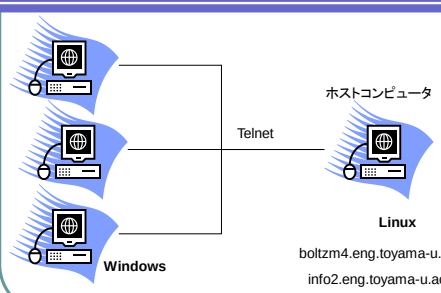
Linuxについて

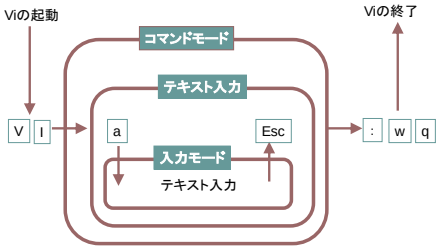
ネットワーク組織図



boltzm4.eng.toyama-u.ac.jp

info2.eng.toyama-u.ac.jp

Viエディター



Linux

Linuxは、フィンランドのヘルシンキ大学に在籍していたLinux. B. Torvalds氏が21歳のときに、教育・学習用としてのminixというUNIXパッケージに触発され、カーネルと呼ばれるOSの核の部分を自分で開発したのが始まりである。

OHP

What is Linux
- Unix clone
- Source Compatible with "UNIX"
- Binary Compatible

リィナス・トルヴァドゥス

UNIX（UNICS = Uniplexed Information and Computing System）

• UNIXは、1969年、アメリカAT&Tのベル研究所で、Ken ThompsonとDennis Ritchieという、たった2人の研究者の手によって開発された。

• 1973年、Dennis Ritchieは、「C」言語を開発し、プログラムを書き直した。

• 1977年、カリフォルニア大学バークレイ校からBSD(Berkeley Software Distribution)版が配布された。

Cシェル　FreeBSD　Mac OS X

It is called the shell, beacuse, like the shell of a nut or an egg, it is the part that we see from the outside. The inside part is called the kernel.

シェル（Shell）：ユーザ端末から入力された文字列を解釈し、その指示に従って仕事をするプログラム

カーネル（Kernel）：オペレーティングシステムの中枢で、プロセスやメモリ、ファイル等の管理を行う部分

Linuxのコマンド

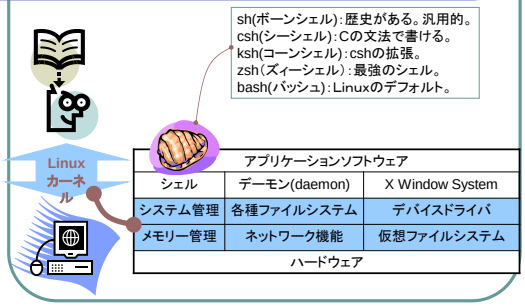
コマンド	機能
login	ログインする
exit	ログアウトする
history	コマンドの履歴を表示する（例：history　5）
which	コマンドの所在位置を表示する（例：which　man）
passwd	パスワードを変更する
users	ログインしているユーザー全員を表示する
last	ログインおよびログアウトの履歴を表示する
ls	ファイル名を表示する
pwd	カレント・ディレクトリを表示する
cd	カレント・ディレクトリを変更する

Linuxのコマンド

コマンド	機能
tree	ディレクトリをツリー表示する
cp	ファイルやディレクトリをコピーする（例：cp file1 file2）
mv	ファイルやディレクトリを移動する（例：mv file1 ¥dir1） ファイル名を変更する（例：mv file1 file2）
rm	ファイルやディレクトリを削除する（例：rm file1, 例：rm –r dir1）
mkdir	ディレクトリを作成する（例：mkdir dir1）
rmdir	ディレクトリを削除する（例：rmdir dir1）
ps	実行中のプロセスを表示する
jobs	バックグラウンドジョブを表示する
jobs -l	バックグラウンドジョブのプロセス番号を表示する
fg %jobs	ジョブをフォアグランドに切り替える

