

# Emacs の簡単な使い方

コンピュータの上での作業は基本的に「ファイルを作成する」ことにつきます。その中でも「テキストファイル」を作りだし、改変 (編集)するのが、「テキストエディタ」と呼ばれる種類のツールです。ここでは Emacs というテキストエディタの簡単な使い方について紹介しましょう。できれば実際に動かしながら読んで下さい。この文書は以下の節からなります。

- |                 |           |             |
|-----------------|-----------|-------------|
| 1 起動と終了         | 2 ファイルの作成 | 3 カーソル移動    |
| 4 操作の取消とコマンドの中断 | 5 ブロック操作  | 6 文字列の検索    |
| 7 バッファとウィンドウ    | 8 日本語入力   | 9 マウスを使った操作 |

## 1 起動と終了

とりあえず Emacs を起動してみましょう。下線で示されている部分を入力して、リターンキーを押します。

```
% emacs
```

図 1 のような画面が現れるはずですが、X-window 環境の場合は新しいウィンドウができて、図 1 のタイトルバーの下にメニューのついた画面が現れます<sup>1</sup>。これが Emacs の起動画面です。ここにはヘルプを見る方法やバージョンなどが書いてありますね。このメッセージは何かキー入力をするると消えてしまいます。

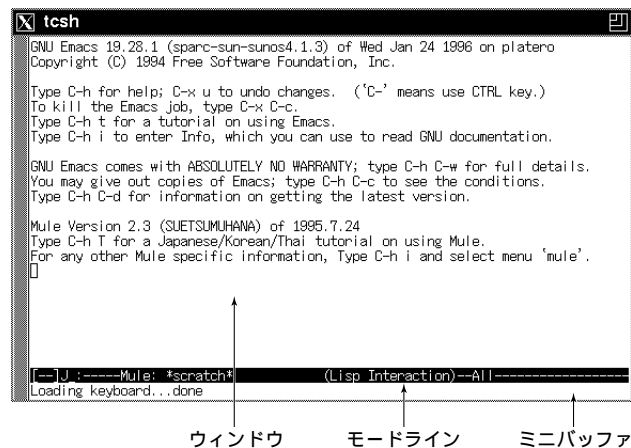


図 1: Emacs のスプラッシュ画面

画面の下の反転表示されている行をステータスラインあるいはモードラインといいます。一つのウィンドウに対して一つ存在して、そのウィンドウの現在の状況を表します。一番下の行はミニバッ

<sup>1</sup> % `nemacs -nw` とすると、x-window 環境でも (1) のような画面になります

ファとよばれ、Emacs に対して発行したコマンド入力が表示されたり、また対話的にコマンドを実行する時に使われます。

起動したばかりですが、終了してしまうことにします。先ほど Emacs と入力したのは unix のコマンドシェルに対する命令ですが、今度は Emacs に対する命令になります。Emacs に命令を発行する時には、主にコントロールキーやエスケープキーと文字キーを組み合わせで入力します。終了するためにはコントロールキーを押しながら x のキーを押す、さらに (コントロールキーから指を離さずに) C のキーを押します。この操作を C-x C-c と表現します。

以降、いろいろなコマンドが出てきますが、それらの説明の時に使う表記についてここにまとめておきます。

表記	キー操作の方法
C-x	コントロールキーを押しながら x を押す
C-x C-x	コントロールキーを押しながら x を押し、さらに x を押す
C-x x	コントロールキーを押しながら x を押し、コントロールキーを離してから x を押す
M-x	エスケープキーを押して離してから x を押す
M-x command	エスケープキーを押して離してから x を押し、続いて command という文字列をタイプする

[補足] C-x の C は “Control” を示し、M-x の M は “Meta” を示しています。Control キーはほとんどすべてのキーボードについていますが、Meta キーなるものは普通のキーボードにはありません。そこで「Meta キーを押しながら」の代替として「Esc キーを押して離してから」という操作が基本的にどんなシステムでも使えるようになっています。その他にも特定のキーを Meta キーとして割り当てているシステムもあります。例えば Sun のワークステーションのキーボードでは、キーが Meta キーとなっています。このようなシステムでも Esc キーによる Meta キー代替は共通に使うことができます。

