

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>

2002.5.29

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



もくじ

- Linux カーネルとは
- カーネル主要機能概要
- 実際のコマンド動作例
- 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.



Linux カーネルとは

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



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

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



Linux システムでのカーネルの役割

コマンド・アプリケーション

init シェル xサーバ ls ... など

ライブラリ

libc.so ld.so libm.so lib X11.so

システムコール インターフェース

Linux カーネル

デバイスドライバ

ハードウェアデバイス



VA LINUX
SYSTEMS
J A P A N

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

カーネル(というソフト)の特徴

- ・ハードウェアの制御を行う
 - ・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ェア

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



VA LINUX
SYSTEMS
J A P A N

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

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



Linux カーネルの主要機能

システムコール インターフェース

時間管理

プロセス管理
(スケジューラ)

空間管理

割り込み管理

ネットワーク

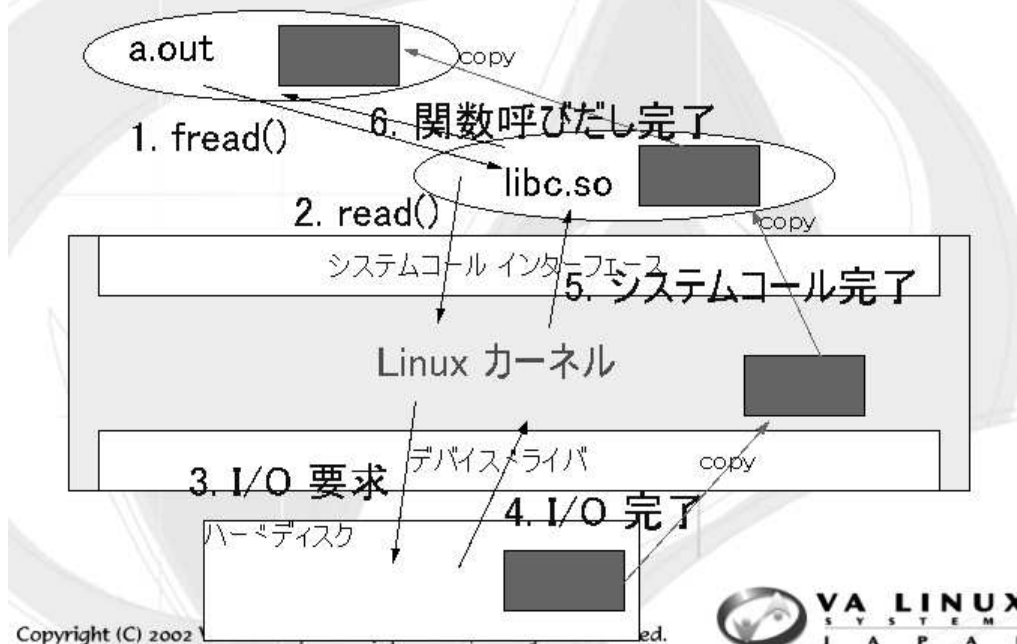
ファイルシステム

ブロックデバイスドライバ | キャラクタデバイスドライバ | ネットワークデバイスドライバ

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



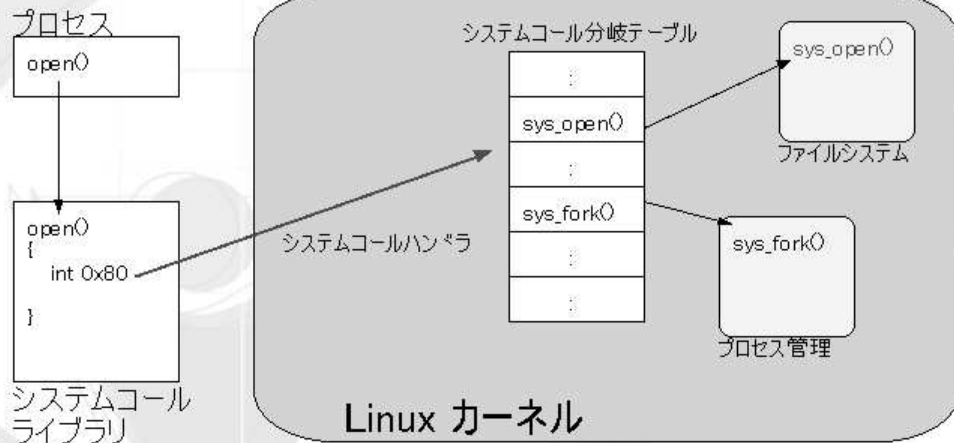
Linux カーネル処理例



Linux カーネル 機能概要

システムコール

