™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
WORLD BIZNet 株式会社
2011年5月21日
矢越昭仁

UNIXの歴史

- 1969年、私的研究対象としてAT&Tベル研 Ken Thompson が基本ソフトUNIXを開発
- 1974年、AT&T がUNIX ver.5を外部に公開
- 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 Proj.)

- 1984年、AT&Tがコンピュータビジネスに参戦（UNIXライセンス問題発生）
- 1987年、X/Open が規格発表
- 1988年、IEEEがPOSIX発表、OSF発足

Linux　概要

- 1991年、ヘルシンキ大（当時）Linus Torvald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

Red Hat 系	Debian 系	その他
Red Hat Enterprise Linux	Debian GNU/Linux	Slackware
Cent OS (RHEL互換)	Ubuntu (NetBook)	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・ベンダーが 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
- 無償入手可能(CentOS、Fedora、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)

主な日本のLinuxコミュニティ

日本Linux協会	http://jla.linux.or.jp/
Linux.com JAPAN	http://jp.linux.com
The Linux Foundation	http://www.linuxfoundation.jp/
Tokyo Linux Users Group	http://tlug.jp/

LPIC

- 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・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）の3つの階級を持つ

参照） http://www.lpi.or.jp/

LPIC レベル1

オープンソース教育100年構想 リナックスアカデミー

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

- 101試験範囲

  Linuxの基本操作・知識
  - システムアーキテクチャ
  - インストールとパッケージ管理
  - GNUとUNIXのコマンド
  - デバイス、ファイルシステム、ファイルシステム階層標準

- 102試験範囲

  システム管理の基本
  - シェル、スクリプト、データ管理
  - ユーザインタフェースとデスクトップ
  - 管理業務
  - 重要なシステムサービス
  - ネットワークの基礎
  - セキュリティ

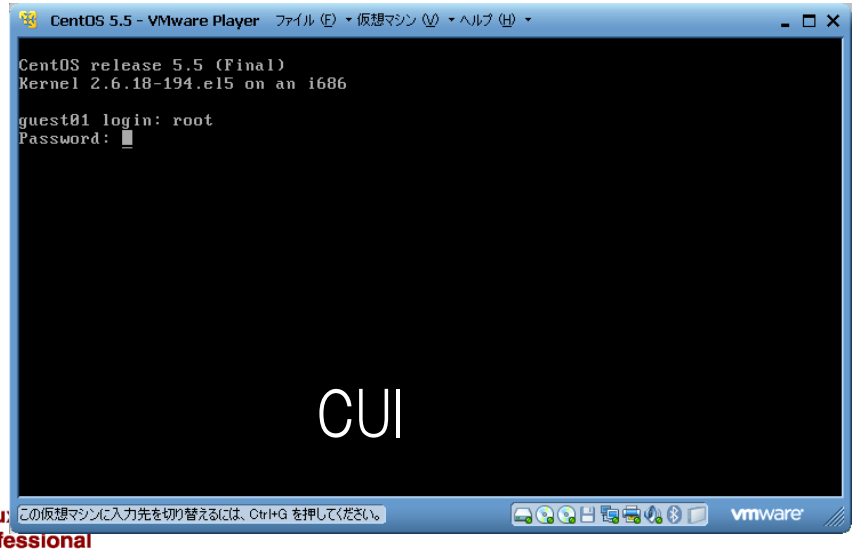
＊学習環境の準備については、別冊参照の事

Linux操作（基本）

Linux の起動から終了まで

- Linux は、普通のPC同様、電源を起動すれば自動的に立ち上がりますが、利用開始にあたってはユーザ名とパスワードを入力する必要があります（ログイン、認証）
  →1つのシステムを複数のユーザで利用する
  ＝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、ユーザ名をシステム管理者に依頼し事前に設定する必要がある
- 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）

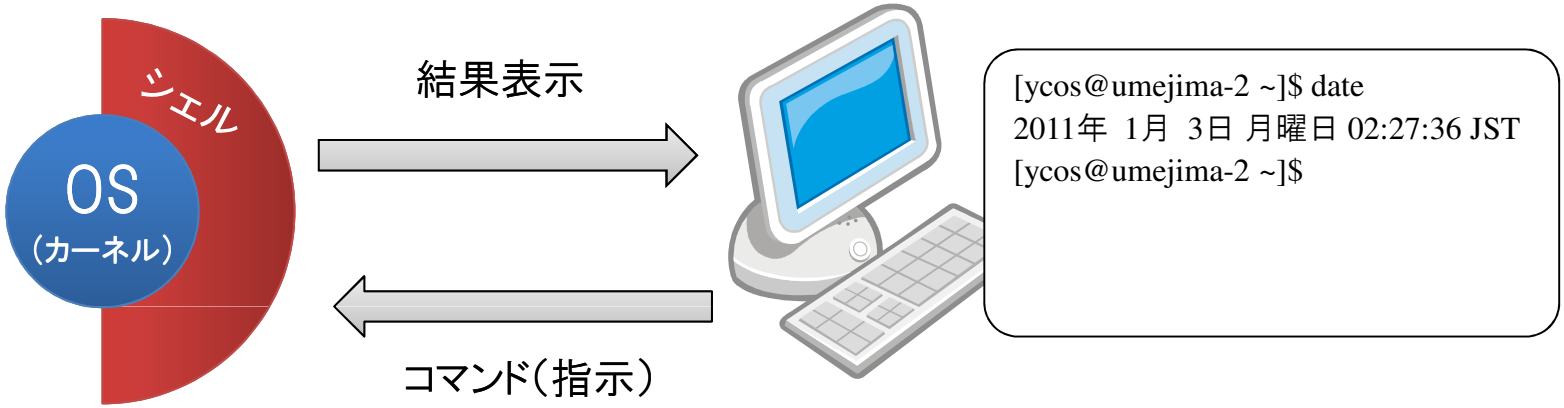
CUI: Command line User Interface

- キーボードから呪文(コマンド)を入力して操作する
→マウスは殆ど使わない
- 空白で区切った最初の単語をコマンド、続く単語を引数と呼ぶ引数のうちハイフン(ー)で始まる物をオプションと呼ぶ
- コマンドを台本のように書き連ねる事で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

- シェルはユーザからのコマンドを解釈し、OS作業を依頼し、その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では複数のシェルを選択することができる(LPIC は bash)
- 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、管理ユーザは # となる

終了に関するコマンド

- 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 や now など
- shutdown はスーパーユーザのみ操作可能

