パナソニックITカレッジ ビジカル BUSICUL

Linux Professional Institute Japan LPI-JAPAN

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)

市川雅士

2012年6月24日

LPICとは

「Linux Professional Institute Certification」
の略称で、
特定非営利活動法人/Linux技術者認定機関
「LPI」の実施する、

Linux技術者認定試験

Linux Professional Institute

LPICの特徴

パナソニックITカレッジ ビジカル BUSICUL

- 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。

試験範囲

パナソニックITカレッジ ビジカル BUSICUL

http://www.lpi.or.jp/exam/index.shtml

101試験

主題101：システムアーキテクチャ

主題102：Linuxのインストールと
パッケージ管理

主題103：GNUとUnixのコマンド

主題104：デバイス、
Linuxファイルシステム、
ファイルシステム階層標準

102試験

主題105：シェル、スクリプト、
およびデータ管理

主題106：ユーザインターフェイスと
デスクトップ

主題107：管理業務

主題108：重要なシステムサービス

主題109：ネットワークの基礎

主題110：セキュリティ

Linux Professional Institute

101試験：出題範囲の詳細

パナソニックITカレッジ ビジカル BUSICUL

- LPICレベル1認定には、この試験が必須である。これは、Linuxのすべてのディストリビューションにわたって共通する、Linux技術者にとって必要な基本的な技能をカバーしている。

- 出題範囲の重要度が3の場合、試験ではその出題範囲に関連する問題が3題出題されます。

Linux Professional Institute

本日解説するポイント ： 101試験

パナソニックITカレッジ ビジカル BUSICUL

- 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

Linux Professional Institute

本日解説するポイント ： 102試験

パナソニックITカレッジ ビジカル BUSICUL

- 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

重要度：4

説明　　コマンド行を使用して、シェルおよびコマンドと対話する。
この目標は、bashシェルを使用することを想定している。

主要な知識範囲

- 1つのシェルコマンドおよび1行のコマンドシーケンスを使用して、コマンドラインでの基本的な作業を行う
- 定義することを含めたシェル変数の使用と変更、環境変数の参照とエクスポート
- コマンド履歴の使用と編集
- 定義済みパス内に存在するコマンドおよび存在しないコマンドの呼び出し

103.1 コマンド行で操作する

パナソニックITカレッジ ビジカル BUSICUL

重要なファイル、用語、ユーティリティ

- ✓ .
- ✓ bash
- ✓ echo
- ✓ env
- ✓ exec
- ✓ export
- ✓ pwd
- ✓ set
- ✓ unset
- ✓ man
- ✓ uname
- ✓ history

Linux Professional Institute

シェル変数・環境変数

パナソニックITカレッジ ビジカル BUSICUL

- set

  シェル変数・環境変数の両方表示

- env

  環境変数のみ表示

- unset

  変数の削除

- export

  環境変数にする

コマンド履歴

- history　　履歴一覧
- history –c 履歴の消去

↑	1つ前に入力したコマンドを表示（続けて押せば遡っていく）
↓	1つ後に入力したコマンドを表示
!n	historyの表示結果でn番のコマンドを実行
!-n	n個前に入力したコマンドを実行
!!	直前に入力したコマンドを再実行
!文字列	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
!?文字列?	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
^文字列1^文字列2^	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
:p	:pをつけると、コマンドを実行せずに表示のみ行う

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。
- 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。

103.3 基本的なファイル管理を行う

パナソニックITカレッジ ビジカル BUSICUL

重要度：4

説明　ファイルおよびディレクトリを管理するための基本的なLinuxコマンドを使用する。

主要な知識範囲

- 個々のファイルおよびディレクトリをコピー、移動、削除する
- 複数のファイルおよびディレクトリを再帰的にコピーする
- ファイルおよびディレクトリを再帰的に削除する
- 基本的なものから高度なものまで、ワイルドカード規則をコマンドで使用する
- findを使用して、種類、サイズ、または時刻を基にファイルを見つけて操作する
- tar、cpioおよびddの使用方法

103.3 基本的なファイル管理を行う

パナソニックITカレッジ ビジカル BUSICUL

重要なファイル、用語、ユーティリティ

- ✓ cp
- ✓ find
- ✓ mkdir
- ✓ mv
- ✓ ls
- ✓ rm
- ✓ rmdir
- ✓ touch
- ✓ tar
- ✓ cpio
- ✓ dd
- ✓ file
- ✓ gzip
- ✓ gunzip
- ✓ bzip2
- ✓ ファイルの展開

ファイル操作

- cp　　コピー
- mv　　移動(名前の変更も可能)
- rm　　消去

Linux Professional Institute

ディレクトリ操作

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

Linux Professional Institute

再帰的

- 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。

➢ rm –r ディレクトリ名

で、ディレクトリ内部をすべて削除できる!

