年5月19日

担当：池口 徹

埼玉大学 大学院 理工学研究科 情報数理科学専攻 助教授

Email : tohru@ics.saitama-u.ac.jp

URL : <http://www.nls.ics.saitama-u.ac.jp/~tohru>

# 今日の講義の内容は？

## データ保存のお話

- ディスクへのアクセス
- ディスクドライブの仕組み
- ディスクアクセスの高速化，圧縮

# Chapter10

## How Disk Storage Works

- 磁気ディスク

- データ保存に最も広く用いられている
- 容量：数百 kB ~ 数 GB
- ディスク微少な領域が 0, 1 を表現

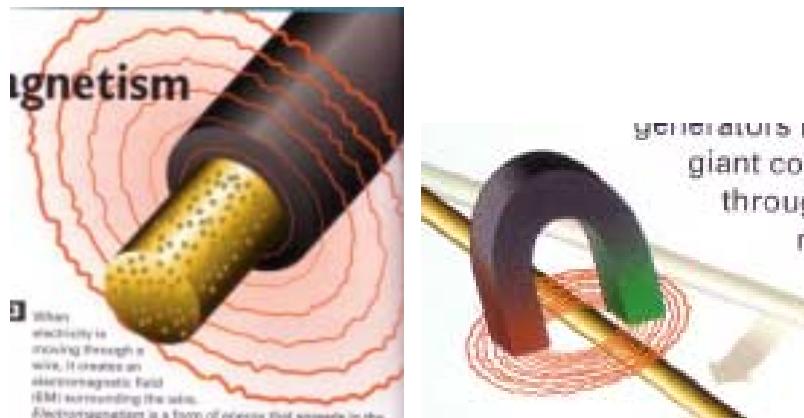
- どのようにディスク上にデータを書き込むか？

DOS → Windows

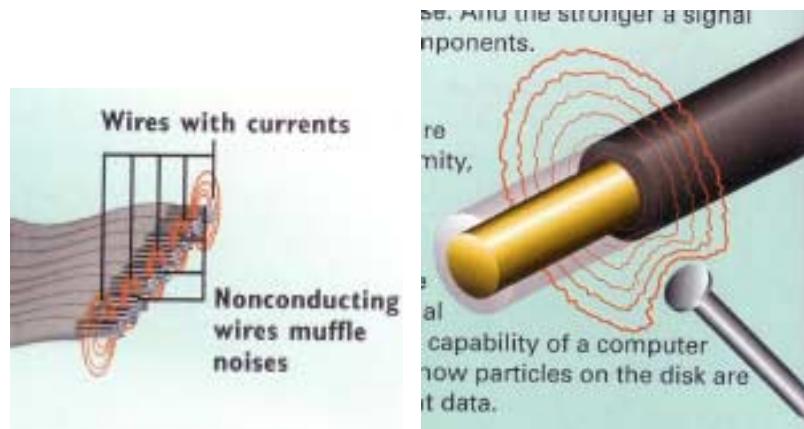
- フォーマット
  - セクタ
  - トランク
- 容量はセクタ数, トランク数による

