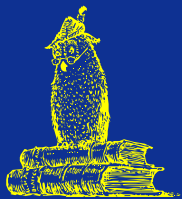


LPICレベル1技術解説無料セミナー

パナソニックラーニングシステムズ(株)

寺井 健一郎

2011年8月18日



LPICの特徴

- 世界標準
LPICは世界共通の国際認定制度です。
- 中立
LPIC はLinuxの技術力を中立公正に判定する試験です。
- 世界最大
LPICは、世界最大のLinux技術者受験者数を有します。



LPICレベル1 試験範囲

- 2009年4月改訂

開発バージョン名 : Version 3.0、リリース名 : Release3

- 出題範囲詳細

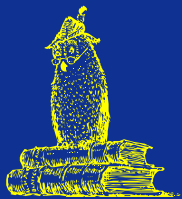
101試験 <http://lpi.or.jp/lpic1/range/range101.shtml>

102試験 <http://lpi.or.jp/lpic1/range/range102.shtml>

- 重要度

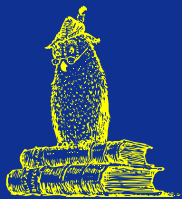
「重要度 3」であれば、

試験ではその出題範囲に関連する問題が3題出題される

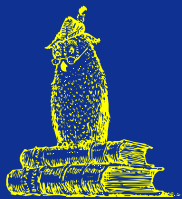
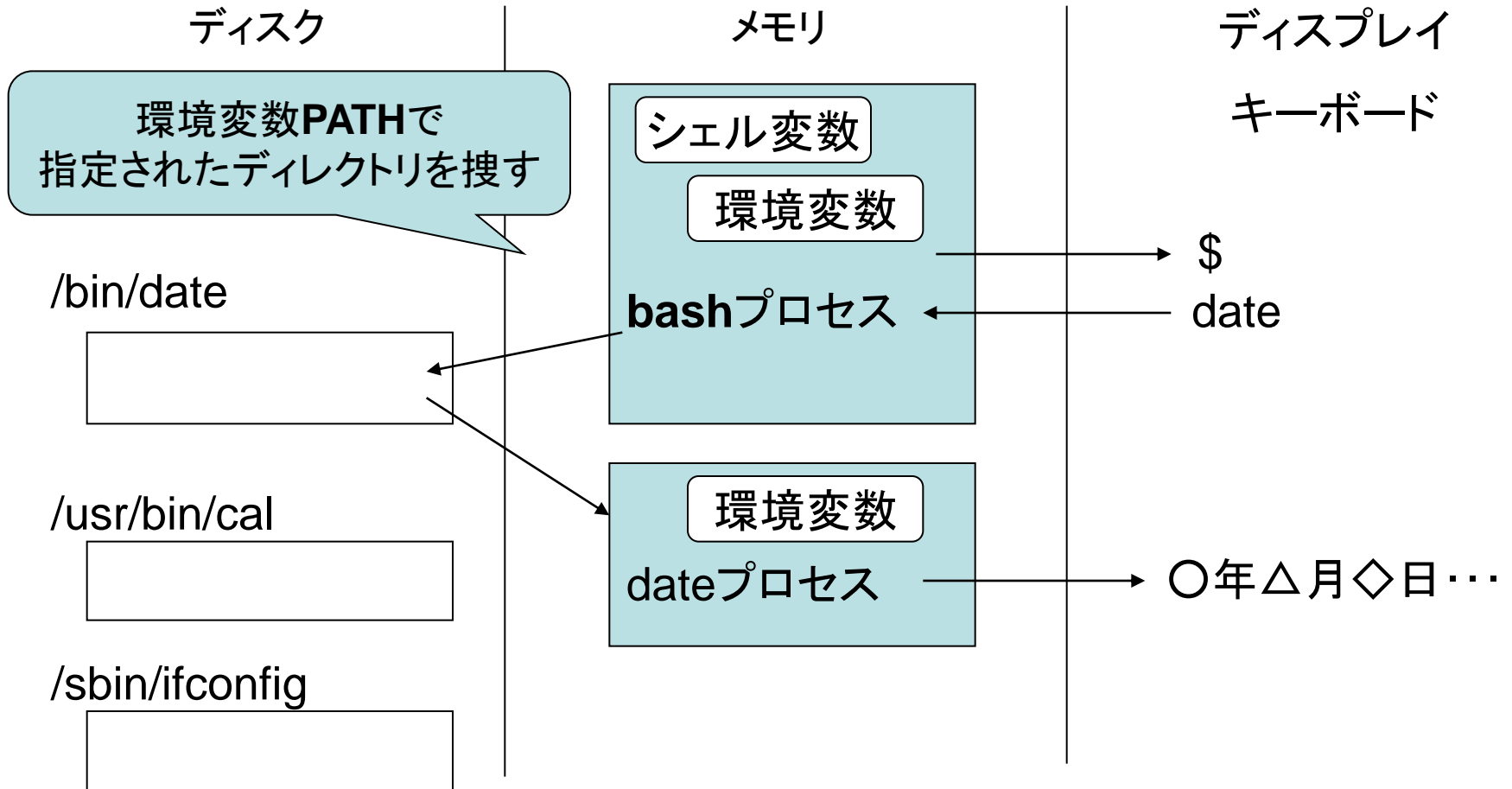


