

LPI-JAPAN 主催

LPICレベル1技術解説無料セミナー



LPI-Japan アカデミック認定校
NEC ネットズエスアイ株式会社
濱田 美奈子



はじめに

本セミナーは、Linuxをこれから学習しようと思っ
ている方、LPIC取得に向けて勉強を始めた
けれど体系的に理解するのが難しいと感じて
いる方を対象に「LPICレベル1取得のポイント」
を解説致します。

LPI101の試験範囲を中心に、基本的なコマンドや
Linuxの概念を実演を交えながらお話していきます。

アジェンダ

- ◆ LPIC(Linux技術者認定試験)の概要
- ◆ Linuxの構成、基本操作
- ◆ 101試験範囲よりポイント解説



会社紹介

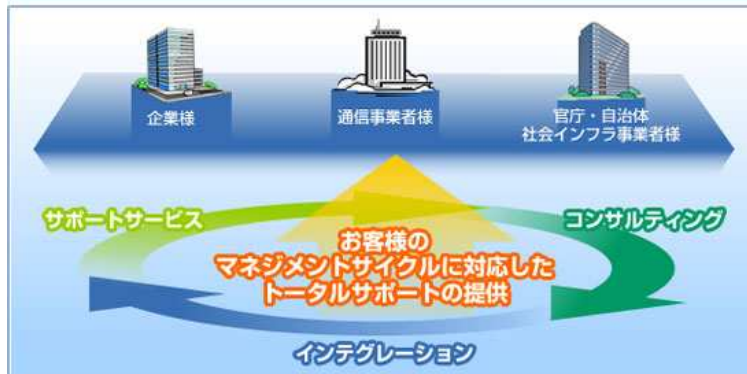
NEC
NECネットエスアイ

▶ 社名 NECネットエスアイ株式会社

<http://www.nesic.co.jp/>



お客様のベストパートナーとして課題解決を強力にサポート。
様々な視点からベストソリューションをご提供いたします。



事業内容：ネットワークシステムに関する、企画・コンサルティングや設計・構築などの提供および国内300ヶ所以上のサポートサービス拠点による、24時間365日対応の保守・運用、監視サービスならびにアウトソーシングサービスの提供。

NECネットエスアイは、人・企業・社会をスムーズにつなぐためのネットワークシステムの構築と、その基盤となる通信インフラの工事、そして、システムをより安心・安全にご利用いただくためのお客様サポートを展開しています。

LPIC(Linux技術者認定試験) の概要



LPICとは?

「LPIC」は、NPO法人/Linux技術者認定機関「LPI」が実施している
全世界共通・世界最大規模・最高品質の「Linux技術者認定制度」です

LPICの特長

- ・オープンソース
 - ー世界的なコミュニティで形成
- ・ベンダーニュートラル
 - ー様々な環境で知識が活かせる
- ・本質的な問題
 - ー技術的な本質を見極める問題
- ・広範囲に渡る出題
 - ー技術レベルを再認識



世界
標準

中立
公正

世界
最大



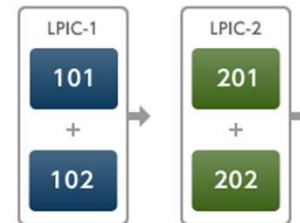
LPIC試験の構成と概要

LPICは3段階のレベルを設けステップアップしていく構成です

LPIC-1

実務に必要なLinuxの基本操作とシステム管理が行えるエンジニアとして認定されます。

Linuxサーバ環境の構築・運用・保守をするための基本的な操作が行えることを証明できます



LPIC-2

Linuxのシステムデザイン、ネットワーク構築が行えるエンジニアとして認定されます

LPIC-3

Linux、Windows、UNIXなどの混合環境や、高負荷に耐えうる大規模システムの構築ができるエンジニアであることを証明できます



101試験の出題範囲

- 主題101: システムアーキテクチャ

この範囲の一部を本日解説

- 主題102: Linuxのインストールとパッケージ管理
 - 主題103: GNUとUnixのコマンド
-
- 主題104: デバイス、Linuxファイルシステム、
ファイルシステム階層標準

101試験範囲詳細: <http://www.lpi.or.jp/lpic1/range/range101.shtml>

101試験：出題範囲詳細(一例)

103.1 コマンドラインで操作する	
重要度	4
説明	コマンドラインを使用して、シェルおよびコマンドと対話する。 この目標は、bashシェルを使用することを想定している。
主要な知識範囲	<ul style="list-style-type: none">• 1つのシェルコマンドおよび1行のコマンドシーケンスを使用して、コマンドラインでの基本的な作業を行う• 定義することを含めたシェル変数の使用と変更、環境変数の参照とエクスポート• コマンド履歴の使用と編集• 定義済みパス内に存在するコマンドおよび存在しないコマンドの呼び出し
重要なファイル用語、ユーティリティ	<ul style="list-style-type: none">• <code>.</code>• <code>bash</code>• <code>echo</code>• <code>env</code>• <code>exec</code>• <code>export</code>• <code>pwd</code>• <code>set</code>• <code>unset</code>• <code>man</code>• <code>uname</code>• <code>history</code>



102試験の出題範囲

- 主題105: シェル、スクリプト、およびデータ管理
- 主題106: ユーザーインターフェースとデスクトップ
- 主題107: 管理業務
- 主題108: 重要なシステムサービス
- 主題109: ネットワークの基礎
- 主題110: セキュリティ

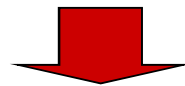
102試験範囲詳細: <http://www.lpi.or.jp/lpic1/range/range102.shtml>

Linuxの構成・基本操作



「オープンソースのOS」

- 1991年に当時フィンランドのヘルシンキ大学在学中だったリーナス・トーバルズが個人で「Linuxカーネル」を開発
- ソースコード（設計図）がインターネット上に公開されている
- OSのカスタマイズが可能(ソフトウェアの改版、再配布も可能)
- 無償入手可能（CentOS、Fedora、Ubuntu etc.）



ベンダーに特化したOSではない



Linuxディストリビューションの構成

本来、Linuxとは**カーネル**と呼ばれるプログラムのことを示しますがカーネルだけではOSとして機能しないため、必要な機能も合わせてパッケージ化された「ディストリビューション」として提供されます

Linuxディストリビューションに含まれるもの

・インストーラ
インストールさせるプログラム

・カーネルモジュール
デバイスドライバなど



・Linuxカーネル
Linux本体 (中核となるプログラム)



・アプリケーション
サーバ機能、コマンドなどの
プログラム



・シェル
ユーザの指示をカーネルに
伝えるプログラム



・オンラインマニュアル
OS上で参照できる
マニュアル





Linuxディストリビューションの種類

ディストリビューションの種類によってパッケージ管理システムが異なる

Red Hat系

パッケージ管理システムとしてRPM形式を採用

CentOS (無償)



RHELのクローンOS

Red Hat Enterprise Linux



Fedoraをベースにして
安定させた「有償版」のOS
略称：RHEL

Fedora (無償)



レッドハットが支援するコミュニティ
「Fedora Project」によって開発され
最新の技術を積極的に取り込んでいる

Debian系

パッケージ管理システムとしてDebian形式を採用

Debian GNU/Linux (無償)



Debian Project によって開発される

Ubuntu Linux (無償)



使いやすいGUIを目指す
Debian系ではシェアが高い

その他 ディストリビューション

OpenSUSE (無償)



