

Meadow について

プログラミング演習 I 担当 時井, 松村

1 エディタ Meadow の紹介

テキストエディタ (text editor) によるプレインテキスト (plain text) の作成方法を学ぶ。プレインテキストとは、フォントを大きくしたり小さくしたり、下線を引いたりといった装飾をしない、文字だけからなるテキストのことである。テキストエディタとは、文字通りプレインテキストを編集するツール (もの) である。「メモ帳」を使って日本語入力を行ったことを思い出して欲しい。「メモ帳」もテキストエディタの一つである。ここでは、プログラミング環境として有益な機能を持つ Meadow について紹介する。Meadow は UNIX 系の OS で利用されている Emacs をベースとしたエディタである。プログラミング演習では Meadow を使用してプログラムを入力することを前提とする。

2 基本的な使用方法

2.1 Meadow の起動と終了

起動は、スタートメニューから行なう。図 1 のように [スタート]-[すべてのプログラム]-[Meadow] と選択する。

Meadow を起動すると、図 2 のように別のウィンドウが開くと思う。これが Meadow のウィンドウである。「メモ帳」を起動したときのことを思い出して欲しい。

Meadow のウィンドウの上の広い部分が実際にテキストを編集する画面である。これをバッファと呼ぶ。言葉の意味は 5.1 で説明する。また、最下行のことをミニバッファと呼ぶ。ここには Meadow

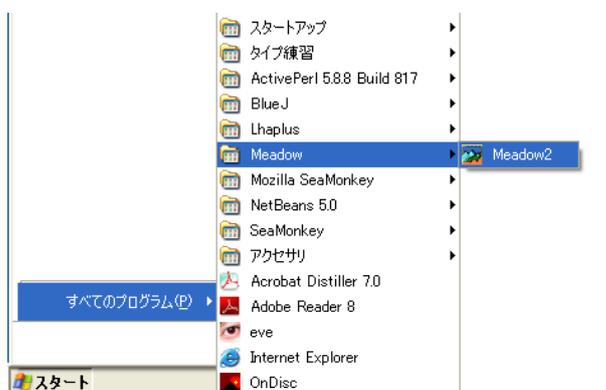


図 1: Meadow の起動

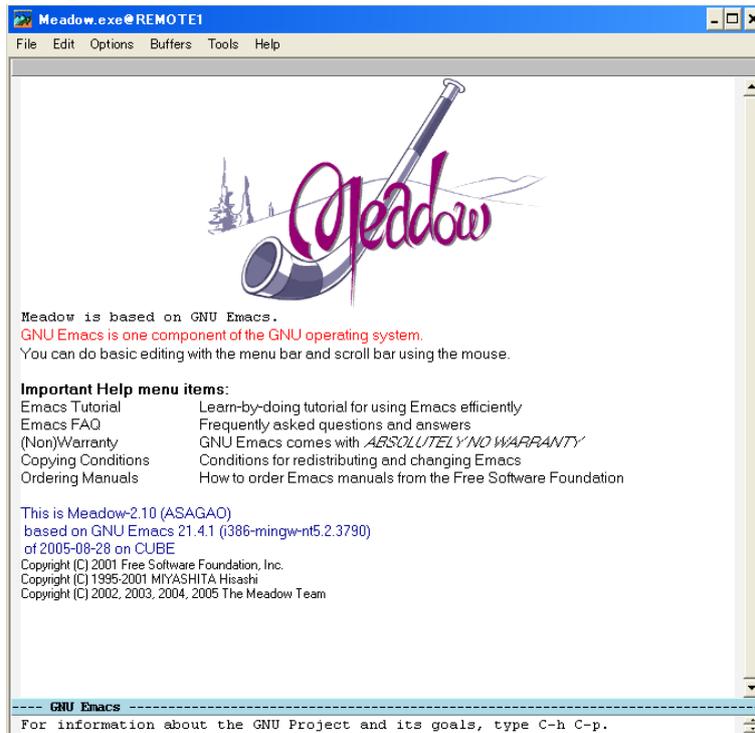


図 2: Meadow 起動時のウィンドウ

からのさまざまなメッセージが表示される。最下行のすぐ上の (反転している) 行はステータス行と呼ばれる。ここにはエディタの状態が示される。

Meadow では指示を与えるのにメニューによる (マウスを使う) 方法とキーボードのみによる方法とがある。メニューによる方法は、直感的であり一般的な他の Windows のソフトウェアとほとんど操作は変わらないので、取っ付きやすいという利点がある。一方、キーボードのみによる方法は、キーの組合せを覚える必要があるため敷居が高いが、操作を素早くできるということや、文字しか利用できない端末エミュレータを利用してリモートで作業をする場合などにも利用できるという利点がある。

