

LPICレベル1 技術解説セミナー

リナックスアカデミー

2017年1月28日
矢越昭仁



- 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
当資料はRed Hat系を採用

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)	Linux Mint	Linux Gentoo

*) 他にも Oracle、SAPといったソフトウェア・ベンダーも Linux OSを提供しており、50を超えるディストリビューターが存在する



- 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、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

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/>



- 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
 - ✓ 最新の市場動向に更新される (2015/6/1よりVer.4.0)

参照) <http://www.lpi.or.jp/>



- 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 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





- 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）



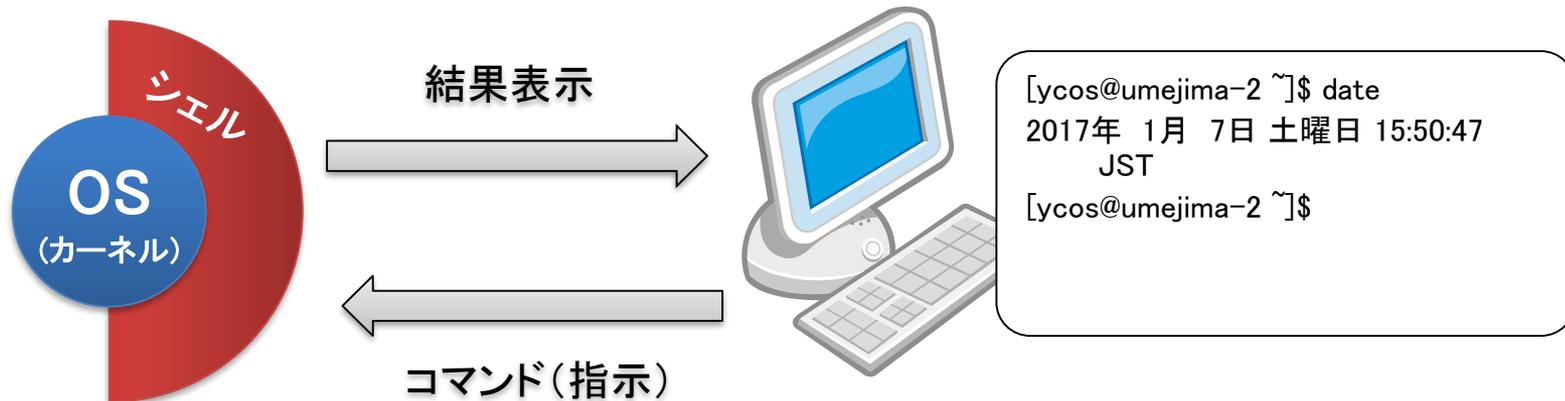
- キーボードから呪文(コマンド)を入力して操作する
→マウスは殆ど使わない
- 空白で区切った最初の単語をコマンド、続く単語を引数と呼ぶ引数のうちハイフン(-)で始まる物をオプションと呼ぶ
- コマンドを台本のように書き連ねる事で半自動的な操作が可能(シェル・スクリプト)
- コマンドはシェルによって処理される

```
$ 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



- 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