※ コマンドの後ろに&を付けると、バックグラウンドで起動される。

Linux

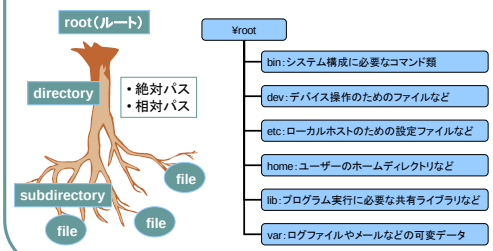


シェルの機能

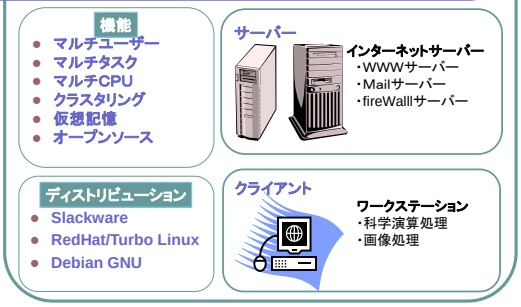
- ヒストリ機能（例：←↑↓→）
- 補完機能（例：¥h → **Tab** → ¥home）
- 自動処理（例：ログインスクリプト）
- パイプライン(例：ls –l | mail itoh@net)
  （複数のコマンドを組み合わせる）
- リダイレクト（例：ls > ¥home¥itoh/log.txt）
  （入出力を切り替える）
- コマンドの一括処理（make；make install）

ディレクトリ(directory)

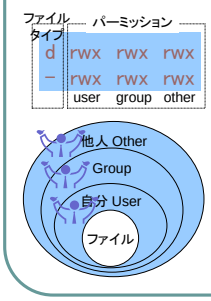
木をさかさまにしたツリー構造をとる



Linux



パーミッション

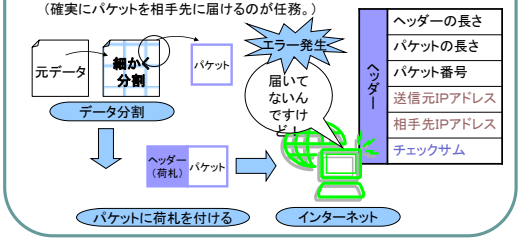


- d：ディレクトリ
- ー：ファイル
- r：読み取り可(read)
- w：書き込み可（write）
- x：実行可（execute）

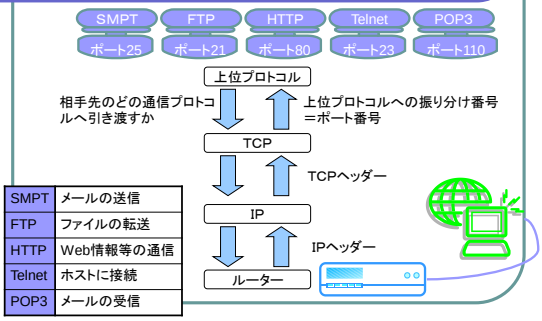
コマンド	機能
ls -la	全てのファイル名と属性を表示する
chmod	ファイルやディレクトリのパーミッションを変更する（例：chmod ug+wx msg.snd）
chown	ファイルやディレクトリの所有者を変更する

TCP/IP

TCP：Transmission Control Protocol：1973年に開発された通信プロトコル。
（エラーが発生しても確実にパケットのやりとりを行う。）
IP：internet Protocol：1979年TCPからIPを分離して別規格とした。
（確実にパケットを相手先に届けるのが任務。）

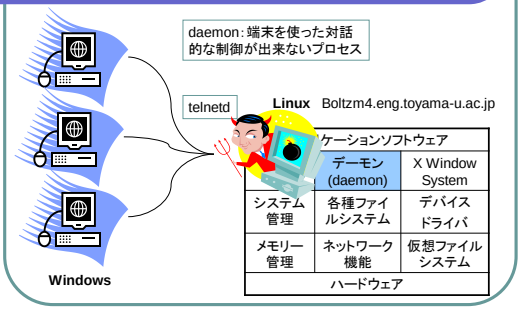


TCP/IP



SMPT	メールの送信
FTP	ファイルの転送
HTTP	Web情報等の通信
Telnet	ホストに接続
POP3	メールの受信

daemon



Vi（ファイルオープン時のキー操作）

キー	機能
Returnキー	改行
BackSpaceキー	左に遷移
Escキー	コマンドモードへ切り替え
vi ファイル名　Returnキー	ファイルを開く。または、新規に作成する
vi　Returnキー	新規ファイルを開く（名前は後で指定）

