

# Linux のインストールに 必要な基礎知識

理学部惑星学科

流体地球物理学教育研究分野 M1

服部 蒼紀

# 目次

1. はじめに
2. パーティション
3. ファイルシステム
4. OS 起動までの流れ
5. 最低限セキュリティ
6. まとめ

# 1. はじめに

# はじめに

## OS をインストールする前に行うこと

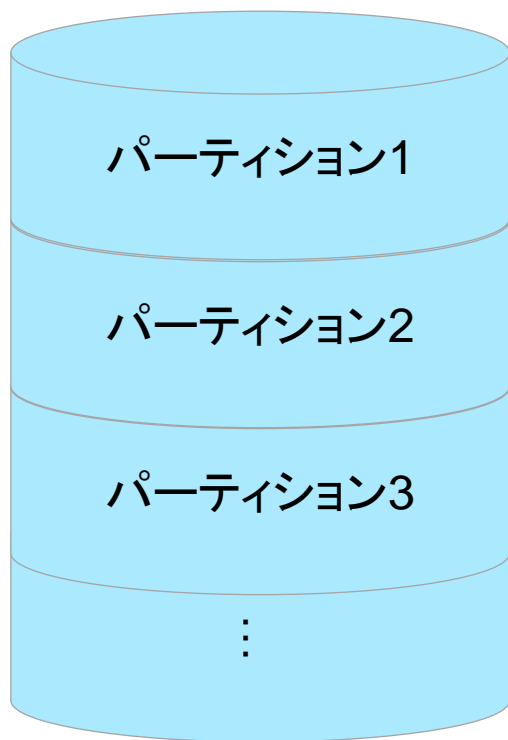
- 言語, 場所, キーボードの設定
- ネットワークの設定
- ユーザとパスワードのセットアップ
- ディスクのパーティショニング
  - パーティションへのファイルシステム作成
- パッケージマネージャの設定
- ハードディスクへの GRUB ブートローダのインストール

## OS をインストールの後に行うこと

- OS を起動
- OS の設定

## 2. パーティション

# パーティションとは

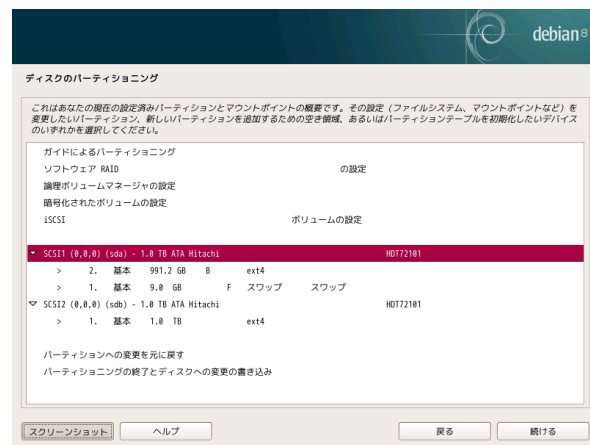


記憶装置

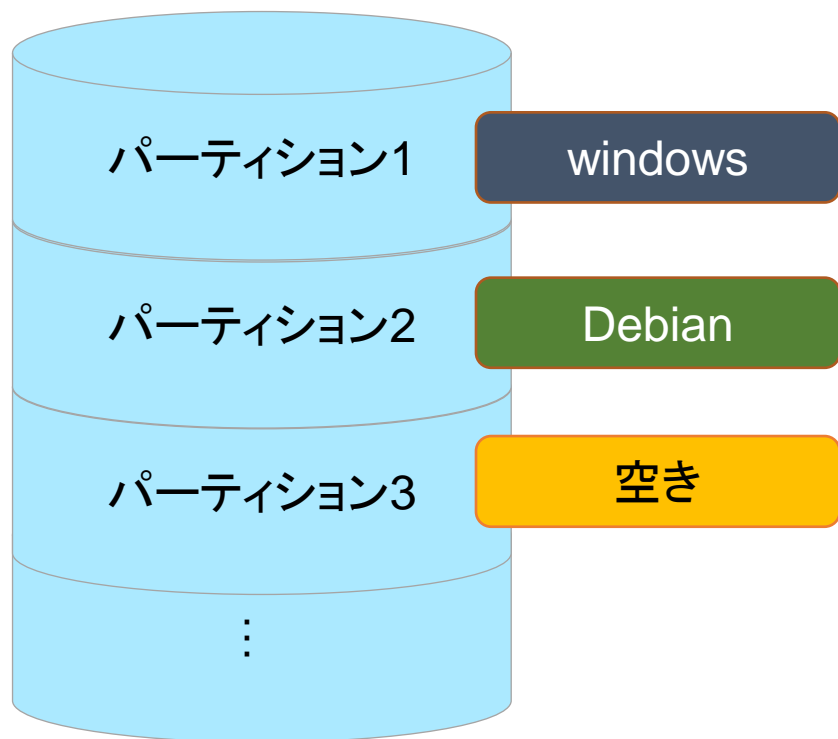
- 記憶装置内に作成できるデータの区画
  - 区画のサイズは自由に設定できる
  - 区画数は上限がある
  - OS をインストールする前に行う
- 分割の方法 (あとで説明)
  - MBR 方式: BIOS 環境で利用
  - GPT 方式: UEFI 環境で利用
- パーティショニング: パーティションを作成すること
  - 「パーティションを切る」とも

