

ゼロからのLinux、 あなたはどう成功する

株式会社びぎねっと

代表取締役社長兼CEO

宮原 徹 (tmiyahar@Beginet)



- 株式会社びぎねっと 代表取締役社長兼CEO
 - 日本仮想化技術株式会社 代表取締役社長兼CEOでもある
- Linux・オープンソースに関するIT技術者教育を中心にビジネスを展開
- 現在は仮想化技術に関するソリューション提案を行っている (VMware・Xenなど)
- LPI-Japan発行 メールマガジン 執筆者



- Linux技術を身につけるには
- Linux基礎技術を学ぶには
 - LPIC (Linux技術者認定試験) の概要
 - ポイント解説
 - Linuxの学習環境作り
 - 102.1 ハードディスクの配置を設計する
 - 103.1 コマンド行で操作する
 - 103.4 ストリーム、パイプ、リダイレクトを使う

LPIの概要を理解し、Linuxの学習を
スタートするためのポイントを掴む

Linux学習の指針



■ 目標の設定

- 実務に役立つスキルの習得
- 客観的なスキルの証明

} バランスよく両立

■ 学習環境の確保

- 無知による無駄を無くす
- 意味のない学習はしない

} 時間を無駄にしない



■ 明確な目標を立てる

- 漠然とした目標: Linuxを使えるようになる
- 明確な目標: Linuxを使ってWebサーバを構築できるようになる

■ 手段と目的をはっきりとさせる

- 手段: Linuxを使う・Webサーバを動かす
- 目的: Webサービスを提供する



- Who? (誰が?)
 - Why? (なぜ?)
 - When? (いつ?)
 - Where? (どこで?)
 - What? (何を?)
 - How? (どのように?)
-
- Linux学習の目的を明確にする



■身につけたスキルの客観的な証明

- スキルレベルの「ものさし」
- 「Linuxできます」では、「どれぐらい」できるのか分からない

■LPICの場合

- レベル1: Linuxの基本的な操作と管理
- レベル2: サーバとネットワークの構築
- レベル3: Linux以外のOSとの統合



■教科書・参考書の確保

- 書籍代はケチるな
- 同一テーマで何冊も買う

■学習環境の構築

- 学習環境構築そのものが勉強

■基礎的なLinuxの操作方法を学習

- インストールからコマンド操作、Linuxシステム管理基礎レベル
- カバーされない範囲については、別途周辺情報で知識を補う(PC自作なども効果的)



全部がLinuxの入門書というわけではありませんが、関連書籍まで含めると、これぐらいにはなります。古くなった本は適宜捨てて、新しい本も随時購入。

この辺りにもう1段



- 今使える環境があるならまずはその環境で学習
 - 基本的にディストリビューションに依存しない
- 色々試してみよう
 - CentOS、openSUSE、Debian GNU/Linux、Ubuntu 等々、無償で試せるディストリビューションは一通り
- 最終的な知識の確認は2.4ベースのディストリビューションも使ってみる
 - 今でも動いている古い環境も多数
 - 個人的にはRed Hat Linux 7.3がオススメ



- 快適な学習環境作りを必死になって考える
 - 自分なりのカラーを出せるか
 - 「試行錯誤」と「育てていく」感覚
- ハードウェアに強くなろう
 - 自作PCのススメ
 - 業務用サーバーを中古で買うのも良い
- 仮想マシンを活用(後述)





- 仮想マシンを使えば、物理マシンは少なくて済む
- CPUとメモリが重要
 - CPUはマルチコア化しているので比較的楽
 - メモリは多めに必要なので、最低でも2GBは欲しい
- 活用を考えると、物理マシンの経験は重要
 - 「鶏と卵」の関係
 - 特にネットワーク周りは分かりにくい
 - とりあえずブリッジがNATを使っておく

