



LPI-Japan 主催 LPICレベル1技術解説無料セミナー



LPI-Japanアカデミック認定校 スキルブレイン株式会社 インストラクター 河原木 忠司



- ■LPIC(Linux技術者認定試験)の概要
- ■受験するに当たって -学習環境の準備-
- ■レベル1試験のポイント





LPIC(Linux技術者認定試験)の概要



LPICとは?

Skill Brain

LPIC(Linux技術者認定試験)は、特定非営利活動法人/Linux技術者認定機関「LPI」(本部:カナダ)が実施している全世界共通の統合的かつ最高品質のLinux技術者認定資格です。

■3つの特徴

- GLOBAL:世界標準資格 世界共通基準で認定を行っており、150カ国以上の方々に受験されています。
- NEUTRAL:中立・公正 LPICはベンダーやディストリビューションに全く依存せず、常に中立公正な立場で、より幅広い 見地からLinux技術力を評価しています。
- STANDARD:世界最大規模
 LPICは2010年9月末には、全世界で27万人以上が受験し、9万人以上の認定者が生まれています。これはLinux技術者認定試験では世界最大です。

※LPI公式サイトより



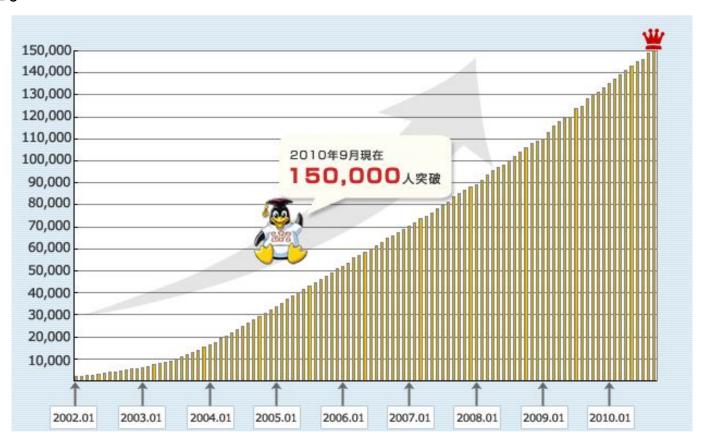
中立

公正

世界



LPICは2010年9月末には日本国内での累計受験者数が15万人を突破しました。



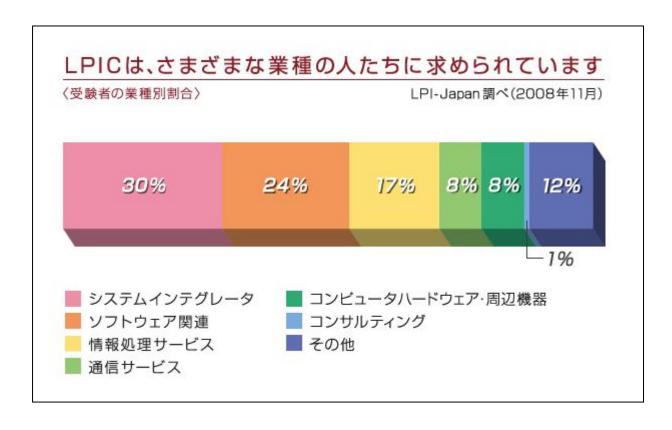
※LPI公式サイトより







- ■「営業効果の高い上位20資格」や「受験費用・講習代などの経費を補助 している上位20資格」で上位にランキング(日経ソリューションビジネス調査)
- ■さまざまな業種の受験者層



※LPI公式サイトより

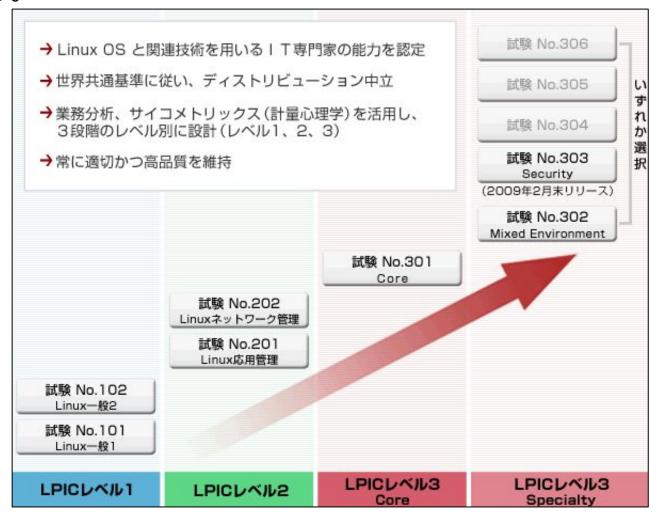




LPIC試験の構成

Skill Brain

■3段階のレベル分けがされており、レベル1試験が入り口の位置づけとなっている。







各レベルのおおよそのイメージ

Skill Brain

■レベル1

Linuxホストについて一通りの操作ができる。

- 101 [Linux一般1]: 基本的なコマンドの使い方など。
- 102【Linux一般2】: システム管理的な操作など。

2科目とも合格すると、レベル1認定

■レベル2

全般的なサーバー構築・管理などの操作ができる。

■レベル3

特定の分野について特化した知識を身につけている。





■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

太字になっている部分が本日このあと紹介する範囲です。

※主題に設定されている全項目を網羅して紹介するわけではありません。

※公式サイトURL

http://www.lpi.or.jp/lpic1/range/







受験するに当たって -学習環境の準備-





受験するに当たって -学習環境の準備- Skill Brain

■想定される学習環境



• 仮想環境









受験するに当たって -学習環境の準備- Skill Brain

■仮想環境導入のメリット

- 既存のOS環境をそのまま利用して、Linux環境を構成することができる。
- 複数の環境を容易に利用することができる。
- 環境を破壊するような操作を試みることもできる。
- テキストの付録などを利用し、手軽に環境を構築することができる。



