

Linux コマンド

[Linux]

複数コマンド

複数にわたるコマンドを対話形式で

- ・行継続のコマンドは、変数 PS2 通常は >
- ・ループ構造を使う

例

基本

```
$for var in ary1 ary2 ary3
> do
> echo $var
> . . . 一連のコマンド . . .
> done
```

再帰的にパターンマッチしたファイルを指定フォルダにコピー

```
$ for f in `find -name "*.java"`
> do
> cp $f /work;
> done
```

複数のコマンドを続けて入力

- ・コマンドセパレータ (セミコロン) を利用

```
$ ls;ps
```

ヒストリ

- ・コマンドはシェルに解釈 (ワイルドカードの展開やコマンドの置き換え) される前にヒストリリストに保存される
- ・シェル変数 HISTSIZE で管理
- ・ヒストリリストの表示には、シェル組み込みの history コマンドを利用する

ヒストリ展開識別子

- ・C- は Ctrl キーを、M- は Meta キー (PC/AT 互換機では Alt キー) を表す

識別子	説明
!!	直前のコマンドを参照
!n	コマンド n を参照
!-n	現在の行番号から n を引いたコマンド
! 文字列	文字列で始まる最後のコマンド

!? 文字列	文字列を含む最後のコマンド
^ 文字列 1^ 文字列 2	文字列 1 を文字列 2 に置き換えた上で直前のコマンドを実行

ヒストリの編集

・ bash のデフォルトではヒストリの編集に Emacs と同様のキーバインドが設定されている

キー	説明
C-p	1 行前へ ()
C-n	1 行後へ ()
C-b	1 文字左へ ()
C-f	1 文字右へ ()
C-a	行頭へ
C-e	行末
C-l	現在行を最上位にして画面をクリア
M-<	ヒストリの先頭へ
M->	ヒストリの末尾へ
C-d	カーソルの 1 文字を削除
C-k	カーソル位置から行末までを削除 (キル)
C-y	カット (キル) した文字をペースト (ヤंक)
M-d	単語を削除
C-r text	text を後方検索
C-s text	text を前方検索

コマンド置き換え

- ・ \$(コマンド) という記述があれば、コマンドの出力結果に置き換えられる
- ・ 出力結果を変数に代入することもできる

```
$ RCSIZE=$(wc -l / .bashrc)
```

- ・ ` コマンド ` でもコマンドの置き換えに使用できる
- ・ 書式で無効なメタキャラクタについて、特別な規則がある

ディレクトリツリー全体でコマンドを再帰的に適用

- ・ 再帰的とは、ディレクトリツリー以下の枝すべてにおいて同じコマンドを実行すること
- ・ コマンドの再帰的な実行は便利だが、影響が大きいため慎重を期す必要がある

再帰機能がオプションとして組み込まれている

- chmod には -R オプションが組み込まれている
- --recursive を使用してもよい

```
$ chmod -R g+w src
```

再帰的な実行によく使われる一般的な方法は find を使うこと

- find コマンド自体が再帰的
- ファイルとディレクトリの情報が再帰的に表示される

```
$ find src
```

- 表示する情報をさらに絞り込むには、-name オプションを使用する

```
$ find src -name "*.c"
```

- find に -exec オプションを指定すると検索結果に対してコマンドを実行することができる

```
$ find src -name "*.c" -exec chmod g+w {} \;
```