本日解説するポイント

- 101試験
 - コマンド行で操作する
 - 基本的なファイル管理を行う
 - ストリーム、パイプ、リダイレクトを使う
 - プロセスを生成、監視、終了する
- 102試験
 - 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
 - シェル環境のカスタマイズと使用
 - 簡単なスクリプトをカスタマイズまたは作成する
 - ネットワーク関連



103.1 コマンド行で操作する シェルの機能

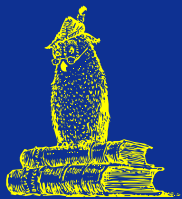


103.1 コマンド行で操作する 環境変数 PATH

- コマンド(プログラム)のある場所を探すPATH(経路)
...サーチパス
- echo \$PATH で内容を表示
- コマンドを実行するには、
 - PATH上のディレクトリに、配置する
 - 絶対もしくは相対パスで、直接コマンドのありかを指定する
- 環境変数PATHにディレクトリを追加
(例) /home/hogehogeを追加

```
PATH=$PATH:/home/hogehoge
```

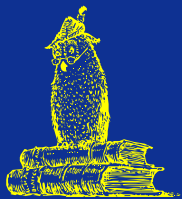
```
PATH=/home/hogehoge:$PATH
```



シェル変数・環境変数

- set シェル変数・環境変数の両方表示
- env, printenv 環境変数のみ表示
- unset 変数の削除
- export 変数名 シェル変数を環境変数にする

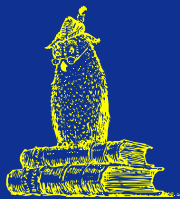
- 一時的に変数の値を変えてコマンドを実行
env LANG=fr_FR.UTF-8 date
LANG=fr_FR.UTF-8 date
- 一時的に変数の値を無効にしてコマンドを実行
env -u LANG date



103.1 コマンド行で操作する コマンド履歴

- `history` 履歴一覧
- `history -c` 履歴の消去
- 環境変数 `HISTSIZE` 履歴の保存件数。デフォルトは1000
- 環境変数 `HISTFILE` 履歴の保存ファイル
デフォルトは `~/.bash_history`

<code>!n</code>	historyの表示結果でn番のコマンドを実行	<code>!?文字列</code>	指定した文字列が含まれるコマンドで、直近に入力したものを実行
<code>!!</code>	直前に入力したコマンドを再実行	<code>^文字列1^文字列2^</code>	直前に実行したコマンドの文字列1を文字列2に変えて実行
<code>!文字列</code>	指定した文字列で始まるコマンドで、直近に入力したものを実行	<code>:p</code>	<code>:p</code> をつけると、コマンドを実行せずに表示のみ行う



