

エディタ Emacs

濱田 龍義 (福岡大学/JST CREST)

平成 21 年 9 月 14 日

1 序

KNOPPIX/Math を用いると、世界中で開発されている数学ソフトウェアを利用することができます。収録されている数学ソフトウェアのうち、特に研究に活用されているシステムについて言えることですが、UNIX の基礎知識 (標準入出力、シェル) や Emacs や vi 等のエディタ (Editor) と呼ばれるアプリケーションの使い方に慣れておくとう便利です。そこで、ここでは Emacs について簡単に解説します。

普段、我々が意識せずに行っている作業ですが、コンピュータに文字を入力したり、マウスを用いて絵を描いたりという操作は、すべてメモリー (主記憶) 上に確保された領域、すなわち バッファ (Buffer) に一旦データを書き込んだり、バッファからデータを読み込んだりすることで実行されます。

例えば Web メールを例に考えてみましょう。文章を入力している最中に、「下書き保存」をせずに、Web ブラウザを終了すると入力した文章は消えてしまいます。これは、入力されたデータがメモリー上のバッファに保存されているだけで、ハードディスクには保存されていないからです。「下書き保存」もしくは「送信」をマウスでクリックすることで、初めてハードディスクにデータが保存されます。

コンピュータ上でお絵書きソフトを使ったことがある方もいると思います。マウスで線を描く、色を塗るといった操作も、メモリー上に用意されたバッファに対して 2 進法の $\{0, 1\}$ データを書き込んでいます。

一般に、メモリーはデータを読み書きする速度が速いのですが高価です。ハードディスクはデータを読み書きする速度は遅いのですが、メモリーに比べれば安価にデータを保存することができます。そこで、メモリー上にバッファという緩衝地帯を設け、適当なタイミングでハードディスクのデータを読み書きすることで、コンピュータの操作性を高めています。また、メモリー上のデータはコンピュータの電源を切ると消えてしまいます。データを長期的に保存するためには、ハードディスク等の補助記憶装置が必要となります。


スクールではエディタを用いて命令を記述したテキストファイルを作成し、数学ソフトウェアに読み込むことで計算を実現します。Emacs の正式名称は GNU Emacs (ぐにゅーいーまっくす) です。フリーソフトウェア (Free Software) の概念を提唱した Richard Stallman が作成し、現在は FSF (Free Software Foundation) という組織で開発が進められフリーソフトウェアとして公開されています。Emacs はスクリーンエディタの一種ですが、Lisp というプログラミング言語によって拡張が可能であり、極めて多彩な能力を備えているため、1 つのコンピュータ環境と言っても良いでしょう。

2 GNU Emacs

2.1 引数無しで起動

この実習では Emacs が備えている豊富な機能のうち、ほんの一部だけ紹介します。既に Emacs を日常的に使いこなしている方は、軽く読み飛ばしてください。

Emacs について詳しいことを知りたい方は、「新 The UNIX Super Text[上]」の第 12 章、第 19 章を読むことで、かなりのことを学ぶことができます。さらに詳しいことを学びたいときは、第 8 章 8.6 を参照してください。参考書等も紹介されています。

まず KNOPPIX/Math 上でコンソール画面を開いてください。デスクトップ下部のメニューから左から 7 番めの黒い画面のアイコン  をシングルクリックすると、コンソールが起動します。ここでは、コンソールからコンピュータに命令を入力します。

コンソール画面を開くと

```
knoppix@Knoppix:~$
```

と表示されますが、knoppix@Knoppix:~\$ の部分を プロンプト(prompt) と呼びます。ここで ~ という記号は特別な意味を持っており、ホームディレクトリ (Home Directory) を表しています。各ユーザが自由にファイルを作成したり、計算結果を出力して保存することができる領域です。

今後、作業を円滑に進めるため、ホームディレクトリの中にディレクトリ (Windows で言うところのフォルダ) を作成して、作業を進めます。ディレクトリを作成するコマンドは mkdir です。

```
knoppix@Knoppix:~$ mkdir tutorial
```

コマンド mkdir の後ろに、新たに作成するディレクトリ名 (この場合は tutorial) を指定します。これをコマンド mkdir の 引数 (ひきすう) としてディレクトリ名を指定すると言います。コマンドでディレクトリ tutorial を作成したらカレントワーキングディレクトリを ~/tutorial に変更して Emacs を起動します。

```
knoppix@Knoppix:~$ cd tutorial
knoppix@Knoppix:~/tutorial$ emacs
```

