

# GNU Emacs (Mule) 入門

電子工学科 桑田正行

1994年1月14日

## 要旨

GNU Emacs(以下 Emacs)は、優れたテキストエディタであるだけでなく、作業環境です。MuleはGNU Emacsを多国語対応に強化したものです。

## 1 Emacs の基本概念

### 1.1 画面を構成する部品

Emacsでは画面をいくつかの領域に分けて、それぞれに独自の情報を表示します。画面上のそれぞれの領域には、次のような名前がついています。

- ウィンドウ** テキスト(ファイルの内容など)を表示する部分
- モードライン** ウィンドウに表示中のバッファの情報を表示する行(通常、反転表示)
- エコー領域** メッセージなどを表示する一番下の行
- ミニバッファ** エコー領域に現れてコマンドや引数の入力に使用されるウィンドウ。

テキスト表示部とモードラインは対になって1つのウィンドウを形成します。この単位で画面を分割して複数のウィンドウを表示できます。エコー領域は画面全体で1つです。

Emacsでは、ファイルの内容を**バッファ**(内容を保存しておくために用意されるファイル名と同じ名前の領域)に読み込んで、そのバッファ上で編集作業を行ないます。ウィンドウにはこのバッファの内容が表示されています。



図 1: Emacs の画面構成

### 1.2 マルチプルバッファとマルチプルウィンドウ

Emacsでは、バッファを複数もつことができます。複数のファイルを別々のバッファに読み込むことによって、同時に編集することができます。また、Emacsでは、画面の表示を複数のウィンドウに分けることができます。分けられたウィンドウには異なるバッファの一部分、もしくは1つのバッファの異なる部分が表示できます。現在画面に表示されていて、カーソルが表示され、入力が可能なバッファをカレントバッファといいます。

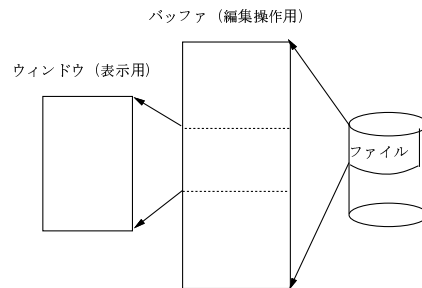


図 2: ウィンドウ、バッファおよびファイルの関係

### 1.3 モードライン

モードラインには、次のようなバッファのいろいろな情報が記述されています。

```
[--]J:--**Mule: *scratch*          (Lisp Interaction)--All-----
  ↑ ↑ ↑          ↑          ↑          ↑
  (a) (b) (c)      (d)      (e)      (f)
```

(a) Mule での多国語入力時のモードを表示します。

例えば、Mule に附属している egg(日本語入力ツール) が使用します。C-\ を入力すると表示が [あ] に変わり、文字を入力すると「ローマ字かな変換」され、その状態で SPC を入力すると「かな漢字変換」されます。RET を入力すると確定します。確定するまでは、文字列の両側は | で囲まれたフェンスモードになっています。フェンスモードのとき C-\ を入力すると [aA] といった表示になり、alphabet をそのまま入力できます。

(b) Mule でのバッファのコード系ニーモニック表します。

Mule のバッファには、ASCII 文字、その他の 1 バイト文字、漢字のような 2 バイト文字が共存できます。ASCII の 1 バイト文字のことを alphanumeric と呼びます。

Mule は、ファイル入出力、入力、画面出力、プロセス間通信について、それぞれ独立にコード系を指定させることができますが、通常はファイル用のコード系のニーモニックだけ表示しています。最初の一文字がコード系のニーモニック、次の ‘:’ が現在英語以外の文字(日本語、中国語など)もきちんと表示するというを示しています。

ニーモニック	意 味
=	無変換
+	自動変換
S	MS 漢字コード (シフト JIS )
s	MS 漢字コード (シフト JIS ), 行末は CRLF
J	JUNET で使われているコード (およびそれに類するもの)
X	X で使われているコンパウンドテキストのコード
E	日本語 EUC コード
K	韓国語 EUC コード
k	韓国のネットワークで使用されているコード
C	中国語 EUC コード
I	96 文字セットの呼び出しに SS2 を用いる ISO2022 コード系
i	96 文字セットの呼び出しにロッキングシフトを用いる ISO2022 コード系
B	BIG5 コード
b	BIG5 コード, 行末は CRLF

また、変数 ‘mc-verbose-code’ に ‘nil’ 以外の値をセットすると、モードラインにファイル用、キー入力用、画面表示用の三つのコード系のニーモニックを表示するようになります。

(c) バッファの状態を示します。

表示	意味
--	バッファの内容は変更されていない.
**	バッファの内容が変更され、保存されていない.
%%	バッファが書き込み禁止.

