第一回輪講
Linux

渡邊研究室 4年　早川顕太

本の紹介

- Linuxエンジニア養成読本

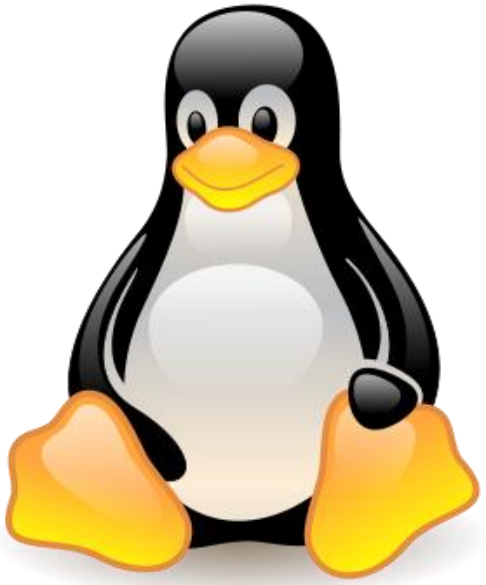
編集：SoftwareDesign編集部
出版社：技術評論社
発売日：2011年4月8日

Linuxとは？

- Linuxとは

1991年にリーナス・トーバルズ氏（当時、21歳）により作られたUNIX互換なカーネル

- 開発動機
  - Minixは教育用で機能が劣る
  - 商用UNIXは高価



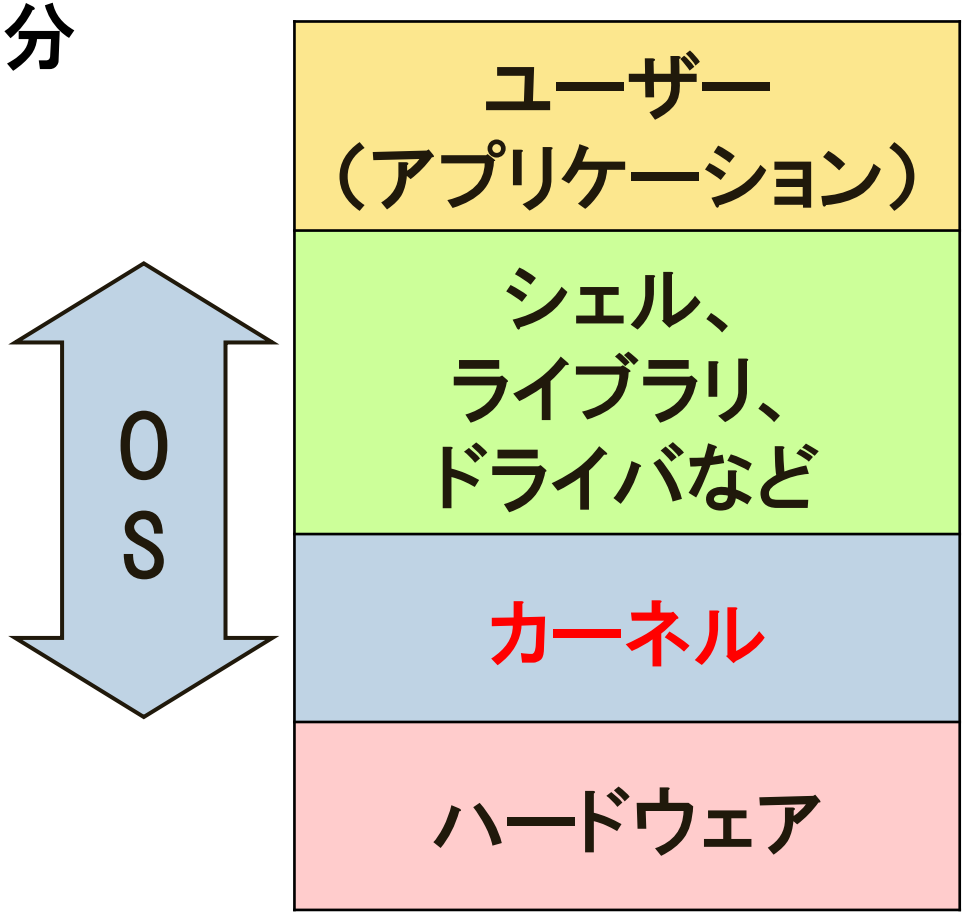
Linuxのマスコットキャラクター「TUX」
（この画像の著作権者Larry Ewing）

