オープンソース教育100年構想
リナックスアカデミー

Linux Professional Institute Japan LPI-JAPAN

# LPICレベル1技術解説セミナー

リナックスアカデミー
2014年6月21日
矢越昭仁

Ver. 1.5

# UNIXの歴史



- 1969年　AT&Tベル研 Ken Thompson が私的研究として基本ソフトUNIXを開発
  ARPAがパケット網の実験を開始（後のInternet）

- 1974年　AT&T がUNIX ver.5を一般に公開
- 1977年　BSD版をBill Joyが中心に開発し公開
- 1983年　Richard StallmanがFSF設立(GNU Project)

- 1984年　AT&TがIT市場に参戦（UNIXライセンス問題）
- 1987年　X/Open が規格発表
- 1988年　IEEEがPOSIX発表

# Linux概要

- 1991年、ヘルシンキ大（当時）Linus Torvaldsが、ゼロから開発したPOSIX準拠カーネルを公開
（GPLライセンス　＝　非AT&Tライセンス）
- Linux OSは、ディストリビュータにより、大きくRed Hat系とDebian系に分類される
- LPICは原則ディストリビューション非依存だが、パッケージ管理は両系統とも対象となる

| Red Hat 系 | Debian 系 | その他 |
|---|---|---|
| Red Hat Enterprise Linux | Debian GNU/Linux | Slackware |
| Cent OS（RHEL互換） | Ubuntu（Desktopに注力） | openSUSE |
| Fedora（RHEL試作） | KNOPPIX（軽量） | Monta Vista(組込) |

*）他にも Oracle、SAPといったソフトウエア・ベンダーも Linux OSを提供している

# Linux特徴

- ソースコード（設計図）が公開されている
- 対応プラットフォーム（CPU等のHW環境）が多彩
- OSのカスタマイズが可能
- 無償配布（CentOS、Fedora、Debian、Ubuntu　etc.）
- 幅広い適用範囲（組込から大規模まで）
- 世界中の技術者が共同作業で開発（コミュニティ）
  - 「伽藍とバザール」（The Cathedral and the Bazaar）Eric S. Raymond
    http://www.catb.org/~esr/writings/cathedral-bazaar/

主な日本のLinuxとOSSのコミュニティ

| | |
|---|---|
| 日本Linux協会 | http://jla.linux.or.jp/ |
| The Linux Foundation | http://www.linuxfoundation.jp/ |
| Tokyo Linux Users Group | http://tlug.jp/ |
| オープンソースカンファレンス | http://www.ospn.jp/ |
| 日本OSS推進フォーラム | http://www.ossforum.jp/ |

- Linux に関する技術レベル認定試験（ Linux Professional Institute Certification）
- カナダのNPOであるLPIが実施する認定試験
  - 国際的な認定制度で世界共通のスキルを認定
  - ベンダーやディストリビュータに中立な内容
  - 世界最大規模の受験生を誇る
  - レベル1（基本な操作とシステム管理）
    レベル2（応用的システム管理、ネットワーク構築）
    レベル3（大規模システムの専門分野）
    上記、3つの階級がある

参照） http://www.lpi.or.jp/

# LPIC試験のレベル分け

- LPIC試験各レベルの概要と、その対象者を整理すると下記のように3段階＋αとなる。

ITSS*1 レベル

レベル3

Linux について基本的な知識があり、利用者（ユーザ）として操作ができる。
システム単体の基本的なシステム管理ができる。

LPIC3
**300: Mixed Environment**
- Linuxプロフェッショナル
- 大規模システム管理
- 分野別専門家

**303:Security**
- セキュリティ

**304: Virtualization & High Availability**
- クラウドコンピューティング

2014年4月 301/302試験は
300 試験に統合されました。

レベル2

LPIC2
- Linux上級エンジニア
- サーバのシステム管理（インターネットサーバ）
- 上級システム管理者

レベル1

LPIC1
- LPIC試験の第一関門
- 基本操作と基本的なシステム管理（スタンドアロンシステム）
- Linux ユーザ、初級管理者

*1） IT Skill Standard : ITスキル標準
経済産業省が示したIT関連スキルの習熟度
http://www.ipa.go.jp/

- レベル2、レベル3取得の前提条件＝登竜門
- 101試験、102試験両方合格が必要

101試験のテーマ

| ID | テーマ | 重 |
|---|---|---|
| 103.1 | コマンドラインで操作する | 4 |
| 103.3 | 基本的なファイル管理を行う | 4 |
| 103.4 | ストリーム、パイプ、リダイレクトを使う | 4 |
| 103.5 | プロセスを生成、監視、終了する | 4 |
| 101.2 | システムのブート | 3 |
| 101.3 | ランレベルの変更とシステムのシャットダウンまたはリブート | 3 |
| 102.4 | Debianパッケージ管理を使用する | 3 |
| 102.5 | RPMおよびYUMパッケージ管理を使用する | 3 |
| 103.2 | フィルタを使ってテキストストリームを処理する | 3 |
| 103.8 | viを使って基本的なファイル編集を行う | 3 |
| 104.3 | ファイルシステムのマウントとアンマウントをコントロールする | 3 |
| 104.5 | ファイルのパーミッションと所有者を管理する | 3 |
| 101.1 | ハードウェア設定の決定と構成 | 2 |
| 102.1 | ハードディスクのレイアウト設計 | 2 |
| 102.2 | ブートマネージャのインストール | 2 |
| 103.6 | プロセスの実行優先度を変更する | 2 |
| 103.7 | 正規表現を使用してテキストファイルを検索する | 2 |
| 104.1 | パーティションとファイルシステムの作成 | 2 |
| 104.2 | ファイルシステムの整合性を保持する | 2 |
| 104.6 | ハードリンクとシンボリックリンクを作成・変更する | 2 |
| 104.7 | システムファイルを見つける、適切な位置にファイルを配置する | 2 |
| 102.3 | 共有ライブラリを管理する | 1 |
| 104.4 | ディスククォータを管理する | 1 |

102試験のテーマ

| ID | テーマ | 重 |
|---|---|---|
| 107.1 | ユーザアカウント、グループアカウント、および関連するシステムファイルを管理する | 5 |
| 105.1 | シェル環境のカスタマイズと使用 | 4 |
| 105.2 | 簡単なスクリプトをカスタマイズまたは作成する | 4 |
| 107.2 | ジョブスケジューリングによるシステム管理業務の自動化 | 4 |
| 109.1 | インターネットプロトコルの基礎 | 4 |
| 109.2 | 基本的なネットワーク構成 | 4 |
| 109.3 | 基本的なネットワークの問題解決 | 4 |
| 107.3 | ローカライゼーションと国際化 | 3 |
| 108.1 | システム時刻を維持する | 3 |
| 108.3 | メール転送エージェント(MTA)の基本 | 3 |
| 110.1 | セキュリティ管理業務を実施する | 3 |
| 110.2 | ホストのセキュリティ設定 | 3 |
| 110.3 | 暗号化によるデータの保護 | 3 |
| 105.3 | SQLデータ管理 | 2 |
| 106.1 | X11のインストールと設定 | 2 |
| 106.2 | ディスプレイマネージャの設定 | 2 |
| 108.2 | システムのログ | 2 |
| 108.4 | プリンターと印刷を管理する | 2 |
| 109.4 | クライアント側のDNS設定 | 2 |
| 106.3 | アクセシビリティ | 1 |

| | | |
|---|---|---|
| 一般ユーザ操作 | 12 – 11 | 9 – 11 |
| システム管理 | | |

参照） http://www.lpi.or.jp/lpic1/range/

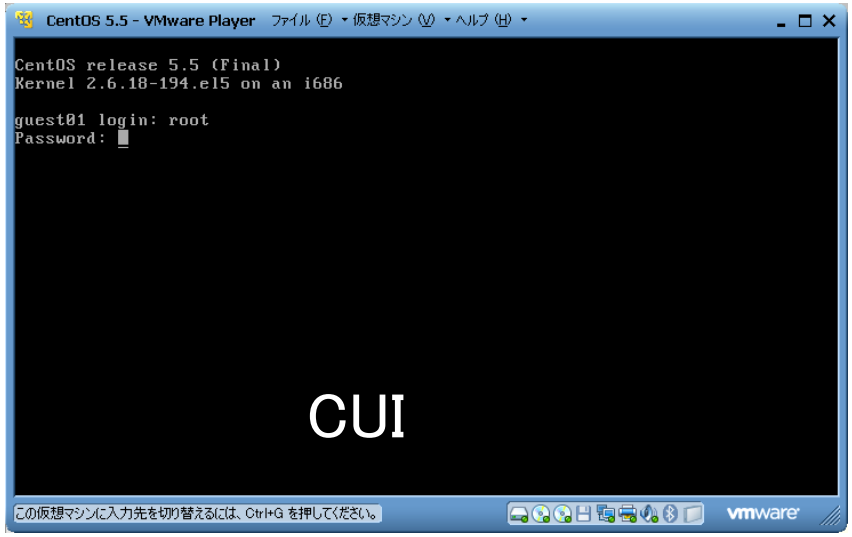



# Linux操作（基本）

# Linux の起動と利用開始

- Linux は、普通のPC同様、電源を起動すれば自動的に立ち上がる
- 利用開始はユーザ名とパスワードを入力（ログイン、認証＝本人確認）1つのシステムを複数ユーザで利用
→　マルチ・ユーザシステム

