

LPIC Level1技術解説無料セミナー

～LPIC Level1試験に向けての準備とポイント解説～

2013/8/24

株式会社エイチアイ
末永貴一



自己紹介

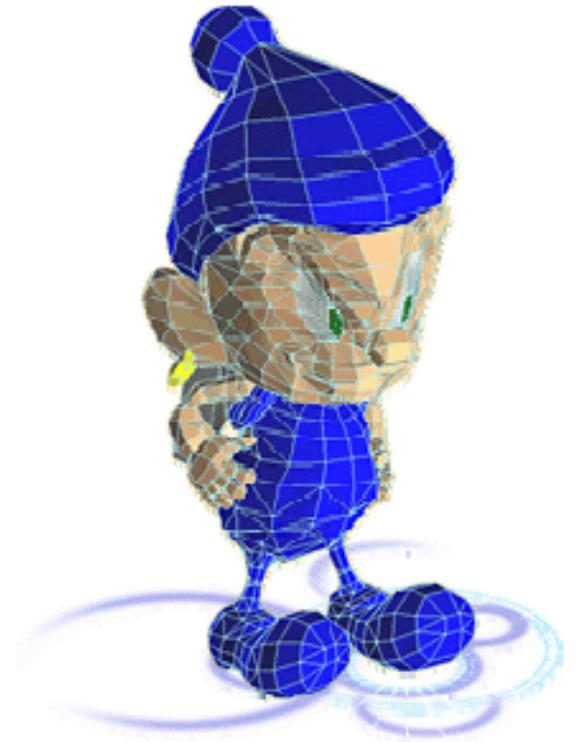
株式会社エイチアイ CE事業開発部 部長

– 組込み機器向けソフトウェアの研究開発

<http://www.hicorp.co.jp>

Linux関連文章の執筆

- LPIC Level1,Level2認定テキスト
- LPI試験 レベル1標準教科書（オーム社）
- LPI試験 レベル2標準教科書（オーム社）
- LPI-Japan コラム【Linux道場 入門編】
- @IT 「Linuxをいまから学ぶコツ教えます」
- @IT 「Linuxに触れよう」
- 日経Linux 「Xと次世代「Wayland」を知る」
など





本日のアジェンダ

1. LPICとは
2. LPICの特徴
3. LPIC Level1試験範囲と特徴
4. LPIC Level1試験の準備
5. LPIC Level1のポイント解説
6. その他の試験範囲へのアドバイス



LPIC試験とは…

- ニュートラルなLinux技術者資格試験

● 特徴

- ベンダーニュートラル
- コミュニティで問題を作成

<https://group.lpi.org/cgi-bin/publicwiki/view/Examdev/WebHome>

- 広範囲かつ様々なケースを想定した出題



LPICには3つLevelの試験がある

各レベルでLinux技術者としての位置づけを認定

- Level1 – ファーストレベルLinux専門家
- Level2 – アドバンストレベルLinux専門家
- Level3 – 市場価値の高いLinuxプロフェッショナル