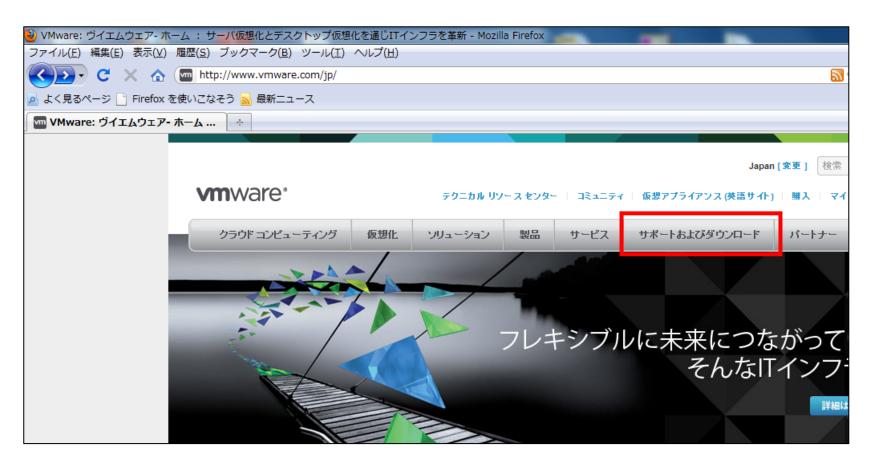


仮想環境(VMWare Player)の導入

Skill Brain

■VMWare

http://www.vmware.com/jp/







仮想マシンの作成



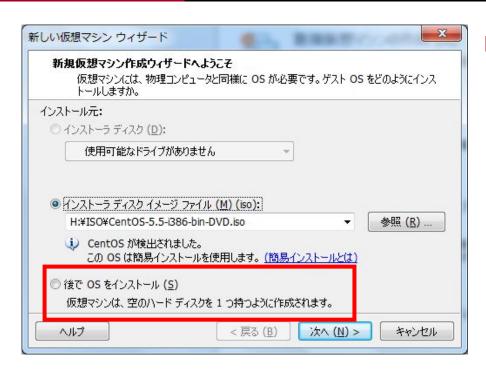
- ■[新規仮想マシンの作成] ーからインストールを行う場合に利用。
- ■[仮想マシンを開く] 既存の仮想マシン環境が存在する場合 に利用。
 - ※テキストによっては付録で仮想マシン 環境を用意してくれているものもある。





仮想マシンの作成

Skill Brain

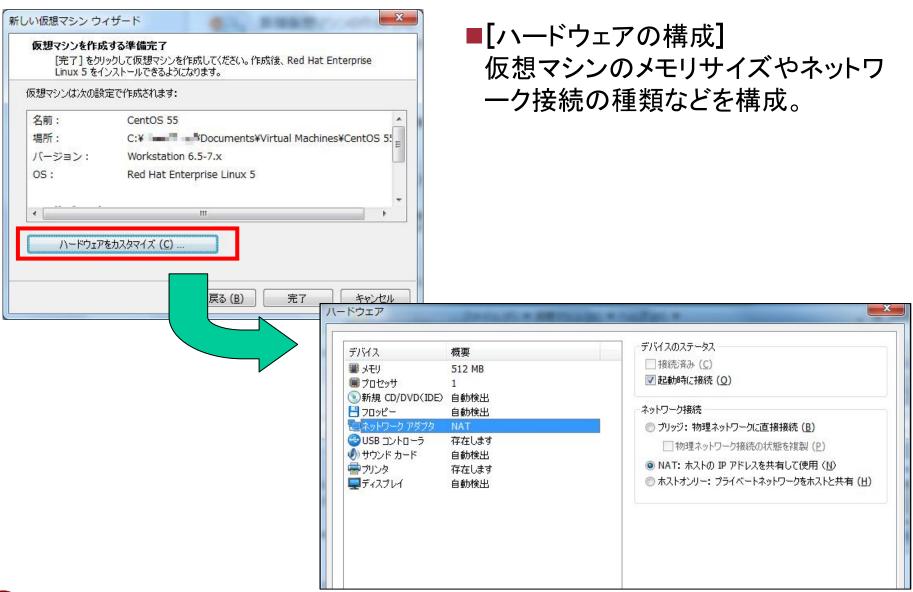


■[後でOSをインストール]を選択 [インストーライメージファイル]を選択 し、簡易インストール機能を利用する こともできるが、自動的にインストール 処理が実行されるため、インストール 構成をカスタマイズできない。