カーネルとは？

カーネル：OSの中核部分

カーネルの主機能

- プロセス管理
- メモリ管理
- 割り込み管理
- ファイルシステム
- ネットワーク
- etc



Linuxの特徴

- 豊富なディストリビューション
- 多くのアーキテクチャに対応
- POSIX準拠
- モノリシックカーネル
- モジュールカーネル
- 多くのファイルシスムに対応
- オープンソース

（特徴その1）
豊富なディストリビューション

- Linuxカーネルはソースコードで配布
  → **コンパイルが必要**
- アプリケーション、ライブラリ、ドライバなどは何もない
  → **ユーザーが用意**

一般ユーザーが導入するのは困難

- Linuxディストリビューションの登場
  LinuxOSとして必要なものをパッケージ化し、それを簡単に導入できるようにしたLinuxカーネルの配布形態

  現在、Linuxディストリビューションは300種以上

（Linuxの特徴その2）
多くのアーキテクチャに対応

Linuxカーネルはソースコードを

- アーキテクチャに依存する部分
- アーキテクチャに依存しない部分

に明確に分割

最新バージョン
Linux-3.8.7では、
28種ものアーキテクチャに対応

→これにより、移植が比較的に容易となり、多くのアーキテクチャをサポート

→インストールするシステムに合わせてコンパイル

※ただし、ディストリビューションでサポートしないのも多い

（特徴その3） POSIX準拠

- POSIX(Portable Operating System Interface)
  POSIXとはUNIXに関する標準的なAPIを定めた規格
  APIの例：open()、read()、write()、fork()、…etc
  (API:アプリケーションプログラミングインタフェース)

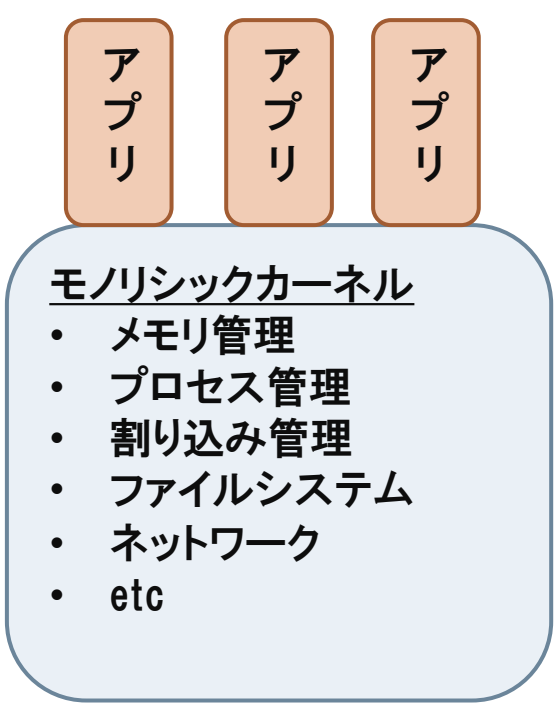
LinuxはUNIXから直接派生したOSではないが、POSIXに準拠しているため、UNIXとのソースレベル互換を実現

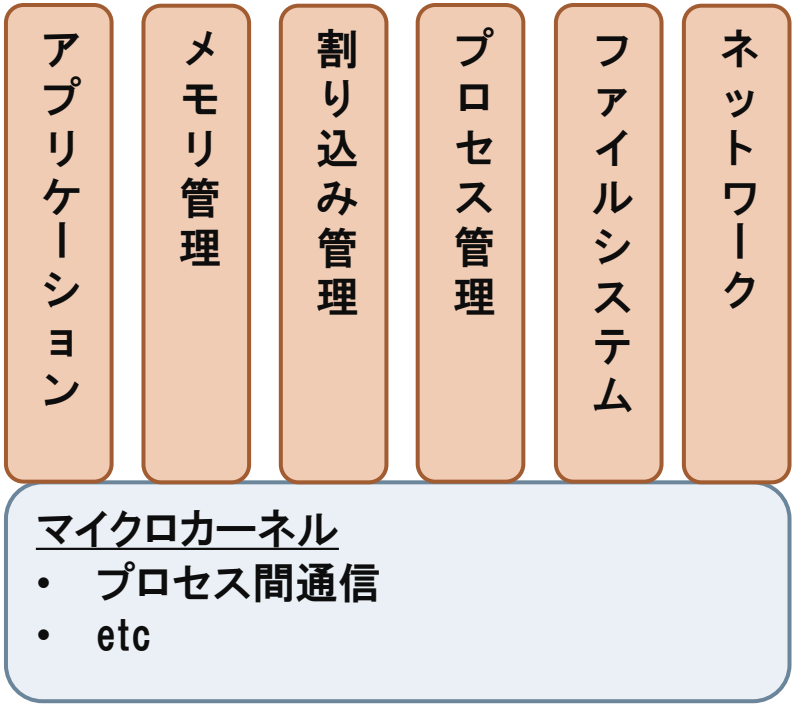
（特徴その4） モノリシックカーネル

カーネルの設計方針

- モノリシックカーネル：OSの機能をカーネルメモリ空間内に実装
- マイクロカーネル：OSの機能をアプリケーションとして実装
- ハイブリッドカーネル：モノリシック方式とマイクロ方式の混合方式

