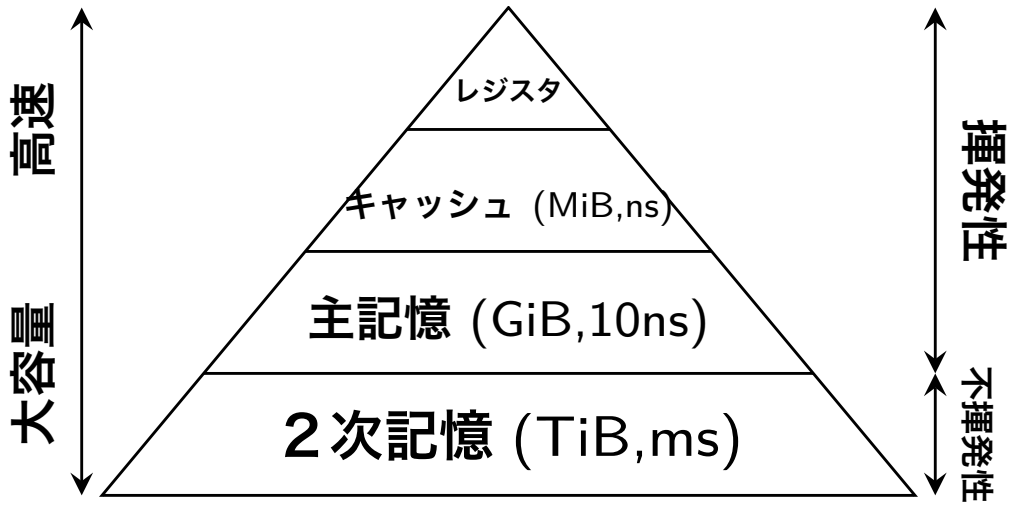
オペレーティングシステム
第１４章 二次記憶装置

https://github.com/tctsigemura/OSTextBook

記憶装置の階層（1）



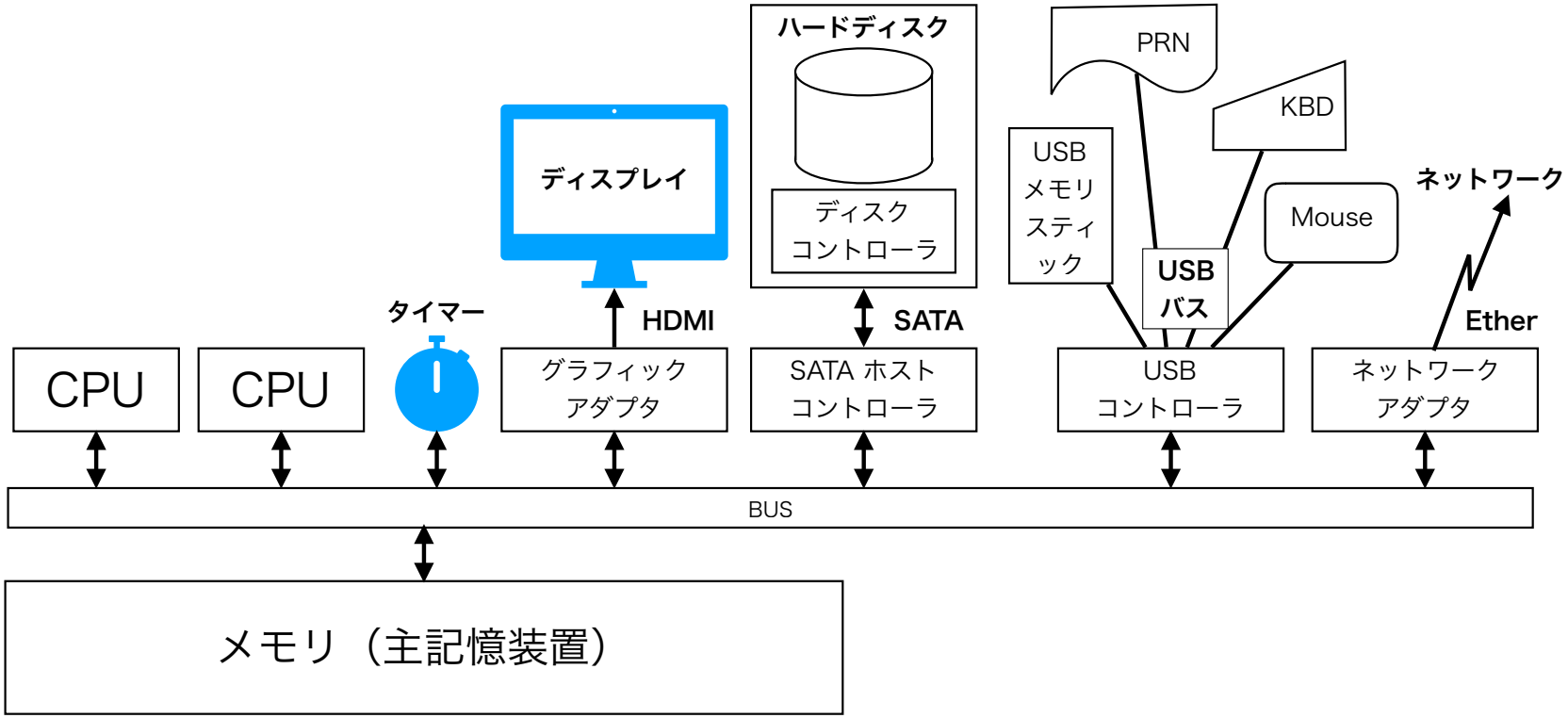
- **レジスタ**は CPU レジスタのこと．
  容量は数十バイト程度，高速アクセスが可能，揮発性
- **主記憶（メモリ）**
  アクセス時間は数ナノ秒〜十数ナノ秒程度
  容量は数 Gi バイト〜数十 Gi バイト程度，揮発性
- **二次記憶装置**
  ハードディスクや SSD（Solid State Drive）のこと．
  アクセス時間は数ミリ秒〜数十ミリ秒（ハードディスク），不揮発性