ファイル操作・ディレクトリ操作

- ファイル操作

- cp コピー
- mv 移動(名前の変更)
- rm 消去

- ディレクトリ操作

- mkdir ディレクトリ作成
- rmdir ディレクトリ削除
(中身が空であること)

再帰的な(Recursive)

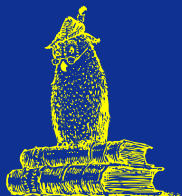
ディレクトリツリーをたどって
全てのファイルやディレクトリを
なめていく...と云うこと

```
cp -r dir1 dir2
```

で、dir1をまるごとdir2にコピー

```
rm -r dir1
```

で、dir1とその内部をすべて削除



103.3 基本的なファイル管理を行う findコマンド

find 基点ディレクトリ [オプション] 検索条件 [アクション]
(例) find /etc -name "httpd*" -ls

- 検索条件

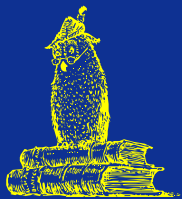
-name ファイル名 指定した名前のファイルを検索
ほかに -mtime (最終更新日時)、-atime(最終アクセス日時) など

- アクション

-print 検索結果をディスプレイに表示(デフォルト)
-ls 見つかったファイルの詳細情報(ls -lと同等の内容)を表示
-exec コマンド {} \; 見つかったファイルに対してコマンドを実行(即座に)
-ok コマンド {} \; 見つかったファイルに対してコマンドを実行(確認後)

- オプション

-maxdepth n n番目のディレクトリ階層まで検索



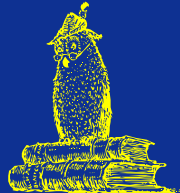


アーカイブと圧縮

- tarコマンド (Tape Archive)
 - 複数のファイルを1つにまとめる(アーカイブする)
 - `tar cvf ファイル名.tar ファイル1 ファイル2 (ディレクトリ) ...`
指定したファイルやディレクトリをtarファイルにまとめる
 - `tar xvf ファイル名.tar`
tarファイルからファイルを抽出する
 - `tar tvf ファイル名.tar`
tarファイル内のファイルのリストを表示

- gzipコマンド
 - gzip形式でファイルを圧縮
 - 圧縮すると、ファイル名の末尾に自動的に.gzがつく
 - `gzip ファイル名` 圧縮
 - `gunzip ファイル名` 解凍(gzip -d ファイル名 でも同じ)