```
[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、# はスーパーユーザのみ限定



- 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検索時は「manpage」を添えて検索

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(シングルユーザーモード)にするために用いられる。shutdown に y と Y 以外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/init.d/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
```

Google search results for "manpage shutdown".

約 148,000 件 (0.20 秒) 検索オプション

Manpage of SHUTDOWN ☆

2010年4月25日 ... 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 ...

archive.linux.or.jp/JM/html/SysVinit/man8/shutdown.8.html - キャッシュ

Manpage of SHUTDOWN ☆

shutdown(8) は、socketd に関連づけられているソケットによる全二重接続 (full-duplex ...

archive.linux.or.jp/JM/html/LDP_man-pages/.../shutdown.2.html - キャッシュ

shutdown(8): bring system down - Linux man page ☆ - [このページを訳す]

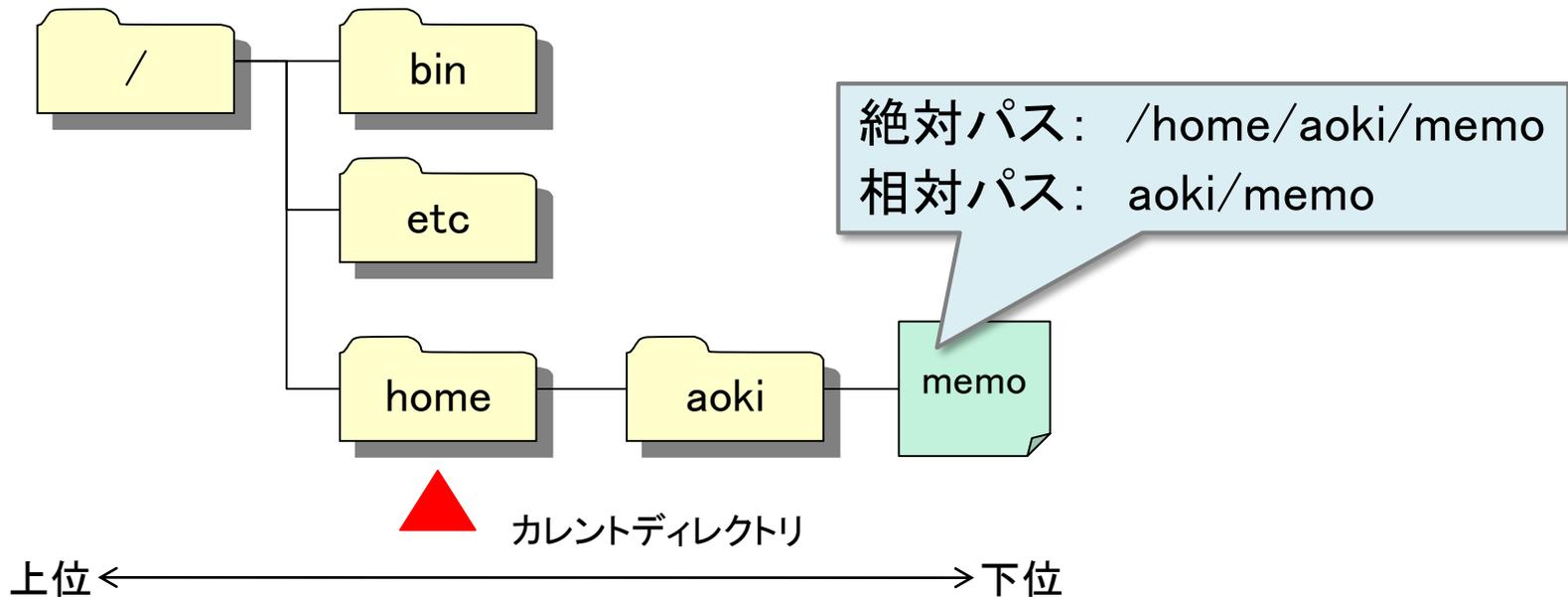
shutdown brings the system down in a secure way. All logged-in users are notified that the system is going down, and login(1) is blocked. It is possible to ...



- 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



- 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





- ディレクトリの移動 change directory
 - ✓ \$ cd [移動先ディレクトリ名: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、# はスーパーユーザのみ



• 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



- 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



- 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 CentOS 5.x/6.x/7.x での実測値



- 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 固定値X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 はスーパーユーザ、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	正式には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。現在は固定値X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



- 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]
 - ✓ 指定したユーザに切り替え(要パスワード)
 - ✓ オプション - は、環境の初期化を行う



- 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



- 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」グループのみ、「ユーザ:」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



• 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)

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



- 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(ugo)
 - ✓ +/- (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)



- 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



- 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

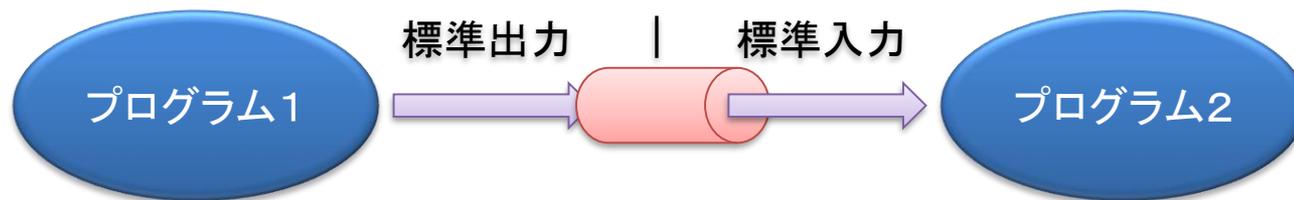
→ 失われる

```
[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]$
```



- 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
16 -rw-rw-r-- 1 ycos ycos 0 1月 4 10:07 c2
```

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]$ █
```



- 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 ~]#
```



