Linux カーネル入門

VA Linux Systems ジャパン(株) OSDN 事業部
安井 卓 <tach@valinux.co.jp>

2002.10.9

VA LINUX SYSTEMS JAPAN

もくじ

・Linux カーネルとは
・カーネル主要機能概要
・そのほかのトピック
・参考文献・ポインタ

Copyright (C) 2002 VA Linux Systems Japan K.K., All Rights Reserved.

VA LINUX SYSTEMS JAPAN

Linux カーネルとは

VA LINUX SYSTEMS JAPAN

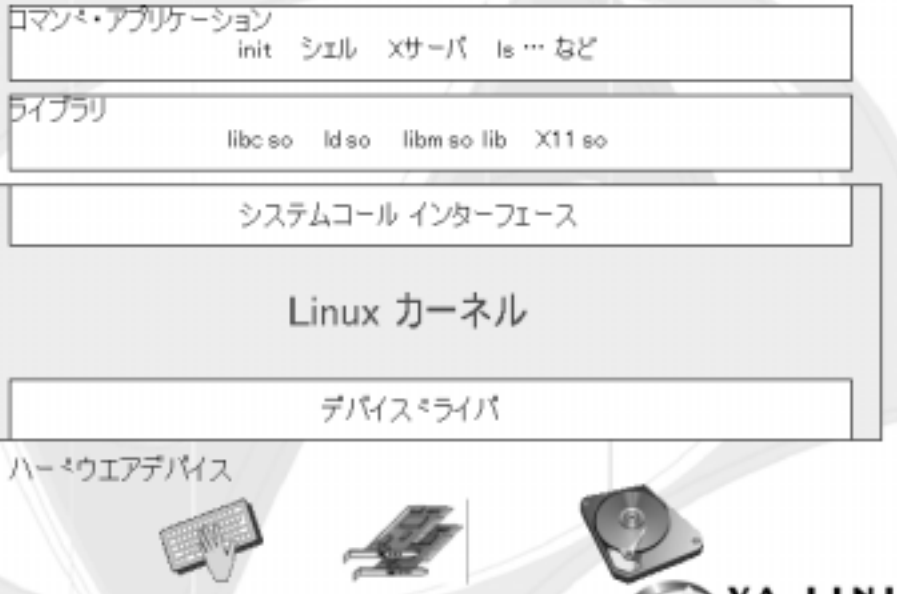
Linux システム

ディストリビューション

様々なツールやライブラリと一緒に「Linux」として配布しているが，本当に Linux といえるのはカーネルのみ

VA LINUX SYSTEMS JAPAN

Linux システムでのカーネルの役割



VA LINUX SYSTEMS JAPAN

カーネル(というソフト)の特徴

・ハードウエアの制御を行う
  ・デバイスドライバ
  ・割り込み処理

・アプリケーション(ユーザプログラム)に実行環境を提供する
  ・システムコール
  ・メモリの管理

・カーネルは自発的になにもしない
  ・システムコールや割り込みなど、イベント駆動型のソフトウエア

VA LINUX SYSTEMS JAPAN

Linux カーネルの特徴

・POSIX 準拠の OS である(UNIX 互換)