記憶装置の階層（2）

夫々の特性に合った使い方をする.
二次記憶装置の特性は次の通り.

- 大容量（ビット単価が安い）
  オペレーティングシステム，アプリケーション，データなどの
  全てを格納できる.
- 不揮発性（電源を切っても消えない）
  プログラムやデータの永続的な置き場所として適している.

二次記憶装置の種類（1）



接続方式

- CPU からはホストコントローラを介してアクセスする.
- 二次記憶装置は SATA や USB バスの先に接続される.
- USB メモリスティックやポータブルハードディスクは取り外し可能.
- 取り外し可能 => データ交換，バックアップ用途にも適する.

二次記憶装置の種類（2）

テープ型装置

- データのバックアップや輸送用（ビット単価が安い）
- **シーケンシャルアクセス**専用
- 読み出し位置まで進むために数分！！

二次記憶装置の種類（3）

ディスク型装置

- **ランダムアクセス**が可能
- ハードディスクのこと（CD-ROM などの光ディスクも仲間）
- SSD，USB メモリ，その他メモリカードも仲間

ハードディスク（1）

- システムの起動ドライブ（OS，アプリ，データ全てが置かれる）
- 仮想記憶のバックストレージとしても使用される．
- ハードディスク管理が，OS の性能や使い勝手を左右する．
- ファイル管理機構はハードディスクを前提にしていることが多い

ハードディスク（2）

セクタ・トラック・シリンダ

- 同心円の**トラック**（*Track*）
- トラックを区切った**セクタ**（*Sector*）
- トラックをまとめた**シリンダ**（*Cylinder*）

ハードディスク（3）

セクタのアドレッシング

512 バイト（4KiB）のセクタのアドレス付け方法

- *CHS*（*Clinder Head Secor*）方式
  - Clinder Head Secor の三次元アドレス.
  - Head は Track と同じ意味.
  - CHS は PC の世界で使用されてきた用語.
  - ハードディスクの物理的な構造通りのアドレッシング.
  - 過去，長く使われてきた方式.
- *LBA*（*Logical Block Addressing*）
  - セクタの通し番号（一次元）を用いる.
  - ハードディスクブラックボックス化（物理構造通が不明）
  - CHS は煩雑なだけでメリットがなくなった.