ログイン画面例




# ユーザ

- ログインするためには、システム管理者が事前にユーザを作成する必要がある
- ユーザは、一般ユーザとスーパーユーザに分かれ、後者はシステム管理者を指す
- システム管理者のユーザ名 はroot（ルート）という
- rootはシステム上で絶対的な権限を持つので、操作する時は細心の注意が必要（破壊的な操作を含め）

# CUI: Command line User Interface

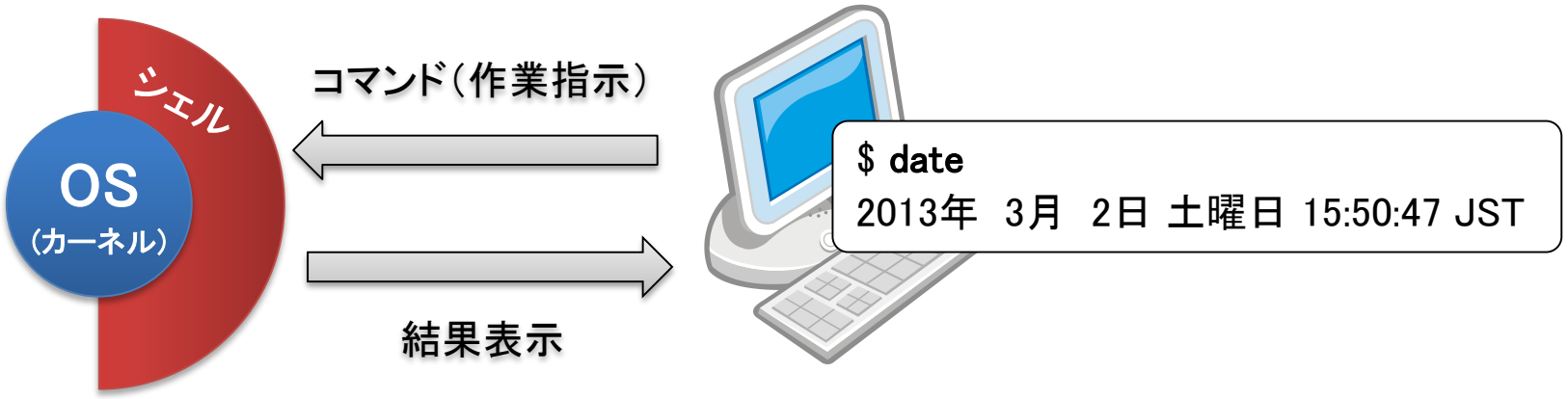


- キーボードから呪文(コマンド)を入力して操作
  →マウスは殆ど使わない
- 最初の単語をコマンド、空白に続く単語を引数
- 引数のうちハイフン(ー)で始まる物はオプション
- コマンドを台本のように記述し半自動的な操作が可能
  (シェル・スクリプト)
- コマンドはシェルによって処理される

```
$ コマンド 引数1 引数2 … [Enter]
$ コマンド ― オプション 引数1 引数2 …[Enter]
```

# シェル

- シェルはユーザのコマンドを解釈し、OSに作業を依頼し、結果を再びユーザに返す仲介者




- OSとユーザの間にある殻に見えるところから、シェルと呼ばれる
- Linuxでは複数のシェルから選択できる(LPIC は bashを採用)
- プログラミング言語としての機能も備える
- $ はプロンプト(入力促進記号)と呼び、シェルの入力待ちを表す
  一般ユーザのプロンプトは「$」で、スーパーユーザは「#」となる

# システムの終了

- 利用終了する場合は logout コマンドを入力
- システムを停止する場合は shutdown コマンド
  - # shutdown -r/-h 時間
  - オプション「 -r」はシステムの再起動 (reboot)
  - オプション「-h」はシステムの停止(halt)
  - 引数「時間」はシャットダウン実行時刻
    hh:mm 、+n分、 now など
- shutdown はスーパーユーザのみ操作可能

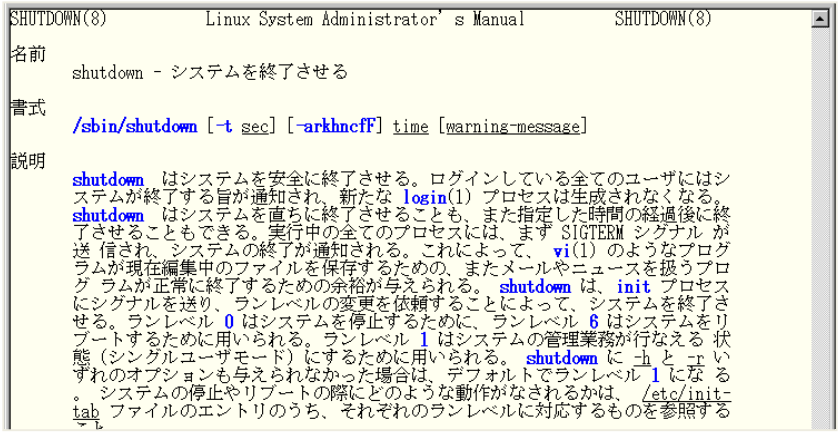
```
# shutdown -h 17:25
Broadcast message from student@localhost.localdomain
        (/dev/pts/0) at 13:58 ...

The system is going down for maintenance in 207 minutes!
```

＊)コマンドの書き方において、先頭の $ は一般ユーザ操作可、# はスーパユーザのみ限定

# オンラインマニュアル

- コマンドの詳細は、man コマンドで調べる
  - $　man　コマンド名
  - $　man　-k　キーワード(英文のみ)
  - man コマンドは、[空白]で次頁、[ u ] で前頁、[ q ]で終了
    $ ( **export LANG=C ; man　コマンド名** ) で英語版を表示
- Web検索も有効。「manpage」を加えて検索

**manpage　コマンド名/キーワード**　　**検索**



SHUTDOWN(8)　　Linux System Administrator's Manual　　SHUTDOWN(8)

名前

shutdown - システムを終了させる

書式

/sbin/shutdown [-t sec] [-arkhncfF] time [warning-message]

