

## 5章「ファイル編集/版数管理/印刷」

中島康彦

### §5. 0 今日の作業ディレクトリを作る

1. % `cd` ⇒ ホームディレクトリへ移動
2. % `mkdir chap05` ⇒ ディレクトリchap05を作成
3. % `cd chap05` ⇒ ディレクトリchap05へ移動
4. % `pwd` ⇒ 作業ディレクトリの確認

そろそろ、ファイル編集ができるようになろう。  
ネットワーク経由では、日頃使い慣れた道具は使えません。

## §5.1 ファイル編集(vi)

- ▶ vi ⇒ 少ないキー操作で複雑な編集が可能。  
漢字入力には適さない。

1. `% vi test` ⇒ コマンドモードになっている  
■  
~
2. `iabc[Enter]def[Esc]` ⇒ iで入力モード, [Esc]でコマンドモード  
abc  
def■  
~
3. `kdd` ⇒ 上に移動し1行削除  
def  
~
4. `p` ⇒ 削除した行を下に移動  
def  
abc  
~
5. `ZZ` ⇒ ファイルtestに書き込んで終了

## §5.1 ファイル編集(vi続き)

### 位置移動

<code>0, ^, \$</code>	⇒ 行頭, 行頭の文字, 行末へ移動
<code>H, M, L</code>	⇒ 画面最上行, 中央行, 最下行へ移動
<code>←, h (8←, 8h)</code>	⇒ 1文字戻る (8文字戻る)
<code>→, l (8→, 8l)</code>	⇒ 1文字進む (8文字進む)
<code>↑, k (8↑, 8k)</code>	⇒ 1行戻る (8行戻る)
<code>↓, j (8↓, 8j)</code>	⇒ 1行進む (8行進む)
<code>G (8G)</code>	⇒ 最終行へ移動 (8行目へ移動)
<code>b, B (8b, 8B)</code>	⇒ 単語の先頭へ戻る (8単語戻る)
<code>w, W (8w, 8W)</code>	⇒ 単語の先頭へ進む (8単語進む)
<code>e, E (8e, 8E)</code>	⇒ 単語の末尾へ進む (8単語進む)
<code>fc, Fc (8fc, 8Fc)</code>	⇒ 文字cまで進む, 戻る (8個進む, 戻る)
<code>tc, Tc (8tc, 8Tc)</code>	⇒ 文字cの手前まで進む, 戻る (8個進む, 戻る)

### 画面表示

<code>[Ctrl]+g</code>	⇒ 現在位置の表示
<code>[Ctrl]+b, [Ctrl]+f</code>	⇒ 1ページ戻る, 進む
<code>[Ctrl]+u, [Ctrl]+d</code>	⇒ 半ページ戻る, 進む
<code>[Ctrl]+y, [Ctrl]+e</code>	⇒ 1行戻る, 進む

### 検索

<code>/, ?</code>	⇒ 正規表現による前方, 後方検索
<code>n, N</code>	⇒ 次を検索, 前を検索

## §5. 1 ファイル編集(vi続き)

入力モードに遷移(終了は [Esc])

i, I (8i, 8I) ⇒ カーソルの左, 行頭から (8回コピー)  
a, A (8a, 8A) ⇒ カーソルの右, 行末から (8回コピー)  
o, O (8o, 8O) ⇒ カーソルの下, 上に行を追加 (8回コピー)  
r, R (8r, 8R) ⇒ 1文字上書き, [Esc] 押すまで上書き (8回コピー)  
s, S (8s, 8S) ⇒ 1文字削除, 1行削除 (8文字削除, 8行削除)  
c<移動>, C ⇒ 移動先, 行末まで削除  
cc (8cc) ⇒ 1行削除 (8行削除)

削除のみ

d<移動>, D ⇒ 移動先, 行末まで削除  
dd (8dd) ⇒ 1行削除 (8行削除)  
J (8J) ⇒ 行末の改行を削除 (8文字削除)  
x, X (8x, 8X) ⇒ 現在位置, 手前の1文字を削除 (8文字削除)

