

GNU Emacs の使い方

2015 年 10 月 2 日

目次

1	GNU Emacs を使おう	1
1.1	キー表記法	1
1.2	起動と終了	1
1.2.1	起動方法	1
1.2.2	終了方法	1
1.2.3	練習	2
1.3	画面構成	2
1.4	困ったときには	3
1.4.1	操作の中断	3
1.4.2	分割された Window を一つにする	4
1.5	Emacs におけるファイル編集の基本操作	4
1.5.1	例題 1 — ファイルの新規作成	4
1.5.2	例題 2 — 既存ファイルの編集	5
1.6	ファイルとバッファ	5
1.7	まとめと練習	7
1.7.1	まとめ—Emacs によるファイル編集の流れ	7
1.7.2	練習	8
1.8	その他の操作	8
1.8.1	ファイル名を指定してファイルに保存	8
1.8.2	Emacs 終了時のファイル保存	9
1.9	Emacs Tutorial	9
1.10	主な Emacs コマンドの一覧	10

1 GNU Emacs を使おう

UNIX 系の OS で利用可能なエディタは vi を始めとして、いろいろとあります。その中でも、vi と並んで代表的なエディタとして GNU Emacs があります。本章では、この GNU Emacs の使い方の基本を紹介します¹。

1.1 キー表記法

GNU Emacs の説明文書では、次のキー表記を使うのが一般的です。

C-文字 コントロールキー (<CTRL>) を押したまま、文字を押します。例えば、C-f はコントロールキーを押したままで f のキーを押すことです。

M-文字 メタキー (普通は <ALT> がメタキーです) を押したまま、文字を押します。または、<ESC> を押して離してから文字を押します。

1.2 起動と終了

1.2.1 起動方法

GNU Emacs (以下、単に Emacs という) を起動するためのコマンドは emacs です。X Window System が動作している環境では、コマンド行オプションの有無によって、二通りの動作形態があります。

1. emacs (オプション無しで起動) : 起動時に新規ウィンドウを生成して動作する。

Windows のアプリケーションを操作するのと同様の感覚で、マウスを使って Emacs を操作できます。

2. emacs -nw または emacs --no-windows : コマンドを入力したウィンドウ内で動作する。

X Window System を用いずに Emacs の操作をすべてキーボードで行うことになります。

X が動作していなければ、いずれの起動法を使っても、Emacs はコマンドを入力したウィンドウ内で動作します。ただし、この資料は X が動作していることを前提として記述しています。

これから、Emacs の使い方を学びますが、できるだけマウスを使わないで操作する方法までを習得することが望ましいです。それは、マウス操作よりもキーボード操作の方が作業効率が良いことが多いし、マウスはいつも使えるとは限らないからです。

1.2.2 終了方法

マウスを使って Emacs を終了するには、メニューから

```
File -> Quit
```

を選択します。

キーボード操作による場合は

```
C-x C-c
```

です。

¹GNU Emacs は UNIX の標準コマンドではありませんので、必ずしもすべての UNIX で利用できるとは限りません。

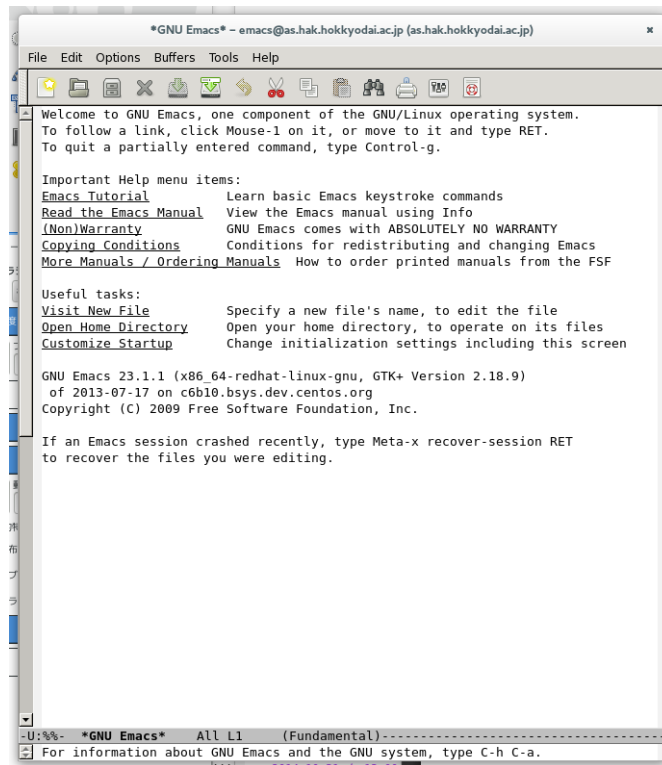


図 1: Emacs の起動画面例

1.2.3 練習

1. コマンド行に `emacs` と打って Emacs を起動しましょう。起動元のウィンドウに新しいプロンプトが現れないことと、Emacs の起動画面を観察したら終了しましょう。
2. コマンド行に `emacs -nw` と打って Emacs を起動しましょう。起動画面を観察したら、キーボード操作で終了しましょう。