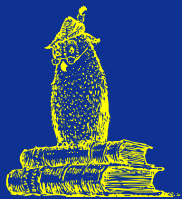
tar zcvf :
アーカイブして圧縮
tar zxvf :
解凍して抽出



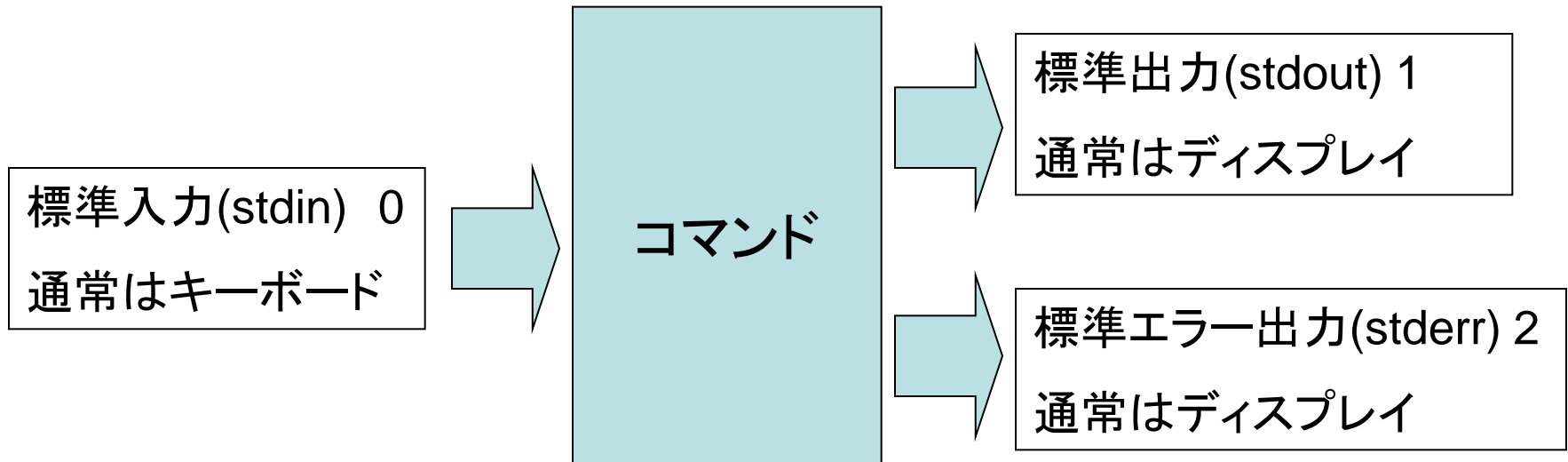
cpioおよびdd

- cpioコマンド (CoPy I/O)
 - 複数のファイルを1つにまとめる
ls | cpio -ov > ファイル名.cpio
カレントディレクトリにあるファイルをcpio形式でアーカイブする
 - cpio -iv < ファイル名.cpio**
ファイルを抽出する
- ddコマンド
 - 1つのファイルを別ファイルに送る
dd if=入力ファイル of=出力ファイル bs=ブロックサイズ count=回数

(例) **dd if=/dev/sda of=/tmp/mbrdump bs=512 count=1**
ディスクの先頭512バイト(マスターブートレコード)を
/tmp/mbrdumpファイルにコピー

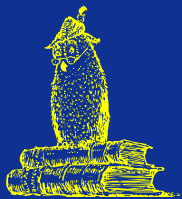


標準入力・標準出力・標準エラー出力



コマンドを実行する場合、通常は標準入力(キーボード)からデータを読み、処理結果を標準出力(ディスプレイ)に出力する。「ストリーム」

catコマンドで引数にファイル名を指定しなかった場合は、キーボードから読み込んだデータをディスプレイに表示する処理を行う



リダイレクト

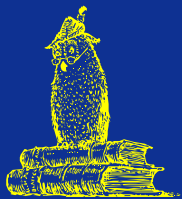
- リダイレクト 標準入力、標準出力などをファイルに切り替えること

標準出力の切り替え	コマンド > ファイル名 (上書き) コマンド >> ファイル名 (追記)
標準エラー出力の切り替え	コマンド 2> ファイル名
標準出力と標準エラー出力を同じファイルに書く	コマンド > ファイル名 2>&1
標準入力の切り替え	コマンド < ファイル名
ヒア・ドキュメント (here document)	<<の後ろと同じ文字列が現れるまで入力を続ける

```
tr a-z A-Z < file01 > file02
```

file01を読み、小文字を大文字に変換して、file02へ出力

```
cat > file03 <<EOF  
Hello Linux World  
hogegege EOF  
EOF
```



103.4 ストリーム・パイプ・リダイレクトを使う パイプ

- パイプ

コマンドAの標準出力を、コマンドBの標準入力として使うことができる

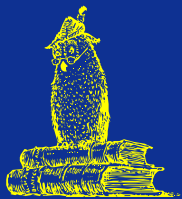
コマンドA | コマンドB

(例) ファイルから重複している行を見つける

```
cat data.txt | sort | uniq
```

(例) 人気ログインシェルランキング?

