

# 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～

株式会社エイチアイ

末永貴一

2008.11.12

---

# 自己紹介

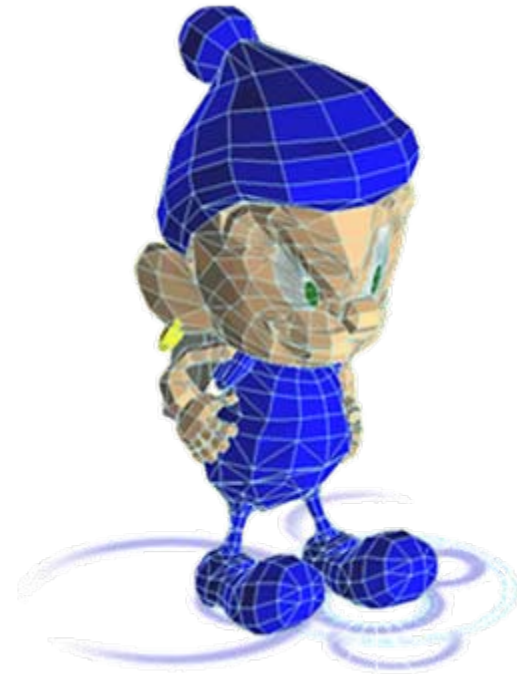
## 株式会社エイチアイ 研究開発部 部長

－ 組み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」  
など



# 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

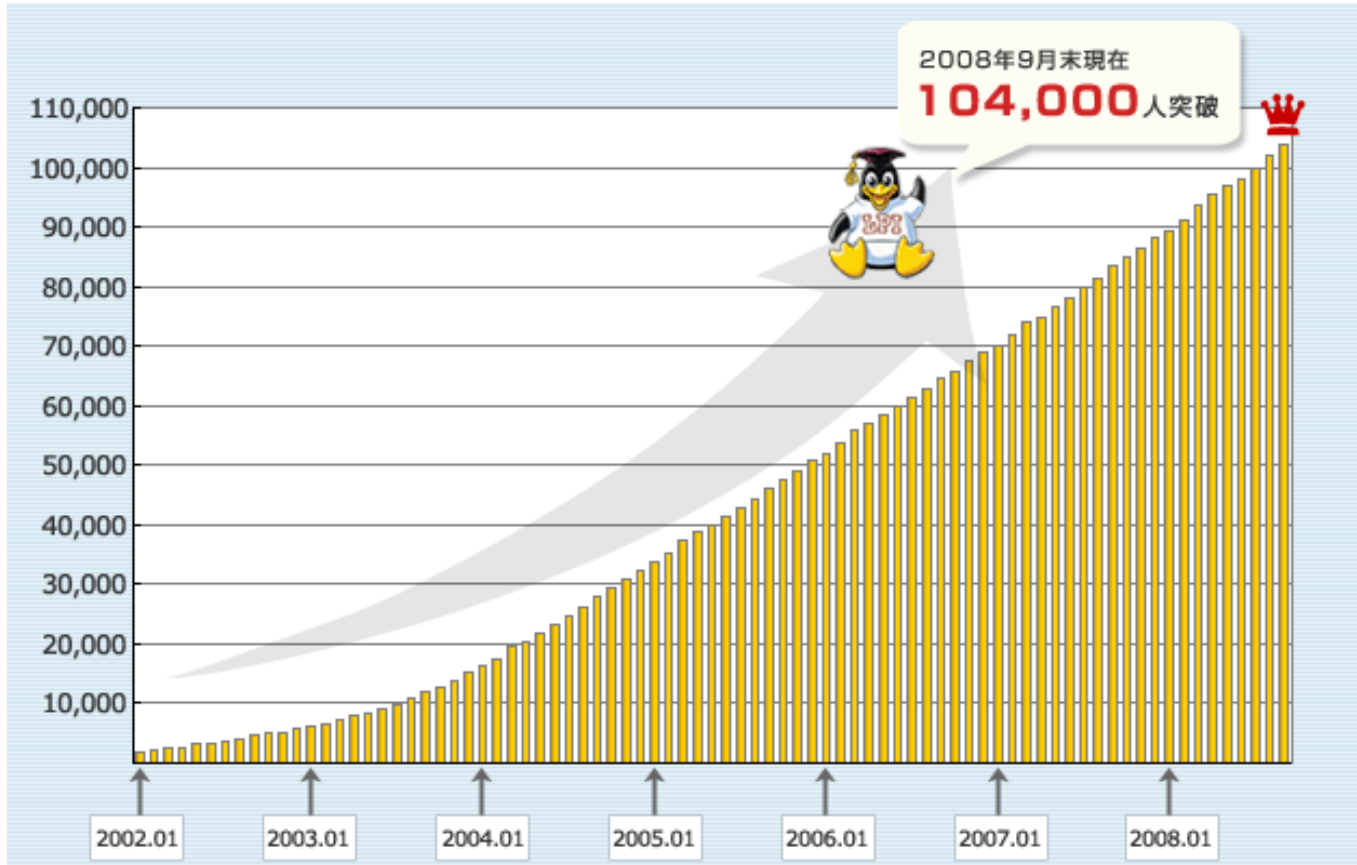