- 任意のファイル名を表す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

```
[lycos@umejima-2 tmp2]$ ls
a al all a2 a2 b b1 b11 b12 b2 c cl c11 c12 c2
[lycos@umejima-2 tmp2]$ ls a*
a al all a2 a2
[lycos@umejima-2 tmp2]$ ls ??
al a2 b1 b2 c1 c2
[lycos@umejima-2 tmp2]$ ls [bc]??
b11 b12 c11 c12
[lycos@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。



- 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「変数とその値に置換(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

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



• 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、-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で更新しておく必要があります。



• 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
- ✓ 他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

```
[lycos@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
www-ldap-gw 1760/udp  www-ldap-gw # www-ldap-gw
www-dev    2784/tcp  www-dev     # world wide web - development
www-dev    2784/udp  www-dev     # world wide web - development
```



- 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
MinSpareServers   5
```

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ーションで括る
(この例ではシングルでも可)



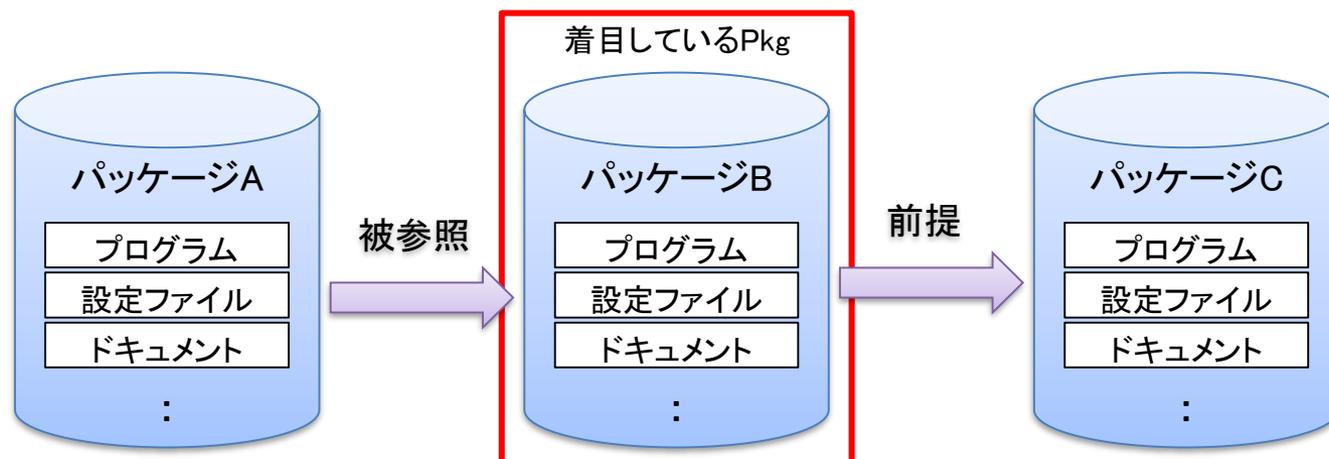
- ファイルの先頭部分を表示
 - ✓ `$ head [-n 行数:省略時10] [ファイル]`
- ファイルの末尾部分を表示
 - ✓ `$ tail [-n 行数:省略時10] [ファイル]`
- 行番号を付けてファイルを表示 number line
 - ✓ `$ nl [ファイル]`
 - ✓ `cat -n` もほぼ等価だが、`nl` は空行に番号を付けない
- 文字数を計数 word count
 - ✓ `$ wc [オプション] [ファイル]`
 - ✓ オプションは `-l`: 行数、`-w`: 単語、`-c`: バイト(文字)数を表示省略時はこの3つ全てを表示



システム管理



- パッケージとは、ソフトウェ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)