(d) バッファの名前を表します.

Emacs を引数なしで起動した場合、**\*scratch\*** という名前のバッファが作成されます.

通常、バッファ名は読み込んだファイル名になります.

(e) モード名を表します.

モード名は、バッファの表示の仕方や入力されたキーの解釈方法を決定するモードを示します.

各バッファは常に1つのモードにあります. Emacs では、読み込んだファイルの拡張子からふさわしいモードを選択したり、選択するモード名をファイルの中で陽に指定する機能があります.

(f) 現在表示されているテキストが、全体のどのくらいの位置にあるかを示します (数字による % 表示, All: テキスト全体, Top: テキストの先頭, Bot: テキストの一番最後の部分).

#### 1.4 Emacs の入力モード

Emacs はキャラクタエディタであり、入力する文字はできるだけテキストとして扱う方がわかりやすいので、基本モードはテキスト入力モードとし、コマンドの入力は特定の手順 (通常、コントロールキーまたはメタキーと ASCII 文字 1 文字の組合せ) に従って行なう方式を取ります. コマンドは入力とその実行を終了すると、自動的にテキスト入力モードに切り替わります.

#### 1.5 Emacs で取り扱う文字

Emacs で扱う文字セットは、ASCII コード (7 ビット) を基本としています. この中で、通常表示可能な文字以外は、次に示すようなコントロール文字として扱います.

**C-x (Control-x)**      Control キーと文字 **x** とを同時に入力.

また、Emacs では、7 ビット ASCII コードを 8 ビットに拡張して 1 ビットを付加し、256 種類のコマンドを扱えるようにしています. この拡張部分は次に示すようにメタキーを用いて入力します.

**M-x (Meta-x)**      メタキーと文字 **x** を続けて入力.

メタキーのない端末では、通常 ESC キーを用い、ESC キーを入力した後で次の文字を入力します.

各文字の表示は次のように行なわれます.

ASCII 表示可能文字	対応する表示可能文字
C-x	^x
M-x	M-x
これ以外の文字	\8 進数 (例えば \234)

一方 Mule では、ASCII 文字セット (7 ビット) や ISO Latin-1 の文字セット (8 ビット) ばかりでなく、日本語、中国語、韓国語などの文字セット (16 ビット) やタイ語も扱うことができます。さらに Mule のテキストバッファはこれらの言語の文字を一つのバッファに混在して保持することができます。

これらの文字は、編集の対象となるテキストとして入力するか、あるいは各種のコマンドとして入力します。

## 1.6 キーとコマンド

Emacs では、システムが基本的にもっている機能の他に、いくつものコマンドを Lisp 言語で記述して作るにより拡張できます。

コマンドは、ファイルに対するもの、バッファに対するもの、ウィンドウに対するもの、またこれらの処理を組み合わせた処理を考えることができます。

あらゆるコマンドは Lisp 言語で記述された関数で、基本的にはユーザがこの関数を指定し、起動させることにより実行します。この関数の指定をあるキーに結び付けておくことにより、起動させることもできます。これが関数のキーバインディングです。このキーバインディングされている関数は、そのキーを入力することにより、起動されます。

関数とキーとの関係 (キーマップ) を変更する機能も備わっていて、ユーザのカスタマイズ機能の 1 つです。

## 2 参考文献

この文書を書くにあたって、参考にしたり引用した書籍を以下に挙げておきます。

- [1] 矢吹道郎監修, 宮城史朗著: 「初めて使う GNU Emacs」 (啓学出版 1992).
- [2] 斉藤信男編: 「ユーザーズ UNIX」 (岩波書店 1988).
- [3] Richard Stallman 著, 竹内郁雄・天海良治監訳: 「GNU Emacs マニュアル」 bit 別冊 (共立出版 1988).
- [4] 「Mule に附属のドキュメント」