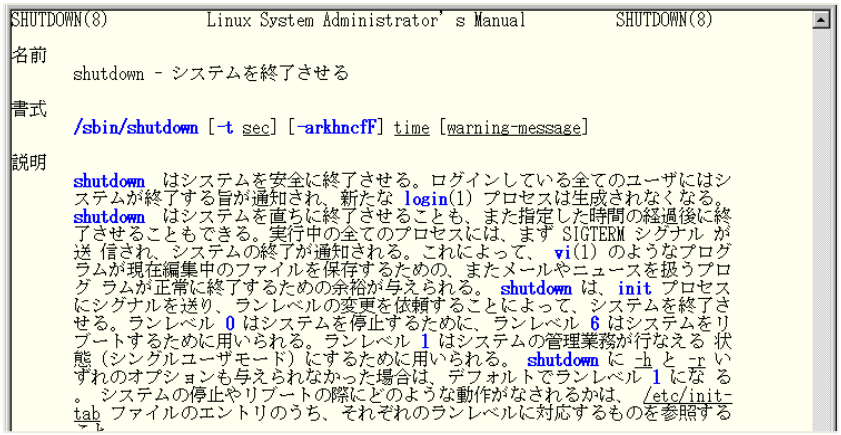
```
[root@umejima-2 ~]# shutdown -h 17:25
Broadcast message from root (pts/1) (Sun Jan  2 17:15:57 2011):
The system is going DOWN for system halt in 10 minutes!
```

＊)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

コマンドを調べる

- 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)
  - man コマンドは、[空白]で次頁、[u] で前頁、[q]で終了
- Web検索も有効。”manpage” を添えて検索

manpage コマンド名/キーワード　[検索]

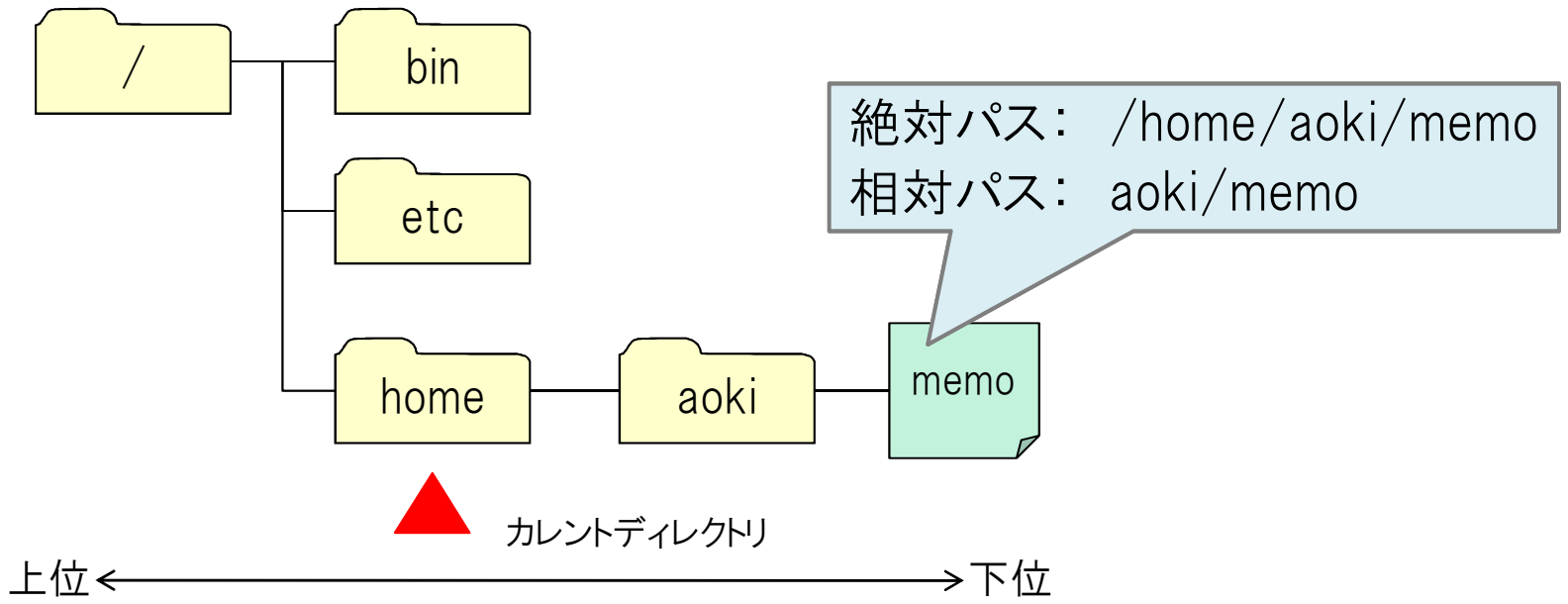


ファイルとディレクトリ

- 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(.)の文字が利用可能
  - 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) から始まるシステムで一意の名称
- 相対パス:　現在の作業場所(カレントディレクトリ)から始まる名称



ファイル操作コマンド(1)

- ディレクトリの移動　change directory
  - $ cd [移動先ディレクトリ(パス)名:省略時はログイン時の値] *
  - 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:省略時は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、[ ] は省略可能を表す先頭の $ は一般ユーザ利用可、# はスーパユーザのみ

ファイル操作コマンド(2)

オープンソース教育100年構想 リナックスアカデミー

- 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されるため、注意が必要
- 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されるため、注意が必要
  - 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…
  - 削除するにあたって特に確認はしない

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　ディレクトリ親/ディレクトリ子
- 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、そのディレクトリの下に作成される
- 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（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
- UID、GIDとも 0～ 4,294,967,294 の範囲
- 1つのユーザは最低1、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 に記載されている

＊）上限値は Kernel ver 2.6.x 、CentOS 5.5 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

項番	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
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。本名、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
  - 特に指定しない場合、ユーザ名と同じグループも合わせて作成
- 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
  - オプション –r でそのユーザの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
  - プライマリとしてメンバーが存在する場合はエラーとなる
- 所属グループ表示 identify
  - $ id [ユーザ名：省略時は自分自身]
  - 指定ユーザのUID、GID、所属グループ表示
  - $ groups [ユーザ名：省略時は自分自身]
  - 指定ユーザの所属グループ名表示
- ユーザの切り替え substitute user
  - $ su [-] [ユーザ名：省略時は root]
  - 指定したユーザへの切り替え（なりすまし） 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

```
[ycos@umejima-2 ~]$ 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

- 日記や営業報告のように、個人だけで留めておきたい情報と、チームで共有したい情報を、Linuxではユーザ（個人）とグループ（複数人）に分けて管理
- ファイル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…

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び、読取(Read)、書込(変更･削除) (Write)、実行 (eXecute)の3種類がある
- 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、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事が可能)
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)

上記ディレクトリに関する操作例:
(詳細は後述)