## 2 ファイルの作成

前の節では Emacs をただ起動してみただけですが、本来は既にあるファイルを読み込ませて編集したり新しいファイルを作成したりするわけです。Emacs にファイルを読み込ませるにはコマンドラインで

```
% emacs filename
```

とタイプします。filename で指定したファイルが既に存在している時にはそのファイルが Emacs 読み込まれ、存在していない時には新しいファイルを作成することになります。試しに

```
% emacs test.text
```

としてみましょ。今度は始めから真っ白い画面が出てきます。左上に黒い四角が表示されていますね。これを簡単に「カーソル」と呼ぶことにします。キーボード上の文字キーをタイプすると、相当する文字がカーソルのところに入力されます。

タイプミスをした場合には、BackSpace キー<sup>2</sup> でカーソルの一つ前の文字を消すことができます。C-d でカーソルの置いてあるところの文字が消えます。

<sup>2</sup> Enter キーの上の、 とプリントしてあるキーです

何か書き込んでみたら、先ほどの Emacs 終了コマンド C-x C-c をしてみてください。さっきは何もいわずに終了しましたが、今回はミニバッファに何かメッセージを出してきているはず。基本的に Emacs からのメッセージは全部英語ですがちゃんと読みましょう。

```
Save file /home/user8/sumiya/test.text? (y, n, !, ., q, C-r or C-h)
```

「test.text というファイルを保存しますか?」と聞いていますね。それに対する答をメッセージの最後の ( ) の中から選ぶわけです。この場合保存しておきたければ y をタイプし、したくなければ n をタイプします。Emacs のコマンドでそのコマンドを発行した後に Emacs から何らかの問い合わせがある場合はほとんどこのような形になっています。( ) の中の選択肢の意味がわからない場合には C-h とするとヘルプが出ます。

これで新しく test.text という名前のファイルができました。再び

```
% emacs test.text
```

としてみてください。Emacs が起動して、さっき作ったファイルが表示されます。

さて、今 Emacs が起動している状態ですが、この状態でファイルを読み込むには C-x C-f とします。するとミニバッファにカーソルが移動し

```
Find file: ~/
```

と表示されます。~/ は自分のホームディレクトリという意味です。ここに読み込みたいファイル名をタイプし、リターンを押します。ファイル名をはっきりと覚えていない場合には、覚えているところまでタイプしてから Space キーを押すと、Emacs がディレクトリを調べてファイルの名前を教えてください (これをファイル名補完機能といいます)。例えば ~/te までタイプして Space キーを押すと自動的にミニバッファに ~/test.text と表示されるはず。もし te から始まるファイルが複数ある場合にはウィンドウが二つに分割され、相当するファイルがリストされます。

以下に、ファイルの読み込み・保存関係の命令をまとめておきます。

% emacs file	Emacs を起動し、file を読み込む (コマンドラインから)
C-x C-f	新しくファイルを読み込む (Emacs 内から)
C-x C-i	ファイルを読み込んでその内容をカーソル位置に挿入する
C-x C-r	ファイルを書き込み保護された状態で読み込む
C-x C-s	Emacs を終了せずにファイルの変更を保存する
C-x C-w	ファイルの名前を変えて保存する

### 3 カーソル移動

この節ではカーソル (画面に出ている黒い四角) の動かし方について述べます。まず適当なファイルを Emacs で開きます。login ファイルを開いてみましょう。

```
% emacs .login
```

とタイプします。すでに Emacs が起動している時は C-x C-f .login でしたね。

ファイルを開いた状態ではカーソルはファイルの一番先頭にいます。画面でいうと左上です。ここからカーソルを一文字先に (右に) 動かすには C-f とします。一文字後に (左に) 動かすには C-b です。C-f を何度か押して行の終りを越えると、カーソルは一つ下の行の一番先頭に移動します。一行下にカーソルを動かすには C-n とし、一行上に動かすには C-p とします。