- RPM系

Redhat Package Manager
Yellowdog Updater Modified

- ✓ # 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

- Debian系

Advanced Packaging Tool

- ✓ # 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



- 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



- 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

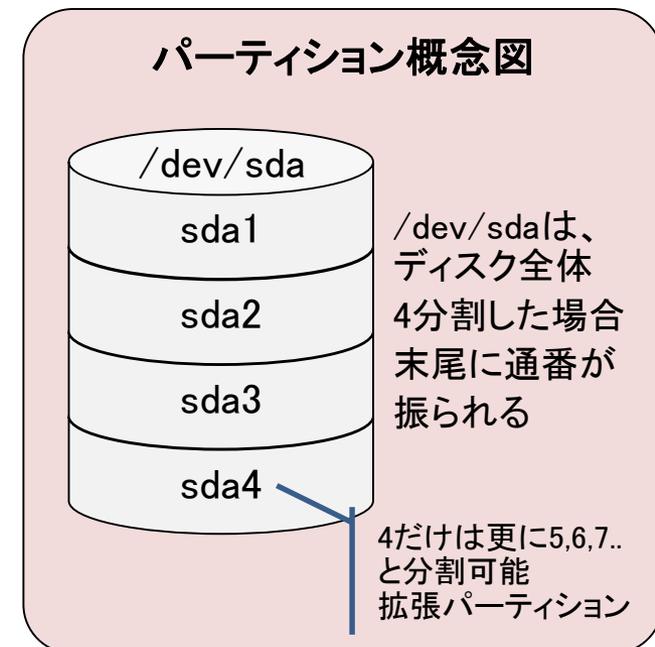
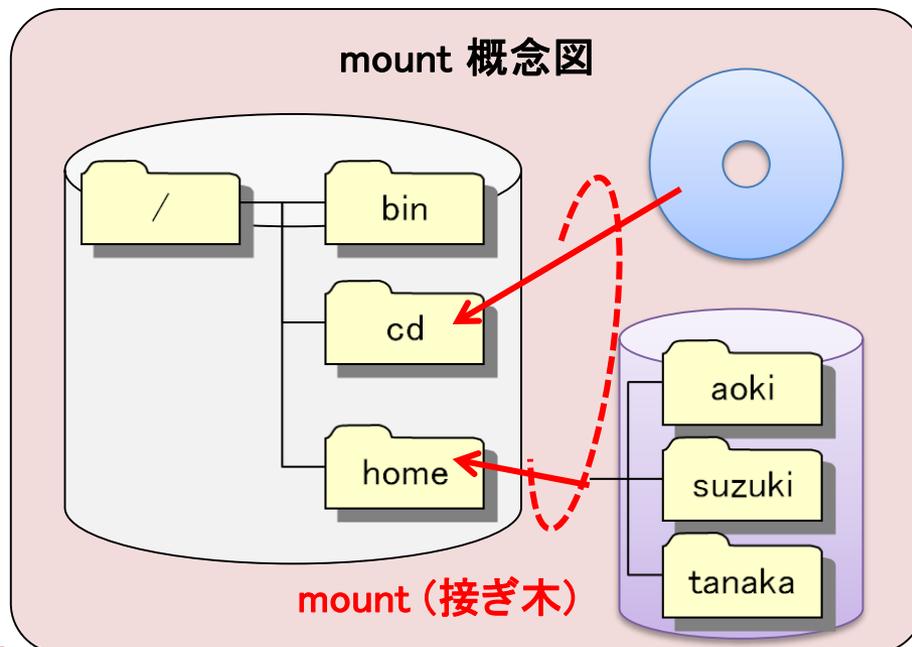
FS	最大容量*	特徴
ext3	32TiB	ext2の改良版、現在最も多くのディストリビューションで採用
ext4	1EiB	ext3の後継、パフォーマンス、セキュリティの向上
Reiserfs	16TiB	高速で、小容量ファイルを多数扱う用途向き
XFS	8EiB	高速かつ堅牢(技術的に安定)、大容量ファイル向き
NTFS	256 TiB	Windows のファイルシステム
FAT32	2TiB	Windows, USBやメモリーカード(デジカメ等)で利用
UDF	128 TiB	DVDのファイルシステム、ISO 9660 (CD)の拡張

*) TiB = 1024³Bytes, EiB = 1024⁴Bytes

CentOS 5.x では ext3、CentOS 6.xでは ext4、CentOS7.x では XFS が標準



- 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)





• 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
gdisk、partd はGPT(GUID Partition Table)に対応
- ✓ # 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

