# オペレーティングシステム2006
# 第1回 概要

2006年10月12日

海谷 治彦

# 目次

- スケジュールとやり方
- 内容の概要
- OSの概要

# スケジュールとやり方

- 詳細はホームページ参照
- 適宜，演習(宿題)を出します
- テストはやりません．

# 内容の概要 (演習の都合で変わるかも)

1. OSの役割
2. プロセス･リソース管理
3. メモリ管理
4. デバイス管理
5. ファイル管理
6. ユーザー管理
7. ネットワーク
8. ユーザーインタフェース (CUI, GUI)
9. デーモン･サーバーサービス(httpd, smtp等)
10. OSセキュリティ

2〜5: OSの基本機能 (文献1 p.9)

# 基本的な方針

- 一般論(や太古のOSの話)はつまらんので，なるだけLinux/UNIXの具体例にあわせて話を進める．

# 参考文献・リンクについて

- 詳細はページのほうを参照.
- 授業スライドのほうでも適宜ふれるが別に買う必要はない.
- しかし, 買ってもいいかも.

- リンクについても適宜, 参考にしてください.

# OSの概要編

# OSの実体と役割

- OSも(アプリと同様) プログラムの一種である.
- 2つ役割 (2面性)
  - ユーザーが計算機に処理(計算)を依頼するための統一的な窓口となる.
  - 計算機に接続された機器を依頼された処理を遂行するために管理･運用する.

# ユーザーとOSの範疇

- OSにおけるユーザーとは二つの異なる意味を持つ.
  - まさにパソコンの前に座っている貴方自身.
  - OSの各種管理機能以外のプログラム (いわゆるアプリケーションプログラム)
- 上記に対応したOSの二種類の意味
  - GUIやファイル表示ツールも含めOS
    - ディストリビューションやデスクトップ等
  - POSIX等で定義された管理機能を実際に処理している核となるプログラムのみをOSとみなす. ⇒ カーネル
    - カーネルは同じだが, GUIや付属アプリが違う場合がある.
    - 授業ではこっちの意味でOSを取り扱う.

# kernel version

- Linux kernelに限らず多数のOSはある周期でバージョンアップをしている.
- バージョン番号を示すのがkernel version.
- Linux kernelの場合,
  - 最新公開版は 2.6?
  - 2.x の xの部分は偶数版のみ公開される奇数版は開発テストバージョンらしい.
  - uname –r コマンドで使ってるLinuxのversionがわかる.
  - バージョンが違うとプログラムも大きく違う場合がある.

# Linuxディストリビューション

- kernelだけでは，ユーザーが業務のためにOSを使うことができない．
  - コマンド(アプリ)，Window System，コンパイラ等はkernelには含まれない．
- ユーザーのためのGUIや，アプリ管理者のための道具などをセットにして配布しているのがディストリビューション．
- 主な系列は，
  - Slackware系
  - ReadHat系 Laser5やTurboはコレに属する
  - Debian系

  とあるらしい．

# 窓口を統一するとは？

- 異なるハードでも同じ手順で処理を依頼できる.
  - CPUの種類や数が異なっても, 同じ手順で作業できる.
- POSIX (Portable Operating System Interface for UNIX) とかに準拠したインタフェースを提供.
  - IEEEによって規定されたUNIX互換OSが最低限もっているべきインタフェース(関数の種類)の仕様.
  - 大雑把にいうと標準関数の仕様.
    - 同じ名前, 同じ引数の関数で, OSの機能を呼び出すことができるということ.

# 例

- ファイルにデータを読み書きする.
  - read関数
  - write関数
- 日時のデータを得る
  - time関数
  - gettimeofday関数 (これはPOSIXではない)

CPUやディスク，発信機等の種類が変わる度に使う関数(仕様)が変わったらプログラマの身がもたん!