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

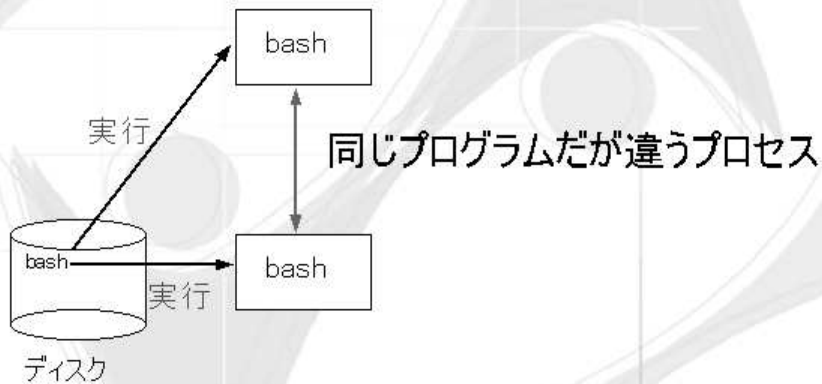


Copyright (C) 2002 VA Linux Systems Japan K.K., All Rights Reserved.



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



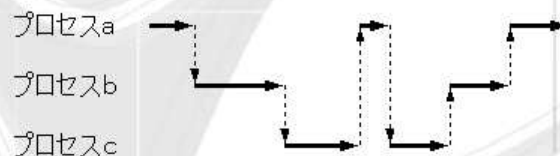
Copyright (C) 2002 VA Linux Systems Japan K.K., All Rights Reserved.



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)



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



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

プロセス情報

プロセス情報

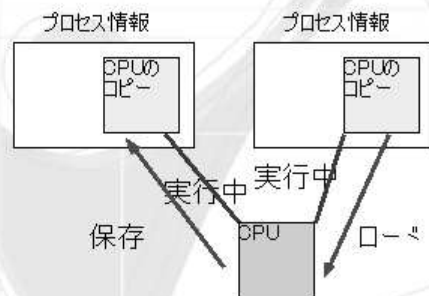
プロセス情報

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



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

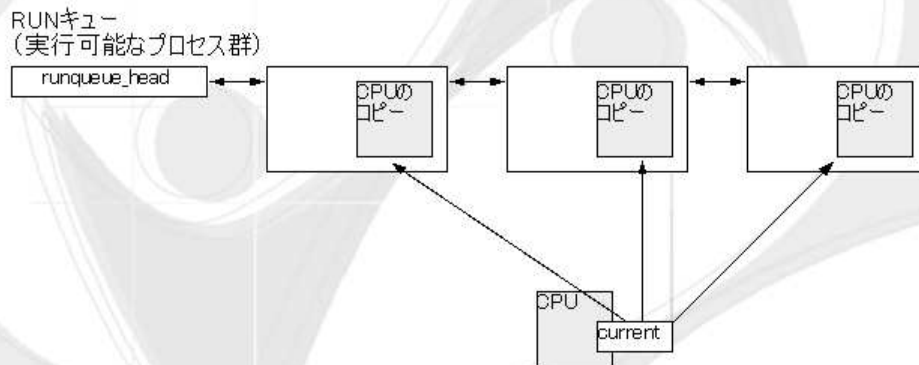


Copyright (C) 2002 VA Linux Systems Japan K.K., All Rights Reserved.



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



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



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

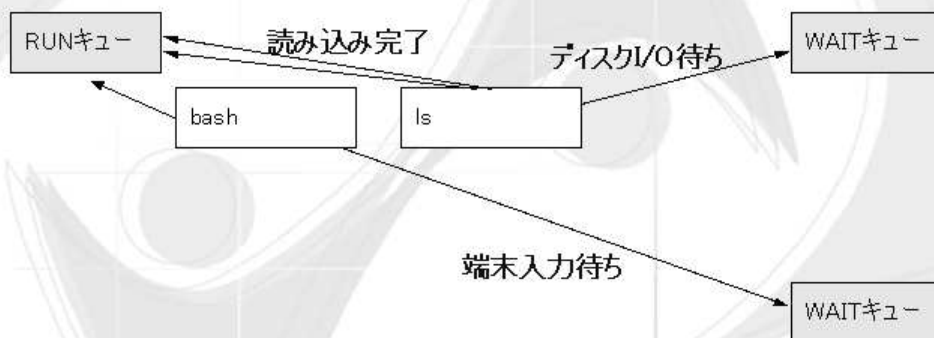
全プロセスの持ち時間がなくなったら優先度再計算
以下繰り返し…

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



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

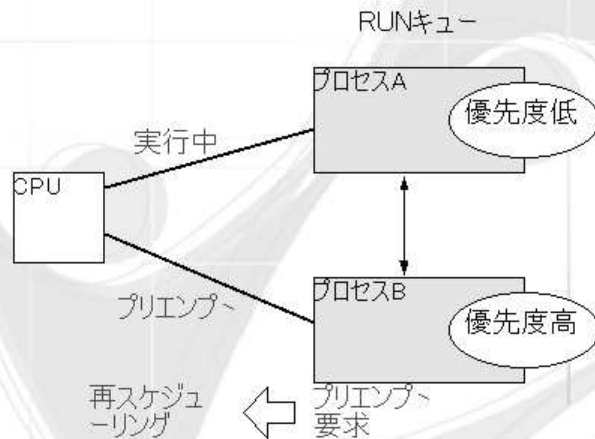


Copyright (C) 2002 VA Linux Systems Japan K.K., All Rights Reserved.