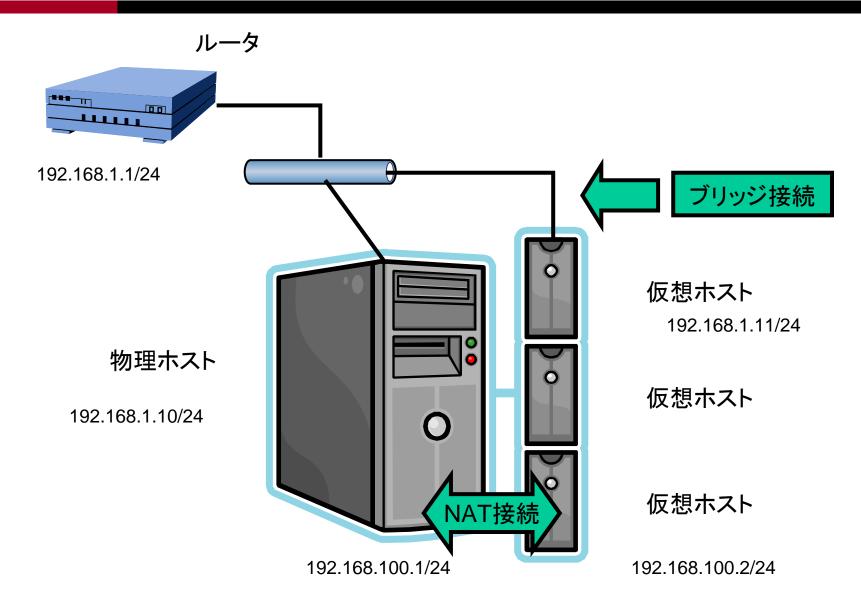
仮想マシンの作成







ネットワーク接続の種類







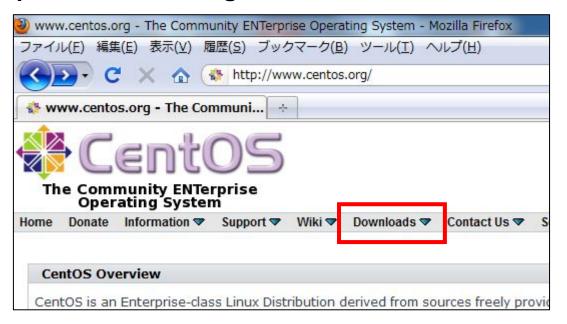
Linuxのインストール

Skill Brain

- ■各ディストリビューションのインストールイメージ(iso)ファイルを入手。
- ■インストールイメージファイルをマウントし、仮想ホストを起動。
- ■インストーラが起動するので、セットアップを実行。