# オンラインマニュアルを見てみる

- man 2 intro **システムコールの概要**
- man 2 read
- man 2 write
- man 2 time
- man 2 gettimeofday

**等**

# IEEE

- **電気電子工学関係の学会**
- **各種の標準なども規定**(IEEE1394**とか**)

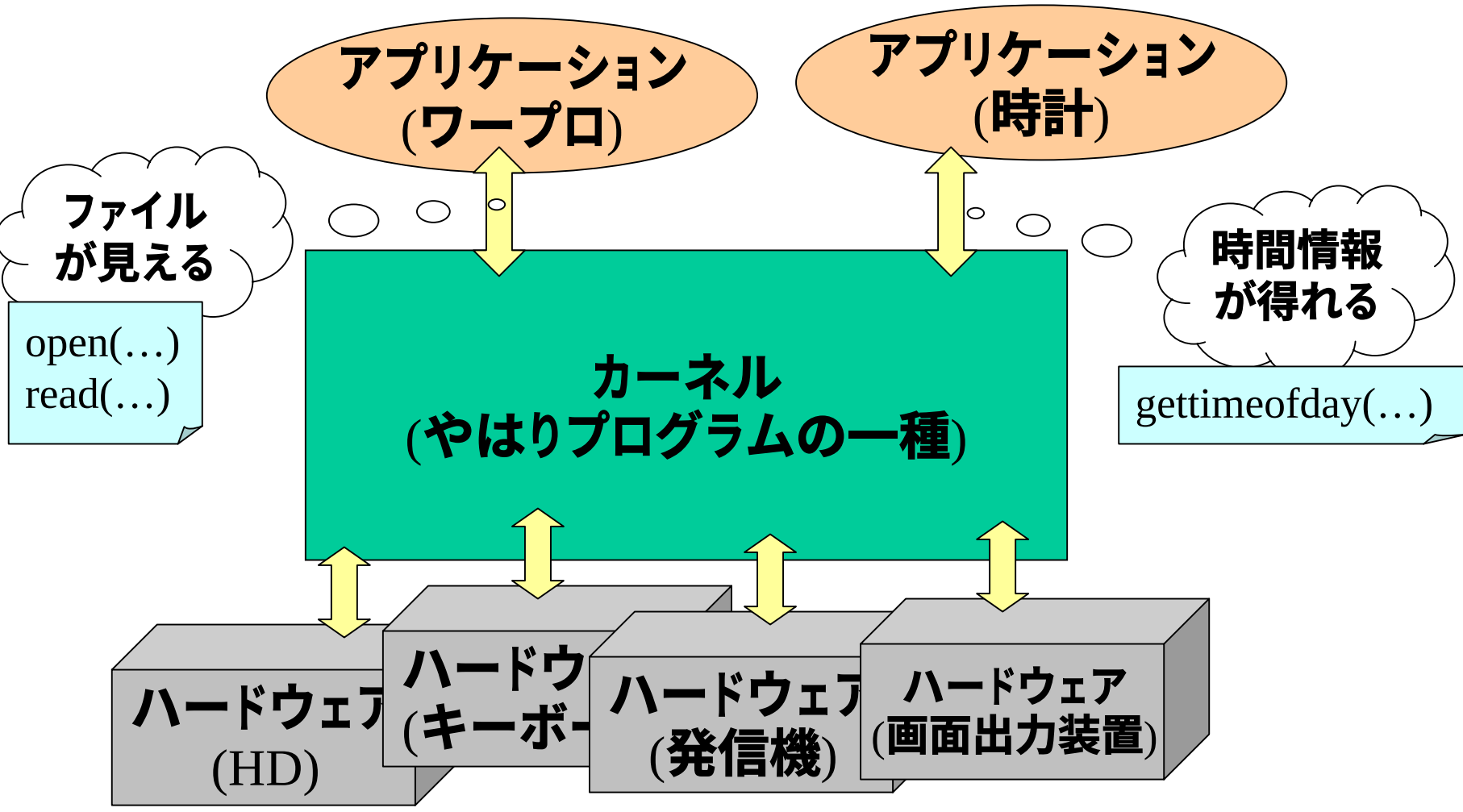
MACのファイヤーワイヤー
SONYのiLinkのこと

## About the IEEE

The IEEE (Eye-triple-E) is a non-profit, technical professional association of more than 380,000 individual members in 150 countries. The full name is the Institute of Electrical and Electronics Engineers, Inc., although the organization is most popularly known and referred to by the letters I-E-E-E.