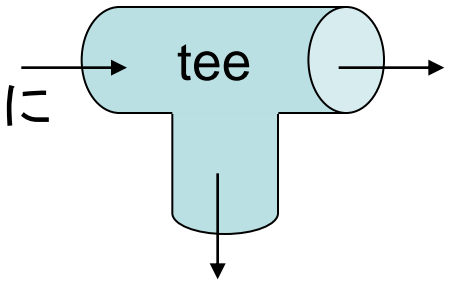
```
cat /etc/passwd | cut -d : -f 7 | sort | uniq -c
```



teeコマンド、xargsコマンド

- コマンド | tee ファイル名

前のコマンドの出力結果(標準出力)を受け取り、それをディスプレイに表示し、かつ指定したファイルに書き込むもの。水道のT字管のイメージ



- コマンド > /dev/null 2>&1

コマンドの出力結果を、どこにも出力しない

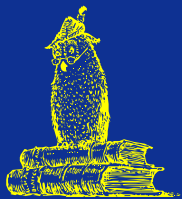
/dev/nullは特殊なファイルで、入力されたものをすべて消去してしまう

- コマンドA | xargs コマンドB

コマンドAの出力結果を、コマンドBの引数とする

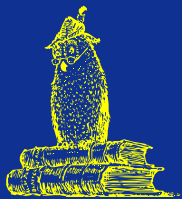
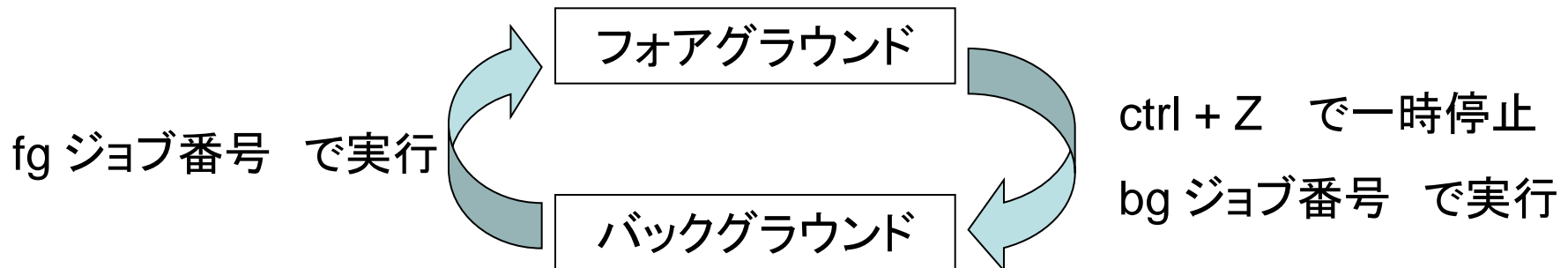
(例) `find . -name "*.c" -exec cat {} \;`

`find . -name "*.c" | xargs cat`



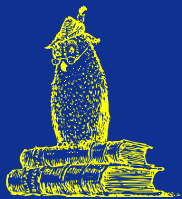
103.5 プロセスを生成、監視、終了する ジョブ

- シェル上でコマンドを実行すると、「ジョブ」という単位で処理される
 - フォアグラウンドジョブ: キーボードからの入力を受け取れる状態のジョブ
 - バックグラウンドジョブ: 陰で動いているジョブ
- バックグラウンドジョブとして起動するには、コマンド名の後ろに & をつける
- 現在稼働中のジョブの一覧は jobs コマンドで表示できる



103.5 プロセスを生成、監視、終了する プロセス

- Linux上での、処理の単位 (ジョブ \geq プロセス)
 - プロセスはプロセスID(pid)で管理される
 - システム内で重複しないように番号がつけられる
 - 同じコマンドであっても、実行するたびにプロセスIDは変わる
-
- ps 現在の端末で自分が実行したプロセスを表示
 - ps ax システムで稼働中の全プロセスを表示
 - ps l 詳細情報を表示(親プロセスIDなど)
 - ps u 詳細情報を表示(プロセスの所有者名など)
 - ps axlw 折り返して全体を表示(通常は1行80文字で切られる)
 - ps axlf ツリー状に表示 (≒pstreeコマンド)

