Text file editing

The two most commonly Unix editors are `vi` and `emacs`. `emacs` is very big and it is trying to be everything — `emacs` users usually do almost everything under `emacs`, including reading/sending e-mail, reading news-groups, etc. A "minimal" `vi` can be very lightweight and because of that at least some version of `vi` is always installed on most systems.

VI family — `vi`, `vim`, `gvim`

The original `vi` (stands for "visual editor") `vi` is very limited compared to its modern clones. One of the clones, `vim` (stands for "**VI**mproved") is backwards-compatible with `vi`, but has many modern features not present in `vi`. `vim` is console-based, but has an XWindows version - `gvim`. To start `vi` (`vim` or `gvim`), run `vi filename` (`gvi filename` or `gvim filename`).

Miscellaneous VI commands

`:help ↵` or `:help <subject> ↵` — get help (for example, `:help set ↵`)

`:set <option> ↵` or `:set <option>=<value> ↵` — set an `<option>`. Running `:set ↵` would list current options and `:set no<option> ↵` would unset an `<option>`

`:nn ↵` or `nnG` — go to line `nn`

`:x ↵` — save file and quit. `:w ↵` — write (save) file, `:q ↵` — quit, `:q! ↵` — quit without saving.

Editing mode in VI

`vi` has two main modes — editing and command. When started, `vi` enters the command mode. After one of the editing commands is entered, `vi` enters editing mode.

- `i` insert text if front of the cursor
- `I` insert text at the beginning of the current line
- `a` add text after the cursor
- `A` add text at the end of current line
- `o` add text after the current line
- `O` add text before the current line
- `R` replace text under cursor

To return to command mode, press `<ESC>`.

Search and replace

`/<string> ↵` — search for a `<string>` and `/ ↵` (or just `n`) — repeat previous search. `?` and `N` — same as `/` and `n` for going up in the file.

`:<range>s/<string1>/<string2>/<options> ↵` — substitute `<string2>` for `<string1>`.

Ranges: `mm,nn` — lines `mm` through `nn` (inclusive), you can use `$` instead of `nn` to stand for "the last line in the file" and you can use `%` to stand for `1,$` ("the whole file"). If the range is omitted, it means "current line".

Options: usually you do not need any. The only interesting one is `g` — replace all occurrences (instead of the first one in each line).