### 3 Emacs のコマンド

#### 3.1 記法について

C-x コントロールキー (Ctrl) を押しながらか x を押すこと

ESC エスケープキー (Esc) を押すこと

DEL デリートキー (Del) を押すこと

SPC スペース・バーを押すこと

M-x ESC に続いて x を押すこと

#### 3.2 Emacs の起動と終了

##### (1) 起 動

シェルのコマンドラインから次のようにして起動します。

```
% emacs
```

または

```
% mule
```

引数を与えて起動することもできます。詳細はマニュアルを参照して下さい。

##### (2) 終 了

C-x C-c バッファの退避等をユーザに問い合わせしてから Emacs を終了

C-z Emacs を中断し、親プロセスのシェルへ制御を戻す。fg コマンドにより、中断されていた Emacs が再開。

#### 3.3 Emacs の基本コマンド

Emacs では、画面上のカーソル位置のことを **ポイント** といい、編集コマンドはポイントの指している位置に対して適用することを原則とします。また、ポイントと対で使用するために、画面上のテキストの任意の場所に **マーク** をつけることができます。ポイントとマークとの間にある部分を **リージョン** と呼びます。

以下、主なコマンドを示します。

#### (1) 基本編集コマンド

##### ●テキストの入力と挿入

Emacs の基本モードはテキスト挿入モードで、入力した印刷可能な文字 (Mule の場合は多国語も含む) はカーソルの位置に挿入されます。通常はカーソルとそれより後ろにある文字は右に移動します (“C-x C-k r RET” を実行すると異なった動きをします。元の状態にする場合は、再度 “C-x C-k r RET” を実行します)。コントロール文字またはメタ文字をテキストデータとして入力するときは、次のコマンドを用います。

C-q 1 文字 1 文字 (コントロール文字も可) を挿入

C-q 8 進 3 桁 対応する値をもつ文字を挿入

##### ●テキストの削除・コピー・移動

DEL カーソルの前の 1 文字削除

C-d カーソルの指している 1 文字削除

C-k カーソル位置から行末までを削除し削除リングへ移動

C-w リージョンを削除し削除リングへ移動

C-y 現在カーソル位置に削除リングからコピーして挿入

M-w リージョンを削除リングにコピー

**テキストの移動**は、削除して、カーソルを移動後、挿入することにより行なえます。**削除リング**は、カットアンドペーストのために使用される特別なバッファです。

##### ●カーソルの移動

C-f 1 文字前方に移動

C-b 1 文字後方に移動

C-a 行の先頭に移動

C-e 行の末尾に移動

C-n 1 行下に移動

C-p 1 行上に移動

C-v 1画面分下方に移動  
M-v 1画面分上方に移動  
M-< バッファの先頭に移動  
M-> バッファの末尾に移動

## ●テキストの探索・置換

### (a) インクリメンタル探索

C-s 順方向へインクリメンタル探索  
C-r 逆方向へインクリメンタル探索

ユーザが探索パターンを入力し始めると、部分的に一致している文字列を指定した方向に探索します。

### インクリメンタル探索で特別な意味をもつ文字

DEL インクリメンタルの文字列を消去  
ESC インクリメンタルを終了  
C-q インクリメンタルにおいて文字の特別な意味を取る  
C-s 順方向のインクリメンタル探索を繰り返す  
C-r 逆方向のインクリメンタル探索を繰り返す  
C-y バッファの行の残りの部分を探索文字列へ取り入れる  
C-w バッファの次の単語を探索文字列へ取り入れる

### (b) 利用者に質問しながらの置換

M-% old-str RET new-str RET  
対話的に old-str を new-str に置換

### 置換で特別な意味を持つ文字

SPC old-str を new-str で置換  
DEL 置換せずに先へ進む  
ESC これ以上置換せずに終了  
.(ピリオド) 現在の文字列を置換してから終了  
! これからは質問することなしに、残りの文字列をすべて置換