LPIC (Linux技術者認定試験) の 概要



■ オープンソース

- 世界的なコミュニティで形成

■ ベンダーニュートラル

- 様々な環境で知識が活かせる

■ 本質的な問題

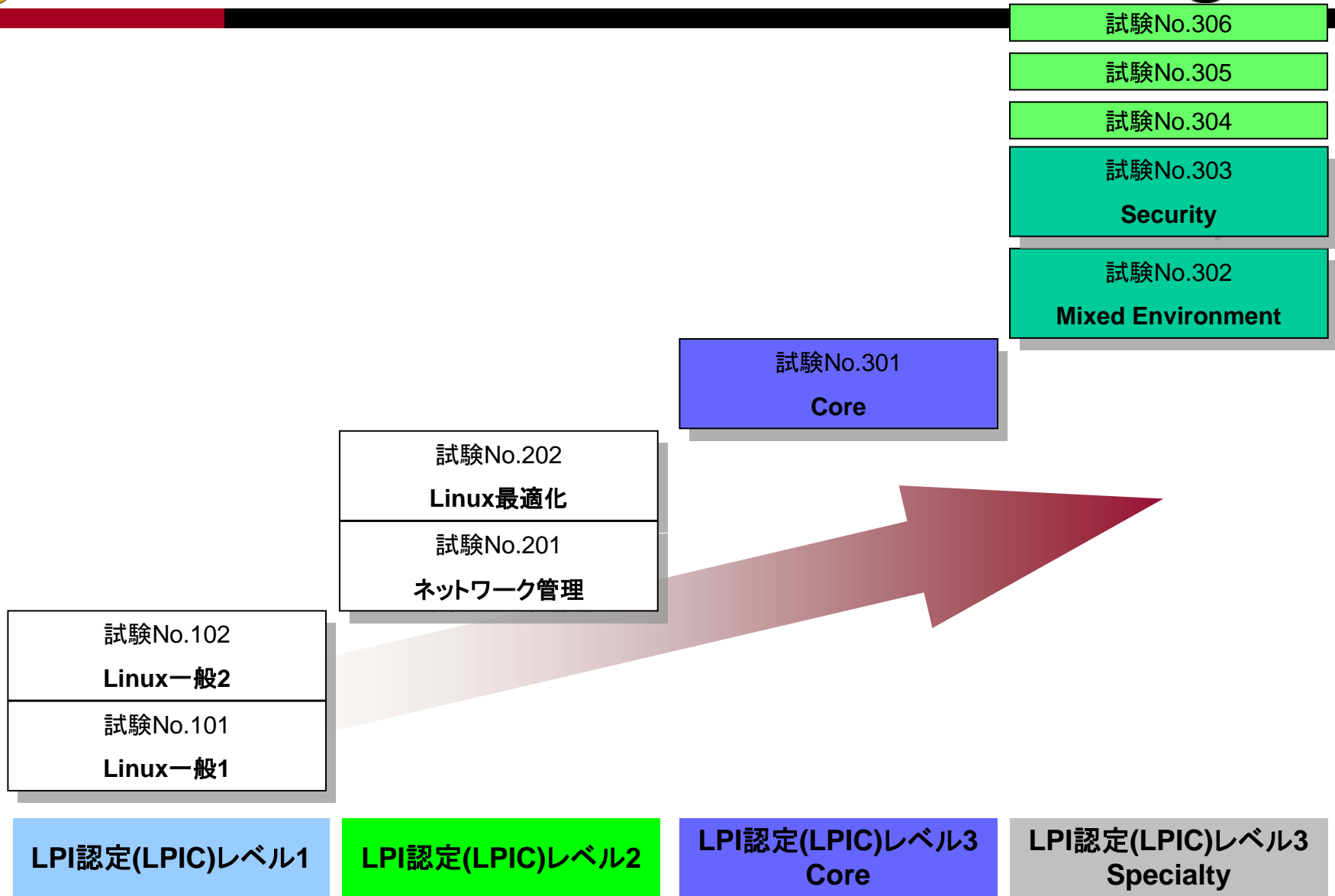
- 技術的な本質を見極める問題

■ 広範囲に渡る出題

- 技術レベルの再認識

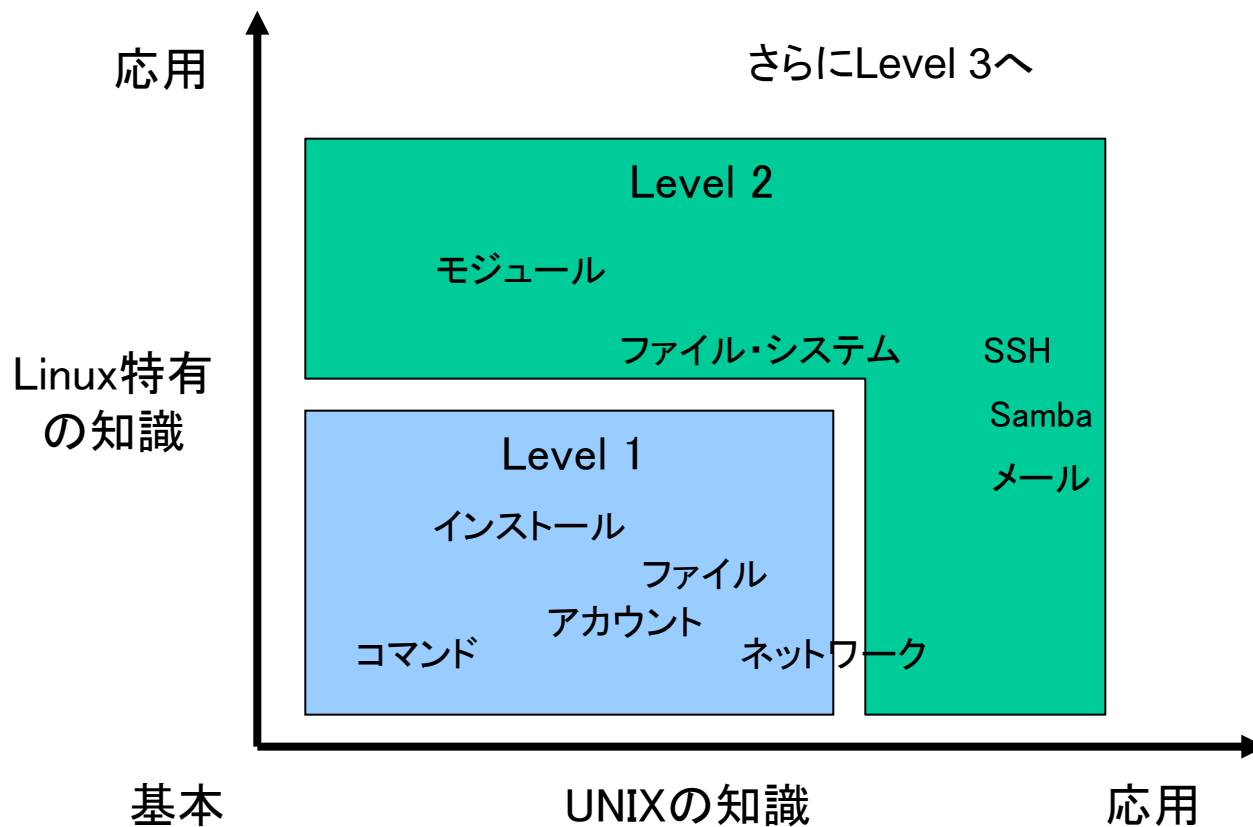


試験の構成





試験のカバーしている範囲



LPICレベル1 試験出題範囲と対策



- 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



- 主題105: シェル、スクリプト、およびデータ管理
- 主題106: ユーザー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



■ 出題範囲に幅がある

- 知らないポイントを無くす
- 得意なポイントを作る

■ うろ覚え、ケアレスミスを減らしたい

- コマンドのオプションの意味をしっかりと
- 出題範囲詳細に出てくるファイルやコマンドはmanコマンドなどで調べておく



■ 出題範囲をしっかりと把握

- 関連キーワードはすべて調べる

■ 基礎的なLinuxの操作方法を学習

- インストールからコマンド操作、Linuxシステム管理基礎レベル(ユーザ管理等)
- カバーされない範囲については、別途周辺情報で知識を補う(PC自作なども効果的)