説明

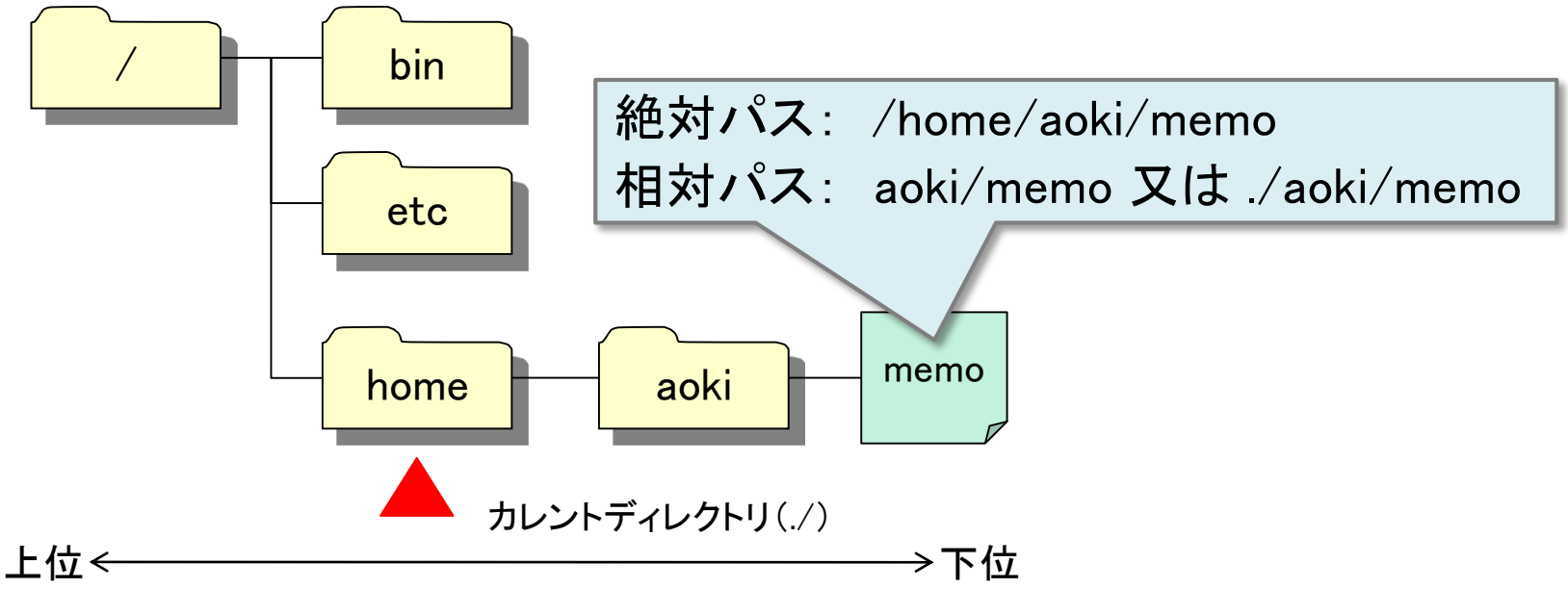
shutdown はシステムを安全に終了させる。ログインしている全てのユーザにはシステムが終了する旨が通知され、新たな login(1) プロセスは生成されなくなる。shutdown はシステムを直ちに終了させることも、また指定した時間の経過後に終了させることもできる。実行中の全てのプロセスには、まず SIGTERM シグナルが送信され、システムの終了が通知される。これによって、vi(1) のようなプログラムが現在編集中のファイルを保存するための、またメールやニュースを扱うプログラムが正常に終了するための余裕が与えられる。shutdown は、init プロセスにシグナルを送り、ランレベルの変更を依頼することによって、システムを終了させる。ランレベル 0 はシステムを停止するために、ランレベル 6 はシステムをリブートするために用いられる。ランレベル 1 はシステムの管理業務が行なえる状態（シングルユーザモード）にするために用いられる。shutdown に -h と -r いずれのオプションも与えられなかった場合は、デフォルトでランレベル 1 になる。システムの停止やリブートの際にどのような動作がなされるかは、/etc/inittab ファイルのエントリのうち、それぞれのランレベルに対応するものを参照する

# ファイルとディレクトリ

- Linuxでは全ての情報(入出力装置含む)をファイルという単位で保存する
- ファイルはディレクトリに収納して整理する
- 全てのファイルとディレクトリは／(ルートディレクトリ)を頂点とした階層構造に収納される
- ファイル(ディレクトリ)名には以下の制限
  - 英数字、下線( _ )、ハイフン( - )、ドット( . )が利用可能
  - 英文字の大文字・小文字は区別する
  - 255文字(Bytes)までの長さ

# パス名（ファイル名）

- 目的のファイルまでの経路にあるディレクトリをスラッシュ（／）で区切って表記
- 絶対パス：／（ルート）で始まるシステムで一意の名称
- 相対パス：現作業場所（カレントディレクトリ）からの名称。作業場所により、異なるファイルとなる事も



# ファイル操作コマンド(1)

- ディレクトリの移動　change directory
  - $ cd ［ 移動先パス名：省略時はログイン時の場所 ］*
  - 特殊なディレクトリ名
    現在地( . )、1つ上( .. )、ログイン時の場所( ~ )
- カレントディレクトリの表示　print working dire.
  - $ pwd
- ファイル一覧　list
  - $ ls［ ディレクトリ名：今いるディレクトリ ］
- ファイル内容の表示　concatenate
  - $ cat　　ファイル名
  - $ less　ファイル名　（画面単位に表示）

＊)コマンドの書き方において、[ ] は省略可能を表す。コロン( : )以降は省略された場合の規定値。

# ファイル操作コマンド(2)

- ファイルのコピー　Copy
  - $ cp　元ファイル　コピー先ファイル
  - $ cp　ファイル…　コピー先ディレクトリ/
  - コピー先ファイルは上書き注意
- ファイルの移動　Move
  - $ mv　元ファイル　移動先ファイル
  - $ mv　元ファイル…　移動先ディレクトリ/
  - コピー先ファイルは上書き注意
  - ディレクトリが同じであれば、ファイル名変更と同じ意味
- ファイルの削除　Remove
  - $ rm　ファイル…
  - 削除にあたって確認はしない

# ファイル操作コマンド(3)

- ディレクトリの作成 Make directory
  - $ mkdir ディレクトリ…
  - $ mkdir -p 親ディレクトリ/子ディレクトリ (Parents)
- ディレクトリのコピー (Recursive)
  - $ cp -r 元ディレクトリ コピー先ディレクトリ
  - コピー先ディレクトリ存在時は、その下に作成される
- ディレクトリの削除 Remove directory
  - $ rmdir ディレクトリ…
  - ディレクトリが空でない場合は、エラーとなる(安全)
  - $ rm -r ディレクトリ…
  - 指定したディレクトリ以下を、ごっそり削除(とても危険)



# ユーザと権限

# ユーザとグループ

- Linuxでは利用者をユーザとグループで識別し、ファイル等のアクセスを制限している
- ユーザ、グループとも背番号（UID、GID）が割り当てられLinuxはその番号で識別する（名前ではない）
- UID、GIDとも 0～ 4,294,967,294 の範囲
- 1つのユーザは最低ひとつ、最大65,536のグループに所属することができる
- ユーザー情報は /etc/passwd、グループ情報は /etc/group に記載

＊）上限値は Kernel ver 2.6.18 、CentOS 5.9 での実測値

# パスワード情報(/etc/passwd)

- パスワード情報は以下の形式で、各項目はコロン(:)で区切られている

(1)ユーザ名:(2)パスワード:(3)UID:(4)GID:(5)コメント:(6)ホームディレクトリ:(7)シェル

| # | 項目名 | 意味 |
|---|---|---|
| 1 | ユーザ名 | ログイン時に用いるユーザ名、システムで一意。ログイン名、アカウント名とも。 |
| 2 | パスワード | 古くは暗号化されたパスワードを指す、現在は X が固定値で、実際のパスワードは別ファイルに記録(/etc/shadow) |
| 3 | UID | ユーザに割り振った背番号、0 はスーパユーザ、2ケタはシステム用 |
| 4 | GID | 所属するグループの番号、主たる所属でありプライマリグループと呼ぶ |
| 5 | コメント | 正確にはGECOS領域。利用者の氏名、連絡先などを記録 |
| 6 | ホームディレクトリ | ログイン時に割り当てられる作業用ディレクトリ |
| 7 | シェル | ログイン後にコマンドを解釈するシェル名(原則的に /etc/shells から選択) |

# グループ情報(/etc/group)

- グループ情報は以下の形式で、各項目はコロン(:)で区切られている

(1)グループ名:(2)パスワード:(3)GID:(4)所属ユーザ…

| 項番 | 項目名 | 意味 |
|---|---|---|
| 1 | グループ名 | グループ名、システムで一意 |
| 2 | パスワード | 古くは暗号化されたプライマリグループ切換えパスワードを指す。現在は Xが固定値で、実際のパスワードは別ファイルに記録(/etc/gshadow) |
| 3 | GID | グループの番号 |
| 4 | 所属ユーザ | セカンダリグループとして所属するメンバー名。複数ある場合はカンマ(,)で区切って指定 |

# ユーザ操作コマンド(1)

- ユーザの作成　user add
  - # useradd ユーザ名
  - 指定しない限り、ユーザ名と同じグループも同時作成
- パスワード変更　pass word
  - $ passwd ［ ユーザ名：省略時は自分自身 ］
  - 他のユーザのパスワード変更は、スーパーユーザのみ可
- ユーザの削除　user delete
  - # userdel ［ -r ］ ユーザ名
  - rオプションでホームディレクトリも削除
- グループの作成　group add
  - # groupadd グループ名