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

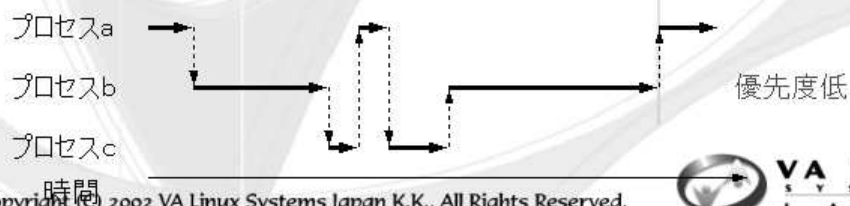


Copyright (C) 2002 VA Linux Systems Japan K.K., All Rights Reserved.



プライオリティ(優先度)

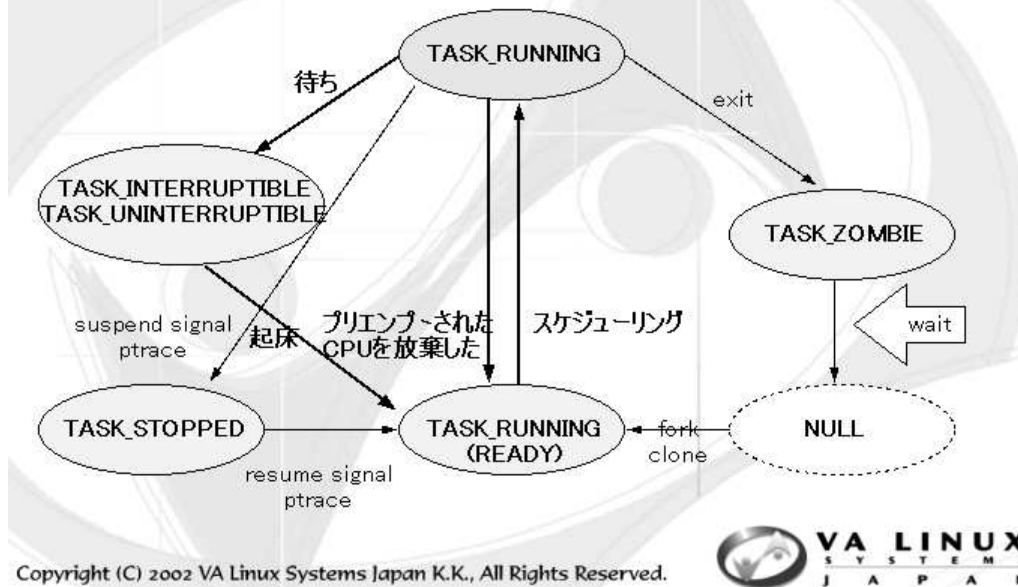
- 固定プライオリティ
 - niceコマンドなどで与えられる優先度
- 変動プライオリティ
 - 実行時間により動的に変更される
 - 実行時間が長いものほど優先度が下がる
 - I/O処理などが多いプログラムは優先度高
→応答性がよくなる
 - 優先度がそのままCPU時間に