# 電磁氣

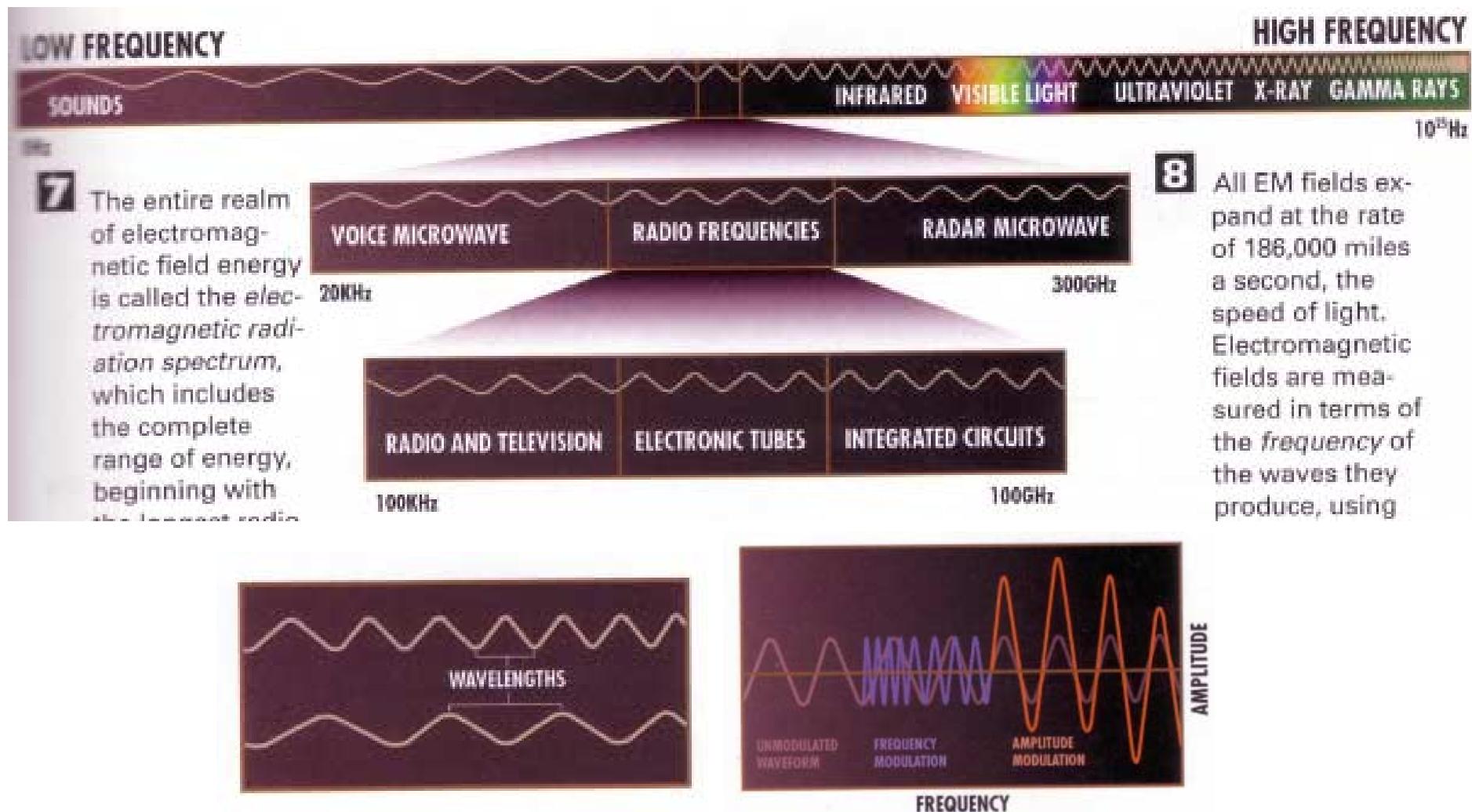
## ● 電流变化 $\leftrightarrow$ 磁界



## ● 磁界への影響 → 磁界変化

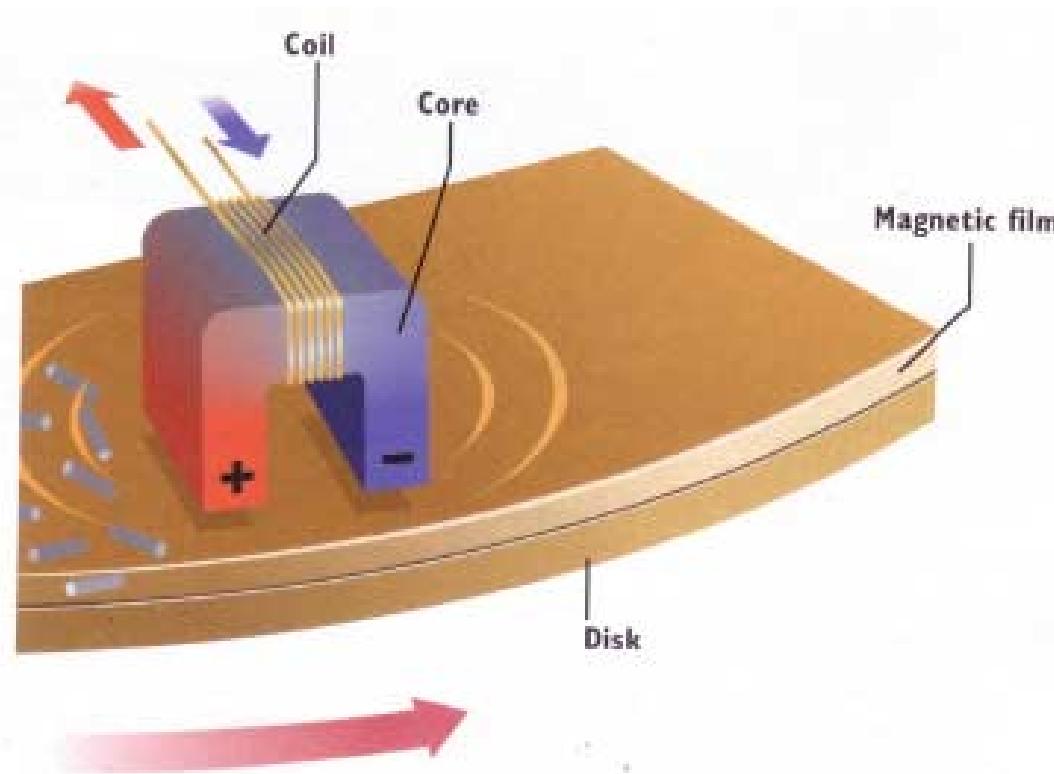


# 周波数帯



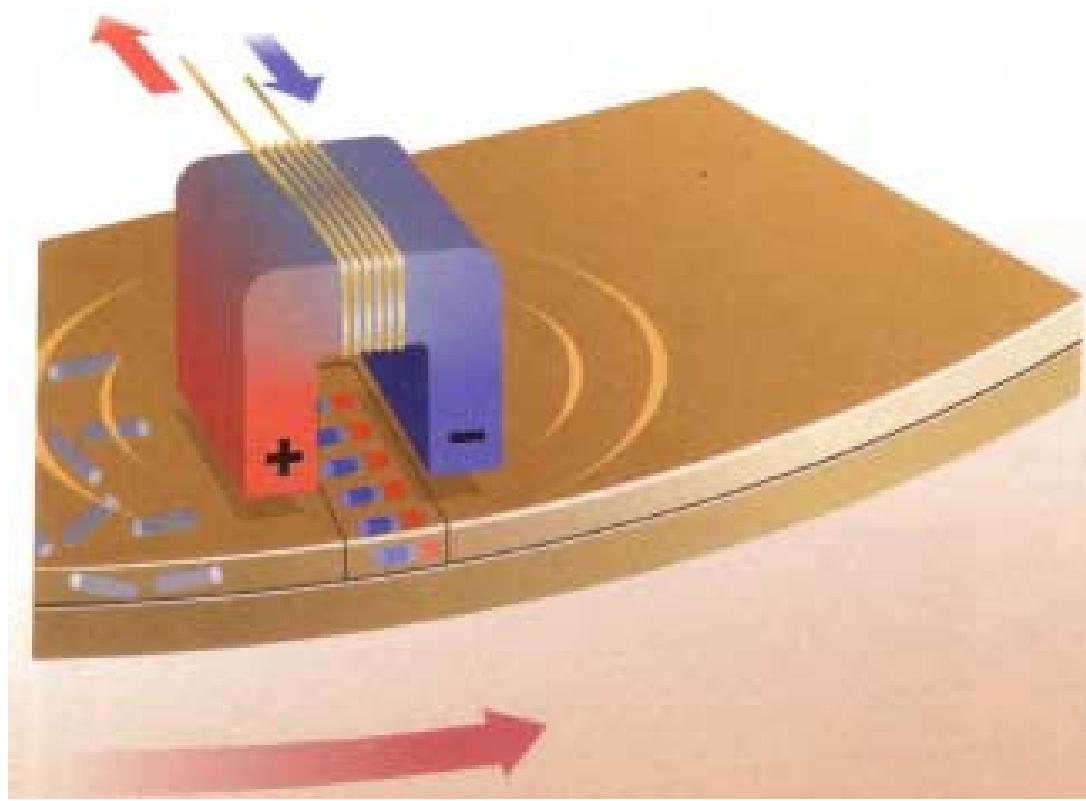
# ディスクへの読み書き

- 磁性粒子がランダムにばらまかれている
- 電流 → 磁界



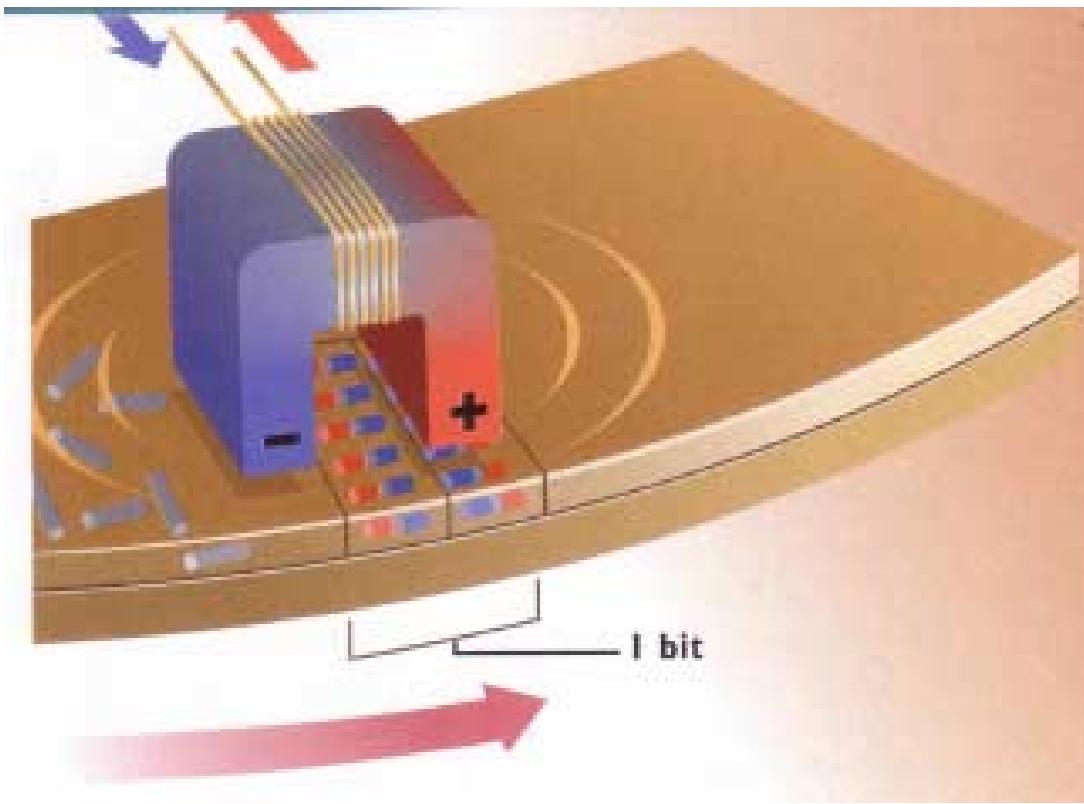
# ディスクへの読み書き

- 磁界 → 粒子の向き



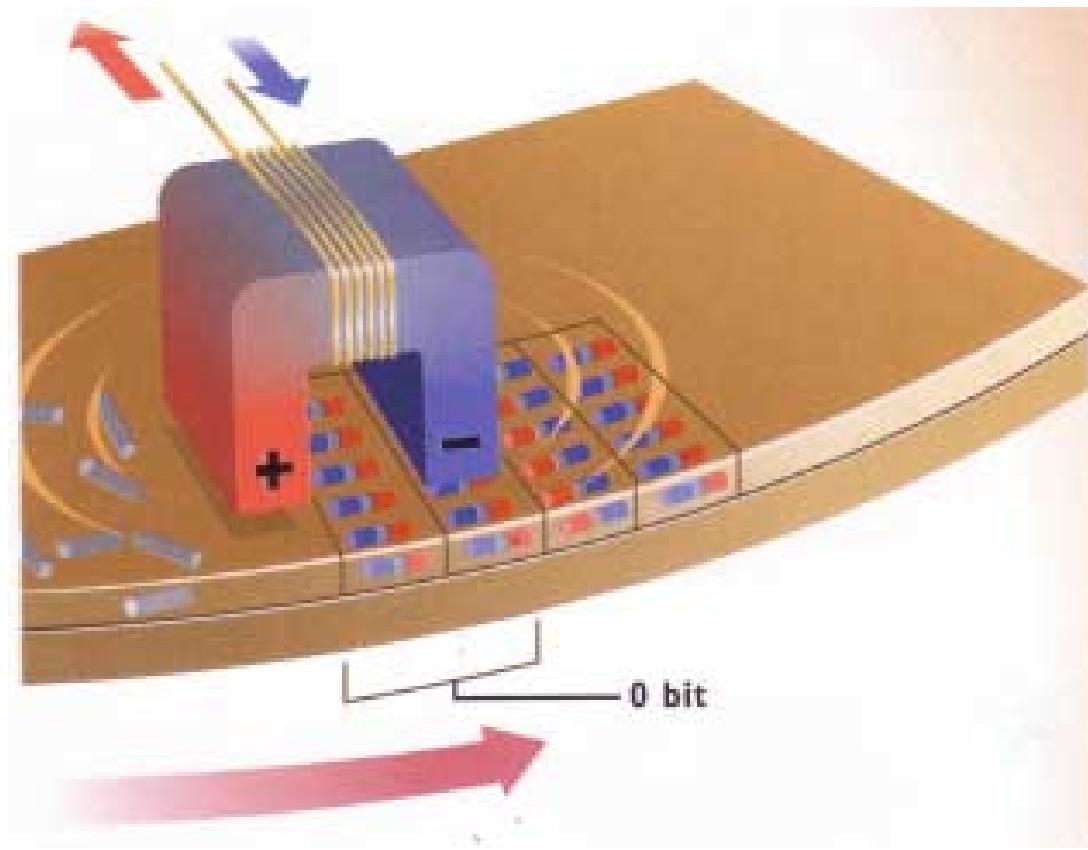
# ディスクへの読み書き

- 2バンド = 1ビット
- 2バンドの粒子のならびの順逆で0, 1が決まる



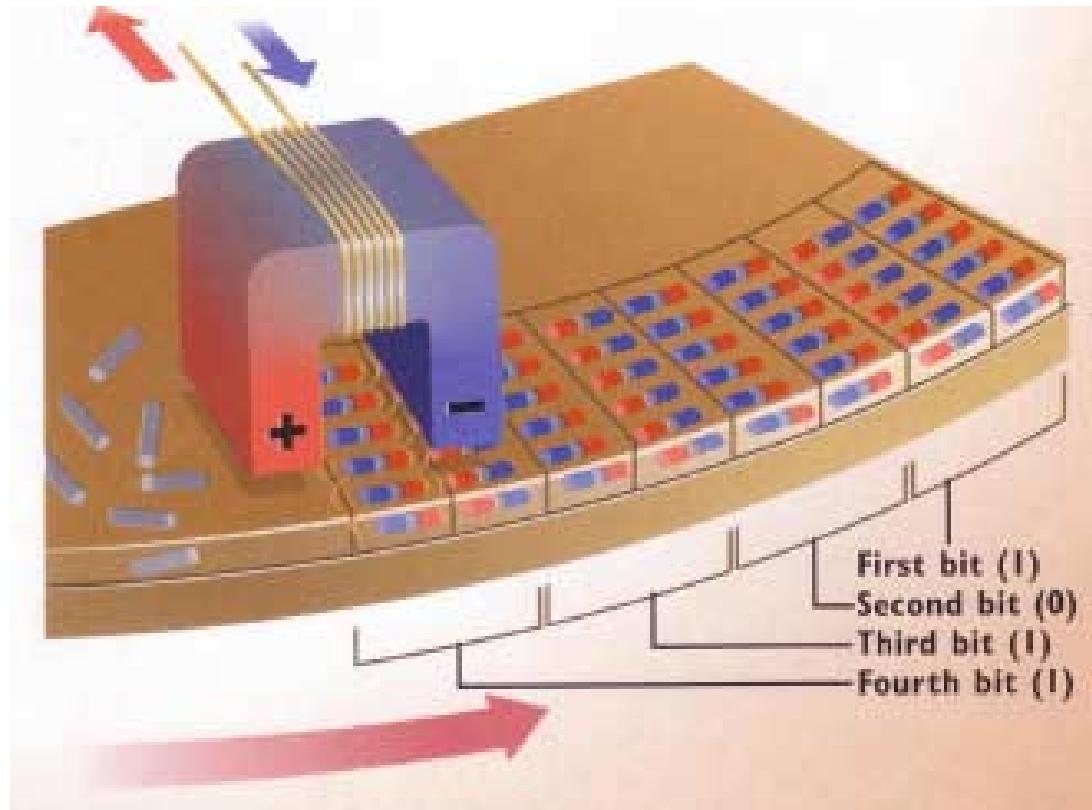