ここでは emacs を引数無しで起動しています。Emacs を起動すると Welcome メッセージが出ます。まず一番に重要なことは、Emacs の終了の仕方です。Emacs を終了するには、C-x C-c を行います。これは、「Ctrl」キーを押しながら、2 個のキー「x」、「c」を続けて押すという動作です。

次に適当なキーを押してみましよう。すると、

```
;; This buffer is for notes you don't want to save, and for Lisp evaluation.
;; If you want to create a file, visit that file with C-x C-f,
;; then enter the text in that file's own buffer.
```

と出ます。このバッファは、今の段階では使いません。一応、簡単に訳しておく

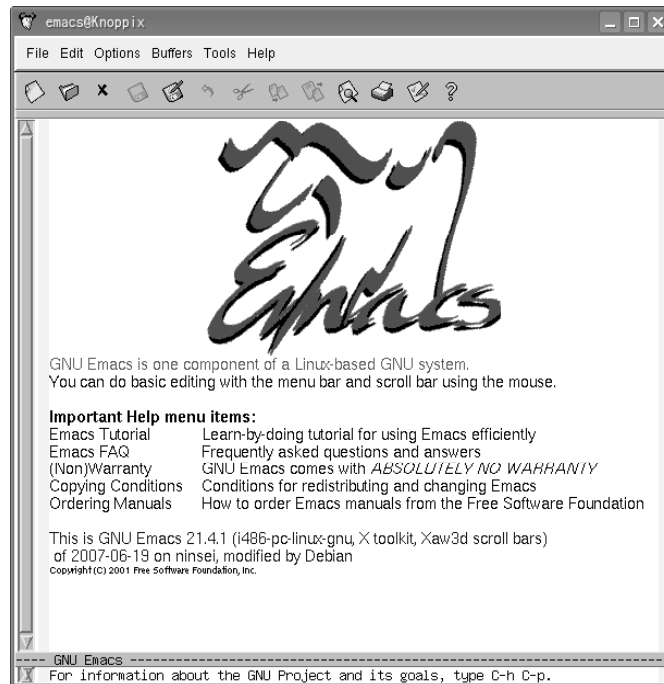


図 1 KNOPPIX/Math における Emacs の起動メッセージ

```
;; このバッファは、ちょっとしたメモや、Lisp の評価用です。
;; もし、ファイルを作成したいならば、C-x C-f を用いてファイルを開き、
;; そのファイルのバッファに文字を入力してください。
```

と書いてあります。それでは、エディタ Emacs を用いて、内容が “Emacs is a screen editor.” という 1 行だけのファイル emacs.txt を作成します。上記にも書いてあったように、まずは、ファイルを作成するためのバッファを用意します。上のメッセージにもあったように、ファイルを作成するバッファを用意する命令は C-x C-f です。「Ctrl」キーを押しながら、「x」と「f」のキーを続けて押すと、エコー領域に

```
Find file: ~/tutorial/
```

と表示されるはずですが、そこで、次のように入力して Enter キーを押します。

```
Find file: ~/tutorial/emacs.txt
```

ファイル~/tutorial/emacs.txt を作成するためのバッファが用意されました。エコー領域を見ると (New file) と表示されているはずですが、あとは入力したい文字列をバッファに入力するだけです。

まずは、1 行だけ

```
Emacs is a screen editor.
```

と入力します。行末のピリオドを入力したら「Enter」キーを押して改行してください。

Emacs では、「Ctrl」キーや「Meta」キーを用いて命令します。C-x という記号は「Ctrl」キーを押しながら「x」を押す操作です。例えば C-x o と書かれていたら、「Ctrl」キーを押しながら「x」を押し、「Ctrl」キーから指を離してから「o」を押す動作となります。

また、M-x という記号は「Meta」キーを押しながら「x」を押す操作です。しかし、一般の PC には「Meta」キーは存在しないので、「Alt」キーを「Meta」キーの代わりに用います。また、一般的には、「Esc」キーをポンと押してから指を離して「x」キーを押す操作で代用することの方が多ようです。

それでは、emacs.txt という名前が付いたバッファのデータをファイルに保存しましょう。ファイルに保存する命令は、C-x C-s です。この場合、C-s の s は save (保存する) という意味です。C-x C-s という命令を与えるとエコー領域に

```
Wrote /home/knoppix/tutorial/emacs.txt
```