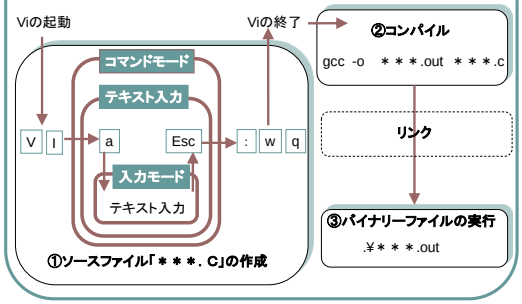
vi（カーソル移動関連のコマンド）

キー	機能
h,左矢印キー,BackSpaceキー	左に移動
j,下矢印キー	下に移動
k,上矢印キー	上に移動
l,右矢印キー、Spaceキー	右に移動
3h	3文字左に移動
3j	3行に移動
3k	3行上に移動
3l	3文字右に移動
G	ファイルの最終行に移動
1G	ファイルの先頭行に移動
7G	7行目に移動
Ctrlキー＋G	現在の行を状態表示行に表示

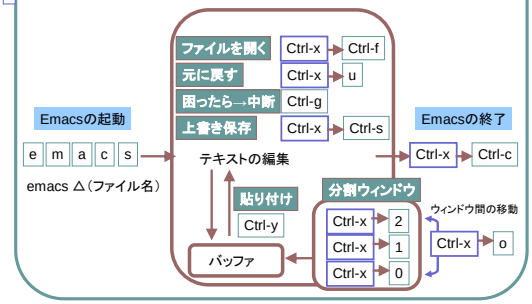
vi（テキスト入力関連のコマンド）

キー	機能
i テキスト入力　Escキー	カーソルの左にテキストを挿入
o テキスト入力　Escキー	カーソルが置かれている行の下にテキストを挿入
a　テキスト入力 Escキー	カーソルの左にテキストを追加
x	カーソルの置かれている文字を削除
dd	行を削除
U	カーソルが置かれている行の変更部分を全て取り消す
yy, Y	行をコピー
p	コピーまたは移動の対象をカーソルの右側（または行の下）に挿入

コンパイル



Emacs （Microsoft Wordと同様の利用が可能）



Emacs利用前の設定

① ファイル：.bashrcに
export△term =
vt100
を書き込む。

② Tera Termにおいて
Setup→Terminal
✓ term size = win size
にチェックを入れる。

③ Tera Termにおいて
Setup→Keybord
✓ Backspace key
✓ Delete key
✓ Meta key
にチェックを入れる。

Emacs（ファイル・ウィンドウ関連）

キー	機能
emacs ファイル名　Returnキー	ファイルを開く。または、新規に作成する。
emacs　Returnキー	新規ファイルを開く（ファイルは後で指定）。
Ctrl-x → Ctrl-f	起動後、ファイルを開く。
Ctrl-x → Ctrl-s	上書き保存する。
Ctrl-x → Ctrl-w	ファイルに名前をつけて保存する。
Ctrl-x → 2	現在カーソルのあるウィンドウを2分割する。
Ctrl-x → 1	現在カーソルのあるウィンドウ以外を削除する。
Ctrl-x → 0（ゼロ）	現在カーソルのあるウィンドウを削除する。
Ctrl-x → o（オー）	カーソルを分割されたウィンドウ間で移動する。
Ctrl-x → b	バッファーを切り替える。

Emacs（カーソル移動関連）

キー	機能
マウスや矢印のキーも使用可である	
Ctrl-f	右に移動 (forward)
Ctrl-b	左に移動 (backward)
Ctrl-p	上に移動 (previous line)
Ctrl-n	下に移動 (next line)
Ctrl-a	行の先頭に移動
Ctrl-e	行の末尾に移動
Ctrl-v	次ページに移動
Esc→v	前ページに移動
Esc→>	バッファ先頭に移動
Esc→<	バッファ最後に移動

Emacs（テキスト入力関連）

キー	機能
Delete, Backspace	カーソルの左側の文字を削除
Ctrl-d	カーソルの置かれている文字を削除
Ctrl-k	カーソル位置からその行の行末までを削除（削除リングへ）
コピー＆貼り付け，切り取り＆貼り付け関連	
Ctrl-space	マーク設定
Ctrl-w	マーク設定した箇所からカーソル前までを削除（削除リングへ）
Esc→w (またはAlt-w)	マーク設定した箇所からカーソル前までを削除せずにコピー（削除リングへ）
Ctrl-y	削除リングの内容をカーソルの右側にペースト