1.3 画面構成

コマンド行に `emacs &` と打って、Emacs を起動してください。新しいウィンドウに図 1 のような起動画面が表示されましたね。& をつけて Emacs を起動したので、起動元のウィンドウに新しいプロンプトが現れていることも確認してください。

- 最上行の File Edit Options ... 等と表示されている部分をメニューバー (menu bar) と呼びます。ここをマウスでクリックして、メニューから Emacs の操作を選択することができます²。すぐ下の行には、マウス操作のためのボタンが並んでいます。
- 最下行の For information about ... と表示されている行をエコー領域 (echo area) と呼びます。ここに Emacs からの様々なメッセージが表示されます。

²すべての操作がメニューでできる訳ではありません。

この行に、Emacs で編集したいファイル名等を入力することもあります。この行は、キーボードからの入力を受け付ける状態になったとき、ミニバッファ (minibuffer) と呼ばれます。

- 残る部分を (Emacs における) ウィンドウと呼びます。図 1 では、様々な文字が表示されている一番広い場所がウィンドウです。ここで文書の編集等を行います。
なお、X Window System における “ウィンドウ” (これまでウィンドウと呼んできたもの) のことを Emacs ではフレーム (frame) と呼びます。
- ウィンドウの最下行 (エコー領域を含めて数えると下から 2 行目) をモード行 (mode line) といいいます。ここにはウィンドウに関する様々な情報が表示されます。

ここで少し Emacs を操作してみましょう。

1. キーボードから何か文字を打ってください。
Emacs のウィンドウに文字を入力できましたか。
2. マウスを使い、メニューバーの File の項目から Split Window を選択してください。
ウィンドウが分割 (split) されて二つになります。Emacs では複数のウィンドウを使ってファイルを編集できます。
3. 同じく File メニューの Remove Splits を選択してください。
予想どおりになりましたか。

次に、ウィンドウ分割をキーボード操作で行ってみます。

1. まず、File メニューをクリックしてプルダウンメニューを表示し、メニュー項目をよく観察してみましょう。
項目の右側に、括弧で囲まれた記号があります。この記号は、当該操作をキーボードで行うためのキーを表しています。
2. Split Window の項目の右には (C-x 2) とあるのを確認したら、この表示に従って、キーボードから C-x 2 をタイプしてください。C-x 2 は、コントロールキーを押しながら x を押し、続いて (空白は打たずに) 2 だけを押すことを意味します。
3. キーボード操作でウィンドウを一つにしましょう。File メニューをマウスで開き、キー操作による方法を確認したら、実行してください。

1.4 困ったときには

1.4.1 操作の中断

Emacs を使っていて良く分からない状況に陥ったときのために、次の Emacs のコマンド (キー操作) を是非覚えておいてください。

Emacs におけるコマンドの取り消し: C-g

1 回押しただけでうまくいかなければ 2 回押しします。

たとえば、キー操作でウィンドウを二つに分割しようと思ったけど、途中で止めたくなったとします。ウィンドウ分割のためにまず C-x を押しすが³、続いて 2 を押す代わりに C-g を押しすと、その操作を取り消すことができます。

1.4.2 分割された Window を一つにする

Emacs では、操作の途中で自動的にウィンドウが二つになることがあります。その場合には、ウィンドウ内に表示された指示に従って対処するか、よく分からなければウィンドウを一つにする操作 (C-x 1 やメニューの Remove Split など) をしてください。

1.5 Emacs におけるファイル編集の基本操作

1.5.1 例題 1 — ファイルの新規作成

ファイルの中身が

```
ls - list directory contents
mv - move files
```

である `commands_file` という名前のファイルを Emacs を使って作成します。以下の操作を行ってください。

1. Emacs が起動していなければ `emacs &` で起動してください。
2. ファイルを新しく作成するには、C-x C-f を押しします。このコマンドが第 1.10 章の Emacs コマンド一覧のどこに載っているかを確認しておきましょう。