というメッセージが出ます。これでメモリー上のバッファのデータがハードディスクに書き込まれました。

一旦、Emacs を終了します。終了の命令は C-x C-c でした。

それではファイルが作成されているか確認します。

```
knoppix@Knoppix:~/tutorial$ ls emacs.txt
emacs.txt
```

ファイルの大きさを調べます。ファイルの大きさが 26byte であることを確認してください。

```
knoppix@Knoppix:~/tutorial$ ls -l emacs.txt
合計 4
-rw-r--r-- 1 knoppix knoppix 26 2009-09-14 10:34 emacs.txt
```

次にファイルの内容を表示しましょう。

```
knoppix@Knoppix:~/tutorial$ cat emacs.txt
Emacs is a screen editor.
```

ここで、コマンド od を用いてみましょう。od は octal dump の省略形です。octal とは 8 進法のこと、dump とは、本来は「投げ捨てる、放り出す」などの意味ですが、コンピュータ用語では記憶されたデータを出力するという意味になります。まずは以下の表で普段使い慣れている 10 進法との対応を確認しましょう。2 進法は {0, 1}, 8 進法は {0, 1, 2, 3, 4, 5, 6, 7}, 10 進法は {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, 16 進法は {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f} という文字で数字を表す方法です。

2 進法	8 進法	10 進法	16 進法
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	a
1011	13	11	b
1100	14	12	c
1101	15	13	d
1110	16	14	e
1111	17	15	f

od はオプションを付けることでファイルの内容を 8 進法だけでなく 16 進法でも表示することができます。
(下の表示では、メモを書き込めるように行を空けています。)

```
[sm090001@pcs1p001-( ~/tutorial )-1008]od -Ad -tx1 emacs.txt
0000000 45 6d 61 63 73 20 69 73 20 61 20 73 63 72 65 65

0000016 6e 20 65 64 69 74 6f 72 2e 0a

0000026
```

ここで表示されているのは、ハードディスク装置に ASCII コード と呼ばれる形式で記録されている情報です。ファイル名 `emacs.txt` は、ファイル自身には含まれていないことに注意してください。左側の 6 桁の数字は保存されているデータの 1byte 単位のアドレスを 10 進法 で表しています。つまりデータに対して 0000000 から 0000025 までのアドレスが振られているということです。さらにデータは 16 進法で表されています。45='E', 6d='m', 61='a', 63='c', 73='s'... と対応しているのがわかるでしょうか? 16 進法で記述された 2 桁の数字 1 個が 1byte に相当します。全部で 26 個あることを確認してください。また、ところどころに空白を表す 20=' ' や最後が 0a='LF'、つまり「改行」を表す制御文字で終了していることにも注意してください。

2.2 起動時に引数でファイル名を指定

コマンド `emacs` は引数としてファイル名を指定すると、指定されたファイル名のファイルを編集することができます。

```
[sm090001@pcs1p001-( ~/tutorial )-1009]emacs emacs.txt
```

Emacs はファイルを開くと、ファイルの内容を表示します。

```
Emacs is a screen editor.
```

2.3 テキストの削除、挿入

それでは、screen という単語を text に修正しましょう。ファイルを開いたとき、行頭の E の字が点滅しているはずですが、キーボードで文字を入力すると、この位置に文字が表示されます。この点滅している部分を カーソル(cursor) と呼びます。カーソルは、コントロールキーを利用して移動することができます。C-f を用いて screen の n の字の次の位置まで移動したら、BS キーで screen という文字を削除し、text と入力してください。バッファが修正されて、以下ようになります。

```
Emacs is a text editor.
```

この時点でカーソルは text という単語末の t の字の次の位置にあるはずですが。

2.4 複数行の編集

今度は複数行の編集をしましょう。C-n で次の行に移動します。editor. という文字の後ろには改行の制御文字がありますので、カーソルキーの移動により、2 行目に入力することができます。

```
Emacs is a text editor.  
Emacs is a computer environment.
```

Emacs では編集機能の多くは Emacs-Lisp というプログラミング言語の一種によって実現されています。文字の入力やカーソルの移動などの編集操作は対応する Lisp 関数 を呼び出すことによって実現されています。関数名を指定して Lisp 関数を呼び出すには M-x を使います。M-x と入力すると、エコー領域にミニバッファが開いて、関数名の入力を促します。

```
M-x
```