複写のみ

y<移動>, Y ⇒ 移動先まで, 1行を複写  
yy (8yy) ⇒ 1行複写 (8行複写)

貼付(削除/複写したものが対象)

p, P (8p, 8P) ⇒ カーソルの右/下, 左/上に貼り付ける (8回実行)

取り消し/読み込み/終了

u, U ⇒ 最後の操作, 現在行を元に戻す  
:rファイル ⇒ ファイル読み込み  
:wファイル ⇒ ファイル書き込み  
:q! :w! ZZ ⇒ 保存せずに終了, 終了せずに保存, 保存して終了

## §5. 1 ファイル編集(vi続き)

その他

~ (8~) ⇒ 英大文字⇄英小文字変換 (8文字変換)  
:%s/文字列1/文字列2/ ⇒ sedと同様(各行の最初の文字列1を文字列2に置換)  
:%s/文字列1/文字列2/g ⇒ sedと同様(全ての文字列1を文字列2に置換)  
. ⇒ 最後に実行した編集を再度適用  
!!UNIXコマンド ⇒ UNIXコマンドの実行結果を現在行に追加  
モードがわからなくなったら, [Esc]を2~3回押すとコマンドモードに戻る

1. % vi test ⇒ コマンドモードになっている
2. dG ⇒ 最終行まで削除
3. !!ls -al ⇒ ls -alの実行結果を取り込む
4. j100~ ⇒ 100回分, 英大文字⇄英小文字変換
5. u ⇒ 元に戻す(再度uを押すと再度戻る)
6. /:[Enter] ⇒ :を探す
7. nnnn ⇒ 次候補を順に探す
8. r@ ⇒ :を@に変換
9. n.n.n.n. ⇒ 次候補を順に変換
10. :%s/@/---/g ⇒ @を---に全て変換
11. u ⇒ 元に戻す
12. ZZ ⇒ 保存して終了

## §5.2 ファイル編集(mule)

▶ mule ⇒ さらに複雑な編集が可能.  
漢字入力に適する.

1. `% mule test` ⇒ 入力モードになっている
2. `abc[Enter]def` ⇒ abcdefと入力  
`abc`  
`def`
3. `↑[Ctrl]+a[Ctrl]+k[Ctrl]+k` ⇒ 上に移動し1行削除  
`def`
4. `↓[Ctrl]+y` ⇒ 削除した行を下に移動  
`def`  
`abc`
5. `[Ctrl]+x[Ctrl]+s` ⇒ ファイルtestに書き込む
6. `[Ctrl]+x[Ctrl]+c` ⇒ 終了

## §5.2 ファイル編集(mule続き)

位置移動([Ctrl]+を^と表記)

- |   |                    |
|---|--------------------|
| <code>^a, ^e</code>                             | ⇒ 行頭, 行末へ移動        |
| <code>←, ^b</code> ( <code>^u8←, ^u8^b</code> ) | ⇒ 1文字戻る (8文字戻る)    |
| <code>→, ^f</code> ( <code>^u8→, ^u8^f</code> ) | ⇒ 1文字進む (8文字進む)    |
| <code>↑, ^p</code> ( <code>^u8↑, ^u8^p</code> ) | ⇒ 1行戻る (8行戻る)      |
| <code>↓, ^n</code> ( <code>^u8↓, ^u8^n</code> ) | ⇒ 1行進む (8行進む)      |
| <code>[Esc]&lt;, [Esc]&gt;</code>               | ⇒ 先頭行, 最終行へ移動      |
| <code>^←</code> ( <code>^u8^←</code> )          | ⇒ 単語の先頭へ戻る (8単語戻る) |
| <code>^→</code> ( <code>^u8^→</code> )          | ⇒ 単語の末尾へ進む (8単語進む) |
| <code>^↑</code> ( <code>^u8^↑</code> )          | ⇒ 段落の先頭へ戻る (8段落戻る) |
| <code>^↓</code> ( <code>^u8^↓</code> )          | ⇒ 段落の先頭へ進む (8段落進む) |

