Program development

- emacs (and vi)
- flex and bison
- makefiles
- source level debugging
- diff
- rcs and subversion
- gprof
- glade

is a superior text-based program development environment over `vi`, and it is easy to install.

Why use emacs? The way that emacs %

- While not "standard", as is `vi`, it is very common and it is generally very easy to install these days.
- It is completely programmable. In fact, it takes the idea of programming to a much higher level in that it maps arbitrary sequences of keystrokes to arbitrary functions.
- emacs lisp is a pleasant programming language. If you like other languages, other versions of emacs support: MacLisp, scheme, guile, Common Lisp, ObjectCaml, even teco.
- emacs has also been called "Eight Megabytes And Continuously Swapping." Despite that moniker, it is actually reasonably efficient.
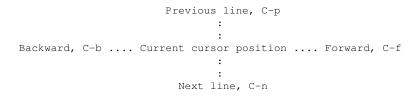
The tutorial

Most of the verbatim material here is taken "verbatim" from the Emacs Tutorial. You can use `ctrl-h t` to display this tutorial in `emacs`:

```
The following commands are useful for viewing screenfuls:
      C-v    Move forward one screenful
      M-v    Move backward one screenful
      C-l    Clear screen and redisplay all the text,
             moving the text around the cursor
             to the center of the screen.
```

```
                         Previous line, C-p
                                 :
                                 :
   Backward, C-b .... Current cursor position .... Forward, C-f
                                 :
                                 :
                           Next line, C-n
```

>> Move the cursor to the line in the middle of that diagram
   using C-n or C-p. Then type C-l to see the whole diagram
   centered in the screen.

A quick summary of most useful "move around" commands

```
C-f     Move forward a character
C-b     Move backward a character
M-f     Move forward a word
M-b     Move backward a word
C-n     Move to next line
C-p     Move to previous line
C-a     Move to beginning of line
C-e     Move to end of line
M-a     Move back to beginning of sentence
M-e     Move forward to end of sentence
M-<     Move to top of the buffer
M->     Move to bottom of the buffer
```

The mode line has several parts: the first indicates your coding system
(use `c-h  C` to find more information about your current one.
It then has some status information: a `%%` indicate that the buffer is
read-only, `**` indicate that the buffer is modified,

The menu bar

If you like menu bars, you can access the one in emacs with m- `.

Creating windows

You can split your current window vertically with `c-x 2`.
You can split your current window horizontally with `c-x 3`.
You can jump around windows with `c-o`. You can even scroll another buffer with `c-m-v`
You can `c-x 1` to get rid off all but one window.

Buffer control

You can list your current buffers with `c-x c-b`. You can even use
`c-x o` to leap into that buffer and then use the "o" key to go directly
to that buffer, or the "k' key to mark the buffer for removal (does not
affect the file), and the "x" to do the marked removals.
You can also use `c-x b` to switch buffers.
Finally, `c-x s` will let you save all modified buffers.

You can record simple macros in `emacs` with `c-x` `(` and `c-x` `)`.

To play the macro, use `c-x e`

You can give an argument to a function with `c-u NUM`; giving one to a keyboard macro invocation causes that macro to be called that many times.