・モノリシックカーネルである

・カーネルモジュールの動的なロードができる

・オープンソース(GPL)である

・多くのアーキテクチャに対応している

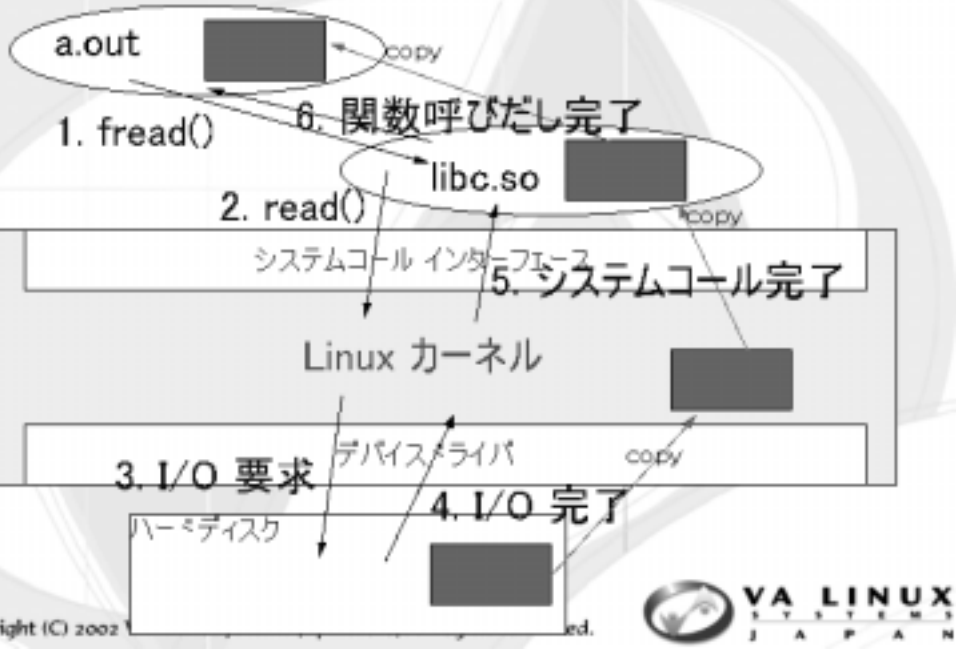
・開発速度が速い

VA LINUX SYSTEMS JAPAN

Linux カーネルの主要機能

VA LINUX SYSTEMS JAPAN

Linux カーネル処理例



VA LINUX SYSTEMS JAPAN

Linux カーネル 機能概要

VA LINUX SYSTEMS JAPAN

プログラムとプロセス

プロセス … プログラムが動いている状態

VA LINUX SYSTEMS JAPAN

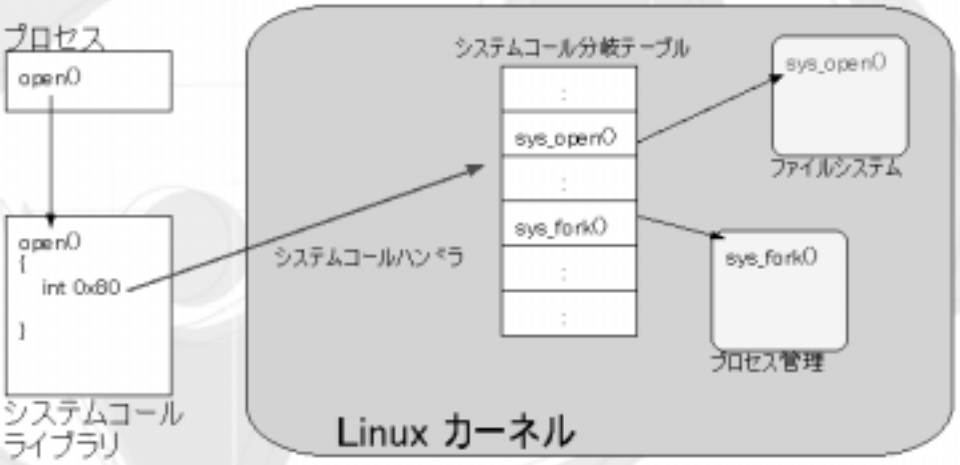
カーネルモード・ユーザモード

カーネルモード = 特権モード
・すべてのカーネルデータ・コードにアクセスできる
・プロセスはふだんはユーザモードで動く

VA LINUX SYSTEMS JAPAN

システムコール

・Linux カーネルに対するソフトウエアインターフェース
・カーネルに対するソフトウエア割り込み



VA LINUX SYSTEMS JAPAN

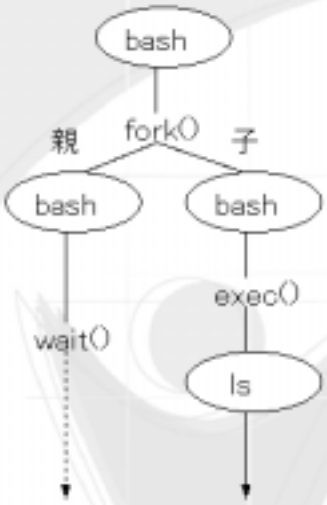
プロセスの情報

各プロセス毎にプロセス情報構造体におさめられている

・プロセスの識別子
・プロセス名
・プロセスの状態
・オープンしているファイルの情報（ファイルディスクリプタ）
・メモリ情報
・優先度（プライオリティ）
・コンテキスト保存領域
・シグナルハンドラ
…など

VA LINUX SYSTEMS JAPAN

プロセスの生成



1.fork()システムコールで自分自身(親プロセス)の複製を作成

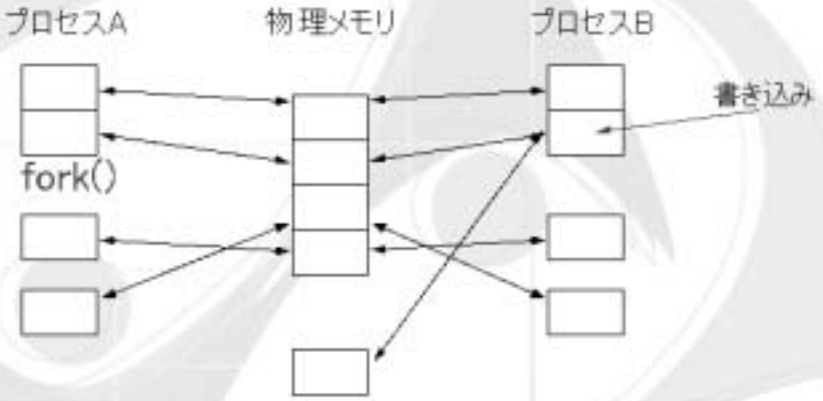
2.(子)exec()システムコールで、自分自身をlsコマンドで置き換え

3.(親)wait()システムコールで、子の終了を待つ

fork システムコール

- 自分自身の複製プロセスを生成するシステムコール
- プロセスを生成する唯一の手段
- 親プロセスの持つすべての資源も複製する
  - ファイルディスクリプタ
  - シグナルハンドラ
  - プロセス空間(コピーオンライト属性)
- プロセスIDはカーネルが適当に割り当てる

コピーオンライト



VA LINUX SYSTEMS JAPAN

exec システムコール

- 自分自身を新しいプロセスに置き換える
  - すべてのプロセス空間を解放し、新しいプロセス空間を設定
  - ファイルディスクリプタなどの管理情報は引き継ぐ

VA LINUX SYSTEMS JAPAN

exit システムコール

・プロセスの終了時に呼び出す
  ・すべてのプロセス空間を解放
  ・ファイルディスクリプタなどの管理情報は残す

VA LINUX SYSTEMS JAPAN

プロセス管理

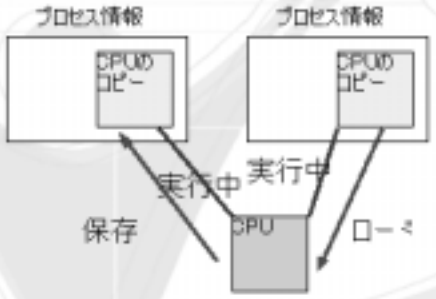
Linuxはマルチタスクosなので、(CPUがひとつでも)複数のプロセスが同時に動く。

…というより、動いているように見える。
実際は時分割で複数のプロセスを切り替えながら動く。
(プロセスディスパッチ)

VA LINUX SYSTEMS JAPAN

プロセスディスパッチ(切り替え)