- **上にも書いてあるが「あいとりぷるいー」と読む.**
- **詳細は** www.ieee.org **でもよんで.**

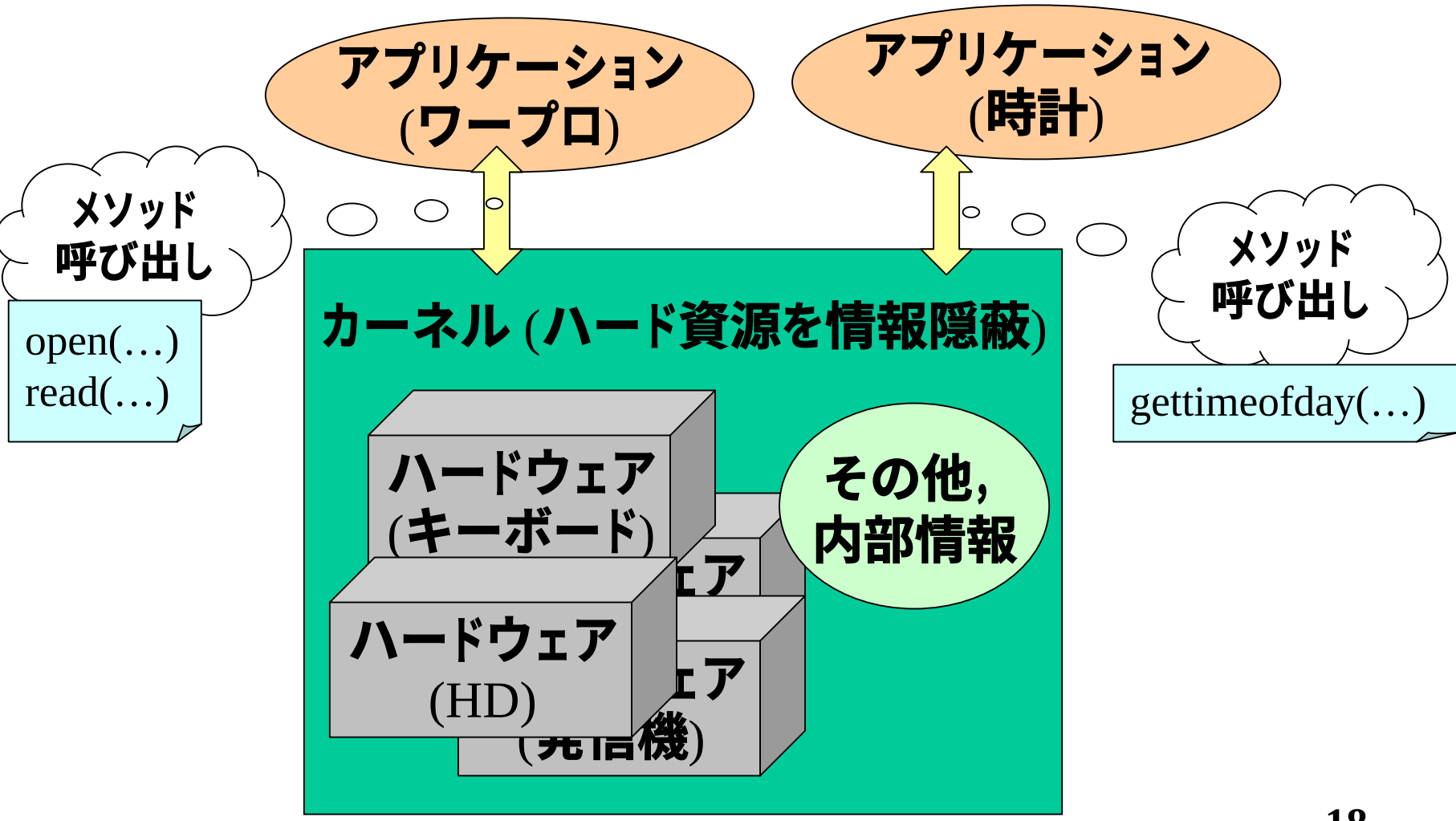
# カーネルの周辺


アプリケーション
(ワープロ)
アプリケーション
(時計)
ファイル
が見える
open(…)
read(…)
カーネル
(やはりプログラムの一種)
時間情報
が得れる
gettimeofday(…)
ハードウェア
(HD)
ハードウ
(キーボー
ハードウェア
(発信機)
ハードウェア
(画面出力装置)

