vi(vim)の使い方

情報システム・セキュリティ実験 I
(最所分)

2020/12/03	初版
2021/04/19	第2版
2021/04/21	第3版

20G470 竹原一駿

最所研究室

https://air.eng.kagawa-u.ac.jp/doku.php

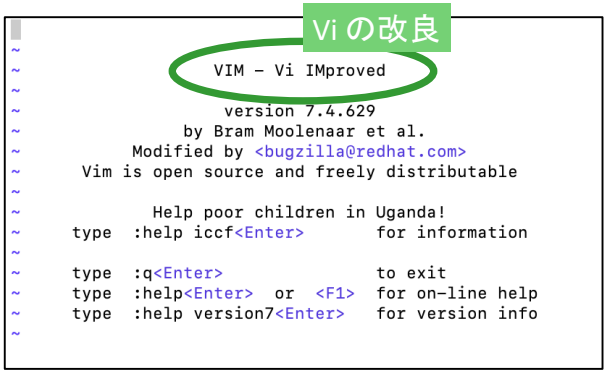
viとは

- ほとんどにUNIXシステムに最初から入っている
  - POSIXと呼ばれるUNIXの標準規格で定義
  - Emacsは規格上，定義されていない
    - ✓ Emacsが入っていない環境も少なくない

エディタ戦争

- 機能拡張されたvimがある
  - 拡張を除いたvimをviとしていることもある

- どのモードで動作しているのか注意する
  - ノーマルモード(viモード)
  - コマンドラインモード
  - 挿入モード(Insertモード)



`vi` の実行結果

viを立ち上げよう

■ プロンプトが出ている状態で，emacsと同様に，

vi［ファイル …］

と入力する

- たとえば`vi /var/log/secure`
- ファイルの指定がない場合は新規作成となり，指定した場合はその(それら)のファイルを順に編集する
- Rオプションで，Read Only モード
  - ✓ `vi -R /var/log/secure`
  - ✓ 書き込みたくないファイル(ログファイル等)に使う
- vimのチュートリアルを起動 `vimtutor ja`