このテキストでは、操作に不慣れであろうキーボードのみによる操作方法について主に説明する。キーボードのみによる方法では、コントロールキーとメタキーを多用する。コントロールキーは `Ctrl` と書かれたキーのことである。メタキーとは一般的なキーボードの場合、`Alt` と書かれたキーのことである。以降、次の表記を多用するので覚えておいてほしい。

`C-x` コントロールキーを押しながら `x` を入力する。

`M-x` メタキーを押しながら `x` を入力する。

ただし、キーボードによってはメタキーがない場合がある。そのような時には `Esc` (エスケープ) キーが使われる。このキーは通常、キーボードの左上の方にある。Esc キーを使う場合の `M-x` の入力方法は、次のようにする。キー操作が少し異なっているので慣れるまでは注意が必要である。

`M-x` エスケープキーを一度押し離す。そうすると画面最下行に `ESC-` と出てくるので、この段階で `x` を押す。ミニバッファに `M-x` という表示が出れば成功である。



図 3: メニューによる Meadow の終了

ESC キーはさらに、C-[で置き換えが出来る。すなわち、M-x は C-[x と入力すれば良いことになる。この方法を利用すると、コマンド操作はすべてコントロールキーを利用することで可能となる。これが便利と思うかどうかは好みの問題なので各自で好きな方法をとればよいであろう。

さて、Meadow の終了方法であるが、メニューによる方法では、[File]-[Exit Emacs (C-x C-c)] を選択する(図 3)。一方、キーボードによる方法では、Meadow の画面中で C-x C-c と入力する (C-x とやって続けて C-c とする。この操作は、コントロールキーを押したまま、x、c と順に押してもよい)。もし、なんらかのファイルが編集集中で保存されていないならば、保存するかどうかを y か n で答える。n と答えた場合には、保存せずに終了してよいかどうかを再度聞かれるので、yes か no で答える。

なお、メニューを見て分かる通り、メニューの各操作と同等のキーボード操作が表示してある。

演習

1. Meadow を起動してみよ。
2. Meadow を終了させよ。その際、メニューによる方法とキーボードによる方法と両方試してみよ。

2.2 使い方 – チュートリアル –

Meadow にはチュートリアルがついている。これを読めば一通りの Meadow の操作方法が分かるというものである。チュートリアルの起動方法は C-h t とする (メニューからの場合は [Help]-[Emacs Tutorial] である)。

そうすると上の方に「あなたが現在見ているのは Emacs 入門ガイドです。」と表示された画面になる。ここを読んでいくと Meadow の基本的な使い方が理解できる。

もしまちがって変な状態になってしまったら、C-g を (場合によっては数回) 入力する。最下行に Quit と出ればもとの状態に戻ったことになる。

Meadow における基本的な操作のいくつかを表 1 に示しておくので、適宜参照されたい。

表 1: Meadow における基本的なキー操作

キー	機能
基本操作	
C-x C-c	終了
C-g	わけがわからなくなったとき
カーソルの移動	
C-p	上の行へ移動
C-n	下の行へ移動
C-f	一文字右へ移動
C-b	一文字左へ移動
C-a	行頭へ移動
C-e	行末へ移動
C-v	一画面分先に移動
M-v	一画面分前に移動
M-<	ファイルの先頭へ移動
M->	ファイルの末尾へ移動
文字の削除, カットアンドペースト	
C-d	カーソルがある文字を削除
C-k	カーソル位置から行末まで削除 (行末の改行コードは含まない)
C-y	C-k で切り取った部分をペースト
C-x u	取り消し (undo)
ファイル操作	
C-x C-f	バッファ(ファイル)を開く
C-x C-s	そのバッファの内容を保存
C-x C-S	すべてのバッファの内容を保存
C-x C-w	そのバッファを名前を付けて保存
C-x i	カーソル位置に (別の) ファイルを挿入