# ユーザ操作コマンド(2)

- グループ削除 group delete
  - # groupdel グループ名
  - プライマリのメンバーが存在する場合はエラー
- 所属グループ表示
  - $ id [ ユーザ名:省略時は自分自身 ] identify
  - 指定ユーザのUID、GID、所属グループ表示
  - $ groups [ ユーザ名:自分自身 ]
  - 指定ユーザの所属グループ名表示
- ユーザの切り替え switch user
  - $ su [ - ] [ ユーザ名:root ]
  - 指定したユーザに変更、要パスワード
  - オプション - は、環境の初期化を行う

# ファイルの属性

- Linux のファイルにはいろいろな属性があり、ファイル一覧詳細( ls -l コマンド ) で確認できる

```
$ ls -l
 合計 52
 drwxrwxr-x 11 ycos apache    4096 12月 26 15:03 public_html
 -rw-rw-r--  1 ycos ycos      263 12月  6  2009 revoke.asc
 ①    ②     ③  ④    ⑤      ⑥        ⑦          ⑧
```

| # | 項目名 | 意味 |
|---|---|---|
| 1 | ファイル種別 | - : ファイル、d: ディレクトリ、l:シンボリックリンク　等 |
| 2 | モード | ファイルのアクセス許可（読み、書き、実行） |
| 3 | ノード数 | ファイルに付けられた名前の総数（1 : 本名、2～ : 別名） |
| 4 | 所有ユーザ | ファイル（ディレクトリ）の所有ユーザ |
| 5 | 所有グループ | ファイル（ディレクトリ）の所有グループ |
| 6 | ファイルサイズ | ファイルの大きさ(Bytes) |
| 7 | タイムスタンプ | 特に指定しない限り、最新更新日時（他にアクセス、状態変更） |
| 8 | ファイル名 | ファイル名称 |

# ファイルの所有者

- 情報の公開範囲を、ユーザ（所有者本人）とグループ（複数人）に分けて管理
- 所有者は操作を行ったユーザと、プライマリグループが用いられる
- 所有者の変更　　change owner
  - # chown ［ -R ］ ユーザ:グループ ファイル/ディレクトリ…
  - オプション -R はディレクトリ以下、全て同じ操作を適用
  - 「:グループ」でグループのみ、「ユーザ」でユーザのみ変更
  - スーパユーザ以外はユーザ変更不可
  - 所有グループの変更
    $ chgrp ［ -R ］ グループ ファイル/ディレクトリ…
    （一般ユーザは自分の所属グループへのみ変更可能）

# ファイルのパーミッション（権限）

- ファイルのアクセス権はパーミッションと呼ばれ読取 (Read)、書込(変更・削除) (Write)、実行(eXecute)の3種類
- 権限の対象は、所有ユーザ(User)、所有グループ(Group)、第三者(Other)の3者となる

| u | | | g | | | o | | |
|---|---|---|---|---|---|---|---|---|
| r | w | x | r | w | x | r | w | x |

例）

```
-rw-rw-r--  1 ycos ycos      263 12月  6  2009 revoke.asc
```

revoke.asc は、ユーザ ycos本人と、グループ ycos に所属するメンバーは、参照・変更可能で、第三者は参照のみ可能

# ファイルのパーミッション(権限)詳細

- パーミッションの詳細
  - ファイルとディレクトリでは挙動が若干違う
  - スーパーユーザは何でもできる(権限を無視できる)

| 権限 | ファイルに対して | ディレクトリに対して |
|---|---|---|
| R | 参照する事ができる(表示可能) | ファイル一覧取得ができる( ls 可 ) |
| W | 変更・削除・上書が出来る | 新しいファイルの追加、既存ファイルの削除ができる(ファイル一覧に影響する操作) |
| X | プログラムを実行できる | ディレクトリへの移動が可能 ( cd 可 ) |

例)
ディレクトリに書込み権(W)が無い場合、ファイル一覧に影響する操作(ファイルの追加、削除)はできない。

```
$ ls -ld tmp2
dr-xr-xr-x. 2 student student 4096  6月  7 13:55 2014 tmp2
$ ls -l tmp2
合計 4
-rw-rw-r--. 1 student student 43  6月  7 13:55 2014 data
$ date >> tmp2/data
$ rm tmp2/data
rm: cannot remove `tmp2/data': 許可がありません
```

# パーミッション変更(記号型)

- パーミッションの変更は chmod で、「誰に」どの「権限」を「どうする」かを指定する　change mode
  - $ chmod (誰)+/−(権限)　ファイル/ディレクトリ…
  - 誰に:
    u : ユーザ、 g: グループ、o: 第三者、 a: 全員(u+g+o)
  - +/− (どうする):
    + : 権限付与、 − : 権限剥奪
  - 権限:
    r : 読取、w : 書込、x : 実行

実行例:
グループの書込権と第三者の読込権利を剥奪する

```
$ ls -l a.txt
-rw-rw-r--. 1 student student 93  6月  7 13:50 2014 a.txt
$ chmod g-w,o-r a.txt
$ ls -l a.txt
-rw-r-----. 1 student student 93  6月  7 13:50 2014 a.txt
```

# パーミッション変更(数値型)

- パーミッションの各権限に数値(重み)をつけ、その値を指定することで変更する
  - $ chmod 数値 ファイル/ディレクトリ…
  - 権限付与する場合は、その数値を加算

| | u | | | g | | | o | | |
|---|---|---|---|---|---|---|---|---|---|
| 記号 | r | w | x | r | w | x | r | w | x |
| 数値 | 400 | 200 | 100 | 40 | 20 | 10 | 4 | 2 | 1 |

実行例:
パーミッションを rw-r----- にするには、400+200 + 40 + 0 = 640

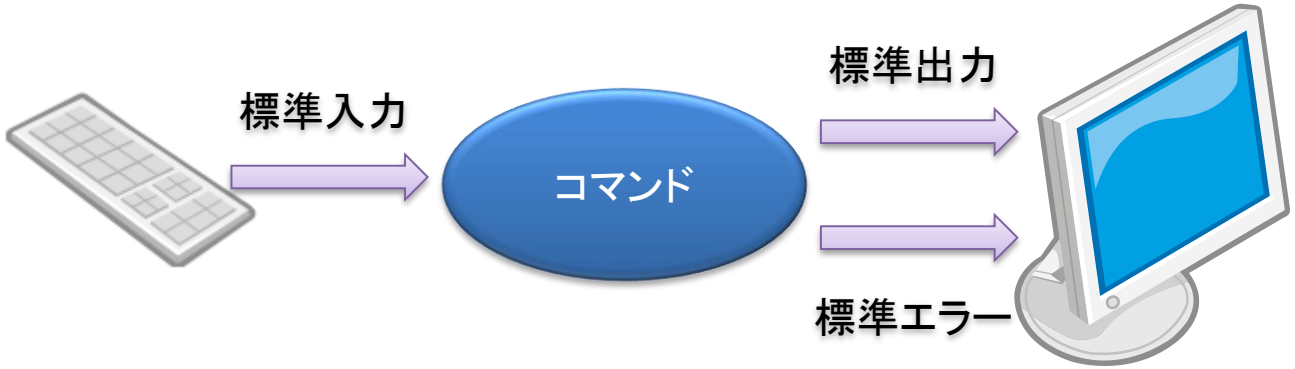
```
$ ls -l a.txt
-rw-rw-r--. 1 student student 93  6月  7 13:50 2014 a.txt
$ chmod 640 a.txt
$ ls -l a.txt
-rw-r-----. 1 student student 93  6月  7 13:50 2014 a.txt
```

記号型は現在の状況によって最終的なパーミッションは変化する。
数値型では指定したパーミッションに付け替える。



# Linux操作(応用)

- コマンドには予め入出力装置が割り当て済み
  - 0 *1.標準入力(stdin)
    データ入力、動作の制御(規定はキーボード)
  - 1.標準出力(stdout)
    処理結果表示(規定はディスプレイ)
  - 2.標準エラー*2出力(stderr)
    エラーメッセージ表示(規定はディスプレイ)




*1) 数値は「ファイルディスクリプタ」と呼ばれる入出力装置の連番
*2) 標準診断出力とも呼ばれる

# リダイレクション

- 標準入出力(エラー)はファイルや他の入出力装置へ切り替える事ができる
  → リダイレクションという
  - >標準出力の出力先を切り換える(上書き)
  - >>　　標準出力の出力先を切り換える(追加書込み)
  - <標準入力の入力元を切り換える
  - 2>　　標準エラー出力の出力先を切り換える