※レベル1は、101試験と102試験、両方に合格すること。

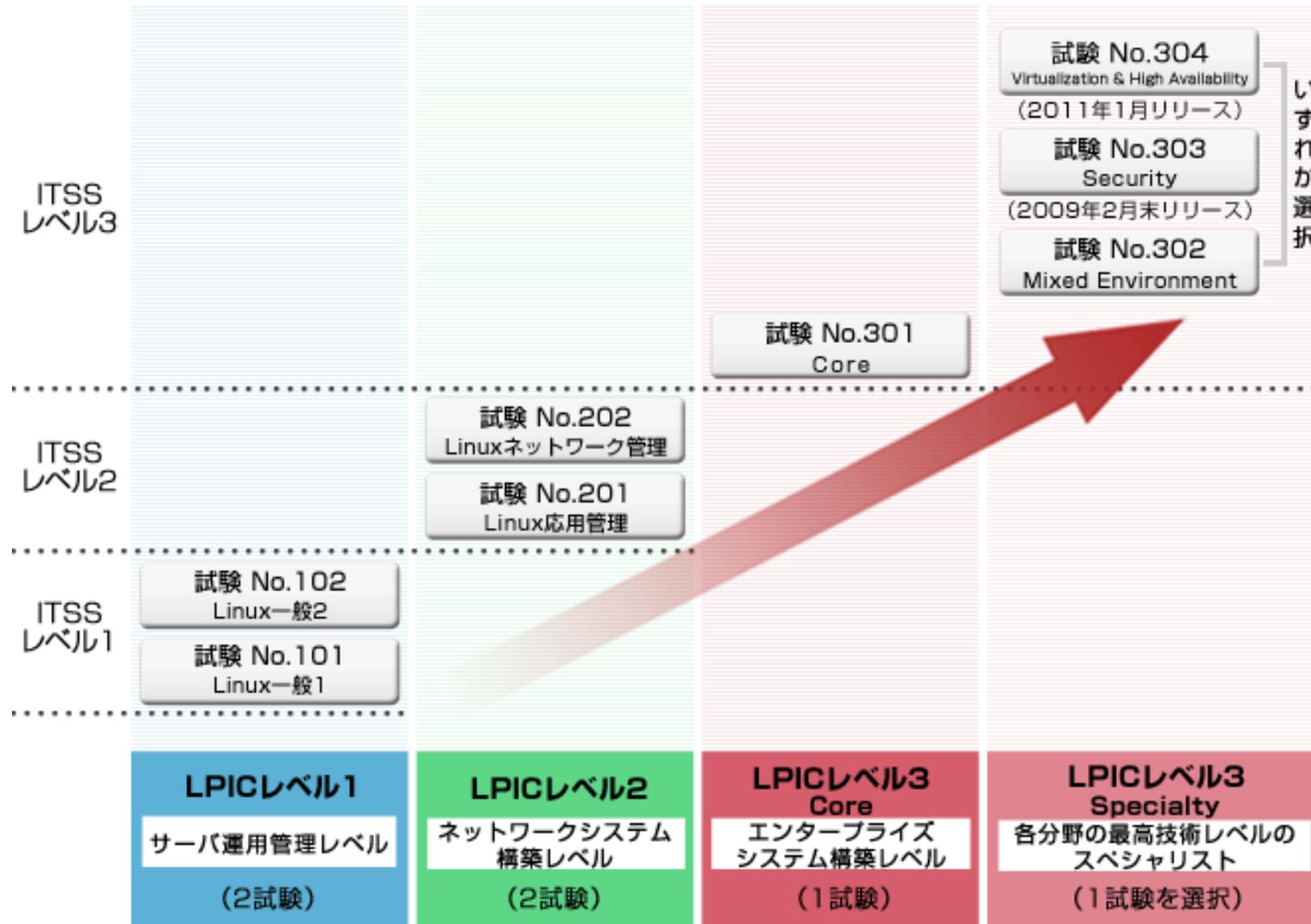
※レベル2は、201試験と202試験、両方に合格すること。

※レベル3は、CoreとSpecialtyの2段階がある。

レベル3 Coreは、301試験に合格すること。レベル3 Specialtyは、301試験に合格し、かつ302試験以降のいずれか1つに合格すること。



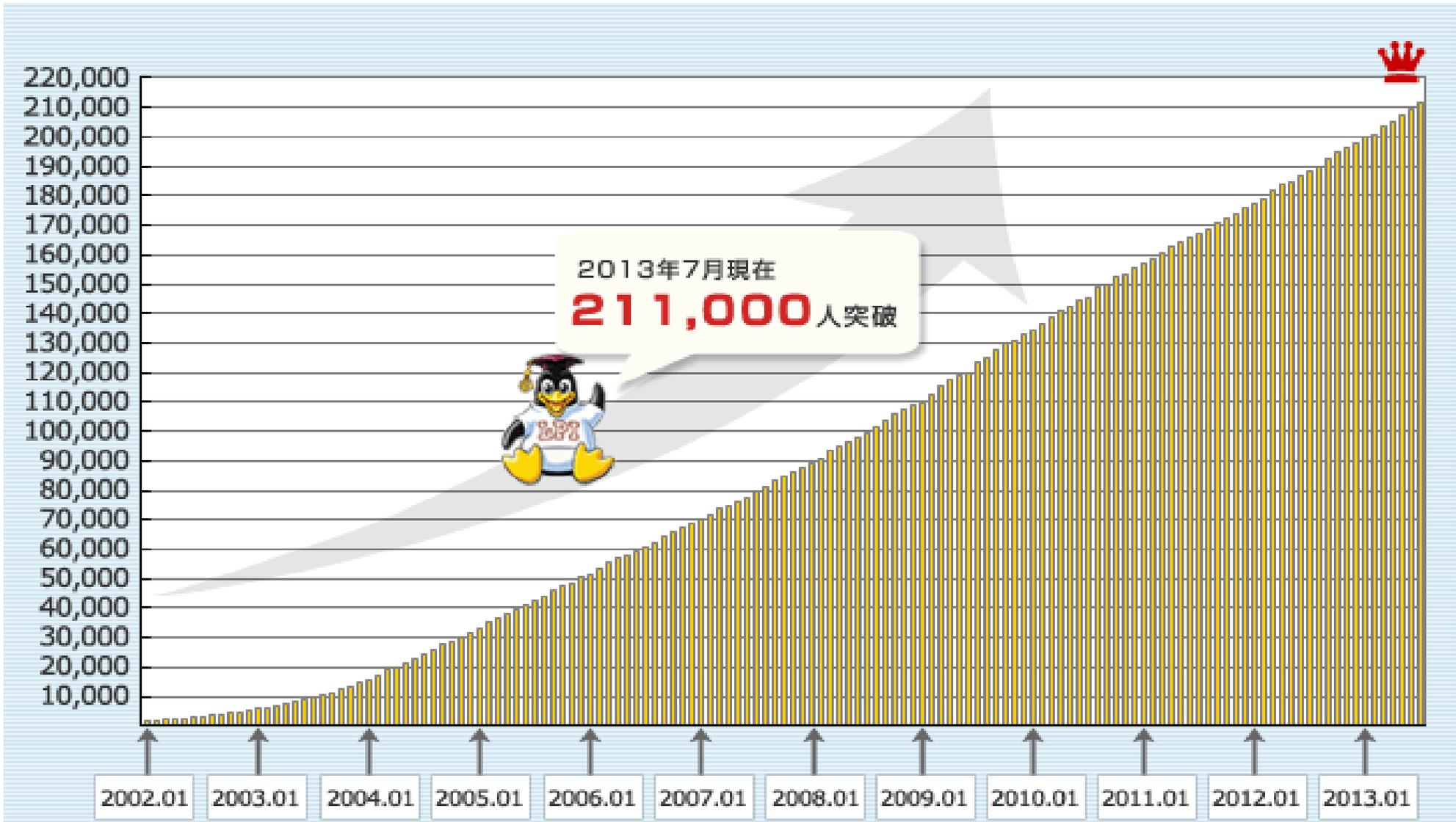
LPIC3つのLevel試験



※ 上位レベルの試験を受けるためには直下のレベルの資格が必要



■国内受験者数: 211,000人を突破

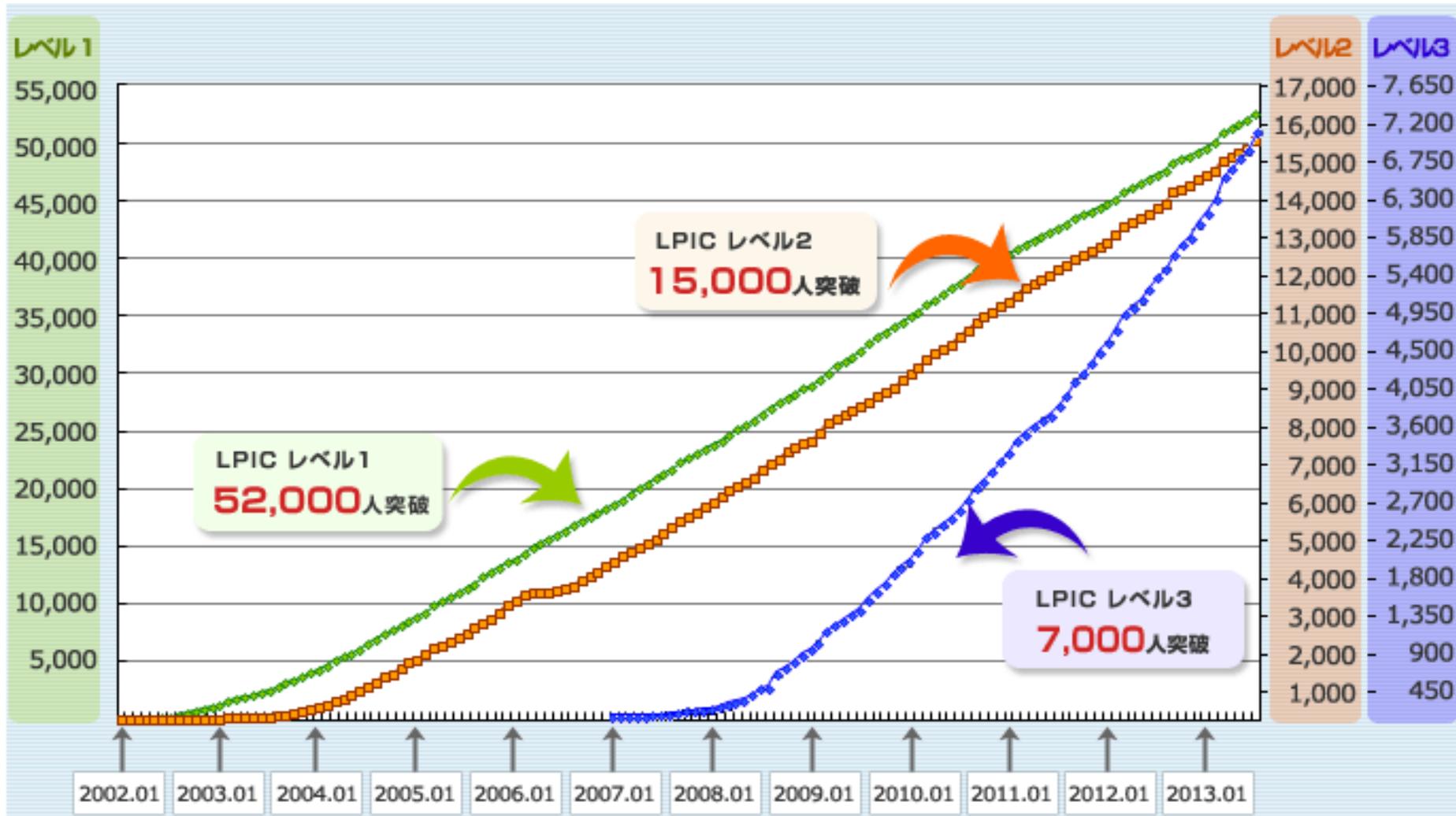




LPIC各レベルの認定者数



- レベル1: 52,000人、レベル2: 15,000人、レベル3: 7,000人
各レベルでそれぞれ突破(2013年7月現在)





2012年10月1日よりv3.5.0スタート！

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、及びデータ管理
- 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

主題には変化なし。テーマの追加と削除が行われる。



v3.5.0で見直されたテーマ

101、102試験の以下の改訂が行われた。

- 追加項目

- GRUB 2 (102.2)
- ext4 (104.1)
- systemdおよびUpstart (101.3)
- IPv6 (109.1)
- LVM (102.1)

- 削除項目

- LILO (102.2)

詳細はLPIC HPで確認。



- コンピュータの基礎的からネットワーク、セキュリティまで幅広い範囲をLinuxを題材として出題
- Linuxの知識もちろん、コンピュータ、ネットワークなどLinux自体の知識以外も問われる
- Linuxの知識も通常の操作や管理では意識しない部分や場合によってはあまり利用しない機能も含まれる
- シェルスクリプトを題材としたプログラム基礎

広範囲な知識が必要



- 出題範囲の詳細をよく確認する
 - 説明部分の知識を体系的に知る
 - 知識範囲は必ず実行する
 - キーワードは必ずチェック
- 勉強用の環境を構築する
 - 繰り返しインストールできる環境