# ディスクへの読み書き

- 1 バンド分 = (遅くとも) 数分の 1 秒程度



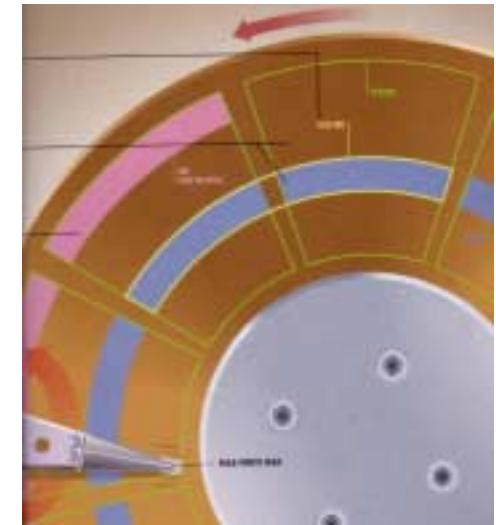
# ディスクへの読み書き

- 磁化粒子 = 磁石 → 電流



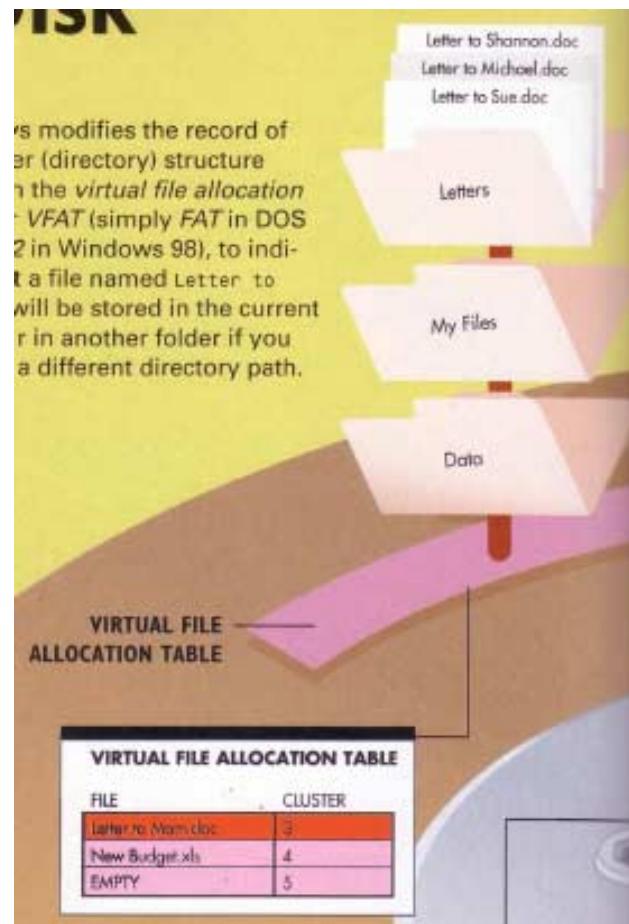
# ディスクのフォーマット

- フォーマット → ディクスを使用するために最初に必要
  - 半径方向 → セクタ, 周方向 → トラック
- 2つ以上のセクタ = クラスタ (最小の記録単位)
- File Allocation Table (at the 0-th sector) ← DOS, 16bit
- Virtual FAT
  - Win, 32bit, long file name
- FAT32 ≈ Win 98 → NTFS
- 保持される情報
  - ディクスのディレクトリ構造
  - どのクラスタにどのファイル?