ここまででてきた f, b, n, p はそれぞれ forward-char, backward-char, next-line, previous-line の頭文字をとっていると思われます。始めはなんと使いにくい割り当て方か、と思われそうですが、使っていると慣れてきて他のエディタでもこの割り当てにしたくなってしまうから不思議です。また、矢印キー<sup>3</sup> をカーソル移動に使える環境もあります。

カーソル移動のコマンドは対になっているので、それをまとめておきます。

C-f, C-b	一文字進む/戻る
C-n, C-p	一行進む/戻る
M-f, M-b	一単語進む/戻る
C-v, M-v	一画面進む/戻る
C-a, C-e	行の先頭へ/末尾へ
M-<, M->	ファイルの先頭へ/末尾へ

## 4 操作の取消とコマンドの中断

直前の操作を取り消す (undo する) には C-x u とします。間違っても必要な場所を消してしまった時には C-x u です。

実行中のコマンドを中断するには C-g とします。期せずしてキー反応しなくなってしまうとか、何だかよくわからなくなってしまう時には C-g とすると元に戻れることがあります。よく覚えておいて下さい。

C-x u	操作の取消
C-g	実行中のコマンドの中断

## 5 ブロック操作

ここでは既に入力されているテキストを動かしたり、まとめて削除したりする方法について述べます。これらの操作をする際に重要になるのが、「マーク」です。マークはテキスト内にユーザーが指定する点で、C-Space もしくは C-@ とすると、現在のカーソル位置にマークが置かれます。マークから現在のカーソル位置までを「リージョン」とよびます。図 2 を見て下さい。C-Space を打つとミニバッファにマークを設定した旨の表示がなされます (図 2-(1))。次にカーソルを指定したい部分の最後まで動かします。図 2-(2) のようにカーソルを動かしたとすると、図の点線で囲った部分がリージョンとなります。リージョンの始めは 'b' からですが、終りは 'w' の直前であることに注意してください。Emacs には、このリージョンに対して削除したり、移動したりする機能があります。

実際にやってみるとわかりますが、マークを置いてリージョンを設定しても、外見上は何も変化はありません。Emacs はマークを私達の目に見えるようにはしてくれないのです。基本的には自分でどこにマークを置いたかを憶えておかななくてはならないわけですが、マークの確認として使える機能にカーソルとマークを入れ替えるというものがあります (C-x C-x)。つまり C-x C-x でマークのある位置にカーソルを移動させ、もう一度 C-x C-x でカーソルをもとあった位置に戻す、という操作で現在のマーク位置 (とリージョン) を確認できるわけです。

テキストの編集でしばしば必要になるのはテキストのひとまとまり (ブロック) を 1. 削除, 2. 移動, 3. 複写することです。これらをリージョンを使って行うには次のようにします。

<sup>3</sup> とプリントしてあるキーのことです

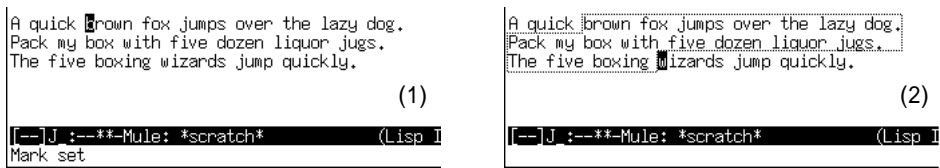


図 2: リージョンの設定

(1) 削除

- (1.1) 削除したい部分の先頭にカーソルを持っていき、C-Space とします (マークのセット)
- (1.2) 削除したい部分の最後にカーソルを持っていきます。この時点で、削除したい範囲は「リージョン」となっています。
- (1.3) C-w とします