# パーティションとは

- 「パーティションを切る」には？
  - 専用ツールを使う  
例 : cfdisk, Partition Magic, GNU Parted など
  - 今回使うインストーラにはパーティションを操作する機能がついている
- 一般に、パーティションを変更すると過去のデータは読めなくなる  
⇒ **データの損失に注意！**



# パーティションの利点



- ハードディスク内に障害が起こったとき、影響を受ける範囲をパーティション単位に収められる
- 1つのハードディスクに、異なるOSやファイルシステムを入れられる (マルチブート)



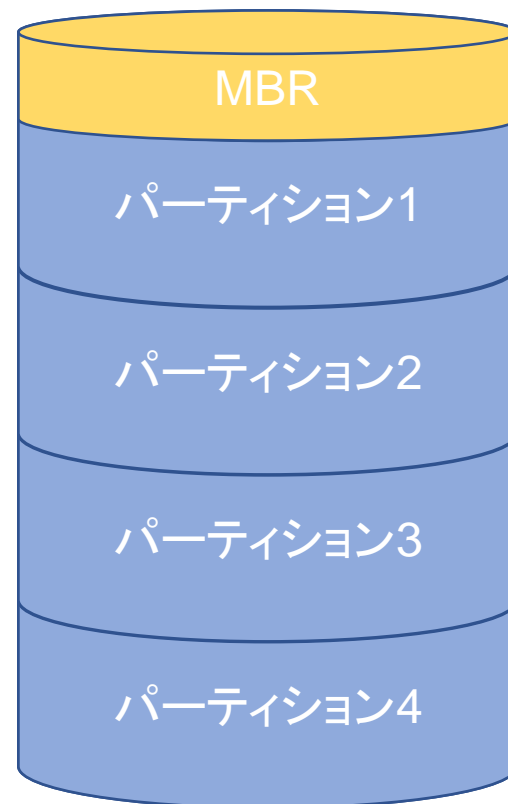
# 用途別のパーティションの呼び方

- スワップパーティション
  - メモリとして使うことができるパーティション (メインメモリの容量が足りない時に使う)
- ルートパーティション
  - ルートディレクトリ (ディレクトリ階層構造の頂点) が入っているパーティション
- アクティブパーティション(ブートパーティション)
  - OS を起動できる状態になっているパーティション