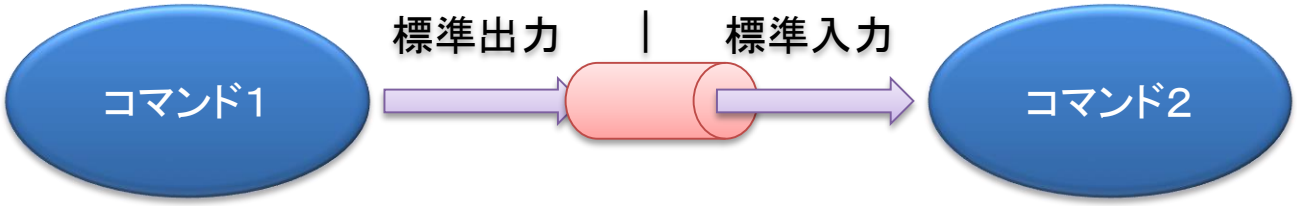
実行例:上から
- 標準出力
- 標準出力(追加)
- 標準エラー出力

```
$ date
2014年　6月　7日 土曜日 13:48:08 JST
$ date > date.out
$ date >> date.out
$ cat date.out
2014年　6月　7日 土曜日 13:48:10 JST
2014年　6月　7日 土曜日 13:48:13 JST
$ ls uso800
ls: cannot access uso800: そのようなファイルやディレクトリはありません
$ ls uso800 > ls.err
ls: cannot access uso800: そのようなファイルやディレクトリはありません
$ ls uso800 2> ls.err
```

**注意)** 出力先のファイルが存在する場合、その内容は上書きされる　→　失われる

# パイプライン

- 2つのコマンドの標準出力と入力を連結する事をパイプラインと呼ぶ
- 2つのコマンドを縦棒( | )でつなぐ




例）

ls でファイル詳細一覧、cat -n で行番号、headで最初の部分表示。

```
$ ls -l | cat -n | head
     1  合計 64
     2  drwxr-xr-x.   5 root root 4096  6月   7 13:14 2014 cache
     3  drwxr-xr-x.   3 root root 4096  6月   7 12:43 2014 db
     4  drwxr-xr-x.   3 root root 4096  6月   7 12:43 2014 empty
     5  drwxr-xr-x.   2 root root 4096  9月  23 20:47 2011 games
     6  drwxr-xr-x.  15 root root 4096  6月   7 13:14 2014 lib
     7  drwxr-xr-x.   2 root root 4096  9月  23 20:47 2011 local
     8  drwxrwxr-x.   4 root lock 4096  6月   7 12:43 2014 lock
     9  drwxr-xr-x.   4 root root 4096  6月   7 13:14 2014 log
    10  lrwxrwxrwx.   1 root root   10  6月   7 12:42 2014 mail -> spool/mail
```

この時、ls、cat、head の3つのコマンドは同時並行に実行される。

# シェル変数（1）

- シェルの動作を変更したり、任意の値を保存するための記憶領域をシェル変数と呼ぶ
  - $　変数名=値　　シェル変数の定義
  - $　echo　$変数名　　シェル変数の参照（名前の前に$）
    なお echo は引数を表示するコマンド
- シェル変数は他のプログラムに引き継がれないが、環境変数は引き継ぐ
  - $　export　シェル変数名
  - $　export　環境変数名=値
    …定義と環境変数宣言を同時に行う

シェル変数

```
$ name=yakoshi
$ echo $name
yakoshi
```

環境変数

```
$ echo $LANG
ja_JP.UTF-8
$ date
2014年　6月　7日 土曜日 13:41:06 JST
$ LANG=C
$ date
Sat Jun  7 13:41:12 JST 2014
```

# シェル変数(2)

- その他シェル変数にまつわるコマンド
  - $ set
    シェル変数、環境変数の一覧表示
  - $ env または printenv
    環境変数の一覧表示
  - $ unset 変数名
    変数の削除
- 環境変数 PATH
  - ディレクトリをコロン(:)で区切って指定
  - コマンドとは、この変数の先頭(左)から順に検索し最初に見つかったプログラムのこと

```
$ echo $PATH
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/studen
t/bin
$ su -
パスワード:
# echo $PATH
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin c
```

- 任意のファイル名を表す為の特殊な文字をメタキャラクタと呼び、主なものは以下の通り

| メタキャラクタ | 意味 | 例 | |
|---|---|---|---|
| * | 0個以上の任意の文字列 | A* | Abc, A123, A など’A’で始まる |
| ? | 任意の1文字 | A?? | Abc, A12, Aaa など‘A’で始まる3文字 |
| […] | […]内の文字のうち1文字 | [abc] | a, b, c のいずれか |
| ¥x | 続くメタキャラクタxの打ち消し | ¥? | ‘?’そのまま |

```
$ ls
a  a1  a11  a12  a2  b  b1  b11  b12  b2  c1  c11  c12  c2
$ ls a*
a  a1  a11  a12  a2
$ ls ??
a1  a2  b1  b2  c1  c2
$ ls [bc]?
b1  b2  c1  c2
```

メタキャラクタはワイルドカードとも呼ばれ、Linux以外のOSでも利用されますが、その機能や種類は異なります。

# クォーテーショニング（引用符）

- Linux では空白やメタキャラクタを含む値を引数にするために引用符を用いる。機能別に以下の3種類
  - シングル・クォーテーション（’）　[Shift] + [7]
    囲まれた内容を記載の通り、そのまま値とする
  - ダブル・クォーテーション（“）　[Shift] + [2]
    囲まれた内容のうち「変数をその値に置換（評価）」する
  - バック・クォーテーション（`）　[Shift] + [@]
    囲まれた内容をコマンドとして実行した結果と置換する
  - これらは組合せて（入れ子）使う事もできる

```
$ echo 'id $USER'
id $USER
$ echo "id $USER"
id student
$ echo `id $USER`
uid=500(student) gid=500(student) 所属グループ=500(student)
```

# シェルの便利な機能

- 履歴
  - $ history
    今までに実行したコマンドの履歴一覧
  - $ !n
    history で表示された n 番目のコマンドを再実行
    [↑][↓]キーでも過去のコマンド参照可能
- エイリアス
  - $ alias 別名='コマンド列'
    自分用にアレンジしたコマンド(別名)の定義

```
[student@localhost ~]$ history | tail -5
   24  history
   25  cal
   26  alias dir="ls -l | less"
   27  dir /etc
   28  history | tail -5
[student@localhost ~]$ !25
cal
      6月 2014
日 月 火 水 木 金 土
 1  2  3  4  5  6  7
```

```
$ alias dirs="ls -la | less"
$ dirs
合計 1220
drwxr-xr-x. 61 root root   4096  6月  7 13:16 2014 .
dr-xr-xr-x. 21 root root   4096  6月  7 12:45 2014 ..
                           :
```

# ファイル操作コマンド(4)

- ファイルを探す
  - $　locate　ファイル名(含むワイルドカード)
  - $　find　ディレクトリ　検索オプション(検索式、下表)

| 検索式 | 意味 |
|---|---|
| -name | ファイル名(メタキャラクタ含む) |
| -size | ファイルサイズ [Bytes]、 +n で n　以上 / -n で以下補助単位として k,M,G |
| -atime | 最終アクセス時刻が n 分前、ーで、n分以内。同様に更新時刻 -mtime、状態変更-ctime |
| -perm | パーミッション(数値)、+nnn で含む(+444 少なくとも全員読める) |
| -type | ファイル属性、f : ファイル、d: ディレクトリ　など |
| -a / -o | 条件の結合-、a は　and 、 -o は or |

例)
今いるディレクトリで、2MB以上の大きさのファイル、および変更が10日以内にあったファイル

```
[student@localhost ~]$ find . -size +2M
./LA-LPIC1-1.5-20140621-MSP.ppt
[student@localhost ~]$ find . -mtime -10 -exec ls -ld {} ¥;
drwx------. 2 student student 4096  6月  7 13:19 2014 .
-rw-------. 1 student student 33  6月  7 13:07 2014 ./.bash_history
```

