

Title	GNU Emacs 入門 : GNU Emacs マニュアルが読めるようになるまで
Author(s)	古林, 紀哉
Citation	大阪大学大型計算機センターニュース. 1989, 74, p. 26-36
Version Type	VoR
URL	https://hdl.handle.net/11094/65841
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

GNU Emacs 入門

— GNU Emacs マニュアルが読めようになるまで —

古林 紀哉

大阪大学基礎工学部情報工学科

nk@ouics1.ics.osaka-u.junet

1. はじめに

GNU Emacs は、UNIX 上の非常に優れたエディタであることは誰もが承知のことだと思います。GNU Emacs (以後、単に Emacs と書いたりします)は、既に日本語をエディットできるように拡張されたもの(ただし、ヘルプメッセージなどはすべて英語。以後、日本語 GNU Emacs または日本語 Emacs と書きます) が利用可能ではありますが、日本語文章やプログラムのエディタとしての機能のみでなく一つの環境としてもとらえることができます。例えば、プログラミング、コンパイル、メールやニュースの読み書きなどをすべて Emacs 上で行い、login してから logout するまで Emacs に入りっぱなしといった具合です。

しかし、Emacs は非常に優れたエディタであるにもかかわらず、ひととおり使いこなせるようになるまでに多くの時間がかかり初心者泣かせの物であることも事実です。初心者泣かせの理由としては次のようなものが考えられます。

- (1) コマンドが多すぎて何から覚えていいのかわからない。
- (2) 誤ったキータイプをした場合、コマンドが多いため何か別のコマンドとして解釈されてしまうことが多く、そこからの回復方法がわからない。
- (3) オンラインマニュアルは、すべて英語で書かれている。
- (4) マニュアルの分量が大きい。(リファレンスマニュアル:(原版[1])約300ページ、日本語版[2]約250ページ)
- (5) 初心者用に書かれた入門書がない。
- (6) LISP をしらない。

ところが、いったん Emacs のある程度の構成を理解してしまえば、利用者のレベルに応じて Emacs の機能を引き出すことができ、そこからは加速度的に慣れていくことができます。さらには、自分専用に拡張することも可能です。

そこで以下では、初心者用に Emacs の基本的な構成とキー操作を解説して行きます。ある程度の Emacs の構成が理解できれば、そこからは必要に応じてマニュアルを参照して行けばいいわけです。そういう意味で、副題を「GNU Emacs マニュアルが読めようになるまで」とつけました。

2. GNU Emacs の特徴

以下に、Emacs の代表的な特徴をあげます。

- (1) 複数のファイルを同時に編集することができる。
- (2) 同じドキュメント、別々のドキュメントにかかわらず、複数のウインドウを開いて編集することができる。(マルチウインドウ)
- (3) テキストの変更を、ほぼ無限にさかのぼって取り消すことができる。
- (4) C、LISP、Pascal などのプログラムテキストのインデント(字下げ)を自動的に行なう。
- (5) バックアップ機能、不慮の事故に対してのファイル回復機能を備えている。
- (6) ニュースやメールの読み書き、コンパイルや対話的シェルの実行などを行える。
- (7) 豊富なヘルプ機能と、オンラインマニュアルを備えている。
- (8) Emacs の下位部分はC言語、編集機能の大部分は Emacs Lisp で記述されている。Emacs Lisp で記述されている部分はユーザが書き換えることができるので、個人にあわせてカスタマイズや拡張が可能である。
- (9) GNU Emacs は、ソースコードを含めて無償で入手可能である。

3. エディタの基本的な構成

3.1. バッファ

編集しているテキストが置かれているところのことをバッファと言います。ファイルを読み込むごとに、そのファイルのテキストを保持するためのバッファが作られます。ファイルの編集だけでなく、メールの読み書きやシェルの実行などのテキストはすべてそれを保持するためのバッファが作られます。ヘルプメッセージやオンラインマニュアルにもそれ用のバッファが作られます。

それぞれのバッファは名前をもっていて、その名前(バッファ名)をつかってバッファを指定します。通常、ファイルの読み込みによって作られたバッファにはそのファイルの名前がバッファ名としてつけられます。

個々のバッファはそれぞれ次の項目を記憶しています。

- (1) 読み込んだファイル名
- (2) バッファが修正されたかどうか
- (3) メジャーモード、マイナーモード
- (4) ローカル変数

例えば、あるファイルを編集する場合の流れは、ファイルを読み込みそれによって作られたバッファにたいして変更作業を行い、そしてその内容をファイルに書き戻すというふうになります。