# MBR 方式のパーティション構造

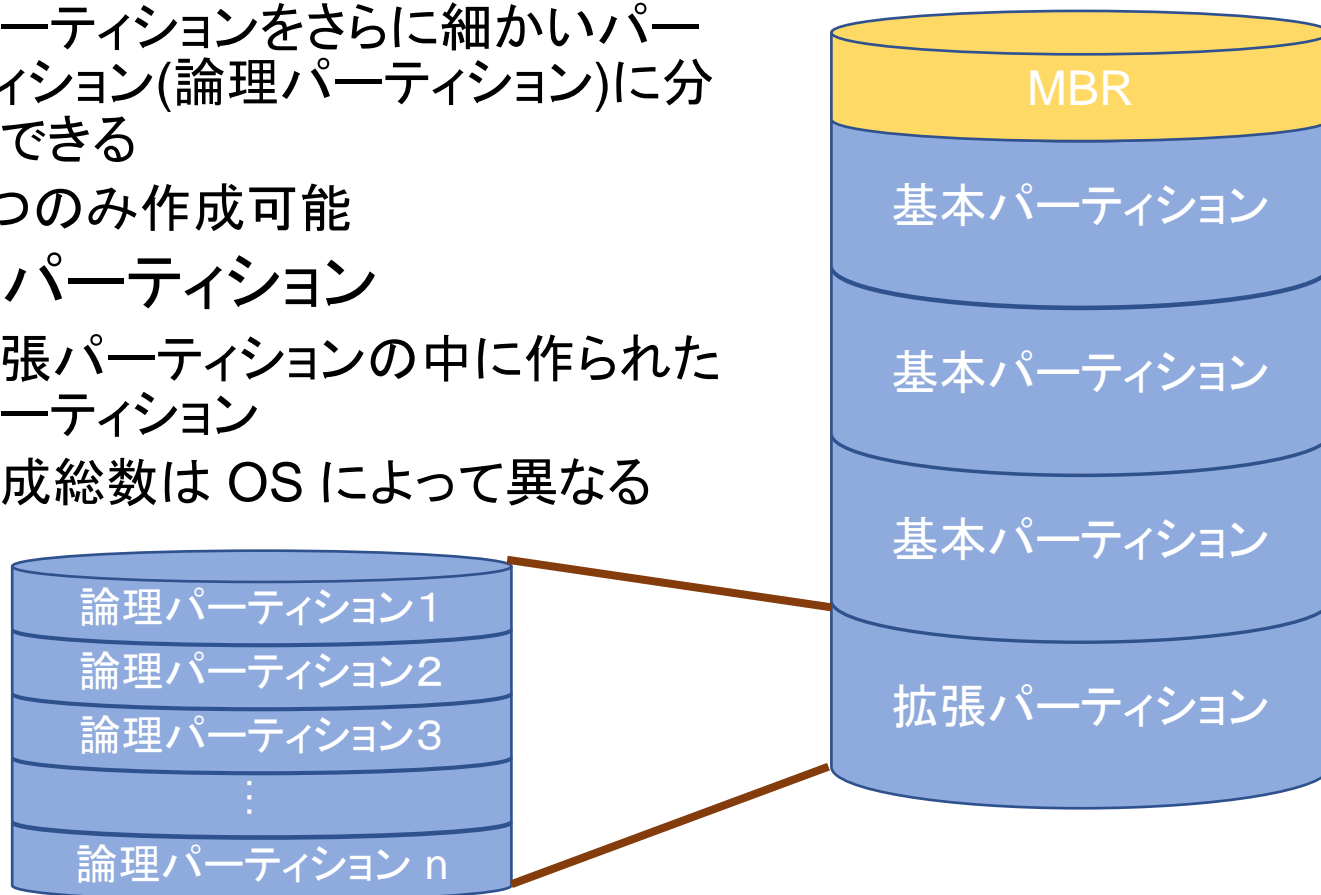
古い方式, 2TB までの HDD しか認識できない

- マスターブートレコード(MBR)
  - BIOS を読んだ CPU がハードディスクにアクセスするとき最初に見るセクタ (512 バイト)
- 基本パーティション
  - OS をインストールできるパーティション
  - 1つの記憶装置につき4つまで作成できる
  - さらにパーティションを増やしたい場合は拡張パーティションを使う



# MBR 方式のパーティション構造

- 拡張パーティション
  - パーティションをさらに細かいパーティション(論理パーティション)に分割できる
  - 1つのみ作成可能
- 論理パーティション
  - 拡張パーティションの中に作られたパーティション
  - 作成総数は OS によって異なる



# MBR (Master Boot Record) とは

- BIOS を読んだ CPU がハードディスクにアクセスするとき最初に見るセクタ (512 バイト)
  - PC 起動時にこの領域にあるプログラムが実行
- MBRの中身
  - パーティションテーブル
    - パーティションの情報の記録
    - 4 パーティション分の情報が書き込める
  - ブートシグニチャ
    - MBRを有効にするための署名
    - 第一セクタの最後の 0xAA55 というマジックナンバーのこと
  - ブートローダ
    - どの OS を使うかを指定するプログラム
    - 選択される OS が入っているパーティションの先頭にあるカーネルローダをメインメモリへ転送するようCPUに命令する
    - 今回はGRUBというブートローダを使用

# GPT 方式のパーティション構造

新しい方式, 今回はこちらを使う

MBR方式と比べて作成できるパーティションの最大数, 認識できる HDD の最大容量が大幅に増加 (128 個, 8 ZB)

- **MBR**
  - 旧式の BIOS に対応
- **GPT ヘッダ**
  - パーティションテーブルや EFI システムパーティションの位置情報を保持
- **パーティションテーブル**
  - パーティション情報の保持
    - 位置やファイルシステム
- **EFI システムパーティション**
  - ブートローダを保持



# 3. ファイルシステム

# ファイルシステムとは

- OSの機能のひとつ
- パーティションを作成しただけではデータを読み書きできない
  - 読み書きをするには**セクタ**を指定しなければならない
  - ディスクはセクタの情報 (使用中か空いているか) を管理していない
  - セクタの情報を管理する存在が必要  
=ファイルシステム
- **セクタ**: 補助記憶装置のデータを読み書きする最小単位
  - HDD では 512 バイトで 1 セクタ