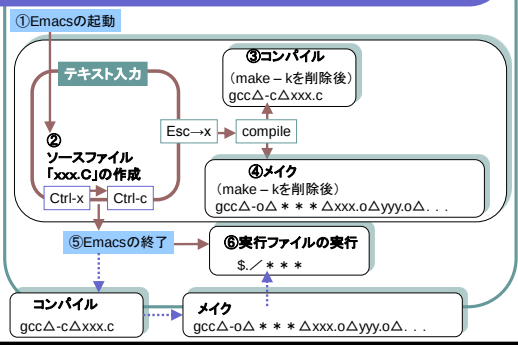
Delete, Backspace利用するためには、以下の設定をする！

Tera Termで
Setup→Keybord
✓ Backspace key
✓ Delete key
✓ Meta key
にチェックを入れる。

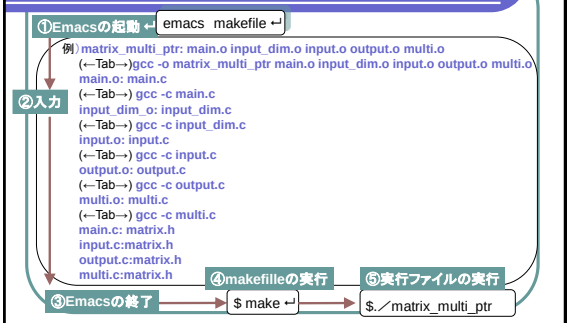
Emacs（検索・置換）

キー	機能
Ctrl-s	ファイルの終わりに向かって文字の検索。（検索の続行。）
Ctrl-r	ファイルの先頭に向かって文字の検索。（検索の続行。）
Ctrl-x →「query-replace△(検索する文字列)」の入力 → return キー →「(置換後の文字列)」の入力	文字の置換 置換するとき:「y」または「space」 置換しないとき:「n」 置換処理を終了したいとき:「q」
Ctrl-g	中断する。
Esc→x→goto-line	行番号を指定して移動する。

コンパイル



Makefileの作り方



プログラムの基本

- 文には、セミコロン（;）をつける
- コードを読みやすくする（ブロック単位で記述する）

```
int  main(void)
{
        ....
インデント  returen  0;
}
```

プログラムの本体は
main()関数

- 先頭から順番に処理される
- C言語は、関数で構成されている

C言語の命令

コマンド	機能
#include	他のファイルを読み込む
printf(“＊＊＊”);	＊＊＊が画面に出力される。
¥n	改行
%c	文字を出力
%d	整数を出力
%f	整数を出力
=	変数に値を入力する
scanf	キーボードから数値を入力する
&	変数の前に&をつける
＋－＊／	四則演算

C言語の命令

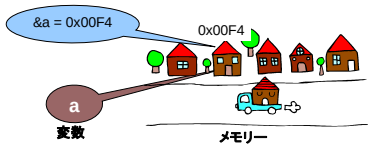
コマンド	機能
#	コンパイル前に実行される。（プリプロセッサ）
#define	マクロを定義し、置き換えることができる。
sin(＊)	サインを返す。
cos(＊)	コサインを返す。
pow(＊,＊)	べき乗を返す。
For文	繰り返し処理をする。（回数指定）
while文	繰り返し処理をする。（条件指定）
if文	条件に応じた処理を行う。
＞、＜	より大きい、未満
＞＝、＜＝	以上、以下
＝＝、！＝	等価、非等価

アドレス（address）

変数aの値がメモリ上の「どの位置」に記憶されているのかをすることが出来る。

printf("変数aのアドレスは%pです。¥n",&a);