ノベルがスポンサーを務め、コミュニティ
が開発する



Linuxの構成と101出題範囲

アプリケーション

主題102: Linuxのインストールとパッケージ管理

コマンド

主題103: GNUとUNIXコマンド

シェル

主題103: GNUとUNIXコマンド

ライブラリ

主題102: Linuxのインストール

Linuxカーネル

プロセス管理

主題103: GNUとUNIXコマンド

ファイルシステム管理

主題104: Linuxファイルシステム

データ入出力管理

主題103: GNUとUNIXコマンド

ハードウェア管理

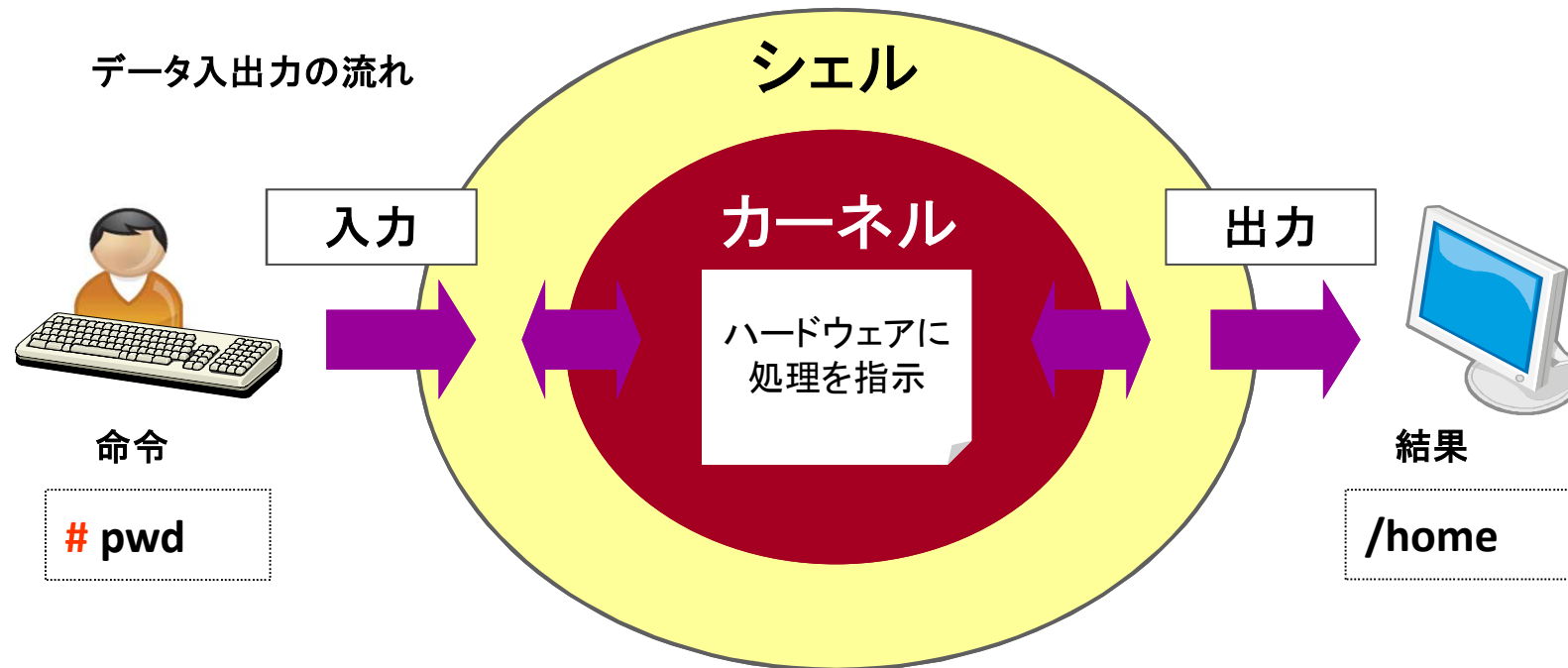
主題101: システムアーキテクチャ

ハードウェア



カーネル・シェルの役割

「カーネル」は、LinuxOSの中核となるプログラムで、プロセス管理、メモリ管理、ファイル管理、デバイス管理、データ入出力管理などを行います



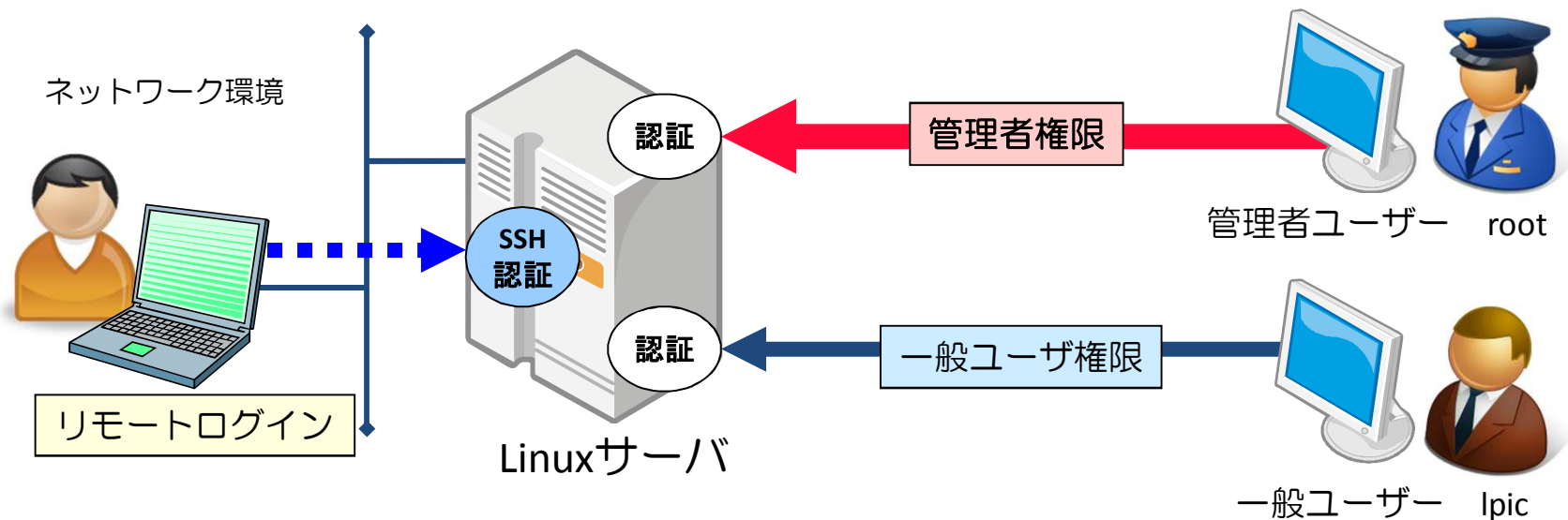
カーネルは機械語でしか理解できず、ユーザが入力したコマンドを解釈できないため「シェルプログラム」が間に入って仲介します



マルチユーザシステム

Linuxは複数のユーザで同時に利用することができます

- Linuxを利用するためには、ユーザ名とパスワードによるログイン認証が必要
- ユーザによって利用権限(ファイル閲覧、プログラム実行)が異なる
- リモートログインにはSSHを利用する

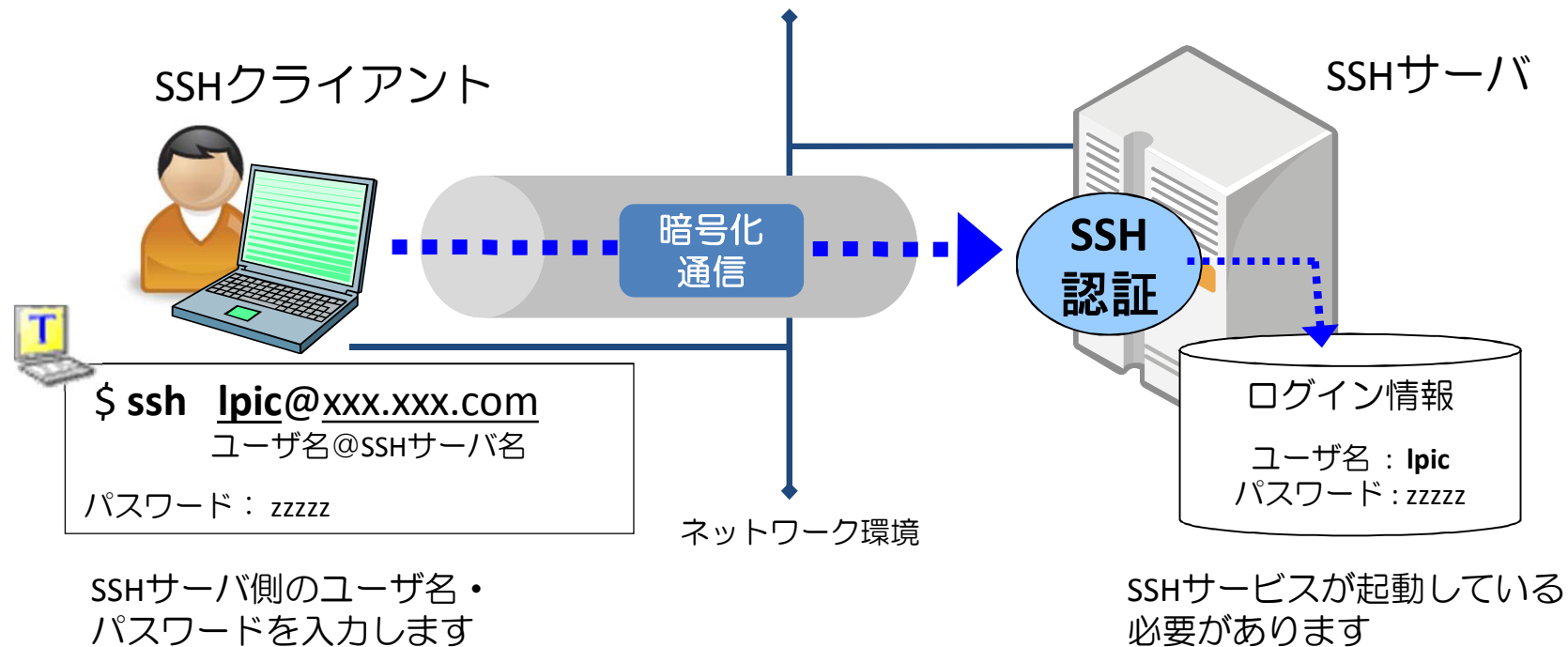




SSHによるリモートログイン

SSH(Secure SHell)はネットワークで接続された他のコンピュータを遠隔操作するためのプロトコルです

- SSHサーバ設定は LPI102試験「110.3 暗号化によるデータの保護」で学びます





ユーザーの種類

- Linuxのユーザは「管理者」と「一般ユーザ」の2つに分けられます
- ユーザは必ず1つ以上のグループに所属し、ユーザ権限とグループ権限が与えられます

管理者 = スーパーユーザ = root



すべてのコマンド実行権限がある
すべてのファイルにアクセス権がある
管理者権限を持つアカウントは1つだけ
インストール時にパスワードを設定する

一般ユーザ(例：lpic)



実行できるコマンドに制限がある
アクセスできるファイル・ディレクトリに制限がある
インストール後に root権限でユーザを作成する



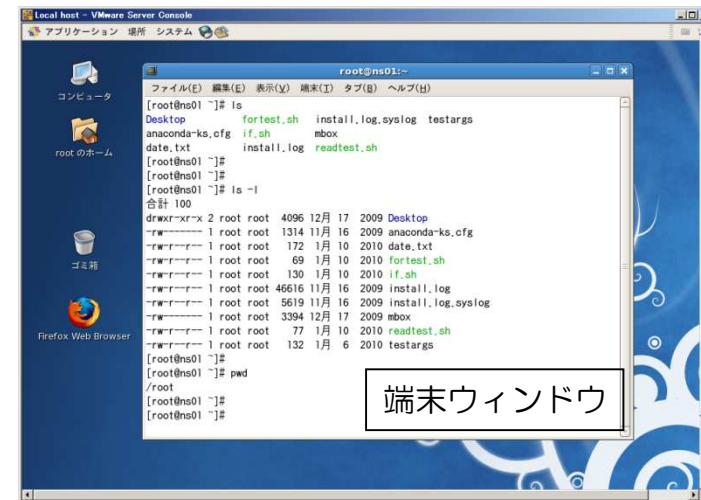
ユーザインターフェース

LinuxのユーザインターフェースはCUIとGUIがあります

```
CentOS release 5.4 (Final)
Kernel 2.6.18-164.11.1.el5 on an i686

ns01 login: root
Password:
Last login: Wed Jan 26 12:52:14 on tty1
[root@ns01 ~]#
[root@ns01 ~]#
[root@ns01 ~]# _
```

CUI



GUI

- Linuxは「コマンド実行」でOS操作を行うのが主流
- GUI環境ではターミナル(端末)ウィンドウを起動しコマンドを実行する
- コマンド操作に慣れることがポイント



ログインの実行(CUI) ランレベル3

- ログイン(利用開始)

```
CentOS release 5.4 (Final)
Kernel 2.6.18-164.11.1.el5 on an i686

ns01 login: root
Password: _
```



```
CentOS release 5.4 (Final)
Kernel 2.6.18-164.11.1.el5 on an i686

ns01 login: root
Password:
Last login: Wed Jan 26 12:52:14 on tty1
[root@ns01 ~]#
[root@ns01 ~]#
[root@ns01 ~]# _
```

- ① 「login:」のあとに root を入力して Enter
「password:」のあとに パスワード を入力して Enter (※パスワードは表示されません)