### ● 特徴

- 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

## 国内受験者数: 10万4000人を突破



## 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。

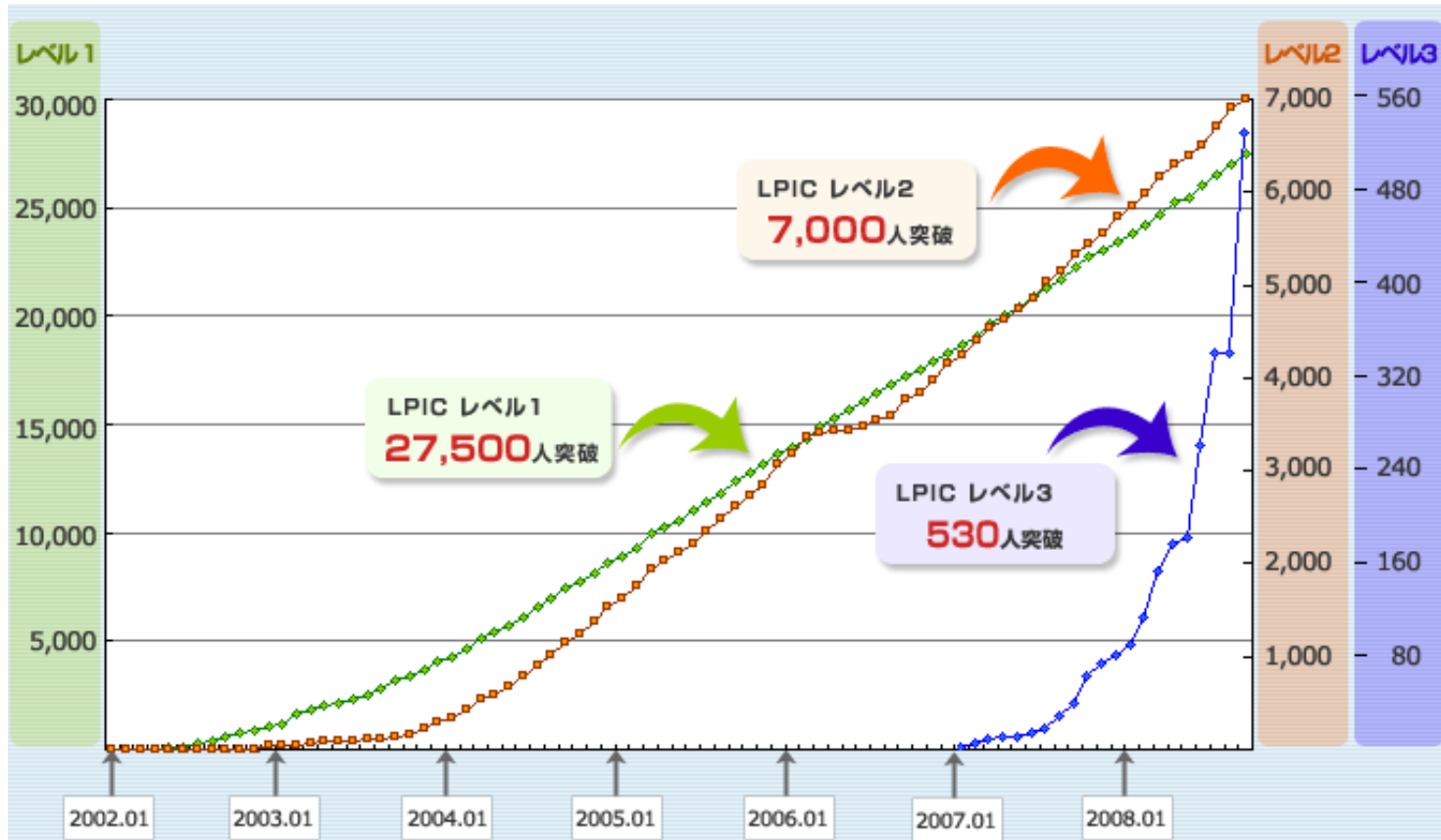
レベル3 Coreは、301試験に合格すること。レベル3 Specialtyは、301試験に合格し、かつ302試験以降のいずれか1つに合格すること。

