知能情報工学演習 I

柳本 豪一

平成 18 年 4 月 27 日

1　エディタ

Linux 上でよく使われるエディタである Emacs について説明します。Emacs とは、UNIX 環境でもっとも広く普及しているテキストエディタであり、多機能であることが知られています。Richard Stallman によって最初のバージョンが作成されました。

この授業では、Emacs の派生バージョンである XEmacs を利用して演習を進めていきます。Linux 上でプログラムを作成したり、レポート、論文を書いたりするときには XEmacs を必ず利用することになるので、この機会に基本的な操作を覚えてください。

2　XEmacs の起動と終了

まずはじめに、XEmacs の起動と終了について説明します。起動するためには、ターミナルで「xemacs &」と入力します。しばらくすると、画面に新しいウィンドウの枠が現れます。

XEmacs を終了するためには、「C-x C-c」と入力します。今後よく出て来ますが、「C-x」は、**コントロールキーを押しながら、x キーを押す**ことを表しています。したがって、「C-x C-c」は**コントロールキーを押しながら x、コントロールキーを押しながら c を押す**操作となります。終了前にファイルの編集内容が保存されていないときには、XEmacs から保存せずに本当に終了して良いか確認が来るので、メッセージを良く読んで操作してください。

3　基本操作

ここでは XEmacs の基本操作を説明していきます。はじめは難しく思うかもしれませんが、XEmacs をキーボードだけで操作できるようにキーバインド (キーの入力に割り当てられた機能) を覚えましょう。慣れてしまうと、キーボードから手を離してマウスを操作することに比べ、快適に作業を進めることができます。キーバインドを覚えるため、以下のことをこの授業では禁止します。

- XEmacs を何度も起動しなおすこと
- 複数の XEmacs を立ち上げること
- カーソルキー (矢印キー) を使うこと
- XEmacs をマウスで操作すること

表 1: 基本編集キーバインド

キーバインド名	キーバインド
カーソルの前に 1 文字移動	C-b
カーソルの次に 1 文字移動	C-f
カーソルの次に 1 行移動	C-n
カーソルの前に 1 行移動	C-p
行頭へ移動	C-a
行末へ移動	C-e
ファイルの先頭へ移動	E-<
ファイルの最後へ移動	E->
指定行に移動	E-g
カーソル位置の文字を削除	C-d
リージョンのマーク	C-Space
リージョンのコピー	E-w
リージョンのカット	C-w
カーソル位置から行末までのカット	C-k
コピーやカットのペースト	C-y

表 1 と表 2 に今回の授業で勉強するキーバインドを載せます。表 1 は、カーソルの移動やコピー、ペーストなどのキーバインドです。一方、表 2 は、ファイルの読み込み、保存などの操作に関するキーバインドです。

表中の「C-文字」は、コントロールキーを**押しながら**、文字キーを押すことを示します。「E-文字」は、ESC キーを**押してから**、文字キーを押すことを示します。また、Space はスペースキーを表します。

3.1 ファイルの中を移動する

表 1 のキーバインドを参考にして実際に XEmacs を利用してみましょう。はじめに`~thy00057/ecsis1/2/sample.log`と`~thy00057/ecsis1/2/sample2.log`を自分のホームディレクトリにコピーしてください。コピーが終了したら、「xemacs sample.log &」と入力し、sample.log ファイルに対して以下の操作を行います。

では、カーソルキーやマウスを使わずに XEmacs 上のカーソルを上下左右に移動させましょう。コントロールキーを**押しながら**操作することを忘れないようにしましょう。これからファイルの編集を行うときには、XEmacs を使うことになるので、少し面倒ですが自由自在にカーソルを移動できるようになってください。

