Editors

CIS 2230 Linux System Administration

Lecture 10

Steve Ruegsegger

Review – regex, sed, awk

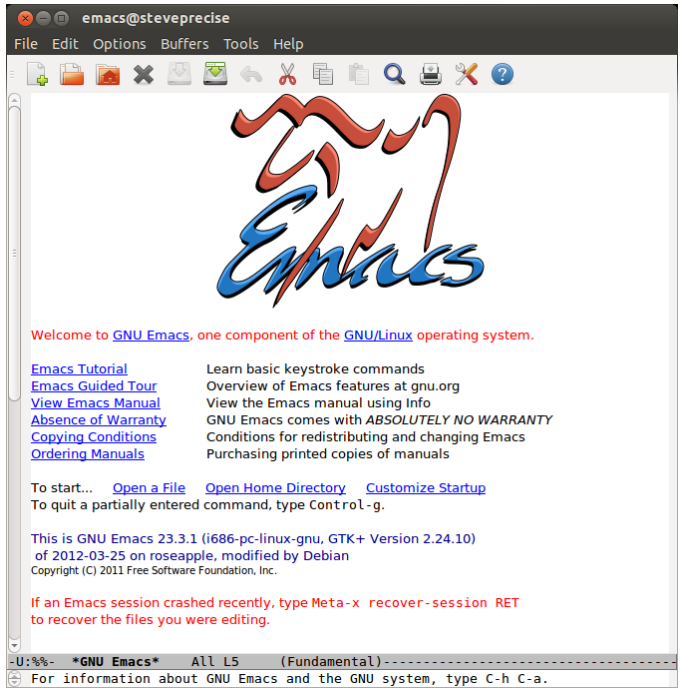
- What are the 2 *anchors* in regex?
- How do you specify a character range in regex?
- What is the regex *wildcard* character?
- In regex, what are 3 ways to specify the char or range repeats?
- What is this regex looking for? Explain.

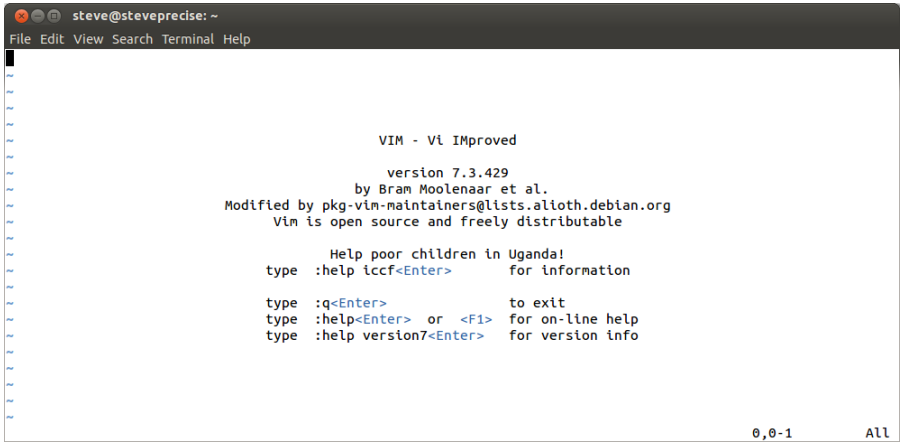
```
/^.*[ ,]?VT +?[0-9]{5}-?[0-9]{4}/
```