(2) 移動

- (2.1) 移動したい部分の先頭にカーソルを持っていき、C-Space とします
- (2.2) 移動したい部分の最後にカーソルを持っていきます
- (2.3) C-w とします。ここまでは削除の時と同じですね。実は C-w で消えた部分はすっかりなくなってしまったわけではなくて、Emacs の内部の特別な場所 (キル・リングといいます) に移されて保存されています
- (2.4) ブロックを移動したい場所にカーソルを移動します。
- (2.5) C-y とします。移動した後も Emacs はブロックの内容を保持していますから、何度でも C-y できます。

(3) 複写

「移動」の (3) で C-w としたところを M-w とすると、リージョンは削除されずにその内容だけが Emacs の内部 (キル・リング) に記憶されます。そのあと C-y で好きなところに複写できます。

もうひとつ憶えておくくと便利なのが、C-k です。C-k とすると、現在のカーソル位置からその行の最後までを削除し、その内容をキル・リングに保存します。このコマンドは、現在のカーソル位置が行の最後か途中かで少しふるまいが違います。カーソルが行の途中にある場合は、行の最後まででのテキストを削除し改行コードは残しますが、カーソルが行の最後にある場合は改行コードを削除します (図参照)。どちらの場合も C-k で削除した行を後から C-y で他の所にコピーすることができます。

以下にブロック操作のコマンドをまとめます。

C-Space	現在のカーソル位置をマークとして設定する (以降マークからカーソル位置までがリージョンとなる)
C-x C-x	カーソルとマークを入れ替える
C-w	リージョンを削除し、キルリングに保存する
M-w	リージョンを削除しないで、キルリングに保存する
C-y	キルリングの内容をカーソル位置に挿入する
C-k	カーソルから行末までを削除し、キルリングに保存する

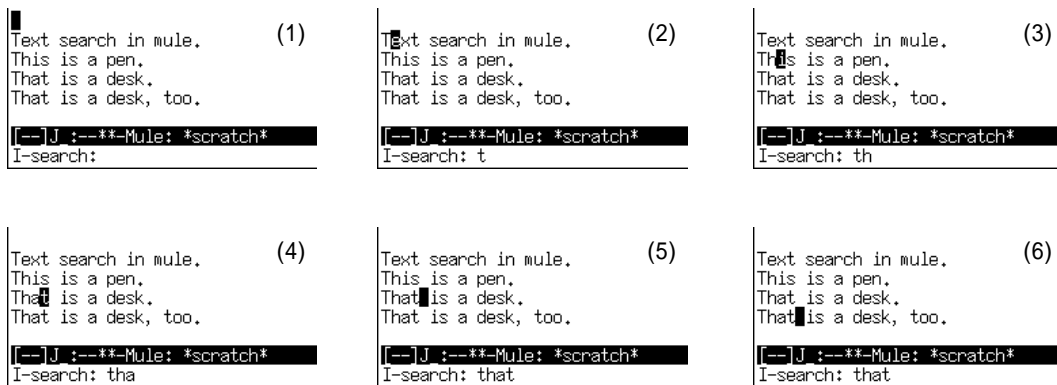


図 3: 文字列の検索

## 6 文字列の検索

文字列を検索するには、C-s とします。画面の一番下のミニバッファにカーソルが移り i-search: と表示されます (図 3-(1))。例えば “that” と検索したい場合はここでももろに ‘t’ とタイプします。するとテキストの中で ‘t’ が最初に出現するところにカーソルが移動します (図 3-(2))。C-s の検索では大文字小文字は区別されませんので、図では ‘Text’ の ‘T’ のところにカーソルが移動しています。さらに続けて ‘hat’ と入力すると、‘that’ のあるところまでカーソルが移動しますね (図 3-(5))。これで “that” がどこにあるのかを見つけることができました。この状態でさらに C-s とすると、次に ‘that’ が出現するところまでカーソルが移動します (図 3-(6))。このように繰り返して C-s することで、目的の場所を探すことができます。

検索を終了するには、Enter を押します<sup>4</sup>。この時ミニバッファに

```
Mark saved where search started
```

と表示されます<sup>5</sup>。「検索を開始した時点での (最初に C-s を打った時の) カーソル位置にマークを保存した」と言ってますね。4 節で見た C-x C-x で検索を始めた場所にカーソルを戻すことができるわけです。