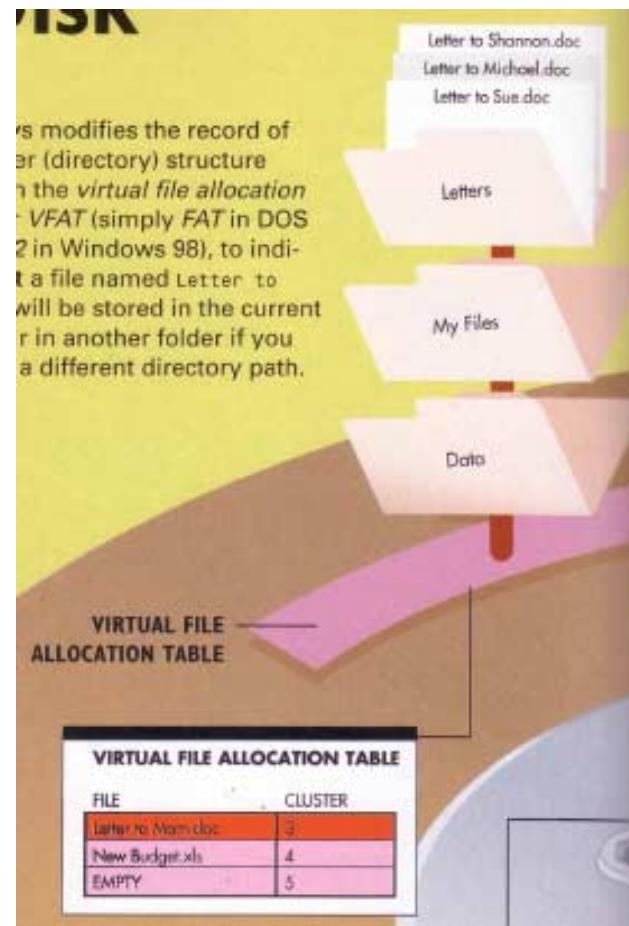
# ディスクに書く

- (V)FAT(32) 内の情報 (だけ) を変更する



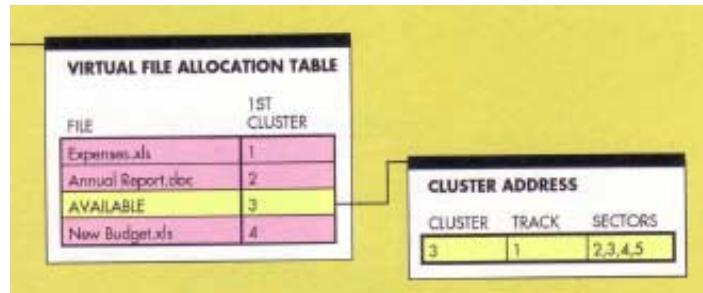
# ディスクに書く

- (V)FAT(32) 内の情報(だけ)を変更する
    - どのクラスタ(トラック, セクタ)に書き込むか



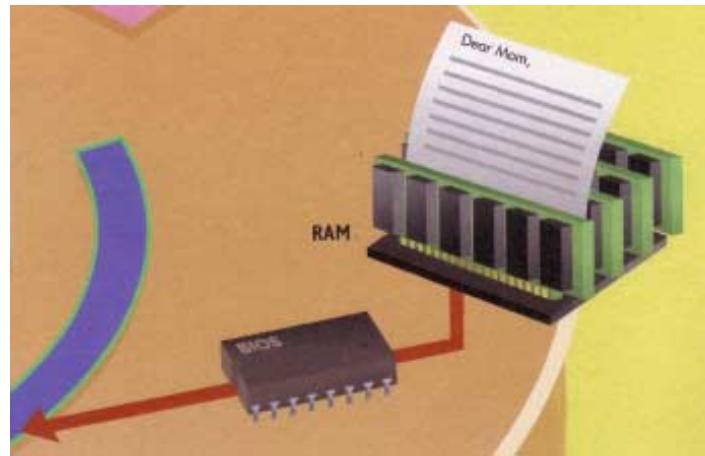
# ディスクに書く

- (V)FAT(32) 内の情報 (だけ) を変更する
  - どのクラスタ (トラック, セクタ) に書き込むか



# ディスクに書く

- (V)FAT(32) 内の情報 (だけ) を変更する
  - どのクラスタ (トラック, セクタ) に書き込めるか
  - BIOS へ情報を送る



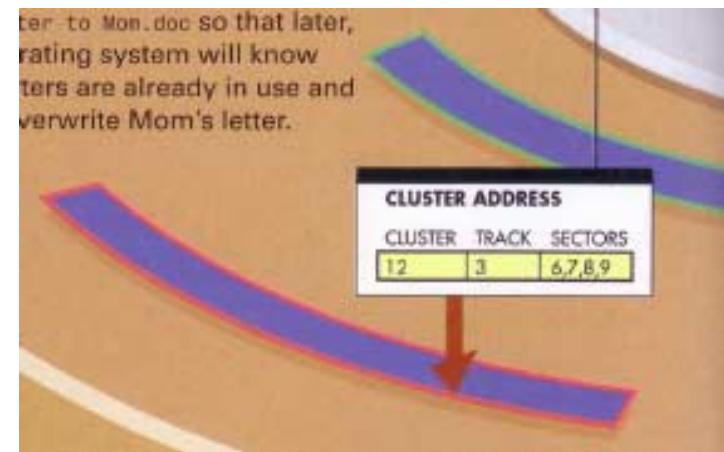
# ディスクに書く

- (V)FAT(32) 内の情報 (だけ) を変更する
  - どのクラスタ (トラック, セクタ) に書き込むか
  - BIOS へ情報を送る
  - 該当するトラック, セクタに情報を書き込む

# ディスクに書く

- (V)FAT(32) 内の情報 (だけ) を変更する
  - どのクラスタ (トラック, セクタ) に書き込めるか
  - BIOS へ情報を送る
  - 該当するトラック, セクタに情報を書き込む
  - クラスタ 1 つで足りない場合は, 別のクラスタ (隣でなくても良い) に書き込む

FILE	1ST CLUSTER
AVAILABLE	12
AVAILABLE	13
Memo to Boss.doc	14



# ディスクに書く

- (V)FAT(32) 内の情報 (だけ) を変更する
  - どのクラスタ (トラック, セクタ) に書き込めるか
  - BIOS へ情報を送る
  - 該当するトラック, セクタに情報を書き込む
  - クラスタ 1 つで足りない場合は, 別のクラスタ (隣でなくても良い) に書き込む
  - 実際に書き込んだという情報に FAT を変更

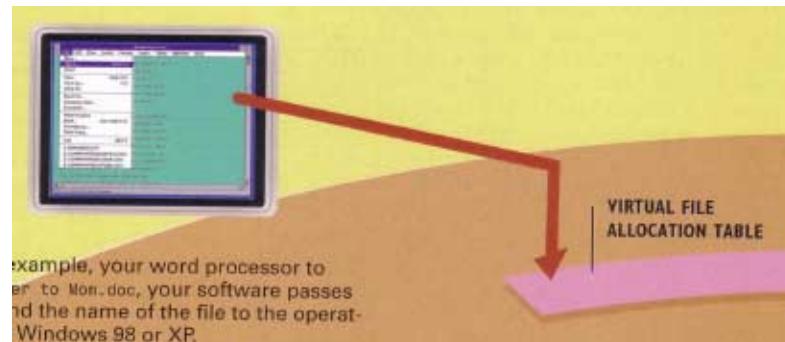
# ディスクに書く

- (V)FAT(32) 内の情報 (だ)
  - どのクラスタ (トラック) に書くか
  - BIOS へ情報を送る
  - 該当するトラック, クラスタを何に書くか
  - クラスタ 1 つで足りない場合、隣のクラスタ (隣でなくとも良い) に書き込む
  - 実際に書き込んだという情報に FAT を変更
- データを消去しても, 実際に変更されるのは FAT のみ  
→ データそのものは残っている

VIRTUAL FILE ALLOCATION TABLE	
FILE	Available
CLUSTER	
Letter to Mom.doc	3
New Budget.xls	4
EMPTY	5
Old Budget.xls	6

