

Linuxの正しい歩き方

～道に迷わないために～

2015/4/26

株式会社GOOYA ICT事業部

藤木 信明



■株式会社GOOYA

「人と世界を笑顔でつなぐ」

新たな雇用の創出、その雇用を継続するための人材育成・教育支援、独立採算で成り立つ組織の構築などに積極的に取り組みます。

そして、これらの企業活動を通じ、GOOYAではたらく仲間の「目標や夢の実現」を目指し、GOOYAと関わる全ての人々と幸せと感動、笑顔を共有し、世界中で地域社会への貢献を推し進めていきます。

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

■講師紹介

「藤木 信明」

オープンソースソフトウェアを中心とした、ITテクニカル教育を15年。

もともと、ネットワーク&サーバエンジニアの経験を活かしたカリキュラムを構成し、独自の教育方針で未経験エンジニアの育成に携わっています。



ぼくがシステムを教える際に、かなり重要とっており、セミナーの際、いつもみなさんをお願いすることがあります。

それは「イメージしながら触ってもらいたい」ということです。

記憶術の一つにもあるんですが、何かを覚える際にはとても役に立つと思われれます。

単純なイメージよりは非現実的なイメージを持った方が、覚えるのはもちろん、後々記憶を引き出しやすくなります。

今日のお話も、随時イメージしてください、というところがありますのでみなさん、だまされたと思ってやってみてください。



LPIC概要とLevel 1 Ver.4.0に関して



Linux技術者認定制度

世界共通基準で認定、150カ国以上で受験

特定の製品メーカーや企業に依存しない

世界で40万人以上が受験、14万人以上の認定者

LPICのレベル

LPIC-1

パーソナル

LPIC-2

グローバル

LPIC-3

専門家レベル



改定の時期

2015年6月1日（あくまでも予定）

これまでのVer.3.5からVer.4.0に改定される。

メジャーディストリビューションへの対応

サービス管理のデフォルトであった、「SysV」から「systemd」に置き換わるなど、Linuxにもかなりの改変が行われている。新しいツール等を取り入れることで、認定者は最新のテクノロジスキルを証明できる。

受験方式

Ver.4.0リリース後6か月間は、旧バージョン（Ver.3.5）も受験可能。

申し込みの際に、どちらかを選択して受験することになる。

Level 1 認定には、101、102双方の合格が必要だが、それぞれ、バージョンが異なってもかまわない。



Linuxで飛び交う用語の意味とその覚え方



LinuxシステムをLinux未経験者に説明をするとき、

Linuxシステムは、シェアハウスだ

と言い切ります。

カーネルとは、その家のインフラ全般を決めるもの。

例えばオール電化であったり

キッチンに備わっている器具が充実していたり

例えば、200V電源が備わっていない家では対応の電化製品を使うことはできません。

どうしても使う必要がある場合は、カーネルをリフォームしなければいけません。

これがカーネルのアップデートです。



シェアハウスに関して、絶対的な権限を持つユーザです。
いわば、シェアハウスの管理人

rootがなにか命令を出した場合、逆らうことは基本的にできません。

そもそもユーザとは、そのシェアハウスに出入りする住人ということになります。

大きく分けて、

管理人（スーパーユーザ）

住人（一般ユーザ）

職人（特殊ユーザ）

という分別になるかと。

ちなみに、職人は出入りしません。
住み込みです。



シェアハウスにおいて、何かをしたい場合、ユーザはコマンドという命令文を用いて命令を出します。

この命令を受け止めるのがシェル。

住人がシェアハウスに入った時点で、隣で付き添うマネージャーのようなもの。

シェルの仕事としては、
ユーザの命令を聞き、それを実行する動きをする。
ユーザの命令をすべてメモに記録する。

住人は、自分と気の合うシェルを選ぶことができます。

bashというシェルは、こんな姿をしています
[root@hostname ~]#



さて、我々ユーザはシェルに対してコマンドを命令することで、やりたいことを実現することができます。

命令を受けたシェルは、それをどう解釈するのでしょうか。

例えば、

ls

と命令したとします。

シェルはまず、命令を実行させるために ls を探しに行きます。
この時どこを探すかは、前もってきめられています。

ls が見つければ、その旨、命令を伝えます。

もし見つからなければ、Command not found. と愚痴を言ってきます。



先ほどの `ls` を例にコマンドを分解してみます。

コマンドは仕事の役割が決められています。

`ls`

現在のカレントディレクトリの内容をリストアップして表示する

①



命令を受けた `ls` は、役割を実行し、また待機に入ります。

ちなみに、カレントディレクトリとは、
シェアハウスの中で、ユーザが現在いる部屋の事です。



それでは、決められている役割を変更したり、追加する場合どうするかというと、オプションや引数を指定することで実現させます。

```
ls -la /tmp/
```