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

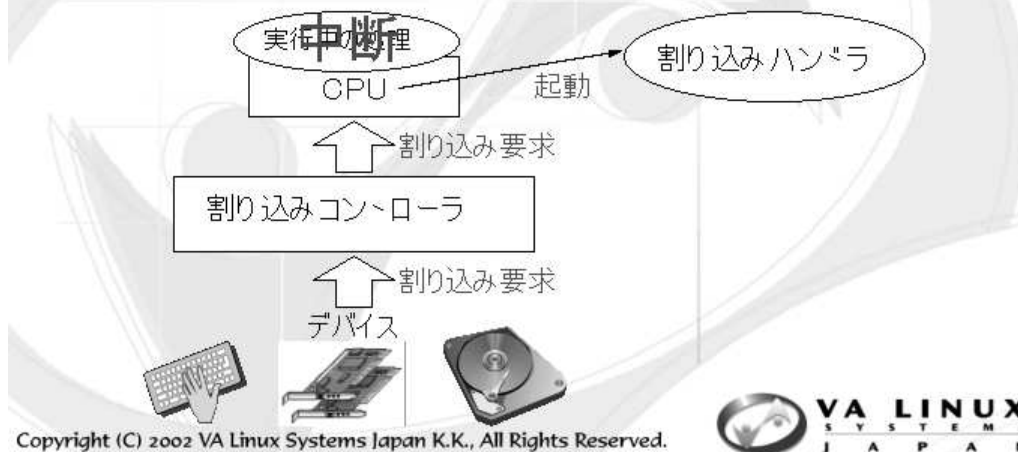


プロセスの状態



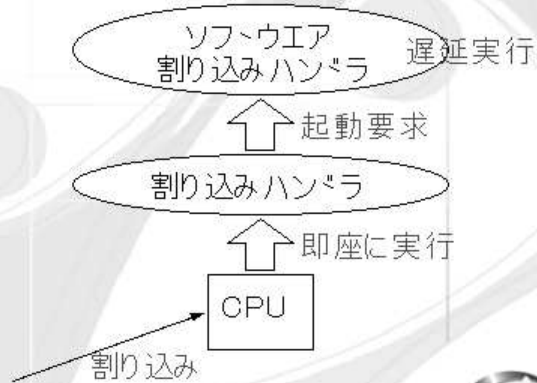
割り込み

ハードウェ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



割り込みハンドラ

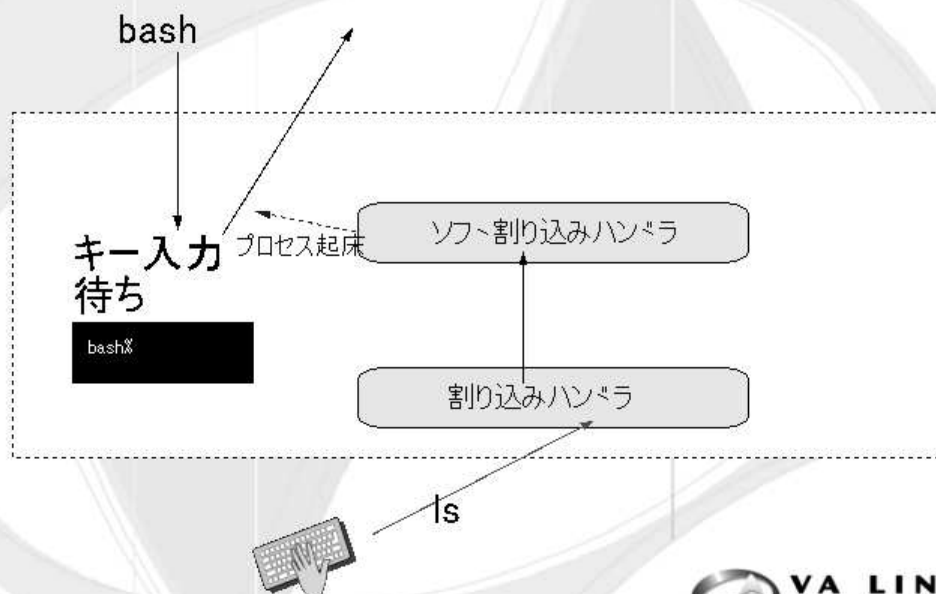
- 2つのハンドラ
 - ・割り込みハンドラ
 - ・ソフトウェア割り込みハンドラ(BHハンドラ)



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



典型的なI/O割り込み処理例

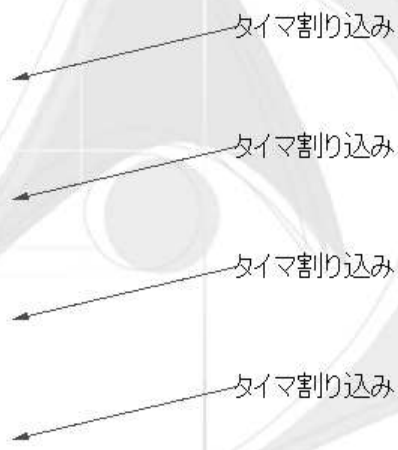


Copyright (C) 2002 VA Linux Systems Japan K.K., All Rights Reserved.



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



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

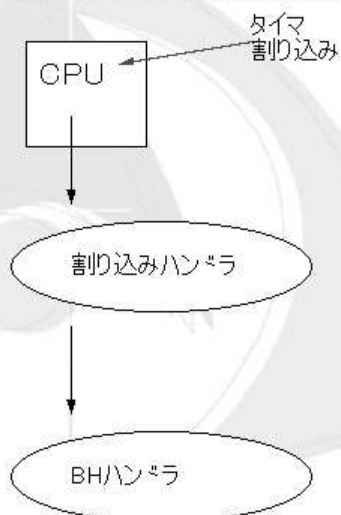


タイマ割り込みハンドラ

- ・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

- ・カレンダー(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



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



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

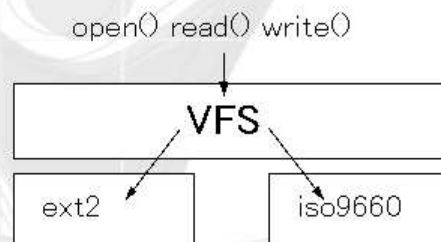


Copyright (C) 2002 VA Linux Systems Japan K.K., All Rights Reserved.