Linuxはモノリシックカーネル採用

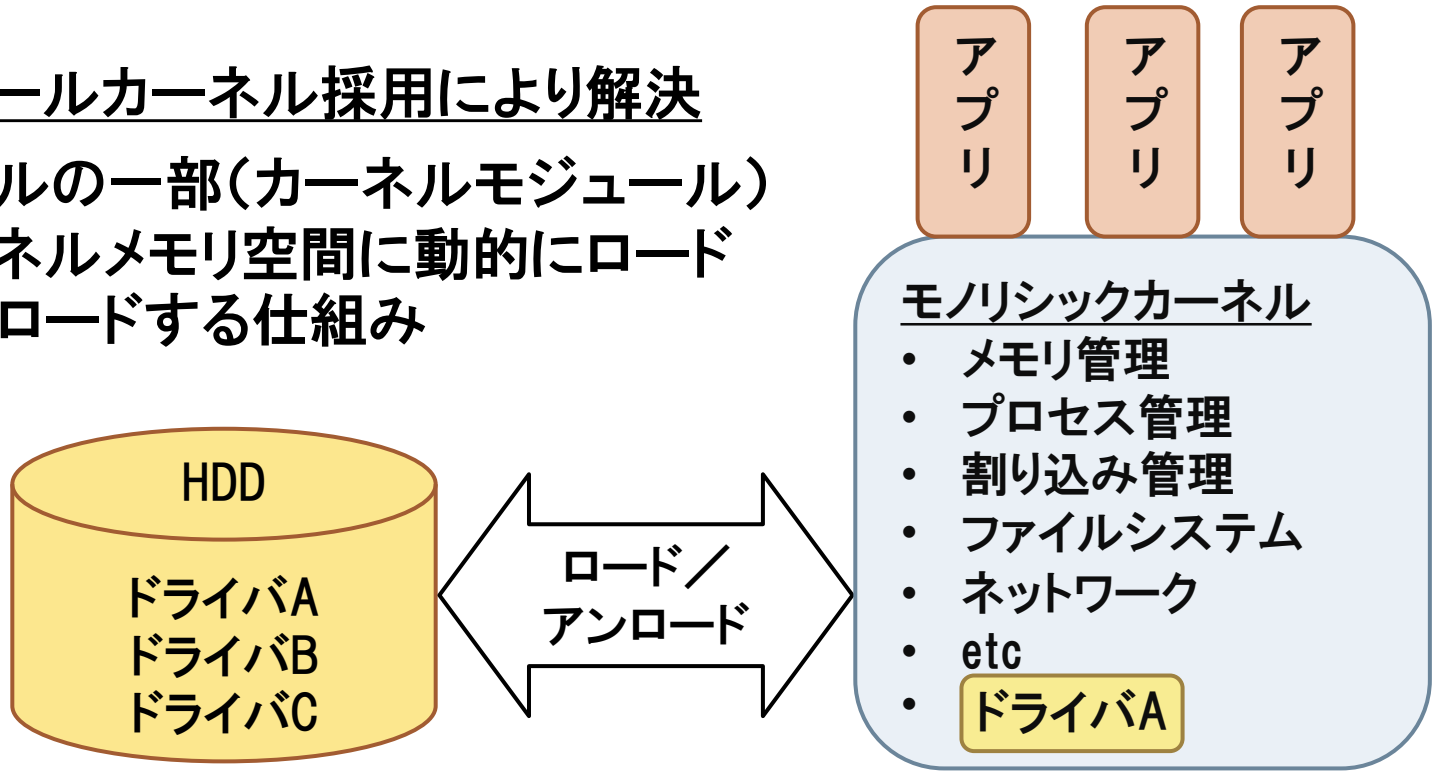




（特徴その5） モジュールカーネル

- モノリシックカーネルの欠点
  全ての機能を単一のカーネルメモリ空間に配置
  →カーネルの肥大化（メモリの消費）
  →機能追加でカーネルの再コンパイルが必要