^ 誤って置換したときに前の位置へ戻る

## ●マークの設定とリージョンの定義

C-SPC ポイント (現在のカーソル位置) にマークを設定  
C-@ 同上  
C-x C-x マークとポイントの位置を交換  
M-h 現在の段落をリージョンにする  
C-x h 全バッファをリージョンにする

コマンドによってはリージョンを指定してから使うものがあります。マークの部分からポイント (現在のカーソル位置) までがリージョンとなります。コマンド C-SPC によってポイントにマークが設定されます (見た目は何の変化もなく、エコー領域に **Mark set** と表示されるだけです)。それから設定したいリージョンの他方にカーソルを移動し、リージョンを使うコマンドを入力します。

## ●とりやめと取り消し

Emacs のバッファのテキストの変更はすべて取り消すことができます。これは約 8000 文字まで可能です。それぞれのバッファは変更を個別に記録しています。変更取り消しコマンドは、常にカレントバッファに適用されます。

C-g 実行の取りやめあるいは途中まで入力したコマンドのとりやめ  
C-x u 望ましくない 1 回分の変更を元に戻す  
C-\_ 同上

## (2) バッファとファイルの操作コマンド

### ●バッファ操作

C-x k バッファを削除  
C-x C-q バッファを読み出し専用にしたり、書き込み可能にする  
C-x C-b バッファのリストを表示

C-x b 指定した名前のバッファを選択するか生成し、ウィンドウに表示

C-x 4 b 指定した名前のバッファを選択するか生成し、他のウィンドウに表示

### ●ファイルの読み込み・書き込み

C-x C-f 指定したファイルのバッファへの読み込み(ディレクトリを指定した場合は、ディレクトリエディタ Dired を起動)

C-x 4 C-f 指定したファイルのバッファへの読み込み、別のウィンドウへ表示

C-x i カーソル位置にファイルを読み込み、挿入

C-x C-s カレントバッファの内容を読み込んだファイルに書き込む

C-x C-w カレントバッファの内容を指定したファイルに書き込む

M-~ カレントバッファの修正の印を消去

C-x d ディレクトリエディタ Dired (4.2節参照) の起動

### (3) 画面とウィンドウの操作コマンド

#### ●表示の制御

C-v ウィンドウ内を前方向にスクロール

M-v ウィンドウ内を後方向にスクロール

C-l 画面全体を再表示

#### ●ウィンドウ操作

C-x 2 カーソルのあるウィンドウを上下2つに分割

C-x 5 カーソルのあるウィンドウを左右2つに分割

C-x 0 カーソルのあるウィンドウを消去

C-x 1 カーソルのあるウィンドウを残して他のウィンドウを消去

C-x o 他のウィンドウに移動

C-M-v 次のウィンドウをスクロール

### (4) ヘルプ機能

C-h a キーワードとなる単語を入れると、キーワードを含むコマンド名を表示

C-h b キーバインド(どのキーにどのコマンドが結びついているかということ)を表示

C-h i Info というドキュメントリーダー(4.3節参照)を起動(終了はq)

C-h k 指定したキー名で実行されるコマンド名を表示

C-h m カレントバッファの主モードの説明を表示

C-h t Emacs の入門編(英語)を表示(終了はC-x k)

C-h T Mule の入門編(日本語)を表示(終了はC-x k)

C-h はバックスペースキー(Bs)によっても出力されるので、間違えないようにしましょう。(間違った場合は C-g でとりやめ)

### (5) 入力補完機能

ミニバッファでは入力の補完機能が働きます。補完機能というのは、入力した文字から確定できる部分がある場合、自動的にその部分を補ってくれる機能です。補完を行なおうとして候補が複数ある場合は、その候補が表示されます。

TAB ミニバッファのテキストを可能な限り補完

SPC 1語分だけ補完(-を越えては行なわない)

RET 補完を行なった後、ミニバッファから脱出

? ミニバッファのテキストから補完可能な候補を表示

## 4 作業環境としての Emacs

Emacs は優れたフルスクリーンエディタですが、さらにその中から、UNIX のいろいろな機能を利用することができます。極端な話、Emacs から 1 歩も出ないで、UNIX のほとんどの機能を使用できます。また、バッファに固有なモードの機能を利用していろいろなアプリケーションのインタフェースを構築できます。