画面表示

- |                         |                   |
|-------------------------|-------------------|
| <code>[Esc]v, ^v</code> | ⇒ 1ページ戻る, 進む      |
| <code>^xl</code>        | ⇒ 現在位置の表示         |
| <code>^xo, ^xk</code>   | ⇒ 他の画面に移動, 現画面を削除 |
| <code>^x0, ^x1</code>   | ⇒ 現画面, 現画面以外を隠す   |
| <code>^x2, ^x3</code>   | ⇒ 上下, 左右に画面を分割    |

検索

- |                                   |                   |
|-----------------------------------|-------------------|
| <code>^s, ^r</code>               | ⇒ 前方, 後方検索        |
| <code>^s^s, ^r^r</code>           | ⇒ 次を検索, 前を検索      |
| <code>[Esc]^s, [Esc]^r</code>     | ⇒ 正規表現による前方, 後方検索 |
| <code>[Esc]^s^s, [Esc]^r^r</code> | ⇒ 次を検索, 前を検索      |

## §5.2 ファイル編集(mule続き)

上書きモードに遷移(終了は [Insert])

[Insert]	⇒ [Insert]押すまで上書き
フィルカラム設定	
<sup>^</sup> xf ( <sup>^</sup> u8 <sup>^</sup> xf)	⇒ フィルカラム数の設定
[Esc]q ( <sup>^</sup> u[Esc]q)	⇒ 段落をフィルカラム以下に揃える(均等割り付け)
リージョン設定	
<sup>^</sup> @	⇒ マークを付ける. カーソルとの間がリージョン
削除のみ	
[Esc]d ( <sup>^</sup> u8[Esc]d)	⇒ 1語削除 (8語削除)
<sup>^</sup> k ( <sup>^</sup> u8 <sup>^</sup> k)	⇒ 行末まで削除 (8行削除)
<sup>^</sup> d ( <sup>^</sup> u8 <sup>^</sup> d)	⇒ 1文字削除 (8文字削除)
<sup>^</sup> w	⇒ リージョンを削除
複写のみ	
[Esc]w	⇒ リージョンを複写
貼付(削除/複写したものが対象)	
<sup>^</sup> y	⇒ 貼り付ける
取り消し/読み込み/終了	
<sup>^</sup> /	⇒ 操作を元に戻す
<sup>^</sup> xi	⇒ ファイル読み込み(現バッファに追加)
<sup>^</sup> x <sup>^</sup> f	⇒ ファイル読み込み(新たなバッファを開く)
<sup>^</sup> x <sup>^</sup> w	⇒ ファイル書き込み
<sup>^</sup> x <sup>^</sup> s, <sup>^</sup> x <sup>^</sup> c	⇒ 終了せずに保存, 終了

---

## §5.2 ファイル編集(mule続き)

その他

[Esc]c, [Esc]u, [Esc]l	⇒ 1語を英大文字, 英小文字に変換
[Esc]%	⇒ 文字列を対話的に置換
[Esc]!UNIXコマンド	⇒ UNIXコマンドの実行結果を新バッファに追加
モードがわからなくなったら, <sup>^</sup> gを2~3回押すと入力モードに戻る	
画面が崩れたら, <sup>^</sup> lを押す	

1. % mule test ⇒ 入力モードになっている
2. <sup>^</sup>@[Esc]><sup>^</sup>w ⇒ 先頭にマークし最終行に移動し削除
3. [Esc]!ls -al ⇒ ls -alの実行結果を取り込む
- <sup>^</sup>xo ⇒ 実行結果バッファに移動
4. <sup>^</sup>u10[Esc]u ⇒ 10語分, 英小文字⇒英大文字変換
5. <sup>^</sup>/<sup>^</sup>/<sup>^</sup>/<sup>^</sup>/ ⇒ 元に戻す(一順すると元に戻る)
6. <sup>^</sup>s: ⇒ :を探す
7. <sup>^</sup>s<sup>^</sup>s<sup>^</sup>s<sup>^</sup>s ⇒ 次候補を順に探す
8. [Esc]<[Esc]% : @ ⇒ 先頭から:を@に変換
9. [Space][Space][Space] ⇒ 次候補を順に変換
10. [Esc]<[Esc]% @ --- ! ⇒ @を---に全て変換
11. <sup>^</sup>/<sup>^</sup>/<sup>^</sup>/<sup>^</sup>/ ⇒ 元に戻す
12. <sup>^</sup>x<sup>^</sup>s test <sup>^</sup>x<sup>^</sup>c ⇒ testに保存して終了