```
[ycos@umejima-2 ~]$ ls  -ld  tmp2
dr-xr-xr-x 2 ycos ycos 4096  1月  4 01:27 tmp2
[ycos@umejima-2 ~]$ ls  -l   tmp2
合計 0
-rw-rw-r-- 1 ycos ycos 0  1月  4 01:35 xxxxx
[ycos@umejima-2 ~]$ rm  tmp2/xxxxx
rm: cannot remove `tmp2/xxxxx': 許可がありません
[ycos@umejima-2 ~]$ date  >  tmp2/xxxxx
[ycos@umejima-2 ~]$ date  >  tmp2/yyyyy
-bash: tmp2/yyyyy: 許可がありません
[ycos@umejima-2 ~]$
```

パーミッション変更(シンボリック型)

- 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

```
[ycos@umejima-2 ~]$ ls -l a.txt
-rw-rw-r-- 1 ycos ycos 304 12月 26 15:54 a.txt
[ycos@umejima-2 ~]$ chmod  o-r,g-w  a.txt
[ycos@umejima-2 ~]$ ls -l a.txt
-rw-r----- 1 ycos ycos 304 12月 26 15:54 a.txt
[ycos@umejima-2 ~]$
```

実行例:
元のパーミッションから、第三者の読込権とグループの書込権を剥奪する

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

```
[ycos@umejima-2 ~]$ ls -l a.txt
-rw-r----- 1 ycos ycos 304 12月 26 15:54 a.txt
[ycos@umejima-2 ~]$ chmod  664  a.txt
[ycos@umejima-2 ~]$ ls -l a.txt
-rw-rw-r-- 1 ycos ycos 304 12月 26 15:54 a.txt
[ycos@umejima-2 ~]$
```

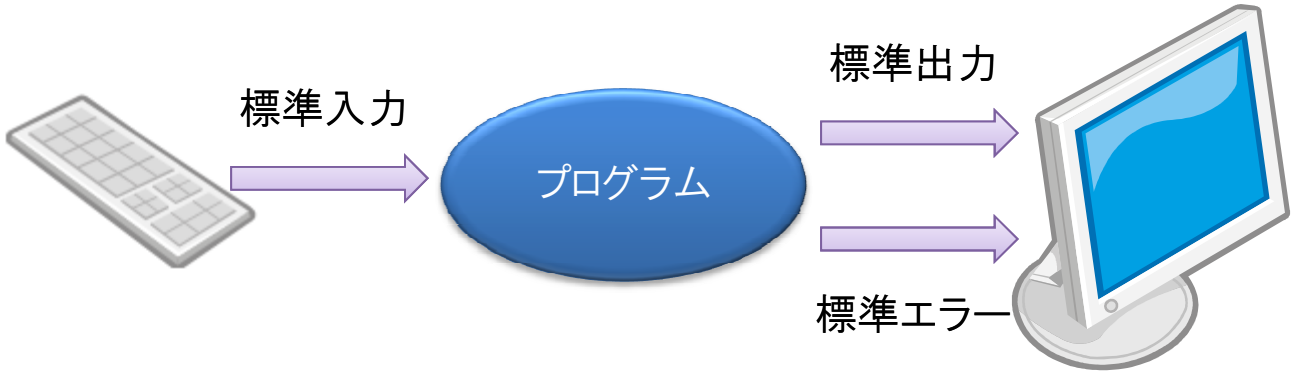
実行例:
パーミッションを rw-rw-r– にするには、
400+200　+　40+20　+　4　=　664

シンボリック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しますが、数値型では指定したパーミッションに付け替えます。

Linux操作（応用）

標準入出力

- プログラムには予め入出力装置が割り当て済み
  - 0.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*出力(stderr)
    エラーメッセージ表示(規定はディスプレイ)



*)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)はコマンド実行時に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

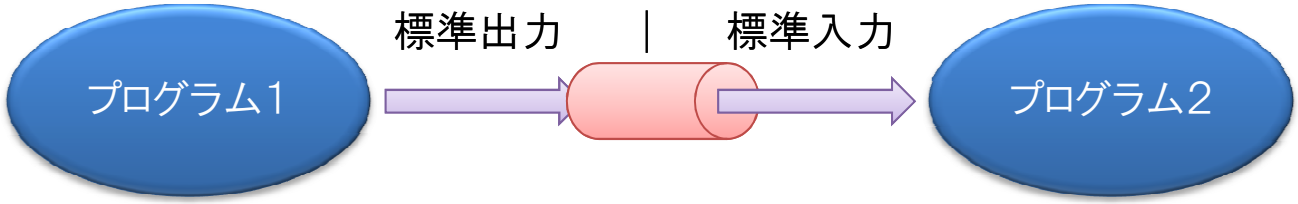
実行例:上から
- 標準出力
- 標準出力(追加)
- 標準エラー出力

注意)
標準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

```
[ycos@umejima-2 tmp2]$ date
2011年  1月  4日 火曜日 12:39:35 JST
[ycos@umejima-2 tmp2]$ date > date.out
[ycos@umejima-2 tmp2]$ date >> date.out
[ycos@umejima-2 tmp2]$ cat date.out
2011年  1月  4日 火曜日 12:39:39 JST
2011年  1月  4日 火曜日 12:39:53 JST
[ycos@umejima-2 tmp2]$ ls  uso800
ls: uso800: そのようなファイルやディレクトリはありません
[ycos@umejima-2 tmp2]$ ls  uso800 >  ls.out
ls: uso800: そのようなファイルやディレクトリはありません
[ycos@umejima-2 tmp2]$ ls  uso800 2>  ls.out
[ycos@umejima-2 tmp2]$
```

パイプライン

- 二つのプログラムの標準出力と標準入力を連結し絞り込む事をパイプラインと呼ぶ
  - 2つのコマンドの間に縦棒(|)を加える




```
[ycos@umejima-2 tmp2]$ ls -l | cat -n
     1  合計 8
     2  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 a
     3  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 a1
     4  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 a11
     5  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 a12
     6  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 a2
     7  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 b
     8  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 b1
     9  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 b11
    10  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 b12
    11  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 b2
    12  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 c
    13  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 c1
    14  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 c11
    15  -rw-rw-r-- 1 ycos ycos  0  1月  4 10:07 c12
```

例）
ls　コマンドはファイルの詳細一覧、
cat は標準入力をそのまま、出力へ
　ただし n オプションは行番号を付与

```
$ ls –l  > ls.out
$ cat –n  ls.out
```

と等価だが、パイプラインでは ls も cat も同時並行で動作する

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

```
[ycos@umejima-2 tmp2]$ name=yakoshi
[ycos@umejima-2 tmp2]$ echo $name
yakoshi
[ycos@umejima-2 tmp2]$ echo Hello, $name-san
Hello, yakoshi-san
[ycos@umejima-2 tmp2]$
```

環境変数

```
[ycos@umejima-2 tmp2]$ echo $LANG
ja_JP.UTF-8
[ycos@umejima-2 tmp2]$ date
2011年　1月　4日 火曜日 13:57:10 JST
[ycos@umejima-2 tmp2]$ LANG=en_US
[ycos@umejima-2 tmp2]$ date
Tue Jan  4 13:57:17 JST 2011
[ycos@umejima-2 tmp2]$
```

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)から順に検索した結果最初に見つかったプログラムのこと