ポイント解説



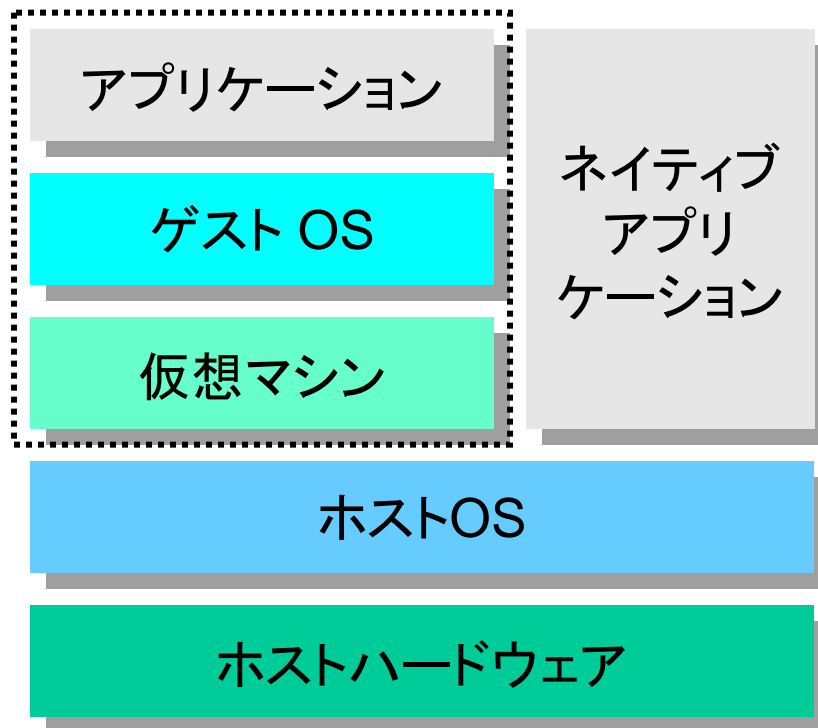
- Linuxの学習環境作り
- 102.1 ハードディスクの配置を設計する
- 103.1 コマンド行で操作する
- 103.4 ストリーム、パイプ、リダイレクトを使う



- 理想を言えば、PCを2台〜3台用意
 - たとえばWindowsクライアントにLinuxサーバ
- HDD交換可能にする
 - 色々なディストリビューションを試せる
 - 再インストールも簡単
- 物理的な問題から仮想マシンで代替
 - LinuxやWindowsで無料で使えるものとしてVMware ServerやVirtualBox、XenやLinux KVMなど
 - MacならParallelsやVMware Fusion、VirtualBoxなど



仮想マシンとは



- ソフトウェアでもう1台のPCを再現する技術
- ホストOSにはLinuxやWindowsを使用
- 仮想マシン内で別のOSを実行可能
- HDDやメモリの許す限り、複数の仮想マシンを実行可能



- メモリは実際のPC同様に消費
 - ホストマシンにできるだけ沢山のメモリを搭載
- 仮想ハードディスクを使用する
 - ホストOS上のファイル＝仮想マシンのHDD
- ゲストOSのインストール元はISOイメージも可能
 - 光学式ドライブはホストOSと共用
 - ISOイメージからインストールも可能
- 仮想ネットワークの設定は用途に応じて
 - ブリッジ: 外部から接続可能・DHCP設定の場合にはDHCPサーバが必要
 - NAT: 外部から接続不可・DHCP設定の場合には仮想マシンソフトがDHCPサーバを準備