問題なく上下左右に移動できるようになったら、行頭、行末、ファイルの先頭、ファイルの最後に移動できることを体験してください。長いプログラムなどを作っているときに、新しいヘッダファイル (stdio.h などの C 言語で関数が定義されているファイル。プログラムを作る上で必ず必要となるものです) を追加するときなど、ファイルの先頭と最後を行き来する状況が発生します。このようなとき、ファイルの先頭と最後に移動できるキーバインドが非常に便利です。

表 2: 基本操作キーバインド

キーバインド名	キーバインド
XEmacs の終了	C-x C-c
ファイルのロード	C-x C-f
バッファの移動	C-x b
バッファの一覧を見る	C-x C-b
バッファの上書き保存	C-x C-s
バッファの別名保存	C-x C-w
カーソル位置から順方向に検索	C-s
カーソル位置から逆方向に検索	C-r
対話的置換	E-%
一括置換	E-x replace-string
コマンドの直接入力	E-x コマンド名
コマンドの中止	C-g
Undo	C-x u

最後に、行数を指定して移動する方法を覚えましょう。これも、プログラムをコンパイル (エディタで作成したプログラムを実行できる形に変換する操作のこと) しているときに、エラーが発生した行にすぐに移動するために、必要なキーバインドです。

以上で、基礎的なカーソルの移動方法を説明してきました。他にも、一単語分だけ移動や一文だけ移動などいろいろあるので、興味のある人は各自調べて使えるキーバインドを増やしていってください。

3.2 文字を削除する

エディタはファイルを編集するツールなので、文字の削除が用意されています。XEmacs でも、「C-d」が用意されています。「C-d」はカーソルがある場所の文字を 1 文字だけ削除します。動作を確認するため、今開いているファイルの文字を削除してください。

他に文字を削除する方法として、次のセクションでも説明しますが、「C-k」があります。これは、カット&ペーストのカットに対応する操作ですが、カーソルのある位置から後ろの文字を全部削除するときに便利です。いろいろな位置で、「C-k」を実行して、どのように削除されるか確認してください。

3.3 文字列をコピーする

エディタの便利な機能として、文字列のカット&ペーストやコピー&ペーストがあります。XEmacs でも、この機能は用意されており、うまく利用することで入力、編集作業が快適になります。

Windows のエディタを例に挙げて考えましょう。まず、コピー (カット) したい文字の範囲を反転表示させて指定します。次に、コピーするのかカットするのかを指定します。最後に、文字列を挿入したい場所にカーソルを移動させて、コピー (カット) した文字列を挿入します。

XEmacs では、コピー (カット) したい文字列を指定するため、**リージョン**により範囲を指定します。コピー (カット) したい文字列の先頭にカーソルを移動し、「C-Space」を押します。次に、コピー (カット) したい文字列の最後に移動して、「E-w」か「C-w」を入力します。「E-w」はコピー、「C-w」はカットに対応します。以上の操作で、コピー (カット) する文字列を指定することができます。あとは挿入したい場所にカーソルを移動して、「C-y」を入力すれば終わりです。

他にも、カーソルの現在ある位置から行末までカットする場合には、先ほど少しだけ説明した「C-k」を使うこともできます。自分がしたい操作を考えて、使い分けるようにしましょう。

それでは、今まで説明したキーバインドを利用して、ファイルを編集してみてください。今回のファイルはあとでは使わないので、どのように変更してしまっても問題はありません。

3.4 編集ファイルを指定する

エディタでは、いろいろなファイルを編集することになります。このため、エディタから編集するファイルを新しく開く必要があります。Windows では、マウスを使ってメニューから指定していたと思いますが、XEmacs では、すべてキーバインドを利用して操作します。

今、XEmacs では、sample.log を開いて編集している思います。ここで、sample2.log を編集したくなったときには、XEmacs に sample2.log を読み込まなくてはいけません。これを実現するキーバインドが「C-x C-f」です。これを実行すると、ファイル名を聞かれるので、「sample2.log」と入力します。