ワイルドカード規則

(ブラケット外部の) '? ' はあらゆる単一の文字にマッチする。
(ブラケット外部の) '*' はあらゆる文字列にマッチする。空文字列 (empty string) にもマッチする。

文字クラス (character class)

"*[...]*" と言う表記は、先頭の '[' に続く最初の文字が '!' で なければ、ブラケットの中に含まれている文字のどれか一つにマッチする。ブラケットの内部に含まれる文字列は空であってはならない。したがって ']' も最初の文字に指定すればブラケットの内部に含めることが できる (つまり "*[][!]*" は '[', ']', '!' の 3 文字のどれかにマッチする)。

領域指定 (range)

特殊な表記法が一つ存在する。‘-’ を挟む二つの文字は領域指定となる。(つまり “*[A-Fa-f0-9]*” は “*[ABCDEFabcdef0123456789]*” と等価となる。) ‘-’ 文字そのものを入れたい場合は、ブラケットの先頭または最後の文字に指定すればよい。(つまり “*[]-]*” は二つの文字 ‘]’ と ‘-’ にマッチし、

"*[--0]*" は '-', '.', '0' の 3 文字にマッチする。この間の '/' にはマッチしない。)

補集合 (complementation)

"*[!...]*" と言う表記は、ブラケットの内部に含まれない単一の文字にマッチする (ただし先頭にある '!' は除外)。(つまり "*[!]a-]*" は ']', 'a', '-' 以外のすべての文字の、どれか一つにマッチする。)

バックスラッシュ ‘＼’ を前置すれば、‘?’, ‘*’, ‘[’ は通常の文字として扱われる。またはシェルのコマンドラインの一部に指定する場合は、クォートで囲っても同じ効果が得られる。ブラケットの内部では、これらの文字はその文字自身だけを意味する。すなわち “*[[?* ＼]*” は ‘[’, ‘?’, ‘*’, ‘＼’ のどれか

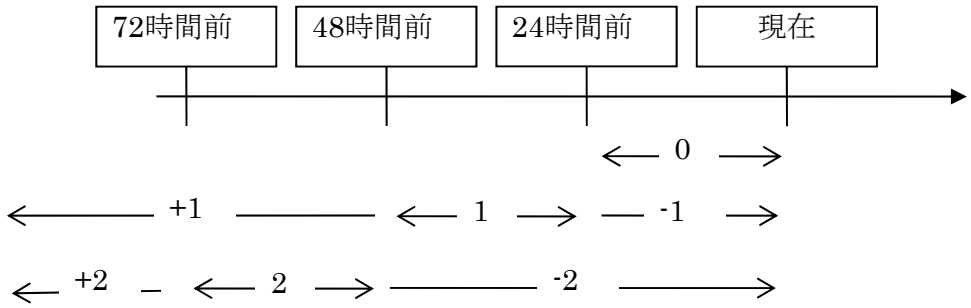
1文字にマッチする。

find 1/2

find 基点ディレクトリ 検索条件 [アクション]

[検索条件]

- **-name** ファイル名
  指定したファイルを検索。ファイル名にワイルドカードを使う場合は、ファイル名全体を””で囲む。
  find . -name “sample*”
- **-mtime n**
  (n+1)x24時間前 から **nx24時間前** の間に変更(書き込み)があったファイル。
  + をつけるとそれよりも過去、
  - をつけるとそれより直近になります。



find . –mtime -1 または find . –mtime 0
カレント・ディレクトリ以下で、
24時間以内に変更があったファイルを検索
find . –mtime +7
1週間以上(=168時間以内に)
変更がなかったファイルを検索

find 2/2

find 基点ディレクトリ 検索条件 [アクション]