この検索では、‘t’ ‘h’ ‘a’ ‘t’ と一文字打つ毎に検索が実行されてカーソルが動いていきます (図 3-(2) ~ (5))。これは例えば ‘th’ まで打った段階で目的とする文字列が見つければそれ以上打つ必要はないわけで、よく考えてある方法だと思いませんか？

テキスト中の “This” を “That” に変えたい、というような場合には置換機能を使います。M-% とすると、ミニバッファにカーソルが移動し、

```
Query replace:
```

と表示されます。変換したい文字列を入力し、Enter キーを押すと

```
Query replace This with:
```

<sup>4</sup> NEmacs では Escape を押す

<sup>5</sup> NEmacs では Mark set と表示されます

と表示が変わります。置き換えたい文字列を入力し Enter キーを押すと置換が始まります。条件にあう文字列を発見するたびに、Emacs は変換するかどうかを聞いてきます。変換したい場合は y または Space を、変換せずに飛ばして次の文字列にいきたい場合 n または Delete を押します。途中で変換操作をやめたい場合は q または Enter を押します。

C-s	検索 (C-s で次検索, Enter で終了)
M-%	置換 (Space で変換, Delete でスキップ, Enter で終了)

## 7 バッファとウィンドウ

いくつかのファイルを参考にして新しいファイルを作成したい、といったことはよくあります。このような場合、すべてのファイルに対して Emacs を起動する必要はありません。Emacs は幾つでもファイルを読み込んでそれらを並列に編集することができます。

Emacs に読み込んだファイルを編集しても、終わるときに保存しなければファイル自身はもとのままです。このことから想像できるように、Emacs で編集しているのはファイルそのものではなくて Emacs 内部にもっているテキストです。そのテキストが入る場所を「バッファ」といいます。Emacs にファイルを読み込むということは、そのファイルと同じ名前のバッファを作ってファイルの内容をコピーする、というように思えばよいでしょう。そのバッファはファイルの名前だけでなく場所なども憶えているので、たとえ同じ名前の (違うディレクトリにある) ファイルを幾つか読み込んだとしても C-x C-s すれば正しく元のファイルが変更されます。また、ファイルを読み込まずに Emacs を起動したときにはバッファの名前は \*scratch\* となります。Scratch というのは何もないスタート状態のような意味合いで、ここに何かを書き込むことはできますが、まだどんなファイルとも関連づけられていない状態です。このようにバッファにはディスク上に存在しているファイルと関係ないものもあります。

ウィンドウに表示されているバッファは一つですが、Emacs は内部にバッファを幾つでも持つことができます。C-x b でウィンドウに表示するバッファを切り替えて編集することができます。C-x C-b とすると、現在どんなバッファがあるかの一覧を見ることができます。

また、ウィンドウを複数作ることもできます。C-x 2 とすると、現在のウィンドウが水平に二つに分割され、両方のウィンドウにそれまで表示されていたバッファが表示されます。C-x 3 とすると縦に分割されます<sup>6</sup>。そこで C-x C-f (ファイルの読み込み) や C-x b (バッファの切り替え) をすると二つのファイルの内容を同時に見ることができます。分割されたウィンドウの間でカーソルを行き来させるには C-x o を使います。

図 4 は、ファイルとバッファ、そしてウィンドウの間の関係を示しています。

C-x C-b	バッファの一覧を表示
C-x b	バッファを切り替える
C-x k	バッファを削除する
C-x 0	現在カーソルがあるウィンドウを削除する
C-x 1	現在カーソルがあるもの以外のすべてのウィンドウを削除する
C-x 2	ウィンドウを横に分割する
C-x 3	ウィンドウを縦に分割する
C-x o	他のウィンドウにカーソルを移動させる