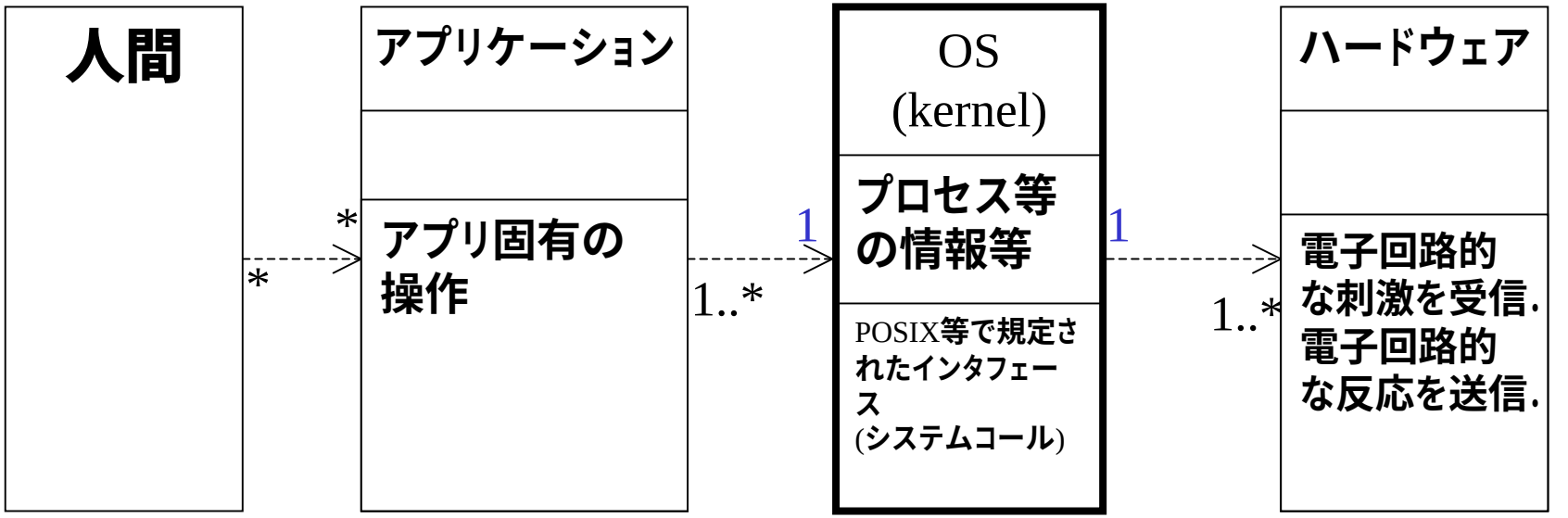
# カーネルはオブジェクト

- カーネルはオブジェクト指向におけるオブジェクトとみなすことができる. (文献1, p.14)
  - ハードウェア資源を内部データとして情報隠蔽している.
  - アプリは, システムコールというメソッドを通してのみ, それらの資源を活用できる.
  - ハード資源以外の内部情報も管理のために保持している.
  - 正確にはADT(抽象データ型)といったほうが良い.