先ほどの `ls` コマンドに、なんか文が追加されました。

前ページの、↑の部分にオプションが加わります。

①の部分が、引数で置き換えられます。

現在のカレントディレクトリの内容をリストアップして表示する



`/tmp/`の内容を[詳細に][全て]リストアップして表示する



- l オプションは、[詳細に]という言葉を追加し、
- a オプションは、[全て]という言葉を追加します。

指定の仕方は、-la でも、-al でも機能します。

詳細に全て、でも
全て詳細に、でも意味が通じます。

/tmp/ という引数は、赤字の部分を置き換えます。
オプションとは違い、引数を指定する順番は①②と決められています。



先ほどの**シェアハウス**をイメージしてみまじょうか。

管理人（スーパーユーザ）

住人（一般ユーザ）

職人（特殊ユーザ）

rootユーザの箇所でも触れましたが、ユーザとは、簡単に言えば、シェアハウスの中をうろうろする人ということになります。

Linuxシェアハウスでは、このユーザをわかりやすくするためにグループにまとめて管理する。
という手段を取っています。



グループに関しては、以下のルールがあります。

1. グループには、メインとサブの2種類がある
2. ユーザは必ずどこかのグループに属さなければならない

ちょっと専門用語で言い換えると

メイン → プライマリ

サブ → セカンダリ

必ず一つのプライマリグループに属さなければならない

ということになります



グループは、新規で自由に作成することができ、
管理しやすい環境を構築することが可能です。

#groupadd 新規グループ名
で作成可能です

/etc/group
に情報が書き込まれていきます。

必ず一つのプライマリグループに属さなければいけない
セカンダリグループには、複数所属させることができる

というルールがありますが、
プライマリもセカンダリもシステム的には特に区別はありません。



さて、ここまで理解いただけたら、次はユーザの登録について話をします。

ユーザはシェアハウスの利用者ですから、
名簿に登録しておく必要があります。

とりあえず、名前の情報があれば登録は可能です。

/etc/passwd

という名簿に情報が書き込まれていきます。

```
root:x:0:0:root:/root:/bin/bash
```

```
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
.  
.

```



先ほどの内容について、rootを例に簡単に説明します。

```
root:x:0:0:root:/root:/bin/bash
```

情報は全部で7つあります。

1. ユーザ名
2. パスワード
3. ユーザID
4. グループID
5. コメント
6. ホームディレクトリ
7. ログインシェル



最小でも、ユーザ名があれば登録はできますが、登録時に7つの情報を指定することももちろん可能です。

ただし、パスワードに関しては、コマンドログが残るため、別途設定することが望ましいと思われます。

```
#useradd [オプション] ユーザ名
```

- u ユーザID
- g プライマリグループ
- G セカンダリグループ
- c コメント
- d ホームディレクトリ (システムによっては-mを併用)
- s ログインシェル



useraddコマンドで、指定しなかった情報にはデフォルト値が割り当てられます。

ユーザID：一般ユーザは500番からの連番

グループID：プライマリには、ユーザ名と同じ名前のグループが自動で作成されそこに所属する。セカンダリには所属しない。

コメント：空欄

ホームディレクトリ：/home 以下に、ユーザ名と同じ名前のディレクトリが作成される

ログインシェル：/bin/bash

管理者が運用しやすいように適度なグループ管理をすべきなのですが、先述したとおり、プライマリもセカンダリも違いはないので、一般的には（私だけかも・・・）ユーザ名と同じ名前のグループをプライマリとし、必要に応じてセカンダリを増やす、という管理が楽です。



ディストリビューション差分



例えば、Ver.6では、httpdサービスを起動する際に、
service コマンドを使用したと思います。

```
# service httpd start
```

もしくは

```
# /etc/init.d/httpd start
```

Ver.7では、systemctl コマンドに置き換わっています。

```
# systemctl start httpd.service
```

systemdに対応しているソフトウェアは、`/usr/lib/systemd/system`
に.serviceファイルが用意されます



詳しくは述べませんが、従来のserviceコマンドでも起動できるような配慮がなされています。

/etc/init.d 以下に、スクリプトがあればそれを実行し、なければsystemctlコマンドにサービス名を引き渡して実行、という流れになります。

またその際には、リダイレクトされた旨のメッセージが出力されます。



Ver.6では、chkconfig を使用していました。

サービスの登録

```
# chkconfig -add [サービス名]
```

Ver.7では、こちらにもsystemctlを使用します。

サービスの登録

```
# systemctl enable [サービス名]
```

確認

```
# systemctl list-unit-files --type=service
```

この場合も、従来のコマンドで実行した場合、メッセージが出力されます。



Ver.7では、iptables に変わり、firewalld という概念が採用されています。完全に置き換えと言うことではなく、iptables のフロントエンドのようなものです。

/etc/sysconfig/iptables にルールを書いて、起動させれば適用されますが、**システム自体を再起動した場合は、その設定を読み込みません。**

このことから、Ver.7からは、iptables は停止し、firewalld を使用する、といった使い方がいいようです。



firewalld のコマンドで、
firewall-cmd --list-all と実行すると

```
public (default, active)
  interfaces: eno16777736
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

と表示され、現在は、public というゾーンがデフォルトであると確認できます。



firewalld の設定ファイルは、
/etc/firewalld/zones 配下にゾーンごとのxmlファイルを配置されています。

先ほど見て頂いたデフォルトが、public ですので、
/etc/firewalld/zones/public.xml
を編集し、httpd を許可してみます。

この後、
systemctl restart firewalld.service
として、firewalldを再起動すれば、httpアクセスが許可されたことが確認
できます。



なぜ、こういう教え方をするかというと、解りやすさはもちろんあると思うのですが、覚えたことを自分なりのイメージに変換できること、これが理解だと思っています。

それをほかの人に「教える」ことで、それを確認することもでき、その際はさらに理解度が深まるはずです。

ものを覚えるためには、まず自分が理解すること。
それを人に教えることで、初めて自分の知識になる。

あ、強制ではないので、あくまでもご参考まで。



ご清聴ありがとうございました