viのモード

viモード

- カーソル移動や文字列の検索などを行う
- 起動直後のモード
- 挿入モードやコマンドラインモードに移行する

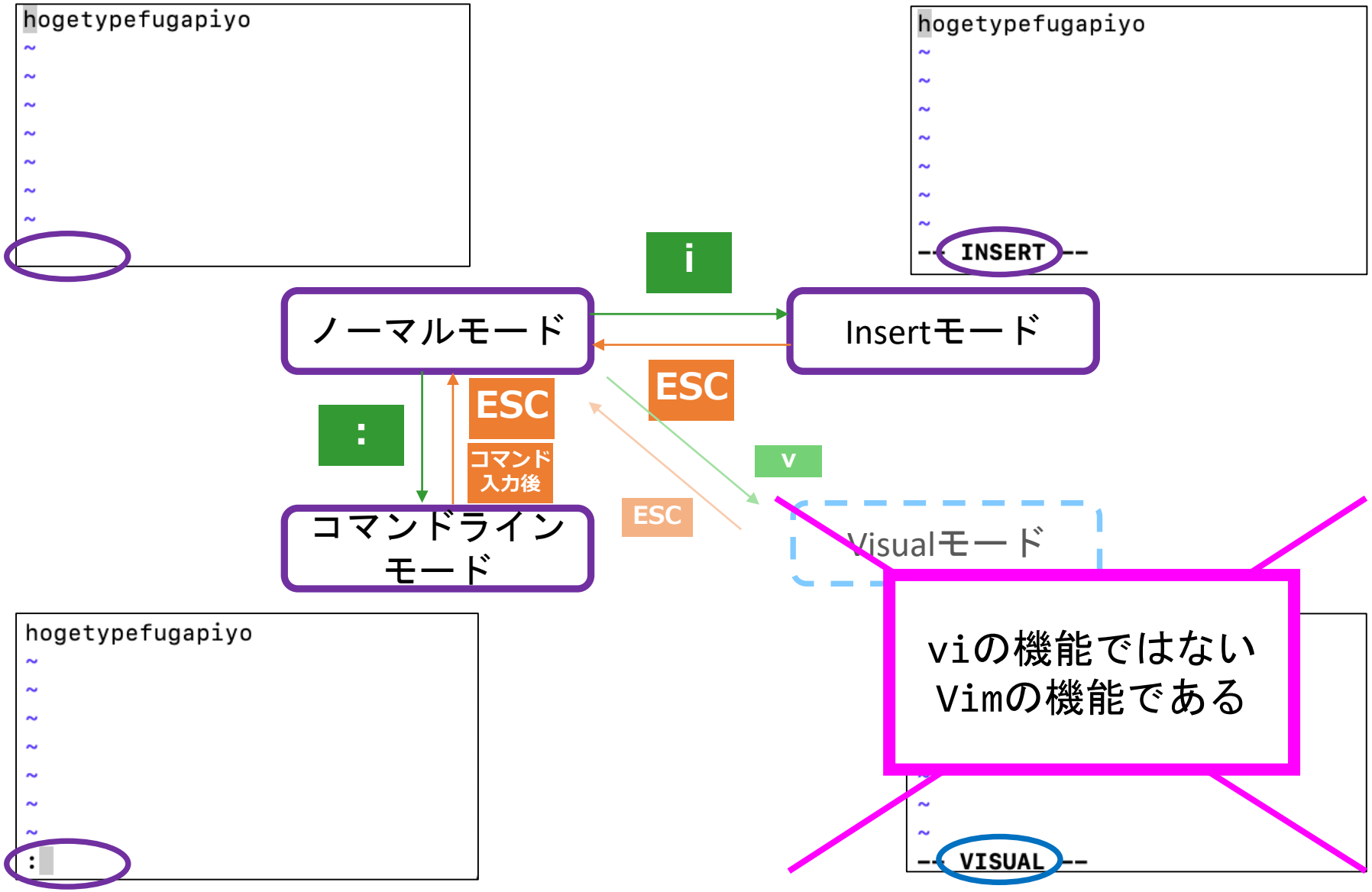
挿入モード

- 文字入力を行うモードである
- **挿入 (‘i’や‘A’などで移行)** または**上書き (‘r’ や ‘R’などで移行)**
- それ以外は**ESCキー** を入力することでviモードに戻る
- 1文字上書きで移行した場合は1文字入力後に自動的に戻る

コマンドラインモード

- コマンドを用いて編集する
- **‘:’ (コロン)で移行**
- ファイルの保存や終了，一括変換などを行う
- コマンドを実行すると自動的にviモードに戻る

モード切り替え


hogetypefugapiyo
hogetypefugapiyo
-- INSERT --
i
ノーマルモード
Insertモード
ESC
:
ESC
コマンド
入力後
v
ESC
Visualモード
viの機能ではない
Vimの機能である
hogetypefugapiyo
:
-- VISUAL --