- ② ログイン認証が完了したら
コマンドプロンプトが表示され
コマンド入力が可能となります

- ログアウト

```
# logout
# exit

上記コマンドを実行
```

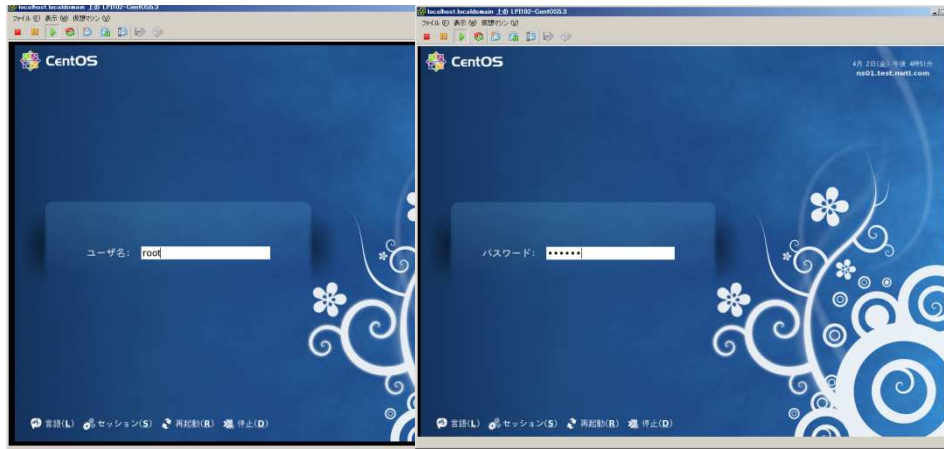
又は

「Ctrl + D」キーを
押下する

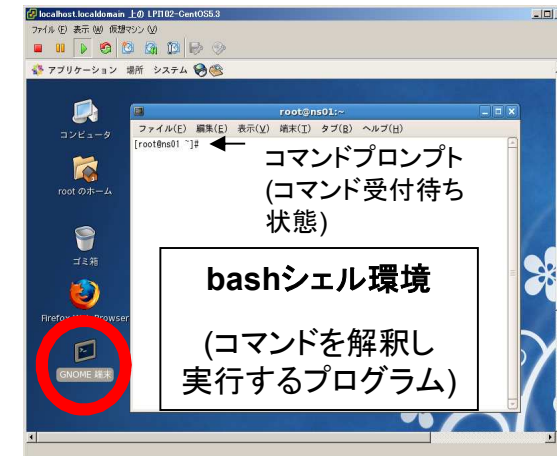


ログインの実行(GUI) ランレベル5

- ログイン(利用開始)



- ① 「ユーザー名」のあとに **root** を入力して Enter
「パスワード」のあとに **パスワード** を入力して Enter
(※パスワードは表示されません)



- ② Linuxが起動したら
「GNOME端末」アイコンを
クリックして、コマンド実行の
準備をします

- ログアウト(終了)

「GNOME端末」を終了し、メニューバーからログアウトを選択する

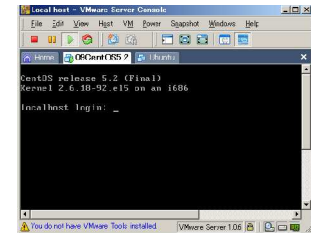


ランレベル (起動レベル)

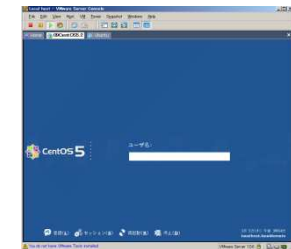
- ランレベル(起動レベル) • • Linuxシステムの実行状態を示す(RedHat 系)

- 0 停止 (シャットダウン)
- 1 シングルユーザモード (システムメンテナンスモード)
- 2 マルチユーザモード (CUIログイン、NFSなし)
- 3 マルチユーザモード (CUIログイン)
- 5 マルチユーザモード (GUIログイン)
- 6 再起動(リブート)