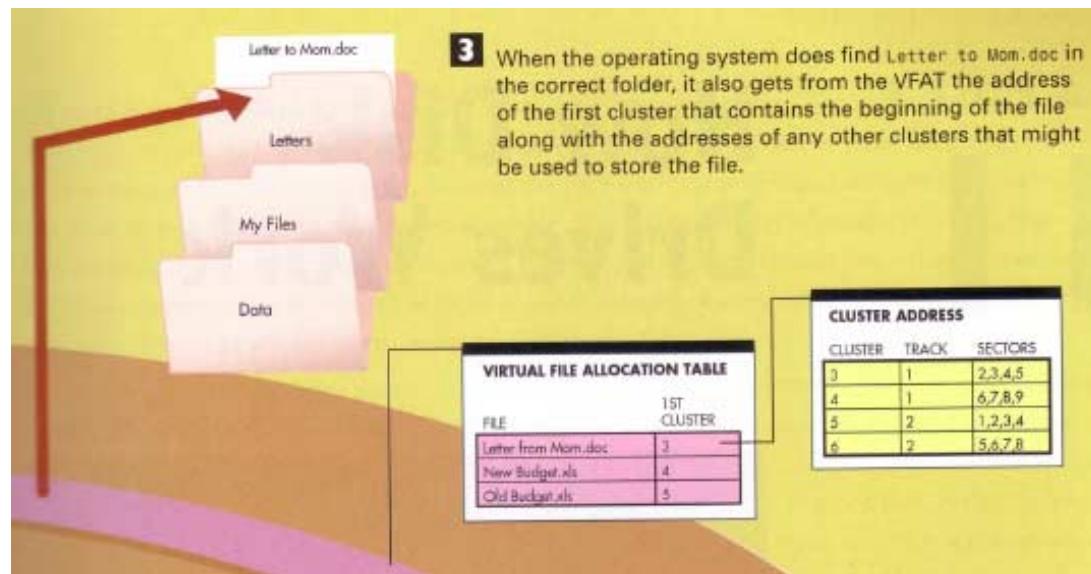
# ディスクから読む

- アプリケーションからの読み込み要求



# ディスクから読む

- アプリケーションからの読み込み要求
- VFAT を検索



# ディスクから読む

- アプリケーションからの読み込み要求
- VFAT を検索
- VFAT からクラスタ情報を抽出 → OS



# ディスクから読む

- アプリケーションからの読み込み要求
- VFAT を検索
- VFAT からクラスタ情報を抽出 → OS
- OS→BIOS → DISK



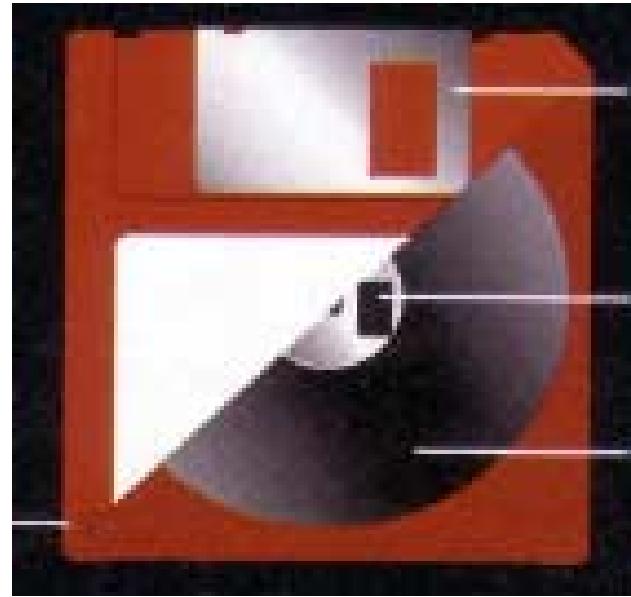
# Chapter 11

## How Disk Drives Work

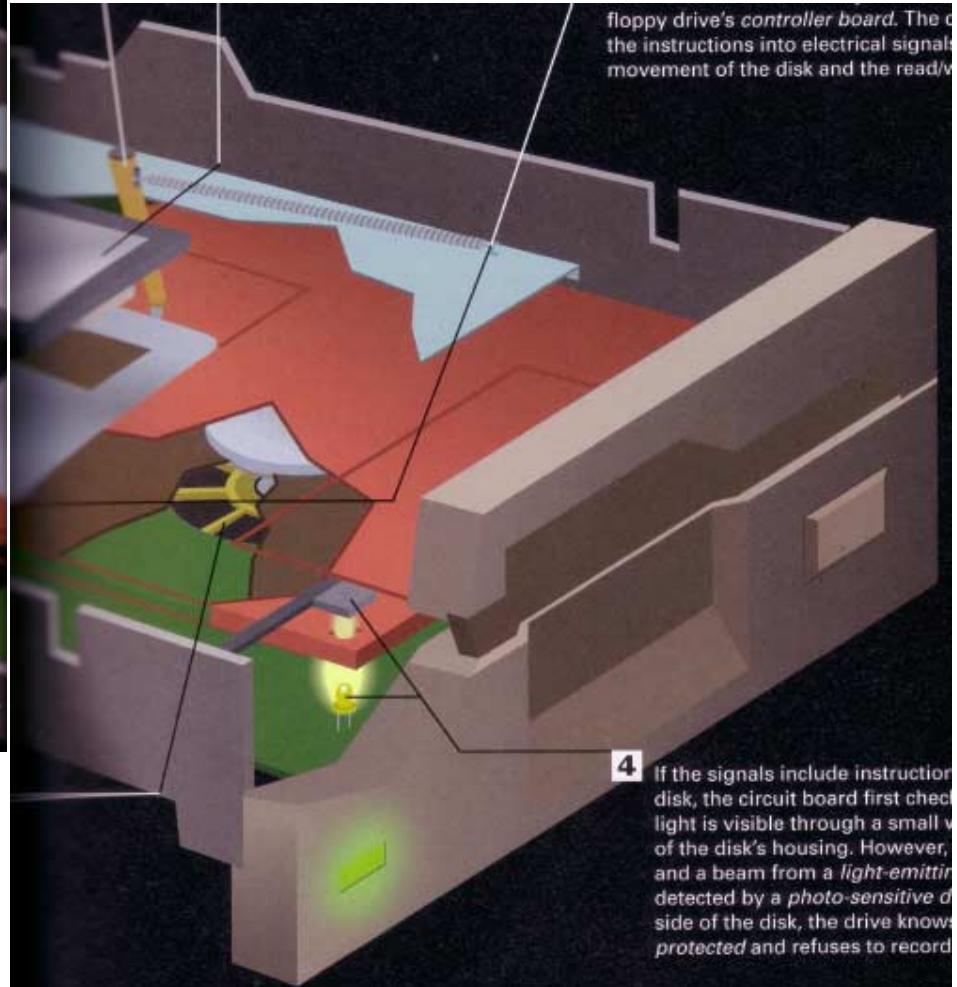
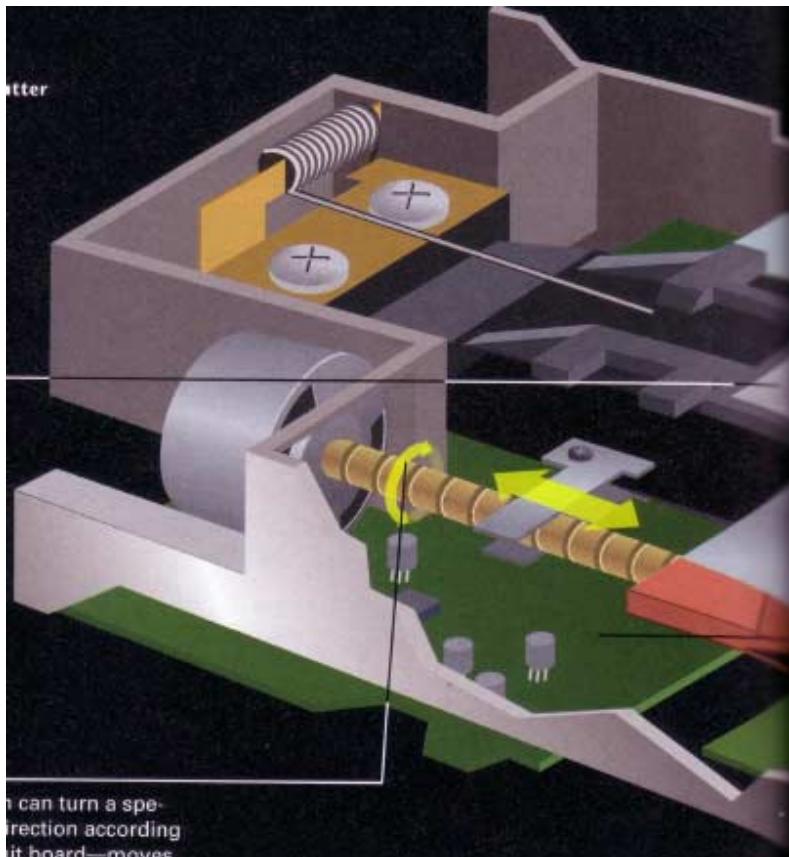
- 種類

1. ハードディスク
2. フロッピーディスク

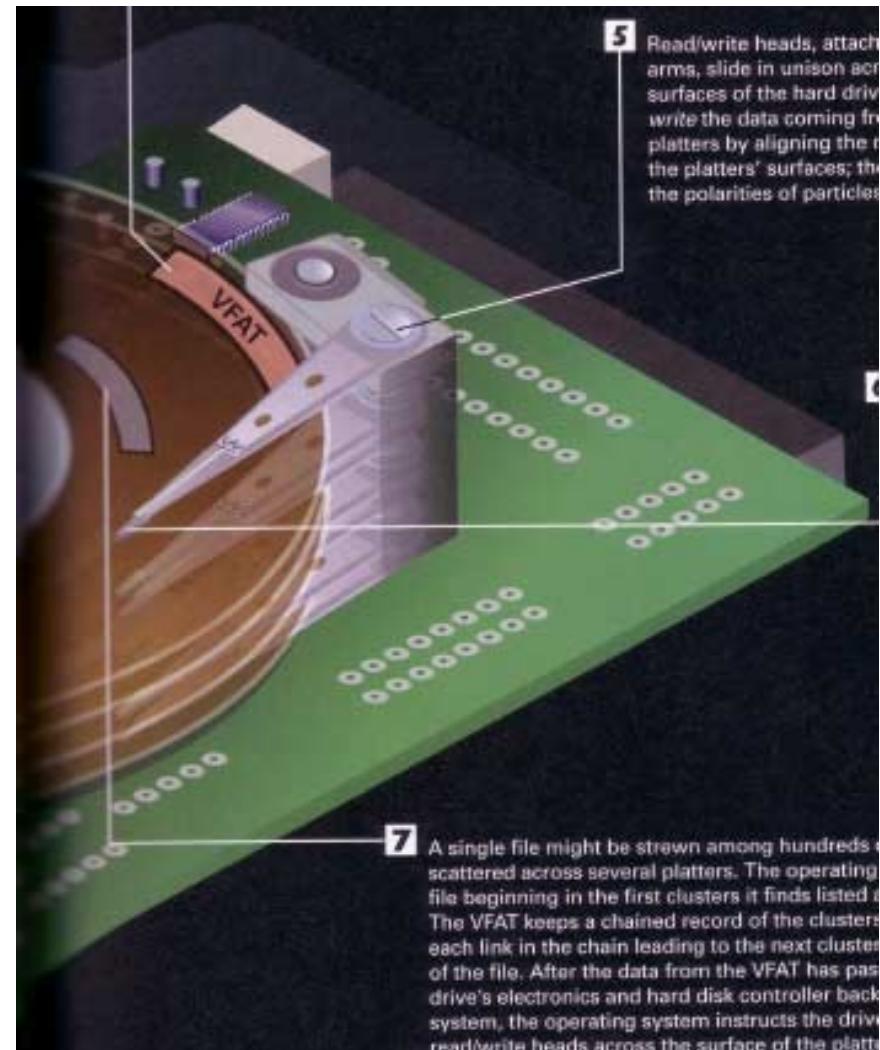
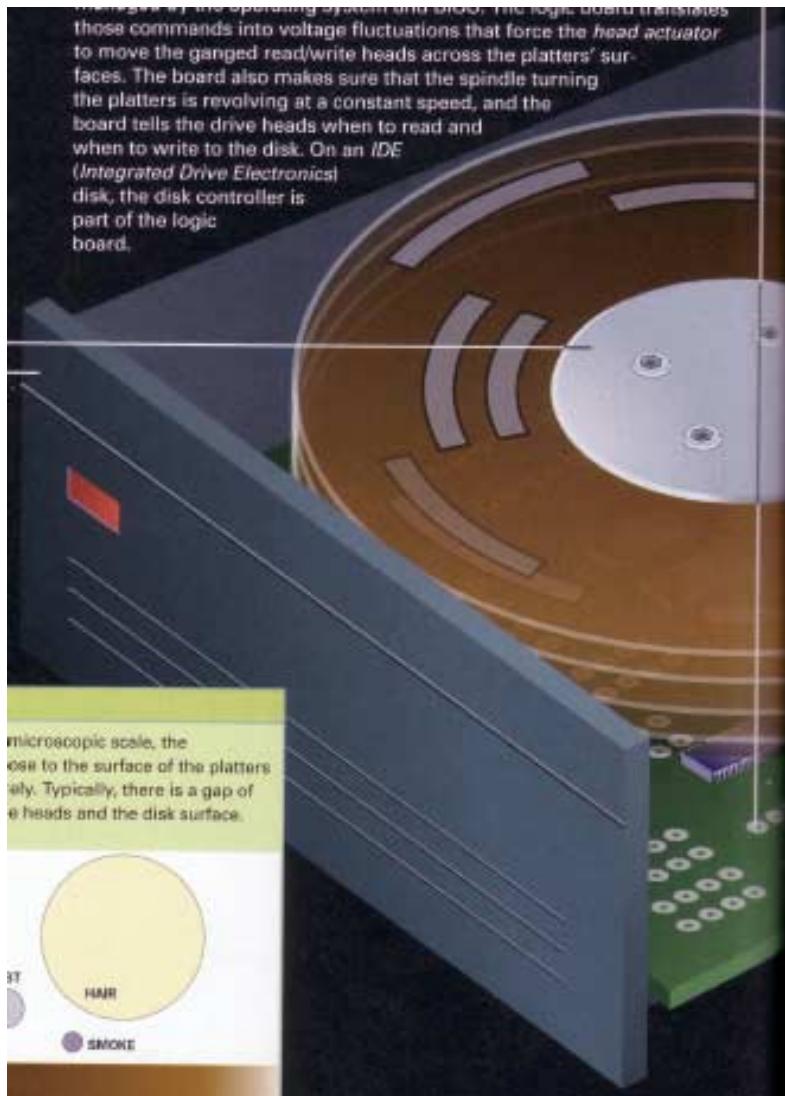
# 3.5 Inch Floppy Drive