### 4.1 シェルコマンドの実行

Emacs では、UNIX のシェルコマンドを Emacs の中から実行できます。

#### (1) Emacs からのシェルコマンドの実行

- M-! 指定されたシェルコマンドを実行し、その出力を\*Shell Command Output\* バッファ (別のウィンドウ) に表示
- M-| 指定されたシェルコマンドをリージョンの内容を入力として実行。また、数引数を与えるとそのリージョンをコマンドの出力で置換可能。
- M-x shell サブシェルを実行する。このとき入出力は Emacs の \*shell\* バッファ (Shell モード) を介して行う。コマンドの入力は対話的に行える。

#### (2) Shell モードにおける特別なキー割り当て

- RET バッファの最後にいるときは、その行を入力として送る。そうでなければ現在行をバッファの最後にコピーしてからそれを送る。このとき、変数 shell-prompt-pattern にマッチする行頭のテキストはコピーされない。
- C-c C-d end-of-file を入力として送る。通常はシェルあるいはサブシェルを終らせる。
- C-c C-c シェルまたはそのサブジョブにインタラプトをかける。
- C-c C-z シェルまたはそのサブジョブを stop

- C-c C-u 入力として送ろうとしているテキストをすべて削除する。
- C-c C-w ポイントの前の単語を削除する。
- C-c C-o シェルからの最後の 1 回分の出力を削除する。
- C-c C-r 最後の 1 回分の出力をウィンドウの最上部へスクロールする。
- C-c C-y 直前のシェルへの入力をコピーし、バッファのポイントの前に挿入する。最後の改行は挿入されず、RET が入力されるまでコピーされた入力を送らない。

## 4.2 ディレクトリエディタ Dired

ファイルを操作するために便利な機能が、このディレクトリエディタ Dired です。バッファ名はディレクトリ名、モード名は Dired です。そのディレクトリにあるファイルのリストの表示、削除、コピーをはじめ、ほとんどすべてのファイル操作をすることができます。

### (1) Dired の起動

C-x d Dired を起動

C-x 4 d 別のウィンドウで Dired を起動

ファイルを読み込むコマンド C-x C-f の引数としてディレクトリを与えた場合にも Dired は起動されます。

### (2) Dired での編集

Emacs の通常のカーソル移動コマンドは、すべて使用できます。それ以外に、特別の目的に使えるコマンドがあります。

SPC カーソルを次の行のファイル名の先頭に移動 (n, C-n も同様)

C-p カーソルを前の行のファイル名の先頭に移動 (p も同様)

DEL カーソルを前の行へ移動し、その行の消去の印を消去

### (3) Dired によるファイルの消去

d ファイルに消去の印を付加

u ポイントのある行の消去の印を取る

DEL カーソルを前の行へ移動し、その行の消去の印を消去

x 消去の印のついたファイルを実際に削除

# 自動セーブファイル (ファイル名の前後に '#' があるもの) すべてに消去の印を付加

~ バックアップファイルすべてに消去の印を付加

### (4) Dired ですぐに実行されるコマンド

c 現在行のファイルをコピー

f 現在行のファイルを読み込む

o f と同様だが、ファイルのバッファの表示に別のウィンドウを使用

h Dired モードの説明を表示

r 現在行のファイルの名前を変更

v M-x view-file を使って現在行のファイルを閲覧

C 現在行のファイルを compress

U 現在行の compress ファイルを uncompress

M 現在行のファイルの保護モードを変更

G 現在行のファイルのグループ id を変更

O 現在行のファイルの持ち主 id を変更

### (5) Dired の拡張機能

一部のホストでは、Dired の拡張した機能を利用できます。

k 現在の Dired バッファを削除

m サブディレクトリを新規に作成

z 現在行のファイルを gzip を用いて compress

D 現在行のファイルとのもっとも古いバックアップファイルとの差分を表示 (ディレクトリ同士の比較も可能)