 - 壊しても問題ない環境
 - すぐに試せる環境



- 基本的には専用PCがあったほうが望ましい
 - H/Wの知識なども含めて学ぶ必要がある
 - ネットワーク構築、サーバ構築が理解しやすい
- バーチャルマシンを利用する
 - VmwareやVirtual PCなどを利用する
 - 複数台のマシンを用意する必要ない
 - 手軽に環境構築可能
- コマンド実行のみなら、Cygwinを利用する
 - LinuxのコマンドがWindows上で利用できる
 - ただし実行結果の相違やLinux特有の情報がない

何でも試せる物理的環境が望ましい

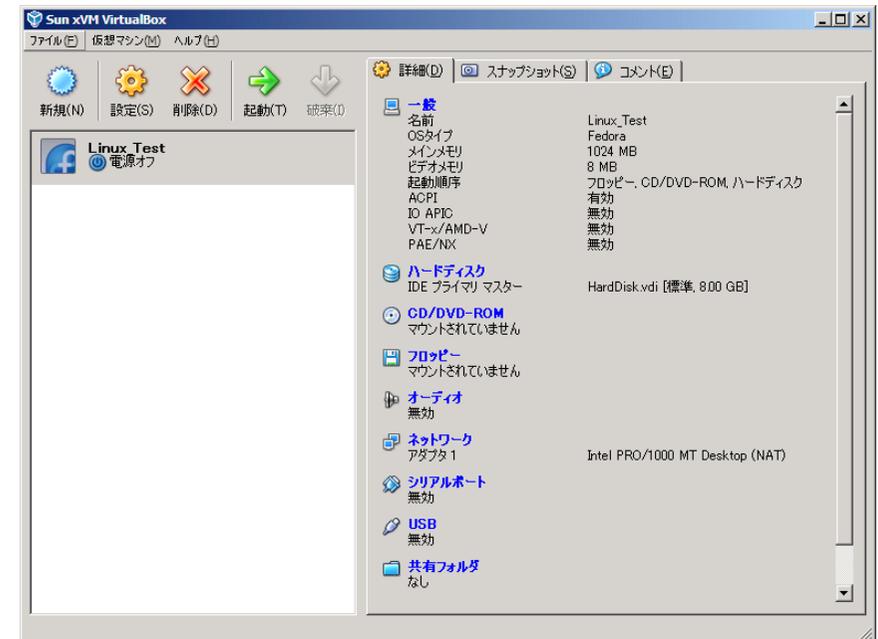


Oracleから提供されているフリーのバーチャルマシン環境

- VmwareやVirtual PCと同等の機能を持つ
 - 仮想マシン上にOSをインストール
 - ゲストOSは複数インストール可能
 - 各種デバイスエミュレーション機能
 - 仮想ネットワークも構築可能

VirtualBox.org

<http://www.virtualbox.org/>





WindowsでVirtualBox



The screenshot displays a Windows desktop environment with several open windows:

- Microsoft Internet Explorer:** Shows the HI CORP. website.
- Sun xVM VirtualBox:** A window titled "Linux_Test [実行中]" showing a terminal window with a file listing. The listing includes permissions, user/group, size, date, and filename for various system files and directories.
- VirtualBox Settings:** A settings window for the "Linux_Test" VM. The "ハードディスク" (Hard Disk) section is expanded, showing a primary IDE hard disk named "HardDisk.vdi" with a capacity of 8.00 GB.