- モジュールカーネル採用により解決
  カーネルの一部（カーネルモジュール）をカーネルメモリ空間に動的にロード／アンロードする仕組み

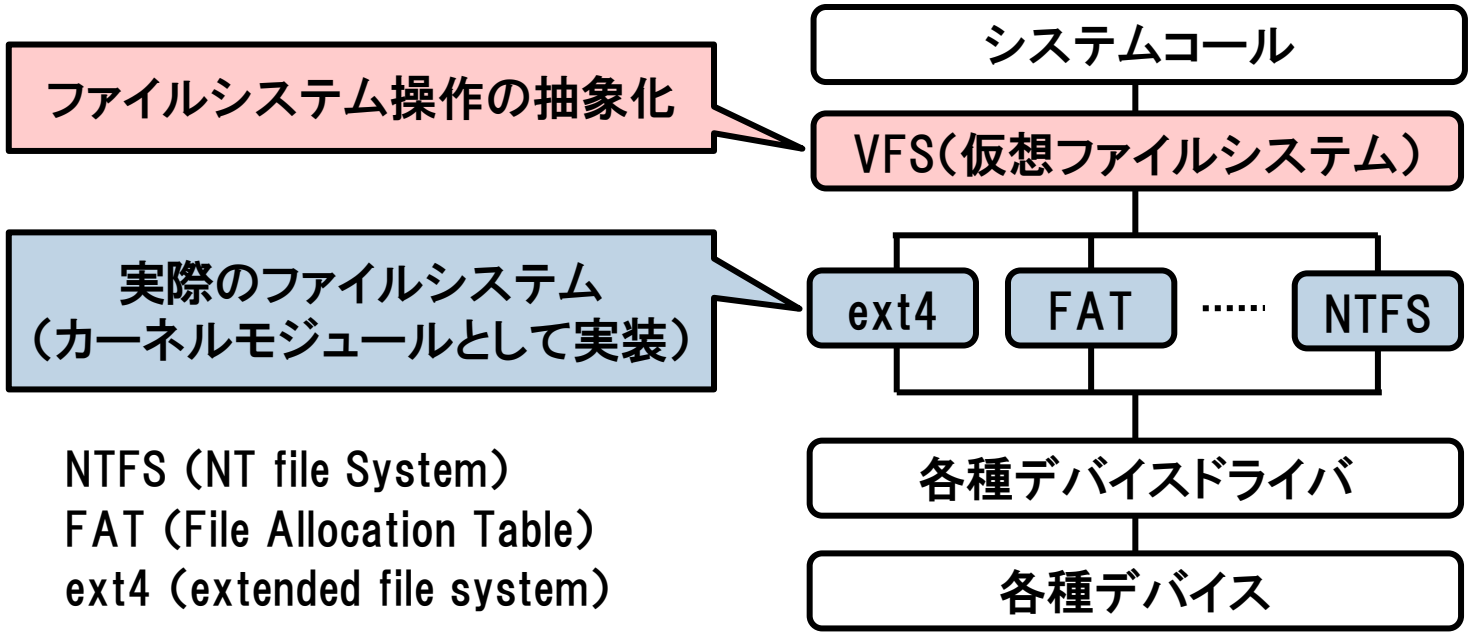


（Linuxの特徴その6）
多くのファイルシステムに対応

- Linuxのファイルシステムの構成

  Linuxは多くのファイルシステムが利用可能

  それら各々のファイルシステムの差異を吸収するため、抽象化層として仮想ファイルシステムがある



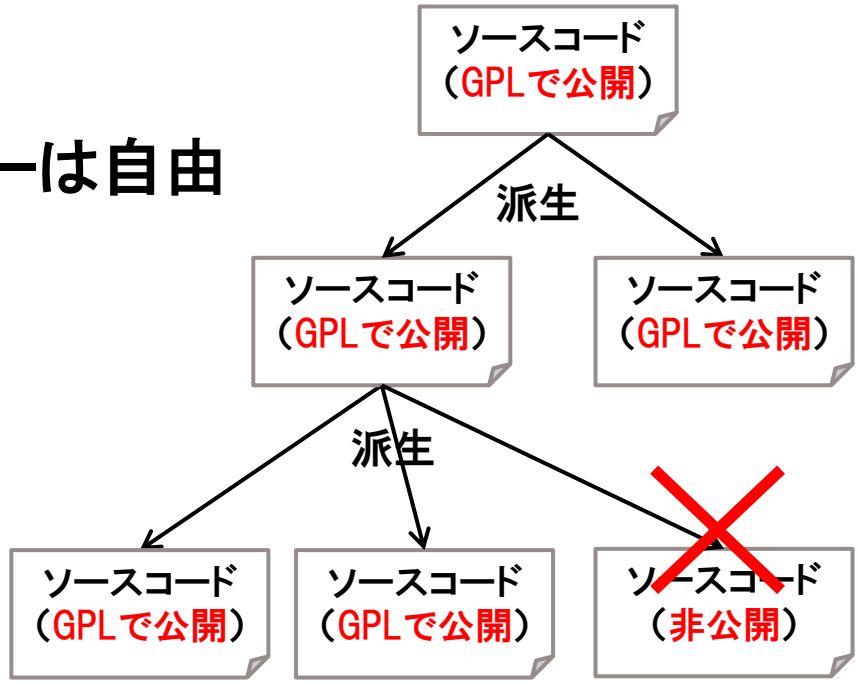
NTFS (NT file System)
FAT (File Allocation Table)
ext4 (extended file system)

（特徴その7） オープンソース

Linuxはオープンソースライセンスのひとつである
「GNU　General Public License (GNU GPL)」の下で配布

- 「GNU　GPL」の基本的な概念
  - 利用、改変、再配布、コピーは自由
  - 再配布の条件
    - ソースコードを公開
    - 同一ライセンス下で配布

※著作権フリーとは異なる



Linuxが使われる場面

- サーバ、スパコン

  2010年では、上位500のスパコンの内90%以上はLinuxを使用している

  理由： 無料、カスタマイズ可能、安定動作するため

- 組み込みシステム

  スマートフォンやタブレット端末、カーナビ、テレビなど

  理由： 多くのアーキテクチャに対応しているため

- 家庭用OS

  理由： ディストリビューションが豊富であり、導入が簡単でGUI(グラフィカルユーザインタフェース)を備えるものが多い

LinuxとWindowsの比較

	Linux	Windows
配布形態	オープンソース（無償）	クローズドソース（有償）
対応アーキテクチャ	豊富	x86、x64のみ