R 現在行のディレクトリファイルを削除

T 現在行のファイルの tar ファイルを作成

X 現在行の tar ファイルからファイルを抽出

L 現在行の tar ファイルの内容を表示

Z 現在行の tar + compress ファイルの内容を表示

C-d 現在行のファイルを削除

M-u 現在行の uuencode ファイルを udecode

M-z 現在行の tar + compress ファイルからファイルを抽出

### 4.3 ドキュメントリーダ Info

Emacs には、階層化されたドキュメントファイルを見るためのプログラム Info があります。

#### (1) Info の起動

C-h i で Info プログラムを起動します。

バッファのモードは Info になっています。ここで “h” を入力すると Info の使い方に関する説明が得られます。

#### (2) Info バッファでのコマンド

Info バッファの最初の行には、  
File: 現在のファイル名  
Node: 現在のノード名  
Next: 次のノード名 ( n コマンドで見れます)  
Previous: 前のノード名 ( p コマンドで見れます)  
Up: 上のノード名 ( u コマンドで見れます)

のような情報が表示されています。

行頭にある “\*” は、メニュー項目を表します。

#### (a) メニューの選択

- メニュー項目の上にカーソルを持っていて、m RET でそのメニューの項目を見れます。
- また、“m” に続けて「メニュー項目名」を入力すれば、そのメニュー項目を見れます。

#### (b) スクロール

- `[SPC]` キーでウィンドウを上方向にスクロールします。
- `[DEL]` キーでウィンドウを下方向にスクロールします。

#### (c) ノード間の移動

- n キーで次のノードに移動します。
- p キーで前のノードに移動します。
- u キーで上のノードに移動します。

#### (c) 終了

- q キーで Info バッファから脱出します (Info バッファは削除されません)。
- q で脱出した場合は、再度 C-h i で、脱出する前に見ていたノードに入ります。

#### (2) Info の終了

次の、いずれかの方法で Info を終了できます。

- 先の q コマンドで終了します。
- C-x k で完全に Info バッファを削除します。

### 4.4 キーボードマクロ

キーボードマクロとは、キーの列を短縮するためにユーザが定義するコマンドのことです。実際に動作を確認しながら定義できます。

C-x ( キーボードマクロの定義開始

C-x ) キーボードマクロの定義終了

C-x e 最新のキーボードマクロを実行

C-u C-x ( 最新のキーボードマクロを実行し、定義にさらにキーを追加

C-x q マクロの実行時にこの点にきたら確認を求める

M-x name-last-kbd-macro  
最新のキーボードマクロにコマンド名を付加

M-x insert-kbd-macro  
キーボードマクロ定義を Lisp コードにしてバッファに挿入

## 4.5 文書の作成支援

ここで述べる以外にも、自然言語の文書を編集するために便利なコマンドがたくさんあります (2章の参考文献 [3] の 20 章を参照).

### (1) 日本語の入力

Mule での日本語の入力方法として、Mule の標準である egg を用いる方法と、mlh を用いる方法を紹介します.

#### (a) egg による日本語の入力

透過モード (通常のモード) で C-\ を入力すると、入力文字変換モードに入り、モードライン (1.3節) の左端が [--] から [あ] のように変わります. 透過モードに戻るには、もう一度 C-\ を入力します.

入力文字変換モードでのコントロールキーやメタキーは通常の意味をもつので、カーソル移動などはこのモードのままでも実行できます. 通常、文字を入力するとカーソルの位置に縦棒 | が 2 つが現れ、その間に入力した文字をローマ字かな変換したひらがなが表示されます.

この縦棒をフェンスと呼び、この状態をフェンスモードといいます.

フェンスモードで SPC を入力すると、かな漢字変換を開始し、モードラインの左端は [漢] に変わります. この状態を漢字変換モードといいます. 漢字変換モードに入って初めて egg は jserver と接続し、かな漢字変換が開始されます.