CentOS公式サイト

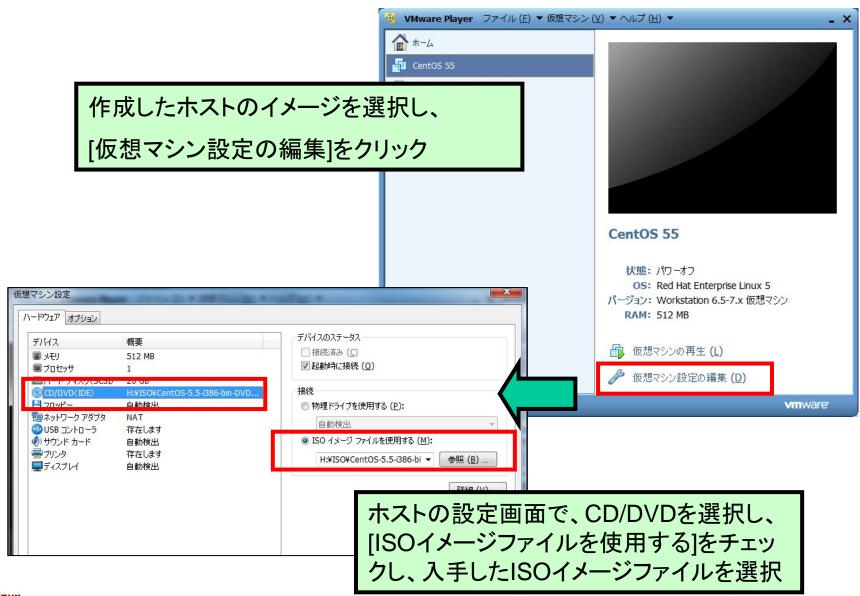
http://www.centos.org/







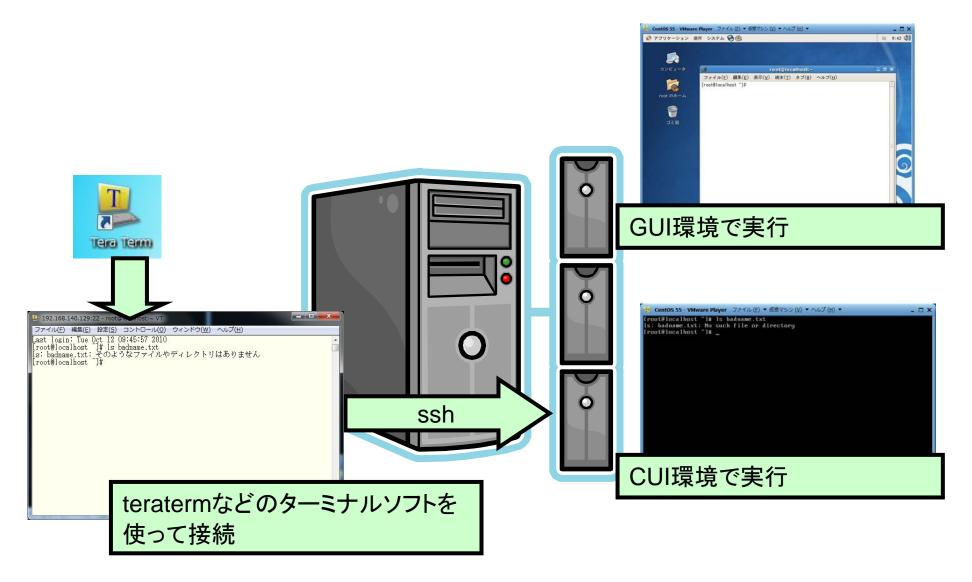
Linuxのインストール













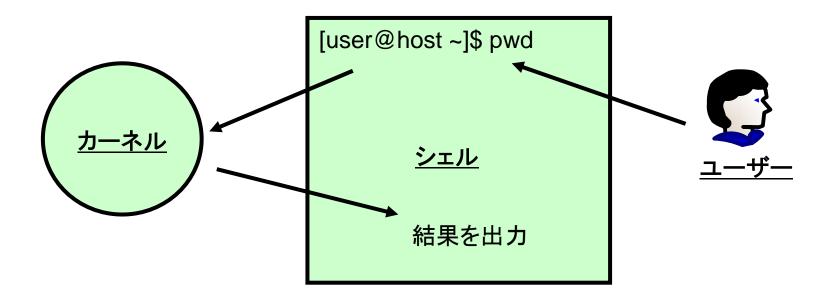




LPICレベル1 試験のポイント













[user@localhost ~]\$ ls —I /etc

プロンプト

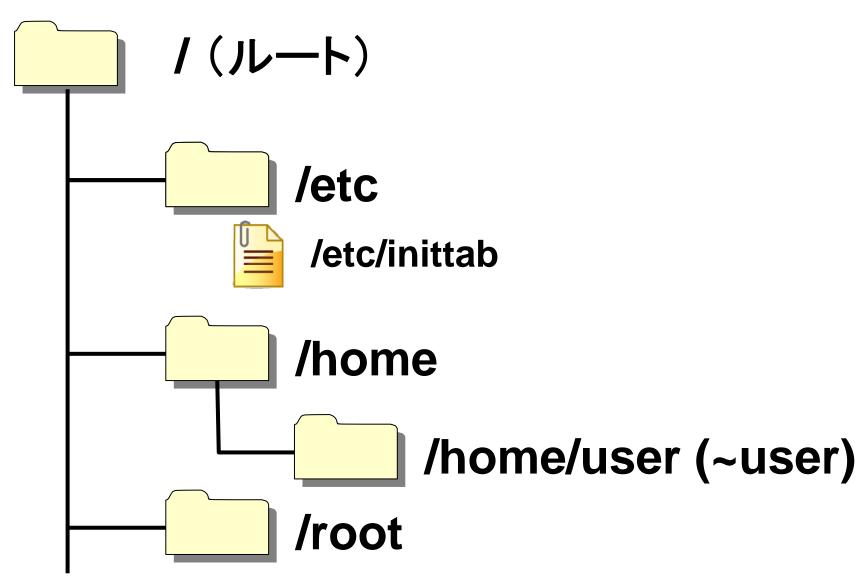
コマンド オプション

引数

| プロンプト | ログインしているユーザー名やホスト名、 カレントとなっているディレクトリを表示 |
|-------|--|
| コマンド | 実行するコマンドを指定 |
| オプション | 必要に応じて、オプションを設定し、コマンドの挙動を変えることができる |
| 引数 | 必要に応じて、引数を設定し、コマンド実 行時に値を渡すことができる。 |











パッケージ管理形式

- ■Linuxディストリビューションでは、アプリケーションをパッケージという単位で、管理している。
- ■それぞれのパッケージ管理形式で、インターネット上のリポジトリからパッケージを入手できる機能が用意されている。

| パッケージ管理形式 | コマンド | 設定ファイル |
|-----------|---------------|------------------------------------|
| RPM形式 | rpm, yum | /etc/yum.conf /etc/yum.repos.d/ |
| Debian形式 | dpkg, apt-get | /etc/sources.list |





yumコマンド

| 使い方 | 説明 |
|------------------------|--|
| yum install パッケージ名 | 指定したパッケージをインストールする |
| yum search キーワード | 指定したキーワードによりパッケージを検索する |
| yum check-update | アップデート対象となるパッケージを一覧表示する |
| yum update パッケージ名 | 指定したパッケージをアップデートする パッケージ名を省略し、インストール済みのパッケ ージをアップデートすることもできる |
| yum grouplist | パッケージグループを一覧表示する |
| yum groupinstall グループ名 | 指定したパッケージグループに属するパッケージ をインストールする |





必要なパッケージの導入

Skill Brain

■既存の仮想マシンイメージを利用する場合、以下のパッケージがインストールされていない可能性がある。

yum groupinstall "X Window System"

文環境の学習で
yum groupinstall "GNOME Desktop Environment"

文理 主題106

文環境の学習で
利用

yum groupinstall "MySQL Database"

主題105

SQLコマンドの学習で利用
yum install ntp

主題108

NTPの学習で利用





ランレベルの設定

Skill Brain

■ランレベルとは?