アプリケーションの互換性	ソースレベル互換（再コンパイルの必要あり）	バイナリレベル互換（再コンパイルの必要なし）
カーネルの設計方針	モノリシックカーネル	ハイブリッドカーネル

まとめ

- Linuxの概要
- Linuxの特徴
- Linuxが使われる場面
- Windwsとの比較

参考文献

- IT pro Linuxカーネルの基本機能
  http://itpro.nikkeibp.co.jp/article/COLUMN/20080501/300463/
- Wikipedia「Linux」
  http://ja.wikipedia.org/wiki/Linux
- eのらぼらとり　パソコン実習室　[ブートストラップ]
  http://park12.wakwak.com/~eslab/pcmemo/boot/index.html

17

付録

Linuxの仕組み

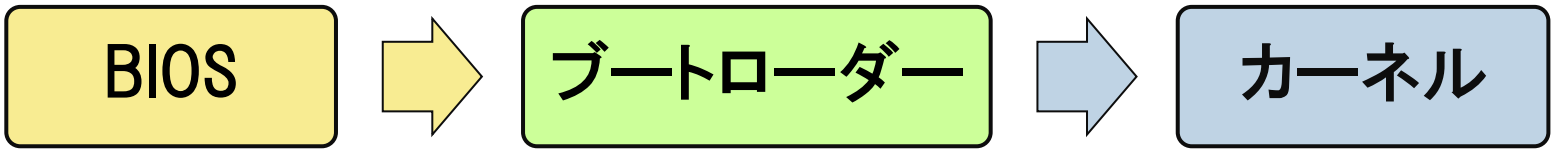
- ブートストラップ
- ファイルシステム

ブートストラップ

- ブートストラップとは？

  コンピュータシステムの電源を入れたときにOSを起動するまでの処理の流れのこと

  - ブートストラップの手順



- **問題点**

  プログラムをメモリにロードするにも、そのロードするプログラムがメモリ上にないといけない

- **解決策**

  不揮発性のROMを用意し、メモリ上にマッピング

Linuxのブートストラップ(その1)

- IntelアーキテクチャでのLinuxのブート手順

BIOS (Basic Input/Output System)

BIOS ROMがメモリ上にマッピングされており、そこからCPUはプログラムを実行。
① POST(Power On Self Test)と呼ばれるハードウェアの自己診断を行う。
② 起動ディスクの先頭1セクタ(MBR:マスターブートレコーダ)をロードし、実行。