s, S, single シングルユーザモード



CUIログイン



GUIログイン

ランレベルの変更コマンド

```
# init 3
```

```
# telinit 3
```

ランレベル3に切り替える

ランレベル設定ファイル

```
/etc/inittab デフォルトのランレベル設定
```

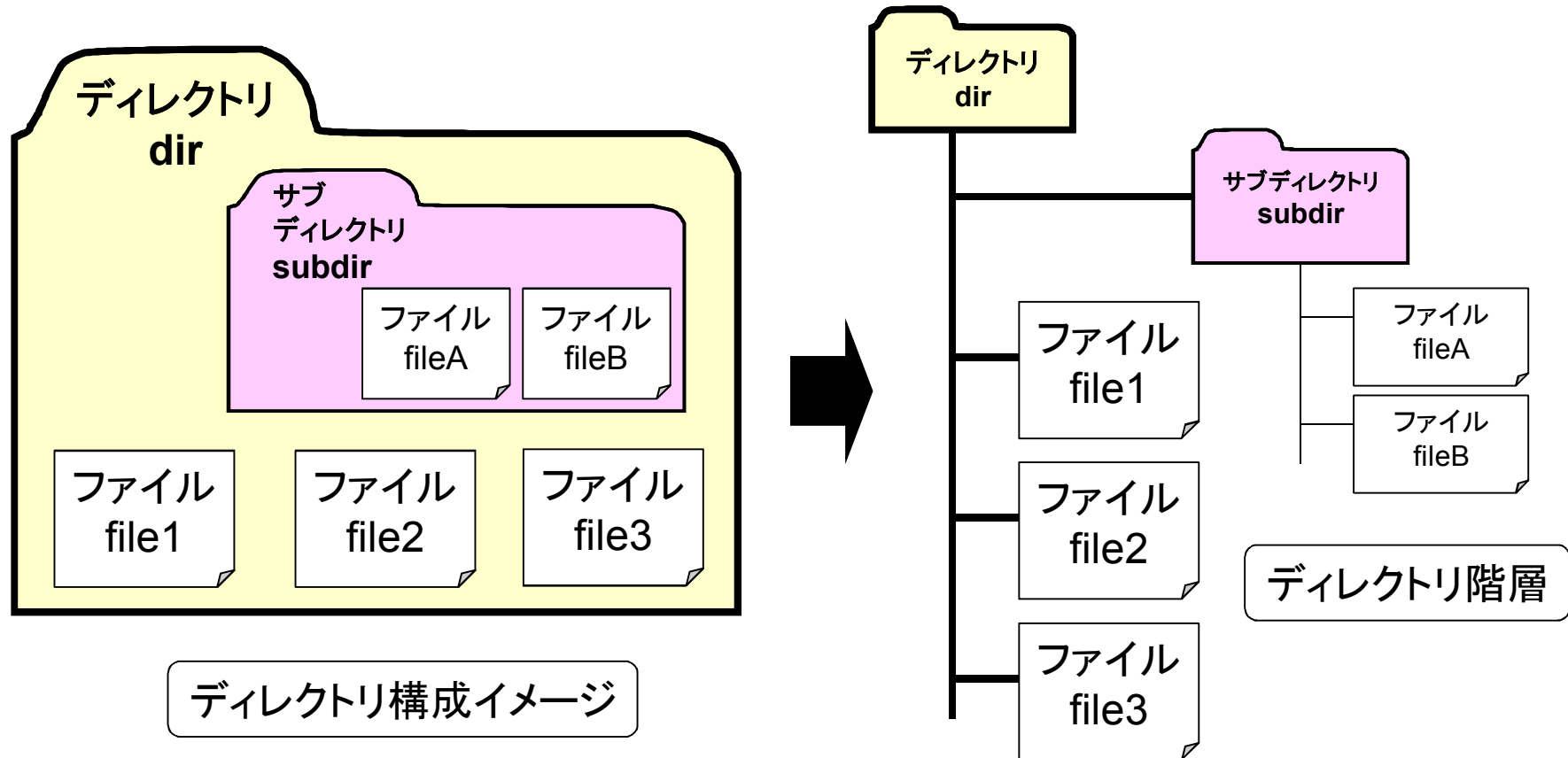
ランレベル確認コマンド

```
# runlevel
```



ディレクトリとは

ファイルを格納する入れ物のことを Windows ではフォルダと
いいますがLinuxでは『**ディレクトリ**』と表します。

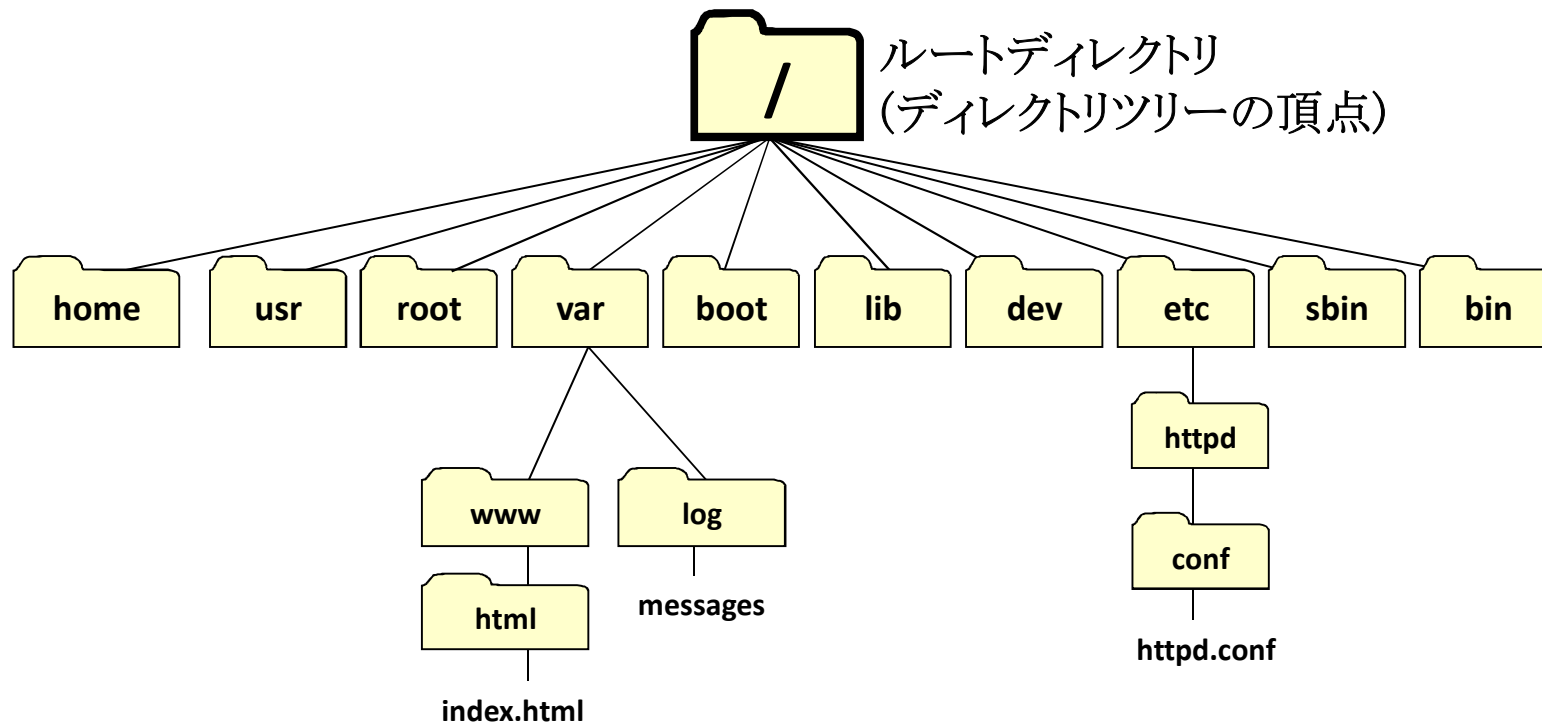




ルートディレクトリ

ディレクトリはツリー状の階層構造で管理されており、頂点のディレクトリを『ルートディレクトリ』と呼びます。

Linuxでは全てのファイル・ディレクトリをルートディレクトリ以下に格納します。

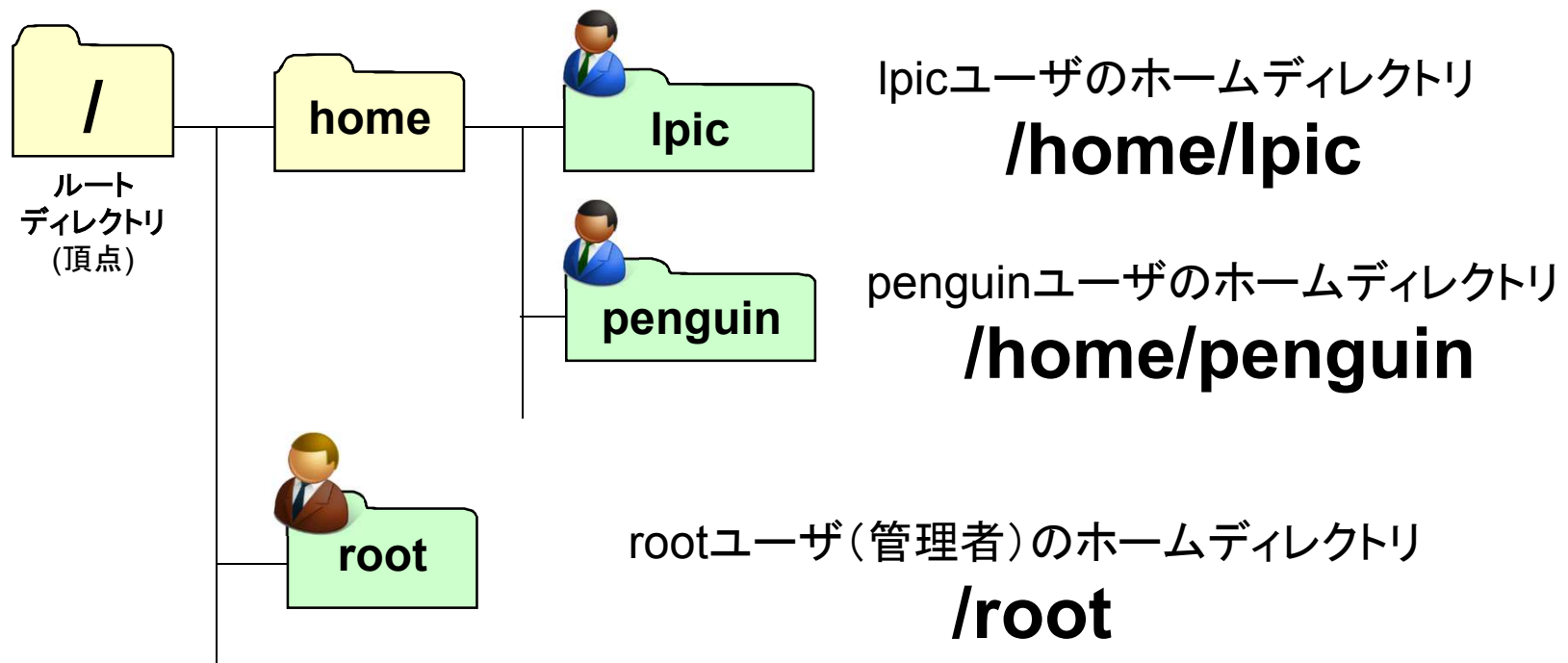




ホームディレクトリ



ユーザ毎に与えられるディレクトリで、ログイン直後のカレントディレクトリ(ユーザが作業を行う場所)です



101 試験範囲よりポイント解説



本日解説するポイント

- ◆ 101試験範囲よりポイント解説
 - 102.5 RPMおよびYUMパッケージ管理を使用する
 - 103.1 コマンドラインで操作する
 - 103.3 基本的なファイル管理を行う
 - 103.5 プロセスを生成、監視、終了する
 - 103.8 viを使って基本的なファイル編集を行う

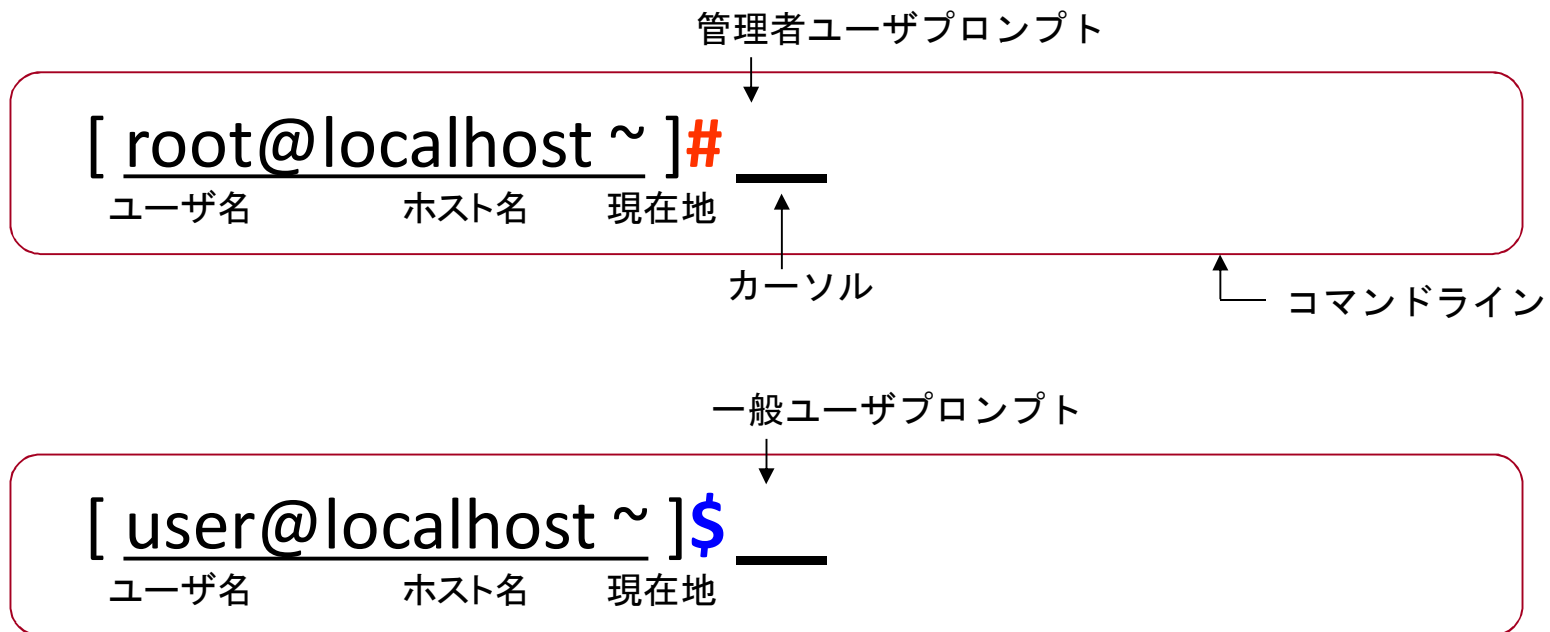
103.1 コマンドラインで操作する



コマンドプロンプト

- ・コマンドプロンプトとコマンドライン

システムがコマンドの入力を待っている状態を示す記号をプロンプト、またコマンドを入力する行のことをコマンドラインといいます





コマンドの基本構文

- コマンド入力で気をつけること

半角英数字を使う

コマンド、オプション、引数の間は半角スペースを空ける

大文字と小文字を区別するので正しく入力する

入力の最後に Enter キーを押す

オプションはハイフン(-)を付けて指定する

- コマンドの基本構文

引数・・・コマンドに与える値 (対象ディレクトリ・ファイルなど)

オプション・・・コマンドの機能拡張で利用する

コマンド

```
# ls  
コマンド
```

コマンド 引数

```
# ls /tmp  
コマンド 引数
```

コマンド オプション

```
# ls -l  
コマンド オプション
```

コマンド オプション 引数

```
# ls -l /tmp  
コマンド オプション 引数
```



基本的なコマンド

cd	ディレクトリを移動する
\$ cd /tmp	/tmp ディレクトリへ移動する
\$ cd ~lpic <small>移動先ディレクトリ</small>	lpicユーザのホームディレクトリへ移動する

pwd	カレントディレクトリを表示する
\$ pwd /etc	現在のカレントディレクトリを表示する → /etc ディレクトリ上にいることが分かる

特殊記号	説明	使い方
~	ホームディレクトリの指定	# cd Δ ~lpic lpicユーザのホームディレクトリへ移動する
.	カレントディレクトリの指定	# cat Δ ./a.txt カレントディレクトリ以下の a.txt を参照する
..	1つ上(親)ディレクトリの指定	# cd Δ .. 1つ上の親ディレクトリへ移動する



ls コマンド

ls

ディレクトリ内のファイル・サブディレクトリー一覧表示

- # ls カレントディレクトリ以下のファイル・サブディレクトリを表示する
- # ls Δ -l カレントディレクトリ以下のファイル詳細情報も表示する
- # ls Δ -a Δ /home /homeディレクトリ内の全てのファイル・ディレクトリを表示する