## 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

レベル1:27,500人、レベル2:7,000人、レベル3:530人  
各レベルでそれぞれ突破(2008年9月末現在)



## Level1取得には101,102試験に合格する

### 101試験

主題101:ハードウェアとアーキテクチャ

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

主題110:X Window system

### 102試験

主題105:カーネル

主題106:ブート、初期化、シャットダウン、  
ランレベル

主題107:印刷

主題108:ドキュメンテーション

主題109:シェル、スクリプト、プログラミング、  
コンパイル

主題111:管理業務

主題112:ネットワークの基礎

主題113:ネットワークサービス

主題114:セキュリティ

# 2009年4月よりRelease3スタート予定！

## 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
- 主題110:X Window system

## 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

内容も見直され、新しいテーマが追加される。

## Release3から追加された主なテーマ

- **全体的に最近の機能にアップデートされている。**
  - パッケージ管理のyum、sysfsなどの概念、ディスプレイマネージャなど
  - 非IDEデバイスやモデム系通信デバイスなどのレガシーシステム系は削除
- **SQLデータ管理**
  - 基本的なDBとSQL言語の理解
- **アクセシビリティ**
  - 支援技術に関する各種機能(障害者支援、入出力支援など)
- **ローカライゼーションと国際化**
  - 多言語対応、タイムゾーンに関する設定。

- 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

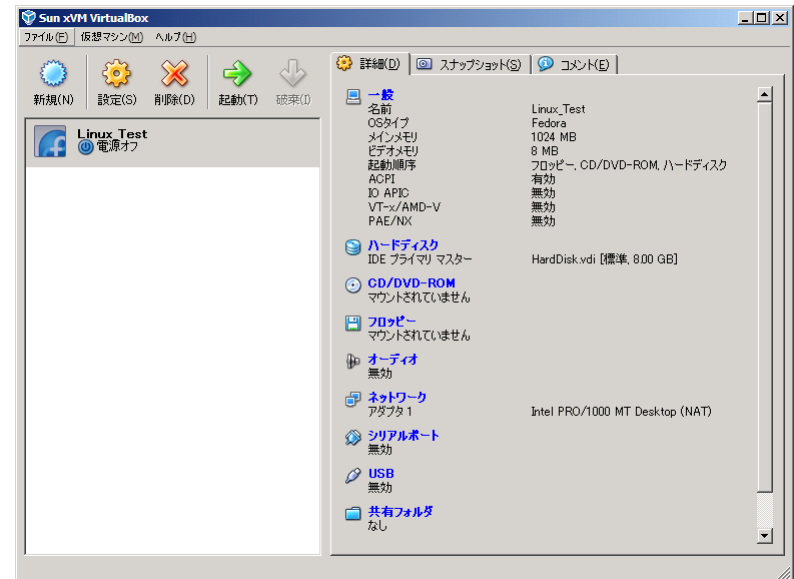
- **出題範囲の詳細をよく確認する**
  - 説明部分の知識を体系的に知る
  - 知識範囲は必ず実行する
  - キーワードは必ずチェック
  
- **勉強用の環境を構築する**
  - 繰り返しインストールできる環境
  - 壊しても問題ない環境
  - すぐに試せる環境

- 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

## Sun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/>

The screenshot displays a Windows XP desktop environment. In the foreground, a Sun xVM VirtualBox window is open, showing a Linux Test virtual machine (VM) running. The terminal window within the VM displays the following file listing:

```

root@vbox_fedora9:~# ls -l
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 02:0
drwxr-xr-x 15 root root 12288 2008-10-29 15:5
drwx----- 2 root root 16384 2008-10-24 01:2
drwxr-xr-x 2 root root 4096 2008-11-01 01:0
drwxr-xr-x 2 root root 4096 2008-04-08 06:4
drwxr-xr-x 2 root root 4096 2008-04-08 06:4
dr-xr-xr-x 132 root root 0 2008-11-01 01:0
drwxr-x--- 26 root root 4096 2008-10-31 16:4
drwxr-xr-x 2 root root 12288 2008-10-29 15:5
drwxr-xr-x 2 root root 4096 2008-10-24 01:2
drwxr-xr-x 2 root root 4096 2008-04-08 06:4
drwxr-xr-x 11 root root 0 2008-11-01 01:0
drwxrwxrwt 33 root root 4096 2008-10-31 16:4
drwxr-xr-x 13 root root 4096 2008-10-24 01:2
drwxr-xr-x 20 root root 4096 2008-10-24 01:4