↓

ブートローダ
GRUB(Grand Unified Bootloader)を使用した場合

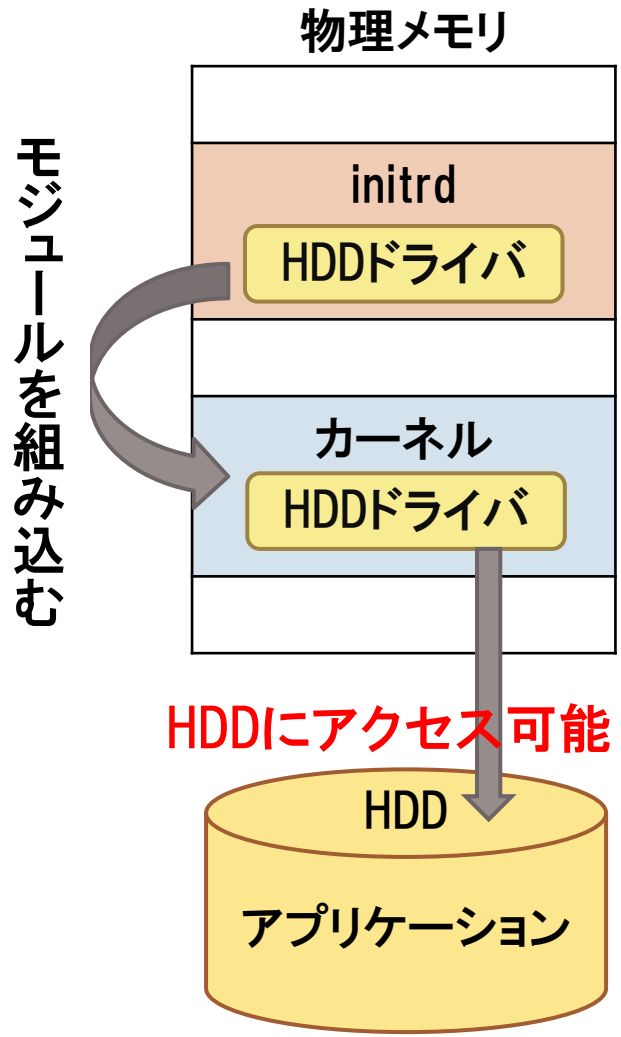
GRUBは3つのプログラムに分かれている。
① MBRにstage1が格納されている。stage1.5をロードし、実行。
② stage1.5はファイルシステムを認識する。ファイルシステム内のstage2をロードして実行。
③ stage2でカーネルとinitrdをロードし、カーネルコードを実行。

↓

Linuxのブートストラップ(その2)