※ファイルの中で、ファイル名の先頭に『.(ドット)』が付いた「隠しファイル」と言われるものがあります。
隠しファイルは ls コマンドに -a オプションを付けないと確認することが出来ません。

```
# ls -l /etc/hosts
```

```
-rw-r--r--  1  root  root  187  3月13  15:44  /etc/hosts
```

↑ ↑ ↑ ↑ ↑ ↑ ↑

アクセス権 リンク数 所有者 所有グループ サイズ 更新日 更新时间 ファイル名

-	rw-	r--	r--
ファイル 種類	所有者	所有グループ	その他
	6	4	4

↑
ファイルの種類

文字	説明
-	通常ファイル
d	ディレクトリ
l	リンクファイル
c , b	特殊デバイスファイル



bashシェルの便利な機能

- 補完機能

bashシェルにはコマンド名やファイル名を補完する機能があります

ファイル名補完

```
# cat /etc/hosts Tab Tab ←途中でTabキーを2回押す  
hosts hosts.allow hosts.deny ←候補があった場合表示される  
  
# cat /etc/hosts.a Tab ←「.a」まで入力しTabキーを押す  
  
# cat /etc/hosts.allow ←残りの部分が補完される
```

コマンド名補完

```
# da Tab ←途中でTabキーを1回押す  
  
# date ←残りの部分が補完される
```



オンラインマニュアルの参照

Linuxではオンラインマニュアルが用意されており、コンソール上で
man コマンドを実行し参照することができます。
マニュアルは内容毎に、セクションが分かれています。

オンライン
マニュアル
「passwd」



番号	内容
1	ユーザ用コマンド
2	システムコール
3	ライブラリ関数
4	デバイスファイル
5	ファイルフォーマット
6	ゲーム
7	その他
8	システム管理用コマンド
9	カーネル用ドキュメント



マニュアルの参照

調べたいキーワードに対して、**man** ページが複数存在する場合があります。その場合は該当するセクション番号を指定します。

man passwd

PASSWD(1)	PASSWD(1)
名前	passwd - ユーザパスワードを変更する
書式	passwd [-f -s] [name] passwd [-g] [-r -R] group passwd [-x max] [-n min] [-w warn] [-i inact] login passwd {-l -u -d -S -e} login
説明	passwd はユーザアカウント・グループ アカウントのパスワードを変更する。~以下省略

man 5 passwd

セクション番号

PASSWD(5)	PASSWD(5)
名前	passwd - パスワードファイル
説明	passwd ファイルには各ユーザアカウントの様々 な情報が記録されている。書かれているのは次の通り。 ログイン名 暗号化されたパスワード (無いこともある) ユーザ ID 番号 グループ ID 番号 ユーザ名またはコメントのフィールド ユーザのホームディレクトリ ユーザのコマンドインタプリタ ~以下省略

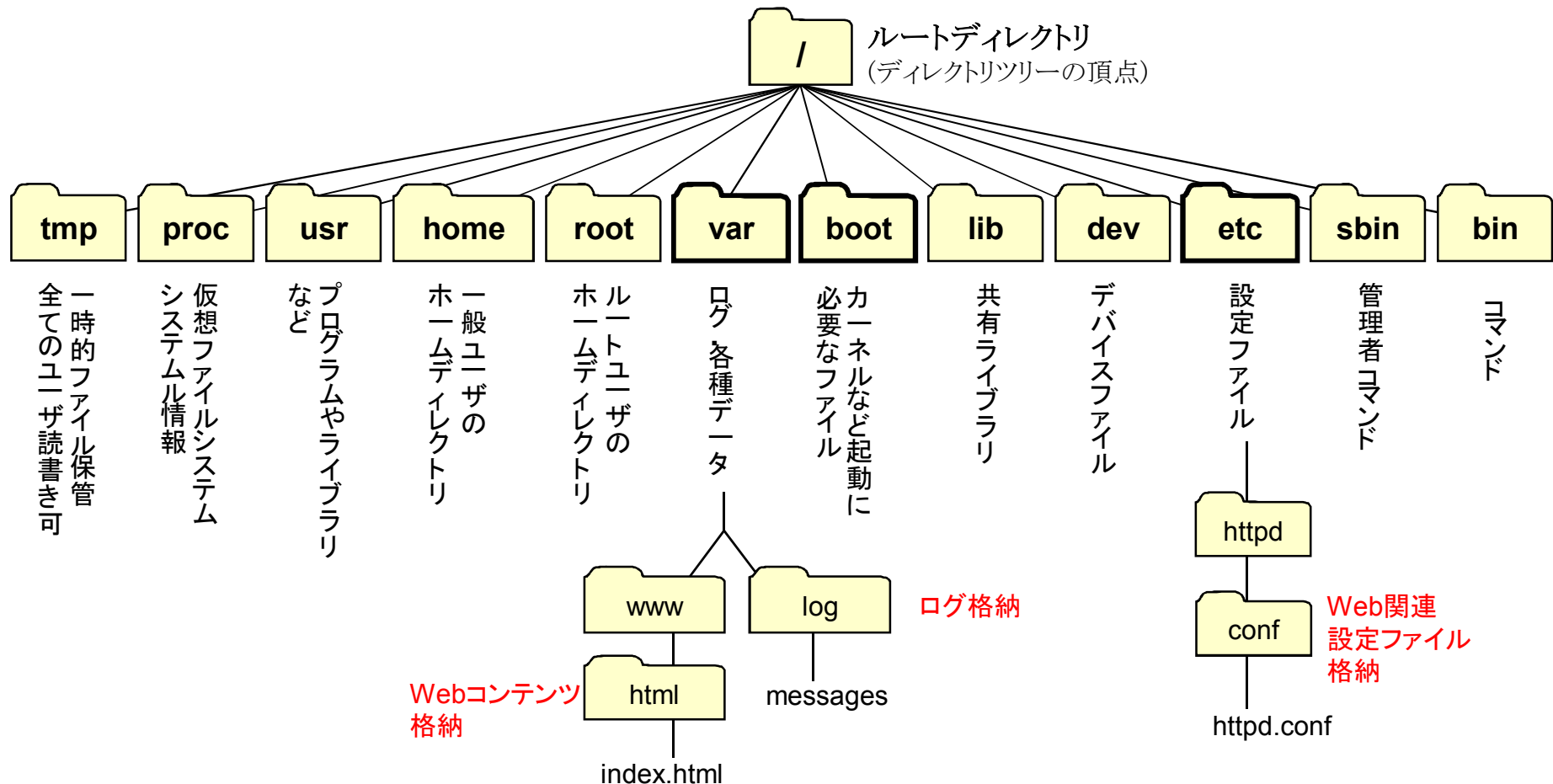
セクション番号の指定がない場合は小さいセクション番号から表示されます。

103.3 基本的なファイル管理を行う



ディレクトリの配置

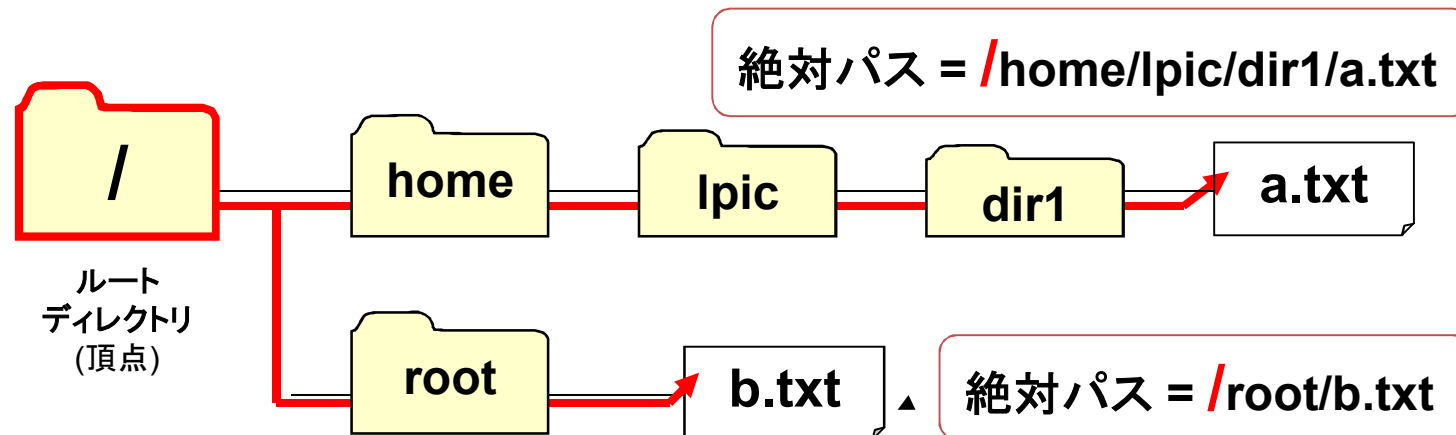
ルートディレクトリ以下でファイル・ディレクトリを配置する基準は『FHS(ファイルシステム階層標準)』という規格で定められています





パスの概念

ファイルを開いたり、コマンドを実行したりするには、対象となるファイルやディレクトリの場所を正確に指定しなければなりません。
ファイルやディレクトリへ辿り着く経路を『パス(Path)』といいます。



- 絶対パス

ルートディレクトリを基点としたパスは『絶対パス(フルパス)』といい必ずルートディレクトリを表す「/」からパスを記述します。

/home/lpic/dir1/a.txt (`/home/lpic/dir1` は `a.txt` までの経路)