```

The VirtualBox settings window is open, showing the following details for the Linux\_Test VM:

- 名前:** Linux\_Test
- OSタイプ:** Fedora
- メインメモリ:** 1024 MB
- ビデオメモリ:** 8 MB
- 起動順序:** フロッピー、CD/DVD-ROM、ハードディスク
- ACPI:** 有効
- IO APIC:** 無効
- VT-x/AMD-V:** 無効
- PAE/NX:** 無効
- ハードディスク:** IDE プライマリ マスター HardDisk.vdi [標準, 8.00 GB]
- CD/DVD-ROM:** マウントされていません
- フロッピー:** マウントされていません
- オーディオ:**

## ・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

## 頭と体で学ぶ！

## ・ 様々な分野に應用が



CE Linux  
Forum



# 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

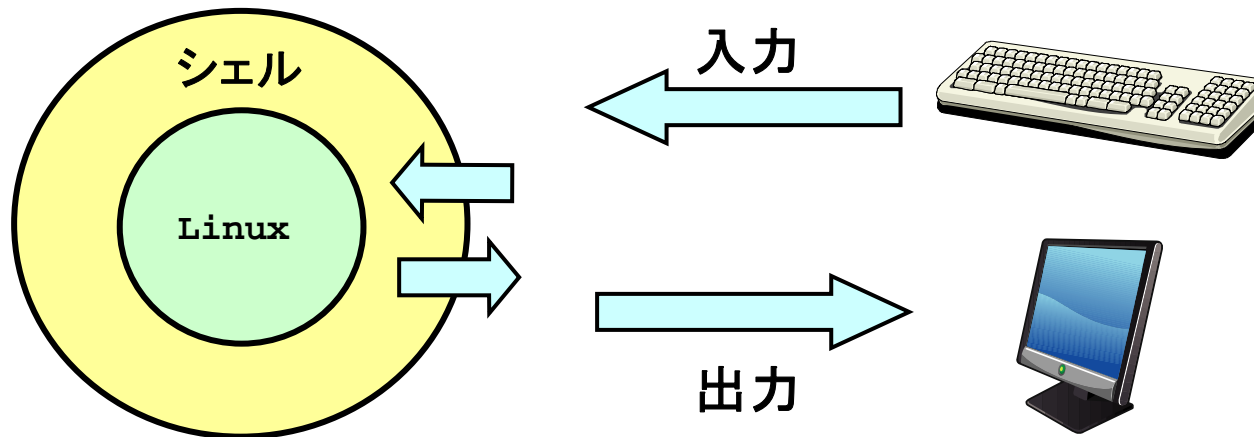