<sup>6</sup> NEmacs では C-x 5

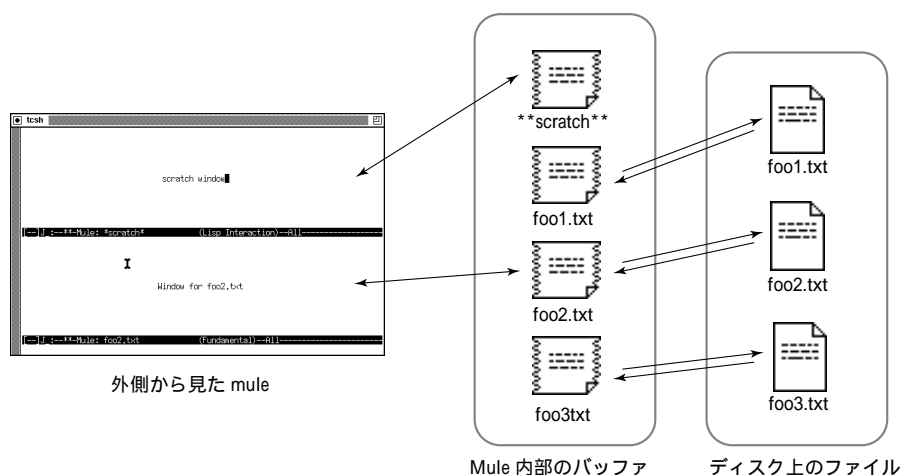


図 4: ファイル, バッファ, ウィンドウ

## 8 日本語入力

Emacs で日本語を入力するにはいくつかの方法があります。一つは SKK というシンプルな日本語入力方法です。これに関しては優れたチュートリアルを Emacs から呼び出して自習することができるので、そちらを参照して下さい。チュートリアルを呼び出すには、`M-x skk-tutorial` と入力します。

他にも「たまご」と呼ばれる Wnn のクライアントがありますし、「かな」を利用することもできます。

ここでは、Emacs 独特の日本語入力方法として、`yc` と呼ばれるものを使ってみましょう。`yc` は “Yet Another Canna Client” の超省略形とのことです。

`yc` を使うためには、`emacs` を起動した後に一つおまじないを唱えておく必要があります。`M-x load-library [RET] yc` として、`yc` を利用する準備をします。<sup>7</sup>

通常の日本語入力環境では、日本語入力モードと英語入力を切替えながら作業を進めますが、`yc` にはそういったモードがありません。`yc` で日本語を入力するには、まずローマ字を打ち込んでおいて、`C-j` とします。すると、打ち込んであったローマ字が一気に漢字へと変換されます。最初は戸惑うかも知れませんが、慣れて来ると英語混じりの日本語、や HTML 文書などを書くときは非常に便利です。

例えば

```
watahahirodainogakuseidesu.
```

とまずはタイプします。打ち終わったら、`C-j` としてみましょう。

```
|私は 比呂だ 伊野 学生です。|
```

と表示されますね。入力された文章「わたしはひろだいのがくせいです。」を適当な四つの文節に区切り、それぞれの文節に対して一つずつの変換候補を表示しています。区切り方が適切ではないため、

<sup>7</sup> `emacs` の初期設定ファイル (`.emacs`) に `(load-library "yc")` と書いておいても OK です。



おかしい変換結果になっていますね。今「私の」のところにカーソルがありますが、これはこの候補でよいかどうかを尋ねてきているわけです。Space または C-n で次の候補を、C-p で前の候補を表示します。変換する文節を移動するには、通常のカーソル移動と同じ C-f, C-b を使います。

今表示されているうちでは 2 番目の「比呂だ」というのが正しくない候補になっていますから、これを直しましょう。C-f を 1 回押してカーソルを「比呂だ」に移動します。ここでは文節の区切りが正しくありませんので、まずこれを直してみましょう。文節の区切りを短くするには C-i とします。長くするには C-o を使います。「ひろだいの」が一つの文節になるはずなので、C-o を二度打って文節を伸ばします。

「ひろだいの」が一つの文節になったら、Space を何度か叩いてみてください。「広大」とはなりませんね。これは「広大」と言う単語がシステムに登録されていないからです。