ファイル名での検索では locate が高速ですが、事前に専用のデータベースをupdatedbで作成しておく必要があります。

# ファイル操作コマンド(5)

- ファイルの中を検索　global regular expression print
  - $ grep [ オプション ] 検索パターン [ ファイル… ]
  - 指定したファイル(省略時はstdin)から、指定されたパターンに一致した行を表示
  - 外に機能は限定されるがより高速な fgrep 、検索パターンがより厳密な egrep がある

| オプション | 意味 |
|---|---|
| -c | マッチした行数のみ表示 |
| -i | 英字の大文字・小文字を無視 |
| -l | マッチした行を含むファイル名のみ表示 |
| -v | マッチしなかった行を表示 |

```
$ grep www /etc/services
#       http://www.iana.org/assignments/port-numbers
http            80/tcp          www www-http    # WorldWideWeb HTTP
http            80/udp          www www-http    # HyperText Transfer
Protocol
www-ldap-gw     1760/tcp                        # www-ldap-gw
www-ldap-gw     1760/udp                        # www-ldap-gw
```

# 正規表現(検索パターン)

- grep での検索パターンはより複雑な組合せが可能で、正規表現(regular expression)と呼ぶ

| メタキャラクタ | 意味 |
|---|---|
| . (ドット) | 任意の1文字 |
| * | 直前の文字の0回以上の繰り返し |
| […] | […] 内のいずれか1文字 |
| [^…] | […] の文字以外 |
| ^ | 行の先頭 |
| $ | 行の末尾 |
| ¥x | メタキャラクタ x の打ち消し(そのまま) |

例)
先頭が#か空白で始まらない行を表示する。
シェルのメタキャラクタと混同されない様にダブルクォーテションで括っている
(この例ではシングルでも可)

```
$ grep "^[^#]"
/etc/httpd/conf/httpd.conf | head
ServerTokens Prod
ServerRoot "/etc/httpd"
PidFile run/httpd.pid
Timeout 1200
KeepAlive Off
      :
```

# ファイル操作コマンド(6)

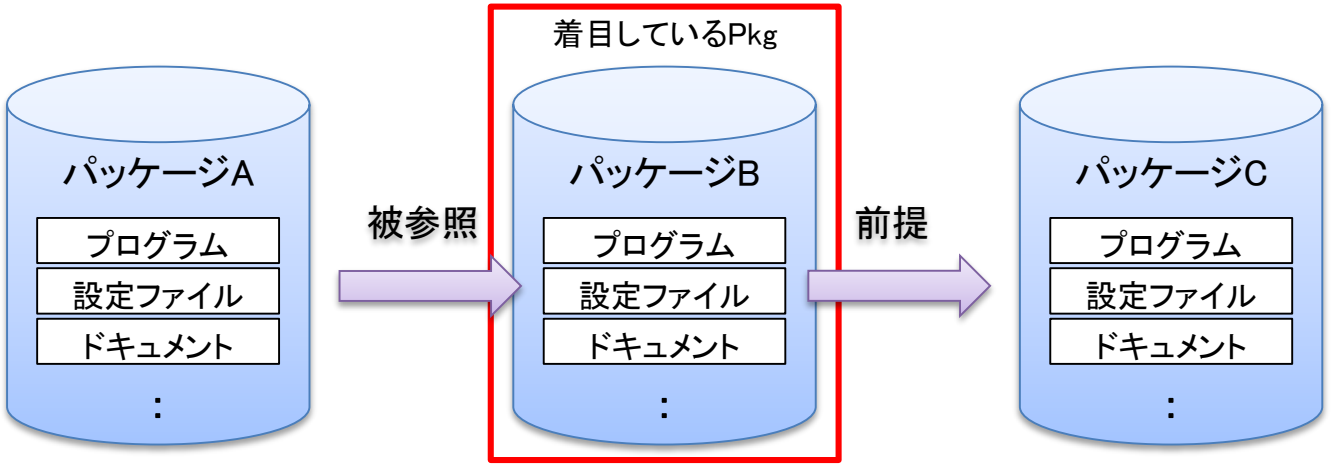
- ファイルの先頭部分を表示
  - $ head [ 一行数:省略時10 ] [ ファイル ]
- ファイルの末尾部分を表示
  - $ tail [ 一行数:省略時10 ] [ ファイル ]
- 行番号を付けてファイルを表示 number line
  - $ nl [ ファイル ]
  - cat -n もほぼ等価だが、nl は空行に番号を付けない
- 文字数を計数 word count
  - $ wc [ オプション ] [ ファイル ]
  - オプションは -l : 行数、-w : 単語、-c : バイト(文字)数を表示省略時はこの3つ全てを表示




# システム管理

# パッケージ

- パッケージとは、ソフトウエアのインストール単位で、実行プログラム、設定ファイル、関連ドキュメント等の集合
- パッケージ管理ツールは、インストール時に前提となるソフト、削除時には参照されているソフトの依存関係をチェックする
- パッケージ管理ツールは大きくRPM系（Red Hat, CentOS等）とDebian系(Debian/Linux、Ubuntu等)に分かれ、互換性はない
- 一般的に Debian系の方がパッケージ数が大きく、パッケージの単位は小さい（より細かい単位で管理）



# パッケージ管理コマンド

- RPM系
  - # rpm オプション [ RPMパッケージ名/ファイル ]
    ファイルからパッケージをインストール、削除などを行う
  - # yum コマンド [ RPMパッケージ名 ]
    ネットワークからパッケージをインストール、削除など行う

Redhat Package Manager
Yellowdog Updater Modified

- Debian系
  - # dpkg オプション [ debパッケージ名 / ファイル ]
    ファイルからパッケージをインストールしたり、削除したりする
  - # apt-get コマンド [ debパッケージ名 ]
    ネットワークからパッケージをインストール、削除など行う

Debian Packaging Tool
Advanced Packaging Tool

# RPM系コマンド

- rpm コマンドの主なオプション

| コマンド | 意味 |
|---|---|
| -ivh パッケージファイル | RPMファイルをインストール |
| -Uvh パッケージファイル | 同上　アップグレード（更新） |
| -e パッケージ名 | パッケージの削除 |
| -qa | インストール済みパッケージ一覧 |
| -qi パッケージ名 | パッケージの説明情報表示 |

- yum コマンドの主なオプション

| コマンド | 意味 |
|---|---|
| update ［ パッケージ名 ］ | パッケージのアップデート、パッケージ名を省略するとシステム全体 |
| install　パッケージ名 | パッケージのインストール |
| remove　パッケージ名 | パッケージの削除 |
| list | ネットで公開されているパッケージ一覧（未インストール含む） |
| info　パッケージ | パッケージの説明情報表示 |

