

【linux に慣れる】: windows のように使えます.

1. メニューからブラウザを立ち上げる
2. <http://www.yahoo.co.jp> にアクセスして, 「C 言語」で検索する
⇒C 言語の勉強のためのサイトがいろいろ見つかります. プログラミング経験のある人は自主的に勉強しましょう!

【unix のコマンド練習】: いろいろな操作は「コマンド」で実行します.

1. 端末エミュレータを立ち上げる
2. 端末エミュレータからブラウザを立ち上げる [%netscape]
3. 数値計算ソフト [%bc]
(ア) 1+2
(イ) 5+2*3
(ウ) a=2, b=3, c=a+b, c
(エ) ctrl-D(Ctrl を押しなが D を入力)でコマンド終了
4. 時刻を知る [%date]
5. 時計の表示 [%xclock]
6. カレンダーを表示する [%cal]
7. どんなファイルがあるかを調べる [%ls]
8. テキストファイルの中身を見る [%cat] [%less]
9. 上記 6, 7 コマンドの使用法を調べる [%man (コマンド名)]
10. 今まで入力したコマンドの一覧 [%history]
11. 端末エミュレータの終了 [%exit]

【C プログラムを作って実行する】

1. 端末エミュレータを立ち上げる
2. %emacs hello.c &でエディタを起動して下記の内容を入力する
・最後の「&」は「バックグラウンドジョブの実行」を指示しており, 端末エミュレータにプロンプトが帰ってきます. 「&」がなくても emacs の起動は正常に行われます.

=====プログラム始まり=====

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    printf("Hello C world!\n");
```

```
}
```

=====プログラム終わり=====

【超基本的なエディタの使い方】

- ・カーソルの移動 [%←] [%↓] [%→] [%↑]
- ・改行 [%ENTER]
- ・文字の削除 [%DELETE] [%C-d]

3. %ctrl-x-ctrl-s でファイルを保存する

4. 入力が終わったら、「ls」および「cat」コマンドでできたプログラムファイルを確認する.
5. 別の端末エミュレータを立ち上げる
6. `%gcc -o hello hello.c` と入力してプログラムをコンパイルする
7. 「ls」コマンドで hello という実行ファイルができていることを確認する.
8. `%. /hello` と入力してプログラムを実行する.
9. もし 5, 6 でエラーメッセージが出力されたり, 正常に実行がされなかったときは emacs エディタでプログラムを修正し, `%ctrl-x-ctrl-s` で修正結果を保存する.
7. `%ctrl-x-ctrl-c` で emacs エディタを終了する

【ファイルのコピー, 名前の変更, 消去】

1. hello.c を showname.c というファイル名でコピー(複製)する. `[%cp hello.c showname.c]`
2. showname.c を emacs エディタで編集して下記のように変更する. (名前は自分の名前にすること.)
=====プログラム始まり=====

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    printf("Hello C world!\n");
```

```
    printf("My name is Takeshi Fujino! \n");    ←追加場所
```

```
}
```

```
=====プログラム終わり=====
```

3. コンパイルする. `[%gcc -o showname showname.c]`
4. 実行する `[%./showname]`
5. showname を outname というプログラム名に変更する `[%mv showname outname]`
6. outname を実行する `[%./outname]`
7. outname を消去する `[%rm outname]`

【linux ログインのためのパスワードを変更する】

1. パスワード変更 `[%yppasswd]`

【Advance】: 演習時間が余った人のために

1. 本日のプログラムは, ソースファイルに書いた文字列を“オウム返し”に出力するだけなので面白くないですね.
2. たとえば, 計算をさせるならば, 出力部を `printf("%d\n", 3+4);`とやれば, 計算結果「7」を出力してくれます. hello.c をコピーして keisan.c を作ってコンパイル&実行してみましょう.
3. 上記の方法では, 小数の計算結果は出力できない. どうすればよいか? これはちょっと難しいかもしれませんね.
`[printf("%d\n", 3+4);を printf("%f\n", 3+4);とする.]`
4. 入力する数字をプログラム時に指定する必要があるのは大変ですね. プログラム実行時に, 任意の数字を入力して計算を実行する方法は, 次週紹介します. お楽しみに!