102.1 ハードディスクの配置を設計する **Beginet**

説明

Linuxシステムにおけるディスクパーティションの構成を設計する。

主要な知識範囲

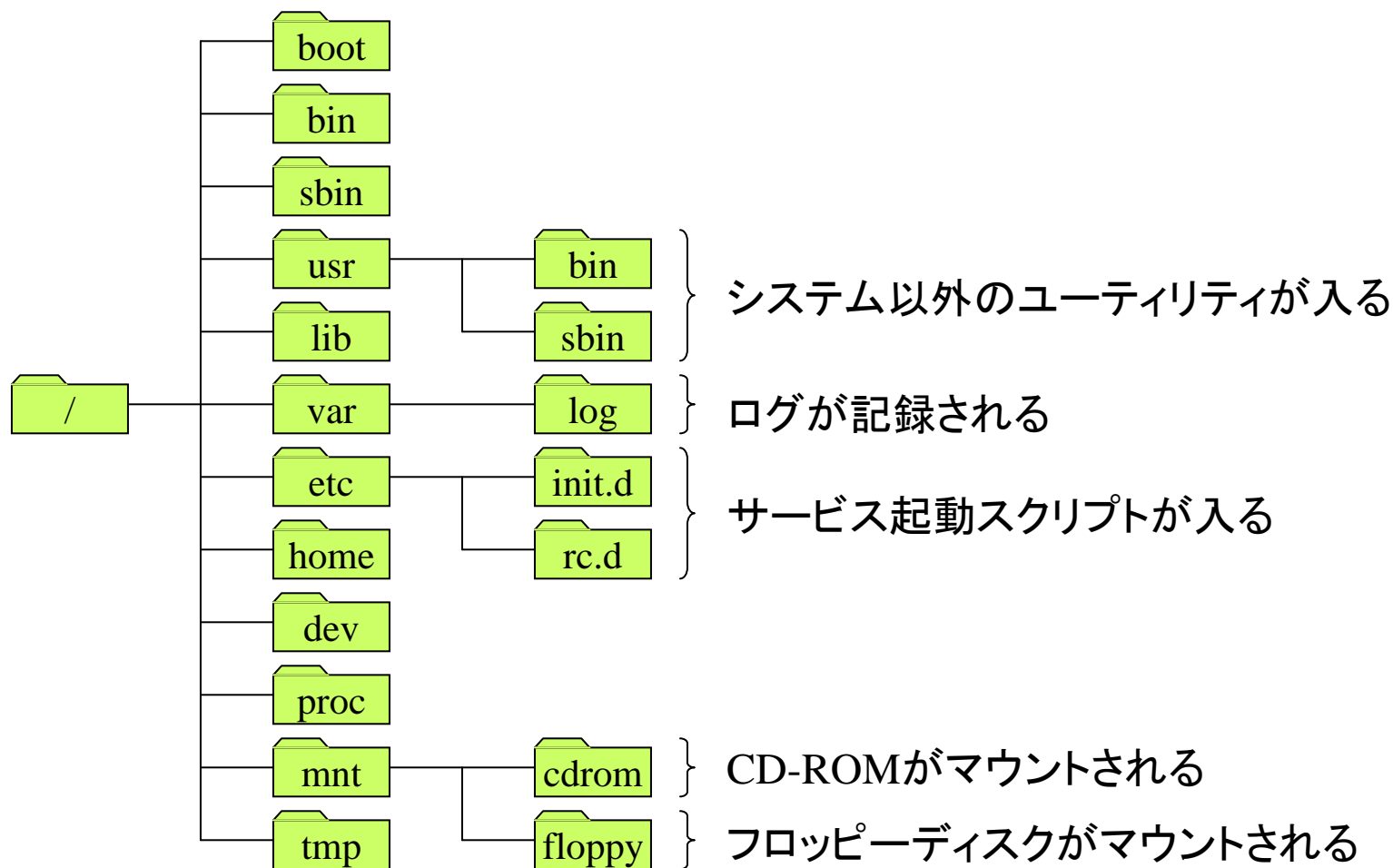
- ファイルシステムおよびスワップスペースを別々のディスクやパーティションに割り当てる
- システムを適切に使用できるように設計を調整する
- ブートに当たって、/bootパーティションがハードウェアアーキテクチャの要件を満たしていることを確認する

重要なファイル、用語、ユーティリティ

- / (root)ファイルシステム
- /varファイルシステム
- /homeファイルシステム
- スワップスペース
- マウントポイント
- パーティション



ディレクトリ名	役割
/boot	Linuxカーネルと起動関連ファイルを格納
/bin	Linuxの最も基本的なコマンドを格納
/sbin	Linuxのシステム管理に関するコマンドを格納
/usr	Linux上のアプリケーションソフトのファイルを格納
/lib	各種ライブラリファイルを格納
/var	ログやメールなどシステムが一時的にデータを格納
/etc	各種設定ファイルを格納
/home	一般ユーザーのホームディレクトリ
/dev	デバイスファイルの入っている特殊なディレクトリ
/proc	システムの各種稼動状態をファイルとして保持
/mnt	フロッピーディスクやCD-ROMをマウントする
/tmp	各種プログラムが利用する一時ディレクトリ