## §5.2 ファイル編集(mule続き)

### 罫線

[Esc] [Esc]K	⇒ 罫線モードをON/OFF
[F8]	⇒ 罫線の太さを変更
[F9], [F12]	⇒ 左,右方向に罫線を引く
[F10], [F11]	⇒ 下,上方向に罫線を引く
[Shift]+[F9], [Shift]+[F12]	⇒ 横方向に罫線を縮小,拡大
[Shift]+[F10], [Shift]+[F11]	⇒ 縦方向に罫線を拡大,縮小

詳細は,

`less /usr/share/emacs/19.34/site-lisp/keisen-mule.doc`

muleは単なるエディタではなく、様々な機能を組み込むことができる。  
遠隔のコンピュータ上でメールを送受信することも必要。

---

## §5.3 ファイル編集(その他)

- ▶ `ispell` ⇒ 英文スペルチェッカ
- ▶ `scp, sftp` ⇒ ファイル転送 (`ssh`が使える場合)
- ▶ `ftp` ⇒ ファイル転送 (`ftp`が使える場合)

UNIX間のファイル転送

Windows上のファイル⇔UNIX上のファイル

一般に、`nkf`による日本語コードの変換が必要

1. Windowsのエディタを使いファイル `nt.txt` を作成
  2. スタート → プログラム → MS-DOSプロンプト
  3. `C:\>cd nt.txtのあるWindowsディレクトリ`
  4. `C:\>ftp UNIXホスト名`
  5. Name: 利用コード
  6. Password: パスワード
  7. `ftp> cd ファイルを置きたいUNIXディレクトリ`
  8. `ftp> put nt.txt unix.txt (Windows⇒UNIX)`  
`ftp> get unix.txt nt.txt (UNIX⇒Windows)`
  9. `ftp> quit`
-

## §5. 4 ファイル版数管理(RCS)

---

- ▶ `ci -l` ファイル名  
チェックイン(現ファイルをRCSに登録)
- ▶ `co -l` ファイル名
- ▶ `co -r`版数1 ファイル名  
チェックアウト(RCSの最新版を取得)  
チェックアウト(RCSの版数1を取得)
- ▶ `rcsdiff` ファイル名
- ▶ `rcsdiff -r`版数1 ファイル名
- ▶ `rcsdiff -r`版数1 `-r`版数2 ファイル名  
RCSの最新版と現ファイルの差分  
RCSの版数1と現ファイルの差分  
RCSの版数1と版数2の差分
- ▶ `rlog` ファイル  
RCSの更新履歴を表示

---

## §5. 4 ファイル版数管理(RCS続き)

1. `% mkdir RCS` ⇒ chap05の中にディレクトリRCSを作成
2. `% vi test` ⇒ viを起動
3. `dG` ⇒ 最終行まで削除
4. `i$Header$[Enter]abc[Esc]ZZ` ⇒ ファイルを作成/保存/終了  
`$Header$`  
`abc`
5. `% ci -l test` ⇒ RCSにチェックインする(RCSの下にtest,vができる)  
`>> KYOTO`  
`>> .`  
`initial revision: 1.1`  
`done`
6. `% cat test` ⇒ testを表示(\$Header\$の部分が自動生成されている)  
`$Header: test,v 1.1 2000/02/21 07:10:01 xxxx Exp xxxx $`  
`abc`
7. `% rm test` ⇒ testを削除
8. `% co -l test` ⇒ RCSからtestの最新版をチェックアウトする  
`RCS/test,v --> test`  
`revision 1.1 (locked)`  
`done`
9. `% cat test` ⇒ testを表示(元通り)  
`$Header: test,v 1.1 2000/02/21 07:10:01 xxxx Exp xxxx $`  
`abc`