・現在動作しているプロセスから、次に動作すべきプロセスに切り替える動作

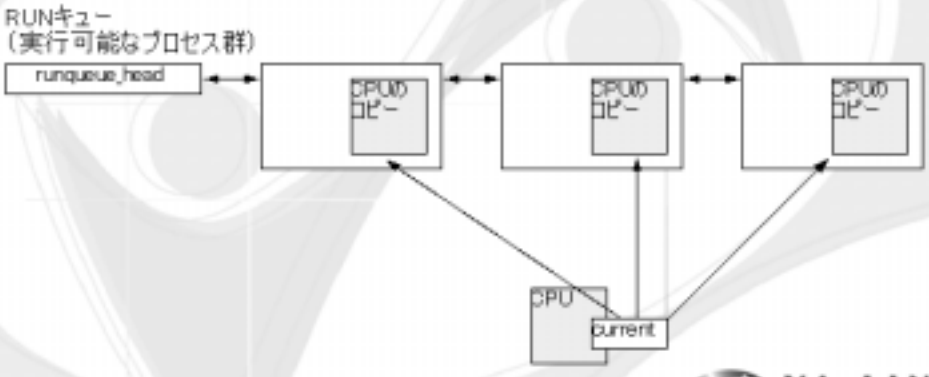


VA LINUX SYSTEMS JAPAN

プロセススケジューリング

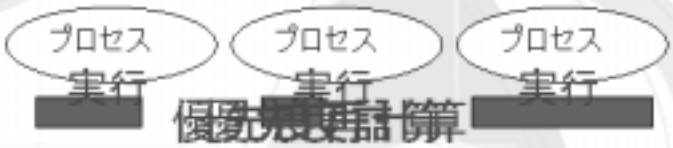
・実際にどのプロセスを動かすかの選択

・プロセススケジューラ



VA LINUX SYSTEMS JAPAN

プロセススケジューラ



プロセスの優先度を計算

各プロセスに対して、CPUの1回の持ち時間を与える
（優先度に応じて、というか優先度そのもの）

優先度が高いものから順番に実行

全プロセスの持ち時間がなくなったら優先度再計算
以下繰り返し…

VA LINUX SYSTEMS JAPAN

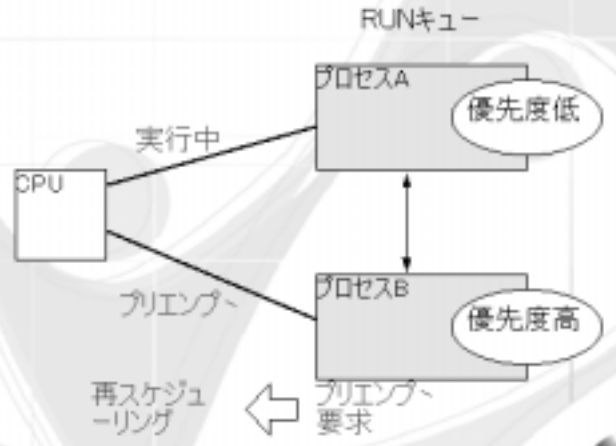
プロセスの待ち合わせ

ほとんどのプロセスは待ち状態にある

VA LINUX SYSTEMS JAPAN

プリエンプション

・あるプロセスが、カレントプロセスからCPU実行権を奪い取る処理

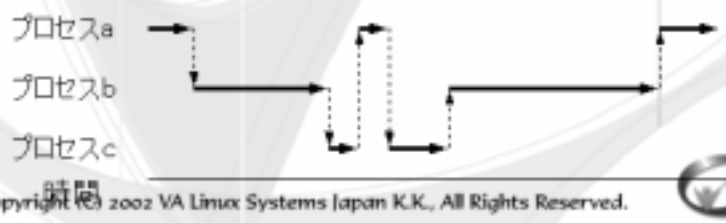


VA LINUX SYSTEMS JAPAN

プライオリティ(優先度)

・固定プライオリティ
  ・niceコマンドなどで与えられる優先度

・変動プライオリティ
  ・実行時間により動的に変更される
  ・実行時間が長いものほど優先度が下がる
  ・I/O処理などが多いプログラムは優先度高
    →応答性がよくなる
  ・優先度がそのままCPU時間に