最下行のエコー領域に Find file: ~/ というメッセージが現れます。カーソルもエコー領域に移動しましたので、ここにキーボードから文字を入力できます。エコー領域は、文字入力できる状態になったとき、ミニバッファと呼ばれるのでしたね。

3. 作成するファイルの名前 `commands_file` をミニバッファに入力して <ENTER> を押ししましょう。入力を誤ったら <ENTER> を押す前に <BS> を使って修正するか、C-g を押して 2. からやり直してください。

ウィンドウ下部のモード行には、これから作成するファイルの名前 `commands_file` が表示されます。カーソルは上部のウィンドウに戻ります。

4. 作成するファイル `commands_file` の中身

```
ls - list directory contents
mv - move files
```

を Emacs のウィンドウに入力してください。

改行するには <ENTER> を押しします。文字の修正には <BS> が使えます。カーソルの移動には、ここでは矢印のキーを使うことにします。

³この状態で少し待って、エコー領域に入力したキー (C-x-) が表示されることを確認しておきましょう。

5. Emacs に入力したテキスト (文書) をファイルに保存するには `C-x C-s` を押します。

エコー領域 (最下行) に `Wrote ...` というメッセージが表示され、保存に成功したことがわかります。これでファイル `commands_file` ができあがりしました。

6. 一度 Emacs を終了し、ファイル `commands_file` の内容を `cat` コマンドで確認してください。

1.5.2 例題 2 — 既存ファイルの編集

第 1.5.1 節の例題 1 で作成したファイル `commands_file` の内容を

```
ls - list directory contents
mv - move files
cat - print files
```

に変更します。また、Emacs でのファイル編集作業と併行して UNIX のコマンドも使ってみます。

1. `emacs &` と打って Emacs を起動してください。
2. ファイル `commands_file` が存在することを、`emacs` コマンドを入力したウィンドウで `ls` コマンドを実行して確認してください。ファイル名が違っていたら、正しいものに変更してください。
3. 編集したいファイル `commands_file` を開きます。方法は第 1.5.1 節 例題 1 でファイルを新規作成したときと同じです。`C-x C-f` を押してください。最下行のミニバッファにはファイル名 `commands_file` を入力して `<ENTER>` を押してください。

既存ファイル `commands_file` の内容が Emacs のウィンドウに表示され (Emacs に読み込まれ)、モード行 (下から 2 行目) には `commands_file` と表示されます。

4. ファイルに追加すべき内容 (`cat - print files`) を Emacs のウィンドウに入力しましょう。
5. 例題 1 と同じ方法で、Emacs のウィンドウに表示されている内容をファイルに保存してください。Emacs はまだ終了しないでください。
6. ファイル `commands_file` の内容を `cat` コマンドで確認しておきましょう。

1.6 ファイルとバッファ

図 2 は、コンピュータの構造を、これまでに行ったことを理解するために必要な部分に限って、ごく簡単に表現したものです。

CPU (central processing unit) はコンピュータの頭脳にあたる役割をし、コンピュータの動作を制御したり、様々な計算を行います。

主記憶装置 (main memory; 以下、単にメモリと呼ぶ) は、実行中のプログラムやデータを格納する場所です。CPU は、処理に必要なデータをメモリから読み取り、処理結果をメモリに書き込みます。

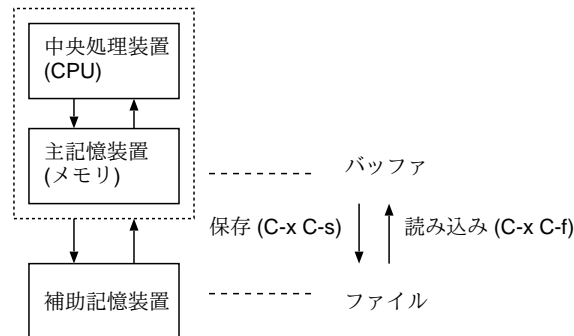


図 2: ファイルとバッファ

Emacs が動いているときには、Emacs のプログラムや編集中のテキスト (文書) はメモリにあります。Emacs がテキストを保持するために使うメモリ内の領域を、Emacs の用語でバッファ (buffer) といいます。

メモリに対するデータの読み書きは高速に行えますが、メモリの容量は比較的小さく、メモリ内のデータはコンピュータを停止すると消えてしまいます。そのため、メモリはデータの永続的な保存には使えません。