ノーマルモードの基本操作

■ 入力位置の移動

l(エル)：右に1文字移動（カーソル移動キーの →）

j：1行下へ （カーソル移動キーの ↓）

k：1行上へ （カーソル移動キーの ↑）

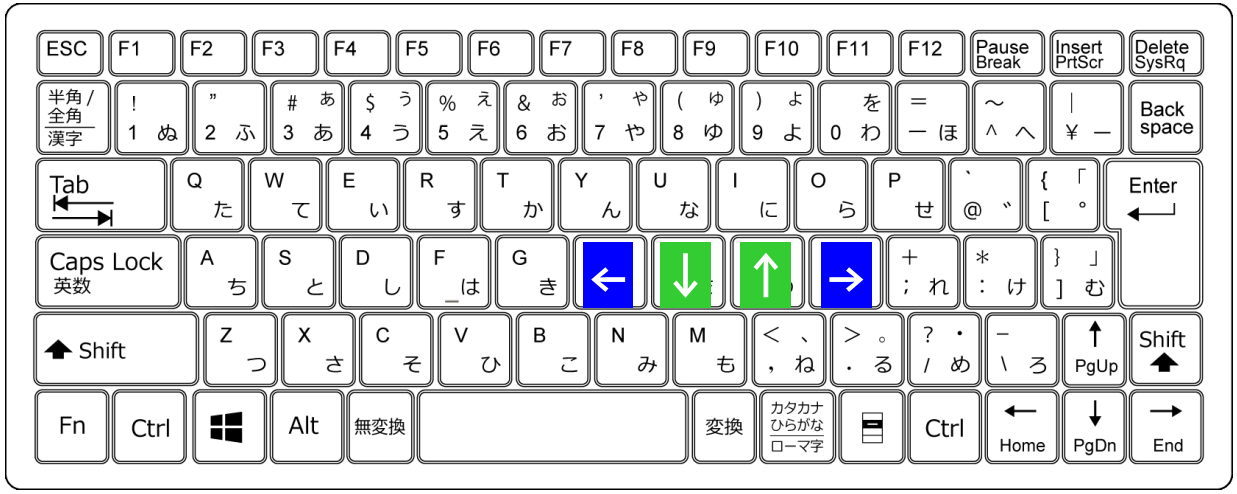
h：左に1文字移動 （カーソル移動キーの ←）

Enter：次の行の空白でない最初の文字の位置に移動

C-f：1ページ次へ (Ctrl押しながらf)

C-b：1ページ前へ

数字の後にG：指定した数字で示される行に移動する



キーボードのイラスト・無料イラスト素材倉庫/人体図イラスト・フリーダウンロード http://www.sharots.com/sozai/keyboard.html 2020/11/10

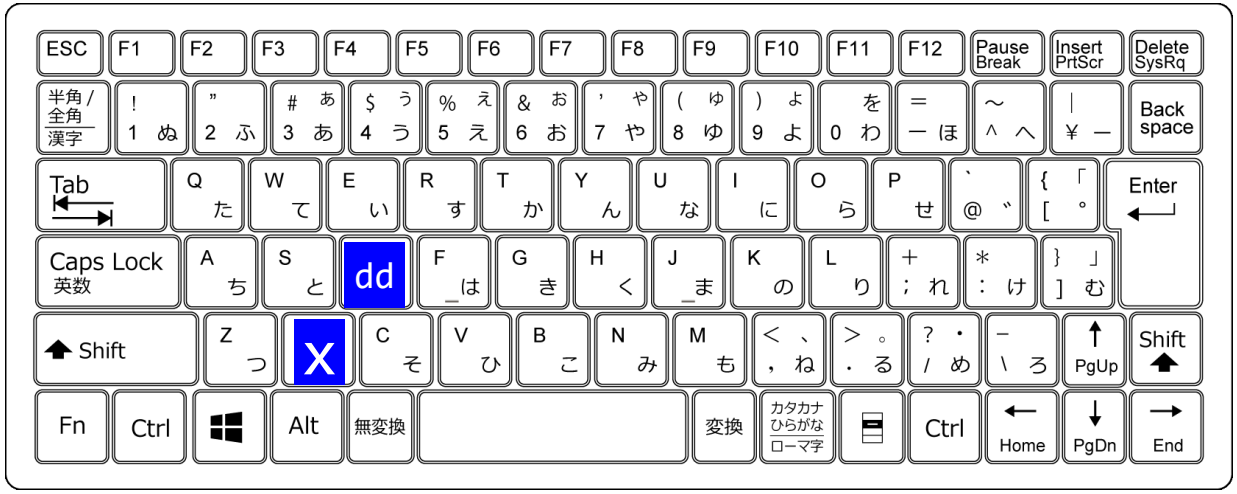
■ 削除 (内容は自動的にコピーバッファにコピーされる)