Linuxシステムの稼働状態を表す。

• init 3 ランレベルの変更 telinit 5

• runlevel 現在稼働しているランレベルの表示

• /etc/inittab initの設定ファイル・デフォルトのランレベルの設定

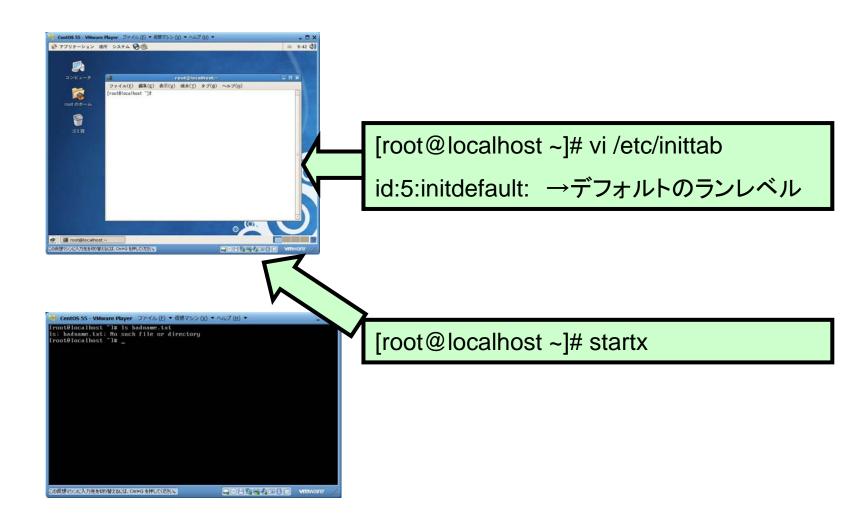
■CentOSのランレベル

| ランレベル | 説明 |
|-------|-----------------|
| 0 | システムの停止 |
| 1 | シングルユーザーモード |
| 3 | マルチユーザーモード(CUI) |
| 5 | マルチユーザーモード(GUI) |
| 6 | システムの再起動 |





X環境の確認







MySQL環境の確認

| [root@localhost ~]# /etc/init.d/mysqld start MySQL を起動中: [OK] [root@localhost ~]# mysql -u root Welcome to the MySQL monitor. Commands end with; or ¥g. Your MySQL connection id is 2 Server version: 5.0.77 Source distribution | mysqldの起動 mysqldに接続 |
|---|------------------------|
| Type 'help;' or '¥h' for help. Type '¥c' to clear the buffer. | |
| mysql> show databases; | データベース一覧を表示 |
| ++ Database | |
| ++ | |
| information_schema | |
| mysql | |
| test | |
| ++ | |
| 3 rows in set (0.00 sec) | |
| mysql> quit | mysqldへの接続を切断 |
| · · · · · · · · · · · · · · · · · · · | THYSYIC YO7]女心ととり間 |
| Bye [root@localhost ~]# | |
| [IOOL@IOCalliost ~]# | |





データベース・テーブルの作成

```
[root@localhost ~]# mysql -u root
mysql> create database testDB;
                                                                    データベースの作成
Query OK, 1 row affected (0.00 sec)
mysql> use testDB;
                                                                    データベースの切り替え
Database changed
mysql> create table staff( id int, name varchar(10) );
                                                                    テーブルの作成
Query OK, 0 rows affected (0.00 sec)
mysql> insert into staff values (1, 'Nakamoto');
Query OK, 1 row affected (0.00 sec)
                                                                    テーブルにレコードを追加
mysql> insert into staff values (2, 'Takagi');
Query OK, 1 row affected (0.00 sec)
mysql> select * from staff;
                                                                    テーブルの参照
    name
  1 | Nakamoto |
                                                                    102試験の出題範囲
  2 | Takagi |
+----+
2 rows in set (0.00 sec)
```





■SQL文の一部で、テーブル内のレコードの追加・検索・更新・削除などを 行なう際に使用する。

■主なDML文

- SELECT 列名 FROM テーブル名 WHERE 列名 = 値;
- INSERT INTO テーブル名 (列名1, •••) VALUES (値1, •••);
- UPDATE テーブル名 SET 列名 = 値 WHERE 列名 = 値;
- DELETE FROM テーブル名 WHERE 列名 = 値;





学習の方法について

Skill Brain

■実践的なスキルを身につけるには、システム管理などの際に使うであろう具体的な操作などをイメージし、いろんな項目を組み合わせながら、実践してみるとよい。

■シナリオ例

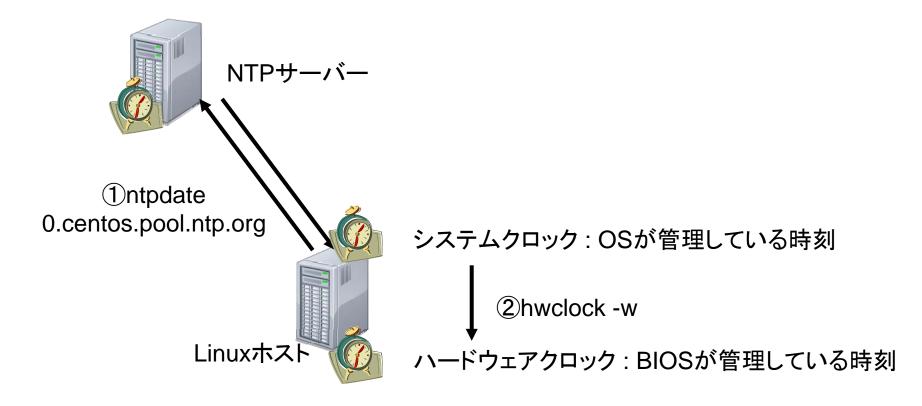
- NTPサーバーと時間を同期し、正確なシステムクロック運用を行う。
- 複数のコマンド実行処理をシェルスクリプトで処理する。
- 作成したシェルスクリプトが定期的に繰り返し実行されるようにする。