3.2. ミニバッファ

ミニバッファは、Emacs のコマンドが引数を読み込むための特殊なバッファで、画面の最下行のエコー領域に現われます。コマンドによって様々な引数を要求しますが、引き数が要求される場合、カーソルが最下行に移動しミニバッファの先頭に何を要求しているかが表示されます。例えばファイルを読み込む場合、

File: ~/

とプロンプトされるので、ファイル名を入力して<RET>をタイプすればよいわけです。

画面の最下行のエコー領域は、一行以内の短いメッセージを表示するための領域です。

3.3. ウィンドウ

Emacs では、画面の最下行はエコー領域として使われ、画面のその他の部分をウィンドウと呼びます。画面を二つ以上のウィンドウに分けることが可能で、それらのウィンドウには異なるバッファの一部分や、同じバッファの異なる部分を表示させることができます。

画面に見えているだけのウィンドウが存在し、一つのウィンドウが一つのバッファと対応しています。例えば、一つのバッファが複数のウィンドウで表示されている場合、一つのウィンドウでテキストを変更すると他のウィンドウでもその変更が表示されます。

各ウィンドウの最下行は、モードラインと呼ばれそのウィンドウの状況を表示しています。

例：

```
--*-NEmacs: *scratch*      (Lisp Interaction)--All--[JJJ]-----
```

常に、一つのウィンドウが選択された状態になっていて、そのウィンドウが表示しているバッファのことをカレントバッファと呼び、そのウィンドウ上にはカーソルが表示されます。テキストの変更はカレントバッファに対してのみ可能です。

3.4. キー

Emacs が一塊として理解するキーストローク列のことをコンプリートキーと呼びます。標準的にはほとんどの一文字 ("a", "b" など) はそれ一つでコンプリートキーを成しています。複数の文字から成るコンプリートキーとしては、C-a (コントロールA)、C-x C-f、C-x 4 b などがあります。

一つのコンプリートキーが一つのコマンドと対応していて、Emacs によりキー入力列がコンプリートキーとして解釈されると、それに対応するコマンドが実行されます。このキーとコマンドの対応のことをバインディングと呼びます。バインディングは固定化されているわけではなく、キーマップと呼ばれる表に記録されているので、ユーザが再定義することも可能です。

例えばキーボードで 'a' をタイプしたとすると、通常、'a' のみでコンプリートキーとして解釈され、キーマップで 'a' のエントリが参照されます。通常、'a' は自身自身を挿入するコマンド (self-insert-command) にバインドされているので、'a' という文字が挿入されます。また C-a の場合には、通常カーソルを行の先頭に移動するコマンド (beginning-of-line) にバインドされていて、カーソルが行の先頭に移動します。

Emacs は多くのキーマップを持っています。その一つグローバルキーマップは、根底になるキーバインディングを定義しています。

3.5. メジャーモード

Emacs は多くのメジャーモード (主モード) を持っていて、個々のバッファは常に一つのメジャーモードの状態にあります。それぞれのメジャーモードは、ある種のテキストを編集するために Emacs をカスタマイズします。直感的に言うと、C言語のプログラムテキストを編集するときにはC言語専用のエディタに、Pascal言語のプログラムテキストを編集するときにはPascal言語専用のエディタに変身するわけです。また、UNIX 上の標準的なエディタである vi や VAX/VMS 上のエディタ EDT に変身することもできます。

その秘密の一つにキーマップが挙げられます。メジャーモードはバッファのローカルキーマップとして自分の持っているキーマップを定義します。キー入力は、まずローカルキーマップにより解釈され、ローカルキーマップで未定義の部分はグローバルキーマップで解釈されます。メジャーモードごとが持っているキーマップに、そのモード専用のコマンドがバインドされているわけです。

バッファのメジャーモードは、ユーザが変更することが可能ですが、その初期値は自動的にほぼ望みのものになります。例えば、ファイル名が ".c", ".h", ".y" で終わるファイルを編集する場合それらのバッファはCモードに、".f" で終わる場合はFORTRANモードに自動的にになります。

ほとんどのプログラム言語には、それぞれのインデントスタイルがありますが、Emacs は、プログラム言語に応じた (メジャーモードごとに) インデントを行なってくれます。

4. GNU Emacs の基本的なコマンド

まず、これから使用する記号の意味を説明します。

C-<文字> コントロール・キーを押したまま、
 <文字>キーを押すことを意味します。

ESC <文字> エスケープ・キーを押してから離し、
 それから<文字>キーを押すことを意味します。

4.1. 起動と終了

Emacs を起動させる場合、通常、シェルで emacs <RET> と入力します。システムによっては nemacs とか xemacs という名前になっている場合や、emacs はもともとの GNU Emacs で nemacs は日本語を編集できるようにした GNU Emacs である場合がありますので、まずシステム管理者に問い合わせることをお勧めします。†