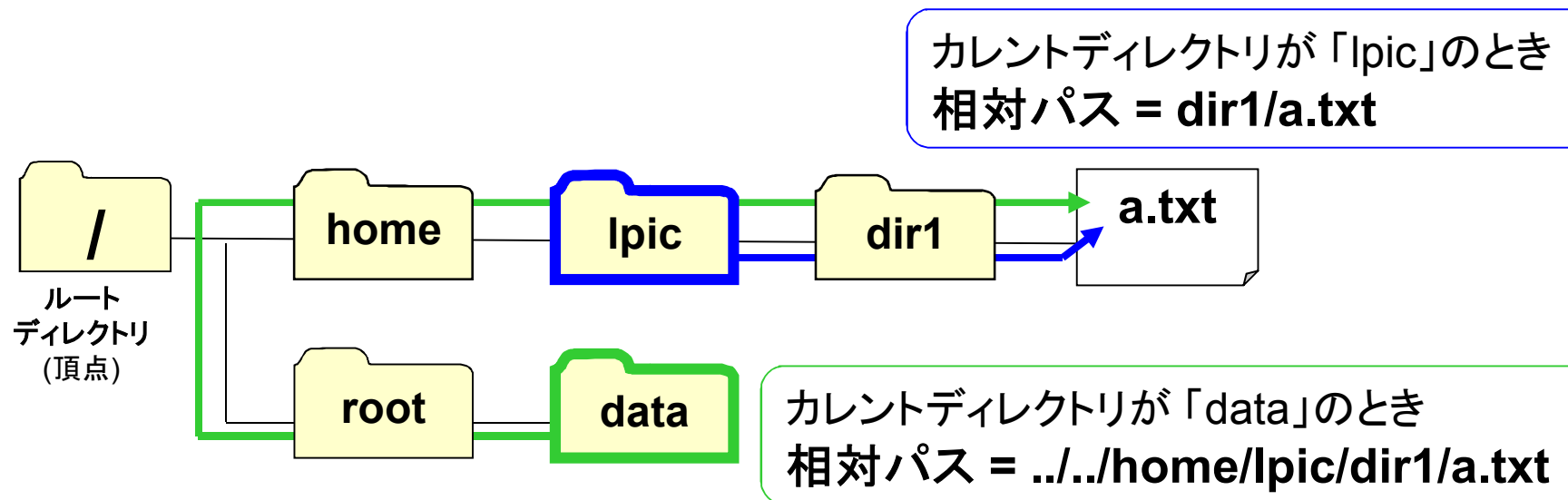


相対パス

- 相対パス

カレントディレクトリ(ユーザの現在地)を基点としてパスを指定する方法を『相対パス』といいます。

カレントディレクトリの場所によってパス表記が変わるのが特徴です。





基本的なコマンド(2)

mkdir

ディレクトリを作成する

mkdir △ /root/data

/root の下に data ディレクトリを作成する

\$ mkdir △ **-p** △ ~lpic/dir1/test

lpicユーザのホームディレクトリ以下にて
親ディレクトリも同時に作成する

親ディレクトリも同時に作成

※ test を作成するにはひとつ上の親ディレクトリ dir1 が存在する必要があります
親ディレクトリも同時に作成する場合は **-p** オプションを使います

rm

ディレクトリを削除する

rm △ **-r** △ /root/data

ディレクトリ内全て削除

/root の下の data ディレクトリを削除する

※ディレクトリ削除のときは **-r** オプションを使います

rm: remove directory `/root/data'? **y** 削除するか聞かれるので y を入力する

rm

ファイルを削除する

rm △ /root/b.txt

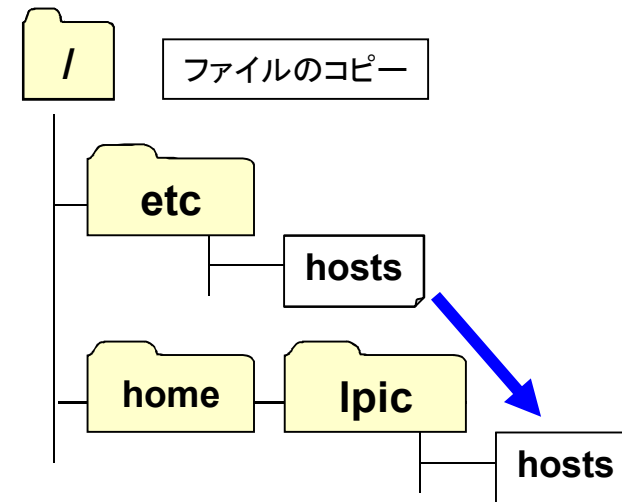
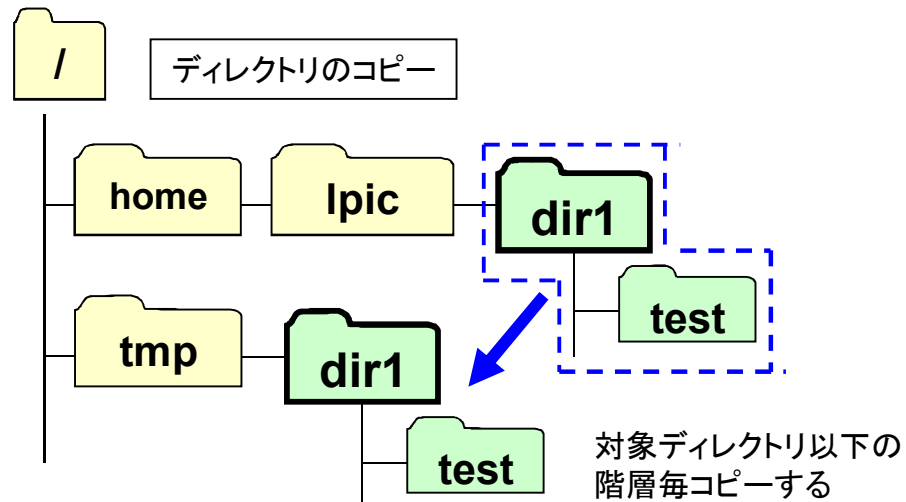
/root の下の b.txt ファイルを削除する
(オプション必要なし)

rm: remove 通常の空ファイル `/root/b.txt'? **y**



基本的なコマンド(3)

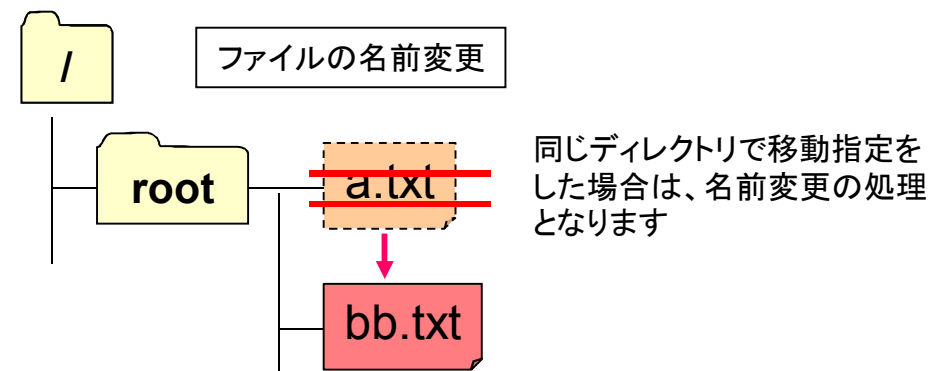
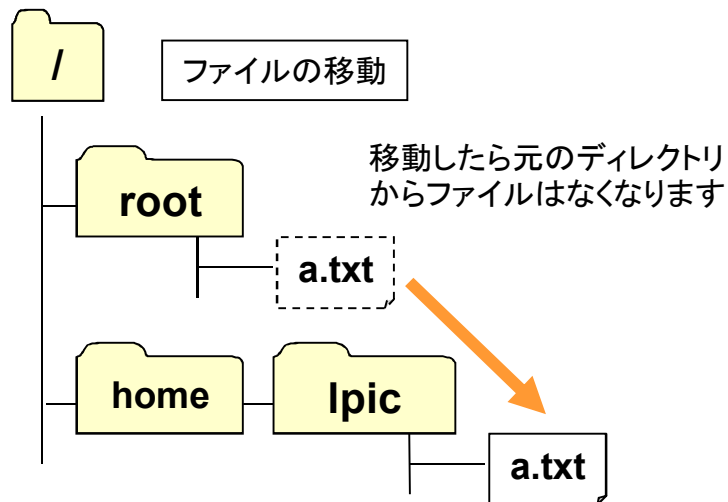
cp	ディレクトリをコピーする
# cp Δ <u>-r</u> Δ <u>~/pic/dir1</u> Δ <u>/tmp</u> ディレクトリのコピー コピー元ディレクトリ コピー先	~/pic/dir1ディレクトリを /tmpディレクトリ以下へコピーする
cp	ファイルをコピーする
\$ cp Δ <u>/etc/hosts</u> Δ <u>~/pic</u> コピー元ファイル コピー先	/etc/hostsファイルをpicユーザのホームディレクトリへコピーする





基本的なコマンド(4)

mv	ファイルの移動
# mv Δ /root/a.txt Δ ~lpic <small>移動元ファイル 移動先ディレクトリ</small>	/tmp 以下の a.txt ファイルを lpicユーザのホームディレクトリへ移動する
mv	ファイルの名前変更
# mv Δ /root/a.txt Δ /root/bb.txt <small>変更前ファイル名 変更後ファイル名</small>	a.txt ファイルを bb.txt へ名前変更する ※同一ディレクトリで移動した場合は、ファイル名の変更となります。





ファイル閲覧コマンド

cat	ファイルの中身を表示
\$ cat Δ <u>/etc/hosts</u> ファイル名	/etc/hostsファイルの内容を表示する (cat コマンドは行数がたくさんあると表示が流れてしまう)
less	ファイルの中身を先頭から表示 (ページャ)
\$ less Δ <u>/etc/inittab</u> ファイル名 q を入力して終了	/etc/inittabファイルの内容を表示する ← lessは実行すると閲覧モードに入ってしまうため、q(quit)で終了する
head	ファイルの先頭部分を表示
\$ head Δ <u>/etc/inittab</u> ファイル名	/etc/inittabファイルの先頭部分を表示する (デフォルトは 10行)
tail	ファイルの末尾部分を表示
\$ tail Δ <u>/etc/inittab</u> ファイル名	/etc/inittabファイルの末尾部分を表示する (デフォルトは 10行)
\$ tail Δ -f Δ <u>/var/log/messages</u> リアルタイム表示 ファイル名	ログファイルをリアルタイム表示する

103.8 viを使って基本的なファイル 編集を行う



vi コマンドモードとインサートモード

テキストファイルを編集するためには『テキストエディタ』を使います。

vi エディタには「コマンドモード」と「インサートモード(文字入力モード)」があり、2つのモードを切り替えながら作業を行います。

vi エディタの起動

vi △ /etc/resolv.conf →

vi コマンドで viエディタを起動します。

ファイルが既存にあれば
「ファイル編集」、
ファイルが存在しなければ
「ファイル新規作成」となります。

vi エディタ コマンドモード
(メニュー操作)

```
search lpic.biz
nameserver 192.168.60.1
nameserver 192.168.50.252
~
~
~
"/etc/resolv.conf" 3L, 111C
```

vi 起動直後はコマンドモード

vi エディタ インサートモード
(文字の入力)

```
search lpic.biz
nameserver 192.168.60.1
nameserver 192.168.50.252
hama ← 新しい文字を入力できる
~
~
-- INSERT -- ← インサートモード
```