VA LINUX SYSTEMS JAPAN

プロセスとスレッド

clone システムコールで生成
・実行単位はプロセスと同じ
・資源のコピーを行わずに共有する

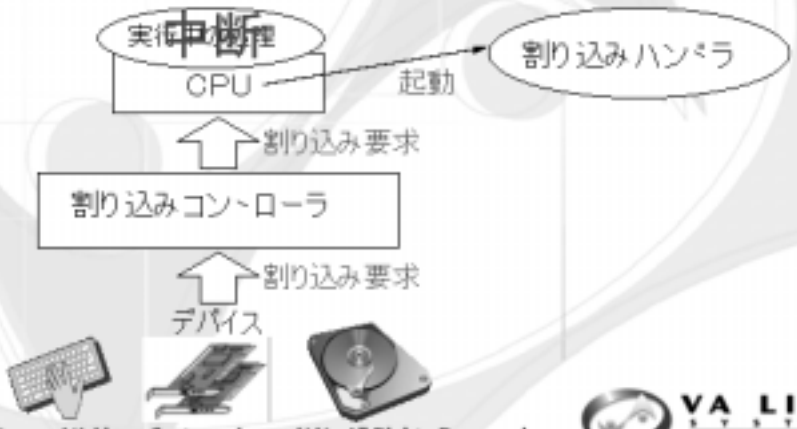
VA LINUX SYSTEMS JAPAN

プロセスの状態

VA LINUX SYSTEMS JAPAN

割り込み

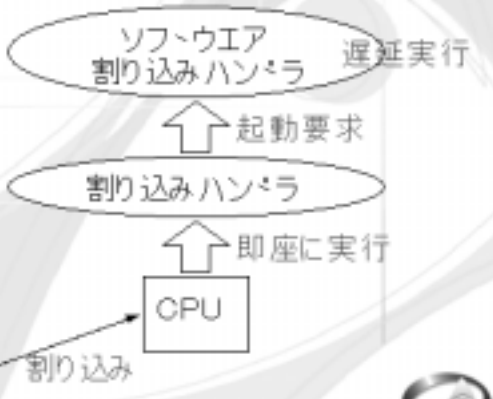
・ハードウエアで発生したイベントを拾ってデバイスドライバを起動するためのしくみ



VA LINUX SYSTEMS JAPAN

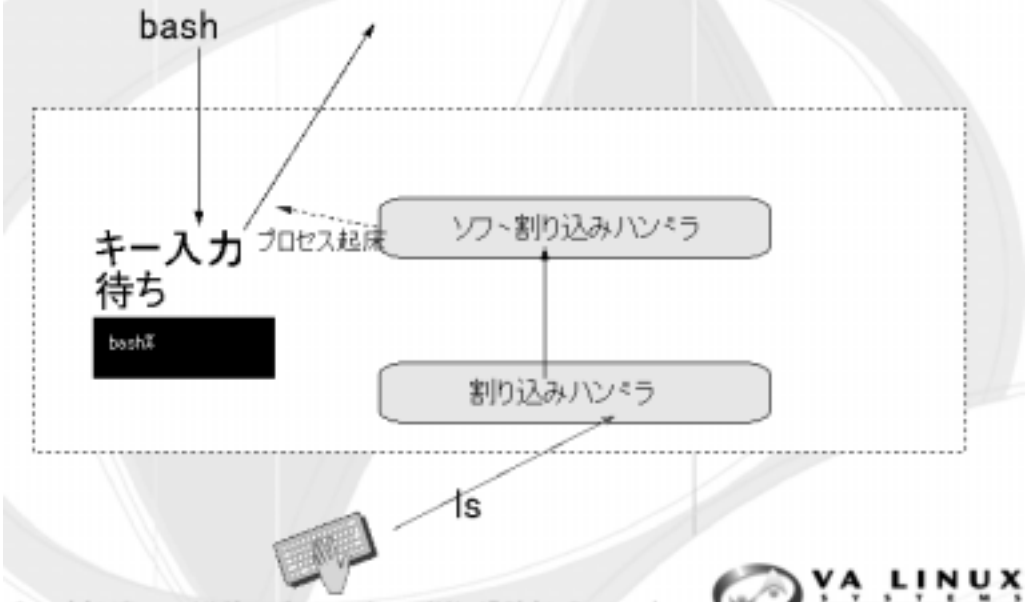
割り込みハンドラ

・2つのハンドラ
  ・割り込みハンドラ
  ・ソフトウエア割り込みハンドラ(BHハンドラ)



VA LINUX SYSTEMS JAPAN

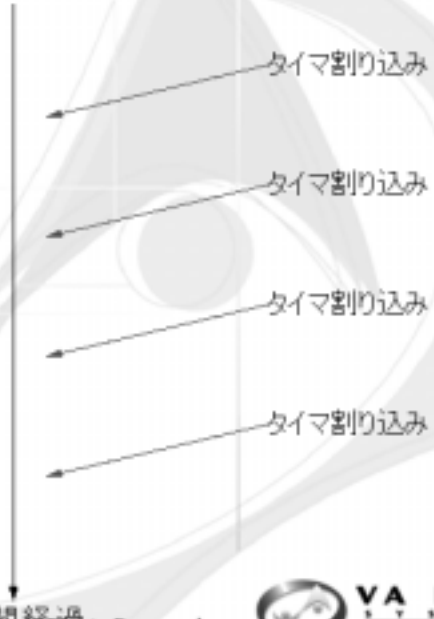
典型的なI/O割り込み処理例



VA LINUX SYSTEMS JAPAN

時間管理

- 時刻を正確に刻む
- 次元処理の実行
- 定期的なタイマ割り込み



VA LINUX SYSTEMS JAPAN

タイマ割り込みハンドラ

・タイマ割り込みハンドラ
  ・jiffies++
  ・2段目の時計（BHハンドラ）の起動要求

BHハンドラ
  ・カレンダ(GMT時刻)の更新
  ・ロードアベレージの計算
  ・タイマリスト（時限処理メカニズム）の実行

VA LINUX SYSTEMS JAPAN

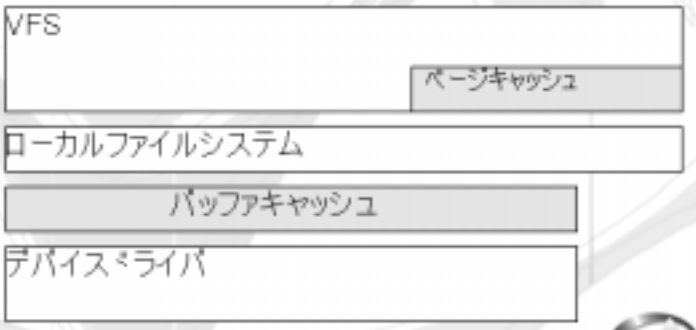
ファイルシステム

・論理的なファイル構造と物理的なディスクブロックの対応を管理
・内部で階層化されている

VA LINUX SYSTEMS JAPAN

ページキャッシュとバッファキャッシュ

ファイルアクセスを高速化するためのキャッシュ
・ページキャッシュ
  ・ファイルそのもののキャッシュ
・バッファキャッシュ
  ・ディスクブロックのキャッシュ



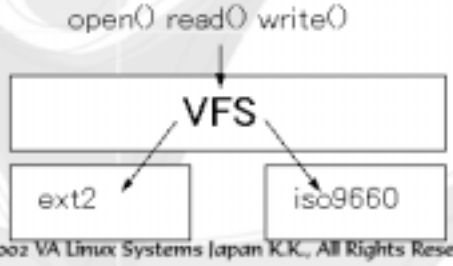
VA LINUX SYSTEMS JAPAN

仮想ファイルシステム(VFS)