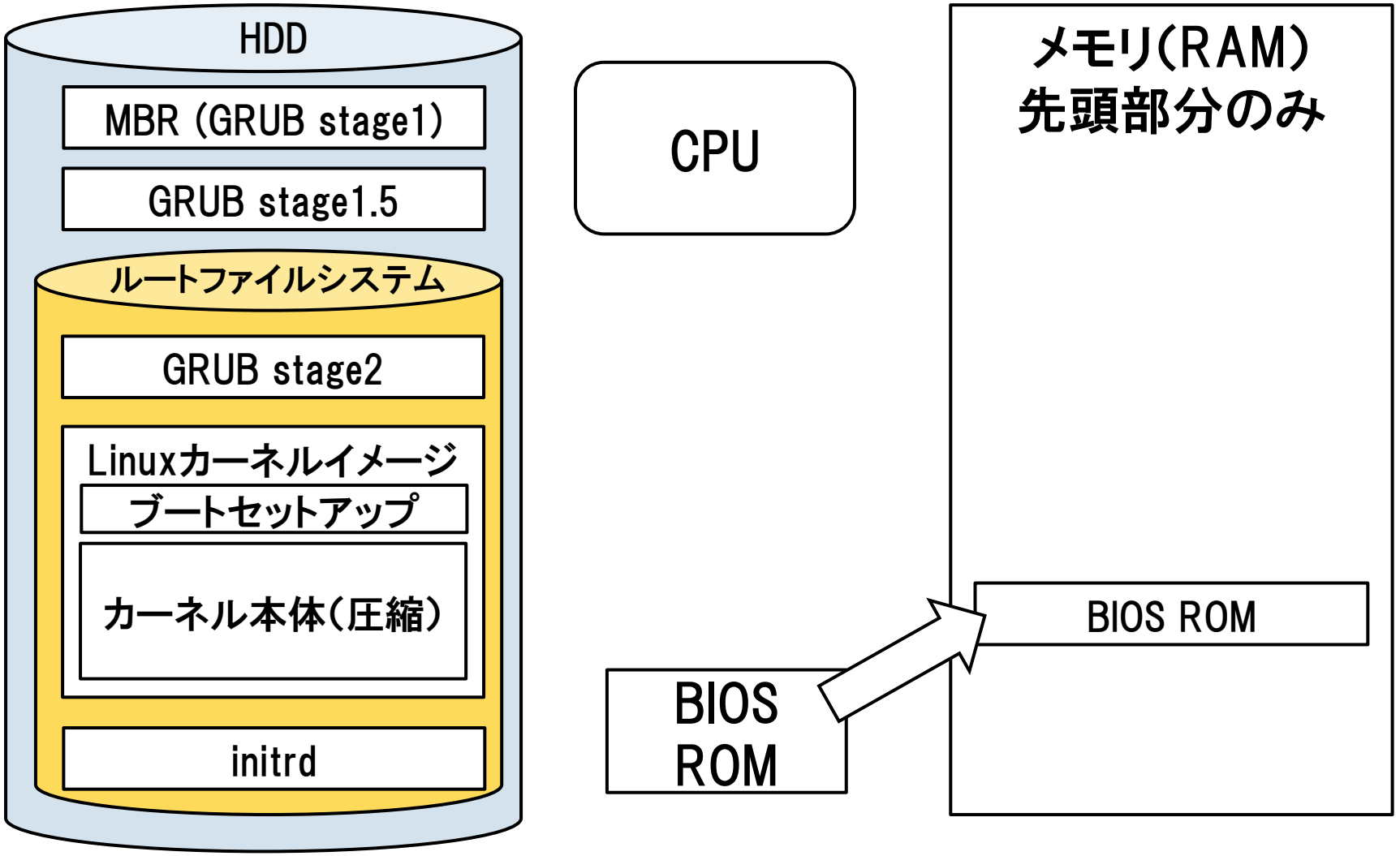
Linuxカーネル

① 周辺機器の初期化。
② 圧縮されたカーネル本体を展開し、実行。
③ カーネルの初期化。
④ 圧縮されたinitrdを展開し、ルートとしてマウント。
⑤ そこからHDDなどのモジュールを読み込む。
⑥ 本来のルートファイルシステムをルートとしてマウント。
⑦ 各種アプリケーションを起動。



- initrd(initial ramdisk)とは？
  カーネルの初期化用のために、メモリ上に構築されるファイルシステム

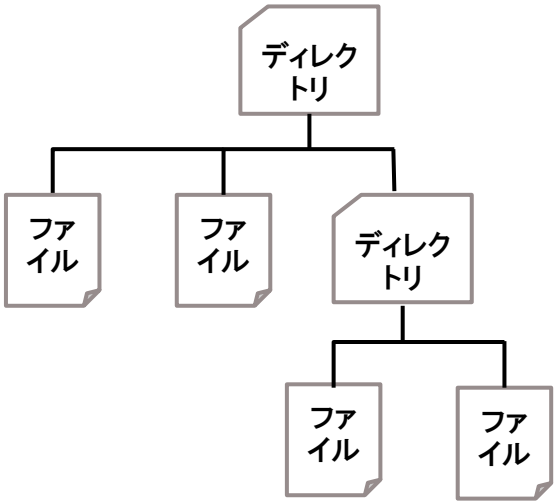
Linuxのブートストラップのアニメーション



ファイルシステム

- （ディスク）ファイルシステムとは？
  - 補助記憶媒体に構造を持たて管理
  - 以下の抽象的な「もの」を提供
    - ①　ファイル　　…　データの入れ物
    - ②　ディレクトリ　…　ファイルの入れ物
    - ③　ツリー構造　…　ディレクトリの階層構造

- ファイルシステムの例
  - NTFS(NT file System)
  - FAT(File Allocation Table)
  - ext4(extended file system)



ext2ファイルシステム(その1)