– **-size** サイズ

単位を省略するとブロック単位(1ブロック=512バイト)、cはバイト単位、kはキロバイト単位。+をつけるとより大きい、-をつけるとより小さいの意味。

find . -size +200

カレント・ディレクトリ以下で、200ブロックより大きいサイズのファイルを検索

find . -size -200c

カレント・ディレクトリ以下で、200バイトより小さいサイズのファイルを検索

[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と同じ内容)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)
- **-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)

tar

- 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
  - 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（デバイス）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=boot.img of=/dev/fd0
    - 　　dd if=/dev/zero of=nulldata bs=512 count=1

103.4 ストリーム、パイプ、リダイレクトを使う

重要度：4

説明　　テキストデータを効果的に処理するためにストリームのリダイレクト　　や接続をする。この作業には標準入力、標準出力、標準エラー出　　力へのリダイレクト、あるコマンドの出力を別のコマンドの入力にパ　　イプする、あるコマンドの出力を別のコマンドの引数として使用する、
　　出力を標準出力とファイルの両方に送るといったことが含まれる。

主要な知識範囲

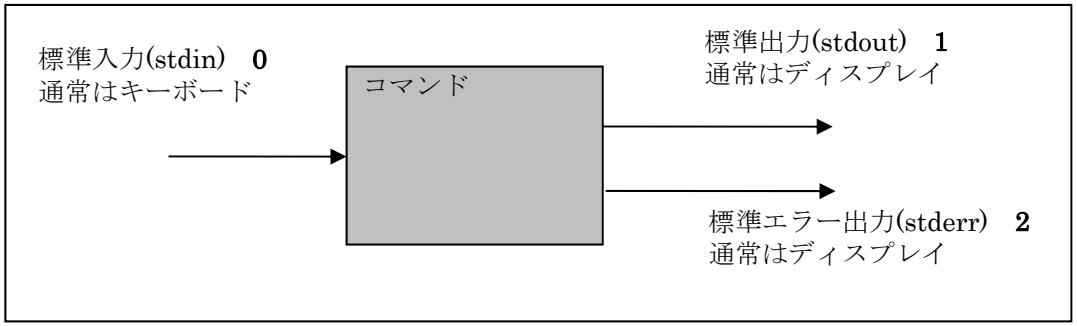
- 標準入力、標準出力、標準エラー出力をリダイレクトする
- あるコマンドの出力を別のコマンドの入力にパイプする
- あるコマンドの出力を別のコマンドの引数として使用する
- 出力を標準出力とファイルの両方に送る

重要なファイル、用語、ユーティリティ

tee

xargs

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)に出力します。

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います。

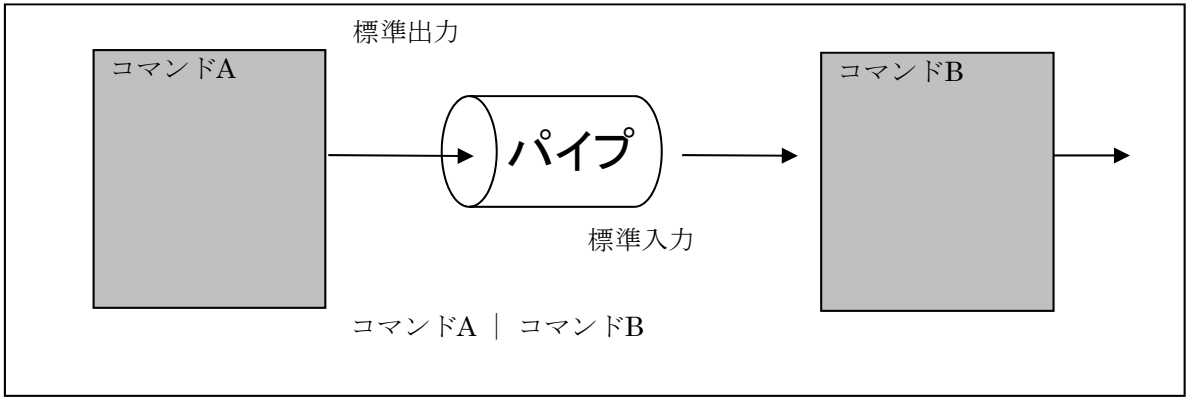
標準入力、標準出力などを切り替えることを、**リダイレクト**と呼びます。

標準入力の切り替え　<

標準出力の切り替え　>(上書き)　>>(追記)

