

# 情報リテラシ (第5回)

## - Unixの基礎(4) -

Editor : emacs

シェルスクリプト

(教科書 第7章、第8章)

## Emacs

- Unix上で最もよく用いられている
- 情報リテラシでは、メールや文書作成システムであるLaTeXで主として利用する。
- 6月からのJavaプログラム作成にも用いる。
- Emacsとは？
  - Richard M. Stallman氏らが開発したエディタ

## Emacsの起動と終了

- **emacs**
  - 新しいemacsの起動
  - `emacs &` とするとどうなるか試してみよ
- **emacs -nw**
  - Emacs起動する時に新しいウィンドウを開かない
- ウィンドウ操作
  - **C-x 2** : ウィンドウを2つにする。
  - **C-x 1** : ウィンドウを1つにする。
  - **C-x o** : ウィンドウの切り替え
- emacsの終了
  - **C-x C-c**
  - `yes`を入力する必要があるときがある(ファイルを更新した場合など)

## 文書作成(1)

- 基本的なカーソル操作
  - 右移動: **C-f**
  - 左移動: **C-b**
  - 上移動: **C-p**
  - 下移動: **C-n**
  - 文字消去: **C-d**
  - 現在のカーソルから行末まで消去: **C-k**
  - 文字は現在のカーソルの後ろに追加される。

## 文書作成(2)

- カット&ペーストの利用
  - 文字列の先頭で**C-Space**、最後で**C-w(M-w)**を入力することにより領域がバッファにコピーされる。
  - **C-y**を入力することにより現在のカーソルの後にバッファの内容がコピーされる。
    - **C-w** :現在の文字列が消去されバッファにコピーされる。
    - **M-w** :現在の文字列はそのままバッファにコピー

## ファイルの操作

- ファイルのオープン
  - **C-x C-f**を入力するとファイル名を聞いてくる。
    - 既存のファイルからの読み込み
- ファイルのセーブ(保存)
  - **C-x C-w**を入力するとファイルにセーブされる。新しいファイルを作る場合はファイル名を新規に入力する。
    - 新しいファイルへの書き込み
    - 既存ファイルの変更
      - 本当に変更する場合はyを入力

## ハマってしまった時

- **C-g** 間違った操作を取り消したい時
  - 実行途中で停止させたり、入力中のコマンドを中止したいときに利用
- **C-x u** 間違った処理を取り消したい時
  - 既に完了したコマンドをundoする場合に利用

## ファイルの作成・編集

### ファイルの作成

1. emacs sample
2. 文章入力
3. C-x C-w を入力後、Enterを入力
4. C-x C-c を入力してemacsを終了

### ファイルの編集

1. emacs sample
2. 文章編集
3. C-x C-w を入力後、Enterを入力
4. C-x C-c を入力して終了

# Emacsに関する雑談

- emacs と
- UNIX上でよく用いられるエディタにviがある。
- UNIXカーネルプログラマはviを使う人も多い。
- emacsは今後も必要か？
- Emacsは強力なカスタマイゼーション機能を持つため、初心者には難しいが慣れれば非常に使いやすいエディタとなる。

このページは  
授業中に  
配布します

## 仮名漢字変換(1)

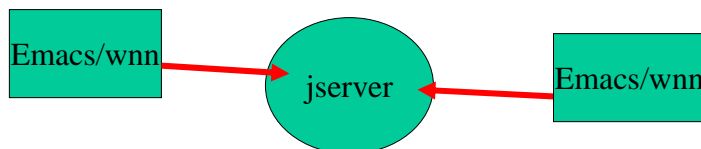
- Emacs上での漢字入力のON/OFF
  - C-¥
  - 注意: 一番最初に漢字入力をONした場合、初期ファイルを生成するための確認を聞かれる場合があるが、すべてyを入力する。
- 変換
  - Spaceキーで変換

## 仮名漢字変換(2)

- C-f
  - 文節の移動(前方へ)
- C-b
  - 文節の移動(後方へ)
- C-o
  - 変換候補の文節を伸張
- C-i
  - 変換候補の文節を短縮

## 仮名漢字変換に関する雑談

- 皆さんのノートPCでは、free Wnn を  
**漢字変換サーバ**として利用
- Wnnはネットワーク上の1つのサーバを複数マシンから共有できる。



ps agx | grep jserverでjserverが動いていることを確認できる

# シェルスクリプト(1)

- (時には外部コマンドも)を組み合わせる作業をするために新たに作ったコマンドがシェルスクリプト(教科書P.181に例)
- 例: WWWでのカウンタ等のプログラムを簡単にシェルを使って記述することが可能。
- 例: ファイル名の先頭に通し番号をつける等、*いちいち人間がやっていたら大変!* だと思ふことをシェルスクリプトで書いておくと便利。

このページは  
授業中に  
配布します

# シェルスクリプト(2)

- 例: WWWのカウンタ (ファイル名: counter.sh)

```
#!/bin/bash                # bashで実行することを指示

counter=/tmp/counter      # /tmp/counterに回数を記憶しておく
nums=`cat $counter`      # counterファイルの内容をnumsに読み込む

nums=`expr $nums + 1`    # 1を加える
echo $nums                # 画面にnumsを表示
echo $nums > $counter     # numsをcounterに保存
```

このページは  
授業中に  
配布します

## ポイント

#!に続けて利用するシェルを書く

#!/bin/bash

変数counterに文字列/tmp/counter  
を代入。

counter=/tmp/counter

nums=`cat \$counter`

Shift + @キー (バッククオート)  
cat \$counterを実行した結果を  
変数numsに代入。

nums=`expr \$nums + 1`

expr以下の式を評価する。  
(\$nums+1の計算を行う)

echo \$nums

echo \$nums > \$counter

標準出力に引数を表示。

2002/5/7

情報リテラシ

15

このページは  
授業中に  
配布します

## シェルスクリプト(3)

- 例: WWWのカウンタ (ファイル名: counter.sh)

\$ chmod +x counter.sh

実行可能の属性を付加

\$ echo 0 > /tmp/counter

counterに初期値0を入れておく

\$ ./counter.sh

1

counter.shを実行すると数字が  
1ずつ増える

\$ !!

同じコマンドを実行するときは!!で

2

2002/5/7

情報リテラシ

16

# 実習

- WWWのカウンタを実現するシェルスクリプトを作成し、実際に実行し動作を確認せよ。
- /tmp/counterのファイル無しで実行するとどうなるかを確認せよ。その原因を考えよ。
- 出力を数字だけではなく「Current Access Number is 1」のように説明文も同時に表示できるように改造してみよ。(ヒント:echoコマンド)