The terminal window output is as follows:

```

root@vbox_fedora9:~
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
drwxr-xr-x 21 root root 4096 2008-11-01 01:07 .
drwxr-xr-x 21 root root 4096 2008-11-01 01:07 ..
-rw-r--r-- 1 root root 0 2008-11-01 01:07 .autofsck
-rw-r--r-- 1 root root 0 2008-10-24 06:10 .autorelabel
drwxr-xr-x 2 root root 4096 2008-10-29 15:53 bin
drwxr-xr-x 5 root root 1024 2008-10-24 01:34 boot
drwxr-xr-x 12 root root 4000 2008-10-31 16:41
drwxr-xr-x 102 root root 12288 2008-10-31 16:41
drwxr-xr-x 3 root root 4096 2008-10-24 02:00
drwxr-xr-x 15 root root 12288 2008-10-29 15:53
drwx----- 2 root root 16384 2008-10-24 01:20
drwxr-xr-x 2 root root 4096 2008-11-01 01:00
drwxr-xr-x 2 root root 4096 2008-04-08 06:41
drwxr-xr-x 2 root root 4096 2008-04-08 06:41
dr-xr-xr-x 132 root root 0 2008-11-01 01:00
drwxr-xr-x 26 root root 4096 2008-10-31 16:41
drwxr-xr-x 2 root root 12288 2008-10-29 15:53
drwxr-xr-x 2 root root 4096 2008-10-24 01:20
drwxr-xr-x 2 root root 4096 2008-04-08 06:41
drwxr-xr-x 11 root root 0 2008-11-01 01:00
drwxrwxrwt 33 root root 4096 2008-10-31 16:41
drwxr-xr-x 13 root root 4096 2008-10-24 01:20
drwxr-xr-x 20 root root 4096 2008-10-24 01:40
[root@vbox_fedora9 ~]#
  
```



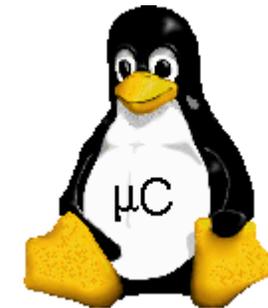
- 知識と実践をバランスよく
 - 知識として背景含めて知ることによる理解の促進
 - 実践することによる知識の安定化
 - 学んだことは即実践、実践しつつ学ぶ



- 様々な分野に 응용が



CE Linux
Forum



montavista™

axLinux





本日解説するポイント

101試験

- 103.1 : コマンド行で操作する
- 103.2 : フィルターを使ってテキストストリームを処理する
- 103.4 : ストリーム、パイプ、リダイレクト
- 104.3 : ファイルシステムのマウントアンマウントを制御する
- 104.6 : ハードリンクとシンボリックリンクを作成・変更する

102試験

- 109.2 : 基本的なネットワーク構成

新試験範囲

- 101.3 : systemdおよびUpstart



範囲の概要

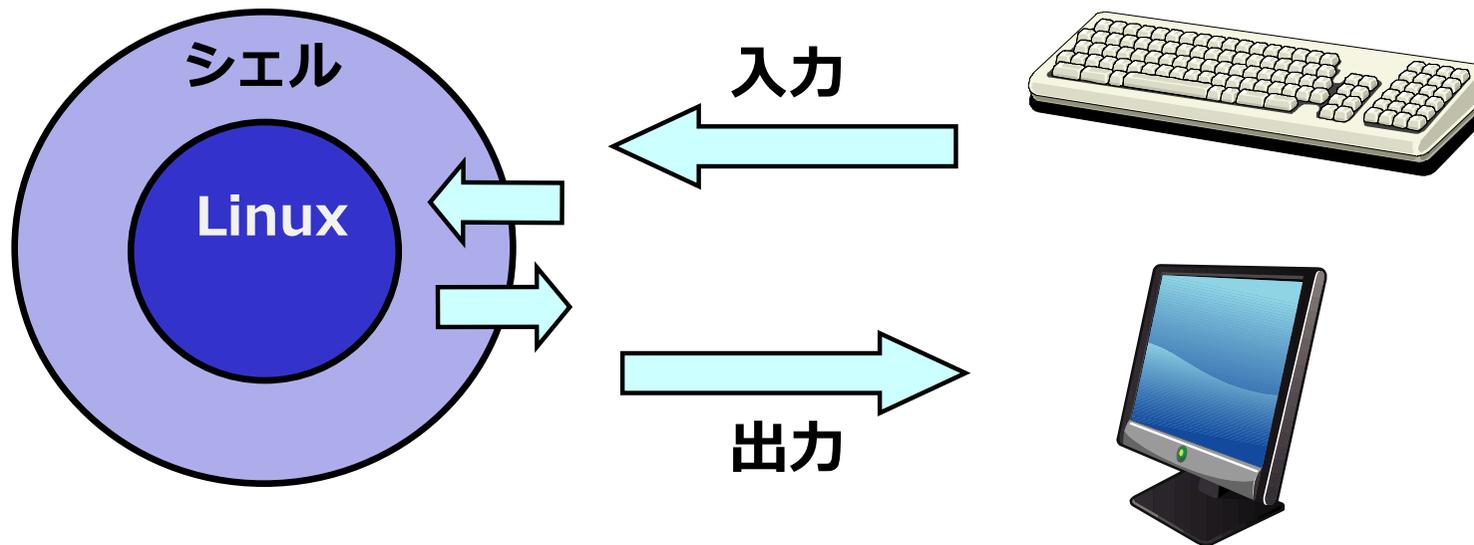
シェルと対話して、コマンド行を使用する。
bashシェルを使用する。

主要な知識範囲

- 1つのシェルコマンド、ないしは、1行のコマンド列を使い、コマンド行で基本的な作業を行う
- 環境変数を定義、参照、ならびにエクスポートするなど、シェルの環境を利用ならびに変更する
- コマンド履歴を利用ならびに編集する
- パスに存在する、存在しないにかかわらず、コマンドを起動する
- ディレクトリツリーに沿って、コマンドを再帰的に実行する



OSを操作するためのインターフェイスであり、OSの外殻となるもの



LinuxのCUIの実態であり、様々な種類が存在する
※ Linuxのデフォルトシェルはbash



コマンド実行の基本フォーマット

コマンド名 [オプション...] [引数 (パラメータ) ...]