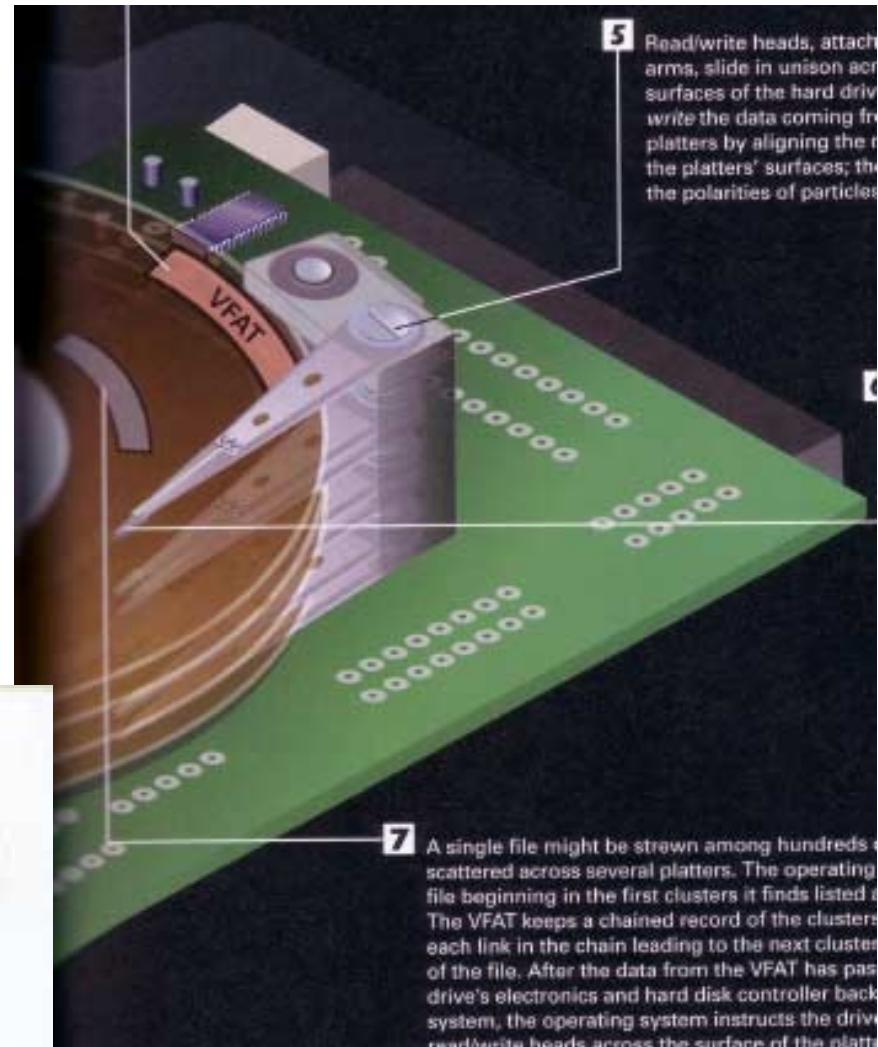
# 3.5 Inch Floppy Drive



# (Fixed) Hard Disk Drive

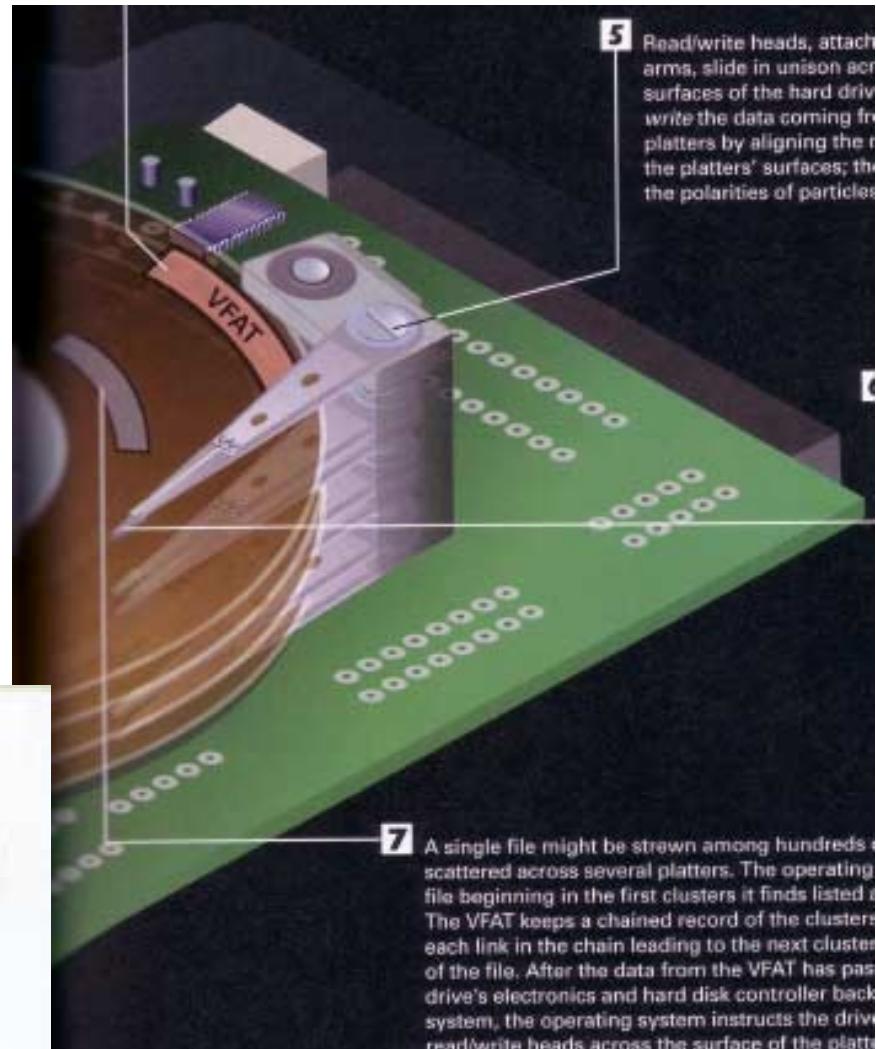
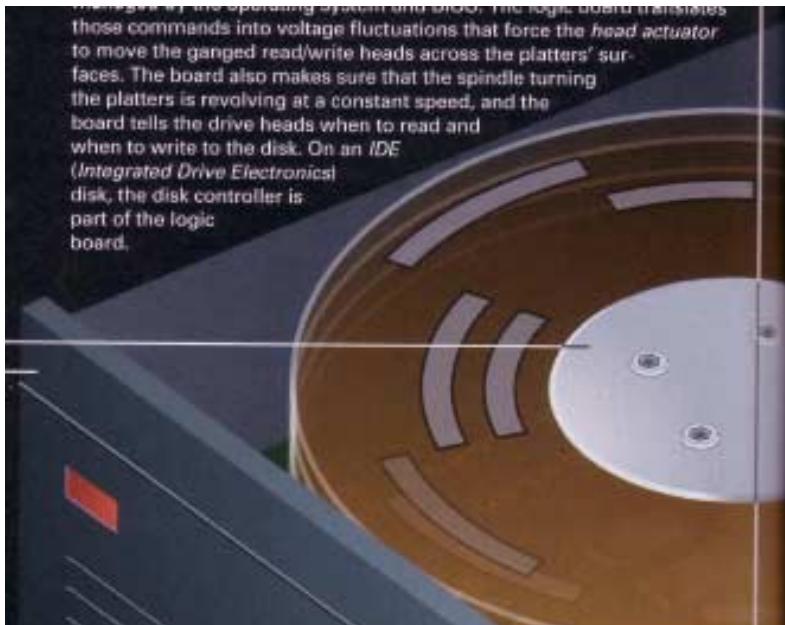


# (Fixed) Hard Disk Drive



# (Fixed) Hard Disk Drive

When the operating system sends a command, the logic board translates those commands into voltage fluctuations that force the *head actuator* to move the ganged read/write heads across the platters' surfaces. The board also makes sure that the spindle turning the platters is revolving at a constant speed, and the board tells the drive heads when to read and when to write to the disk. On an *IDE* (Integrated Drive Electronics) disk, the disk controller is part of the logic board.