# オブジェクトとしてのカーネル

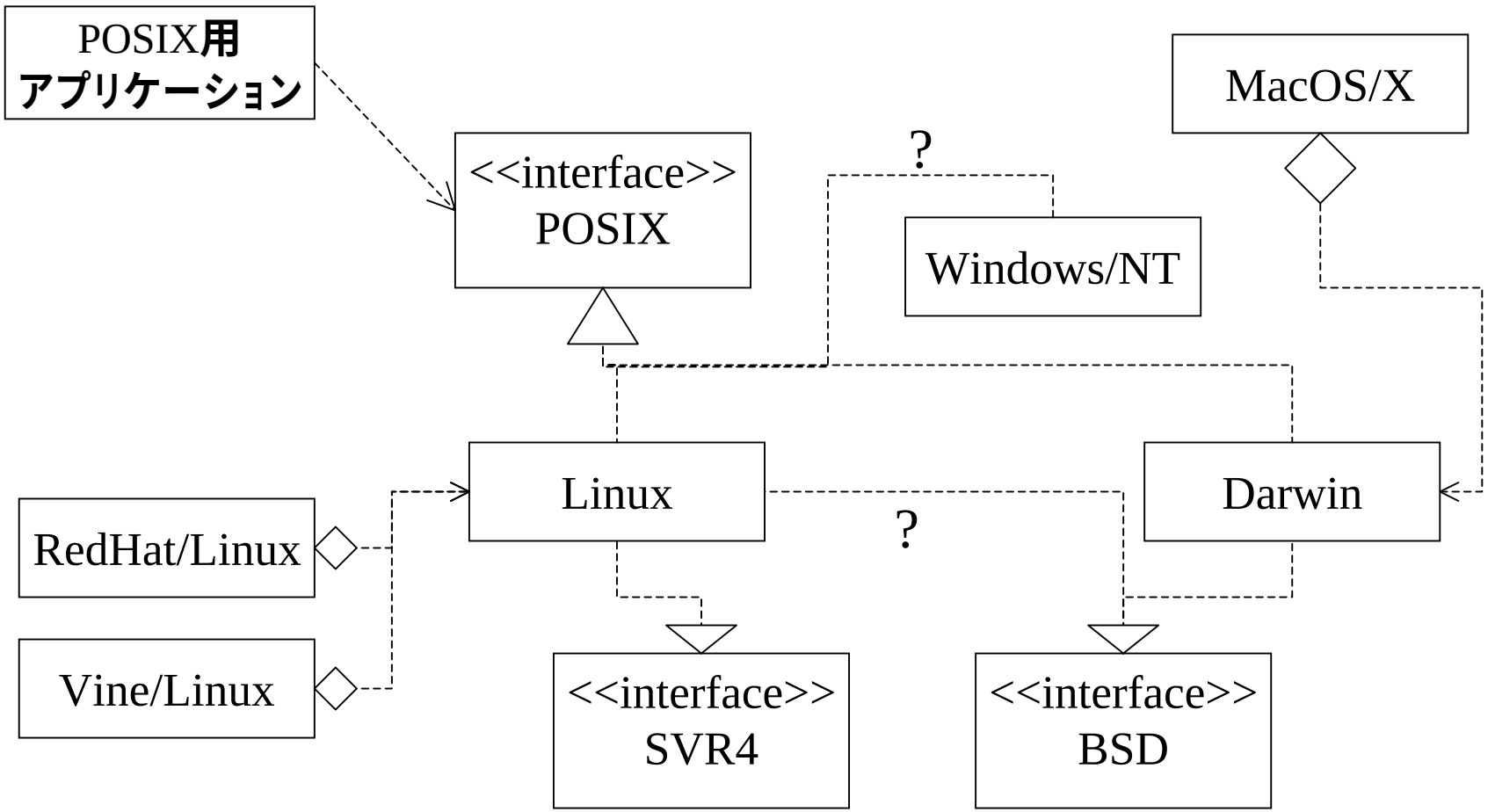


# クラス図風にかくと・・・




※ 青字の1は*の場合のほうが正確.
複数のOSを使うアプリや,
複数のOSに利用されるハードは存在しうる.

# POSIX等の関係




※ 実は実装関係はちょいと不正確かも．実データがよくわからんので．

# システムコール

- カーネル内の機能を利用するための関数群.
- これによって，アプリはプロセス，リソース，メモリ，デバイス，ファイル等，カーネルが管理しているものを利用することができる.

# カーネルの管理対象 1/2

- **プロセス: プログラムのインスタンス. すなわち, 個々の実行中プログラムのこと.**
  - 1**つのプログラムが複数のプロセスにインスタンス化されるのが普通.**
    - **例**: kterm **や** less, bash **など.**
- **システムリソース: プロセスが計算を進めるために必要な資源.** RAM, CPU**等**

# カーネル管理対象 2/2

- デバイス: いわゆる周辺機器. キーボード, マウス, モニタ等.
- ファイルシステム: いわゆるファイル. プログラムを含むデータを格納するために抽象化された記憶装置.

# カーネルの管理機能

- プロセス･リソース管理
- メモリ管理
- デバイス管理
- ファイル管理

‥‥これら順に今後話してゆきます

# Linuxのカーネル

- 基本的には一枚板(モノリシック)なプログラム.
  - プロセス, リソース, メモリ, ファイル管理が1つのプログラムになっている.
  - 複数のプログラムでカーネルを構成するOSもある. (例: Darwin)
- しかし, 以下の二つの部分は着脱容易となっている.
  - 個々のデバイス管理部 /proc/devices
  - モジュール /proc/modules

# 今後の展開

- 基本的には目次どおりに進める
- 時たまCでのプログラミング演習を要求する. (さすがにOSだとJavaではキツい. )
- 最終目標
  - アプリケーションの処理要求が, どのような仕組みで, 最終的にハードウェア等によって実行されるかを理解する.