補助記憶装置 はメモリに比べて動作が遅い反面、記憶容量が大きく、中身はコンピュータを停止しても消えません。ファイルは補助記憶装置にあります。

さて、Emacs のウィンドウに表示されるテキストは、ファイルではなく、バッファの内容です。したがって、Emacs でファイルを作成・編集するには、まず、そのためのバッファを用意する必要があります。これを行うのが C-x C-f です。この操作ではファイル名を入力しますが、その名のファイルが存在しなければ、同名の空のバッファが用意されます (第 1.5.1 節 ファイルの新規作成)。ファイルが存在すれば、ファイルは同名のバッファに読み込まれます (第 1.5.2 節 既存ファイルの編集)。なお、バッファに読み込まれるのはファイルの内容のコピーですから、ファイルを読み込んで元ファイルが無くなる訳ではありません。

Emacs を終了するとバッファは消えるので、保存したいバッファの内容はファイルに入れる必要があります。それをするのが C-x C-s です。

編集中のバッファが保存済みかどうかは、モード行 (下から 2 行目) におけるバッファ名の左側の表示でわかります。ここが -- から ** に変わったら、バッファとファイルの内容に違いが生じたこと、すなわち、バッファの内容を変更したのに未保存であることを表しています。

では、上記のことを確認するために、以下の操作を行きましょう。

1. Emacs が起動していなければ、`emacs &` と打って起動してください。 `commands_file` をバッファに読み込んでいなければ、読み込む操作をしてください。
2. Emacs のウィンドウに表示されている、 `commands_file` という名のバッファを消してみます。メニューから

File -> Close (current buffer)

を選んでください。

3. `ls` コマンドを使って、消したバッファと同じ名前のファイルが存在することを確認してください。

バッファを消してもファイルは消えませんね。

4. Emacs を終了せずに、引き続き `commands_file` を編集するにはどうしたらいいでしょう？

ファイル `commands_file` を再度バッファに読み込んでください。

5. Emacs のモード行 (下から 2 行目) に表示されているバッファ名 `commands_file` のすぐ左側の表示が `--` であることを確認してから、バッファ (Emacs のウィンドウ) 3 行目の `cat` の説明を

```
cat - concatenate and print files
```

に変更してください。

モード行の表示が `**` に変わりましたね。

6. UNIX のコマンドでファイル `commands_file` の内容を確認してください。

バッファの内容を変更しただけでは、ファイルは変わりません。

7. バッファの内容をファイルに保存してください。

モード行の表示が `--` になりましたね。バッファとファイルの内容が同じになりました。

8. ファイル `commands_file` の中身を `cat` コマンドで確認してください。

1.7 まとめと練習

1.7.1 まとめ—Emacs によるファイル編集の流れ

Emacs を用いてファイルを編集するときの流れは次のとおりです。

1. Emacs を起動する。— `emacs &` (X ウィンドウシステムが使えないときには `&` を付けない)
2. 新規ファイル用のバッファを用意する。または、編集したいファイルをバッファに読み込む。
— `C-x C-f`
3. バッファへの文字の追加・削除・変更などの編集作業を行う。
4. バッファの内容をファイルに保存する。— `C-x C-s`
5. 必要に応じて 2 から 4 の作業を繰り返す。複数の文書を編集するときでも Emacs を終了して再度起動する必要はない。
6. すべての作業が終わったら Emacs を終了する。— `C-x C-c`

1.7.2 練習

1. 次の内容を持つファイル `emacs_motion` を Emacs で作成してください。

```
backward forward
character C-b      C-f
line      C-p      C-n
```

ファイル作成後も Emacs を終了しないでください。

2. 起動中の Emacs を使って、既存のファイル `commands_file` に、これまでと同じ形式で、`cp` と `rm` の説明

```
cp - copy files and directories
rm - remove files or directories
```

を追加してください。作業を終えたら Emacs を終了してください。

3. ホームディレクトリ以外に存在するファイルを Emacs に読み込む操作を試みましょう。
 - (a) ホームディレクトリに `unix` というディレクトリがあることを、`ls` コマンドで確認してください。無ければ `mkdir` コマンドで作成してください。
 - (b) `mv` コマンドを使って `emacs_motion` を `unix` に移動してください。
 - (c) ホームディレクトリで Emacs を起動し、`C-x C-f` してください。
 - (d) ミニバッファを Find File: `~/unix/emacs_motion` にしてから `<ENTER>` を押してください。

ミニバッファでの `~` はホームディレクトリを表していて、`~/file` の代りに `~/sub-dir/file` とすれば、`~` の子ディレクトリ (`subdir`) のファイル (`file`) を指定したことになります。このことはファイルの読み込み時に限らず、保存時などでも共通です。