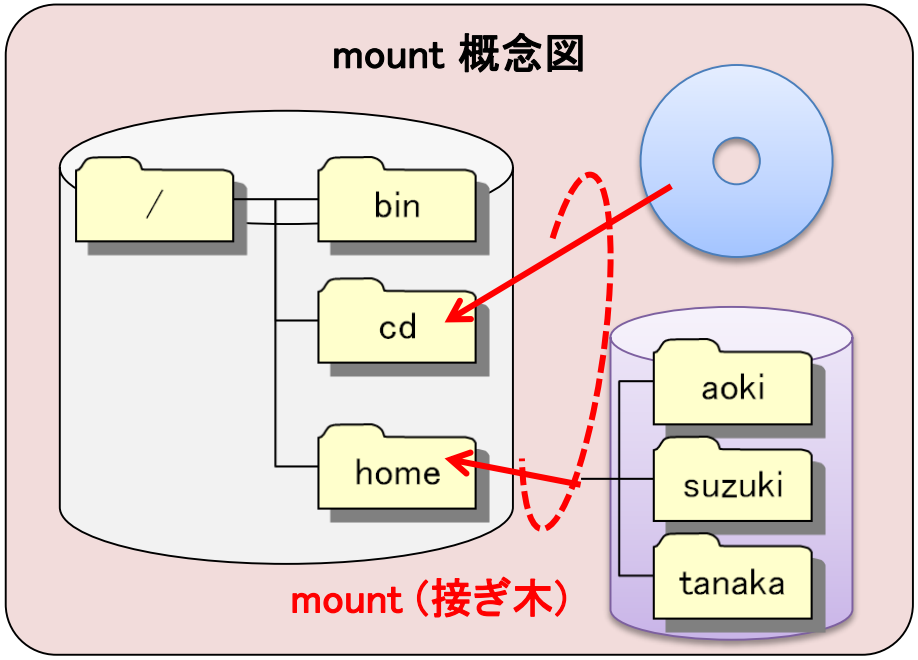
# ファイルシステム

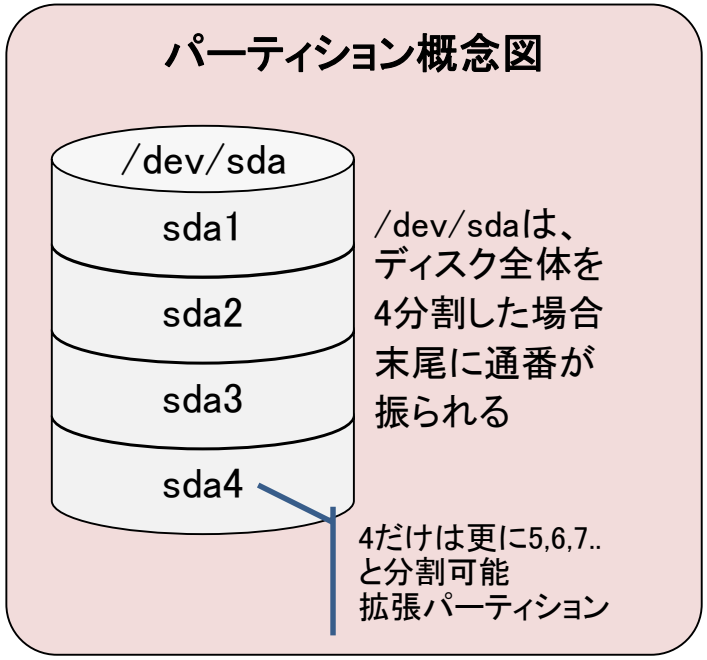
- データをHDDやUSBといった二次記憶装置にファイルとして格納する機構をファイルシステムと呼ぶ
  Linuxでは数多くのファイルシステムに対応

| FS | 最大容量* | 特徴 |
|---|---|---|
| ext3 | 32TiB | ext2の改良版、現在最も多くのディストリビューションで採用 |
| ext4 | 1EiB | ext3の後継ファイルシステム。RHEL 6 標準（↑） |
| Reiserfs | 16TiB | 高速で、小容量ファイルを多数扱う用途向き（↓） |
| XFS | 8EiB | 高速かつ堅牢（技術的に安定）、大容量ファイル向き。RHEL 7標準 |
| NTFS | 256 TiB | Windows のファイルシステム |
| FAT32 | 2TiB | Windows, USBやメモリーカード（デジカメ等）で利用 |
| UDF | 128 TiB | DVDのファイルシステム、ISO 9660 (CD)の拡張 |

*) TiB = $1024^3$Bytes, EiB = $1024^4$Bytes

- Linuxではハードディスクの実態を意識することなく、ひとつの階層構造に取り込む
  - 各ディスクの内容を既存の構造に接ぎ木する mount（Windowsはディスクごとにドライブ割当 C:¥ など）
  - 大容量ディスクは用途別に分割して利用（パーティション）



# ファイルシステム管理コマンド

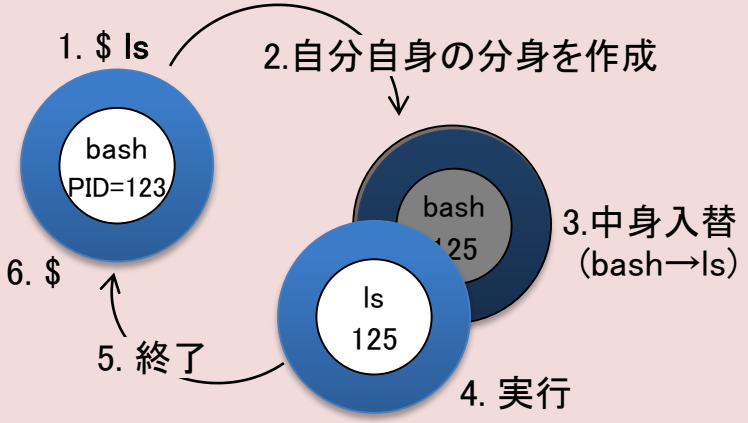
- 主なファイルシステムと管理用コマンド
  - # fdisk デバイスファイル(/dev/sdaなど) DOS FDISK
    HDD の領域を分割(パーティション)する
  - # newfs -t FSタイプ デバイスファイル(/dev/sda1など)
    ファイルシステム初期化(フォーマット)する new file system
  - # mount デバイスファイル ディレクトリ
    ディスクをディレクトリ(マウントポイント)に接続する

補足:主なファイルシステムと、管理コマンド

| | ext2/3 | ReiserFS | XFS |
|---|---|---|---|
| 生成(フォーマット) | mke2fs | mkriserfs | mkfs.xfs |
| 整合性のチェック | e3fsck | reiserfcsk | xfs_check |
| 属性・動作管理 | tune2fs | reiserfstune | xfs_admin |

- Linuxでの処理単位
  - ジョブ　：　入力した一連のコマンド群（[Enter] まで）
  - プロセス：　OSが扱う単位、メモリ・入出力装置の割当
    （LinuxはPID（ Process ID ）で管理）
  - スレッド　：　CPU(Core) が処理する単位
  - 処理は　ジョブ ≧　プロセス　≧　スレッド　という関係

**プロセス生成概念図**




プロセス生成手順:

1. コマンド ls とユーザが入力
2. bash が自身のコピー作成
3. 中身を ls と入れ替える
4. ls を実行
5. ls 終了
6. bash は再びプロンプトを表示
   コマンド入力待ち

# プロセス操作コマンド

- プロセス操作に係るコマンド
  - $ ps ［ オプション ］ process status
    - プロセス一覧、pstree はプロセスの親子関係を図示
    - オプションのハイフン(－)有無で動作が異なる注意
  - $ kill ［ -シグナル(数値または記号):15 ］ PID
    - プロセスに対し処理の指示(シグナル)を送る

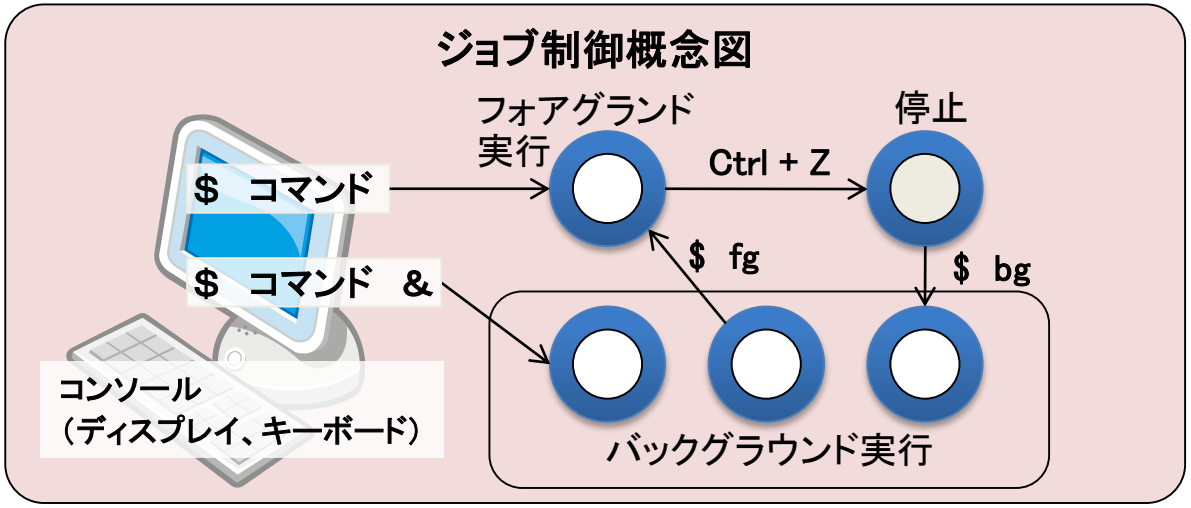
| シグナル | | 処理 |
|---|---|---|
| 1 | HUP | 回線切断による終了 |
| 2 | INT | ^C による中断 |
| 9 | KILL | 強制終了 |
| 15 | TERM | 終了(正常終了、省略値) |
| 18 | CONT | 中断している処理の再開 |
| 19 | STOP | 処理の一時中断 |

$ kill -l でシグナルの一覧表示

```
-bash-3.2$ pstree
init─┬─amavisd───2*[amavisd]
     ├─arpwatch
     ├─atd
     ├─auditd─┬─audispd───{audispd}
     │        └─{auditd}
     ├─clamd───{clamd}
     ├─crond
     ├─dbus-daemon───{dbus-daemon}
     ├─events/0
```

```
-bash-3.2$ csh
[student@ginza ~]$ echo $$
20657
[student@ginza ~]$ kill -9 20657
強制終了
-bash-3.2$
```