#### ●フェンスモードでのコマンド

通常のコマンド C-a, C-b, C-e, C-f, C-d, C-k, DEL の他に

SPC かな漢字変換を開始

RET 文字列を確定し、フェンスモードを脱出

M-h フェンス内の文字をひらがなに変換

M-k フェンス内の文字をカタカナに変換

C-\_ JIS コード入力

C-\ 入力文字変換モードと ASCII 文字入力モードの切替

#### ●漢字変換モードでのコマンド

SPC 次候補を表示

RET すべての変換を確定

C-b 直前の文節へカーソルを移動

C-f 直後の文節へカーソルを移動

C-g 漢字変換モードを中止

C-i 文節の長さを短縮

C-o 文節の長さを伸長

C-p 前候補を表示

M-s 変換候補一覧をミニバッファに表示

#### (b) mlh による日本語の入力

“mlh” では、ローマ字をタイプして最後に “/” を入力したあと SPC を叩くと使えます. “/” の前の文字によって、次のように変換の処理を指定できます. egg のようにモードの切替を行なう必要はありません.

#### ●mLh の変換規則 (一部抜粋)

文字	変換規則
H	ひらがなを漢字に変換
L	Emacs Lisp の式を評価
X	左側を変換対象にしない
Z	1 バイト文字を 2 バイト文字に変換
c	文字列の先頭を大文字にする
d	ユーザ定義辞書による変換
f	ローマ字をひらがなに変換
h	ギリシャ文字に変換
j	JIS コードとみなして変換
k	ローマ字をカタカナに変換
p	文字列を大文字に変換
q	左側を変換対象にしない
s	なにもしない
w	スペースを挿入
上記以外	ローマ字かな漢字変換

## (2) L<sup>A</sup>T<sub>E</sub>X を使用した文書作成の支援

文書を作成するために L<sup>A</sup>T<sub>E</sub>X を使用することがあるでしょう。そのようなときには、AUC<sub>TEX</sub> パッケージを使用すると便利です。(La)TeX コマンドの補完機能、環境の入力時のオプション設定、エラーが起こったときのデバッガの起動等、便利な機能が備わっています(詳細は、Info のメニュー AUC-TEX を選択してマニュアルを見て下さい)。

### 4.6 電子メール

メール用のツールはいろいろありますが、ここでは Emacs に標準として附属しているメールを読むための RMAIL と、メールを出すための MAIL について説明します。

mh-e に関しては、伊藤先生の書かれた「電子メールの使い方 (mh メール)」を参照して下さい。その他 vm というツールもあります。

#### (1) メールの送り出し

Emacs でメールのメッセージを作成し、送りたい場合は、C-x m コマンドを用います。すると、メッセージ編集用のバッファ \*mail\* を新規に作成し、そこに次のような枠組を表示し、Mail モードとなります。

```
To:
Subject:
--text follows this line--
```

ここで、To: の行にはメールを送る相手のアドレスを、Subject: の行にはメールの内容の見出しを記述します。メールの本文は --text follows this line-- の次の行から記述します。

#### ● Mail モードでのコマンド

C-c C-s 編集したメッセージを送出。カレントバッファは \*mail\* のまま

C-c C-c 編集したメッセージを送出後、他のバッファに切替

C-c C-y メール送出的コマンドをメール読み出し用バッファから起動した場合、現在読んでいるメッセージを引用

#### (2) メールの読み出し

メールを読むのは RMAIL です。M-x rmail コマンドを入力すると、メールのプールから自分宛のメールを取り込み、通常は、ホームディレクトリにある RMAIL ファイルに書き込みます。そして、バッファ RMAIL に表示します。RMAIL モードとなります。

#### ● RMAIL モードでのコマンド

n 次のメッセージを表示

p 前のメッセージを表示

d 現在表示されているメッセージに消去の印をつける

h メールの一覧(サマリ)を作成

o 現在表示されているメッセージを Rmail 形式でファイルに保存

C-o 現在表示されているメッセージを Unix メール形式でファイルに保存

SPC 現在表示されているメッセージを前方向にスクロール

DEL 現在表示されているメッセージを後方向にスクロール

m メールを送出するための \*mail\* バッファを開く現在読んでいるメッセージを引用

d 現在表示されているメッセージに対する返事のメールを送出するために \*mail\* バッファを開く

q Rmail を終了

### 4.7 ニュースシステム

ニュースに関しては、私の書いた「ニュースシステムの使い方 (GNUS を使用)」を参照して下さい。