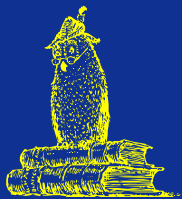


103.5 プロセスを生成、監視、終了する プロセスにシグナルを送る

kill -シグナル番号(またはシグナル名) プロセスID

killall -シグナル番号(またはシグナル名) プロセス名

シグナル名	シグナル番号	意味
HUP	1	ハングアップ。デーモンプロセスの初期化
INT	2	割込み。処理の中断 (= Ctrl+C)
KILL	9	強制終了
TERM	15	終了(デフォルト)
TSTP	20	一時停止 (= Ctrl+Z)
CONT	18	再開



105.1 シェル環境のカスタマイズと使用

alias(エイリアス)、関数

- alias

コマンドに別名をつける

- 設定

```
alias ls='ls -l'
```

- 定義されているaliasの表示

```
alias
```

- 解除

```
unalias ls
```

- 一時解除

```
\ls
```

- 関数

```
function 関数名() { コマンド; }
```

(例)

```
function pwdfs() { pwd; ls; }
```

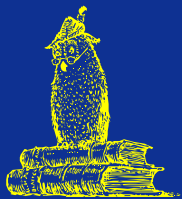
```
function pwdfs2() { pwd; ls -l "$1"; }
```

※\$1は関数を呼び出すときの1つ目の引数

- 定義されている関数の表示

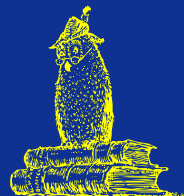
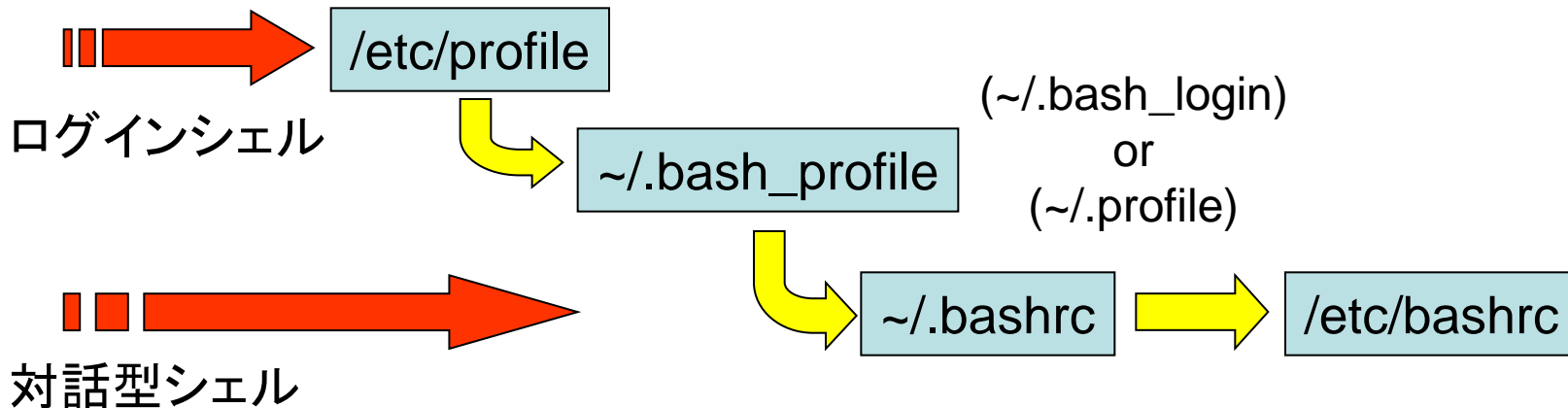
```
declare -f
```

set でも表示できる



bashの設定ファイル

	ログイン時一度だけ (環境変数の設定等)	bashを起動するたび (alias・関数の設定等)
全ユーザ	/etc/profile	/etc/bashrc
個別ユーザ	~/.bash_profile	~/.bashrc