終了は次の通りです。

C-x C-c バッファのセーブを促し、Emacs を終了する

†大阪大学大型計算機センターの場合:
/usr/emacs/bin/emacs GNU Emacs
/usr/emacs/bin/nemacs 日本語 GNU Emacs

終了時にファイルに書き戻されていないバッファがあれば、バッファごとにセーブするよう確認が求められますので、心配はいりません。

4.2. 一文字入力

単に、一文字をタイプすれば、カーソルの位置にその文字が挿入され、カーソルは一文字分右に移動します。このとき、カーソル上及びそれより右にあった文字は右に移動します。

4.3. カーソルの移動

C-v	前に一画面分進む
ESC v	後ろに一画面分戻る
C-l	カーソルのある行をウインドウの中央になるようにし、画面全体を書き直す
C-f	一文字先に進む
C-b	一文字後に戻る
ESC f	一単語先に進む
ESC b	一単語後に戻る
C-n	次の行に移動
C-p	前の行に移動
ESC n	段落の終わりに移動
ESC p	段落の先頭に移動
C-a	行の最初に移動
C-e	行の最後に移動
ESC <	バッファの最初に移動
ESC >	バッファの最後に移動

カーソル移動の最も基本的な前後左右は、それぞれ、previous (前)、next (次)、forward (先)、backward (後) の頭文字をとったものですから簡単に覚えることができると思います。

日本語については、単語の切れ目を簡単には認識することができないので、半角文字、改行、タブの出現を単語の切れ目として扱います。

4.4. テキストの消去

	カーソルの直前の文字を削除
C-d	カーソルのある文字を削除
ESC 	カーソルの直前の単語を削除
ESC d	カーソル位置以降にある単語を削除
C-k	カーソル位置から行末までを削除

4.5. コマンドの中断

2つ以上のキー入力を必要とするコマンドを入力している最中に、それを中断したい場合、C-g をタイプします。

C-g コマンドを中断する

4.6. Undo

変更したテキストをもとに戻したい場合にはときは C-x u を使って直すことができます。この機能のことを Undo 機能とよびます。Undo コマンドを連続して実行させるとで、実用上、無限にさかのぼってテキストの変更を戻すことができます。

C-x u テキストの変更を取り消す
C-_ 上に同じ

4.7. ファイルの入出力

C-x C-f 新しいバッファにファイルを読み込む
C-x C-s バッファの内容をファイルの保存
C-x C-w バッファの内容を指定されたファイルに書き込む

C-x C-f をタイプした場合、Emacs はミニバッファを使ってファイル名を聞いてきます。ファイル名を入力した後、<RET> キーを押すと、ミニバッファは消え、新しいバッファにファイルを読み込み、ウィンドウにはそのバッファが表示されます。

4.8. ウィンドウとバッファの操作

C-x 0 そのウィンドウを画面から消す
C-x 1 他のウィンドウを画面から消す
C-x 2 そのウィンドウを二つに分割する
C-x o 他のウィンドウを選択する
C-x s 現在あるバッファをすべてセーブする
C-x C-b バッファリストを表示する
C-x b そのウィンドウに他のバッファを表示させる

ヘルプや、幾つかのコマンドからの出力を表示するために現れたウィンドウを消すためにも、上記のコマンドを使用します。

4.9. コマンド名でのコマンドの実行

Emacs には非常に多くのコマンドがあり、これらすべてをキーにバインドすることは不可能です。そこで、バインドされていないコマンドや、バインドされてはいるがどのキーにバインドされているのかわからないコマンドを、コマンド名を用いてそれを実行させることが必要になってきます。それには、ESC x に続いてコマンド名そのものを入力してやります。

ESC x コマンド名を読み込みコマンドを呼び出す

4.10. ヘルプ

Emacs は、豊富なヘルプ機能を備えています。下記はその一例にすぎません。

C-h C-h	ヘルプの種類を表示
C-h C-h C-h	ヘルプ機能の解説
C-h b	キーバインド・テーブルを表示
C-h i	オンラインマニュアル呼び出し
C-h m	現在のメジャーモードの解説
C-h t	Tutorial
C-h T	チュートリアル (個別指導)