# ファイルシステムとは

- パーティション上におけるデータの記録・管理形式
- パーティションをフォーマット(初期化)することによって作成
  - ディスク内にデータを管理するための領域を確保
  - ハードディスクに OS をインストールできるようになる
- OS, 記憶装置によって扱えるファイルシステムの種類は異なる
  - ファイルシステムの種類によってパーティションファイルの最大サイズ, ファイル名の最大文字数などが異なる

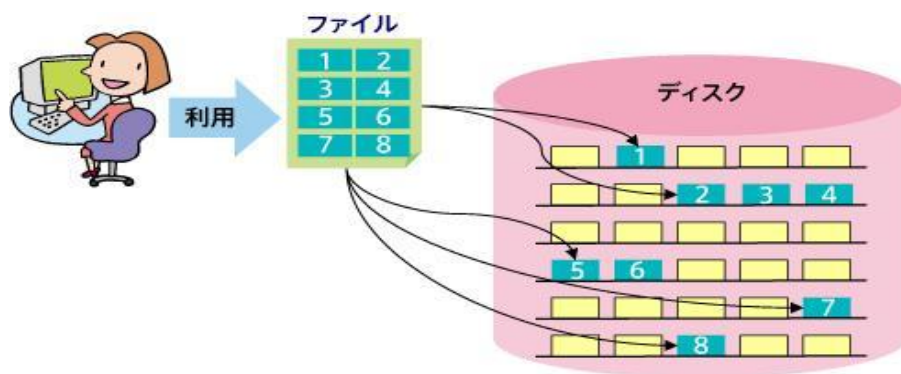


# ファイルシステムの主な役割

1. ファイルイメージの提供
2. ディスクの空きスペースの管理
3. 階層ディレクトリの提供

# 役割 1. ファイルイメージの提供

- セクタを合わせて 1 つのファイルであるかのように見せている
  - Windows や UNIX/Linux では 1 セクタ単位で HDD のデータを読み書きしていることが多い
  - 1 つのファイルの情報がいくつかのセクタに分けられ、ディスク内に存在する

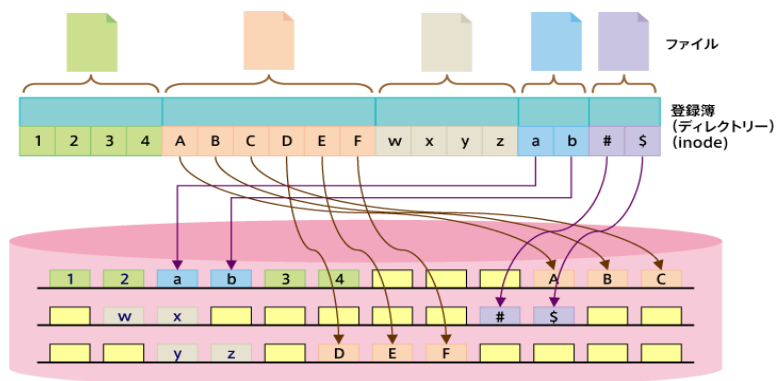


セクタがディスク内であればら  
(データの書き込みを繰り返すので)  
→ディスクだけでは、セクタがどのファイル  
のものか分からない  
→どうやってセクタを管理するか？

引用元  
佐野 正和著  
整理整頓を行う縁の下の力持ち～ファイルシステム  
企業ITとマネジメント Enterprise Zine  
<http://enterprisezine.jp/article/detail/272/>

## 役割 2. ディスクの空きスペースの管理

- ディスクのどのセクタを使っていいか (使用中か空きか) の情報を管理
  - ファイルシステムでは、ディスク内のセクタの情報を **inode** に記録
- inode: ひとつのファイルがどのセクタによって構成されるか (ポインター情報) を記録しておく場所
  - 帳簿のようなもの