シェルのオプション

- >での上書きを禁止する

`set -o noclobber` (解除するときは `set +o noclobber`)

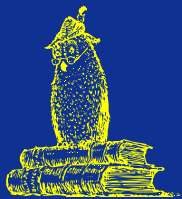
- メタキャラクタ(ワイルドカード)で使用する記号(* や?など)を、すべて文字とみなす

`set -o noglob` (解除するときは `set +o noglob`)

- vi風のキーバインドにする

`set -o vi` (emacs風にするには `set -o emacs`)

操作中にESCキーを押すと、viのコマンドモードと同様にhやlキーでのカーソル移動やxキーでの文字削除ができる



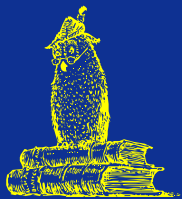
特別な変数、testコマンド

- echo \$? 1つ前に実行したコマンドの終了ステータス
0ならば成功(正常終了)、0以外であれば失敗
- echo \$\$ 現在のシェルのプロセスID
- testコマンド、[]コマンド
引数で指定した条件式が真(正しい)ならば0を、
偽(誤り)であれば0以外の値を、終了ステータスとして返す

```
VAL=100  
test $VAL -gt 10  
echo $?  
0
```

```
VAL=3  
test $VAL -gt 10  
echo $?  
1
```

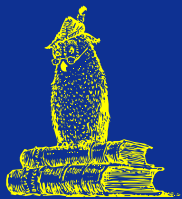
```
VAL=3  
[ $VAL -gt 10 ]  
echo $?  
1
```



105.2 簡単なスクリプトをカスタマイズまたは作成する シェルスクリプトでの制御構造

- **if** 条件文
then
 実行文
elif 条件文
 実行文
else
 実行文
fi

- **for** 識別子 in リスト
do
 \$識別子を使う文
done
- **while** 条件文
do
 実行文
done



if文、for文の例

sample1.sh

```
#!/bin/bash

if [ "$1" -eq 100 ]
then
    echo '引数は100です'
else
    echo '引数は100ではありません'
fi
```

sample2.sh

```
#!/bin/bash

for i in `seq 1 5`
do
    ping -c 1 192.168.0.$i
done
```

``
(バッククォート)

seqコマンド 連番を自動生成

seq 10

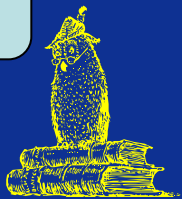
1 2 3 4 5 6 7 8 9 10

seq 5 10

5 6 7 8 9 10

seq 5 2 10

5 7 9



ユーザ情報の設定ファイル

- /etc/passwdファイル
 - 所有者、所有グループはroot、パーミッションは rw-r--r--
 - 7項目を：区切りで記述
ユーザ名、パスワード(xの場合は、/etc/shadowに記述されている)、ユーザID、グループID、コメント、ホームディレクトリの絶対パス、ログインシェル
- /etc/shadowファイル
 - 所有者、所有グループはroot、パーミッションは r-----
 - 9項目を：区切りで記述
ユーザー名、暗号化されたパスワード、パスワードの最終更新日、パスワードの有効期限、ユーザアカウントの有効期限 など



ユーザの登録

- useradd ユーザ名
 - /etc/passwd、/etc/shadow、(/etc/group)ファイルにユーザ情報を追加
 - ホームディレクトリ（デフォルトでは /home/ユーザ名）を作成
 - /etc/skelディレクトリ内のファイルを、ホームディレクトリにコピー

/etc/skel : skeleton (スケルトン: 骨組み)