注) C-x: コントロールキーを押しながら x を入力する。

M-x: メタキー (Alt) を押しながら x を入力する。

演習

1. Meadow を再び起動せよ。
2. C-h と打ち、しばらく待つ。待っていると最下行に C-h (Type ? for further options)- と出てくる。これはヘルプモードに入ったことを表す。チュートリアルを出すために t を入力する (メニューの場合は [Help]-[Emacs Tutorial])

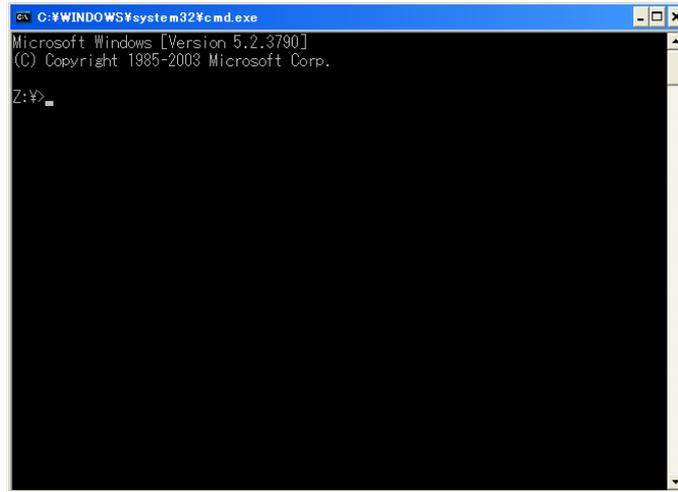


図 4: コマンドプロンプトのウィンドウ

- しばらくチュートリアルにしたがい Meadow の操作を学ぼう。このファイルは、好きなように編集して構わない。

3 ファイル操作

3.1 コマンドプロンプトとコマンドの実行

ファイル操作に先立って、これから作成するファイルを保存したりする作業用のフォルダを作成する方法を説明する。

作業フォルダは次のようにして作成する。まず、コマンドプロンプトを開く。これには、[スタート]-[すべてのプログラム]-[Accessories]-[コマンド プロンプト] を選択すればよい。すると、図 4 のようなウィンドウがあらわれる。

このウィンドウ内のコマンドプロンプト (Z:¥> のこと) に続けてコマンド (文字列) を入力し、Enter を押すことで、いろいろな操作を実行できる。

ここでは、フォルダを作成する (mkdir)、フォルダに移動する (cd)、フォルダの中のファイルのリストを表示する (dir) を紹介する。

フォルダを作成するには、mkdir を使う。以下のように、入力すると progI というフォルダが作成される。

```
Z:¥>mkdir progI      <===== progI フォルダを作成する
```

作成したフォルダへ移動するには、cd を使う。以下のように、入力すると progI というフォルダへ移動できる。progI に移動したことを示すように、コマンドプロンプトが変更される。

```
Z:¥>cd progI      <===== progI フォルダへ移動する  
Z:¥progI>        <===== コマンドプロンプトが変わった
```

フォルダにあるファイルリストを表示するには、`dir` を使う。以下のように、入力すると現在いる `progI` というフォルダのファイルのリストが表示される。今の流れでは作成したばかりなので、ファイルは存在しない。

```
Z:¥progI>dir <===== ファイルのリストを表示する
ドライブ Z のボリューム ラベルは HOME です
ボリューム シリアル番号は 0102-7E5A です

Z:¥progI のディレクトリ

2009/06/16  20:14    <DIR>          .
2009/06/16  15:28    <DIR>          ..
                0 個のファイル                0 バイト
                2 個のディレクトリ  1,233,256,448 バイトの空き領域

Z:¥progI>
```

演習

1. 作業用フォルダを作成する。

プログラミング演習では、作業用フォルダとして `Z:¥progI` や `Z:¥progII` を作成してプログラムのファイルを置くようにするため、あらかじめこれらを作成しておく。なお、`I` は `i` の大文字、`II` は `i` の大文字を 2 つ並べる。記号としてのローマ数字を使ってはならない。