```
[userz@umejima-2 ~]$ echo $PATH
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/userz/bin
[userz@umejima-2 ~]$ su -
パスワード:
[root@umejima-2 ~]# echo $PATH
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/
usr/bin:/root/bin
[root@umejima-2 ~]#
```

メタキャラクタ

- 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、シェルは以下のようなメタキャラクタを持つ

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'で始まり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

```
[ycos@umejima-2 tmp2]$ ls
a  a1  a11  a12  a2  b  b1  b11  b12  b2  c  c1  c11  c12  c2
[ycos@umejima-2 tmp2]$ ls a*
a  a1  a11  a12  a2
[ycos@umejima-2 tmp2]$ ls ??
a1  a2  b1  b2  c1  c2
[ycos@umejima-2 tmp2]$ ls [bc]??
b11  b12  c11  c12
[ycos@umejima-2 tmp2]$
```

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(') キーボード [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(") キーボード [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(`) キーボード [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

```
[ycos@umejima-2 ~]$ echo 'id $USER'
id $USER
[ycos@umejima-2 ~]$ echo "id $USER"
id ycos
[ycos@umejima-2 ~]$ echo `id $USER`
uid=500(ycos) gid=500(ycos) 所属グループ=500(ycos),0(root),1(bin),2(daemon),3(sys),4(adm),
6(disk),7(lp),10(wheel),12(mail),50(ftp),48(apache),25(named),501(student),502(wbuser),89(
postfix)
[ycos@umejima-2 ~]$
```

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
- 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

```
[ycos@umejima-2 tmp2]$ history  |  tail -5
 1017  cd tmp2
 1018  date
 1019  cal
 1020  echo $PATH
 1021  history  |  tail -5
[ycos@umejima-2 tmp2]$ !1019
cal
     1月 2011
日 月 火 水 木 金 土
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
```

```
[ycos@umejima-2 tmp2]$ alias dir='ls -l | less'
[ycos@umejima-2 tmp2]$ dir /etc
合計 6860
drwxr-xr-x 116 root root    12288  1月  3 19:46 ./
drwxr-xr-x  28 root root     4096 12月 22 18:04 ../
drwxr-xr-x   3 root root     4096  9月 14  2009 .java/
-rw-------   1 root root        0  6月  8  2009 .pwd.lock
-rw-r--r--   1 root root     2518  3月  1  2010 DIR_COLORS
-rw-r--r--   1 root root     2420  3月  1  2010 DIR_COLORS.xterm
-rw-r--r--   1 root root    92794  6月  4  2007 Muttrc
-rw-r--r--   1 root root        0  6月  4  2007 Muttrc.local
drwxr-xr-x   4 root root     4096 12月 25 19:46 NetworkManager/
drwxr-xr-x   8 root root     4096 11月 18 16:19 X11/
-rw-r--r--   1 root root     2562  5月 25  2008 a2ps-site.cfg
```

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、ーを指定すると、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

```
[ycos@umejima-2 tmp]$ find . -size +10M
./ImageMagick-6.6.0-10.tar.gz
./access.log
[ycos@umejima-2 tmp]$ find . -atime -10
.
./access.log
[ycos@umejima-2 tmp]$
```

例)
今いるディレクトリで、
1. 10MB以上の大きさのファイル
2. アクセスが10分以内にあったファイル

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

```
[ycos@umejima-2 tmp]$ 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
www-dev         2784/tcp                        # world wide web - development
www-dev         2784/udp                        # world wide web - development
```

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
.	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)