・共通ファイルシステムレイヤ

・ファイルシステム間の違いを意識せずにファイルにアクセス可能

・ファイルが物理的にどう配置されているかは関知せず、論理構造とインターフェースを定義



VA LINUX SYSTEMS JAPAN

共通ファイルモデル

・VFS がサポートするファイルシステムモデル
　伝統的UNIXファイルシステムを反映している

・スーパーブロックオブジェクト
・iノードオブジェクト
・ファイルオブジェクト
・dエントリオブジェクト

ローカルファイルシステム

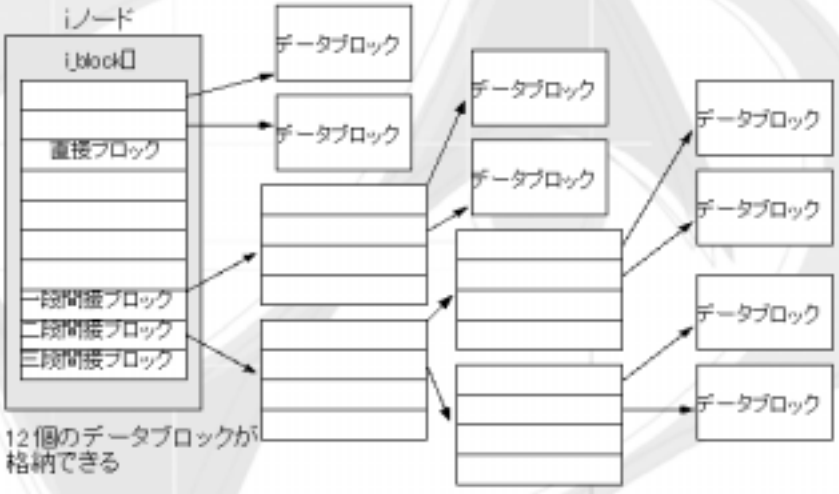
論理的なファイルと物理的なブロックとの対応関係を管理

・ext2 ファイルシステム
　・古典的UNIXファイルシステムとほとんど同じ

・ext3
・reiserfs
・xfs
・jfs

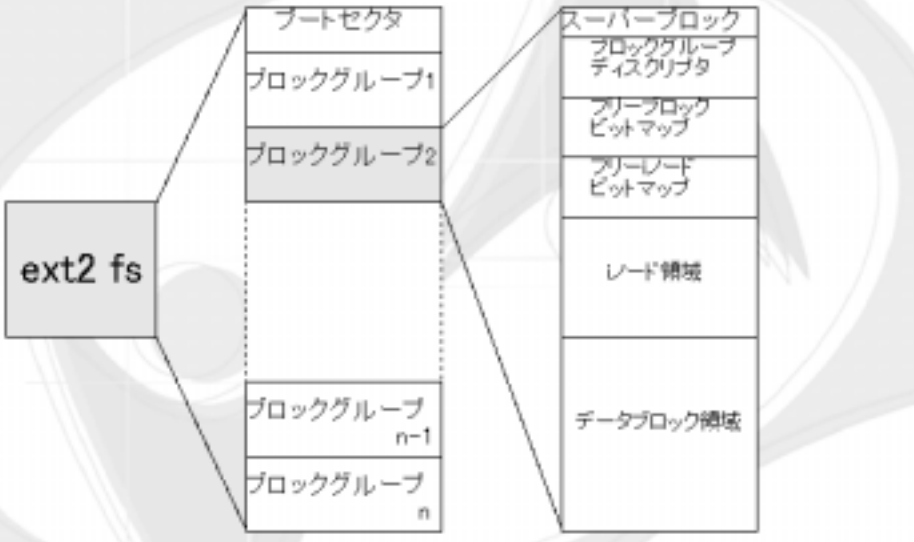
・vfat
・iso9660

iノードとデータブロック



VA LINUX SYSTEMS JAPAN

ブロックグループとスーパーブロック



VA LINUX SYSTEMS JAPAN

iノードが保持している情報

i_mode: ファイル種別(通常ファイル，ディレクトリ，シンボリックリンク，キャラクタ型デバイス，ブロック型デバイス，パイプ，ソケット)，ファイルモード

i_size: ファイルサイズ(バイト)

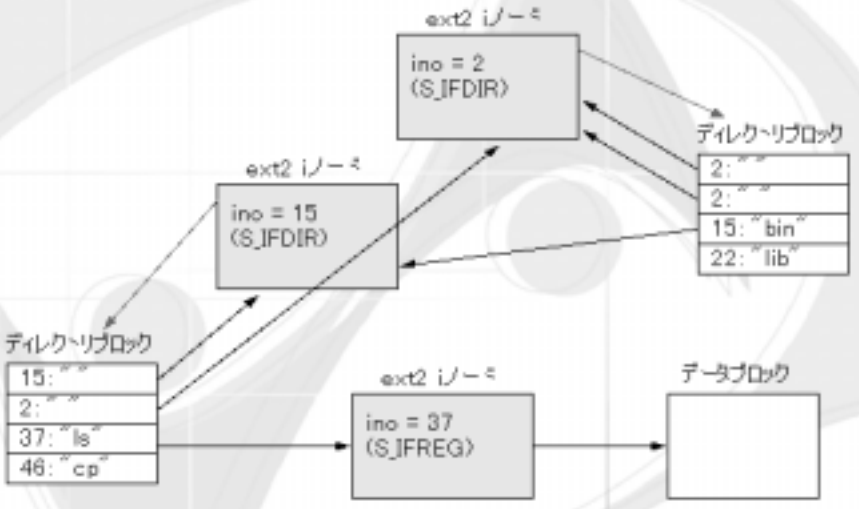
i_nlink: ファイルのリンク数(いくつのディレクトリから参照されているか，ハードリンクされたファイルは2以上になる)

i_uid, i_gid: 所有者情報(ユーザID，グループID)

i_atime, i_ctime, i_mtime: ファイルのタイムスタンプ(アクセス時刻，iノード変更時刻，更新時刻)