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。



sort と uniq

- ファイルから重複している行を見つける。

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

- 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）　|　tee ファイル名

前のコマンドの実行結果（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。

xargs

```
% find . -name “*.c” –exec grep hogehoge {}   ;   \

% find . -name “*.c” | xargs grep hogehoge
```

103.5 プロセスを生成、監視、終了する

重要度：4

説明　基本的なプロセス管理を行う。

主要な知識範囲

- ジョブをフォアグラウンドやバックグラウンドで実行する
- ログアウト後にも実行が継続されるようにプログラムにシグナルを送信する
- 活動中のプロセスを監視する
- プロセス群を選択し、並び替えて表示する
- プロセスにシグナルを送信する

103.5 プロセスを生成、監視、終了する

パナソニックITカレッジ ビジカル BUSICUL

重要なファイル、用語、ユーティリティ

- ✓ &
- ✓ bg
- ✓ fg
- ✓ jobs
- ✓ kill
- ✓ nohup
- ✓ ps
- ✓ top
- ✓ free
- ✓ uptime
- ✓ killall

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。

ジョブの管理

コマンド名 &	バックグラウンドでジョブを起動
jobs	現在稼動中のジョブ一覧を表示 （ジョブ番号、ステータス、ジョブ名）の順
Ctrl+Z	フォアグラウンドジョブをバックグラウンドに移す ジョブは一時停止する

bg [%ジョブ番号] バックグラウンドで停止しているジョブを再開

fg [%ジョブ番号] バックグラウンドジョブをフォアグラウンドへ移動

プロセス

- Linux上での、処理の単位。
  (ジョブ ≧ 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。

プロセスの管理

- ps　現在の端末で自分が実行したプロセスを表示
- ps a　　ほかのユーザーのプロセスも表示
- ps x　　端末を持たないプロセスも表示
- ps ax　　システムで稼動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（ユーザー名など）
- 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)

プロセスにシグナルを送る

パナソニックITカレッジ ビジカル BUSICUL

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

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

重要度：5

説明　　ユーザアカウントを追加、削除、一時停止、変更する。

主要な知識範囲

- ユーザおよびグループを追加、変更、削除する
- パスワード/グループデータベースにある ユーザ/グループ情報を管理する
- 特別な目的を持つ制限付きのアカウントの作成と管理

重要なファイル、用語、ユーティリティ

- ✓ /etc/passwd
- ✓ /etc/shadow
- ✓ /etc/group
- ✓ /etc/skel
- ✓ chage
- ✓ groupadd
- ✓ groupdel
- ✓ groupmod
- ✓ passwd
- ✓ useradd
- ✓ userdel
- ✓ usermod

ユーザ管理

- useradd ユーザ名　　ユーザ追加
- userdel –r ユーザ名　　ユーザ削除
- usermod –L ユーザ名　　ユーザー時停止
  (解除は、–U)

/etc/passwd
/etc/shadow　} に保存。

/etc/skel

- 新規ユーザのホームディレクトリの
  テンプレート(ドットファイル等を仕込んでおく)

Linux Professional Institute

105.1 シェル環境のカスタマイズと使用

重要度：4

説明　ユーザの要求に応じてシェル環境をカスタマイズする。
全体のプロファイルおよびユーザのプロファイルを
変更する。

主要な知識範囲

- ログイン時または新しいシェルを生成したときに、
環境変数（PATHなど）を設定する
- よく使用する一連のコマンド用にBASHの関数を作成する
- 新しいユーザアカウント用のスケルトンディレクトリを保守する
- コマンドサーチパスを適切なディレクトリに設定する