## 範囲の概要

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

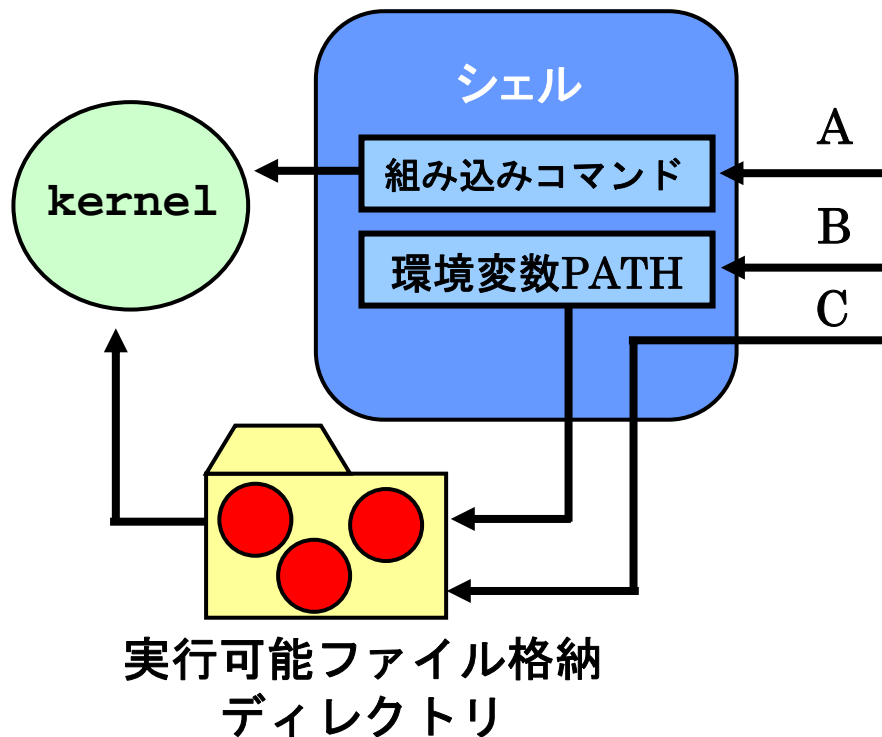
## コマンド実行の基本フォーマット

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



## 組み込みコマンドとは？

### シェルに組み込まれたコマンド

➤ `cd`、`pwd`、`export`など

- 組み込みコマンドは`help`コマンドで確認できる
- あるコマンドが組み込みコマンドかは`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

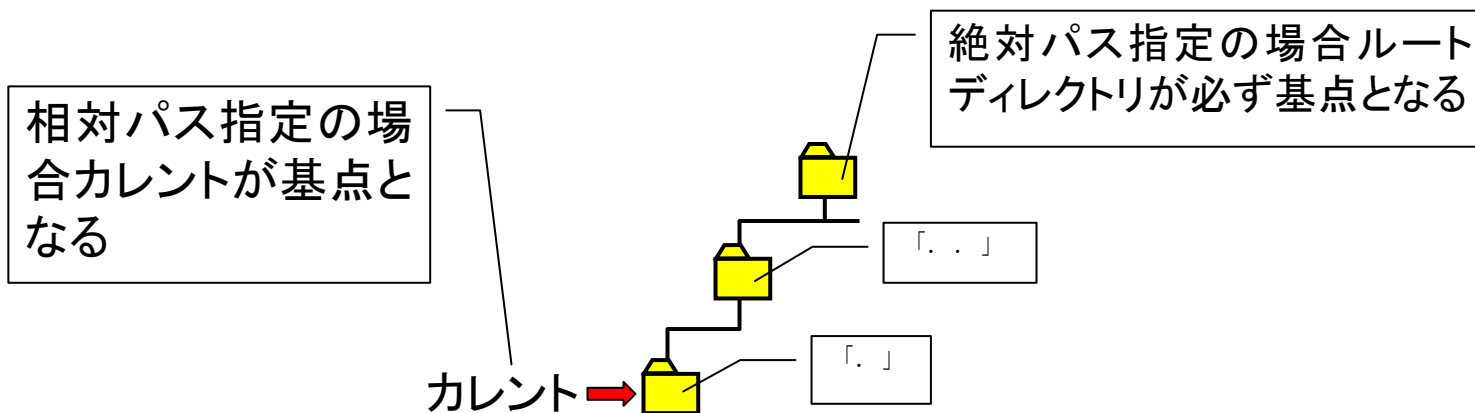
- ・確認はechoコマンドなどで行うことができる
- ・設定されている順番に検索が行われる
- ・whichコマンドでコマンドパスを確認する際にも利用される
- ・環境変数PATHに設定された場所にはないコマンドはパスなしで実行できない