VA LINUX SYSTEMS JAPAN

ディレクトリ階層構造



VA LINUX SYSTEMS JAPAN

パス探索とキャッシュ

ファイルを操作するには，パス名をiノードに変換する必要がある．

パス探索を高速化するためのキャッシュ
・iノードキャッシュ
・ディレクトリエントリキャッシュ

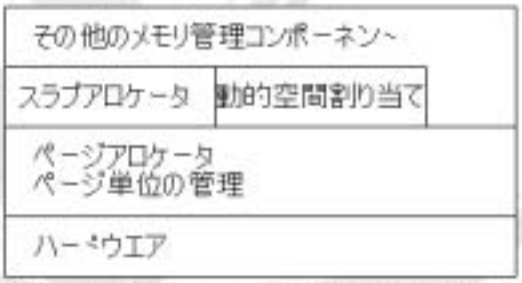
I/Oデバイスとファイル

・デバイスはデバイスファイルという形でVFSに用意される
  ・キャラクタデバイス
  ・ブロックデバイス

・メジャー番号・マイナー番号でデバイスを識別）
  ・デバイスファイル名は（カーネルにとっては）関係ない
  ・デバイスドライバ初期化時にメジャー番号との対応を登録

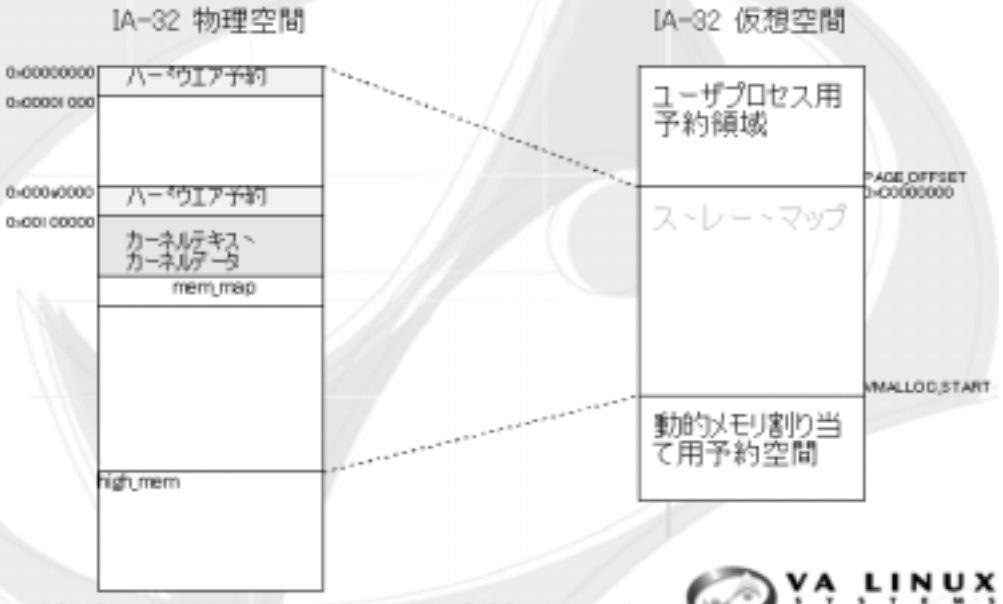
空間管理

- 仮想メモリ空間(VM)を利用する
- ページ単位で管理する(4KB)
- 物理的なメモリと仮想メモリとの対応(ページテーブル)を利用して論物変換を行う(CPU)
- 実際の物理メモリよりも大きなサイズのプログラム群を動かすこともできる