x ：現在位置の文字を1文字削除する

dd：現在行を削除する

■ 文字の修正

r ：現在位置の文字を1文字修正する



■ 検索

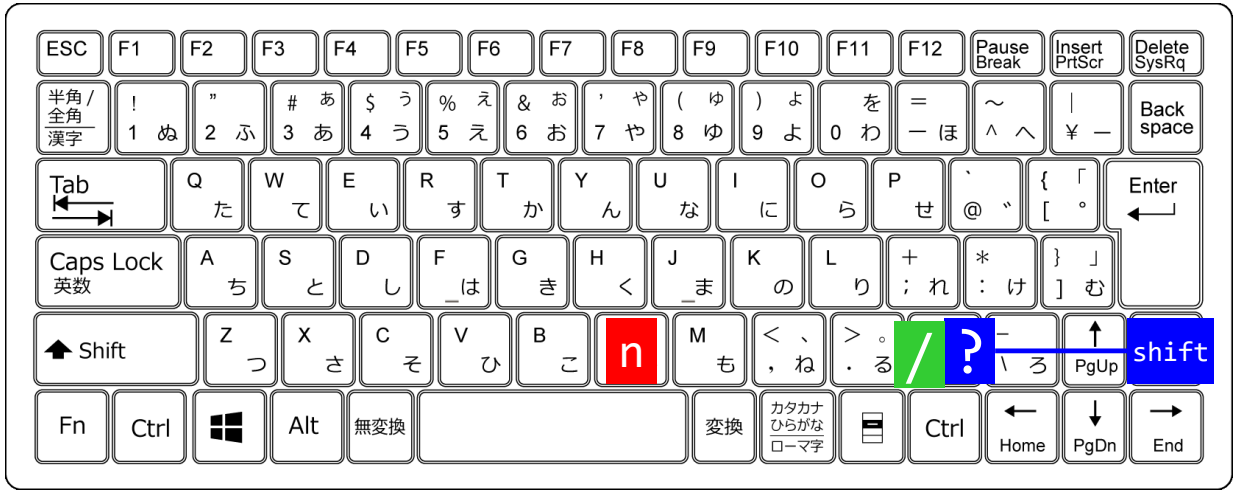
前方検索：

‘/’ の後に検索する文字列を入力し，Enter

後方検索：

‘?’ の後に検索する文字列を入力し，Enter

‘n’を入力すると直前の検索を繰り返す



■ コピー&ペースト

Y ：現在行をコピーバッファに入れる

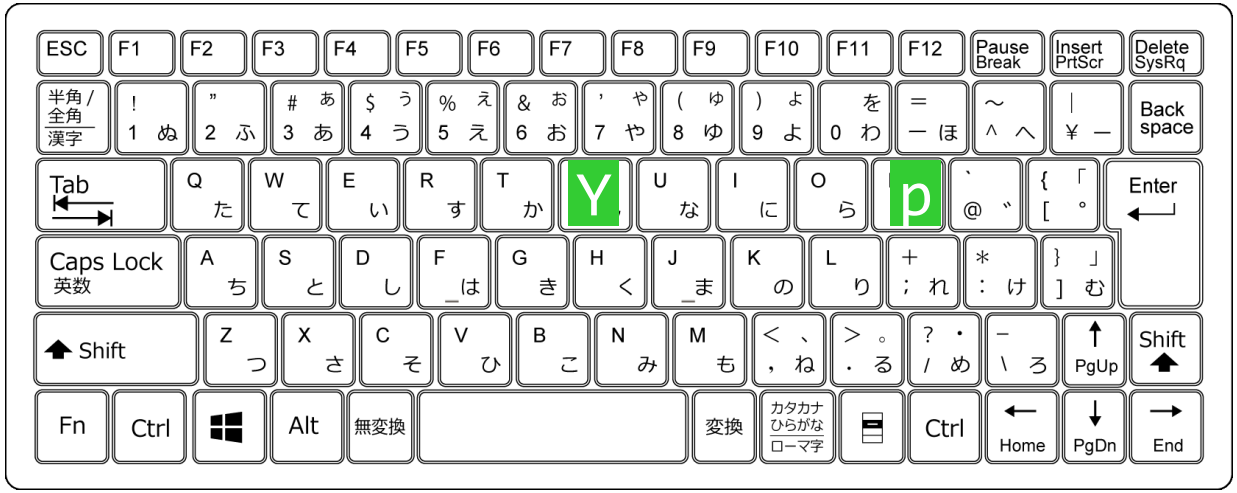
数字を指定した場合は，指定した行数分コピーバッファに入る

p ：コピーバッファの内容をペーストする

文字単位やワード単位では現在位置の後ろに，
行単位の場合は現在行の後ろにそれぞれペーストされる

P(大文字)：ペースト位置が現在位置あるいは，現在行になる

ペースト自体はpと同様である．



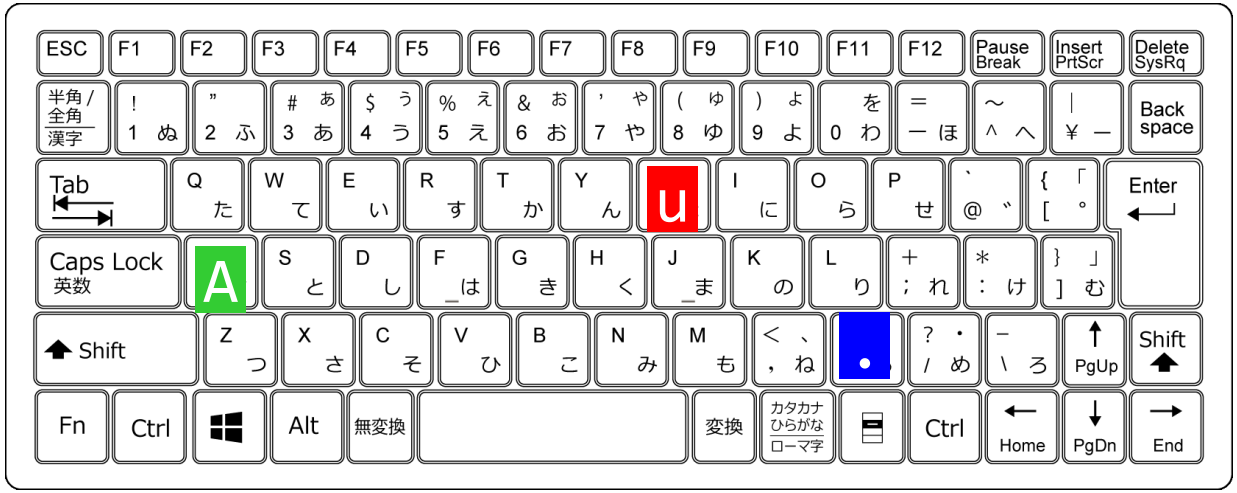
■ 便利な機能

A ：現在行の行末に移動し，挿入モードに移行する

. ：直前に実行した viモードのコマンドを繰り返す

u ：直前に行った viコマンドで行われた変更を元に戻す

vim は複数の変更を戻すことができる



コマンドラインモードの操作

ファイル操作

- ファイルの書込「'w' と入力」
  - ✓ 範囲指定(後述)してから書き込みもできる「'w'の前に範囲指定」
- 名前を指定して書込「'w' の後に空白を挟んでファイル名を入力」
- ファイルの読込「'r'のあとに空白を挟んでファイル名を入力」

終了方法

- viを終了「'q' と入力」
- 保存と同時に終了「'wq' と入力」
- 強制終了「'q!' と入力」

■ 範囲指定

- 範囲を指定する必要があるコマンドの前に入る