- / (ルート) ディレクトリ以外を別のディスクやパーティションに分離する
 - /home を分離して、**ユーザデータ**を分離する
 - /var を分離して、**システムデータ**を分離する
- ディレクトリを分離する理由
 - メンテナンス性の確保 (システムとデータの分離)
 - RAID (ミラー) による冗長性の確保
 - /home をミラーで構成
 - パフォーマンスの向上
 - /var を RAID 1+0 などで構成



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

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



■ コマンドの動作を細かく設定するためにオプションと引数をコマンドに与えます

- オプションはコマンドの設定項目を指定します
- 引数はファイル名などの具体的な値を指定します

■ コマンド基本文法

1. コマンド

2. コマンド 引数

3. コマンド 引数1 引数2

4. コマンド -オプション

5. コマンド -オプション 値

6. コマンド -オプション 値 引数 ...

← オプションが値を取る時は必ず値が必要



- コマンド実行時にディレクトリを指定しなかった場合、暗黙のうちに指定されるディレクトリ
- 「作業(ワーク)ディレクトリ」とも呼ばれる
- 確認には `pwd` コマンドを使用
 - プロンプトにもカレントディレクトリ名が表示される

コマンドラインによる操作上達の近道は
カレントディレクトリを意識すること



- 長いディレクトリ名入力を省くシェルの機能
- TABキーを押すと、ファイル名が前方一致検索され、候補が1つであれば補完される
 - 例) `cat /etc/sysconfig/network`と入力する
 1. `cat /e<TAB>`
 2. `cat /etc/sysco<TAB>`
 3. `cat /etc/sysconfig/ne<TAB>`
 4. `cat /etc/sysconfig/network`
- 補完候補が複数ある場合には、TABキーを2回押すと一覧表示される



- シェルはコマンド作業の履歴を取っている
- 呼び出して編集・再実行可能
- 呼び出し方法
 - カーソルの上下 順番に呼び出し
 - historyコマンド 履歴一覧の表示
 - !履歴番号 履歴番号のコマンドを再実行
 - !文字列 文字列で始まる最新コマンド
 - !! 直前のコマンドを再実行
 - コマンド !* 直前のコマンド引数を再利用



- システム全体の環境を設定する値を格納する変数
- 設定方法
 - A. シェル変数を設定し、exportする
 - 1. 変数=値
 - 2. export 変数
 - B. exportで直接設定する
 - 1. export 変数=値
- 環境変数の確認方法
 - envコマンドで環境変数とその値を一覧できる



- シェルでコマンドを実行する際に、実行するコマンドファイルが環境変数PATHで設定されたディレクトリから検索する(サーチパス)
- ディレクトリの中を、設定されている順に検索を行い、最初に見つかったコマンドファイルを実行する
- コマンドファイルを見つけられなかった場合にはコマンドを実行できない
- whichコマンドで、実際に実行されるコマンドファイルの位置を調べられる



su [-] [username]

- 「-」をつけると、ログインしたのと同じ状態になる
(環境変数が新たに設定される)
 - 「-」をつけないと、前のユーザーの環境変数を引き継ぐ(PATHなどが変わらない)
- usernameを省略すると、rootへの切り替えとなる
 - /sbin,/usr/sbin等へのPATH設定はrootのみ
 - su -でrootでログインしたのと同じ状態



■ manコマンド

- `man [section_number] command`
- マニュアルの種類によってセクションに分かれている
- 同じ名前で複数のセクションに分かれている場合には指定が必要。
- マニュアルファイルは `/usr/man`、あるいは `/usr/share/man` に入っている
- 環境変数 `LANG` が設定されている
と” `/usr/share/man/言語名`” 以下を優先的に参照する
 - `LANG=ja_JP.UTF-8` なら `/usr/share/man/ja` 以下を参照
 - `LANG` を設定しないと英語マニュアルを表示 (オススメ)