文字入力はインサートモード

インサートモードへの
切り替え

i a →

← Esc 文字入力の終了



vi コマンドモードでの操作 基本編

- 基本的なコマンドモードでの操作

インサートモードへの切り替え	
コマンド	説明
i	カーソル前にテキスト入力
a	カーソル後にテキスト入力
I	行頭にテキスト入力
A	行末にテキスト入力
o	カーソルの下に行挿入、テキスト入力
O	カーソルの上に行挿入、テキスト入力

カーソル移動	
コマンド	説明
0	行の先頭へ移動する
\$	行の末尾へ移動する
G	ファイル内の最終行へ移動する
:n	ファイル内 n行へ移動する

編集コマンド	
コマンド	説明
x	カーソル位置の文字削除 (Delete)
X	カーソル位置手前の文字削除 (Backspace)
dd	行の削除 (切り取り) 2行分→ 2dd
yy	行のコピー 5行分→ 5yy
p	カーソルの下に貼り付け
P	カーソルの上に貼り付け
u	Undo 直前操作のやり直し

vi の終了	
コマンド	説明
:q!	内容を保存せずに終了する
:wq 又は ZZ	内容を保存して終了する
:W △ファイル名	ファイルを指定して保存する



vi コマンドモードでの操作 応用編

検索コマンド	
コマンド	説明
/パターン	カーソル位置から下に向かってパターンを検索する
?パターン	カーソル位置から上に向かってパターンを検索する
n	次を検索する
N	次を検索する (逆方向)
:noh	候補のハイライト表示を解除
:%s/AA/BB/	文字列AAを文字列BBに1つ置換する
:%s/AA/BB/g	文字列AAを文字列BBにすべて置換する

vi の設定変更、シェルコマンドの実行、viの終了	
コマンド	説明
:set nu	行番号を表示する
:set nonu	行番号を非表示する
:! Δ コマンド	vi を終了せずにシェルコマンドを実行する
:w!	終了せず保存のみ実行 (上書き保存)

```

1 LPIC Level 1
2 spring
3 summer
4 fall
5 winter
6 LPIC Level 1
6 LPIC Level 1 hamada
7 spring
7 spring hamada
8 summer
8 summer hamada
9 fall
9 fall hamada
10 winter
10 winter hamada
~
~
/hamada ← 下方向へ検索

```

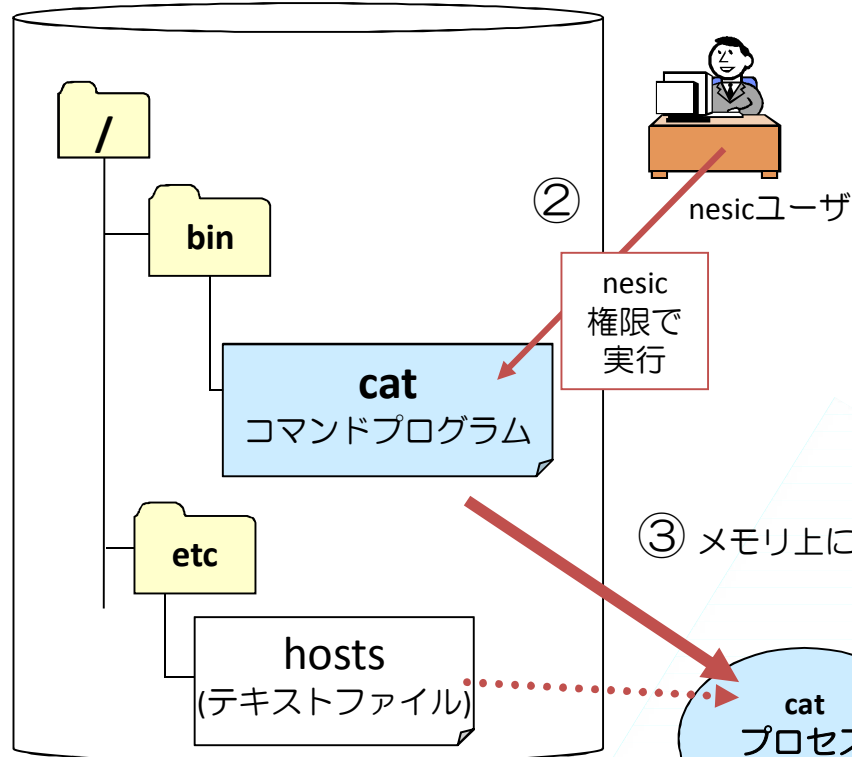
カーソル位置から
下方向に検索

103.5 プロセスを生成、監視 終了する



プロセスとは

① \$ cat /etc/hosts nescユーザでコマンド実行



コマンドやプログラムファイルを実行するとプログラムがメモリに読み込まれ「プロセス」が生成されます。

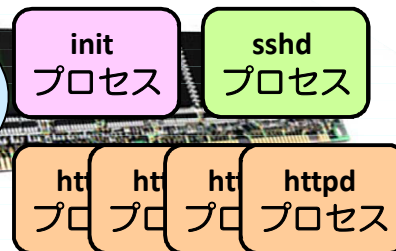
プロセスにはID(PID)が付き、システム内で識別されます。

③ メモリ上に読み込まれる

サーバ機能提供のプロセスは「デーモン」と呼ばれ、メモリ上に常駐し続ける

④ コマンド結果を出力

```
[nesc@ns01 ~]$ cat /etc/hosts
127.0.0.1 localhost.localdomain
192.168.25.50 ns01.test.nwtl.com
192.168.25.50 ns01.test.nwtl.com
```





プロセス監視コマンド

ps	現在実行されているプロセス表示
\$ ps	ユーザ自身が起動しているプロセスを表示
\$ ps Δ aux	システム全体で実行されているプロセスを表示
	a .. 他のユーザのプロセスも表示 u .. 実行ユーザ名も表示 x .. 制御端末のないプロセスも表示する
<pre>USER PID %CPU %MEM VSZ RSS TTY STAT START TIMECOMMAND root 1 0.0 0.0 2060 584 ? Ss Mar25 0:00 init [3] root 2 0.0 0.0 0 0 ? S< Mar25 0:00 [migration/</pre>	

システム全体のプロセスを監視する際は、オプション `aux` を指定することが多いです。

オプション `aux` はハイフン(-)を付けずに指定します。

pstree	プロセスの階層構造を表示
\$ pstree Δ -p	プロセスIDも表示
<pre>init(1)--acpid(4798) --atd(5083) --login(4422)---bash(4424)---ls(4425) --sshd(4430)</pre>	

プロセスは階層構造で管理されています。

元のプロセスを「親プロセス」、親プロセスから起動されたプロセスを「子プロセス」と呼びます。



プロセスの終了

プロセスの動作(停止・再起動)を制御するには、シグナルと呼ばれる命令を送信する必要があります。

kill	指定したプロセスにシグナルを送信
# kill 500 プロセスID	プロセスID 500番のプロセスを正常終了する (シグナル指定がない場合のデフォルト動作)
# kill -9 100	} プロセスID100番のプロセスを 強制終了する シグナルはどの指定でも動きは同じ
# kill -KILL 100	
# kill -SIGKILL 100	

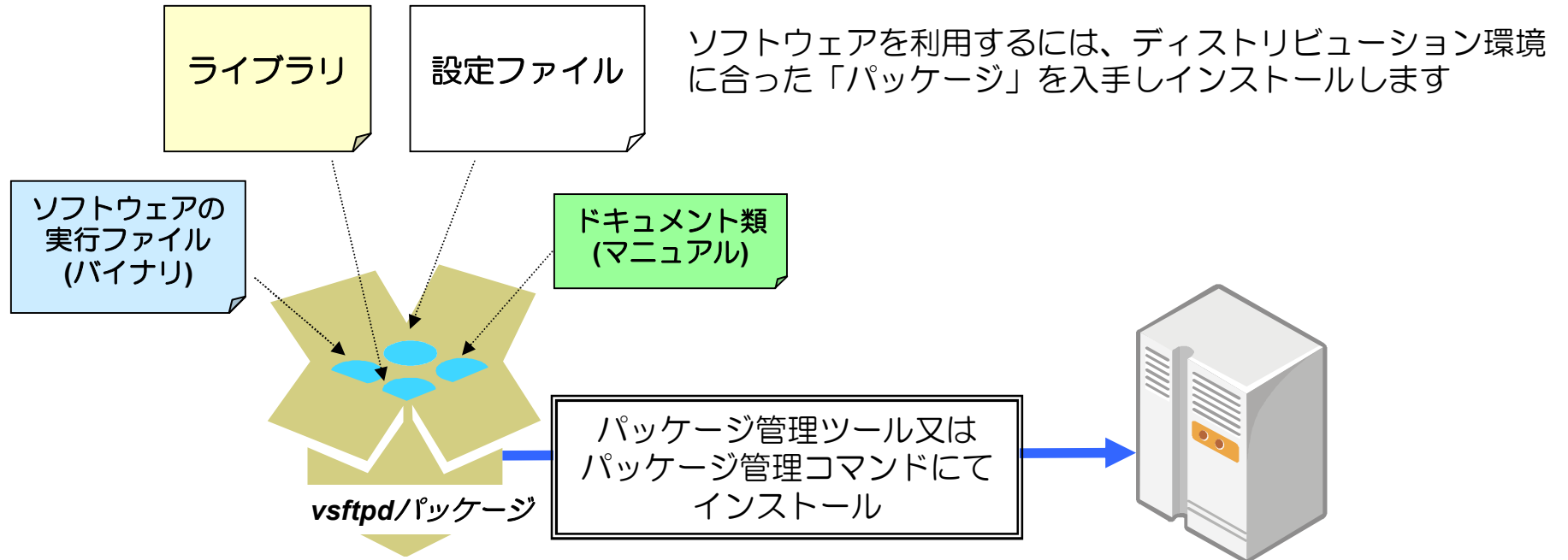
• 代表的なシグナルの種類

シグナル		シグナル番号	動作
HUP	SIGHUP	1	設定再読み込み、プロセスの再起動
INT	SIGINT	2	Ctrl + Cキー押下など割り込み通知
KILL	SIGKILL	9	プロセスの強制終了 ※使用注意
TERM	SIGTERM	15	プロセスの正常終了(デフォルト)