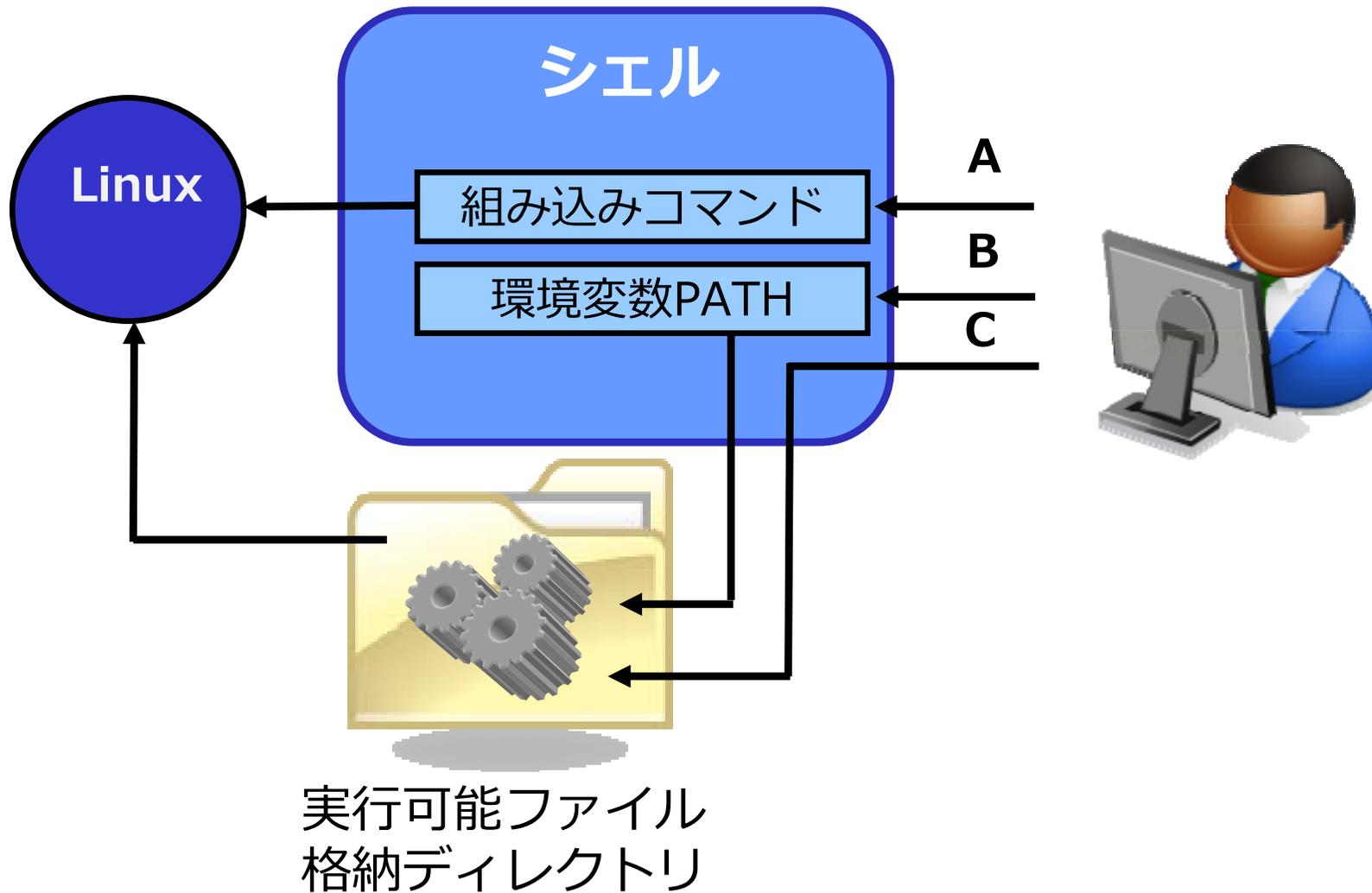
基本知識

- コマンド入力はプロンプト（「#」や「\$」）に続けて入力
 - プロンプトが出力されていないときはコマンド実行中
- オプション形式はコマンドによって異なる
 - Unix98(-)、BSD、GNU(--)などの形式がある
- 相対パス、絶対パスを理解する

- シェルによりコマンド実行補助機能が提供される
 - 履歴、補完、キーバインド等



コマンド実行には3つのフローがある





組込みコマンドとは？

シェルに組み込まれたコマンド

➤ cd、pwd、exportなど

- 組込みコマンドはhelpコマンドで確認できる
- コマンドが組込みコマンドかはtypeコマンドで確認できる
- man bashで確認することも可能



環境変数PATHとは？

指定されたコマンドをシェルが検索するための情報

- シェルはいくつかの環境変数を持ち、PATHはその中の1つ
- 確認はechoコマンドなどで行うことができる
- 設定されている順番に検索が行われる
- whichコマンドでコマンドパスを確認する際にも利用される
- 環境変数PATHに設定された場所にはないコマンドはパスなしで実行できない

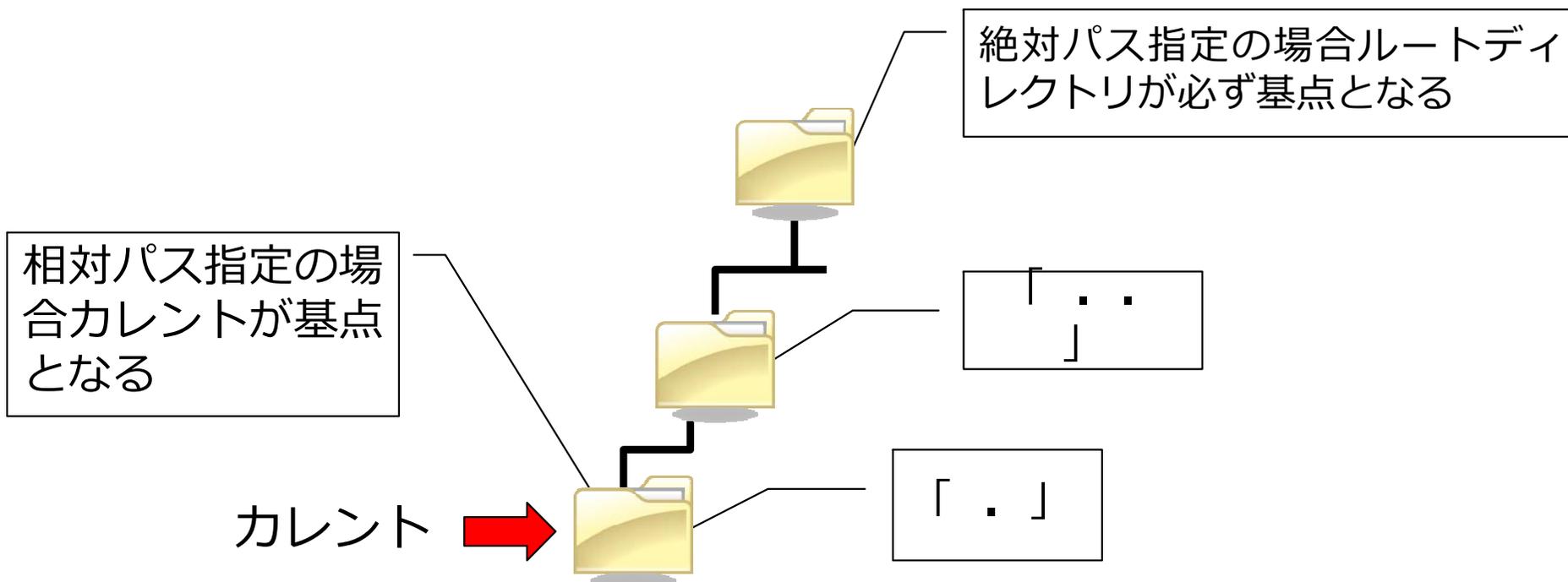


コマンドのダイレクト実行とは？

コマンドの場所をダイレクトに指定し、実行する

➤ 環境変数PATHに登録されてなくても実行可能

- 絶対パス指定、相対パス指定で実行
相対指定は「.」、「..」を利用





範囲の概要

ストリームをリダイレクトして、テキストデータを効果的に処理するために接続する。これには、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への入力にパイプする、出力を標準出力とファイルの両方に送るといったことを含む。