NTPサーバーとシステムクロックの同期

Skill Brain



※pool.ntp.org 公開NTPサーバーの情報を共有し、ラウンドロビンでアクセス させるプロジェクト





時刻設定関連コマンド・サービス

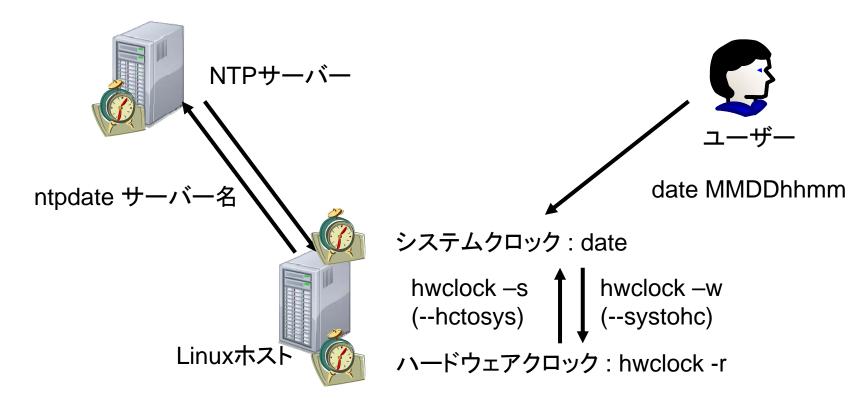
Skill Brain

■date:システムクロックの表示・設定

■hwclock:ハードウェアクロックの表示・設定

■ntpdate: NTPサーバーとシステムクロックの同期

■ntpd:NTPサーバーサービス。設定ファイルは/etc/ntp.conf







dateコマンドの応用例

Skill Brain

./timecheck.sh > timecheck-`date "+%m%d"`.log 2>&1

- ■./シェルスクリプト名 シェルスクリプトの実行。
- ■>, 2>&1 リダイレクト
- ■`コマンド` コマンドを実行した結果を出力
- ■date +%~ 指定した日時情報のみを出力

| 形式 | 説明 |
|----|----------|
| %Y | 年 |
| %m | 月(01~12) |
| %d | 日(01~31) |
| %H | 時(00~23) |
| %M | 分(00~59) |





リダイレクト

Skill Brain

- ■通常、コマンドの結果は画面に出力されるが、これをファイルに出力するように切り替えることができる。
- ■同じ画面に出力されるものでも通常の結果(標準出力)と、エラー(標準エラー出力)で区別されている。

| 書式 | 説明 | |
|------------------|---------------------------------|--|
| コマンド > ファイル | 標準出力の結果をファイルに書き込む(上書き) | |
| コマンド >> ファイル | 標準出力の結果をファイルに書き込む(追記) | |
| コマンド 2> ファイル | 標準エラー出力の結果をファイルに書き込む(上書き) | |
| コマンド > ファイル 2>&1 | 標準出力・標準エラー出力の結果をファイルに書き込む (上書き) | |
| コマンド くファイル | ファイルの内容をコマンドの標準入力に送る | |
| コマンド 〈〈 終了文字 | 終了文字が表れるまで標準入力に送る | |





引用符(クォーテーション)

Skill Brain

- '~': シングルクォーテーション 強い抑止。クォーテーション内の文字を文字データとして扱う。
- ■"~":ダブルクォーテーション 抑止。クォーテーション内の文字を文字データとして扱う。一部の記号に ついては抑止しない。\$変数名など
- `~ `: バッククォーテーション コマンドを実行し、文字情報として出力する。





シェルスクリプトの実行

Skill Brain

■テキストファイルの中にコマンドを記述しておき、これをまとめて実行することができる。

[root@localhost ~]# cat timecheck.sh
#!/bin/bash
ntpdate 0.centos.pool.ntp.org
hwclock -w
[root@localhost ~]# chmod a+x timecheck.sh → スクリプトには実行権限を与える
[root@localhost ~]# ./timecheck.sh → カレントディレクトリに存在するスクリプトを実行する場合、/スクリプトを実行する場合、/スクリプトを実行する場合、/スクリプトをと記述





シェルスクリプトの作成に必要な知識

Skill Brain

■変数

- 変数名=値\$変数名で値を参照することができる
- 特殊な変数

- \$1, \$2 ...: 引数

- \$# :引数の数

■testコマンド、[]

条件を設定し、true、falseを返す

- ・数値による条件
 - 値1 -eq 値2:値1と値2が等しい
 - 値1 -nq 値2:値1と値2が等しくない
 - 値1 -ge 値2:値1が値2以上
 - 値1 -gt 値2:値1が値2より大きい
 - 値1 -le 値2:値1が値2以下
 - 値1 -lt 値2:値1が値2より小さい
- ファイル属性による条件
 - -f ファイル名:ファイルが存在する
 - -x ファイル名:ファイルが実行可能である





```
■<u>if文:条件分岐</u>
```

```
if 条件式
then
true(真)の場合の処理→条件に合致した場合の処理
else
false(偽)の場合の処理→条件に合致しなかった場合の処理
fi
```