```
[userz@umejima-2 ~]$ grep "^[^# ]" /etc/httpd/conf/httpd.conf
ServerTokens ProductOnly
ServerRoot "/etc/httpd"
PidFile run/httpd.pid
Timeout 120
KeepAlive Off
MaxKeepAliveRequests 100
KeepAliveTimeout 15
<IfModule prefork.c>
StartServers       8
```

例)
先頭が#か空白で始まらない行表示
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る
(この例ではシングルでも可)

ファイル操作コマンド(６)

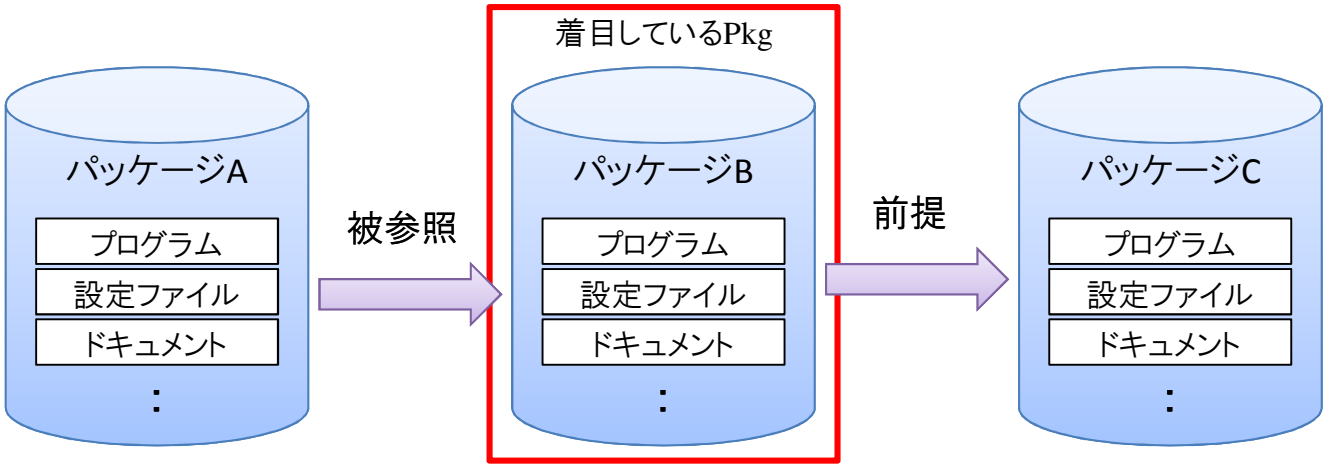
オープンソース教育100年構想 リナックスアカデミー

- 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、関連ドキュメントなどから構成される
- 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など）とDebian系(Debian/Linux、Ubuntuなど)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したり、削除したりする
  - # yum [コマンド] [RPMパッケージ名]
    パッケージのインストールと削除。必要であればネットからダウンロードする

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]
    パッケージのインストールと削除。必要であればネットからダウンロードする

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

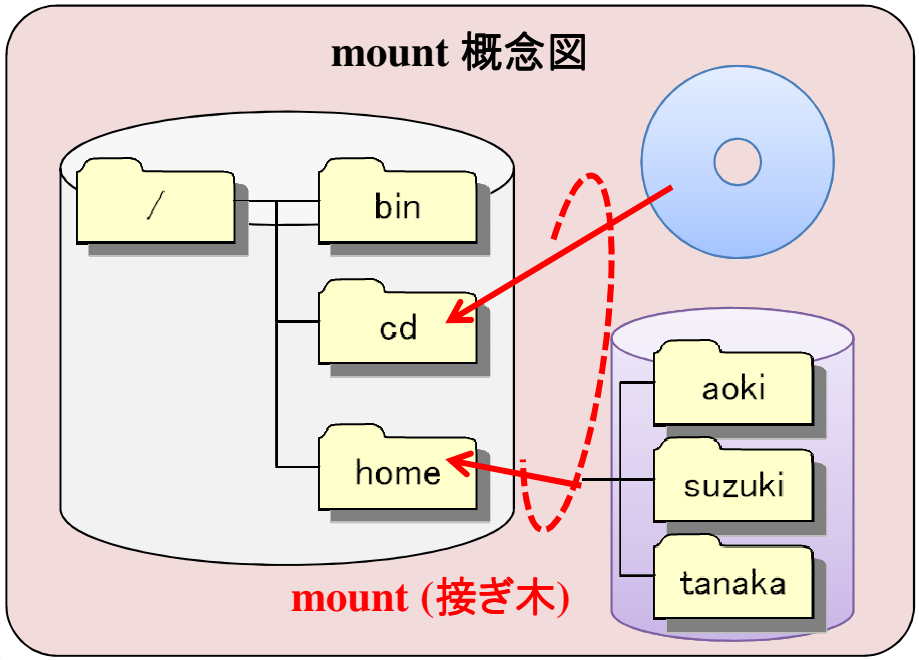
- データをHDDやUSBといった二次記憶装置にファイルとして格納する機構をファイルシステムと呼ぶ