主要な知識範囲

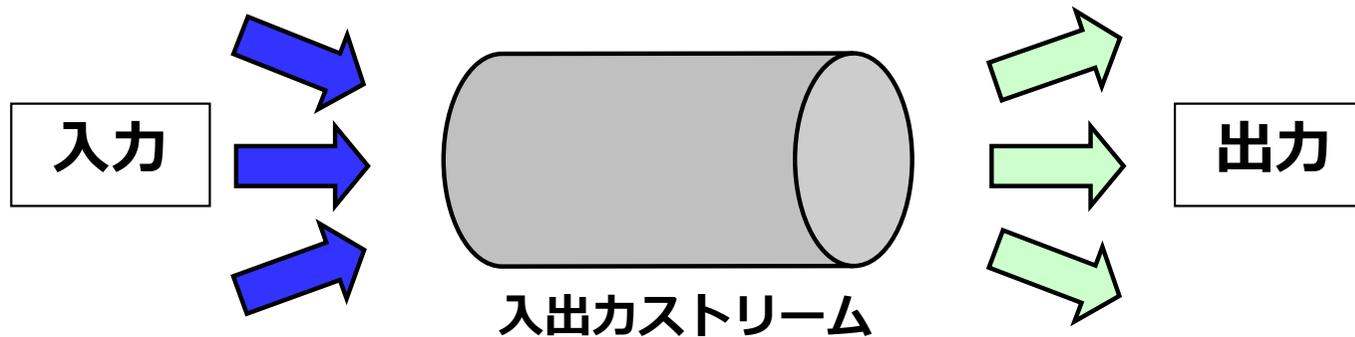
- ・ 標準入力、標準出力、標準エラー出力をリダイレクトする
- ・ あるコマンドの出力を、別のコマンドの入力にパイプする
- ・ あるコマンドの出力を、別のコマンドの引数として利用する
- ・ 出力を、標準出力と、ファイルの両方に送る



ストリームとは？

データの流を表現する概念

➤この概念を使うことにより入出力処理が統一される



- ストリームのデータはシーケンシャル（順次）に流れてくる
- ランダムアクセスできない
- ストリーム操作には標準入力、標準出力、標準エラー出力を利用する



標準入力 (stdin:0)

通常のテキストの入力を扱うストリーム.デフォルトはキーボードの入力.

標準出力 (stdout:1)

通常のプログラムの出力を扱うストリーム.デフォルトは
端末 (画面) への出力.

標準エラー出力 (stderr:2)

コマンドが正常に実行されなかった場合の出力を扱うストリーム.
デフォルトは端末 (画面) への出力.

これらのファイルを利用して、リダイレクト、パイプなどを利用する



リダイレクトとは？

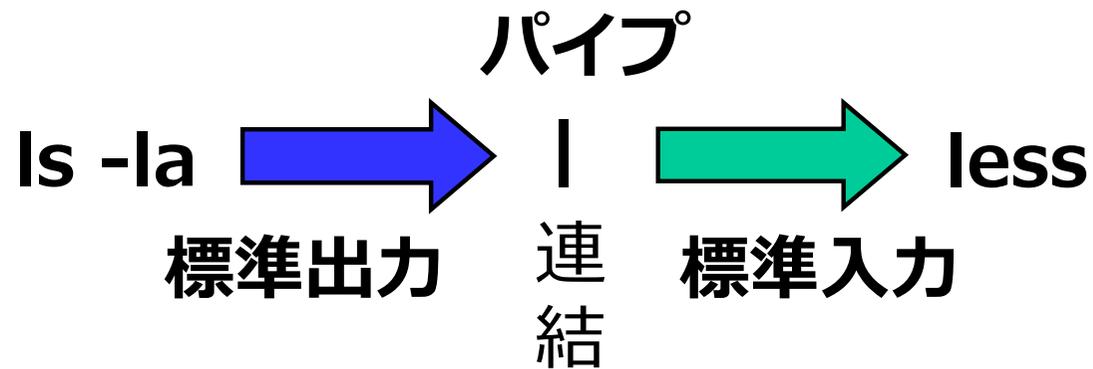
データストリームの入出力を制御し、プログラムの入出力を異なるものに変更する

リダイレクトの指定	意味
cmd > file cmd 1> file	cmdの標準出力をfileに行う
cmd >> file cmd 1>> file	cmdの標準出力をfileに追加する
cmd 2> file	cmdの標準エラー出力をfileに行う
cmd 2>> file	cmdの標準エラー出力をfileに追加する
cmd < file	標準入力をfileから受け取る
cmd > file 2>&1	cmdの標準出力、標準エラー出力をfileに行う
cmd >> file 2>&1	cmdの標準出力、標準エラー出力をfileに追加する