ハードディスク（4）

```
int sem;                                        // ドライバ用セマフォの番号

// restart  : 割り込みハンドラ
interrupt restart() {                           // 割込が発生したら
        semV(sem);                              // 待っていたプロセスを起こす
}

// readSct  : ブロックを読み込む
// 引数 h   : 読み込むブロックの上位ブロックアドレス
//      l   : 読み込むブロックの下位ブロックアドレス
//      buf : データを読み込むバッファ
public void readSct(int h, int l, void[] buf) {
        out(MEM_ADDR, _AtoI(buf));              // buf のアドレスを MEM_ADDR に格納
        out(BLK_ADDR_H, h);                     // BLK_ADDR にブロックアドレスを格納
        out(BLK_ADDR_L, l);
        out(SD_CTRL, READ | INT_ENA);           // 読み込み開始指示、割り込み許可
        semP(sem);                              // ブロック
}
```

- *TacOS* の *uSD* ドライバの例
  - restart()：割り込みハンドラ
  - readSct()：1 セクタ読み出しルーチン

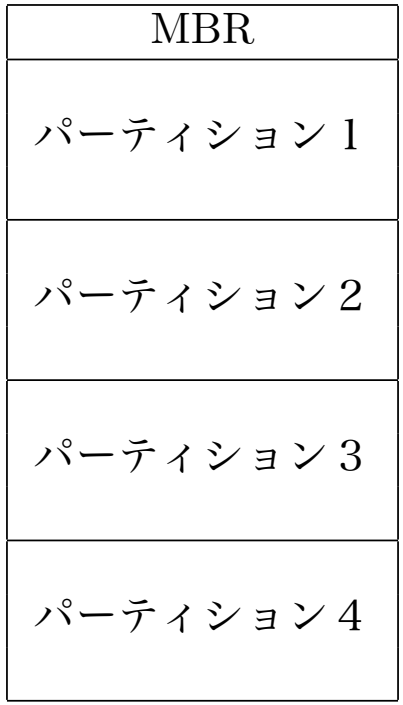
フォーマッティング（1）

ハードディスクの初期化の例

1. 低レベル（物理）フォーマット
   ディスクの表面に磁気的にトラックを書き込む.
2. **パーティション**（区画）に分割
   - 装置全体を一つの**ボリューム** => 大きすぎる
   - 区画に分割し区画をボリュームとして扱う =>
     オペレーティングシステムのパーティション
     ユーザデータのパーティション => ここだけバックアップ
   - 複数のオペレーティングシステムをインストール第 1 パーティション（ボリューム）に Windows
     第 2 パーティション（ボリューム）に Linux
     第 3 パーティション（ボリューム）に FreeBSD
3. 高レベル（論理）フォーマット
   各ボリュームの内部に該当オペレーティングシステムの
   空のファイルシステムを作る.

フォーマッティング（2）

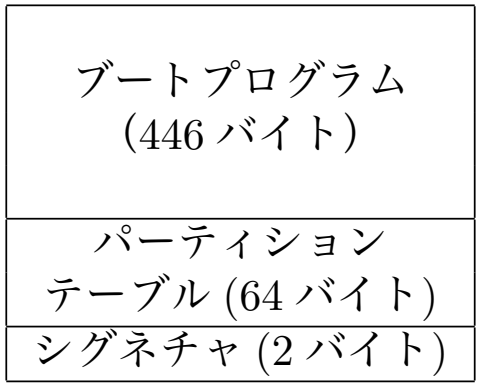
PC 用ハードディスクのパーティションの例



- *MBR*（*Master Boot Record*）
  - ハードディスクの先頭セクタ（LBA0）に格納
  - MBR のサイズは 512 バイト
  - 内容は**ブートプログラム**と**パーティションテーブル**

フォーマッティング（3）

PC 用ハードディスクの *MBR* の内容



- *MBR*（*Master Boot Record*）（*512* バイト）
  - ブートプログラム（446 バイト）
    PC の機械語プログラム（OS を起動するためのプログラム）
  - パーティションテーブル（64 バイト）
    各パーティションの位置と大きさ等を記録する 4 業の表
  - シグネチャ（2 バイト）
    フォーマッティングされている目印（55H, AAH）

フォーマッティング（4）

PC用ハードディスクのパーティションテーブルの例

Flag (1)	Start CHS(3)	Type (1)	End CHS(3)	Start LBA(4)	Size (4)
80H	???	06H	???	0000003FH	00003F00H
80H	???	A5H	???	00003F3FH	0000BD00H
00H	???	???	???	????????	??????
00H	???	???	???	????????	??????