例えば、M-x forward-char と入力すると、カーソルが次の文字へ移動します。

しかし、カーソルの移動のために、いちいち関数名を入力していたのでは面倒なので、良く使う関数に対してはキー入力が割り当てられています。「キーに Lisp 関数を結びつける」という意味で、このようなキー割り当ての事を キーバインディング(key binding) と言います。コンピュータの入力において効率を求めるのであれば、キーボードからは手を離さない方が良いことは明白です。そこで、Emacs では上の表のようなキーバインディングでカーソルを移動させることができます。

これまでに紹介したファイル操作も、それぞれ Lisp 関数をキーに結び付けたものです。

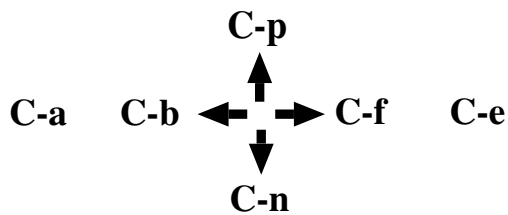


図2 Emacs のカーソル移動

キー操作	意味	関数
C-f	次の文字へ	forward-char
C-b	前の文字へ	backward-char
C-p	前の行へ	previous-line
C-n	次の行へ	next-line
C-a	行頭へ	beginning-of-line
C-e	行末へ	end-of-line

キー操作	意味	関数
C-x C-c	Emacs の終了	save-buffers-kill-emacs
C-x C-f	ファイルを開く	find-file
C-x C-s	ファイルの保存	save-buffer
M-x	Lisp 関数の呼出し	execute-extended-command

練習 1 先ほどの文章に続けて以下の文章を入力してください。

```
Emacs is a text editor.
Emacs is a computer environment.

At its core is an interpreter for Emacs Lisp,
a dialect of the Lisp programming language
with extensions to support text editing.
```

ここまで入力したら、ファイルを保存します。ファイルを保存する命令は、C-x C-s です。この命令も重要ですので、必ず忘れないようにしてください。

練習 2 C-p 等でカーソルを動かして 3 行目に 1 行挿入して

```
Emacs is a text editor.
Emacs is a computer environment.
Emacs is the extensible, customizable, self-documenting, real-time display editor.
At its core is an interpreter for Emacs Lisp,
a dialect of the Lisp programming language
with extensions to support text editing.
```

としてください。

ここでモードラインに注目してみましょう。モードラインには現在の行番号が表示されています。

練習 3 カーソルを上下に動かして行番号を確認しなさい。

Emacs では全ての編集操作にキーが割り当てられているわけではありません。例えば、行番号を表示する Lisp 関数は “what-line” ですので、M-x what-line と入力すると、エコー領域に行番号が表示されます。ま

た、行番号を指定して移動するには、“goto-line” という Lisp 関数を利用します。実際に

```
M-x goto-line
```

を実行すると、エコー領域にミニバッファ

```
Goto line:
```

が表示されますので、行番号を数字で入力して Enter キーを押せば、任意の行に移動することができます。

練習 4 Lisp 関数 goto-line を利用して、5 行目に移動しなさい。また、Lisp 関数 what-line を利用して、行番号をエコー領域に表示しなさい。

練習 5 ファイルを保存せずに、終了の命令 C-x C-c を実行しなさい。どのようなメッセージがエコー領域に表示されますか?

おそらく、

```
Save file /home/knoppix/tutorial/emacs.txt? (y, n, !, ., q, C-r or C-h)
```

と表示されます。このメッセージは、変更されているにも係わらず保存されていないファイル emacs.txt が存在するためです。従って、変更を保存する場合は y、変更を保存しない場合は n を入力します。ここでは y を入力して保存します。

2.5 削除ふたたび

再度、テキストファイル emacs.txt を Emacs で開きます。

```
knoppix@KNOPPIX:~/tutorial$ emacs emacs.txt &
```

命令の最後に & を付けると、バックグラウンドジョブで Emacs を実行できます。つまり、コンソールをそのまま使うことができるわけです。カーソルの位置は、最初の行の先頭の文字、「E」にあるはずですが、カーソル位置の文字を削除するには、Emacs の命令 C-d を用います。C-d を 1 回だけ押して、「E」の字を 1 字だけ消してください。

練習 6 C-d をさらに 4 回押して Emacs という文字を消しなさい。

1 行全てを削除したいときに、その回数分だけ C-d や BS を押すのは面倒です。カーソル位置から行末までの削除命令として、C-k が用意されています。ここで、C-k はカーソル位置からの削除であることに注意してください。行頭への移動 C-a と組み合わせることで、迅速に行を消すことができます。