パイプとは？

先に指定したコマンドの標準出力ストリームを後に指定したコマンドの標準入力ストリームに連結する



ls -laの出力を
lessで表示



範囲の概要

テキストストリームにフィルタを適用する

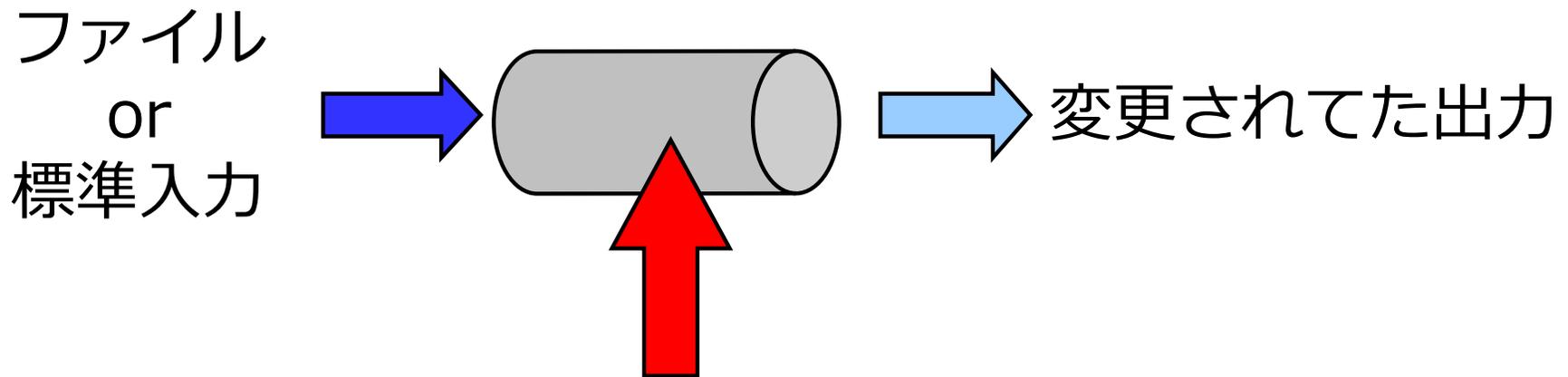
主要な知識範囲

GNU textutilsパッケージにある標準的なUNIXコマンドを使用して出力を加工するために、ファイルや出力ストリームをフィルタユーティリティに送信する



フィルタコマンドとは？

テキストストリームを処理するコマンド



フィルタコマンドによるテキスト編集

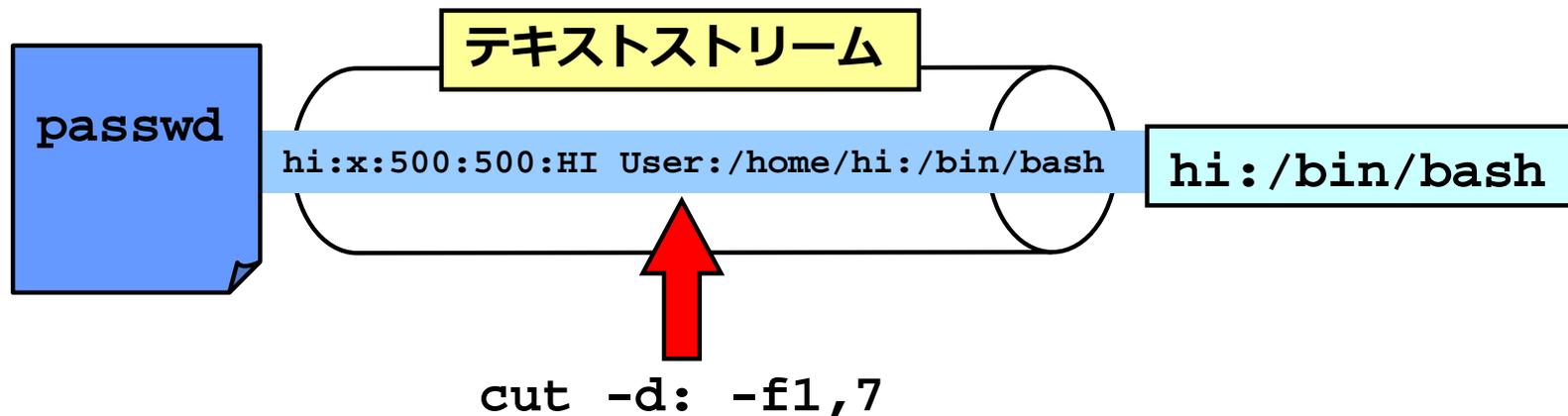


例：cutコマンド

ファイルからの入力ストリームに対して、指定行された行、行の一部を取り出して出力する

実行例

```
$ cut -d: -f1,7 /etc/passwd  
root:/bin/bash  
bin:/sbin/nologin  
daemon:/sbin/nologin  
...
```





類似コマンドに注意

出題範囲には多くのフィルタコマンドが含まれている。
類似するコマンドの注意。

cat、cut、expand、unexpand、fmt、head、tail、od、join、nl、paste、pr、
sed、sort、split、tr、uniq、wc

例えば . . .

expand

タブをスペースに変換

unexpand

スペースをタブ変換



範囲の概要

ファイルシステムのマウントを構成する。

主要な知識範囲

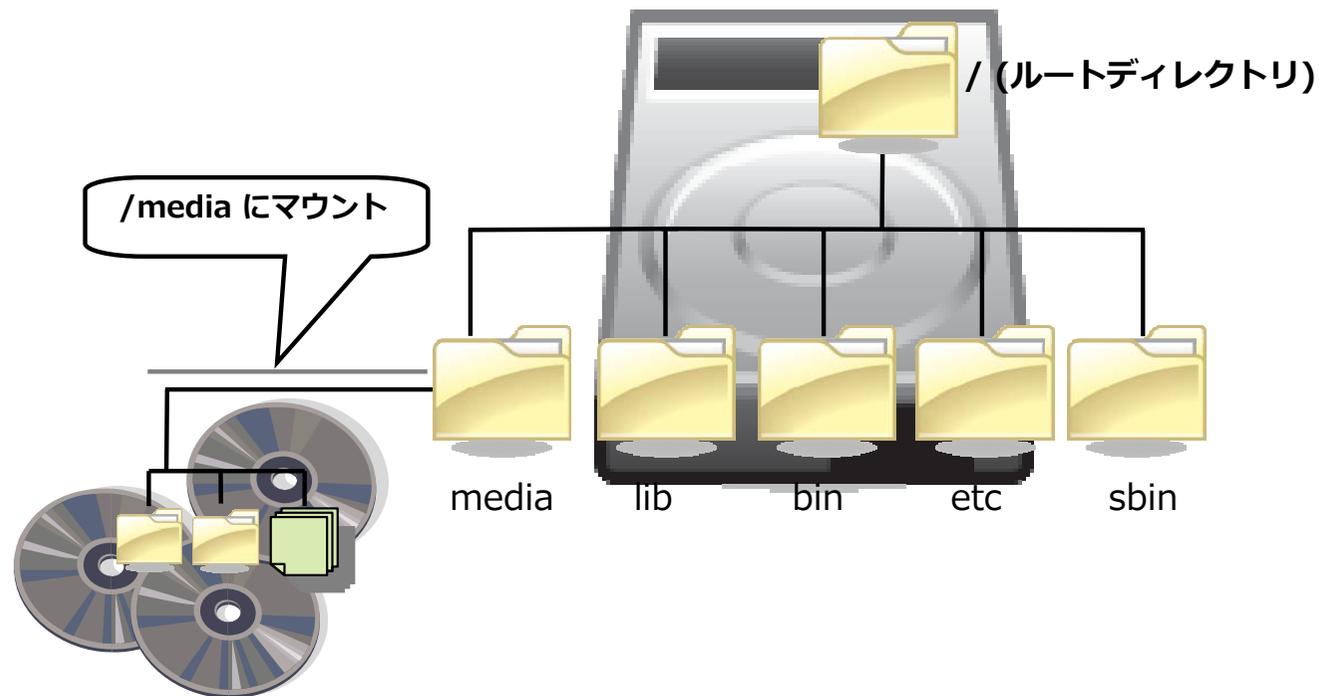
- ファイルシステムを手作業でマウントないしアンマウントする
- ブート時にファイルシステムをマウントするように構成する
- リムーバブルなファイルシステムをユーザがマウントできるように設定する



マウントとは？

OSからファイルシステムを利用可能にする手続き

- マウントなしにファイル参照はできない





- マウント系のコマンド

mount、umountを実行

ex. mount -t vfat /dev/cdrom /media
umount /media

※ マウントしたデバイスを取り出すときにはumountを実行し、マウント解除する

- 定義済みマウント

ファイルシステムテーブル/etc/fstabに記述
起動時に自動的にマウントを行うことが可能



範囲の概要

ファイルに対するハードリンクとシンボリックリンクを作成して管理する。

主要な知識範囲

- ・リンクを作成する
- ・ハードリンクとシンボリックリンクを識別する
- ・ファイルのコピーとリンクの違い
- ・システム管理業務をサポートするためにリンクを使用する