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



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



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

- ひとつのファイルにひとつのiノード
 - ファイルシステム管理情報
 - データブロック

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



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
- バッファキャッシュ

VFS

ページキャッシュ

ローカルファイルシステム

バッファキャッシュ

デバイスドライバ

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



空間管理

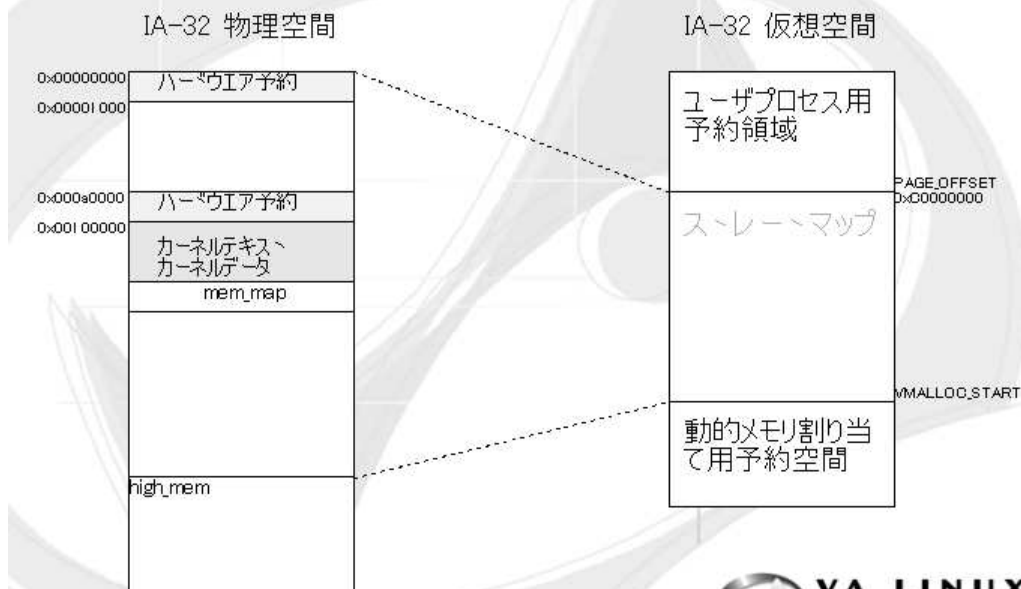
- 仮想メモリ空間(VM)を利用する
- ページ単位で管理する(4KB)
- 物理的なメモリと仮想メモリとの対応(ページテーブル)を利用して論物変換を行う(CPU)
- 実際の物理メモリよりも大きなサイズのプログラム群を動かすこともできる