- (e) `emacs_motion` が読み込まれたことを確認したら、Emacs を終了してください。
- (f) ディレクトリ `unix` の `emacs_motion` を、適切な UNIX コマンドを使って、ホームディレクトリに移動してください。

1.8 その他の操作

1.8.1 ファイル名を指定してファイルに保存

編集中のバッファの内容を、ファイル名を指定して保存するには `C-x C-w` を使います。既存のファイルをバッファに読み込んでからこのコマンドを使うと、元のファイルを異なるファイル名で保存することができますので、UNIX の `cp` コマンドでファイルを複製するのと同様のこともできます。

以下では、ファイル `commands_file` と同じ内容を持つファイル `commands_copy` を作ります。

1. ファイル `commands_file` をバッファに読み込んでください。

2. C-x C-w を押しましょう。

エコー領域 (ミニバッファ) にメッセージ Write file: ~/ が現われます。

3. commands.copy と打って、<ENTER>を押しましょう。

この操作で新しいファイル `commands.copy` が作成されましたが、モード行に表示されているバッファ名も `commands.copy` になったことに注意してください。C-x C-w を実行すると、編集時のバッファの内容が別のファイルに書き込まれるだけでなく、バッファ自体の名前も変わります。引き続きこのバッファで編集作業をすれば、それは元の `commands.file` ではなく `commands.copy` に対してなされます。

1.8.2 Emacs 終了時のファイル保存

バッファの内容を変更したにもかかわらず、それをファイルに保存しないまま Emacs を終了しようとする、バッファの内容をファイルに保存するかどうかを尋ねるメッセージが、ウィンドウ下部のエコー領域に表示されます⁴。その場合には、保存の必要性を判断して、y, n, yes, no 等で答えてください。

例えば、第 1.9 節で紹介するチュートリアルの記事に、何か書き込みをしてから Emacs を終了しようとする、このメッセージが現れます。普通はチュートリアルを保存する必要がないので、Save file ... ? に対しては n を、... exit anyway? に対しては yes を打ってください。

1.9 Emacs Tutorial

Emacs 起動時のメッセージに

Important Help menu items:

Emacs Tutorial Learn-by-doing tutorial for using Emacs efficiently.

と記されているとおり、Emacs をキーボード操作で効率よく使う方法を学ぶチュートリアルがあります。Help メニューには“Emacs Tutorial (C-h t)”という項目がありますので、マウスでこれを選択するか、CTRL-h に続いて t を押せばチュートリアルを始めることができます⁵。

⁴Emacs には、ファイル編集以外の用途に用いられるバッファ (例えば *scratch*) が存在します。それらの内容を変更しても、終了時に保存のメッセージは現れません。

⁵Emacs の標準設定では C-h にヘルプ表示の機能が割り当てられているのですが、C-h が他の機能に変更されているシステムがあるかもしれません。その場合、M-x help <ENTER> を打てばヘルプを表示できます。ヘルプを介さずにチュートリアルを始めるには M-x help-with-tutorial <ENTER> とします。

1.10 主な Emacs コマンドの一覧

C-x C-c	終了
C-g	コマンドの取り消し
C-x C-f	ファイルを開く (バッファへのファイル読み込み)
C-x C-s	現在のバッファをファイルに保存
C-x C-w	現在のバッファに名前をつけて保存
C-x s	編集中のバッファをすべてファイルに保存
C-h	ヘルプ (M-x help <ENTER>)
C-h t	チュートリアル (M-x help <ENTER> t)
C-x u	変更の取り消し (undo)
C-_	変更の取り消し (undo)
C-f	カーソルを 1 文字右 (forward) に移動
C-b	カーソルを 1 文字左 (backward) に移動
C-a	カーソルを行頭に移動
C-e	カーソルを行末に移動
C-p	カーソルを前 (previous) の行に移動
C-n	カーソルを次 (next) の行に移動
C-v	次の画面を見る
M-v	前の画面を見る
<BS>	カーソル左の 1 文字を削除
C-d	カーソル位置の 1 文字を削除
C-k	行末まで消去 (kill)
C-y	最後に保存した kill-ring の内容取りだし (yank)
C-<SPACE>	マーク (領域の始点) の設定
C-w	領域消去 (kill-ring に保存)
M-w	領域を kill-ring に保存
C-x b	バッファの切り替え
C-x C-b	バッファの一覧表示
C-x k	バッファの削除
C-x o	他のウィンドウに移動
C-x 0	現在のウィンドウを削除
C-x 1	現在のウィンドウのみ残す
C-x 2	現在のウィンドウを上下に分割
C-x 3	現在のウィンドウを左右に分割
C-s	検索
M-%	置換