- リンクとは
実体への参照を意味する。
Linuxはハードリンクとシンボリックリンクがある。
- ハードリンク
実体と同一のi-nodeを持つリンク。
同一のファイルシステム上にしか作成できない。
- シンボリックリンク
実体と異なるi-nodeを持つリンク。
異なるファイルシステム間のリンクが可能。



- lnコマンド
ln [オプション] リンク元 リンク先
-s オプション：シンボリックリンクの作成
ex. ln -s /var/log/maillog ./mail.log
- i-node数の確認
dfコマンドの-iオプションでファイルシステムの
i-node数と利用率を確認。



範囲の概要

クライアントホスト上の設定を参照、変更、確認する。

主要な知識範囲

- 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の設定



- N/Wクライアントの設定
 - 設定はディストリビューションごとに異なる
 - 非依存な設定方法が求められる
 - ex. ホストネーム、名前解決、ルーティング
- 起動スクリプトで行われていることを把握する
 - 基本はコマンドで設定するが設定の永続性がない。
 - 設定ファイルを記述して、起動時に実行することで永続性を構築する。



- 設定ファイル
 - ホストネーム : /etc/hostname
 - ローカル名前解決 : /etc/hosts
 - DNS設定 : /etc/resolv.conf
 - Nameサービス変更 : /etc/nsswitch.conf
- コマンド
 - ネットワークI/F設定 : ifconfig
 - ルーティングテーブル設定 : route
 - ネットワークI/F起動 : ifup、ifdown
 - ネットワーク確認 : ping

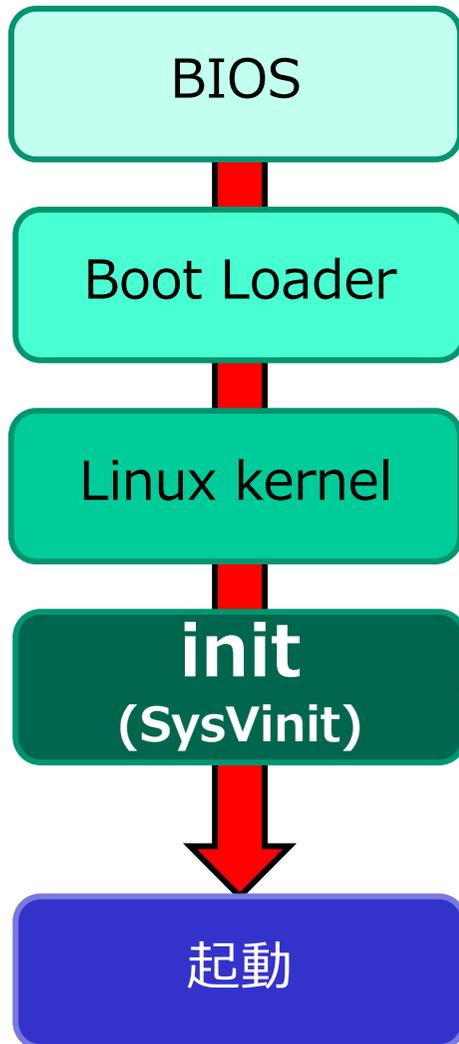


範囲の概要

システムのランレベルを制御する。これには、シングルユーザモードへの変更と、システムのシャットダウンまたはリブートも含まれる。また、ランレベルを変更する前にユーザに警告し、プロセスを適切に終了させる。これには、デフォルトのランレベルの設定も含まれる。

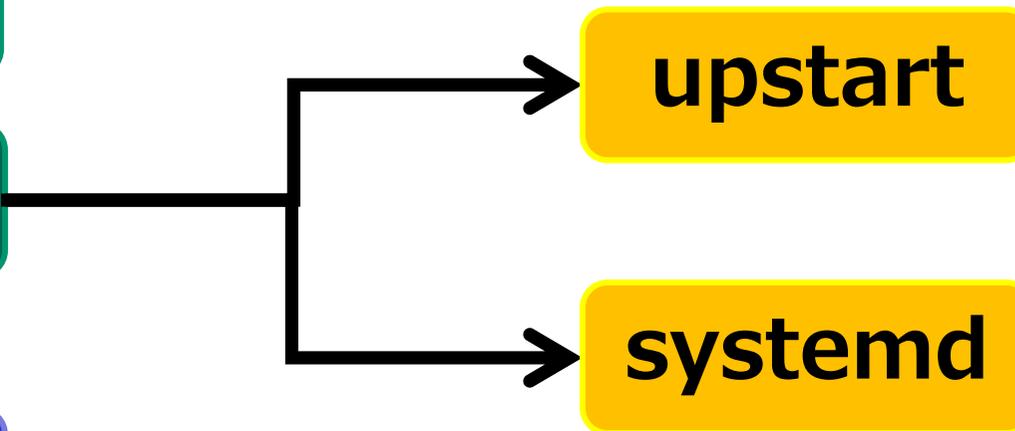
主要な知識範囲

- 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
- デフォルトのランレベルを設定する
- ランレベル（シングルユーザモードを含む）を変更する
- コマンドラインからシャットダウンおよびリブートを行う
- ランレベルの切り替えやその他の重要なシステムイベントの前にユーザに警告する
- プロセスを適切に終了させる
- **systemdおよびUpstartの基本的機能（新規）**



Linuxの起動シーケンスにおいて様々なサービスやアプリケーションを起動してシステムとしてLinuxを使用可能にするプロセスがinit。プロセスID 1

旧initプロセス(SysVinit)の置き換える次世代のinitプロセスとしてupstartとsystemdが登場。





• SysVinit

各ランレベルごとに決められたサービスを設定された順番に従って起動する。起動はシーケンシャルに行われ、同期的に起動される。そのため下位番号に設定されたプロセスについては上位番号のプロセスが起動されるまで待ちの状態となる。

• upstart

起動対象をイベントと関連付けし、イベントが発生するとイベントと関連付けされた対象が非同期に起動する。ただし依存関係の設定により不必要なプロセスまで起動されることがある。

• systemd

upstartの問題点を解決し、より高速で高効率の初期化プロセス。ソケットやD-Busを利用したアプリケーション間のメッセージにより対象を制御し、より細かなプロセスの管理を行うことが可能。



- マルチディストリビューションで学ぶ
Redhat系、Debian系を中心に複数のディストリビューションで学ぶ
- 管理系の範囲は設定で学ぶ
ある場所を管理する設定で設定項目を学ぶ
- ネットワークは実際にネットワークシステムを構築する
単一のサーバだけではなく、サーバを組み合わせてシステム化する
- オリジナルのものを作ってみる
オリジナルカーネルや、オリジナルブートディスクなど

何かしらの目的なり、設定を作るほうが身につきやすい



ありがとうございました。

株式会社エイチアイ

<http://www.hicorp.co.jp>