## 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。

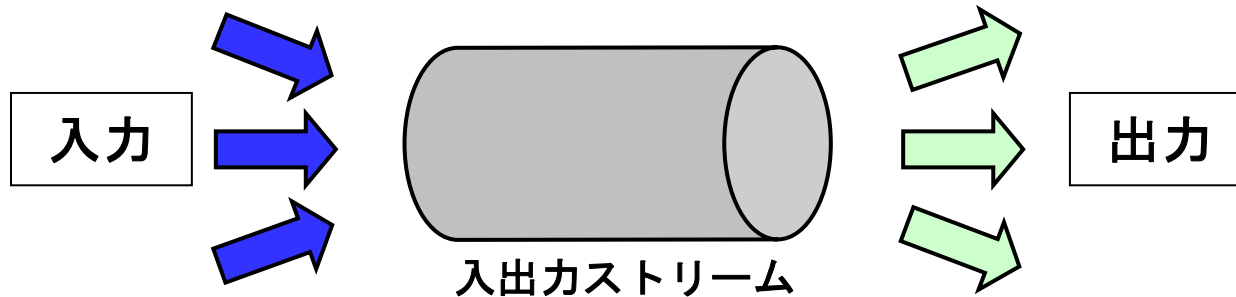
## 主要な知識範囲

- ・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

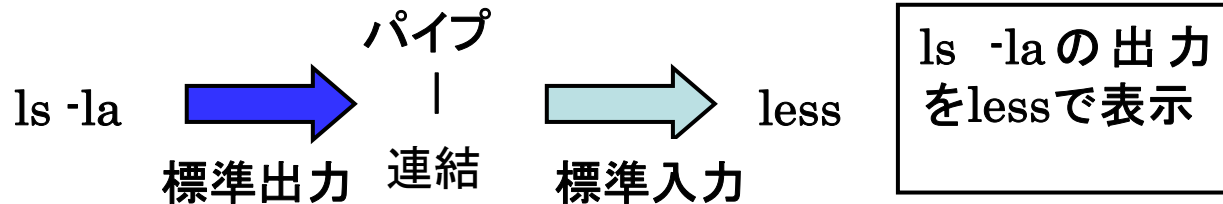
## リダイレクトとは？

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                         |
|---|----------------------------|
| <code>cmd &gt; file</code><br><code>cmd 1&gt; file</code>         | cmdの標準出力をfileに行う           |
| <code>cmd &gt;&gt; file</code><br><code>cmd 1&gt;&gt; file</code> | cmdの標準出力をfileに追加する         |
| <code>cmd 2&gt; file</code>                                       | cmdの標準エラー出力をfileに行う        |
| <code>cmd 2&gt;&gt; file</code>                                   | cmdの標準エラー出力をfileに追加する      |
| <code>cmd &lt; file</code>  | 標準入力をfileから受け取る            |
| <code>cmd &gt; file 2&gt;&amp;1</code>                            | cmdの標準出力、標準エラー出力をfileに行う   |
| <code>cmd &gt;&gt; file 2&gt;&amp;1</code>                        | 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