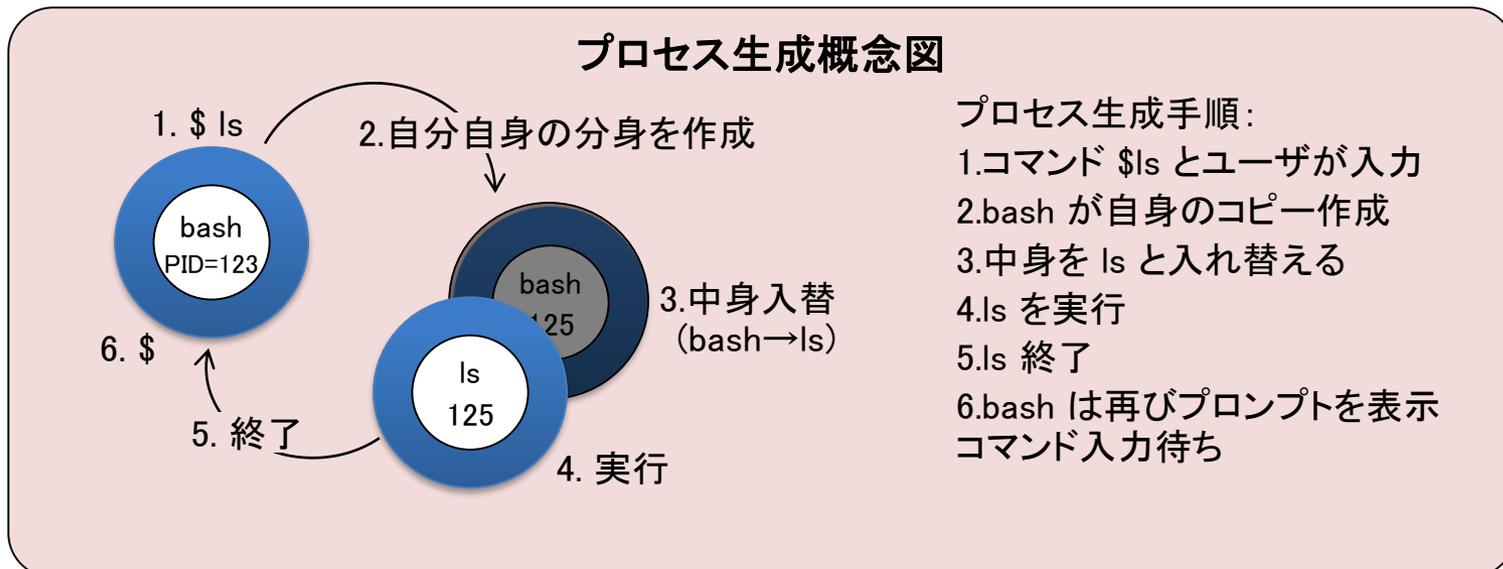
補足: 主なファイルシステムと、管理コマンド

	ext2/3/4	ReiserFS	XFS
生成(フォーマット)	mke2fs	mkreiserfs	mkfs.xfs
整合性のチェック	e3fsck	reiserfck	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 \geq プロセス \geq スレッド という関係





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

process status

✓ \$ 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

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

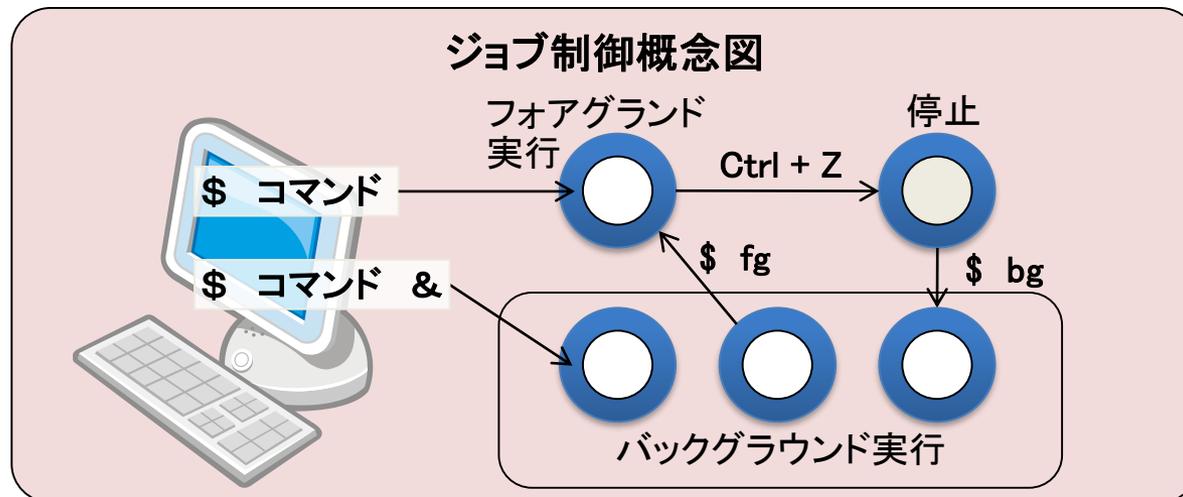
```
[lycos@umejima-2 LAPC]$ bash
[lycos@umejima-2 LAPC]$ echo $$
13553
[lycos@umejima-2 LAPC]$ kill -9 13553
強制終了
[lycos@umejima-2 LAPC]$ █
```

```
[lycos@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ウンドへ移動





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





• プロトコルについて

- ✓ IP (Internet Protocol) はインターネット上で目的地までの経路接続
- ✓ TCP(Transmission Control Protocol)はデータをパケットに小分けにして確実に配送する仕組み。信頼性が高いがパフォーマンスが低い
- ✓ UDP(User Datagram Protocol)はスピード優先で、エラー処理などはプログラムで行う
- ✓ TCP, UDP は紐づくプロセスをポート番号として管理

ポート	サービス	説明
20,21	FTP	ファイル転送、21がコマンド制御、20がデータ転送用
23	TELNET	リモートコンソール、サーバー動作確認用
25	SMTP	メール送信、転送用
53	DNS	名前解決用(UDP)、サーバ情報連携(ゾーン転送、TCP)
80	HTTP	ホームページ閲覧
110	POP3	メール閲覧
123	NTP	ネットワーク時間同期
143	IMAP4	メール閲覧



• ネットワーク設定に関わる主なファイル

- ✓ 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
- ✓ resolv.conf 名前解決 (DNS = インターネットの電話帳) 設定
- ✓ hostname コンピューター名

`/etc/sysconfig/network`