105.1 シェル環境のカスタマイズと使用

重要なファイル、用語、ユーティリティ

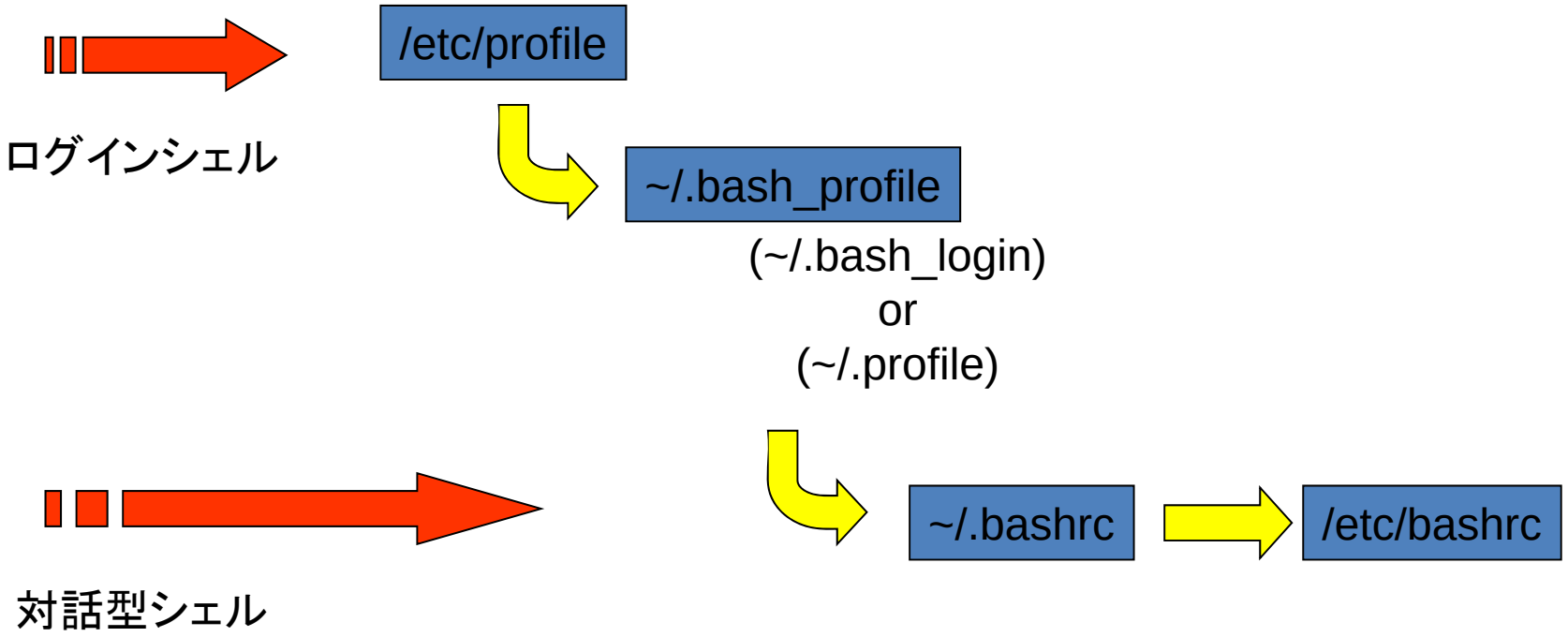
- ✓ /etc/profile
- ✓ env
- ✓ export
- ✓ set
- ✓ unset
- ✓ ~/.bash_profile
- ✓ ~/.bash_login
- ✓ ~/.profile
- ✓ ~/.bashrc
- ✓ ~/.bash_logout
- ✓ function
- ✓ alias
- ✓ lists

bashの設定ファイル

パナソニックITカレッジ ビジカル BUSICUL

	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

設定ファイル実行順



bashの関数

- 書式

  function 関数名() { コマンド; }

- alias よりも複雑な”内部コマンド”を定義可

alias のまね...

function ls() { command ls -CF --color=tty $@ ; }

カスタマイズ?