引用元

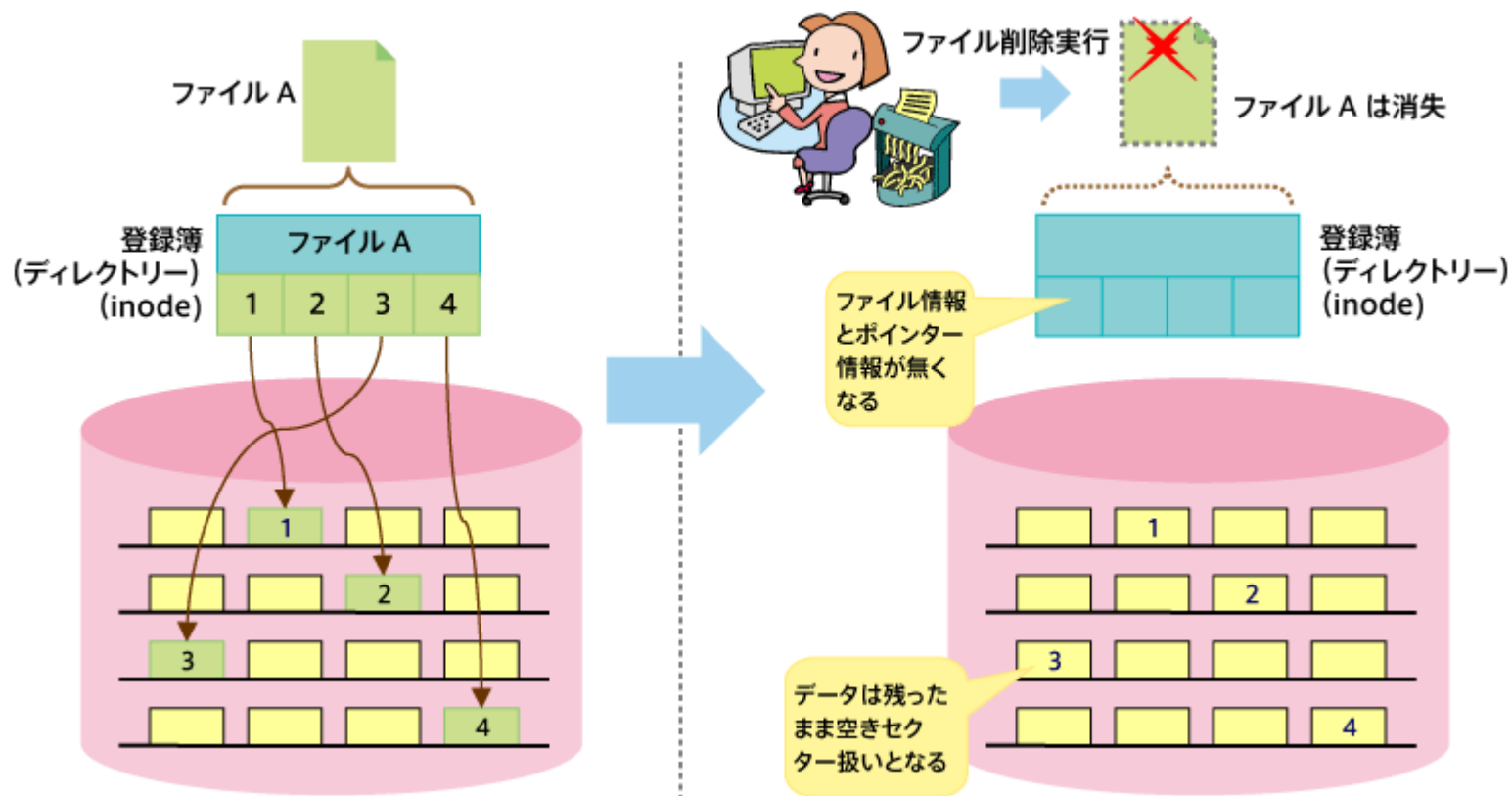
佐野 正和著

整理整頓を行う縁の下の力持ち～ファイルシステム

企業ITとマネジメント Enterprise Zine

<http://enterprisezine.jp/article/detail/272/>

## 役割 2. ディスクの空きスペースの管理



引用元

佐野 正和著

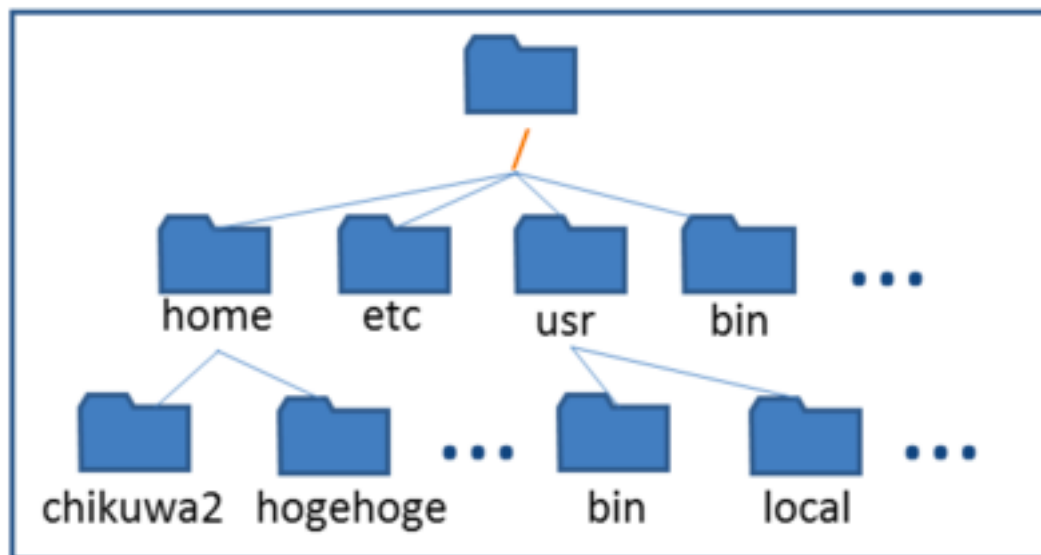
整理整頓を行う縁の下の力持ち～ファイルシステム

企業ITとマネジメント Enterprise Zine

<http://enterprisezine.jp/article/detail/272/>

## 役割 3. 階層ディレクトリの提供

- ファイルは基本的にディレクトリに格納され、ディレクトリは階層的に配置される。
- ディレクトリを使ってファイルを整理することで、たくさんのファイルを上手に管理できる。



# 主なファイルシステムの例

- ファイルシステムは OS とともに発展してきた
  - 各 OS に対応してファイルシステムがある
    - Windows : FAT32, NTFS
    - Macintosh : HFS, HFS+, FAT32
    - Linux : ext4, ext3, ext2, FAT32 ...etc
  - ファイルシステムによって、パーティションの最大容量などが異なる
    - 例: ext4: 1 EB, NTFS: 16EB など
  - 今回は ext4 を利用する