```
NETWORKING=yes  
GATEWAY=192.168.0.1
```

`/etc/resolv.conf`

```
search uso800.com  
nameserver 10.20.250.1  
nameserver 10.20.0.1
```

`/etc/hostname`

```
mssystem.uso800.com
```

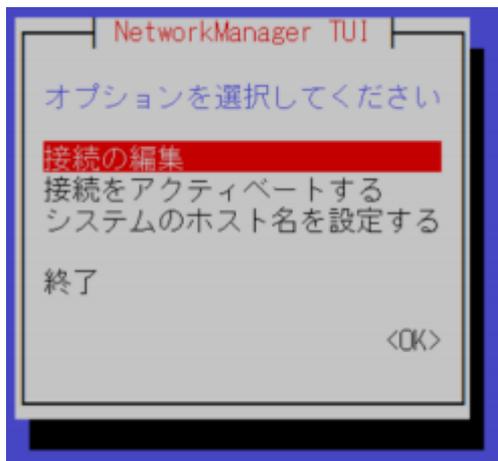
`/etc/sysconfig/network-scripts/ifcfg-enp2s0`

```
TYPE="Ethernet"  
BOOTPROTO="none"  
DEFROUTE="yes"  
NAME="enp2s0"  
UUID="9e2548c9-7e71-48f7-b519-9fd3a4a4dfc0"  
DEVICE="enp2s0"  
ONBOOT="yes"
```



• NetworkManager による自動設定

- ✓ NICを自動検出し、設定を自動化
- ✓ 最近では多くのディストリビューションで採用されている
- ✓ メニュー形式のツールを装備



```
# nmtui
```

NetworkManager Text User Interfaceの例

他にもコマンドとして、nmcli (Command Line Interface)がある



- ip addr show [NIC名 : 省略時は全て]

```
$ ip addr show eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
    link/ether 00:15:5d:03:0d:04 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.12/24 brd 192.168.1.255 scope global dynamic eth0
        valid_lft 7522sec preferred_lft 7522sec
    inet6 2408:210:2d44:6700:215:5dff:fe03:d04/64 scope global dynamic
        valid_lft 14194sec preferred_lft 12394sec
    inet6 fe80::215:5dff:fe03:d04/64 scope link
        valid_lft forever preferred_lft forever
```

(参考)