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

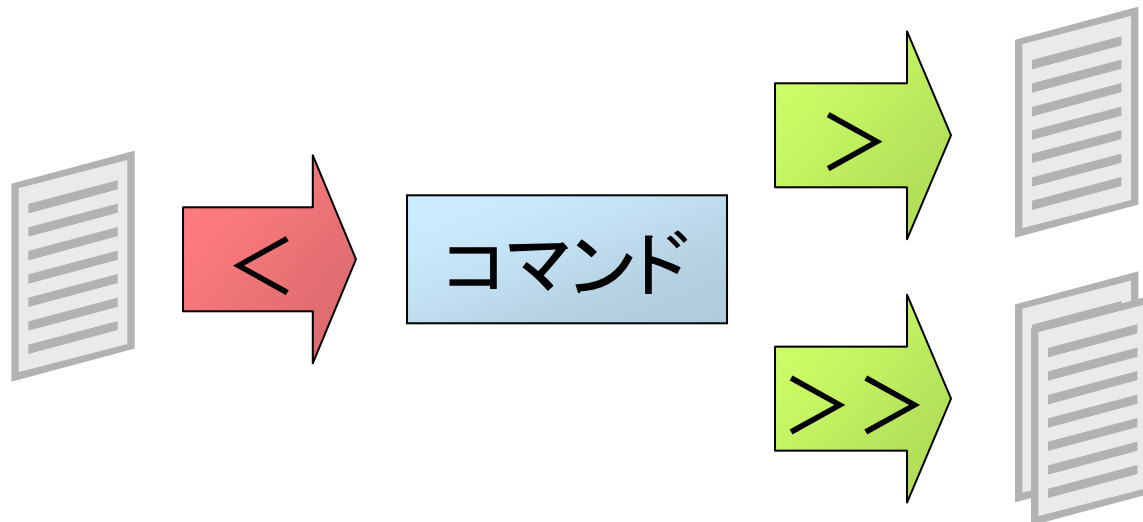


- コマンドは標準入力からデータを受け入れて、処理を行った後に標準出力へと結果を出力する
- エラーが発生した場合には、標準エラー出力へとエラーメッセージを出力する



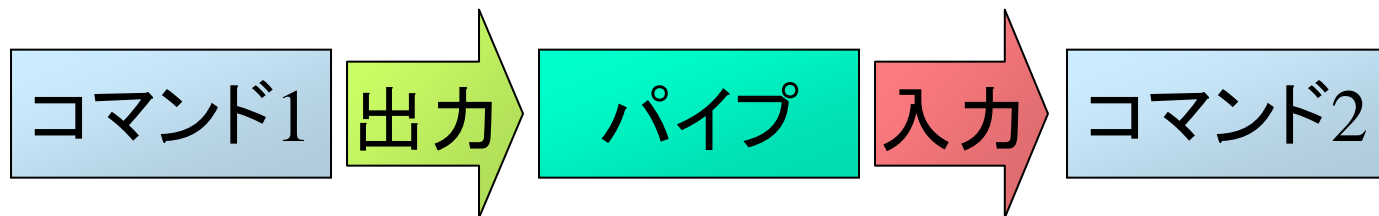


- 標準入出力とファイル間のやり取りを制御する
 - `command > file` 標準出力をファイルにリダイレクト
 - `command >> file` 標準出力を追加でリダイレクト
 - `command < file` ファイルを標準入力にリダイレクト





- あるコマンドの結果を、さらに他のコマンドで処理したい場合に使用
- command1の標準出力をcommand2の標準入力へ渡す
 - `command1 | command2`
 - `command1 > file; command2 < file`と同等





■ 書式

- `grep key [file]`
- `file`または標準入力から`key`を探し出す
- `key`には正規表現を指定できる

■ 正規表現

- `.` 任意の1文字
- `*` 直前の文字の0回以上の繰り返し
- `[文字]` 文字のいずれか1文字
- `^` 行頭
- `$` 行末
- `¥` 続くメタキャラクタを文字として処理(エスケープ)



■ 学習の目標を定めよう

- 例) Linuxを使ったWebサーバー構築

■ できるだけ沢山Linuxに触る

- コマンドライン中心に、苦勞なく触れる程度に
- 色々なディストリビューションを試してみる

■ 自分なりに説明できること

- 教えることが最高の学習



■『LPI通信』

- Linuxオープンソース関連ニュース
- Linuxの利用に役立つTipsやテクニック
- LPI合格者の声

■『LPIC Level2・Level3を受けてみよう！』

- Level2・Level3の例題解説
- Level2・Level3を取得するメリット、活用の事例など
- Level1をお持ちでない方にも役立ちます

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