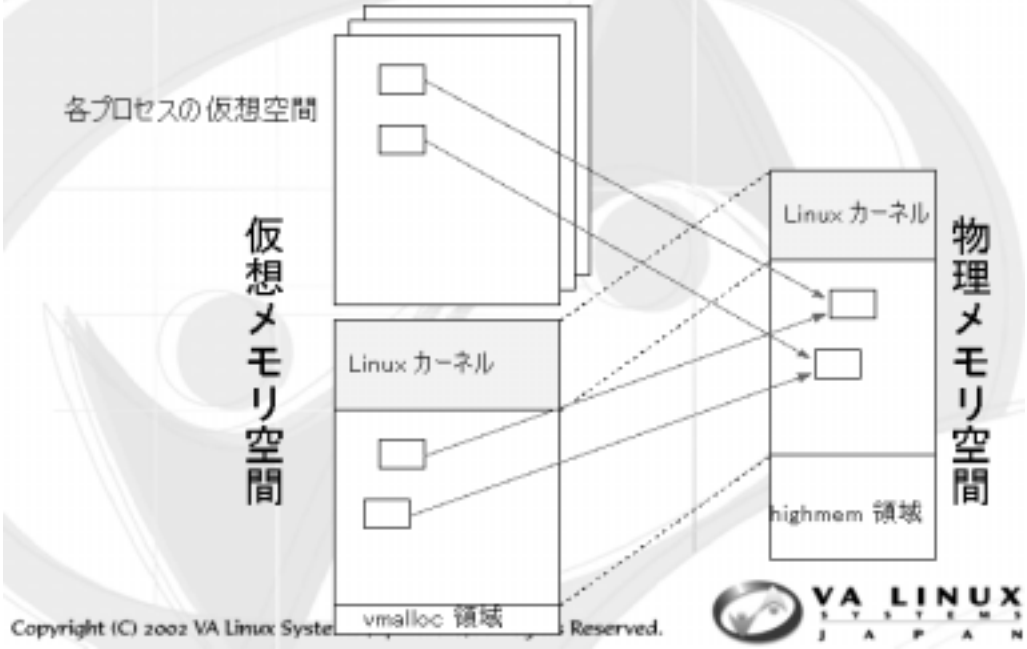
VA LINUX SYSTEMS JAPAN

物理空間と仮想空間



VA LINUX SYSTEMS JAPAN

プロセスメモリ空間



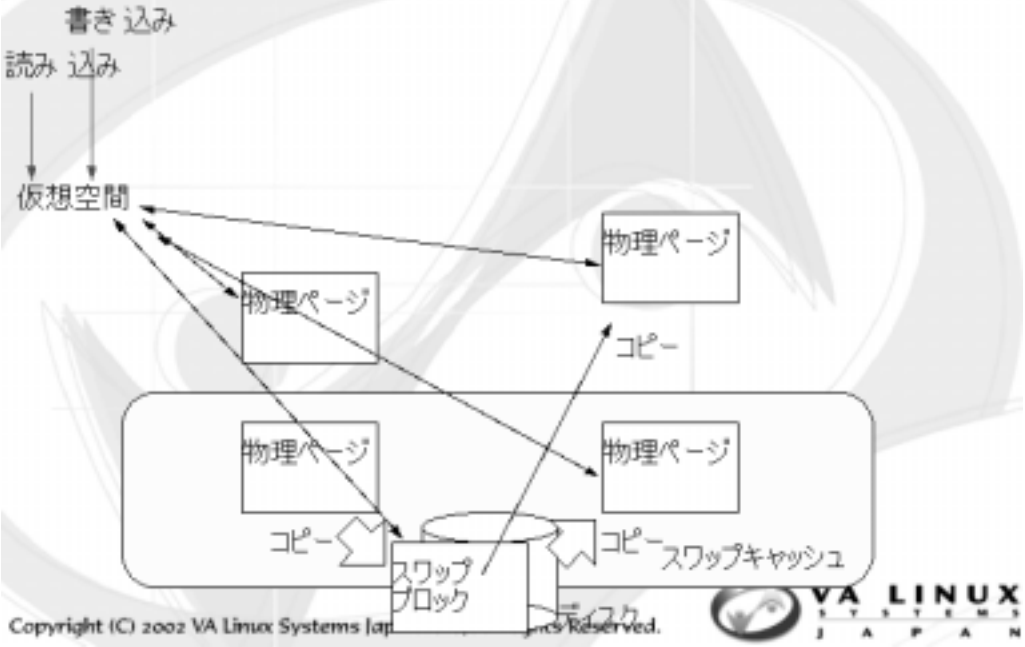
VA LINUX SYSTEMS JAPAN

割り当て(デマンドロード)

できるだけ遅らせる
- デマンドロード
- コピーオンライト(あとで説明)