## xargsコマンド

コマンドのパラメータを標準入力から得ることが可能

```
$ cat create.txt | xargs touch
```

パイプと組み合わせることで便利に

## 範囲の概要

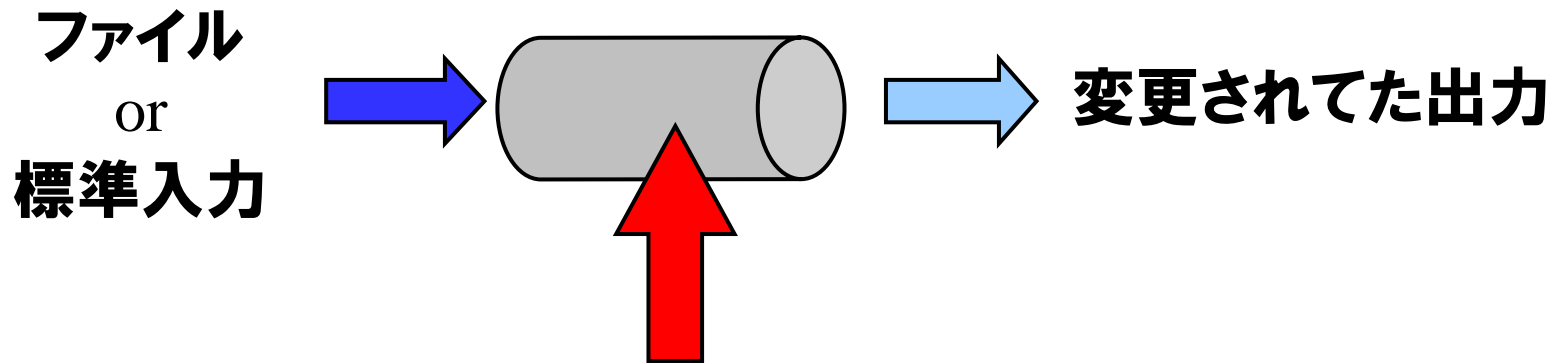
### テキストストリームにフィルタを適用する

## 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



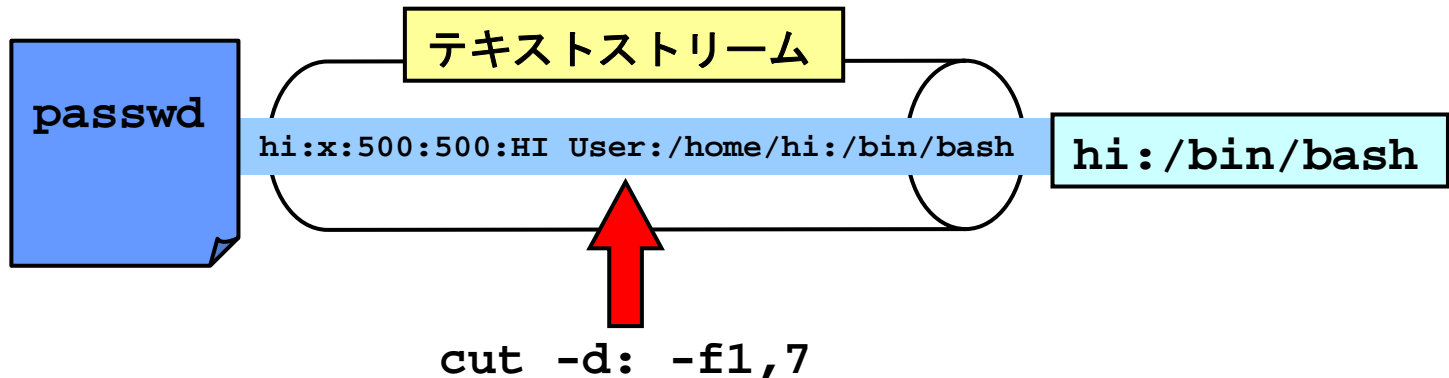
フィルタコマンドによるテキスト編集

## 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  
...
```



## 範囲の概要

ファイルシステムのマウントを構成する。

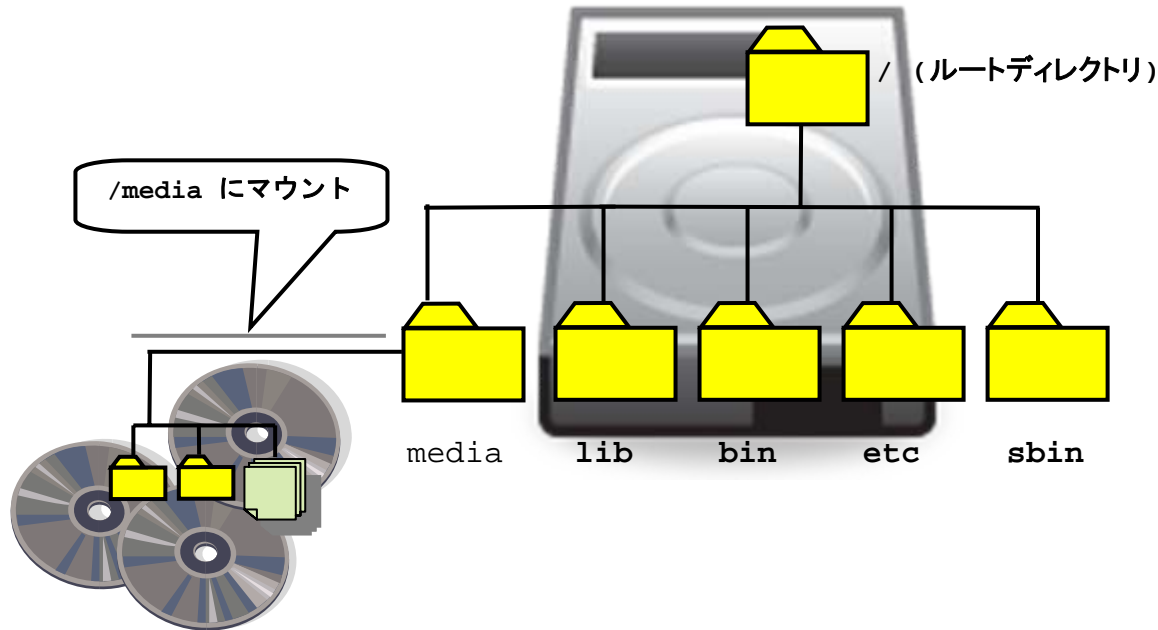
## 主要な知識範囲

- ・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

- **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**

# Q & A

**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>**