その他のメモリ管理コンポーネント	
スラブアロケータ	動的空間割り当て
ページアロケータ ページ単位の管理	
ハードウェア	

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



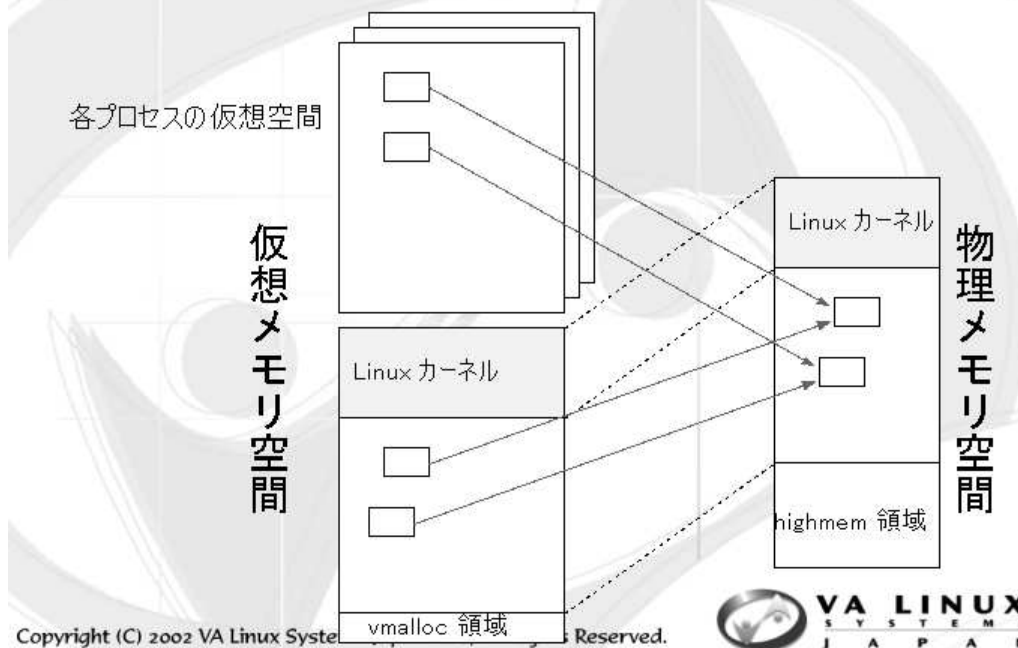
物理空間と仮想空間



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

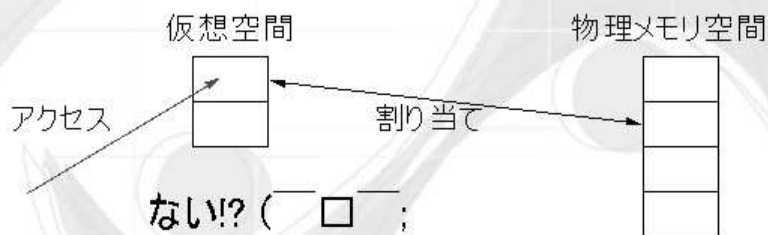


プロセスメモリ空間

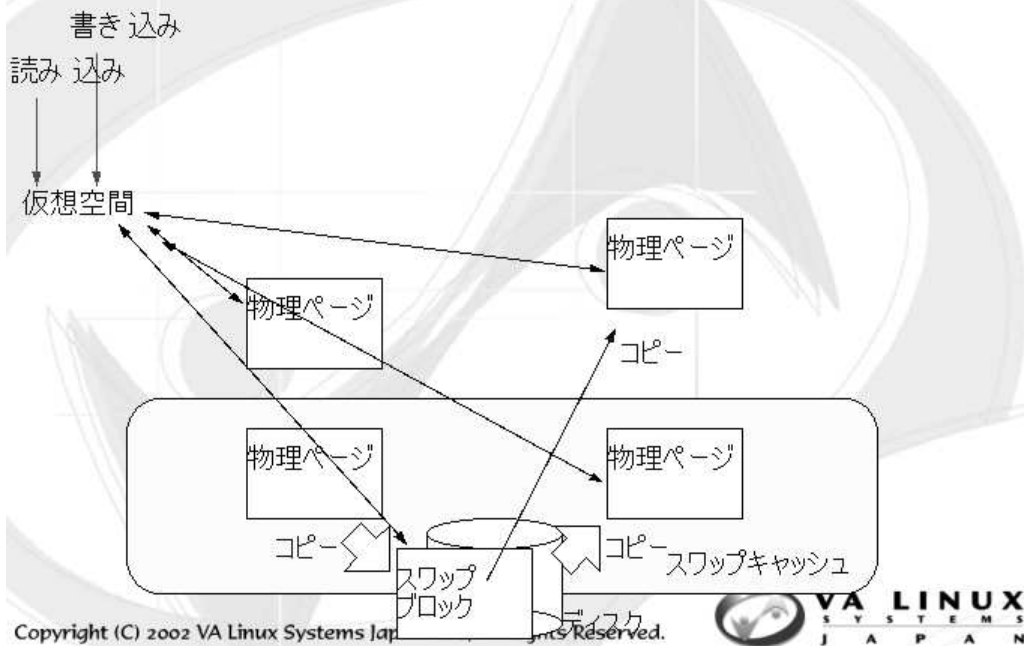


割り当て(デマンドロード)

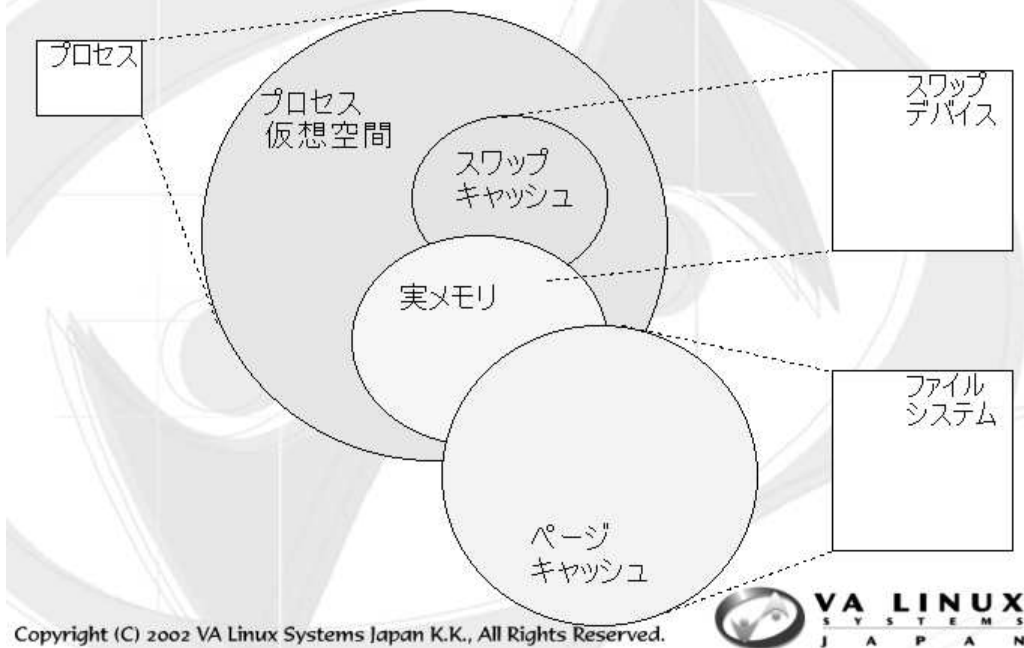
- できるだけ遅らせる
- ・デマンドロード
 - ・コピーオンライト(あとで説明)



ページアウト・ページイン



キャッシュとの関係



ページアロケータ

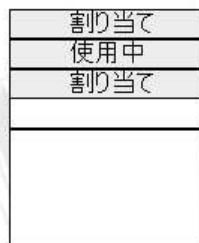
- ページの管理
- ページごとの管理情報
 - 参照カウント
 - 仮想マップ先のアドレス
 - アクティブか?
 - ...