5. GNU Emacs に慣れる

5.1. チュートリアル

GNU Emacs には、初心者が実際に Emacs を使いながら学習できるようにチュートリアルが用意されています。日本語 GNU Emacs の場合は、英語と日本語の二種類のチュートリアルが用意されています。このチュートリアルには基本的なコマンドが解説されています。既に Emacs を使っている人なら誰もが一度は自己学習したことがあると思いますし、これから Emacs を使おうと思っている人ならまず最初にトライしなければならぬのがチュートリアルです。。ゆっくり行なっても約1時間で一通り追えると思います。

Emacs に入ってから、C-h T とタイプすると日本語によるチュートリアルが始まります。次の例はチュートリアルの最初の画面です。

=====
日本語 GNUEMACS(NEmacs) 入門編
=====

注意: この入門編は、「習うより慣れろ」をモットーに作成されています。行の左端の ">>" は、その時何をすべきかを指示しています。

NEmacsのコマンドを入力するときには、一般的にコントロール・キー（キー・トップに、CTRL あるいは、CTL と書いてある）やメタ・キー（普通、エスケープ・キーを使う）が使われます。そこで、CONTROL とか META とか書く代わりに、次のような記号を使うことにします。

C-<文字> コントロール・キーを押したまま、<文字>キーを押します。例えば、C-f は、コントロール・キーを押しながら f のキーを押すことを意味します。

>> それでは、C-v (View Next Screen; 次の画面を見る) をタイプしてみてください。次の画面に進むことができます。

画面の指示にしたがって、どんどん自分でやってみてください。一通り終われば、とりあえず Emacs が使えるようになると思います。

5.2. イニット・ファイル (.emacs)

Emacs はスタート時に、ホームディレクトリの ".emacs" というファイルをロードします。このイニットファイルの内容により、Emacs が初期化またはカスタマイズされます。ところがイニットファイルは、Emacs Lisp を使って記述しなければならないので初心者には少々頭の痛いところです。パラメータの設定など簡単な初期化は初心者でも行えますが、少し手の凝ったカスタマイズとなると Emacs Lisp の知識が必要です。

自分では ".emacs" を書けないが、他の計算機利用者の中でずいぶん使いやすそうに ".emacs" 書いている人がいるとします。そういった場合、その人の ".emacs" を借用してみるというのが世の中の常ですが、簡単にそれを実現する方法があります。

```
emacs -u <usr name> <RET>
```

として、Emacs を立ち上げます。この場合、自分の ".emacs" の代わりに <usr name>さんの ".emacs" をイニットファイルとしてくれます。‡

それでは、イニットファイルのごく初歩的な内容を二三紹介します。最初の例は少し複雑ですが、デリートキーとバックスペースキーの役割を入れ換えるためのものです。

‡大阪大学大型計算機センターでは、ユーザ名 'guest' の ".emacs" にある程度初期化の記述がされていますので、

```
emacs -u guest <RET>
```

とすることで、ある程度初期化された Emacs を立ち上げることができます。

Emacs の初期の設定では、デリートキー () は削除、バックスペースキー (C-h) はヘルプの意味を持っています。通常ヘルプよりも削除のほうを頻繁に使用しますが、自分の使っている端末ではバックスペースキーのほうが押し安いため、バックスペースキーを削除用に使用したいことがあります。UNIX のシェルでは 'stty' コマンドをつかってこのようなことをしますが、シェルでの設定は Emacs 上では無効力です。そこで、".emacs" に例のような Emacs Lisp を入れます。

```
(let ((the-table (make-string 128 0)))
  ;; Default is to translate each character into itself.
  (let ((i 0))
    (while (< i 128)
      (aset the-table i i)
      (setq i (1+ i))))
  ;; Swap Back-Space and Delete
  (aset the-table ?177 ?h)
  (aset the-table ?h ?177)
  (setq keyboard-translate-table the-table))
```

次は load-path を設定している例です。load-path は Emacs がリブライブラリをロードするさいに探索するディレクトリのリストです。Emacs 自身のリスブコードのなかには、事前にロードされているもの、必要に応じてロードされるものがあります。それに加え、利用者個人が記述したリスブコードなどがあります。Emacs 自身のリスブコードが存在するディレクトリは初期値として load-path の中に加わっていますが、個人用のそれは設定する必要があります。この例の中のディレクトリ名の部分は個人に応じて書き換えて下さい。§

```
(setq load-path
  (append
    (list (expand-file-name "~/emacs/lisp")
          (expand-file-name "~/emacs/local"))
    load-path))
```