キー操作	意味	関数
BS	カーソル位置の前の文字を削除	backward-delete-char
C-d	カーソル位置の文字の削除	delete-char
C-k	カーソル位置から行末までの削除	kill-line

それでは、複数行に渡って削除するにはどうしたら良いのでしょうか。

2.6 ポイント/マーク/リージョン

Emacs では、編集を行う際、ポイント(point)、マーク(mark)、リージョン(region) という 3 つの概念を利用します。

概念	意味
ポイント	現在のカーソル位置と前の文字との間
マーク	ユーザがバッファに 1 つだけ記憶させることができるポイント
リージョン	ポイントとマークで挟まれた部分 (矩形とは限らない)

下記のドキュメントを例に説明します。今、カーソル位置が最初の不定冠詞 “a” にあるとします。

```
Emacs is a computer environment.  
Emacs is the extensible, customizable, self-documenting, real-time display editor.
```

このとき、ポイントは空白 “ ” と “a” の間にあります。ここで、マークを設定します。マークを設定する命令は、C-SPC です。SPC はスペースキーを表しています。

カーソルを移動して、定冠詞 the の “t” まで移動します。

```
Emacs is a computer environment.  
Emacs is t he extensible, customizable, self-documenting, real-time display  
editor.
```

このとき、リージョンに含まれるのは、最後の空白も含めて以下の部分となります。

```
    a computer environment.  
Emacs is
```

Emacs では、標準の状態ではマークやリージョンは目に見える形では表示されません。教科書のコラムでは、マーク位置を確認する方法の一つとして C-x C-x を 2 度入力する方法が紹介されています。その他の方法としては、transient-mark-mode という Lisp 関数が存在し、リージョンの表示モードの On, Off を行うことができます。

リージョンを確認したら、削除してみましょう。リージョンの削除は、C-w という命令で行います。削除されたリージョンは 削除リング(kill ring) と呼ばれる領域に保存されます。削除リングに保存されたリージョンは、C-y という命令でカーソル位置に挿入できます。また、リージョンを削除するのではなく、削除リングに複製する命令 M-w もあります。

一般に C-w や C-k による文字列の削除を カット(cut)、M-w による文字列の複製を コピー(copy)、C-y による文字列の挿入を ペースト(paste) と呼び、テキストの移動を「カット&ペースト」、テキストの複製を「コピー&ペースト」と呼びます。

ところで、削除リングは、その名のとおりリング (輪) 状に削除された文字列が記憶されています。C-y の後に M-y を実行することで、順次古い削除内容を呼び出すことができます。

キー操作	意味	関数
C-w	リージョンを削除	kill-region
M-w	リージョンを削除リングに複製	kill-ring-save
C-y	削除リングのリージョンをカーソル位置に挿入	yank
M-y	削除リング内の古いデータを順番に呼び出す	yank-pop

3 編集操作のキャンセル

Emacs を使っている最中にキー操作を間違えたり、関数呼び出しを間違えたりということがあります。また、操作によっては、ミニバッファにメッセージが表示されて意味が理解できないということがあるかもしれません。その場合、操作をキャンセルする命令を行ってください。

操作をキャンセルする方法は C-g です。大変重要な命令です。場合によっては、何度か C-g を実行しないといけないかもしれませんので、注意しておいてください。

4 その他

今回紹介した機能の他に必要と思われる機能を表にまとめておきます。最近の Emacs はマウスでも操作できますが、キーボードからの操作を使いこなせれば、使いこなせるほど、今後の実習の効率が上がりますので是非挑戦してみてください。

キー操作	意味	関数
C-x i	ファイルの内容をバッファに挿入	insert-buffer
C-x C-w	ファイル名を指定して保存	write-file
C-s	前方逐次検索 (前方インクリメンタルサーチ)	isearch-forward
C-r	後方逐次検索 (後方インクリメンタルサーチ)	isearch-backward
M-%	置換	query-replace
C-x o	カーソルを他のウィンドウに移動	other-window
C-x u	Undo(編集取消)	advertised-undo
C-_	Undo(編集取消)	undo

また、Emacs 上で M-x help-with-tutorial を実行すると Emacs の簡単な案内が表示されます。こちらも一度読んでみると良いでしょう。

5 日本語の入力

たぶんスクールでは必要ありませんが、最後に Emacs 上で日本語入力を切替える方法を紹介しておきます。Emacs 上で日本語入力を切替えるには C-¥ を用いてください。

参考文献

[1] 新 The UNIX Super Text [上] 改訂増補版, 山口和紀 他著, 技術評論社