2. 作業用フォルダ `progI` へ移動する。
3. 作業用フォルダにファイルが何も無いことを確認する。

3.2 Meadow によるファイル操作

ここでは、Meadow を使ったファイルの作成・保存・再利用の方法について説明する。Meadow を起動したら、最初にファイルを開く必要がある。ファイルのオープンには `C-x C-f` と入力する。ミニバッファには

```
Find file: ~/bin/
```

のように表示される。なお、`~/bin/` の部分はその時によって異なる。

先ほど作成したディレクトリ以下にファイルを作成するため、`~/bin/` を修正して、`z:/progI/test.txt` などとする。入力したファイル名のファイル (ここでは `test.txt`) が存在する場合は、そのファイルが開き、存在しない場合には、新しいファイルとして開く (メニューの場合は `[File]-[Open File...]` で適切なフォルダに移動して、ファイル名を指定する。)

ファイルの保存は `C-x C-s` を使う。

演習

1. Meadow を起動する。起動したら、上記で作成した `Z:¥progI` フォルダに `ex1.txt` というファイルを作成する。
2. 適当な文字列を数行、打ち込む。打ち込みが終わったら、`C-x C-s` でその内容をファイルに保存する。そして、`C-x C-c` で終了させる。
3. もう一度、Meadow を起動する。さっき編集したファイルを読み込む。さらになにか数行文字列を付け足す。その後、`C-x C-c` で終了させる。今度は編集内容を保存するか聞いてくるので、`y` を入力する。
4. さらにもう一度、Meadow を起動し、先ほどのファイルをオープンする。何か適当に書き足す。今度はこの内容を別のファイル名で保存する。このためには `C-x C-w` を入力する。そうすると画面最下行でファイル名を聞いてくるので `ex2.txt` と打ち込む。そして、`C-x C-c` で Meadow を終了する。
5. もう一度、Meadow を起動しよう。`C-x C-f` と打ち込むと編集したいファイル名を聞いてくるので、`progI/ex3.txt` と打ち込む。(注意: `I` は `i` の大文字) ここでは新規のファイルなので画面には何も出てはこない。ここに `ex1.txt` の内容を取り込んでみる。`C-x i` と打つと取り込みたいファイルの名前を聞いてくるので `progI/ex1.txt` と打ち込む。ここで、`C-x` の次はただの `i` で、`C-i` ではないので注意すること。
次に、この状態でさらに今読み込んだものの後ろに `ex2.txt` の内容を取り込む。まず、`M->` としてバッファの最後に移動する。(`>` や `<` の入力には `shift` キーを使う必要があることに注意。) バッファ内のカーソルが最下行にあることを確認したら、`C-x i` とやって、`ex2.txt` を取り込もう。うまくいったら `C-x C-c` を打ち込み、保存して終了する。
6. Meadow を起動し、`hello.rb` というファイルを開き、次を入力する。(ここで、`\` はキーボード上では `¥` と書かれたキーを使って入力する。)

```
print("Hello, Ruby World !\n")
```

`C-x C-s` を打ち込み、保存する。保存場所は `Z:¥progI` とする。

7. `ruby hello.rb` を実行する。まず、コマンドプロンプトを起動する。以下のようにして、`progI` フォルダへ移動し、プログラムを実行する。

```
Z:¥>cd progI          <===== progI フォルダへ移動する
Z:¥progI>ruby hello.rb
.....                <===== どうなる?
```

もし、なんらかの(エラー)メッセージが出た場合には打ち間違いがないかを確認して修正したあとに、再度実行する。

4 日本語の扱い

Meadow で日本語を扱う方法について述べる。

4.1 日本語入力

日本語入力モードへの切り替えは標準では C-\ または「全角/半角」キーである。(ここで、\ はキーボード上では ¥ と書かれたキーである。) このキー入力があるとステータス行の左の [--]S** の部分が [あ]S** のように変わる。この状態でローマ字入力のモードになっているので何か文字を入力することができる。変換はスペースキーで行う。