```
$ ifconfig eth0
```

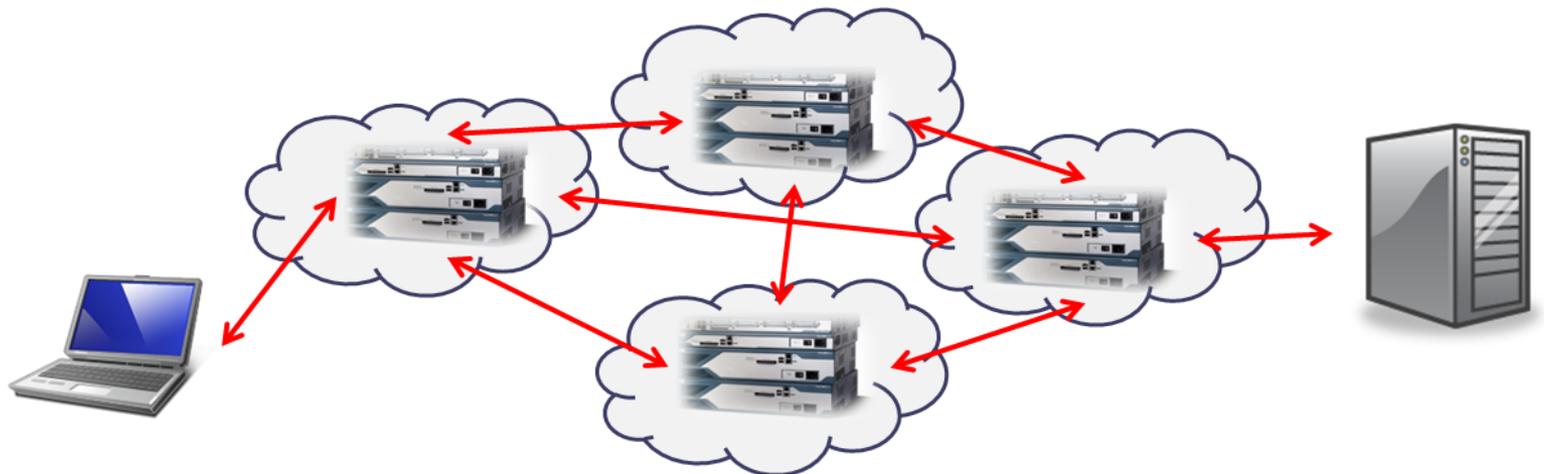
```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.12 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::215:5dff:fe03:d04 prefixlen 64 scopeid 0x20<link>
    inet6 2408:210:2d44:6700:215:5dff:fe03:d04 prefixlen 64 scopeid 0x0<global>
    ether 00:15:5d:03:0d:04 txqueuelen 1000 (Ethernet)
    RX packets 7906 bytes 1846730 (1.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1448 bytes 299939 (292.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



• 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
ネットワーク利用状況表示

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 ホスト名

```
$ ping www.google.com
```

```
PING www.google.com (216.58.200.164) 56(84) bytes of data.  
64 bytes from nrt12s11-in-f4.1e100.net (216.58.200.164): icmp_seq=1 ttl=54 time=5.18 ms  
64 bytes from nrt12s11-in-f4.1e100.net (216.58.200.164): icmp_seq=2 ttl=54 time=2.50 ms  
64 bytes from nrt12s11-in-f4.1e100.net (216.58.200.164): icmp_seq=3 ttl=54 time=2.36 ms  
64 bytes from nrt12s11-in-f4.1e100.net (216.58.200.164): icmp_seq=4 ttl=54 time=2.45 ms  
^C
```

```
--- www.google.com ping statistics ---
```

```
4 packets transmitted, 4 received, 0% packet loss, time 3003ms  
rtt min/avg/max/mdev = 2.367/3.128/5.186/1.189 ms
```

最小/平均/最大/平均偏差(ばらつき)

```
$ ping -c 2 www.yahoo.co.jp
```

```
PING www.g.yahoo.co.jp (182.22.59.229) 56(84) bytes of data.  
64 bytes from f1.top.vip.ssk.yahoo.co.jp (182.22.59.229): icmp_seq=1 ttl=48 time=9.31 ms  
64 bytes from f1.top.vip.ssk.yahoo.co.jp (182.22.59.229): icmp_seq=2 ttl=48 time=7.68 ms
```

```
--- www.g.yahoo.co.jp ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 1001ms  
rtt min/avg/max/mdev = 7.688/8.501/9.314/0.813 ms
```



- nslookup [-query=レコード] ホスト名/ドメイン名/IPアドレス

```
$ nslookup www.linuxacademy.ne.jp
```

```
Server:      10.20.250.1
Address:     10.20.250.1#53
```

Non-authoritative answer:

```
Name:        www.linuxacademy.ne.jp
Address:     203.174.70.235
```

```
$ nslookup -query=mx ezweb.com
```

```
Server:      10.20.250.1
Address:     10.20.250.1#53
```

Non-authoritative answer:

ezweb.com mail exchanger = 10 inbound.ezweb.com.netsolmail.net.

Authoritative answers can be found from:

ezweb.com nameserver = ns64.worldnic.com.

ezweb.com nameserver = ns63.worldnic.com.