- 行単位で先頭行，終了行の形式で指定
  - ✓ 特殊な行として，現在行'.'，最終行'$'を示す
  - ✓ '1,.'とすると，先頭行から現在行を指定
  - ✓ '.,$'とすると，現在行から最終行を指定
- 行番号のあとに，'+1'や'-5'を付けれる
  - ✓ '.,.+10'ならば，現在行から現在行+10行を指定

■ 文字列置換

- 範囲指定の後に's/対象文字列/置換後の文字列/g'と入力
  - ✓ '1,$s/int/double/g'
- 正規表現で置換できる
- 文字列には行頭を表す'^'や行末を表す'$'が使える
- 'g'を指定しないと，同一行に複数の候補がある場合に先頭のもののみ置き換えられる

ファイル編集の流れ

- prog.cファイルを編集する
  - [端末]`vi prog.c` viでprog.cを開く(新規作成)
  - [vi] `i` 挿入モードに入る
  - [vi] プログラムを書く
  - [vi] `:w` 保存する（コマンドラインモードで`w`）
  - [vi] `:q` 終了する（`:wq`保存して終了）
  - [端末]`ls` ファイルを確認できる

- ファイルの中身を確認する
  - `cat`…ファイルの中身を表示するコマンド
  - `cat prog.c` prog.cを表示

```
$ cat prog.c
#include <stdio.h>

#define DEFAULT_LOOP 3 // The number of loops when the entered loop
range is incorrect
#define MAX_LOOP 5 // The maximum number of loops

int main(void)
{
        int loopnum;
        int loop;
```