## 4. OS 起動までの流れ

# OS 起動の大まかな流れ

## BIOS/UEFI 起動

→パーティションテーブル読み込み

- パーティションテーブル:パーティション情報の記録

→ブートローダ実行

- ブートローダ:カーネルローダをメインメモリへ転送するように CPU へ命令

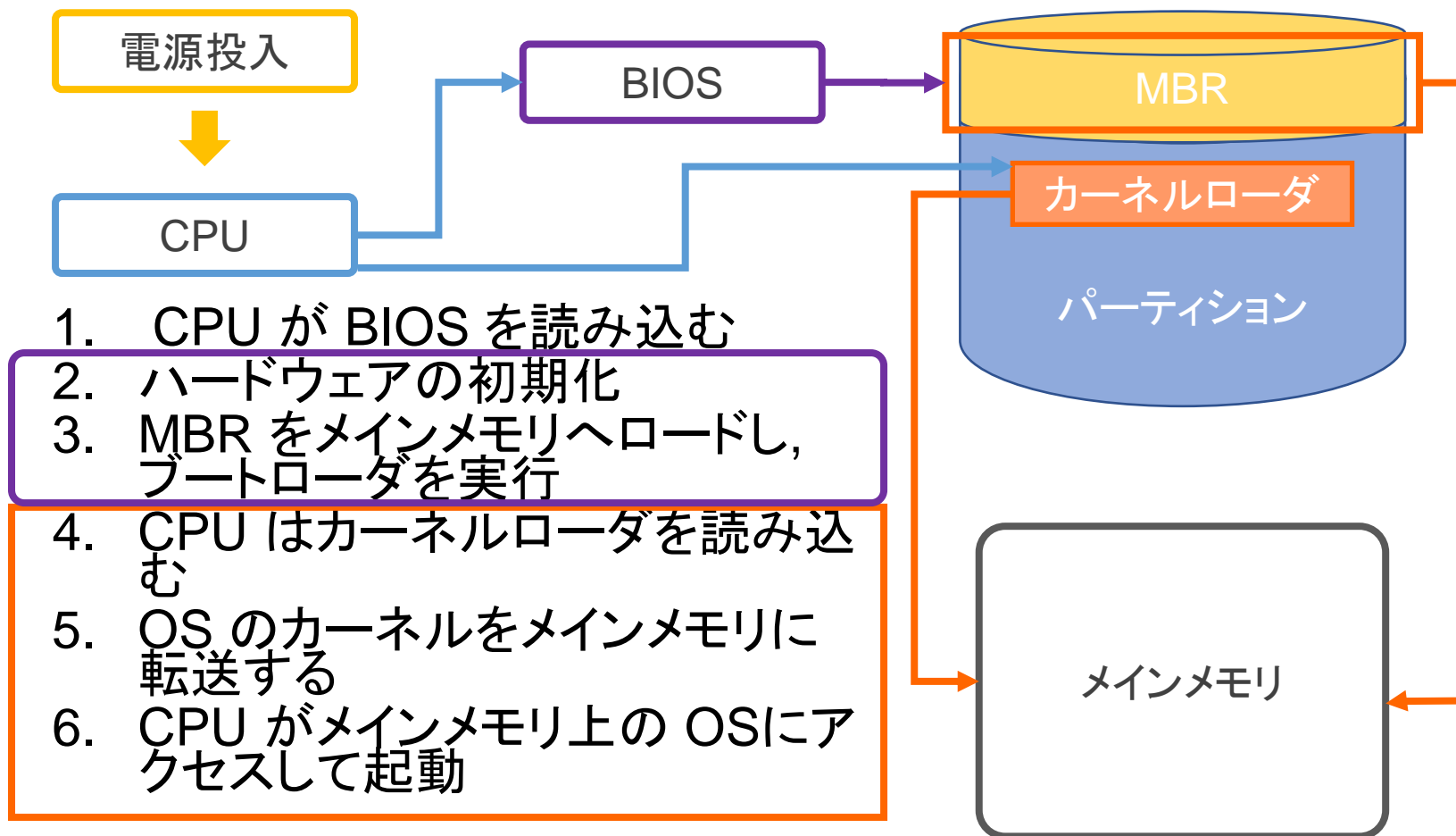
→カーネルローダ読み込み

- カーネルローダ:OS のカーネルをメインメモリに転送