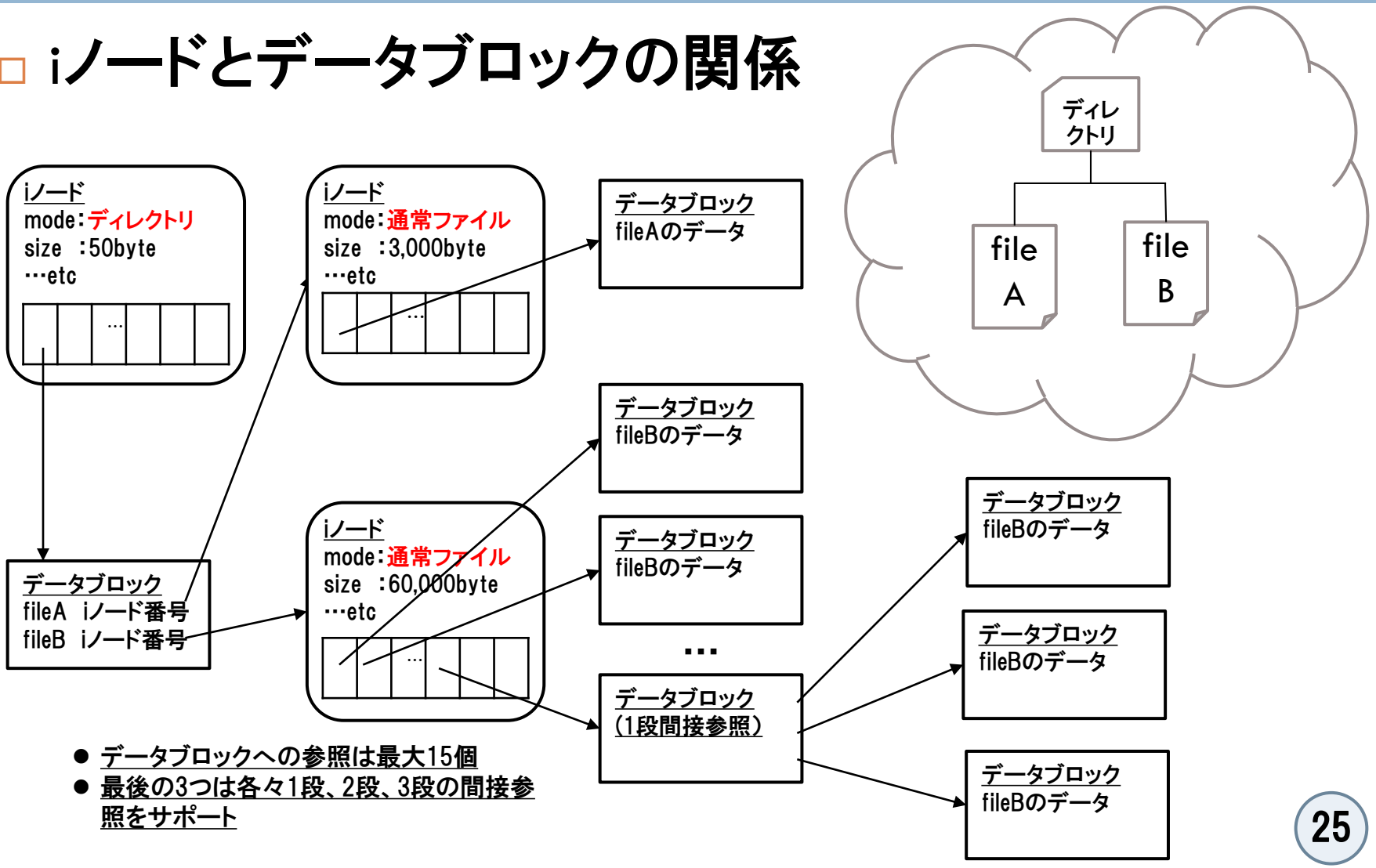
- ext2(second extended filesysytem)
  - Linuxで幅広く利用されるファイルシステム
  - 管理単位はセクタ(512byte)をいくつかまとめたブロック(多くは4Kbyte)
  - i-node方式で、各ファイルはiノードと一対一に対応
  - ブロックとiノードにはそれぞれ、通し番号が割り振られる
  - iノードは対応するファイルの管理情報を保持

    管理情報の例　:ファイルの種類、ファイルサイズ、ファイルデータを保持しているブロック(データブロックという)の番号など。
  - ディレクトリもファイルで定義

    ディレクトリのデータブロックには、そのディレクトリに属する各ファイルのファイル名とiノード番号の組が記されている。

ext2ファイルシステム(その2)

- iノードとデータブロックの関係



- データブロックへの参照は最大15個
- 最後の3つは各々1段、2段、3段の間接参照をサポート

ext2ファイルシステム(その3)

- ext2のパーティション内の構造
  - ブロックグループに分けて管理
  - 1つのファイルのデータブロックは同じブロックグループ内に格納

→断片化が減る

→シークタイムが減る