■case文:条件分岐(複数の処理)

```
case 変数名 in
値1) 処理1 ;;
値2) 処理2 ;;
default) 上記の値以外の場合の処理 ;;
esac
```







- ■for文:値リストを利用した繰り返し処理 for 変数名 in 値リスト do 値リストの値に対する処理 done
- ■while文:条件を指定した繰り返し処理 while 条件式 do 条件に合致している限り繰り返す処理 done





制御構文を使ったシェルスクリプト

Skill Brain

```
[root@localhost ~]# ./iftest.sh
Usage: ./iftest.sh arg
[root@localhost ~]# ./iftest.sh iftest.sh
#!/bin/bash
if [ $# -eq 1 ]
then
  cat $1
else
  echo "Usage: ./iftest.sh arg"
fi
[root@localhost ~]# ./iftest2.sh iftest2.sh
#!/bin/bash
if cat $1
then
  echo "success"
else
  echo "failrure"
fi
success
```

「\$# -eq 1」で引数が1つ指定されているときではないとエラーとして処理する。

引数が1つ指定されている場合、「cat \$1」で引数の値を参照し、catコマンド を実行している。

if文の条件にはコマンドを指定することもできる。コマンドの成功、失敗によって、true、falseを返す。

この例ではコマンドの実行が成功し、 iftest2.shの内容を表示できているため、最終行に「success」と表示されている。





制御構文を使ったシェルスクリプト

Skill Brain

[root@localhost ~]# cat fortest.sh #!/bin/bash for var in aaa bbb do echo \$var done [root@localhost ~]# ./fortest.sh aaa bbb [root@localhost ~]# cat fortest2.sh #!/bin/bash for fileName in *.sh do cp \$fileName \$fileName.bak done [root@localhost ~]# Is *.sh* fortest.sh fortest2.sh iftest.sh timecheck.sh [root@localhost ~]# ./fortest2.sh [root@localhost ~]# ls *.sh*

値リストに「aaa」、「bbb」という値を設 定。それらをechoコマンドで出力する。

値リストにワイルドカードを利用し、合致したファイル名を値リストとして構成することができる。

この例では、「~.sh」という拡張子が ついているファイルがあれば、 「~.sh.bak」という名前のバックアップ ファイルを作成する。

fortest.sh fortest2.sh iftest.sh timecheck.sh fortest2.sh.bak iftest.sh.bak timecheck.sh.bak





制御構文を使ったシェルスクリプト

Skill Brain

```
[root@localhost ~]# cat fortest3.sh
#!/bin/bash
for num in `seq 5`
do
  echo $num
done
[root@localhost ~]# ./fortest3.sh
3
[root@localhost ~]# seq 2 5
3
[root@localhost ~]# seq 1 2 10
3
```

コマンドの実行結果を値リストとして構成することができる。

seqコマンドは指定した引数までの連番を出力する。引数が1つの場合、「1から引数の値まで」連番を出力。

引数が2つの場合、「引数1から引数2の値まで」連番を出力。

引数がの場合、「引数1から引数3の値まで、引数2の値の分を増減値として」連番を出力。





スクリプトを繰り返し実行(cron)

Skill Brain

■ジョブのスケジューリング

- cron:繰り返し実行。crontabコマンドで設定
- at:1回だけ実行。atコマンドで設定

[root@localhost ~]# crontab -e → crontabエントリにジョブを追加する。
no crontab for root - using an empty one crontab: installing new crontab
[root@localhost ~]# crontab -l → 追加したジョブの内容を確認する。

0 * * * * /root/timecheck.sh





crontabの書式

Skill Brain

■分 時 日 月 曜日 実行する処理

| 分 | 0~59で指定 | |
|----|-------------------------|--|
| 時 | 0~23で指定 | |
| 日 | 1~31で指定 | |
| 月 | 1~12で指定、もしくはjanなど文字列で指定 | |
| 曜日 | 0~7で指定。0と7が日曜日、1が月曜日・・・ | |

■ワイルドカード、複数の値の指定

- 0 * * * * → 毎時0分に処理を実行(「分」以外の部分はすべてOK)
- 0 */2 * * * → 2時間おきに処理を実行
- 0 22 * * 2,4 →火曜日と木曜日の午後10時に処理を実行





/etc/crontabファイル

Skill Brain

■crontabコマンドで追加したジョブは、ユーザーごとに構成することができるcronジョブ。このほか、システム用のcronジョブとして/etc/crontabの内容を参照して、実行する。