項目	バイト数	意味
Flag	1	80H アクティブ/ 00H インアクティブ
Start CHS	3	開始アドレス (CHS 表現)
Type	1	ファイルシステムの種類
End CHS	3	終了アドレス (CHS 表現)
Start LBA	4	開始アドレス (LBA 表現)
Size	4	セクタ数 (LBA 表現)

Type	意味
00H	空き
01H	FAT12
04H	FAT16(小)
06H	FAT16(大)
07H	NTFS
0BH	FAT32
83H	Linux(ext2)
A5H	FreeBSD

パーティションを見つけるプログラムの例

```
int[] bpbLba  = { 0, 0 };                          // パーティションの開始 LBA

void readMBR() {
  char[] buf = malloc(BLKSIZ);                     // 1 セクタ分のバッファを確保
  readSct(0,0,buf);                               // MBR を読み込む

  for (int i=446; i<510; i=i+16) {                 // パーティションテーブルについて
    int active = ord(buf[i]);                      // アクティブフラグ
    int fType  =ord(buf[i+4]);                     // ファイルシステムタイプ

    if((active & 0x80)!=0 && fType==0x06){  //アクティブな FAT16 パーティション
      ld32(bpbLba,wordLE(buf,i+10),wordLE(buf,i+8));// パーティションの開始 LBA
      free(buf);                                   // バッファを解放して
      return;                                      // 戻る
    }
  }
  panic("readMBR");                                // 最後まで行くとエラー
}
```

TacOS が PC 用の uSD カードから FAT16 パーティションを見つける.

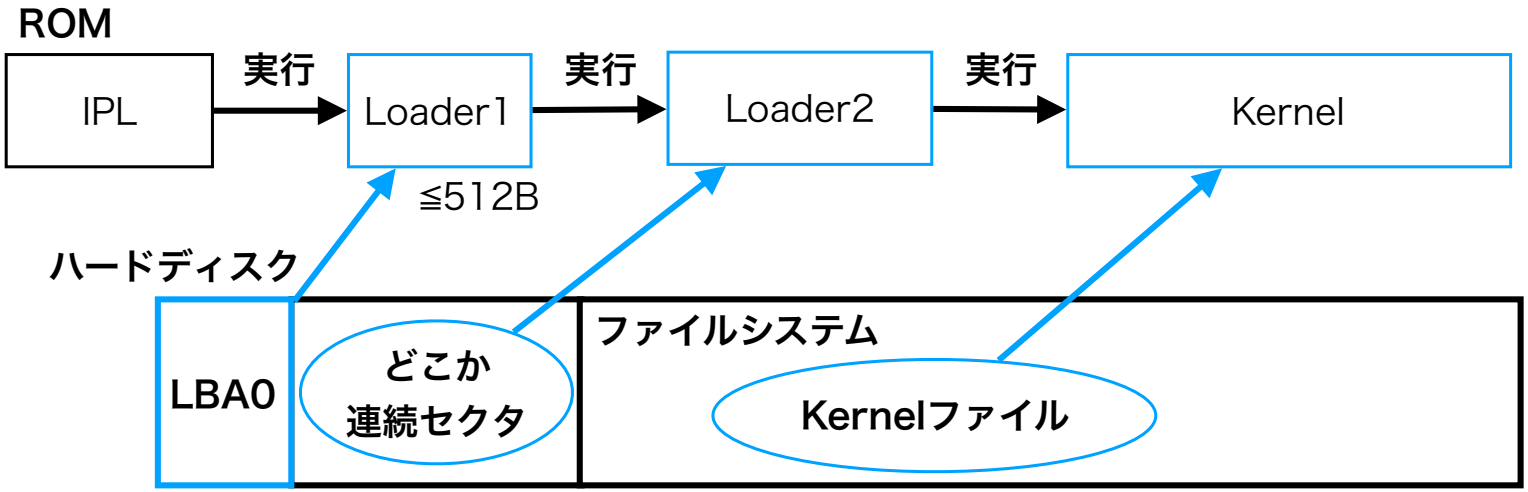
ブートストラップ（1）

PC の場合を例にブートストラップを説明する.