→OS 起動

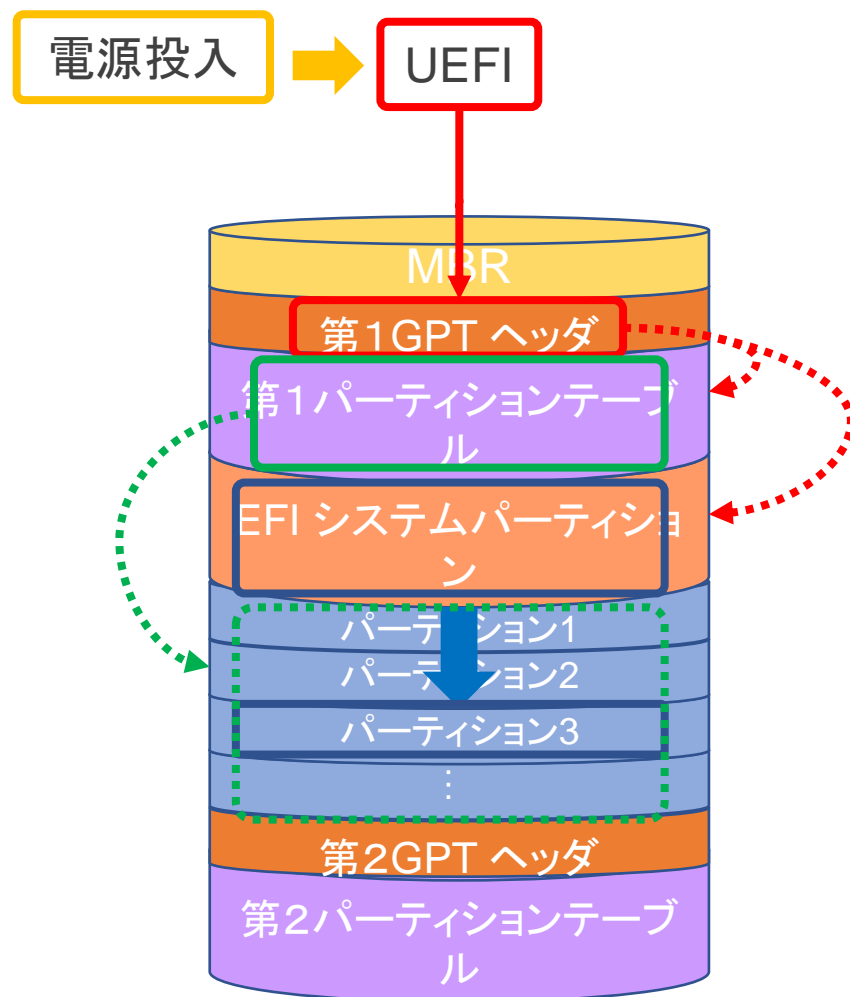


# OS 起動までの流れ(MBR方式)



# OS 起動までの流れ(GPT方式)

1. GPTヘッダでパーティションテーブルとEFIシステムパーティションの位置を把握
2. パーティションテーブルでパーティションの位置とファイルシステムを確認
3. EFIシステムパーティションにあるブートローダでパーティションのカーネルローダを読み込む