(ディスクの動作)

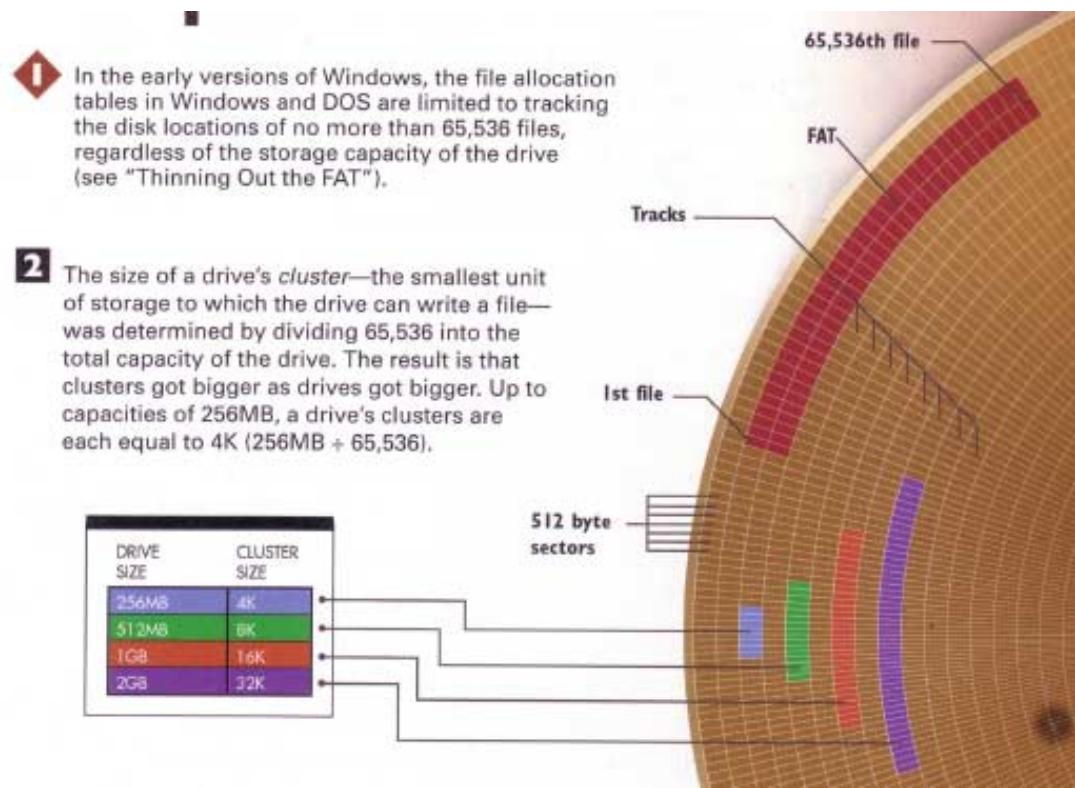
# Chapter 12

## How Disk Drives Increase Speed and Storage

- ディスクの圧縮
- ファイルの圧縮
- デフラグ

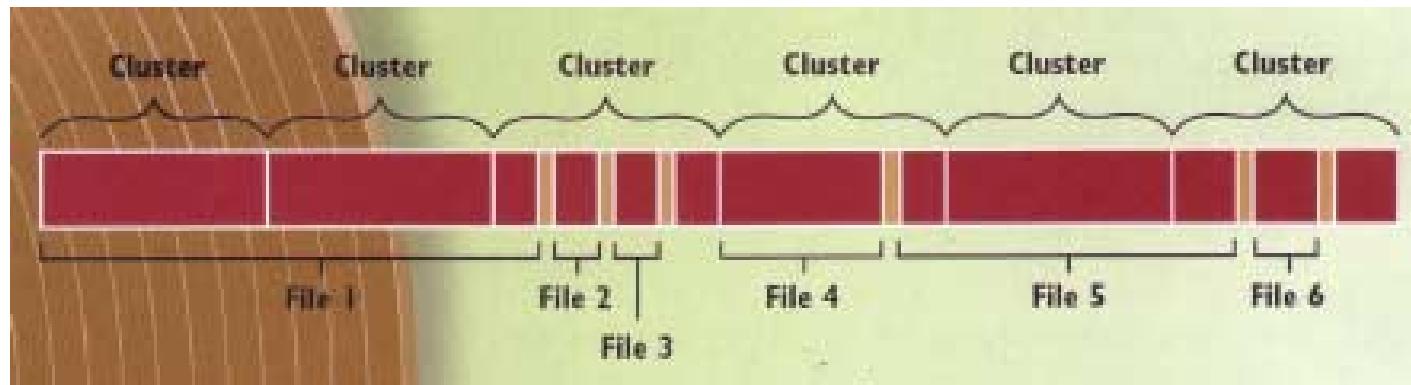
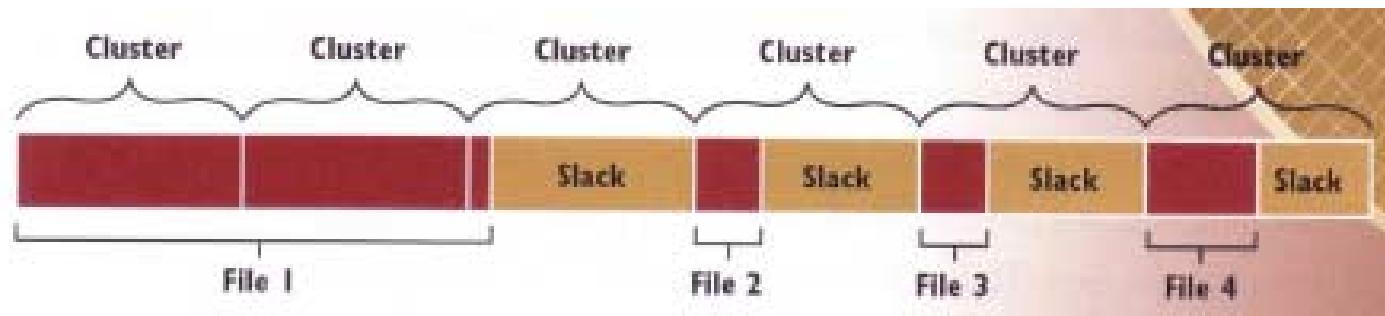
# ディスクの圧縮

- クラスタ = ディスク上の最小単位
  - ディスクサイズと OS に依存
  - FAT :  $2G \rightarrow 2[GB]/65536 = 32[kB]$



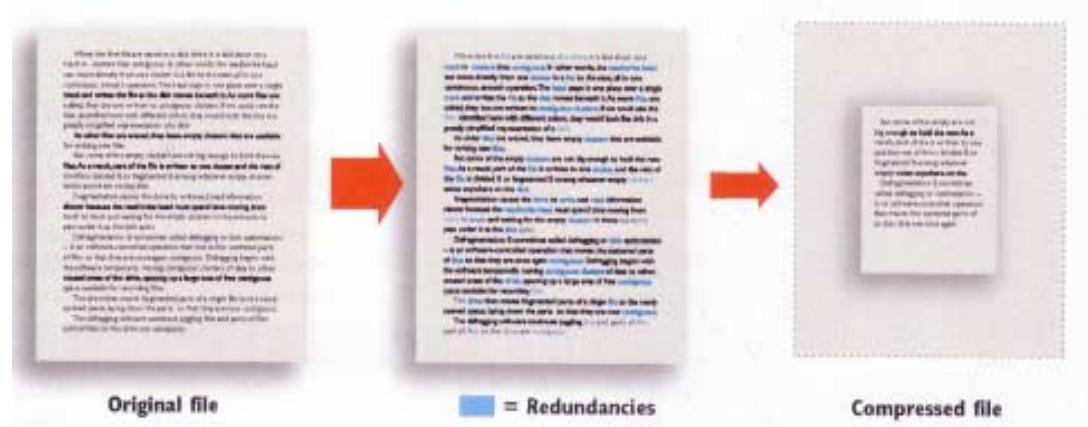
# ディスクの圧縮

- クラスタ = ディスク上の最小単位
  - ディスクサイズと OS に依存
  - FAT :  $2\text{G} \rightarrow 2[\text{GB}]/65536 = 32[\text{kB}]$