## §5.4 ファイル版数管理(RCS続き)

10. `% vi test` ⇒ viを起動
11. `Go` ⇒ 最終行の後ろに追加
12. `def[Esc]ZZ` ⇒ ファイルを更新/保存/終了  
`$Header: test,v 1.1 2000/02/21 07:10:01 xxxx Exp xxxx $`  
`abc`  
`def`
13. `% rcsdiff test` ⇒ RCSの最新版とtestの差分  
`retrieving revision 1.1`  
`diff -r1.1 test`  
`2a3`  
`> def`
14. `% ci -l test` ⇒ RCSにチェックインする(RCSの下にtest,vができる)  
`>> new line is added`  
`>> .`  
`done`
15. `% rlog test` ⇒ testを表示(更新履歴を表示)  
`description:`  
`KYOTO`  
-----  
`revision 1.2 locked by: xxxx;`  
`date: 2000/02/21 11:24:53; author: xxxx; state: Exp; lines: +2 -1`  
`new line is added`  
-----  
`revision 1.1`  
`date: 2000/02/21 07:10:01; author: xxxx; state: Exp;`  
`Initial revision`  
=====

---

## §5.4 ファイル版数管理(RCS続き)

16. `% co -r1.1 test` ⇒ Version1.1をチェックアウトする  
`RCS/test,v --> test`  
`revision 1.1`  
`writable test exists; remove it? [ny] (n): y`  
`done`
17. `% cat test` ⇒ Version1.1が入っている  
`$Header: test,v 1.1 2000/02/21 07:10:01 xxxx Exp xxxx $`  
`abc`
18. `% rcsdiff -r1.1 -r1.2 test` ⇒ 1.1と1.2の差分を表示  
`retrieving revision 1.1`  
`retrieving revision 1.2`  
`diff -r1.1 -r1.2`  
`1c1`  
`< $Header: test,v 1.1 2000/02/21 07:10:01 xxxx Exp $`  
---  
`> $Header: test,v 1.2 2000/02/21 11:24:53 xxxx Exp $`  
`2a3`  
`> def`
19. `% co -l test` ⇒ RCSからtestの最新版をチェックアウトする  
`revision 1.2 (locked)`  
`done`
-



## §5. 5 ファイル印刷

---

UNIXには印刷用のコマンドが豊富にある。

Postscriptと呼ばれる表示/印刷用言語がよく用いられる。(詳細は省略)

```
pr      ⇒ テキストファイルを印刷用に整形
a2ps    ⇒ テキストファイルをPostscriptに変換
psnup   ⇒ Postscript複数ページを縮小して1枚に納める
psresize⇒ Postscriptファイルのページサイズ変更
psselect⇒ Postscriptファイルから指定ページの取り出し
ps2pdf  ⇒ PostscriptファイルからPDFファイルを生成
lpr     ⇒ Postscriptファイルをプリンタへ送信
lp      ⇒ "
```

Windowsでの閲覧には、最近普及したPDF形式が便利。ps2pdfやOpenofficeを使えば無償でPDFが作成できる。

UNIXでの閲覧には、acroreadを使用。

---

## §5. 6 例題

---

講義中に作成したファイル `test` の変更履歴を `rlog` を使って出力せよ。

## §5.7 今日の課題

テキストファイルを作成し、PDF形式に変換してみよ。なお、`acroread`により内容が正常に表示されることを確認すること。

- ▶ 1. `a2ps`を使って、テキストファイルからPSを生成する。
- ▶ 2. `gv`を使って、PSの内容を確認することができる。
- ▶ 3. `ps2pdf`を使って、PSからPDFを生成する。
- ▶ 4. `acroread`を使って、PDFの内容を確認する。

`ls -l` を用いて、3つのファイルの大きさを比較してみよ。なお、PSやPDFをレポートに添付する必要はない。

宛先: `nakashim@econ.kyoto-u.ac.jp`

件名: `unix1-学生番号`

-----  
名前(忘れずに)

今日はここまで