  Linuxでは数多くのファイルシステムをサポートしている

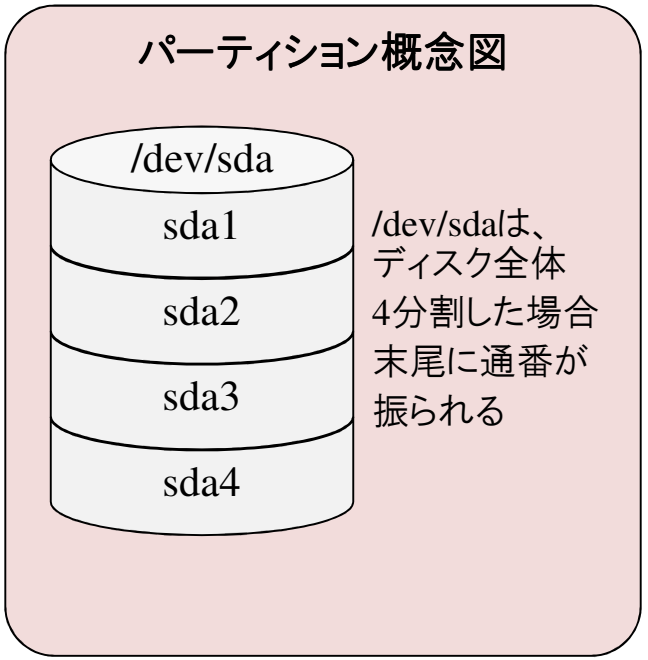
FS	最大容量*	特徴
ext3	32TiB	**ext2**の改良版、現在最も多くのディストリビューションで採用
ext4	1EiB	ext3の後継ファイルシステム
Reiserfs	16TiB	高速で、小容量ファイルを多数扱う用途向き
XFS	8EiB	高速かつ堅牢（技術的に安定）、大容量ファイル向き
NTFS	256 TiB	Windows のファイルシステム
FAT32	2TiB	USBやメモリーカード（デジカメ等）で利用
UDF	128 TiB	DVDのファイルシステム、ISO 9660 (CD)の拡張

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

ディスクとファイルシステム

- 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 などは、ディスクをドライブ毎に分割して管理
  - 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）





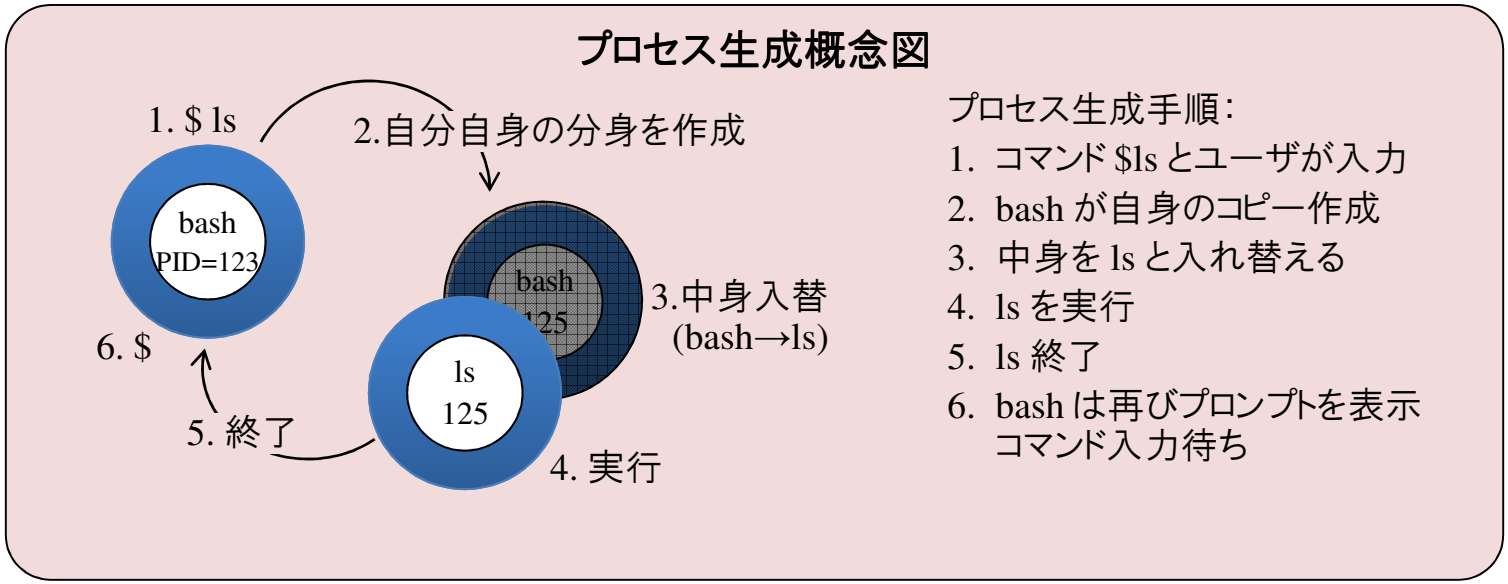
ファイルシステム管理コマンド

- 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
  - 大きさは　ジョブ ≧　プロセス　≧　スレッド　という関係



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]
    プロセス一覧表示、pstree はプロセスの親子関係を図で表示
  - $ kill [-シグナル（数値または記号）] PID
    プロセスに対し処理の指示（シグナル）を出す

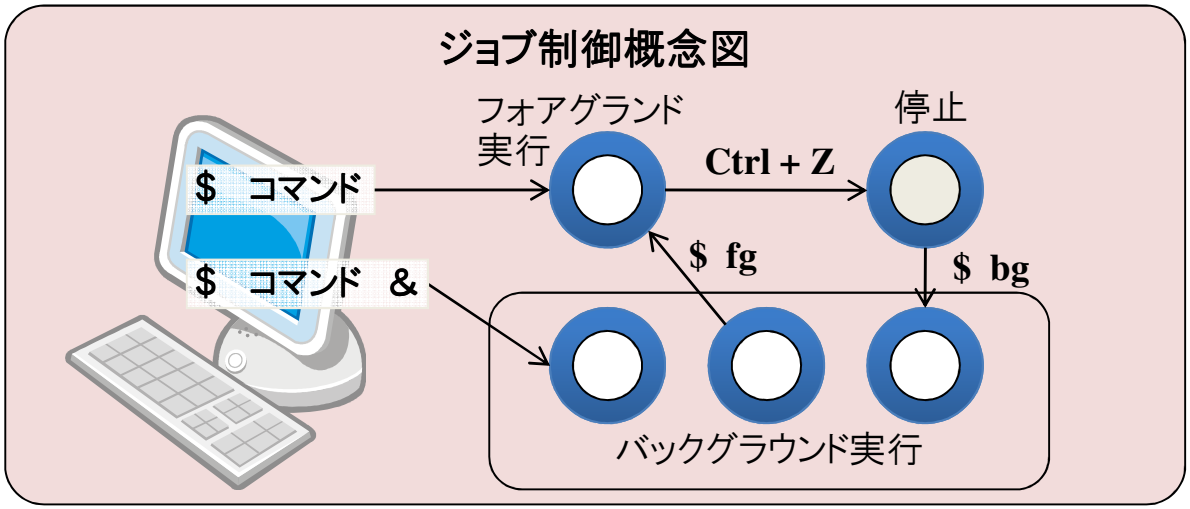
process status

シグナル		処理
9	KILL	強制終了
15	TERM	終了（正常終了、後処理後終了）
18	CONT	中断している処理の再開
19	STOP	処理の一時中断