102.5 RPMおよびYUM パッケージ管理を使用する



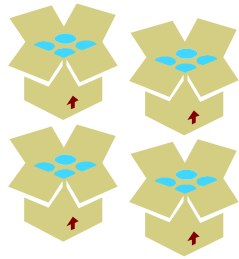
パッケージ管理とは



ディストリビューション	Debian系 (Debian, Ubuntu)	RPM系 (RHEL, CentOS, fedora)
パッケージ管理コマンド (パッケージを入手し手動インストール)	dpkg	rpm
パッケージ管理ツール (ネットワーク経由で自動インストール)	APT	YUM
パッケージファイル名	vsftpd-1-2.3.i386.<u>deb</u>	vsftpd-1-2.3.i386.<u>rpm</u>



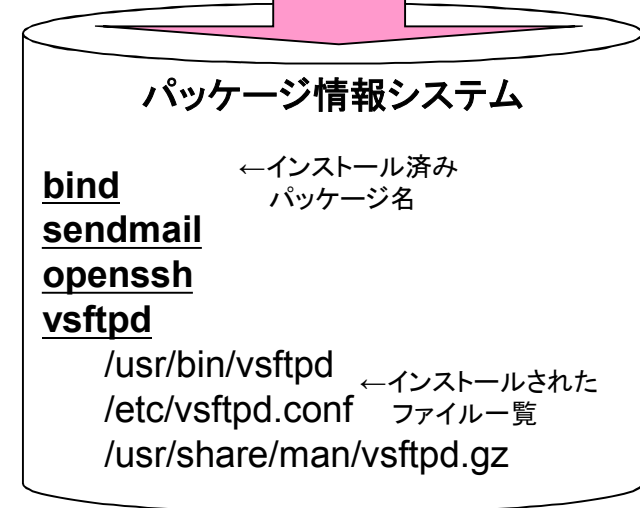
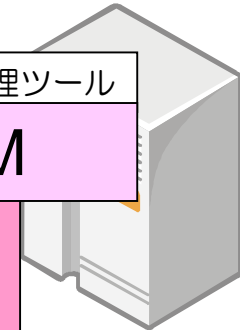
[RPM系] YUMパッケージ管理



RedHat系
パッケージ提供サーバ



ネットワーク経由でサーバから
パッケージを自動ダウンロードし
インストール又はアップデートを行う
〈パッケージ間の依存関係を解決〉



```
YUM パッケージインストール
# yum △ install △ apache2
                サブコマンド      パッケージ名
```

```
YUM 全てのパッケージをアップデート
# yum △ update
```

```
YUM リポジトリ上最新パッケージ情報を取得
# yum △ check-update
```

```
YUM パッケージ情報を検索
# yum △ search △ apache2
                パッケージ名
```



[RPM系] rpmコマンド

rpmは RPM形式のパッケージを管理するためのコマンド
あらかじめサーバに用意されたパッケージを使用する
<パッケージ間の依存関係に問題があった場合はエラーとなる>

rpmコマンド インストール

```
# rpm Δ -ivh Δ apache2.1.i386.rpm  
パッケージファイル名
```

rpmコマンド パッケージのアップデート(なければインストール)

```
# rpm Δ -Uvh Δ apache2.1.i386.rpm  
パッケージファイル名
```

rpmコマンド インストール済みパッケージのみアップデート

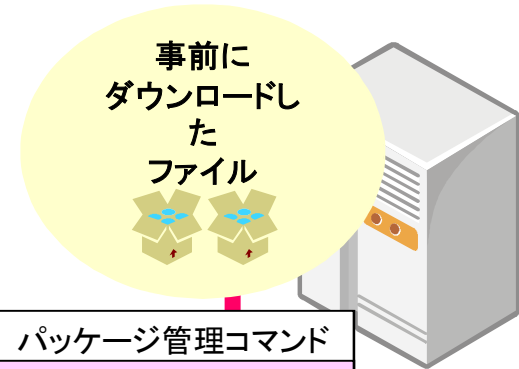
```
# rpm Δ -Fvh Δ apache2.1.i386.rpm  
パッケージファイル名
```

rpmコマンド パッケージをアンインストールする

```
# rpm Δ -e Δ apache2  
パッケージ名
```

rpmコマンド [照会] インストール済みの全てのパッケージ表示

```
# rpm Δ -qa
```

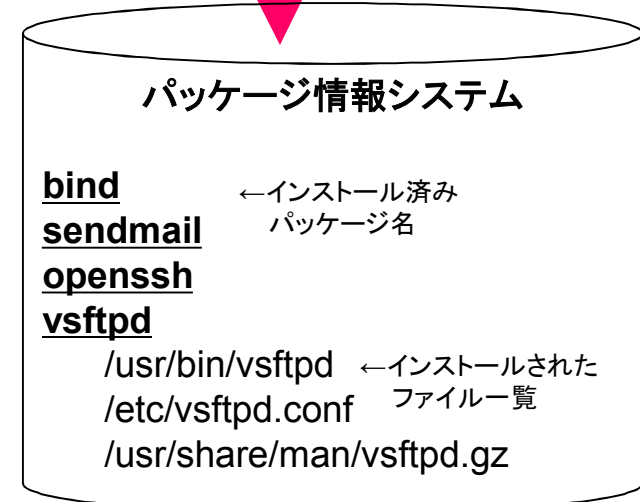


パッケージ管理コマンド

rpm

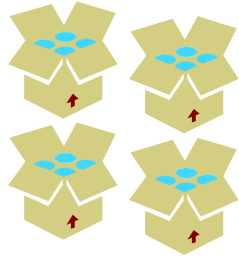
RedHat系
Linuxサーバ

用意されたパッケージの
インストール・アップデートを行う





[Debian系] APTパッケージ管理



Debian系
パッケージ提供サーバ



ネットワーク経由でサーバから
パッケージを自動ダウンロードし
インストール又はアップデートを行う
〈パッケージ間の依存関係を解決〉

パッケージ管理ツール

APT



APT パッケージインストール

```
# apt-get  $\Delta$  install  $\Delta$  apache2
```

サブコマンド パッケージ名

APT 全てのパッケージをアップグレード

```
# apt-get  $\Delta$  upgrade
```

APT リポジトリ上最新パッケージ情報を取得

```
# apt-get  $\Delta$  update
```

APT パッケージ情報を検索

```
# apt-cache  $\Delta$  search  $\Delta$  apache2
```

パッケージ名

パッケージ情報システム

bind ←インストール済み
sendmail パッケージ名

openssh

vsftpd

/usr/bin/vsftpd ←インストールされた
/etc/vsftpd.conf ファイル一覧
/usr/share/man/vsftpd.gz



[Debian系] dpkgコマンド

dpkgは Debian形式のパッケージを管理するためのコマンド
あらかじめサーバに用意されたパッケージを使用する
<パッケージ間の依存関係に問題があった場合はエラーとなる>

dpkgコマンド パッケージのインストール

```
# dpkg  $\Delta$  -i  $\Delta$  apache2.1.i386.deb  
パッケージファイル名
```

dpkgコマンド 設定ファイルを残してアンインストール

```
# dpkg  $\Delta$  -r  $\Delta$  apache2  
パッケージ名
```

dpkgコマンド 設定ファイルも含め全てをアンインストール

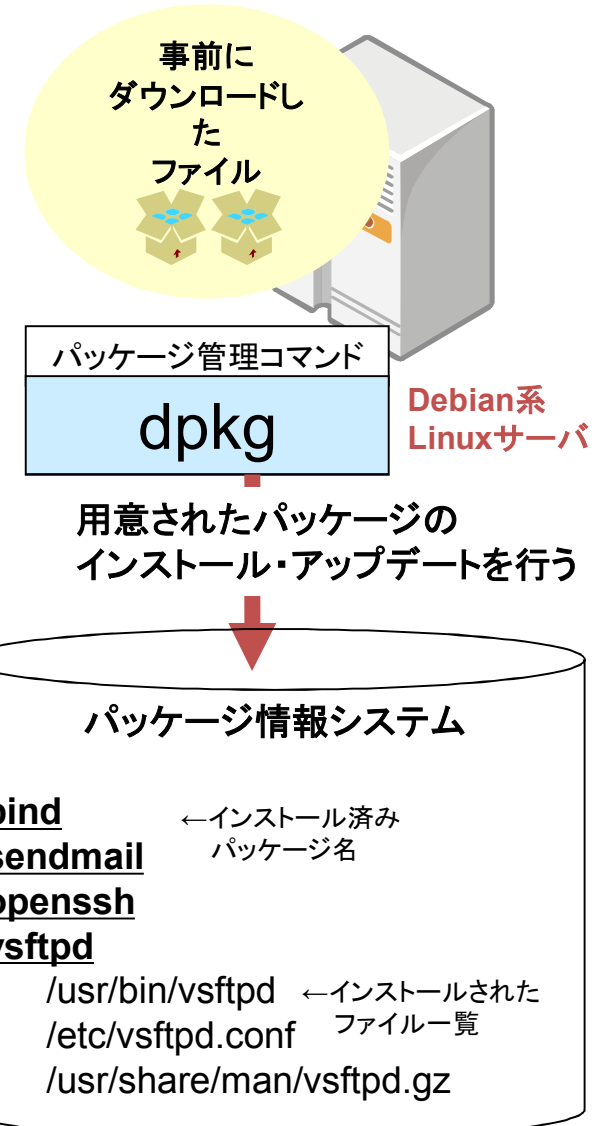
```
# dpkg  $\Delta$  -P  $\Delta$  apache2  
パッケージ名
```

dpkgコマンド インストール済みパッケージの詳細情報を表示

```
# dpkg  $\Delta$  -s  $\Delta$  apache2  
パッケージ名
```

dpkgコマンド インストール済みパッケージを一覧表示

```
# dpkg  $\Delta$  -l
```





参考資料



Linux標準教科書 (Ver1.1.1)
<http://www.lpi.or.jp/linuxtext/text.shtml>
発行：エルピーアイジャパン



Linux 教科書LPICレベル1第4版
中島能和(著) / 濱野賢一郎(監修)
発行：翔泳社
2009/05/12発売
定価3,990円
ISBN- 9784798119311



徹底攻略LPI問題集 Level1/Release3 対応
鳥谷部昭寛、菅蒲淳司 著 / 株式会社ソキウス・ジャパン 編
発行：インプレスジャパン
2009/11/20発売
定価2,835円
ISBN：9784844327943

ご清聴いただきありがとうございました。

ご質問がございましたら、お声掛け下さい。



LPI-JAPAN

<http://www.lpi.or.jp/>

NEC

NECネットエスアイ

NECネットエスアイ株式会社

<http://www.nesic.co.jp/>