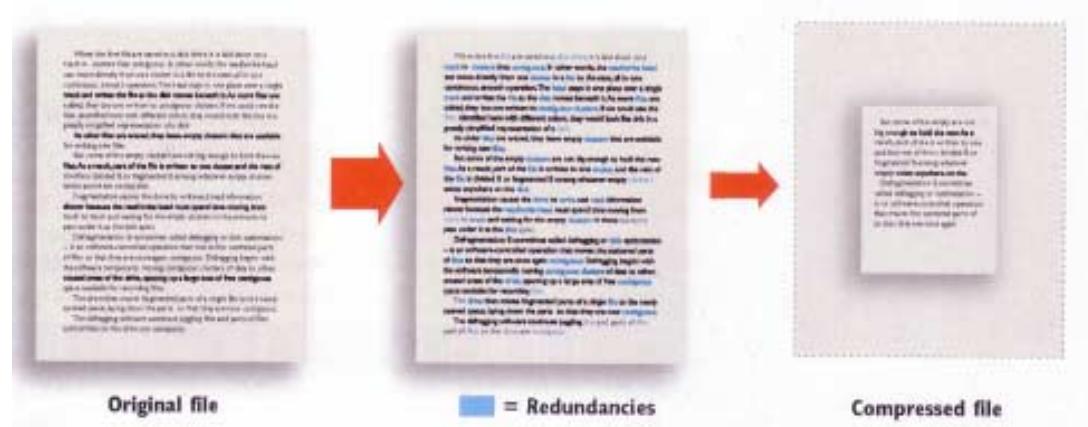
# ファイルの圧縮

## 冗長性 (redundancy)



# ファイルの圧縮

## 冗長性 (redundancy)

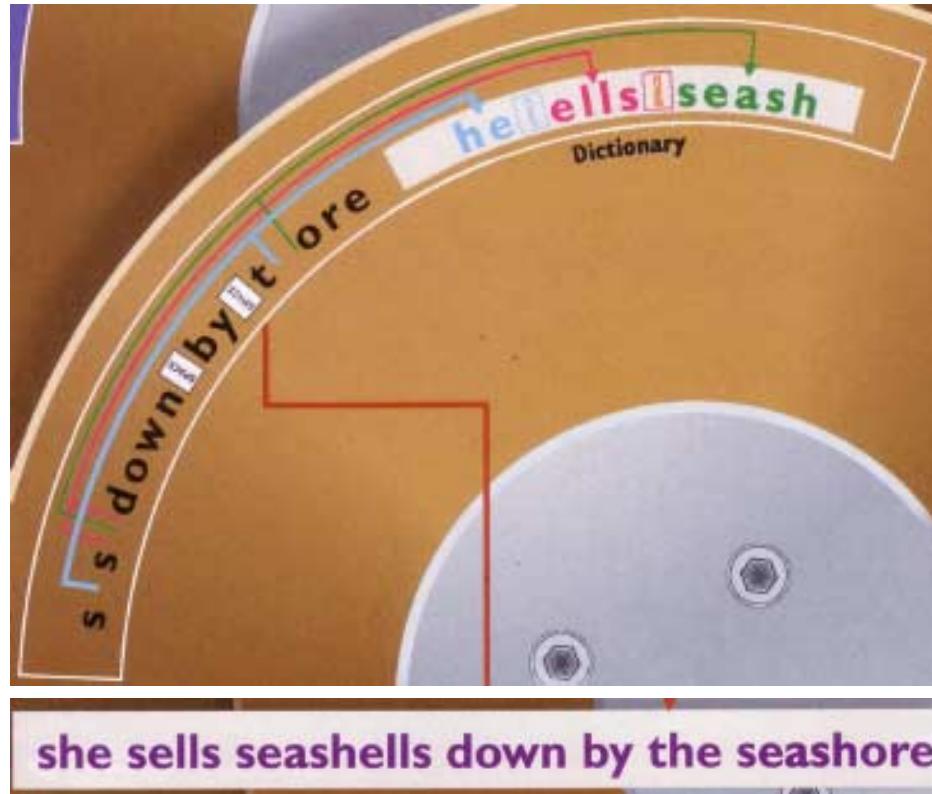


## Lempel-Ziv 圧縮



# 圧縮効率とファイルの復元

- 圧縮効率  
ファイルタイプに依存
- 復元



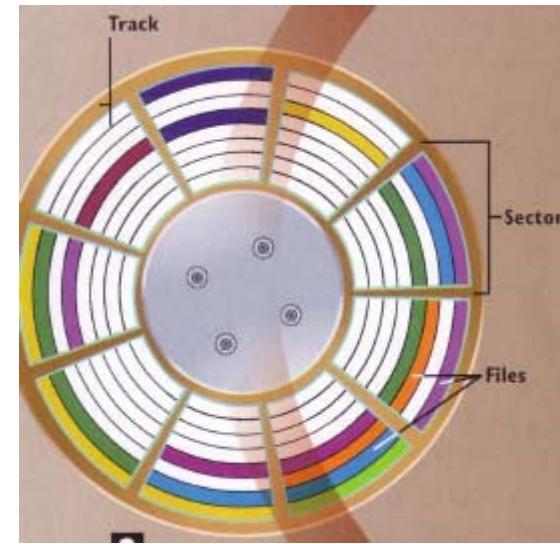
# デフラグ Defragmentation

## 1. 初期状態



# デフラグ Defragmentation

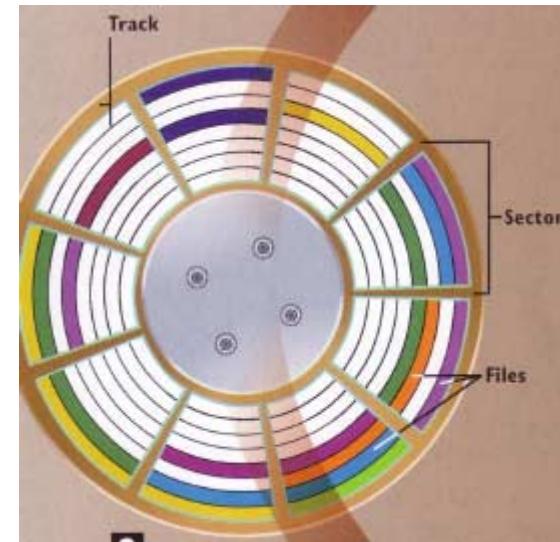
## 1. 初期状態



## 2. 一部データ消去

# デフラグ Defragmentation

## 1. 初期状態



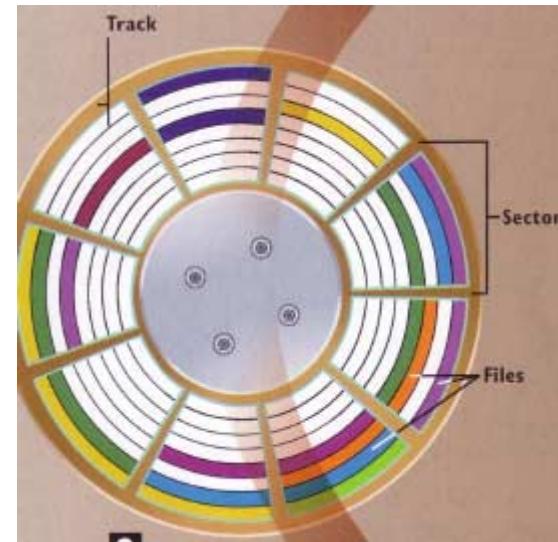
## 2. 一部データ消去



## 3. 新規ファイル分割 →

# デフラグ Defragmentation

## 4. 数クラスタ間に分割



## 2. 一部データ消去



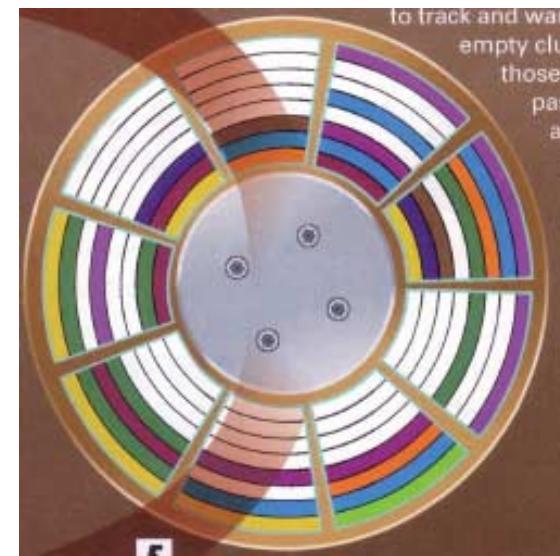
## 3. 新規ファイル分割 →

# デフラグ Defragmentation

## 4. 数クラスタ間に分割



## 3. 新規ファイル分割 →

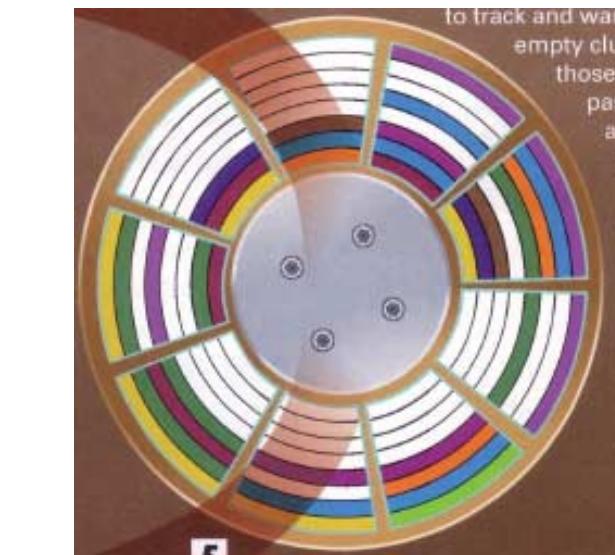


## 5. アクセス速度の低下

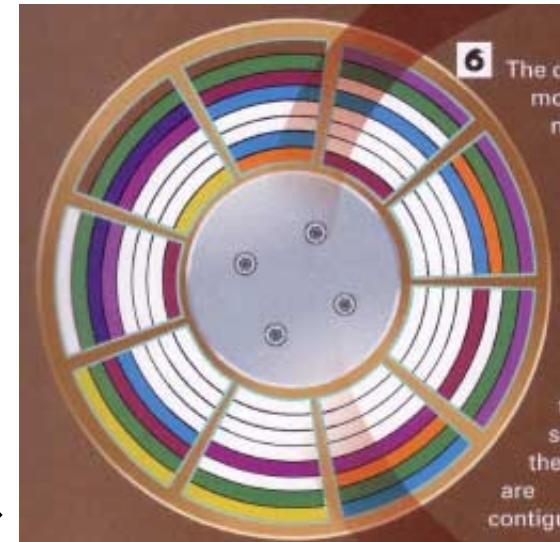


# デフラグ Defragmentation

## 4. 数クラスタ間に分割



## 5. アクセス速度の低下



## 6. デフラグ：隣接するように再配置 →