.bash_profile、.bashrc、.bash_logoutファイルなどがある

- passwd [ユーザ名]
 - パスワードの設定。/etc/shadowファイルを更新
 - rootユーザは、引数にユーザ名を指定して、一般ユーザのパスワードを設定できる
 - 一般ユーザは、引数なしでpasswdを実行し、自分のパスワードを変更

パスワードの有効期限などは、chage (change age)コマンドで変更



107.1 ユーザ・グループアカウント、関連するシステムファイルを管理

ユーザの削除、ログイン禁止

- userdel ユーザ名
 - /etc/passwd、/etc/shadow、(/etc/group)ファイルから、ユーザ情報を削除
 - ホームディレクトリを削除する場合、-r オプションが必要
- ユーザのログインを禁止する
 - 一時的に…パスワードのロック
 - usermod -L ユーザ名 -U でアンロック
 - または passwd -l ユーザ名 -u でアンロック
 - 恒久的に…ログインシェルを /sbin/nologin または /bin/false にする
 - /etc/nologinファイルがあれば、一般ユーザはログインできない



ネットワークインターフェースの設定

- 設定ファイル

 - `/etc/sysconfig/network-scripts/ifcfg-eth0` (RedHat 系)

 - `/etc/network/interfaces` (Debian)

- 確認

 - `ifconfig`

- eth0を一時的に無効にする

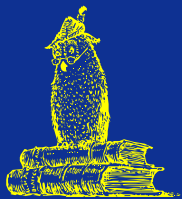
 - `ifconfig eth0 down` または `ifdown eth0`

- eth0を一時的に有効にする

 - `ifconfig eth0 up` または `ifup eth0`

- eth0にIPアドレスを設定

 - `ifconfig eth0 192.168.2.20 netmask 255.255.255.0`



ルーティングテーブル

- ルーティングテーブルの表示

`route` または `netstat -r`

DNSでの名前解決ができない場合、`-n`をつけるとホスト名をIPアドレスで表示する

- ルーティングの追加

```
route add -net 192.168.10.0 netmask 255.255.255.0 gw 192.168.2.254
```

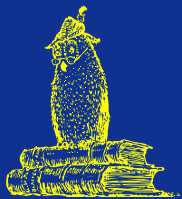
```
route add default gw 192.168.2.1
```

- ルーティングの削除

```
route del -net 192.168.10.0 netmask 255.255.255.0 gw 192.168.2.254
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.10.0	192.168.2.254	255.255.255.0	UG	0	0	0	eth0
192.168.2.0	*	255.255.255.0	U	0	0	0	eth0
default	192.168.2.1	0.0.0.0	UG	0	0	0	eth0



名前解決

- ファイルによる指定
/etc/hosts

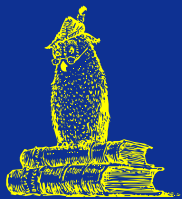
```
127.0.0.1      localhost.localdomain localhost
192.168.2.254  router1
192.168.2.10   server1.hogehoge.jp server1
```

- DNS 参照先の設定
/etc/resolv.conf

```
search localdomain
nameserver 192.168.2.1
nameserver 192.168.2.11
```

- 問い合わせ順
/etc/nsswitch.conf

```
hosts: files dns
```



109 ネットワークの基礎 診断コマンド

- ping 疎通確認(トラブルの切り分け)
ping [-c 回数] IPアドレス または ホスト名
- traceroute 経路確認
traceroute IPアドレス または ホスト名
- tracepath tracerouteと同じ。MTU値も表示
- netstat ネットワーク情報
 - netstat -t TCPソケットで、接続が確立されているポートを表示
 - netstat -ta TCPソケットで、接続待ちのポートも表示
 - netstat -ua UDPソケットで、開いているポートを表示
 - nをつけると、ホスト名やサービス名を、IPアドレスやポート番号で表示する

ping, traceroute,
tracepathは、
ICMPプロトコルを
使用する