ns63.worldnic.com internet address = 207.204.40.132

ns64.worldnic.com internet address = 207.204.21.132

レコード	意味
A	IPアドレス(既定)
NS	ネームサーバー情報
MX	メールサーバー情報
PTR	ホスト名(逆引き)
CNAME	別名
TXT	その他情報



- traceroute ホスト名

```
$ traceroute www.google.com
```

```
traceroute to www.google.com (216.58.200.164), 30 hops max, 60 byte packets
```

```
1 gateway (10.20.0.1) 2.525 ms 2.938 ms 3.357 ms
2 172.30.0.1 (172.30.0.1) 12.399 ms 12.831 ms 12.075 ms
3 c65-4a-R-v302.data-hotel.net (203.174.70.254) 13.402 ms 13.793 ms 14.234 ms
4 480-1-ae14.data-hotel.net (203.174.66.29) 14.774 ms 14.848 ms 15.953 ms
5 125-6-112-1.data-hotel.net (125.6.112.1) 17.669 ms 125-6-112-5.data-hotel.net (125.6.112.5)
  18.111 ms 18.513 ms
6 ex-mx480-1-ae1.data-hotel.net (203.174.64.98) 19.013 ms 15.395 ms 15.745 ms
7 218.100.6.53 (218.100.6.53) 16.177 ms 5.651 ms 5.919 ms
8 108.170.242.97 (108.170.242.97) 8.147 ms 108.170.242.129 (108.170.242.129) 10.714 ms
  108.170.242.97 (108.170.242.97) 10.295 ms
9 66.249.95.41 (66.249.95.41) 13.514 ms 72.14.232.45 (72.14.232.45) 14.017 ms 14.543 ms
10 nrt12s11-in-f164.1e100.net (216.58.200.164) 15.070 ms 15.748 ms 16.337 ms
```



- netstat [オプション]

netstat -antp

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN	1/systemd
tcp	0	0	192.168.122.1:53	0.0.0.0:*	LISTEN	1165/dnsmasq
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	1092/sshd
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN	1008/cupsd
tcp	0	0	0.0.0.0:25	0.0.0.0:*	LISTEN	1269/master
tcp	0	0	127.0.0.1:45024	127.0.0.1:80	TIME_WAIT	-
tcp	0	0	127.0.0.1:45022	127.0.0.1:80	TIME_WAIT	-
tcp6	0	0	:::111	:::*	LISTEN	1/systemd
tcp6	0	0	:::80	:::*	LISTEN	1024/httpd
tcp6	0	0	:::22	:::*	LISTEN	1092/sshd
tcp6	0	0	:::1:631	:::*	LISTEN	1008/cupsd
tcp6	0	0	:::25	:::*	LISTEN	1269/master

\$ netstat -r

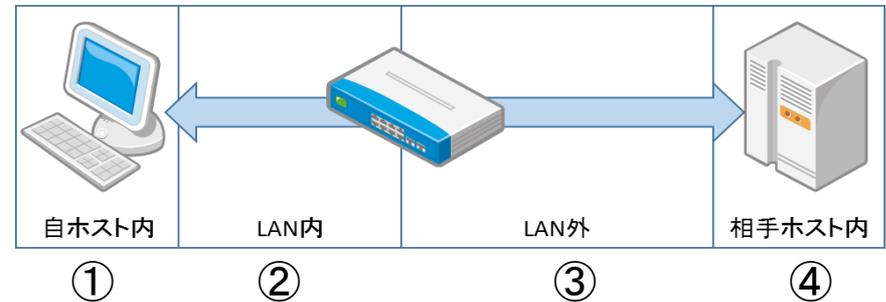
Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
Default	gateway	0.0.0.0	UG	0	0	0	enp2s0
10.20.0.0	0.0.0.0	255.255.0.0	U	0	0	0	enp2s0
192.168.122.0	0.0.0.0	255.255.255.0	U	0	0	0	virbr0



• 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



良くある、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 ...
```



END

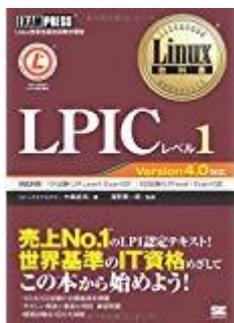


参考文献



ゼロからはじめるLinuxサーバー構築・運用ガイド 動かしながら学ぶWebサーバーの作り方

中島 能和【著】
翔泳社 (2016/07発売)
269p / 24cm / B5判
ISBN: 9784798146379



Linux教科書
LPICレベル1 -Version 4.0対応

中島 能和【著】 濱野 賢一郎【監修】
翔泳社 (2015/06 出版)
550p / 21cm / A5判
ISBN: 9784798141916



これだけで突破[合格ライン]LPIC合格読本 —レベル1攻略

中島 能和/矢越 昭仁【著】
技術評論社 (2014/09発売)
143p / 26cm / B5判
ISBN: 9784774166728

<http://www.lpi.or.jp/lpic1/book.shtml>

Linux 専門スクール



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

LPIジャパン



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

<http://www.lpi.or.jp/linuxtext/text.shtml>