- ジョブ制御に関する用語
  - フォアグラウンドジョブ：　コンソールが直結するジョブ
  - バックグラウンドジョブ：　キーボードから切離されたジョブ
  - フォアグラウンドは、キーボードから制御
    強制終了は Ctrl + [ C ]、一時停止は Ctrl + [ Z ]
  - 起動時にバックグラウンド実行する場合は、末尾に　&　を付ける
  - バックグラウンドジョブは fg コマンドでフォアグランドへ移動





# ネットワーク

- ネットワークの基礎
  - サービスを提供するサーバと、要求するクライアントによる通信
  - 双方が異なるOS、ソフト、ハードでも通信可能（プロトコル）
  - ホスト名（www.yahoo.comなど）は世界で一意、実際の通信には電話番号に相当する IPアドレスが用いられる
  - サーバ名とIPアドレスの変換はDNSと呼ばれる電話帳サーバが行う
    IP Ver.4 と IP Ver.6 があり、アドレス表記は全く異なる。
  - インターネットはネットワークの集合で、経路はルータが決定

# ネットワークの設定（1）

- Linux ではテキストファイルでネットワーク設定を行う
  （代表的な設定ファイル例・共通）
  - resolv.conf
    名前解決（DNS＝インターネットの電話帳）設定
  - hosts
    名前解決（個別ファイル、必要最小限）
  - nsswitch.conf
    名前解決方法の検索順序

**/etc/resolv.conf**

```
search   uso800.com
nameserver 192.168.0.1
nameserver 10.20.30.40
```

**/etc/hosts**

```
127.0.0.1   localhost.localdomain   localhost
::1             localhost6.localdomain localhost6
```

**/etc/nsswitch.conf**

```
passwd:     files
hosts:      files   dns
```

# ネットワークの設定(2)

- Linux ではテキストファイルでネットワーク設定を行う
  (代表的な設定ファイル例・RHEL系)
  - network
    ネットワーク全般にかかわる設定
  - Ifcfg-NIC名
    ネットワークインタフェース個別設定(有線LAN、無線LANなど)

**/etc/sysconfig/network**

```
NETWORKING=yes
GATEWAY=192.168.0.1
HOSTNAME=www.uso800.com
```

**/etc/sysconfig/network-scripts/ifcfg-eth0**

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.0.7
NETMASK=255.255.255.0
ONBOOT=yes
BROADCAST=192.168.0.255
NETWORK=192.168.0.0
```

# ネットワークの設定(3)

- Linux ではテキストファイルでネットワーク設定を行う
  (代表的な設定ファイル例・Debian系)
  - hostname
    ホスト名の設定
  - interfaces
    ネットワークインタフェース設定(有線LAN、無線LANなど)

**/etc/hostname**

```
ww.uso800.com
```

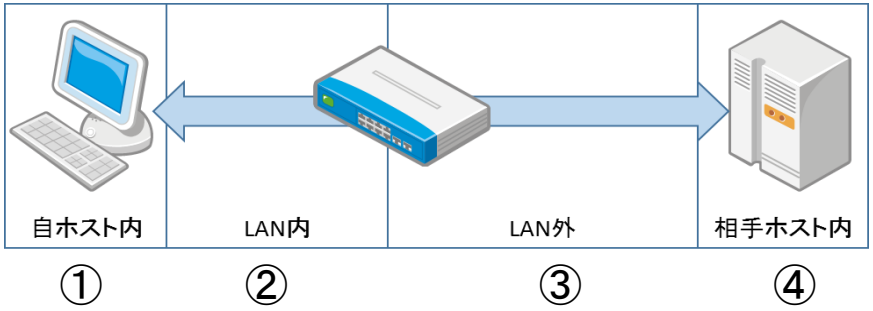
**/etc/network/interfaces**

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
   address  192.168.0.8
   netmask 255.255.255.0
   gateway 192.168.0.1
```

- 主なネットワーク関連のコマンド
  - $ ping ホスト名
    相手と接続確認(疎通確認)
  - $ nslookup ホスト名
    IPアドレスへの変換
  - $ traceroute ホスト名
    目的地までのルータ表示
  - $ netstat
    ネットワーク利用状況表示
  - # ifconfig NIC名 …
    NIC設定、内容表示
  - # route add/del …
    ルーティング設定
  - # nmap ホスト名
    解放ポート一覧

```
$ traceroute www.google.com
traceroute to www.google.com (173.194.126.242), 30 hops max, 40 byte packets
 1  gateway… (192.168.3.1)  0.578 ms  0.687 ms  0.773 ms
 2  softbankxxx.bbtec.net (221.xxx.xxx.xxx)  3.919 ms  3.882 ms  4.683 ms
 3  softbankxxx.bbtec.net (221.xxx.xxx.xxx)  4.180 ms  4.147 ms  4.121 ms
 4  * * *
 5  softbankxxx.bbtec.net (221.xxx.xxx.xxx)  4.402 ms  5.014 ms  4.964 ms
 6  209.85.241.90 (209.85.241.90)  5.706 ms  10.953 ms  10.905 ms
 7  72.14.232.109 (72.14.232.109)  5.944 ms  4.277 ms  4.319 ms
 8  nrt04s08-in-f18.1e100.net (173.194.126.242)  4.344 ms  4.220 ms  4.459 ms
```

- トラブル時の確認順序
  1. 自身の確認
     ケーブル接続、NIC起動確認
  2. LAN内の確認
     疎通確認、名前解決
  3. LAN外の確認
     デフォルトルータ、経路の確認
  4. アプリケーションの確認
     サーバプロセスの起動確認




デフォルトルータ未設定の事例

```
# ping 10.20.40.30
connect: Network is unreachable
# netstat -rn
Kernel IP routing table
Destination Gateway  Genmask         Flags   MSS Window…
10.33.121.0  0.0.0.0  255.255.255.0   U         0 0 …

# route add -net default gw 10.33.121.1
# netstat -rn
Kernel IP routing table
Destination Gateway  Genmask         Flags   MSS Window…
10.33.121.0  0.0.0.0  255.255.255.0   U         0 0 …
0.0.0.0   10.33.121.1   0.0.0.0        UG        0 0 …
```



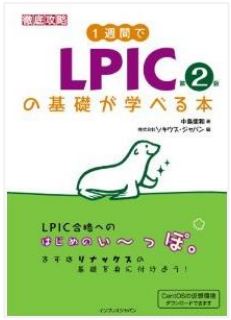
# その他

- 自宅でもできる実習環境の例
  - 仮想マシン
    PC上に仮想的なPCを作成、OSのインストールから体験できる。
    VMware：業界トップのシェア、無償版は再生のみ
    VirtualBox ：無償かつ多くのインストール済み環境も公開
    Hyper-V：Windows 8 から標準装備（Win7はWindows Virtual PC）
  - クラウドサービス
    時間貸し環境、最近は期間限定で無料のサイトも多い
    AWS：Amazon のサービス、1か月間は無料。業界トップシェア
    Microsoft Azure：MSのHyper-Vを利用したサービス、低コストに魅力
    これ以外にもVPSなどのコンピューティングサービスが沢山存在
  - ハードウエアへの導入
    別のハードウエアを用意しインストールする
    PC再生：　不要になったPCにLinuxを導入。HW前提に注意
    開発ボード：　HW勉強用の環境を流用。Raspberry Pi , Intel Galileoなど

# ご参考

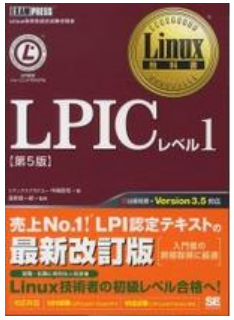


## 参考文献




**1週間でLPICの基礎が学べる本第2版**

中島　能和【著】 ソキウス・ジャパン【編】
インプレスジャパン
インプレスコミュニケーションズ〔発売〕
（2014/01 出版）
294p / 21cm / A5判
ISBN: 9784844335306




Linux教科書
**LPICレベル1（第5版）**

中島　能和【著】 濱野　賢一朗【監修】
翔泳社（2012/09 出版）
535p / 21cm / A5判
ISBN: 9784798127927


**LPI認定試験LPICレベル1《101／102》バージョン3.5対応**
**最短合格テキスト＆問題集**

中島　能和【監修】 橋本　智裕【著】
秀和システム（2014/08 出版）
735p / 21cm / A5判
ISBN: 9784798038728

## Linux 専門スクール


http://www.linuxacademy.ne.jp/

## ビジネス


http://www.wbiznet.co.jp/

http://ycos.sakura.ne.jp － 矢越サイト