- ハードディスクから OS を起動する作業のこと.
- OS のカーネルを格納したファイルを見つけてロード・実行する.
- PC の製造時にはどんな OS がインストールされるか分からない.
  => ブートストラップは後で変更できる必要がある.
- 以下に説明する段階を経て OS をブートする.
- 以下の方法が PC では標準的であるが様々な変種がある.
  (段階が多い場合，強力なブートマネージャを備えている場合)

ブートストラップ（2）

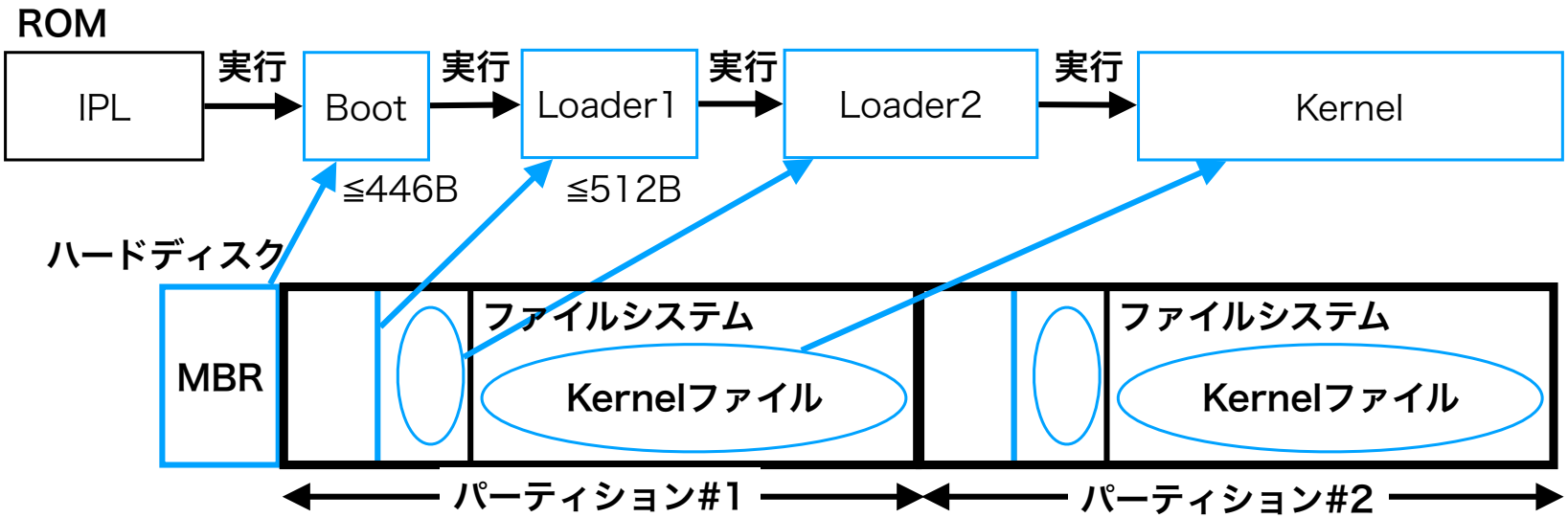
ハードディスク = ボリュームの場合



- *IPL*（*Initial Program Loader*）
  PC の ROM に格納されており電源 ON と同時に動作開始
- ブートローダ（第 1 段階：Loader1）*512* **バイト以内**
  LBA0 に格納され IPL によってロード・実行される.
- ブートローダ（第 2 段階：Loader2）
  ディスク上のどこか連続セクタに格納され Loader1 がロード・実行.
  サイズに制限がない => 高機能にできる.
- OS のカーネル
  ファイルシステムにファイルとして格納され Loader2 がロード・実行.

ブートストラップ（3）

パーティション = ボリュームの場合



- *IPL*（*Initial Program Loader*）
- ブートセレクタ・ブートマネージャ（Boot）*446* **バイト以内**
  LBA0（MBR）に格納され IPL によってロード・実行される.
  メニューを表示してユーザに OS のパーティションを選択させる.
  (勝手に次に進むものもある.)
- ブートローダ（第 1 段階：Loader1）*512* **バイト以内**
- ブートローダ（第 2 段階：Loader2）
- OS のカーネル

練習問題

1. 次の言葉の意味を説明しなさい.
   - 二次記憶装置
   - 揮発性・不揮発性
   - 記憶の階層
   - テープ型装置・ディスク型装置
   - シーケンシャルアクセス・ランダムアクセス
   - セクタ・トラック・シリンダ
   - CHS・LBA
   - パーティション
   - ブートストラップ