ここまでの操作によって、XEmacs で 2 つのファイルが開かれていることになります。はじめに編集していた sample.log をまた操作したくなった場合にはどうすれば良いでしょうか。対象ファイルを変更するために XEmacs では、**バッファの切替**が用意されています。XEmacs では、編集されているファイルはすべて**バッファ**という名前で管理されています。別のバッファ(ファイル) を操作したいときには、「C-x b」としてバッファ名を入力します。直前に操作していたバッファ(ファイル) にしたい場合には、そのままリターンを押せば切り替わります。それより以前に操作していたバッファに切り替えたいときには、そのバッファに付けられた名前を入力します。現在利用しているバッファの一覧を表示するには、「C-x C-b」とすれば表示されます。それでは、いろいろなファイルを XEmacs で開いて中身を見てみましょう。ただし、開いたファイルは絶対編集しないようにし、最後に保存をしないように気をつけてください。設定ファイルを変えてしまった場合、次回うまくログインできないなどの問題が発生するかもしれません。

3.5 ファイルの保存

入力や修正が終わったファイルは、普通は保存します。保存するために XEmacs では、「C-x C-s」と「C-x C-w」が用意されています。これは、上書き保存と別名保存に対応するものです。

それでは、今編集しているファイルを上書き保存で保存してましょう。その後、ターミナルで more を使って修正が行われているか確認してみましょう。次に、今編集しているファイルを sample3.log という名前で保存しましょう。ターミナルからちゃんと sample3.log で保存されていることを確認してみましょう。

3.6 ファイルを検索・置換する

エディタの便利な機能として、文字列の検索や置換があります。検索や置換は大きなファイルの中から必要とする文字列を素早く見つけることができたり、文字列の変更が間違いなく簡単に行えたりします。

まずは、検索について説明します。「C-s」と入力すると、検索する文字列を聞いてきます。文字列を入力するとファイルを下に向かって検索することができます。逆、上に向かって検索するときには「C-r」を利用します。今見つかった場所以外に文字列がないか調べるときには、「C-s」(または「C-r」) をもう一度入力します。それでは、sample2.log のどこに hidekazu という文字列があるか検索して探してみましょう。

次に置換について説明します。置換とは、ある文字列を他の文字列に置き換える操作をすることです。XEmacs では、「E-%」の対話的置換と「E-x replace-string」の一括置換の 2 つがあります。対話的置換では、文字列を置き換えて良いか確認がされますが、一括置換では確認せずに置き換えが実行されます。たくさん置換がある場合には一括置換が便利ですが、予想していない場所に同じ文字列があった場合には、予想外の置換が行われてしまうので、対話的置換が便利です。状況に合わせて使い分けてください。それでは、sample2.log で hidekazu を yanagimoto に置換してみましょう。

3.7 その他の機能

XEmacs にはここでは説明しきれないほど、さまざまな機能が用意されています。書籍やウェブページなどでも紹介されているので、各自調べてみてください。ここでは、表の残りのキーバインドについて説明します。

XEmacs の機能にはさまざまな名前が割り当てられています。この名前を手がかりに機能を実行するため、「E-x コマンド名」が用意されています。置換で利用した「E-x replace-string」は置換のコマンド名を用いて実行している一例です。また、XEmacs は利用できるコマンド名を自動的に補完してくれます。途中まで入力してスペースキーを押すと、残りを補完してくれます。もし、一つに決められない場合には、補完できるところまで補完して、候補となるコマンド名の一覧を表示してくれます。この補完機能は便利なので、どんどん使っていってください。

次にキーバインドを利用していて、処理を中断したい時に有効なキーバインドは「C-g」です。XEmacs で操作が分からなくなった場合には、「C-g」を利用してみると解決することが多いです。

最後に、誤って操作をしてしまったとき、その操作を取り消す操作 (アンドゥ) があります。これは、「C-x u」と入力すると、直前の操作を取り消すことになります。ただし、ファイルの保存などには有効でないので注意してください。