仕方がないので、C-i を三回押して、「ひろ」と「だいの」にしてみます。今度は大丈夫だと思います。正しく「広大の」と表示されたら、Enter キーを押して、漢字を確定します。

「広大」はこれから先よく使うでしょうから、これを個人辞書に登録して見ましょう。正しく変換された「広大」をリージョンとし、M-x yc-toroku-region と打ちます。するとミニバッファで「広大」に対する読みを入力するよう促されます。「ひろだい」と打ち込み、リターンキーを押すと次に品詞を聞いてきます。「固有名詞」とします。これで登録完了です。

C-j	日本語変換の開始
Space	次候補の表示
C-n	次候補の表示
C-p	前候補の表示
C-f	文節移動 (次へ)
C-b	文節移動 (前へ)
C-i	文節縮め
C-o	文節伸ばし
C-h	ひらがなに変換
C-k	カタカナに変換
M-x yc-toroku-region	(リージョンを指定した後) 単語を個人辞書に登録する

## 9 マウスを使った操作

X-window 環境で Emacs を起動すると図 5 のようなウィンドウが新たに作られます。非 x-window 環境のもの (図 1) にメニューバーとタイトルバーが加わった物になっています。Emacs の画面がこのようなになっている場合、今までの操作に加えてマウスを使った操作が可能になります。

カーソルの移動 カーソルを移動させたい場所にマウスポインタを持っていき、左クリックします。ウィンドウを分割している時にも、移動させたい場所を左クリックするだけです。

リージョンの指定 いくつかの方法があります。

- マウスポインタを設定したいブロックの先頭に持って行って左クリック。そして最後に持って行って右クリック。
- マウスポインタを設定したいブロックの先頭に持って行って、左クリック、そのままボタンを押したままでブロックの最後までマウスポインタを持って行ってボタンを離す。(この操作を「ドラッグ」と言います)

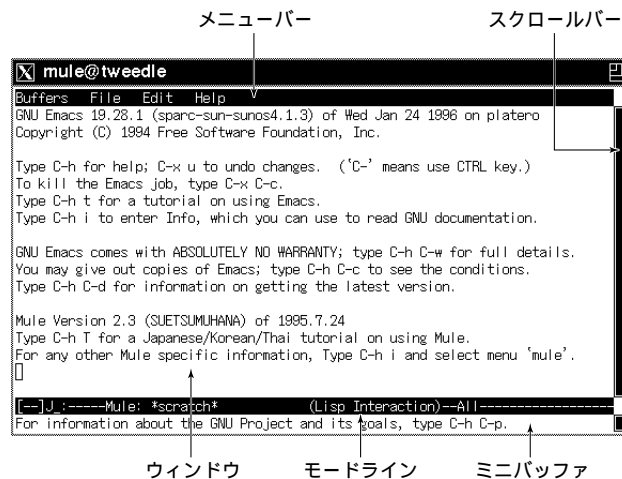


図 5: X-window 環境での Emacs のスプラッシュ画面

- 左ボタンをダブルクリック。これでマウスポインタがある位置の単語がリージョンとして設定されます。ダブルクリックしたあとドラッグすると、同じように単語単位でリージョンを広げることができます。
- 左ボタンをトリプルクリック。これでマウスポインタがある位置の行がリージョンとして設定されます。トリプルクリックしたあとドラッグすると、同じように単語単位でリージョンを広げることができます。

真中のボタンをクリックすると、C-y したときのようにリージョンが複写されます。C-y の場合と違ってカーソルを複写したい場所に持っていく必要はありません。マウスポインタを持って行って中クリックするだけです。

**スクロール** スクロールバーを左クリックするとウィンドウの内容が上にスクロールします。右クリックで下にスクロールします。このとき、スクロールバーの上からどの位の位置をクリックしたかによってスクロールする量が変わります。下にいけばいくほど多くスクロールします。中ボタンでドラッグすると、ポインタの移動に合わせてウィンドウの内容がスクロールします。