VA LINUX SYSTEMS JAPAN

ページアウト・ページイン



VA LINUX SYSTEMS JAPAN

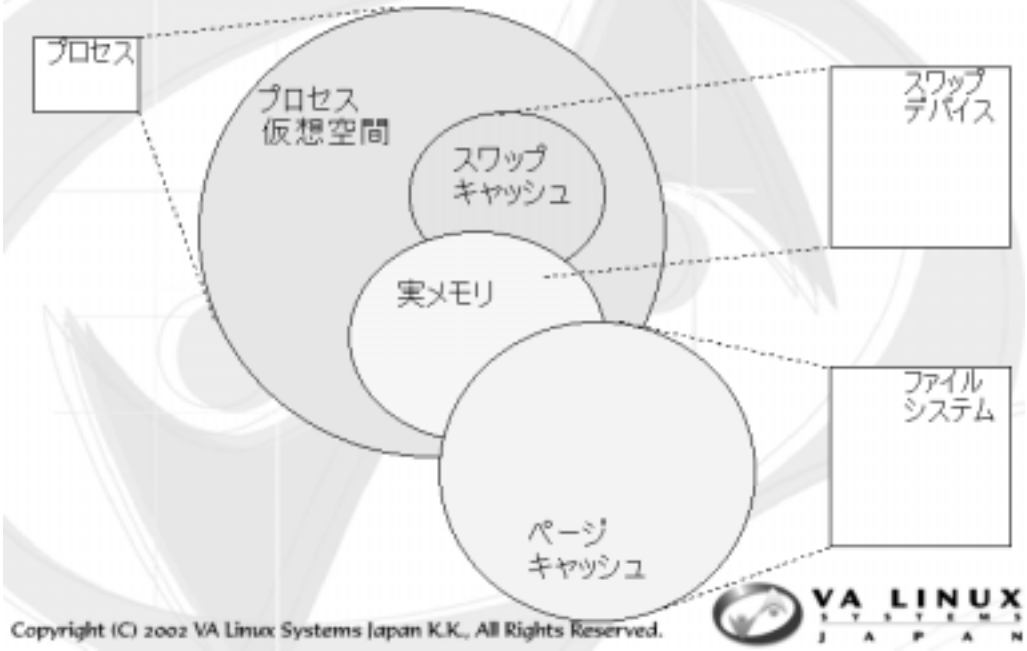
ページ解放について

・可能な限り空きメモリにせず，キャッシュとして利用
・利用価値の高いメモリは解放しない
・高負荷時には早めに多めに確保

カーネルスレッド kswapd による監視

VA LINUX SYSTEMS JAPAN

キャッシュとの関係



VA LINUX SYSTEMS JAPAN

ページアロケータ

・ページの管理

・ページごとの管理情報
  ・参照カウント
  ・仮想マップ先のアドレス
  ・アクティブか?
  ・・・

VA LINUX SYSTEMS JAPAN

バディシステム

・ページアロケータの管理方法
  ・無作為にページを割り当てると、メモリのフラグメンテーションが起きやすい
・2のべき乗個単位でページを管理する

VA LINUX SYSTEMS JAPAN

スラブアロケータ

・ページよりも小さい単位でのメモリ割り当て
・/proc/slabinfo 参照

VA LINUX SYSTEMS JAPAN

その他のトピック

VA LINUX SYSTEMS JAPAN

SMPカーネル

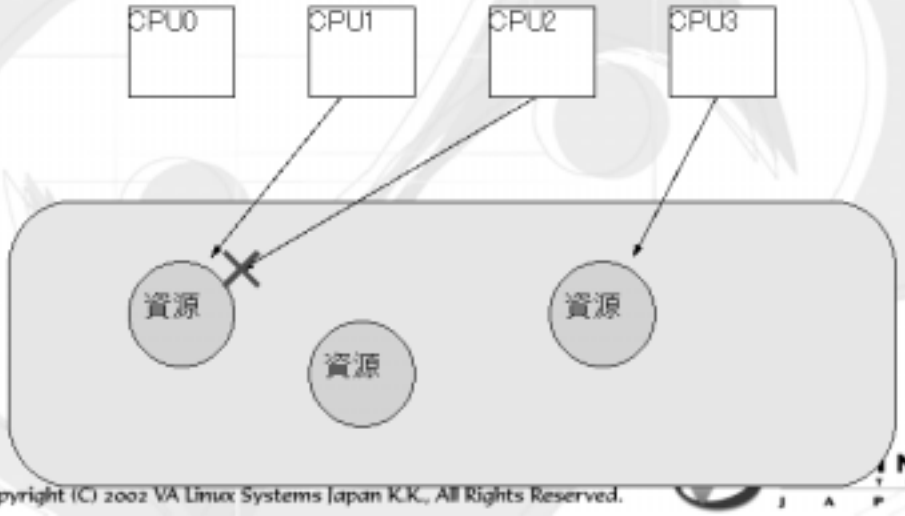
- SMP…Symmetric Multi Processing
  - すべてのCPUがメモリを共有
  - すべてのメモリに均一なアクセスが可能
  - メモリバスの速度が性能に影響する

リソースの競合状態を引き起こさないように
特別な配慮が必要

VA LINUX SYSTEMS JAPAN

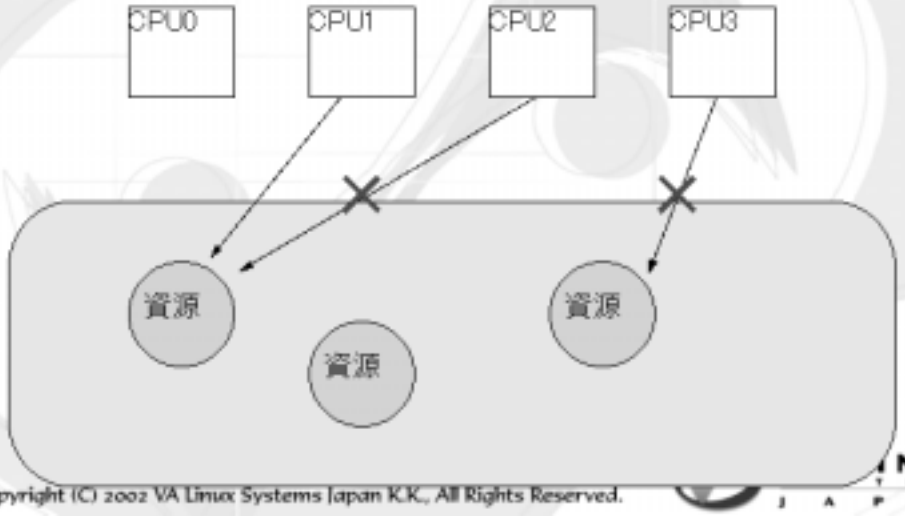
排他制御の仕組み 1

・スピンロック



排他制御の仕組み 2

・カーネルロック（ジャイアントロック）



モジュール

・動作中のLinuxカーネルに対して、新しい機能を追加するしくみ
  ・デバイスドライバ
  ・ファイルシステム

・自動組み込みが可能(kmod)

・カーネルの機能とユーザコマンドで利用する
  ・insmod rmmod modprobe など

モジュールの組み込み・削除

・組み込み
  ・モジュール配置領域の予約
    ◦vmallocで動的に領域確保
  ・未解決の参照の解決
  ・モジュールの組み込み
    ◦init_module()システムコール

・削除
  ・delete_module()システムコール
  ・依存されていなければ削除してvfreeでメモリを解放

ソースツリーについて

```
/usr/src/linux
    6MB Documentation (ドキュメント)
   26MB arch          (アーキテクチャ依存コード)
   73MB drivers       (ドライバ)
   10MB fs            (ファイルシステム)
   20MB includes      (インクルードファイル)
  0.4MB kernel        (カーネルコアコード)
  0.4MB mm            (メモリ管理)
    6MB net           (ネットワーク)
  0.4MB scripts       (スクリプトファイル)
```


参考文献・資料

参考書籍・記事

・詳解Linuxカーネル(オライリージャパン, 2001)
Daniel P. Bovet, Marco Cesati 著
高橋浩和, 早川仁 監訳

・Linux Japan連載 Linuxカーネル2.4の設計と実装
→LJ最終号にPDFで収録

VA LINUX SYSTEMS JAPAN

参考URLなど

・http://www.kernel.org/
Linuxカーネルの一次配布もと

・Linux Kernel Documents
カーネルソースツリーのDocumentsディレクトリ
JF(http://www.linux.or.jp/JF/)に日本語訳がある

VA LINUX SYSTEMS JAPAN