4 日本語の入力

XEmacs 上で日本語を入力することができます。ここでは、日本語入力への切り替えと操作に慣れましょう。

日本語入力を行うときには、「C-\」と入力します。XEmacs から「Input method:」と聞かれるので、「japanese-egg-wnn」と入力します。画面の下のステイタスを表示しているところに [あ] と表示されていれば問題ありません。これ以降、日本語と半角の英語を移るときには「C-\」と入力

すれば切り替えることができます。はじめの場合だけ、「Input method:」と聞かれるということを注意しておけばよいだけです。

かな漢字変換ですが、スペースキーで変換を行い、リターンキーで決定を行います。変換する文節を調整するために、「C-i」と「C-o」が用意されています。「C-i」は、文節を縮めることができ、「C-o」は伸ばすことができます。カタカナにするときは、「E-k」、ひらかなにするときは「E-h」です。まだまだいろいろなコマンドがありますが、上記の操作で大体問題なく入力することができると思います。詳しく知りたい人は、Web ページなどで検索を行って調べてください。Mozilla を利用して検索を行う場合に、日本語を入力したいときがあると思います。XEmacs 以外で日本語を入力するためには、Shift+Space と入力することで日本語を入力することができます。終了する場合も、Shift+Space と入力します。NumLock などが入っていると、日本語の入力モードから戻れないときがあるので、キーボードに変なシグナルが点いてないか確認してください。

それでは、新しいファイルを開いて日本語とアルファベットをいろいろ入力してみてください。

5 参考文献

XEmacs に関する書籍はたくさん出版されています。エディタとしての機能としては Emacs/Mule と同じなので、Emacs や Mule と書かれている書籍を読むことを勧めます。また、XEmacs に用意されているチュートリアルも参考になるので、時間があるときに一度目を通しておくことを勧めます。チュートリアルの実行は「C-h t」です。

6 付録

XEmacs の入力の練習として入力してみましょう。作成されたファイルは、適当なファイル名にして保存してください。ただし、最後に「.c」を付けることを忘れないようにしてください。例えば、test という名前で保存したい場合には、ファイル名を「test.c」とする必要があります。ファイルの保存が終了したら、ターミナルでファイルを保存したディレクトリに移動して、「gcc ファイル名」と入力してみましょう。上の例では、「gcc test.c」となります。その後、「a.out」と入力するとプログラムが実行されます。

6.1 サンプルプログラム

指定された一年間のカレンダーを表示するプログラムです。

```
include <stdio.h>

int daysOfMonth(int year, int month)
{
  static int days[12] = {
    31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

  if (month < 1 || month > 12) month = 1;
```

```
  if (month == 2) return days[1] +
                    (year % 4 == 0 && year % 100 != 0 || year % 400 == 0);
  else return days[month-1];
}

int dayOfWeek(int year, int month, int day)
{
  if (month == 1 || month == 2) {
    year--;
    month += 12;
  }
  return (year + year/4 - year/100 + year/400
          + (13*month+8)/5 + day) % 7;
}

int main(void)
{
  int year;
  void calendar();

  printf("--- 年間カレンダーを表示します ---\n");
  printf("西暦年を入力して下さい（入力例 1996）\n");
  scanf("%d", &year);
  calendar(year);    /* 関数 calendar() を使う */
  return 0;
}

void calendar(int year)
{
  int i, k;
  int month, days;

  for (month = 1; month <= 12; month++) {
    printf("\n    %d 年%d 月\n", year, month);
    printf("日 月 火 水 木 金 土\n");

    k = dayOfWeek(year, month, 1);
    days = daysOfMonth(year, month);

    for (i = 0; i < k; i++)
      printf("   ");
    for (i = 1; i <= days; i++) {
      printf("%2d ", i);
```

```
      if (++k % 7 == 0)
        printf("\n");
    }
    printf("\n");
  }
}
```