# 5. 最低限セキュリティ

# セキュリティホール

- セキュリティホールとは?
  - コンピュータソフトウェアの欠陥(バグ)の一つ
    - 例: 実行権限を越えて、本来できないはずの操作が行えるなど
- セキュリティホールをなくすには?
  - セキュリティ情報をチェック
    - Debianセキュリティ情報
      - <http://www.debian.org/security/>
      - メーリングリストで情報を受け取ることも可能
    - JPCERT
      - <http://www.jpcert.or.jp/>
      - 様々なソフトウェアの注意喚起など、情報が集まっている
- セキュリティホールがあると分かっているソフトウェアは
  - 使わない
  - アップデートする

# OS・アプリケーションの アップデート

- Windowsの更新プログラム

- Windows update (OS そのものの更新プログラム)
- Office update (Word, Excel などの更新プログラム)

- UNIXやLinuxでは

- binary update (パッケージのアップデート機構)
- Debian 系: apt

```
# apt update ←パッケージの最新リストを取得
```

```
# apt upgrade ←アップデートの実行
```

- # は管理者権限でコマンドを入力していることを表す (実際入力するときは sudo で置き換えればよい)
  - apt が使えるのはパッケージで入手したものについてのみ自分でインストールしたものは自分でアップデートする
- CentOS, Red Hat 系: yum

## その他

- 余計なサービスは入れない
- ウイルス対策ソフトを入れる
  - 定期的にスキャンする

## 6. まとめ

# まとめ

- パーティション
  - ハードディスクをいくつかに分けた領域
  - パーティションを切ることでハードディスク内の障害をパーティション単位に収められる
- ファイルシステムの役割
  - ファイルイメージの提供
  - ディスクの空きスペースの管理
  - 階層ディレクトリの提供
- 最低限セキュリティ
  - セキュリティ情報に注意し、こまめにアップデート



# 今回の一冊

- 技術評論社, 2010:  
Software Design  
2010年12月号,  
pp208, ISSN 0916-  
6297
  - 特集 Linux ブートシス  
テムの部分
  - わかりやすい文章
  - トラブル対処法もあり



# 参考 URL

- 神戸大理学部 ITPASS 実習 2020 「Linux インストールに必要な基礎知識」(白濱 理花)
  - [https://itpass.scitec.kobe-u.ac.jp/exp/fy2020/200928/lecture\\_linuxInstall/pub/](https://itpass.scitec.kobe-u.ac.jp/exp/fy2020/200928/lecture_linuxInstall/pub/)
- 北大理学院 情報実験第七回 INEX 2016 「OS インストール・起動」(三上 峻)
  - <http://www.ep.sci.hokudai.ac.jp/~inex/y2016/0617/lecture/pub/>
- Enterprize 「整理整頓を行なう縁の下の力持ち～ファイルシステム」
  - <https://enterprisezine.jp/iti/detail/272>
- Arch Linux Japan 「GUID Partition Table」
  - [https://wiki.archlinuxjp.org/index.php/GUID\\_Partition\\_Table](https://wiki.archlinuxjp.org/index.php/GUID_Partition_Table)

# 参考 URL

- PCトラノマキ「パーティション分割のメリットとやり方を分かりやすく解説します!!」
  - [https://torano-maki.net/guide/course/partition\\_division.html](https://torano-maki.net/guide/course/partition_division.html)
- パソコントラブル解決 PCと解「『プライマリパーティション』と『拡張パーティション』の違い」
  - [https://pctrouble.net/storage/partition\\_difference.html](https://pctrouble.net/storage/partition_difference.html)
- 「分かりそう」で「分からない」でも「分かった」気になれるIT用語辞典「スワップパーティション(swapパーティション)」
  - <https://wa3.i-3-i.info/word1723.html>
- I-O DATA 画面で見るマニュアル AVHD-U シリーズ
  - <https://www.iodata.jp/lib/manual/avhd-u/data/basic.htm>

# 参考文献

- 技術評論社, 2010: Software Design 2010 年 10 月号, pp208, ISSN 0916-6297
- 技術評論社, 2010: Software Design 2010 年 12 月号, pp208, ISSN 0916-6297
- 武藤健志, 2005: Debian GNU/Linux 徹底入門 第3版, ページ, 翔泳社, ISBN4-7981-0286-5
- 高町健一郎, 大津真, 佐藤竜一, 小林峰子, 安田幸弘, 2011: 「Linuxの教科書」改訂版, ページ, 毎日コミュニケーション, ISBN-13: 978-4839938710