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



バディシステム

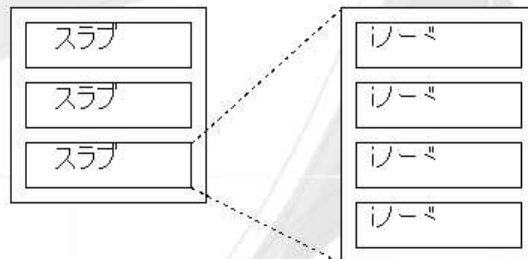
- ページアロケータの管理方法
 - 無作為にページを割り当てると、メモリのフラグメンテーションが起きやすい
- 2のべき乗個単位でページを管理する



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



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

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



プロセスの一生

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



ls コマンドの一生

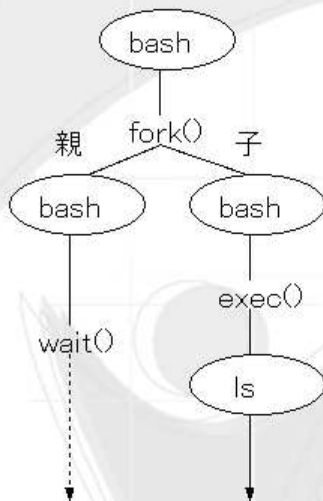
```
% ls
Makefile      dma.o         info.o        module.c      ptrace.c      softirq.c     timer.c
acct.c        exec_domain.c itimer.o      module.o      ptrace.o      softirq.o     timer.o
acct.o        exec_domain.o itimer.o      panic.c       resource.c    sys.c         uid16.c
capability.c  exit.c        kernel.o      panic.o       resource.o    sys.o         uid16.o
capability.o  exit.o        kmod.c        pm.c          sched.c       sysctl.c      user.c
context.c     fork.c        kmod.o        pm.o          sched.o       sysctl.o      user.o
context.o     fork.o        ksyms.c       printk.c      signal.c      time.c
dma.c         info.c        ksyms.o       printk.o      signal.o      time.o
```

1. 親プロセス(シェル)から ls コマンドが実行される
2. ls のプロセスが生成される
3. ディレクトリエントリを読む
4. ファイルを表示
5. プロセスの終了