変換候補の選択等はメモ帳や Microsoft Word での操作方法と同じである。

演習

1. 上記のように Meadow で日本語入力モードにし、日本語の入力を行ってみよ。
2. 日本語の文字列の検索は C-s で Enter キーを押す。検索したい文字列を入力し、Enter キーを押す。再度同じ文字列で検索したい場合は、C-s を 2 度入力する。以後、連続して C-s を入力する間は、同じ文字列の検索を続けられる。

日本語チュートリアルを再び出して、その中で適当に日本語のサーチをしてみよ。

なお、インクリメンタルサーチは日本語入力には対応していない。

5 バッファとウィンドウ

5.1 複数ファイルの切り替え

Meadow は、一度に複数のファイルを扱うことができる。その説明をする前にまず、ファイルとバッファについて説明しておこう。

Meadow ではバッファという言葉が頻繁に出てくる。ファイルとは計算機のハードディスクなどに保存されているものをさす。バッファは、あるファイルの内容を編集するために計算機のメモリ上に読み込んだものをさす。つまり、テキストエディタは、ファイルの中身を読み込んで—すなわち、ファイルの中身をメモリ上にコピーして、ユーザによる編集操作に基づいてそのコピーを変更し、「保存(あるいは書き出し)」するコマンドが指示されたときに初めて、ファイルの内容を書き直すのである。したがって、どんなに編集作業を行っても「保存」作業が行われなければ、ファイル自体には変化が起こらない。C-x C-f を行うと、ファイルの内容を編集するためのバッファが Meadow 上に新たに作られるのである。

いま、C-x C-f でファイル(testA.txt としておこう)のためにバッファを開いたとする。そこでもうひとつのファイルも開きたいと思ったとしよう。このときは、同様に C-x C-f でもうひとつのファイルを開けばよい(こちらは testB.txt にしよう)。これによって、testA.txt を編集集中に testB.txt の一部を切り取り(あるいはコピーし)、testA.txt に貼り付けることもできる。

C-x C-f で複数のバッファを開くと、画面には最後に開いたバッファの内容しか表示されていない。せっかくいくつものバッファを開いたので、開いている別のバッファの内容を見たいのは当然であろう。このため、バッファの切り替えをする必要がある。

バッファの一覧表示は C-x C-b で行う。このキーが入力されると画面が分割され、例えば以下のようなバッファ一覧が現れる。

MR Buffer	Size	Mode	File
-- -----	----	----	----
. testB.txt	0	Text	z:/progI/testB.txt
testA.txt	0	Text	z:/progI/testA.txt
scratch	182	Lisp Interaction	
* *Messages*	136	Fundamental	
;% *Buffer List*	290	Buffer Menu	

ここで、Buffer の欄に表示されているのがバッファの名前であり、File の欄に表示されているのが実際のファイル名である。上の例では、hoge hoge ディレクトリ内の testA.txt と testB.txt をバッファに読み込んで作業を行なっている¹。

一覧を見て切り替えたいバッファの名前が分かったら、今度は C-x b とする。すると、ミニバッファに Switch to buffer: (default testA.txt) のように表示される。(testA.txt の部分は違う場合もある。) もし、切り替えたいバッファの名前が default の後に表示されていたら(例の場合 testA.txt に切り替えたいのなら)、単に Enter キーを押すだけで良い。そうでなければ、バッファの名前を入力して Enter キーを押す。これで、上の画面が新しいバッファに切り替わる。

なお、二つに分かれた画面を一つに戻すには C-x 1 とする。このような画面の分割に関する操作については次の節で説明する。

演習

1. C-x C-f などを利用し、二つ以上のファイルを読みこみ、C-x C-b を用いてバッファ一覧を表示せよ。
2. 次に C-x b でファイルの切り替えをやってみよ。ファイルの中身を変更する必要はない。
3. 現在、画面上にあるバッファに適当に文字列を書き込み、それをコピーしてもうひとつのバッファに移してみよう。例として、ここでは次のような一行を書くことにする。

Meadow の使い方

この行の先頭にカーソルを合わせ(同一行内にカーソルがある場合は C-a)、C-k としてこの行をいったんカットし、C-y で元に戻す。これでいったんカットした内容を Meadow は覚えたことになる²。C-x b を実行して、別のバッファに切り替える。そのバッファで C-y を実行すればコピーは完了する。