[root@localhost ~]# cat /etc/crontab SHELL=/bin/bash PATH=/sbin:/bin:/usr/sbin:/usr/bin MAILTO=root HOME=/

run-parts

01 * * * * root run-parts /etc/cron.hourly

02 4 * * * root run-parts /etc/cron.daily

22 4 * * 0 root run-parts /etc/cron.weekly

42 4 1 * * root run-parts /etc/cron.monthly

[root@localhost ~]# ls /etc/cron.daily/

00webalizer certwatch logrotate prelink

Oanacron cups makewhatis.cron rpm

Ologwatch inn-cron-expire mlocate.cron tmpwatch

/etc/cron.*という名前のディレクトリの中に存在するスクリプトを実行する。

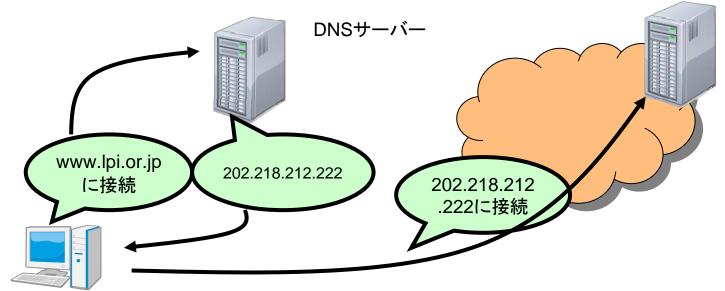




■ターミナルソフトの利用やNTPサーバとの同期の際、IPアドレス、ホスト名を指定



- ■TCP/IPを利用した通信では、IPアドレスを接続先・接続元のホスト情報として利用。
- ■コマンドの実行時など、ホスト名で指定することもできる。その場合、名前解決という処理が行われ、IPアドレスが取得される。







TCP/IP設定

Skill Brain

| 設定項目 | コマンド名、ファイル名 |
|-----------------|------------------|
| IPアドレス、サブネットマスク | ifconfig |
| デフォルトゲートウェイアドレス | route |
| DNSサーバーアドレス | /etc/resolv.conf |

[root@localhost ~]# ifconfig eth0 192.168.120.200 netmask 255.255.255.0

[root@localhost ~]# route add default gw 192.168.120.2

[root@localhost ~]# cat /etc/resolv.conf

search localdomain

nameserver 192.168.120.2

設定ファイルでは、このほか、以下のものが出題されます

•/etc/hosts

ローカルな名前解決

/etc/nsswitch.conf

名前解決の順序などを設定。

(ローカルなhostsファイル→DNSサーバー)





設定の確認

Skill Brain

[root@localhost ~]# ifconfig eth0

eth0 Link encap:Ethernet HWaddr 00:0C:29:0A:7A:DD inet addr:192.168.140.129 Bcast:192.168.140.255 Mask:255 (略)

ethOインターフェースに割り振られているIPアドレスなどを確認

[root@localhost ~]# ping 192.168.140.2

PING 192.168.140.2 (192.168.140.2) 56(84) bytes of data.

64 bytes from 192.168.140.2: icmp_seq=1 ttl=128 time=0.209 ms

64 bytes from 192.168.140.2: icmp_seq=2 ttl=128 time=0.388 ms

--- 192.168.140.2 ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1191ms rtt min/avg/max/mdev = 0.209/0.298/0.388/0.091 ms

[root@localhost ~]# host www.lpi.or.jp www.lpi.or.jp has address 202.218.212.222 ホスト192.168.140.2との疎通を 確認する。

Linuxのpingコマンドはデフォルトでパケットを送り続ける設定になっているため、ctrl+cで停止。

www.lpi.or.jpを名前解決

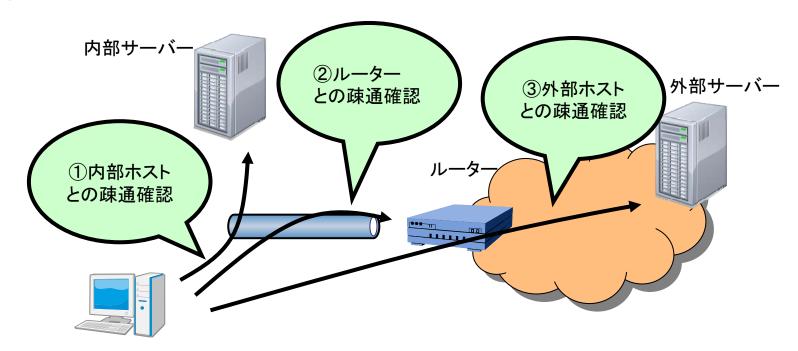




TCP/IP設定のトラブルシュート

Skill Brain

- ■疎通確認を行う場合には、近くのホストから遠くのホストへ実行
 - →トラブルが起きている箇所の切り分け
 - →外部ホストとの疎通確認はtracerouteなども活用
- ■ホスト名での通信が可能なホストに対しては、ホスト名、IPアドレスそれぞれで 疎通確認を行う
 - →名前解決絡みのトラブルとの切り分け





参考資料について

Skill Brain



Linux 教科書 LPICレベル1第4版

中島能和 (著) / 濱 野賢一朗 (監修) 2009/05/12発売 出版社:翔泳社 552ページ 定価3,990円 ISBN- 9784798119311



Linuxサーバー構築標準教科書(Ver1.0.2)

詳しくは下記URLで

http://www.lpi.or.jp/linuxservertext/

発行:エルピーアイジャパン



LPI認定試験 LPICレベル1《101/102》リリース3対応

橋本智裕、中島能和(監) 2009/3/26発行 出版社:秀和システム 608ページ 定価2,835円 ISBN- 978-4-7980-2229-1



Linuxクックブック——Linuxを120%使いこなすレシピ集

Carla Schroder 著、林 秀幸 訳 2005/10発行 出版社: O'Reilly Japan 444ページ 定価3,780円 ISBN4-87311-248-6





質疑応答についてはお気軽にお声掛けください。

ご静聴ありがとうございました。

http://www.skillbrain.co.jp info@skillbrain.co.jp