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



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

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



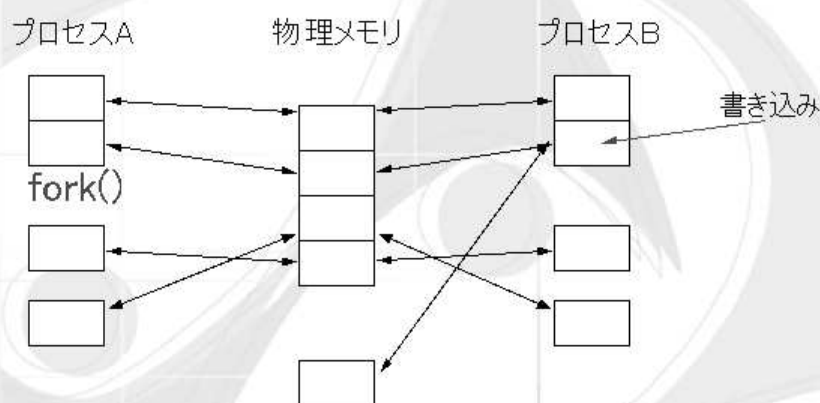
fork()システムコール

- ・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

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



コピーオンライト



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



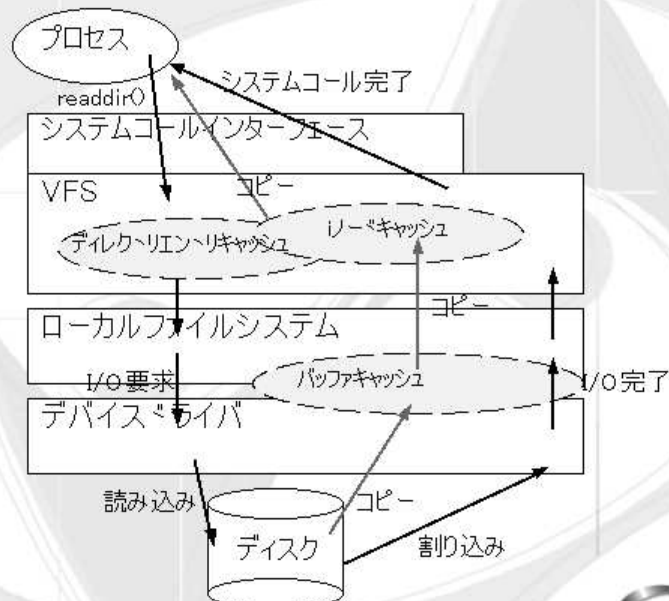
exec()システムコール

- ・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

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



ディレクトリ情報の読み出し



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



プロセスの終了

- `exit()` システムコール(子:ls)
 - `task_struct` 構造体以外のすべての資源を解放
 - 親プロセスが後始末をするまで、TASK_ZOMBIE(ゾンビ)として残る
- `wait()` システムコール(親: bash)
 - 親プロセスが子プロセスの終了を待つ
 - 子プロセスの残骸(`task_struct`)の後始末



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



その他のトピック

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



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

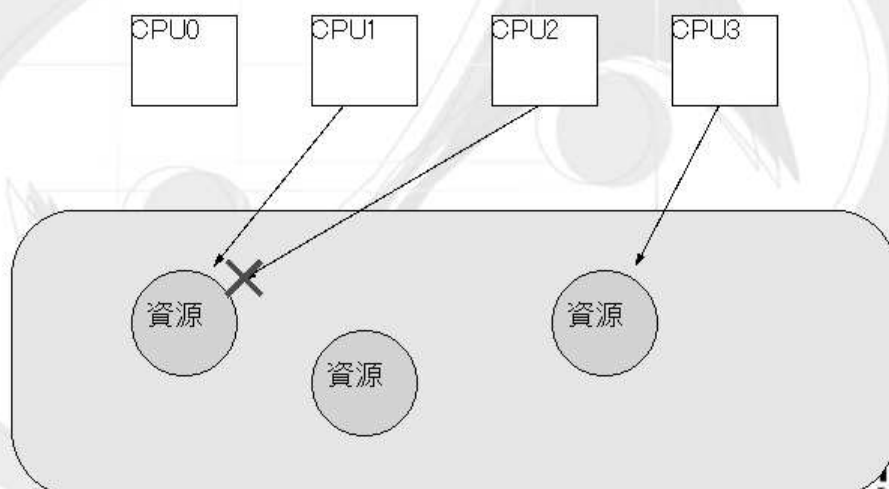


Copyright (C) 2002 VA Linux Systems Japan K.K., All Rights Reserved.



排他制御の仕組み 1

- スピンロック

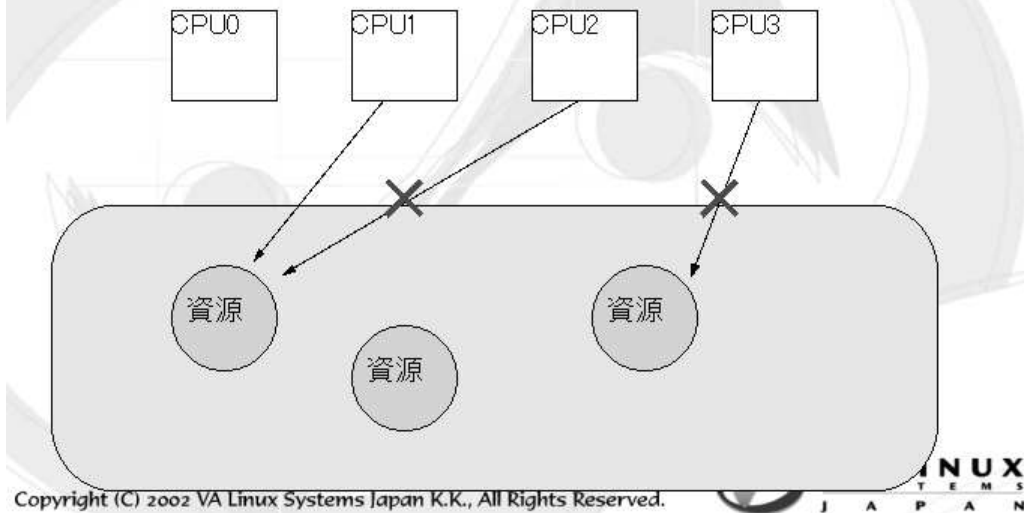


Copyright (C) 2002 VA Linux Systems Japan K.K., All Rights Reserved.



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

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



プロセス間通信(IPC)

- パイプ・FIFO(名前付きパイプ)
 - VFSで実装されている
 - オープンされたファイルと同様のインターフェースを提供
- SystemV IPC
 - セマフォ・メッセージ・共有メモリ
 - ipc()システムコール
- ソケット

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



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)

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



参考文献・資料

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



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今月号にPDFで収録

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



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

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



おわり

続いて、当社が誇る技術陣のセミナーを
お楽しみください

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