- 環境変数PATHに追加

  PATH=$PATH:/home/...
- alias の設定
  - 設定

    alias ls='ls –l'
  - 解除

    unalias ls
  - 一時解除

    ＼ls

105.2 簡単なスクリプトをカスタマイズまたは作成する

重要度：4

説明　既存のスクリプトをカスタマイズするか、簡単なBASHスクリプトを新規作成する。

主要な知識範囲

- 標準的なshの書式（ループ、テスト）を使用する
- コマンド置換を使用する
- コマンドによって返される、成功または失敗を示す戻り値やその他の情報をテストする
- 条件に応じて、スーパーユーザにメールを送信する
- 先頭行(#!)を利用して、適切なスクリプトインタープリターを選択する
- スクリプトの位置、所有権、実行権、SUID権を管理する

重要なファイル、用語、ユーティリティ

- ✓ for
- ✓ while
- ✓ test
- ✓ if
- ✓ read
- ✓ seq

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

条件文

[$? –eq 0] 手前のコマンドが成功なら

- s1 = s2　　文字列 s1 と s2 が同じであれば真
- s1 != s2　　文字列 s1 と s2 が同一でなければ真
- s1 < s2　　文字列 s1 が文字列 s2 に対し、ASCII 順で前なら真
- s1 > s2　　文字列 s1 が文字列 s2 に対し、ASCII 順で後なら真

- n1 -eq n2　　整数 n1 と n2 が等しければ真
- n1 -ne n2　　整数 n1 と n2 が等しくなければ真
- n1 -gt n2　　整数 n1 が n2 がより大きければ真
- n1 -ge n2　　整数 n1 が n2 より大きいか等しければ真
- n1 -lt n2　　整数 n1 が n2 より小さければ真
- n1 -le n2　　整数 n1 が n2 より小さいか等しければ真

ネットワーク関連 1/2

- 109.1 インターネットプロトコルの基礎
  - ネットワークマスクについて理解していることを示す
  - プライベートとパブリックのドット区切り形式のIPアドレスの違いを知っている
  - デフォルトルートを設定する
  - 一般的なTCPおよびUDPのポート(20、21、22、23、25、53、80、110、119、139、143、161、443、465、993、995)について知っている
  - UDP、TCP、およびICMPの違いや主な機能について知っている
  - IPv4とIPV6の主な違いについて知っている

ネットワーク関連 2/2

- 109.2 基本的なネットワーク構成
  - ネットワークインターフェイスの設定を手作業および自動で行う
  - ホストの基本的なTCP/IP設定
- 109.3 基本的なネットワークの問題解決
  - ネットワークインターフェイスおよびルーティングテーブルを手作業および自動的に設定する(これには、ネットワークインターフェイスの追加、起動、停止、再起動、削除、および再設定が含まれる)
  - ルーティングテーブルを変更、参照、設定し、不適切なデフォルトルート設定を手作業で訂正する
  - ネットワーク構成に関連する問題をデバッグする

ホストの基本的なTCP/IPを構成 (1/2)

- ホスト名
  /etc/HOSTNAME ないし /etc/hostname
  (RedHat系は、/etc/sysconfig/network)
- 名前解決
  - 問い合わせ順
    /etc/nsswitch.conf もしくは、/etc/host.conf
  - ファイルによる指定
    /etc/hosts
  - DNS 参照先の設定
    /etc/resolv.conf

ホストの基本的なTCP/IPを構成 (2/2)

パナソニックITカレッジ ビジカル BUSICUL

- ネットワークインターフェース
  - 設定ファイル

    /etc/sysconfig/network-scripts (RedHat 系)

    /etc/network/interfaces (Debian)

  - 確認

    ifconfig

  - 一時設定

    ifconfig eth0 192.168.0.20 netmask 255.255.255.0

DHCP設定は、上記ファイルに
BOOTPROTO=dhcp(RedHat系)
iface eth0 inet dhcp（Debian）　等と、書く

ルーティングテーブル

パナソニックITカレッジ ビジカル BUSICUL

- route

  ルーティングテーブルの表示

- route add

  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

- route del

  route add –net 192.168.10.0 netmask 255.255.255.0 gw 192.168.2.254

診断コマンド

- ping

  疎通確認(トラブルの切り分け)

- traceroute

  経路確認

- netstat

  ネットワーク情報

(DNSトラブルのときは、-n で名前解決を抑止)

その他実際のトラブルシューテイング

- iptables によるパケットフィルタ
  - iptables -L -n などで確認
- 各サーバの設定ファイル、ログ等の確認
- inetd 使用の場合
  - /etc/hosts.allow
  - /etc/hosts.deny 設定の確認
- netstat でソケットの使用状況を確認
  - netstat -at
- tcpdump で、通信をモニタ
  - tcpdump –X –i eth0 port 80