```
[ycos@umejima-2 LAPC]$ bash
[ycos@umejima-2 LAPC]$ echo $$
13553
[ycos@umejima-2 LAPC]$ kill -9 13553
強制終了
[ycos@umejima-2 LAPC]$
```

```
[ycos@umejima-2 LAPC]$ pstree
init─┬─acpid
     ├─atd
     ├─auditd─┬─audispd───{audispd}
     │        └─{auditd}
     ├─blktapctrl───{blktapctrl}
     ├─crond
     ├─dbus-daemon
     ├─dnsmasq
     ├─events/0
     ├─events/1
     ├─gam_server
     ├─gpm
     ├─hald───hald-runner─┬─hald-addon-acpi
     │                    ├─hald-addon-keyb
     │                    └─5*[hald-addon-stor]
     ├─hidd
     ├─hpiod
     ├─httpd───8*[httpd]
     ├─irqbalance
     ├─khelper
```

- ジョブ制御に関する用語
  - フォアグラウンドジョブ： 画面（キーボード、ディスプレイ）から直接操作しているジョブ
  - バックグラウンドジョブ： キーボードから直接操作できないジョブ
  - フォアグラウンドでは、強制終了は Ctrl + [C]、一時停止は Ctrl + [Z]
    一時停止中のジョブは bg コマンドでバックグラウンド化可能
  - 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（プロトコルによる）
  - 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
  - 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

- 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）
  - network
    ネットワーク全般にかかわる設定
  - Ifcfg-*
    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）
  - resolv.conf
    名前解決（DNS＝インターネットの電話帳）設定

/etc/sysconfig/network

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

/etc/resolv.conf

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

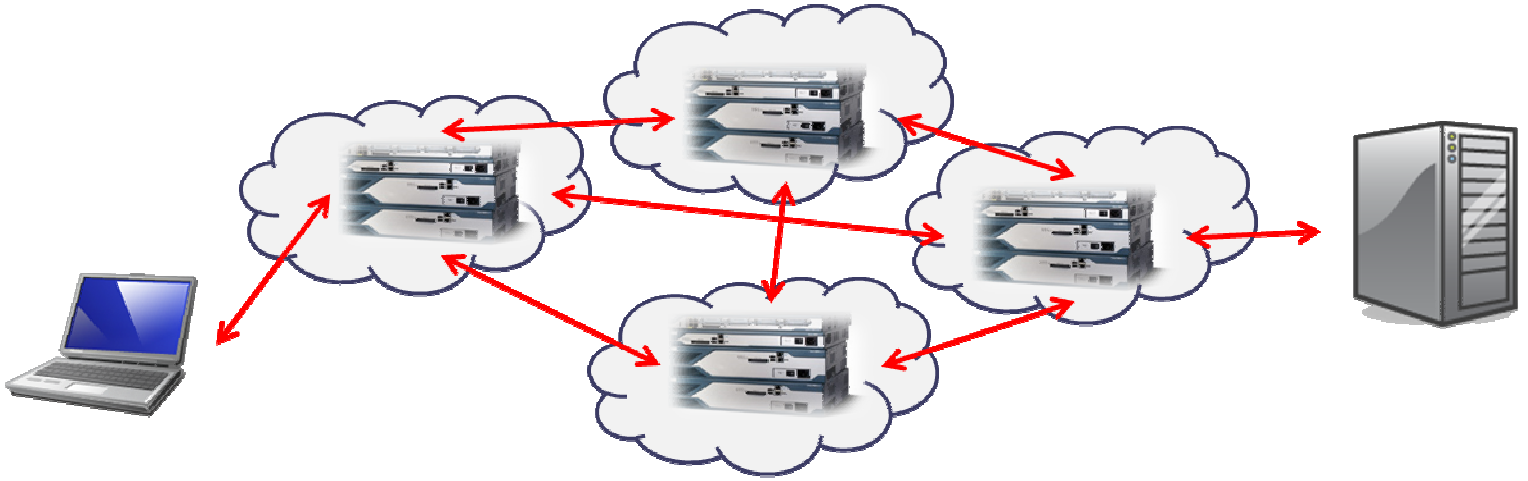