次は変数の値を設定している例です。c-tab-always-indent という変数の値を nil に設定しています（ここでは、変数 c-tab-always-indent の意味については触れません）。このようにただ変数の値を設定するだけならば、関数 setq を用いて行なうことができます。

```
(setq c-tab-always-indent nil)
```

このほかイニットファイルには十人十色の様々な記述が成されます。他の利用者のイニットファイルを研究するのもよい勉強になると思います。

5.3. インフォ

GNU Emacs ではインフォと呼ばれるオンラインマニュアルが使用可能です。ただし、残念ながらすべて英語です。インフォの内容は、GNU が提供している Emacs の

§大阪大学大型計算機センターでは、load-path に (expand-file-name "~/local" exec-directory) も加えることをお勧めします。そのディレクトリには、有用なリスブコードがあります。

リファレンスマニュアルとまったく同じものです(正確には、`texinfo`形式でかかれたリファレンスマニュアルが、オンラインマニュアルにも利用でき、また清書プリントアウトもできるようになっている)。インフォでは、このリファレンスマニュアルを木構造で管理していて効率よく内容を画面で見ることができます。

`C-h i` とタイプするとインフォモードに入ります。次の例のように、木構造のなかのノードを移動する基本的な操作法は画面に現われていますのでここで詳しくは説明しませんが、画面に現われている操作法だけである程度簡単に必要な項目を表示させることができます。そこで `h` をタイプすることでインフォ自身の使い方ことが表示されますので、まず最初にこの項目を読むことをお勧めします。

```
File: dirNode: Top      This is the top of the INFO tree
This (the Directory node) gives a menu of major topics.
Typing "d" returns here, "q" exits, "?" lists all INFO commands, "h"
gives a primer for first-timers, "mTexinfo<Return>" visits Texinfo topic,
etc.
--- PLEASE ADD DOCUMENTATION TO THIS TREE. (See INFO topic first.) ---

* Menu: The list of major topics begins on the next line.

* Info: (info).  Documentation browsing system.

* Emacs: (emacs). The extensible self-documenting text editor.

* Nemacs: (nemacs). Japanese version of Emacs.

* VIP: (vip).  A VI-emulation for Emacs.

* GDB: (gdb).  The source-level C debugger.
```

ただし、お手元に印刷されたリファレンスマニュアルがあり、その索引のほうが便利と感じられる方はそちらを利用したほうが良いでしょう。日本語訳されて印刷されたリファレンスマニュアルもありますので、必要に応じて使い分けるのが良いのではと思います。

5.4. モードのヘルプ

GNU Emacs になれていく過程で最も手助けとなるのがモードのヘルプ (`describe-mode`) です。このコマンドは `C-h m` とタイプすることで起動でき、現在のメジャーモードの簡単な解説とそのバッファで設定されているバインディングすなわちキーコマンドの解説が表示されます。

GNU Emacs には多くのメジャーモードがありますが、その使い方はリファレンスマニュアルを見なくても、`describe-mode` コマンドの表示だけでほとんど理解できると思います。

一つ典型的な例を示します。ファイルをオープン使用として `C-x C-f` をタイプしたとします。画面の最下行に次のような表示がなされたとします。

```
Find file: ~/
```

そこで、ファイル名を入力せずにごく <RET> をタイプしてしまったとします。このとき、良く見慣れた光景が現われると思います。これはディレクトリエディタと呼ばれるものです(あえて、出力例はつけません)。ここで、C-x C-f をもう一度やり直すのも手ですが、ディレクトリエディタの使い方を覚えた方が後々のためになることは言うまでもありません。このような場合に、C-h m をタイプして describe-mode コマンドを実行させ、このモードの使い方を表示させるわけです。Emacs のヘルプメッセージも一つのバッファを使用していますので、ヘルプの内容が画面内に納まっていない場合が多々あります。そのときには、今までに覚えたコマンドをつかって、そのバッファの内容を見て下さい。

6. あとがき

何度も述べますが、今回の内容は GNU Emacs の入口にすぎません。少しでも容易に入門できるよう心がけて書きましたので、残念ながら Emacs の強力な機能は殆ど現われていません。Emacs には非常に多くのコマンドがあり、それをすべて覚えるのは初心者でなくても不可能です。しかし、ユーザが欲するところの機能はほとんどの場合、それをおこなうコマンドが存在し、そのコマンド名も容易に探し出せる機能が Emacs には備わっています。Emacs になれるために Emacs を使うことができれば、もうあなたは初心者の域を脱したと言えるでしょう。

参考文献

- [1] Richard Stallman, "GNU Emacs Manual", Fifth Edition, Emacs Version 18, October 1989
- [2] 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年2月