**ウィンドウ・バッファ操作** モードラインを左ボタンでドラッグするとウィンドウの高さを変えることができます。モードラインを右ボタンでクリックすると、そのウィンドウを消すことができます。コントロールキーを押しながらスクロールバー（モードライン）を中ボタンでクリックすると、その位置でウィンドウが上下（左右に）に分割されます。コントロールキーを押しながらウィンドウの中を左クリックすると、現在 Emacs が持っているバッファの一覧が現れます。

**メニューからコマンドを実行** メニューバーにある Buffers とか File とかの文字列をクリック（すぐにはボタンを離さない）するとメニューがポップアップします。基本的なコマンドをここで実行できます。

**候補の選択** C-x C-f でファイル読み込みをするときにファイル名補完機能を使うとウィンドウが分割されてファイルの一覧がでます。この一覧の中のファイル名を中クリックするとそのファイ

ルが読み込まれます。

C-x C-b でバッファの一覧表を出した時も、中クリックで切替えたいバッファを選ぶことができます。

マウスを使った操作は便利ですが、キーボードの操作の方が慣れると早かったりします。また x-window が使えない環境で Emacs を使わなければならない場合も多いので、キーボードの操作を最初に覚えてしまった方が良いでしょう。

## 10 おわりに

駆け足で Emacs の操作の説明をしましたが、これはほんの入口です。Emacs は多機能なエディタであり、実にさまざまなことができます。また emacs lisp という言語を使って自分の好みに合うようにカスタマイズしたり新たな機能を加えたり、といったこともできるようになっています。計算機室の書架にも何冊か Emacs, NEmacs の解説書がありますので、興味を持たれた方は是非読んで下さい。

(May 13, 2002, 隅谷 孝洋)

ファイルの読み込み・保存

% mule file	Mule を起動し, file を読み込む (コマンドラインから)
C-x C-f	新しくファイルを読み込む (Mule 内から)
C-x C-i	ファイルを読み込んでその内容をカーソル位置に挿入する
C-x C-r	ファイルを書き込み保護された状態で読み込む
C-x C-s	Mule を終了せずにファイルの変更を保存する
C-x C-w	ファイルの名前を変えて保存する

カーソル移動

C-f, C-b	一文字進む/戻る
C-n, C-p	一行進む/戻る
M-f, M-b	一単語進む/戻る
C-v, M-v	一画面進む/戻る
C-a, C-e	行の先頭へ/末尾へ
M-<, M->	ファイルの先頭へ/末尾へ

操作の取消とコマンドの中断

C-x u	操作の取消
C-g	実行中のコマンドの中断

ブロック操作

C-Space	現在のカーソル位置をマークとして設定する (以降マークからカーソル位置までがリージョンとなる)
C-x C-x	カーソルとマークを入れ替える
C-w	リージョンを削除し, キルリングに保存する
M-w	リージョンを削除しないで, キルリングに保存する
C-y	キルリングの内容をカーソル位置に挿入する
C-k	カーソルから行末までを削除し, キルリングに保存する

文字列の検索

C-s	検索 (C-s で次検索, Enter で終了)
M-%	置換 (Space で変換, Delete でスキップ, Enter で終了)

バッファとウィンドウ

C-x C-b	バッファの一覧を表示
C-x b	バッファを切り替える
C-x k	バッファを削除する
C-x 0	現在カーソルがあるウィンドウを削除する
C-x 1	現在カーソルがあるもの以外のすべてのウィンドウを削除する
C-x 2	ウィンドウを横に分割する
C-x 3	ウィンドウを縦に分割する
C-x o	他のウィンドウにカーソルを移動させる

日本語入力

C-j	日本語変換の開始
Space	次候補の表示
C-n	次候補の表示
C-p	前候補の表示
C-f	文節移動 (次へ)
C-b	文節移動 (前へ)
C-i	文節縮め
C-o	文節伸ばし
C-h	ひらがなに変換
C-k	カタカナに変換
M-x yc-toroku-region	(リージョンを指定した後) 単語を個人辞書に登録する