これをいろいろと試してみよう。

¹*のついたバッファは編集するファイルではなく特殊なものである。

²なお、連続して C-k を使って連続した内容をまとめて覚えさせることもできる。

テキストの一行を「記憶する」ために「カットして、戻す」という操作を行うのは少々奇妙に思われるかもしれない。実は、文字どおり「記憶する」という操作も可能である。それには、「カーソルを行頭に合わせた後、C-スペースキーとし、次に、行末までカーソルを移動し、M-w とする」という手順を踏む。どちらが簡単か分かるであろう。

このように、実際やりたいことを概念通りにするより、別の手段で代用すると効率が良い場合がしばしばある。コンピュータに慣れると、そういう手段が身についてくるだろう。

5.2 画面の分割

Meadow では複数のファイルを扱えるのであるが、それを同時に眺めたい場合も出てくる。この場合、画面の分割の機能を使うのがよい。画面の分割の機能は C-x 2 で行うことができる。分割された一つの画面中でさらに分割するということもできる。

分割された画面間でカーソルを行き来させるには、C-x o (小文字のオー) を使う。分けた二つの画面を一つの画面にするには C-x 0 (数字のゼロ) を使う。Meadow 全体でひとつの画面だけの状態に戻すには C-x 1 を使う。

表 2: 画面切り替えのキー操作

キー	機能
C-x 2	画面を二分割する
C-x o (オー)	画面間のカーソル移動をする
C-x 0 (ゼロ)	現在カーソルのある画面を消し、分割した画面を統合する
C-x 1 (イチ)	Meadow 全体を現在カーソルのある一画面に戻す

演習

前節の演習をもう一度やってみよう。ただし、ここでは二つのバッファを開いたあと、その両方を画面で眺めるとする。手順は次のようになる。

1. C-x C-f で testA.txt を開く。(開くファイルは各自で勝手に決めてよい)
2. C-x C-f で testB.txt を開く。(同上)
3. この時点では画面上には testB.txt の内容が出ている。ここで C-x 2 により、画面を二分割する。
4. この時点でカーソルは上の画面上にあるのでここで C-x b を実行する。ミニバッファに testA.txt と入力し、Enter キーを押す。そうすると上の画面が testA.txt の内容にかわる。
5. testA.txt の方に「Meadow の使い方」と書き、その先頭にカーソルを移動して C-k と C-y を実行する。

6. C-x o で下の画面にカーソルを移動させる .

7. C-y でペーストする .

これをいろいろな内容でやってみよう .

6 種々の設定

Meadow は非常に多くの機能を備えている . Meadow を起動してから一々設定するのは大変なので , 設定をまとめてファイルに書いておくことができる . このような設定ファイルがあると Meadow は起動時にそのファイルを読みこみ設定をする . 設定ファイルはホームディレクトリにある .emacs という名前のファイルである .

さて , 設定ファイルの中身であるが Emacs Lisp というプログラミング言語で記述されている . 初心者にはなにがなんだかわからんしろものである . だが慣れてみると結構面白い . 参考文献としては [1] などがある .

Meadow(あるいは Emacs) 上ではメールやニュースの読み書き , Web ページの参照など , やろうと思えば , 何でもできてしまう . 興味のある人は , 参考文献を読んでみると良い . Meadow の参考文献として [2] , [3] を , Emacs の参考文献として [4] , [5] をあげておこう .

参考文献

- [1] Chassell, R. J. *Emacs Lisp プログラミング入門*. 大木敦雄訳. アスキー, 1998, 294p. (ISBN4-7561-1805-4)
- [2] 小関吉則. *入門 Meadow/Emacs*. オーム社, 2003, 280p. (ISBN4-2740-6518-9)
- [3] 松下晃久. *Meadow/Emacs スーパーチュートリアル*. 秀和システム, 2004, 303p. (ISBN4-7980-0894-X)
- [4] Cameron, D., Rosenblatt, B. and Raymond, E. *入門 GNU Emacs*. 福崎俊博訳. 第 2 版. オライリー・ジャパン, 1999, 580p. (ISBN4-9009-0083-4)
- [5] Stallman, R. M. *GNU Emacs マニュアル*. 赤池英夫訳. アスキー, 1999, 671p. (ISBN4-7561-3002-X)