Introduction to the Emacs Editor

Eric Myers

Department of Physics
Randall Laboratory of Physics
University of Michigan
Ann Arbor, MI 48109-1120

Emacs is an extensible, customizable, self-documenting, real-time display editor. One of its many virtues is that it can be used on a wide variety of computers (Suns, Vaxes, and anything from Crays and Connection Machines to PCs and Macs) running a wide variety of operating systems (Unix, VMS, even DOS) and using a wide variety of terminals. The basic commands are the same no matter which kind of computer or terminal you are using, but emacs can also take advantage of the special features of a given terminal or machine. For example, emacs is completely compatible with the X windowing system. Emacs also has a number of features to assist with software development. It is possible with just a few keystrokes to edit a source file, invoke the correct compiler (or the `make` program) to compile the program, and then run the program from a sub-shell in an emacs buffer. If compiler errors are found emacs can move you to the line(s) in the file(s) where problems were found, so that the errors can be immediately corrected. When using a color X terminal window it is possible to have emacs highlight different types of command words in your program using different colors. Emacs is available on all Unix computers at Michigan.

Tutorial

Emacs comes with its own on-line tutorial. To take the tutorial on the computer you are using for the Fortran course you can simply type `learn-emacs`. To take the tutorial on any computer which has emacs you should first start emacs with the simple command

```
emacs
```

and then press `^H` (hold down the control key and press "H" — in the emacs documentation this is written as "`C-h`") to get "Help" from emacs, and then press "T" to start the Tutorial.

You should probably allow about an hour to go through the tutorial, but if you have to stop you can come back later and find the point where you left off.

To exit emacs press `^X-^C` (that's control-X followed by control-C). This is always the way to terminate emacs.

It may also be useful to know that emacs makes a distinction between the Backspace and Delete keys on your terminal. Backspace is the same as `^H`, so if you press Backspace you will get the emacs help menu. To delete a character you will want to use the Delete key.

Editing

Once you have learned the basics of emacs you can edit a file by giving the emacs command with the name of the file to edit on the command line, as in:

```
emacs  hello.f
```

Customization

Like many programs designed for the Unix operating system, emacs can be customized by placing commands in a startup file. Emacs will look for such commands in a file called `.emacs` in your home directory. While it is not required, you may find it easier to use emacs if you obtain a copy of a simple startup file from me. The command to do so is:

```
cp ~myers/fortran/sample.emacs  ~/.emacs
```

If you look in this file you will see that the setup instructions for emacs are written in a dialect of the LISP language. Execept for things that depend on the operating system, these setup instructions can be the same on any computer.

If you are not at Michigan you can copy the `sample.emacs` file (and many other useful files for this course) via anonymous ftp from `ftp://scarlett.vassar.edu/pub/fortran`.

Overview for `pico` users

If you use pine for e-mail then you are already familiar with an editor called `pico`— it is the default editor for pine.

If you already have experience with pico then you will have no trouble learning emacs, except that you may have to re-train your fingers for some of the emacs keystrokes that are different from pico keystrokes. But the basic idea behind emacs and pico is the same: certain editor functions are invoked by typing special keys or sequences of keys, usually "control" keys.

One difference you may notice right away is that emacs does not give you any a menu to remind you which keys to press. Pico is for beginners and very helpfully puts a menu of function keys along the bottom of the screen. Emacs assumes that you know a little bit more about what you are doing, and only gives you such help if you ask for it (with `^H`).

While it may be tempting to continue using pico, you should try to learn emacs instead. Emacs is more powerful than pico, and has some special features that are very useful when you are writing computer programs (rather than just editing e-mail). In the long run you will find it was worth the extra effort to learn emacs, and it's not that difficult once you already know pico.

Overview for `vi` users

The vi editor is standard on all Unix computers, so some people may already know how to use it even if they have not yet learned a programming language.

If you already know the vi editor you will have no trouble learning emacs, except that you may have to re-train your fingers for emacs keystrokes. The basic idea behind emacs and vi is the same: certain editor functions are invoked by typing special keys or sequences of keys. But there are some important differences. Unlike vi, emacs is always in "insert" mode. Any printable character you type is automatically inserted at the position of the cursor (called the "point"). The most basic editing functions are "bound" to control keys. Moving the cursor up, down, left or right is accomplished by typing `^P`, `^N`, `^B` or `^F`. (You can remember these with the mnemonics "previous", "next", "backward" and "forward".) A character can be deleted by pressing `^D`. (In the emacs documentation control-D is written as `C-d`, not `^D`, but it is the same key – hold down the control key and press the D key.)

More complicated editing functions are bound to keys on the "meta" keyboard. To invoke one of these functions you press the "escape" (ESC) key first, followed by some other key. For example, to move the display up one page in the file the command is ESC followed by `v`. In the emacs documentation this is written as `M-v` and read as "meta-vee". Even more complicated functions are bound to the "extended" keyboard; To invoke one of these functions you press control-X and then another key. An important example is the command to save all files and exit emacs: `^X-^C`. Finally, there is a third special keyboard, the "custom" keyboard, which is introduced by the `^C` key. The functions on the custom keyboard can change depeneding on the editing mode you are in, or you can add your own keys to the custom keyboard. The startup file you copied from me defines `^C-C` to be the `compile` command and `^C-N` to be the `next-error` command. Both of these are very useful for editing and debugging programs.

Emacs can also execute commands which are not bound to any keys, much like the "`:`" command of vi. To give one of these commands you enter the command `M-x` (that's ESC, followed by the "x" key). Emacs will then let you type your extended command in the "mini-buffer" area at the bottom of the screen.

Exercise

Once you feel comfortable with emacs, edit the file called `.login` in your home directory to change your name. Near the beginning of this file you should find a line like:

```
setenv NAME "Fortran student"  # <-- change to your name here
```

Change this line so that your own name appears within the quotation marks, such as

```
setenv NAME "Matthew Vassar"  # <-- change to your name here
```

If you don't find such a line, just create one. Be sure that you still have both the opening and closing quote marks. After you do this your full name will appear (along with your userid) in any electronic mail messages you send.

While you are editing your `.login` file you can also select a default printer. Change the name of the `PRINTER` environment variable to the name of the printer you wish to be the default. The next time you log in, you can use the `lpr` command without having to add the "`-P`" flag to specify the printer.