/etc/sysconfig/network-scripts/ifcfg-eth0

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

ネットワークのコマンド

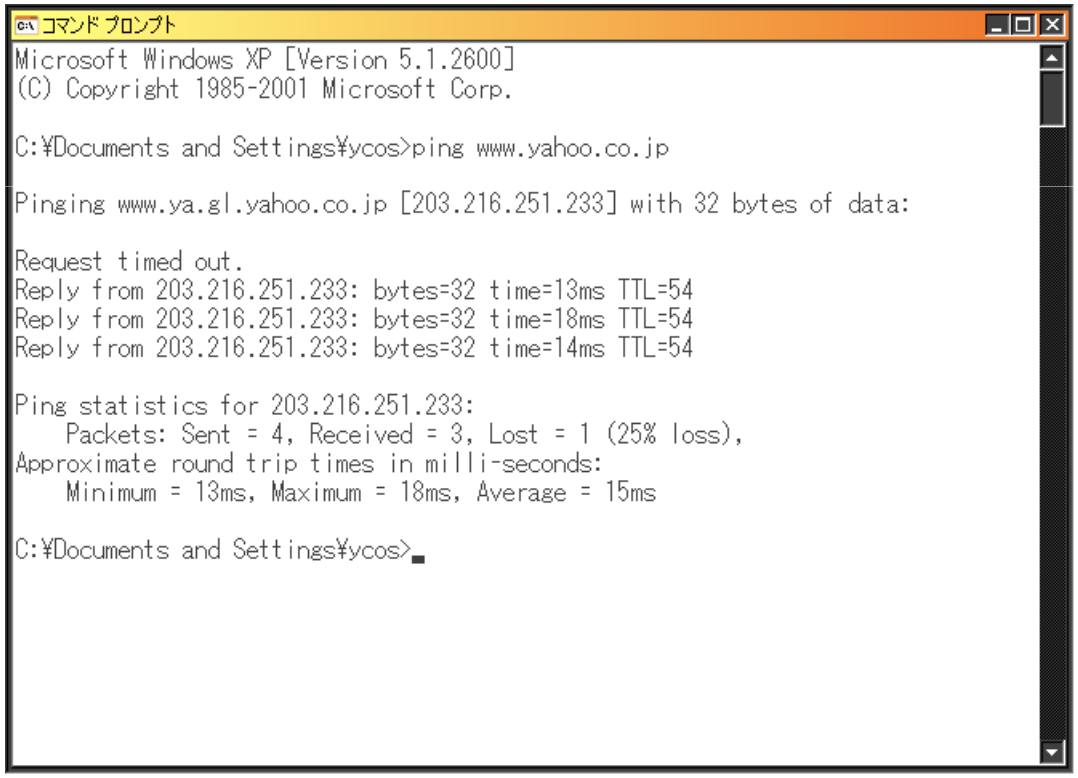
- 主なネットワーク関連のコマンド
  - $ ping ホスト名
    相手と接続できるかどうか、疎通確認
  - $ nslookup ホスト名
    IPアドレスへの変換
  - $ traceroute ホスト名
    目的地までのルータ表示
  - $ netstat
    ネットワーク接続状況表示

Windows でも同等のコマンドがあります。
「ファイル名を指定して実行...」から、cmd を起動し先と同じコマンドが利用できます。
（traceroute のみ tracert ）



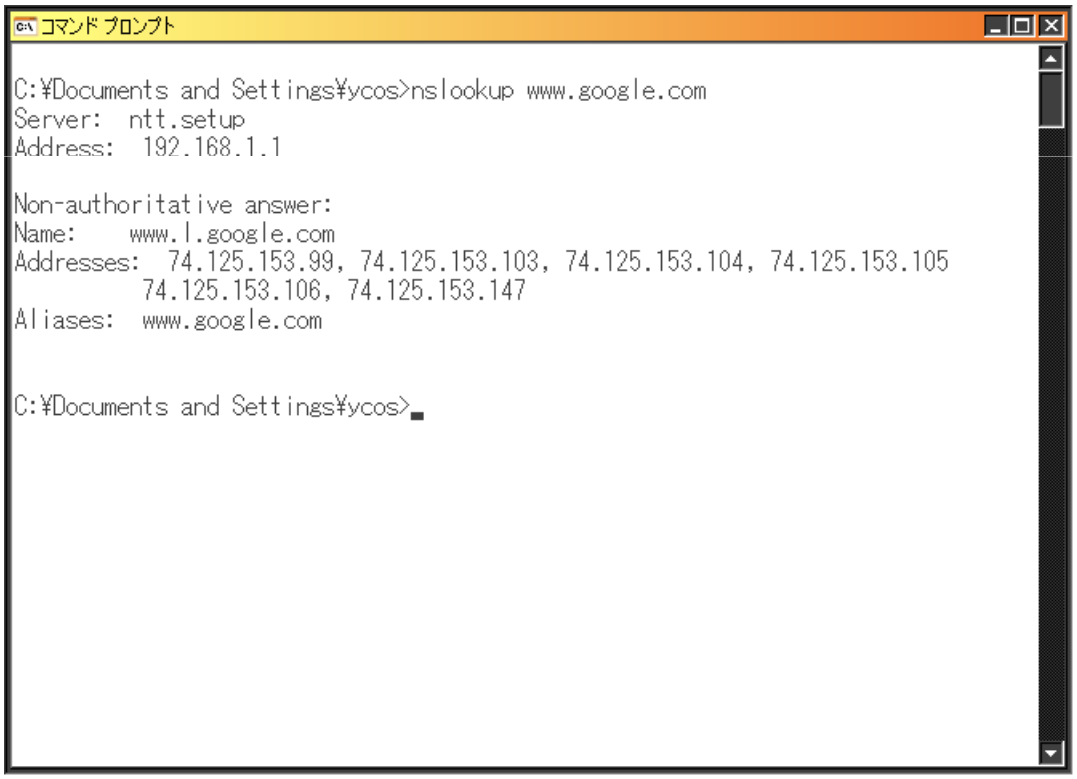
ping

- Windows　の ping 例
  - Windows 版ping は自動的に終了するが、linux 版では Ctrl + C で中断させる



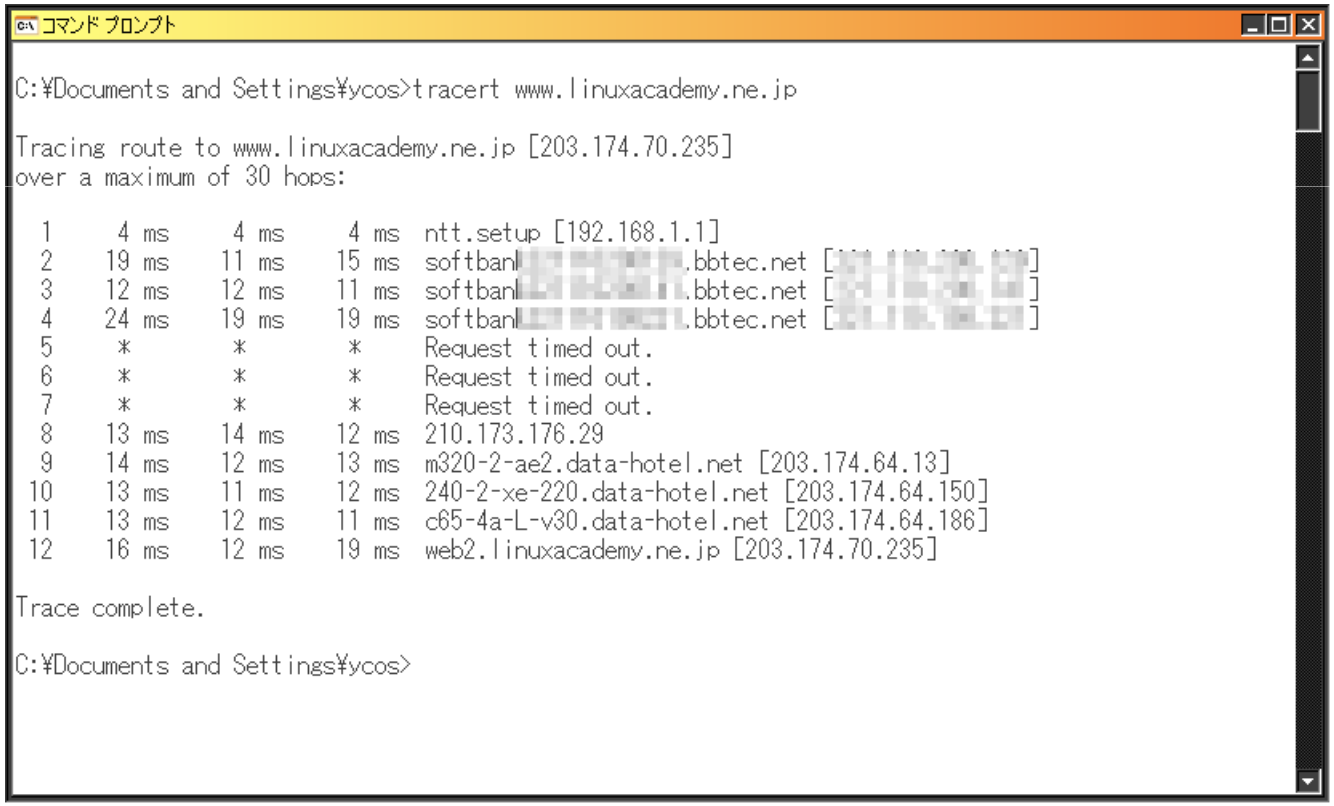
nslookup

- nslookup 例
  - www.google.com は窓口を6つ持っている事が分かる
  - サーバの正式名は www.l.google.com である



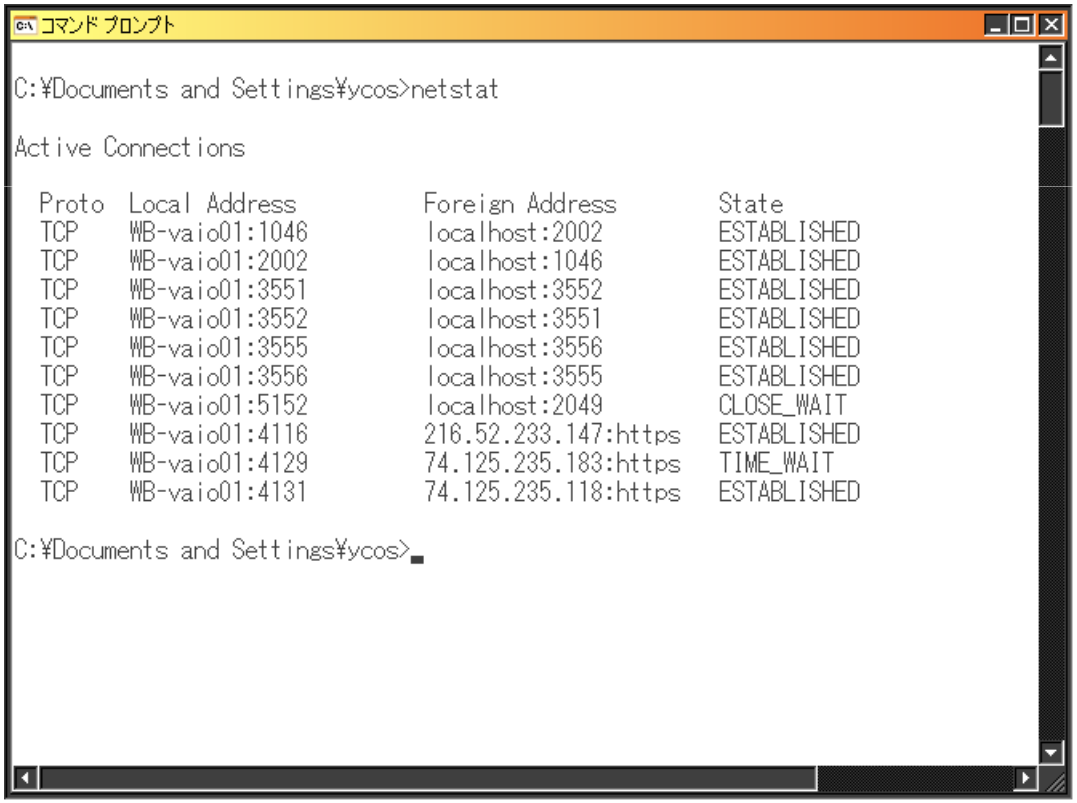
traceroute

- Windows 版は tracert
  - 自宅PC～Linuxアカデミー間でルータ12台経由
  - Linuxアカデミーはデータホテルにサイトを配置



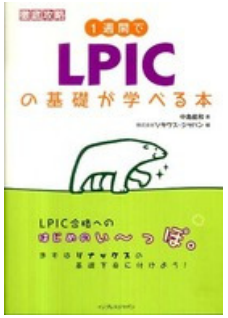
netstat

- netstat は現在のネットワーク接続状況を表示
  - Webサイト3か所と通信している
  - PC内部で7つのプロセス間通信が行われている



ご参考

参考文献



1週間でLPICの基礎が学べる本

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〕
（2010/03/21 出版）
294p / 21cm / A5判
ISBN：9784844328445

Linux教科書
LPICレベル1（第4版）

中島　能和【著】 濱野　賢一朗【監修】
翔泳社（2009/05/12 出版）
522p / 21cm / A5判
ISBN：9784798119311

LPI認定試験LPICレベル1“101／102”リリース3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（2009/04/02 出版）
607p / 21cm / A5判
ISBN：9784798022291

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