コマンド	機能
&変数名	変数のアドレスを表す（アドレス演算子）

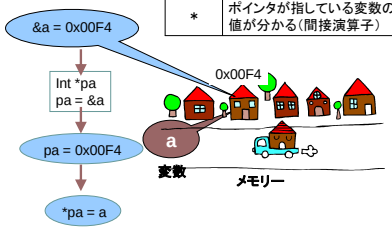


ポインタ（pointer）

ポインタ：アドレスを格納する特殊な変数

int *pa; （型名 *ポインタ名；）

コマンド	機能
*	ポインタが指している変数の値が分かる（間接演算子）



構造体

- 構造体型を宣言して、異なる型をまとめることができる。

ある会社の車を管理するプログラムを作成してもらいたい。

車のナンバー、ガソリン量、走行距離、車種などをまとめて管理すること。

異なる型の値をまとめて新しい型とする。
例）車のナンバー（int型）
ガソリン量（float型）

富山12－34

10．0 ℓ

「車」を表す構造体型の宣言

```
Struct Car{
  int num;
  float gas;
}

Main()
{
 struct Car car1;

 car1.num= 1234
 car1.gas = 10.0;
}
```

メンバ（num、gas）

ナンバーを格納

ｶﾞｿﾘﾝ量を格納

構造体型の変数car1を宣言

Car1のﾅﾝﾊﾞｰは1234です。

Car1のｶﾞｿﾘﾝ量は10.0ℓです。

ドット演算子

ファイルの入出力

ファイル・ポインター（ファイルの管理用）

ファイルをオープンする <fopen>

ファイルを読み/書きする <fscanf>/<fprintf>

ファイルをクローズする <fclose>

ファイルを書く場合

```
FILE *fp;
   :
fp = fopen("data.txt","w")
   :
fprintf(fp, "%f    %f¥n", v_old.x, v_old.y);
   :
fclose(fp);
```

ファイルを読む場合

```
FILE *fp;
   :
fp = fopen("data.txt","r");
   :
fscanf(fp, "%f    %f¥n", &v_old.x, & v_old.y);
   :
fclose(fp);
```

逆行列の計算

$$A=\begin{bmatrix}1 & 2\\ 2 & 5\end{bmatrix}$$

$$\begin{bmatrix}1 & 2 & 1 & 0\\ 2 & 5 & 0 & 1\end{bmatrix}$$

①2行目−1行目×2

$$\begin{bmatrix}1 & 2 & 1 & 0\\ 0 & 1 & -2 & 1\end{bmatrix}$$

②1行目−2行目×2

$$\begin{bmatrix}1 & 0 & 5 & -2\\ 0 & 1 & -2 & 1\end{bmatrix}$$

$$A^{-1}=\begin{bmatrix}5 & -2\\ -2 & 1\end{bmatrix}$$

$$①\begin{bmatrix}1 & 0\\ -2 & 1\end{bmatrix}\begin{bmatrix}1 & 2\\ 2 & 5\end{bmatrix}=\begin{bmatrix}1 & 2\\ 0 & 1\end{bmatrix}$$

$$②\begin{bmatrix}1 & -2\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 2\\ 0 & 1\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$\left(\begin{bmatrix}1 & -2\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ -2 & 1\end{bmatrix}\begin{bmatrix}1 & 2\\ 2 & 5\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}1 & -2\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ -2 & 1\end{bmatrix}A=I$$

$$A^{-1}=\begin{bmatrix}1 & -2\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ -2 & 1\end{bmatrix}=\begin{bmatrix}5 & -2\\ -2 & 1\end{bmatrix}$$

A^{-1}

$$①\begin{bmatrix}1 & 0\\ -2 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}=\begin{bmatrix}1 & 0\\ -2 & 1\end{bmatrix}$$

$$②\begin{bmatrix}1 & -2\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ -2 & 1\end{bmatrix}=\begin{bmatrix}5 & -2\\ -2 & 1\end{bmatrix}$$

ファン・デル・ポールの方程式

- 自励振動（リミットサイクル）を表す。
- 位置のエネルギーを時間をかけて蓄積し、瞬時に解放し速い速度で運動する。
- $\varepsilon = 0$ とすると単振動を表す。

$$\frac{d^2x}{dt^2}-\varepsilon(1-x^2)\frac{dx}{dt}+x=0 \qquad (\varepsilon>0)$$

変形

$$\frac{dx}{dt}=y$$

$$\frac{dy}{dt}=\varepsilon(1-x^2)y-x$$

＜例題＞

$t = 0.01$、$\varepsilon = 5$、$n = 2500$
青色：(-0.001,　0)
緑色：(　-3,　1.5)
水色：(　 3,　 4)