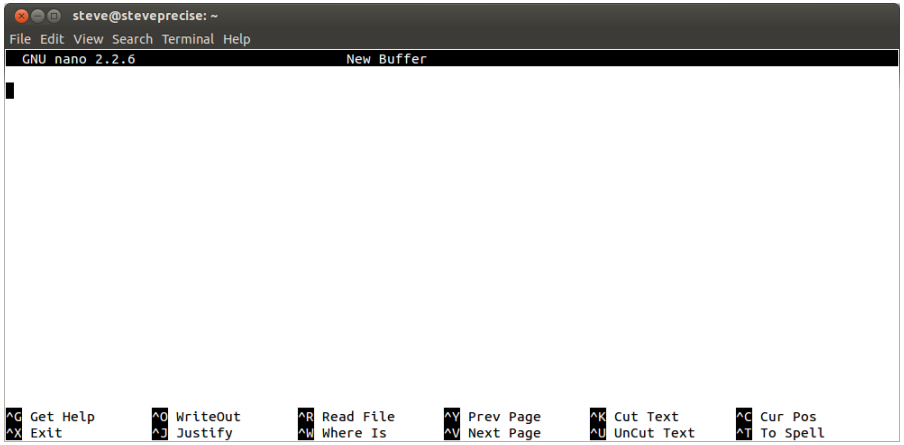
- In AWK, what specifies each column in the input stream?
- In AWK, the format is `$ awk '(expr){expr}'` Describe those 2 expressions.

Editors

- Why?
- Popular ones?
- Why learn them all?

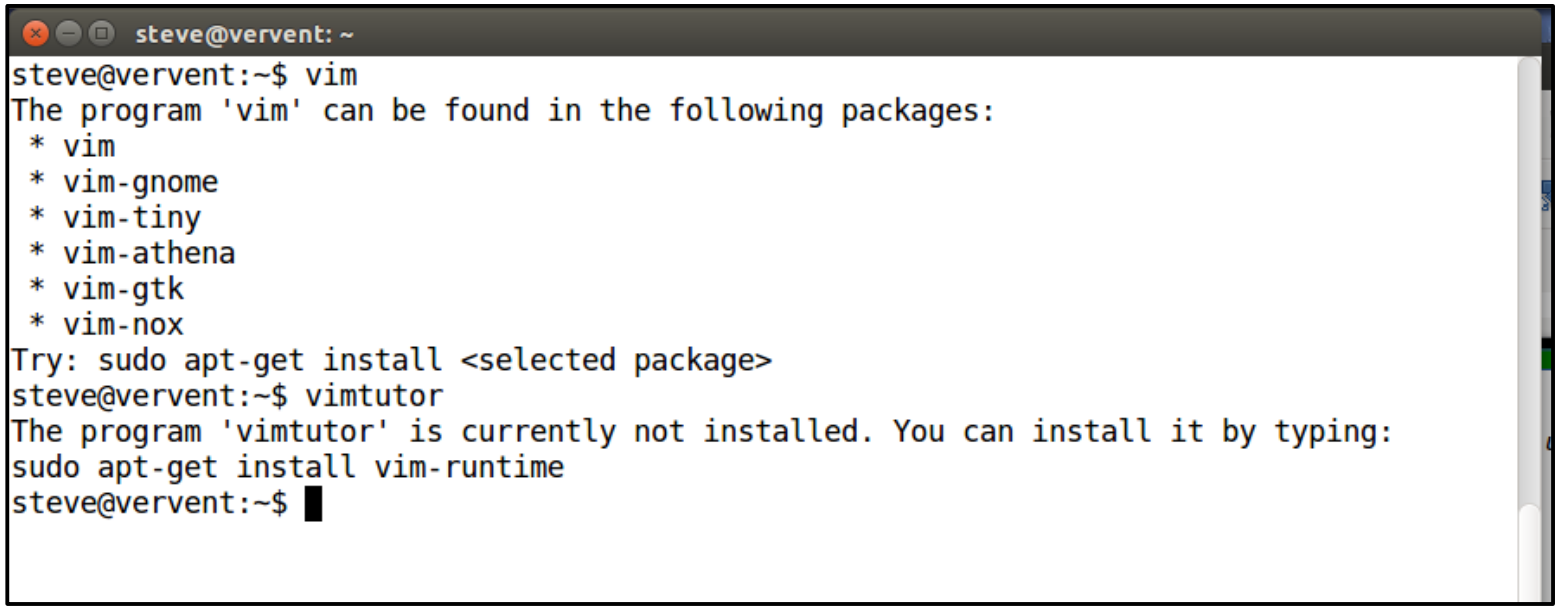






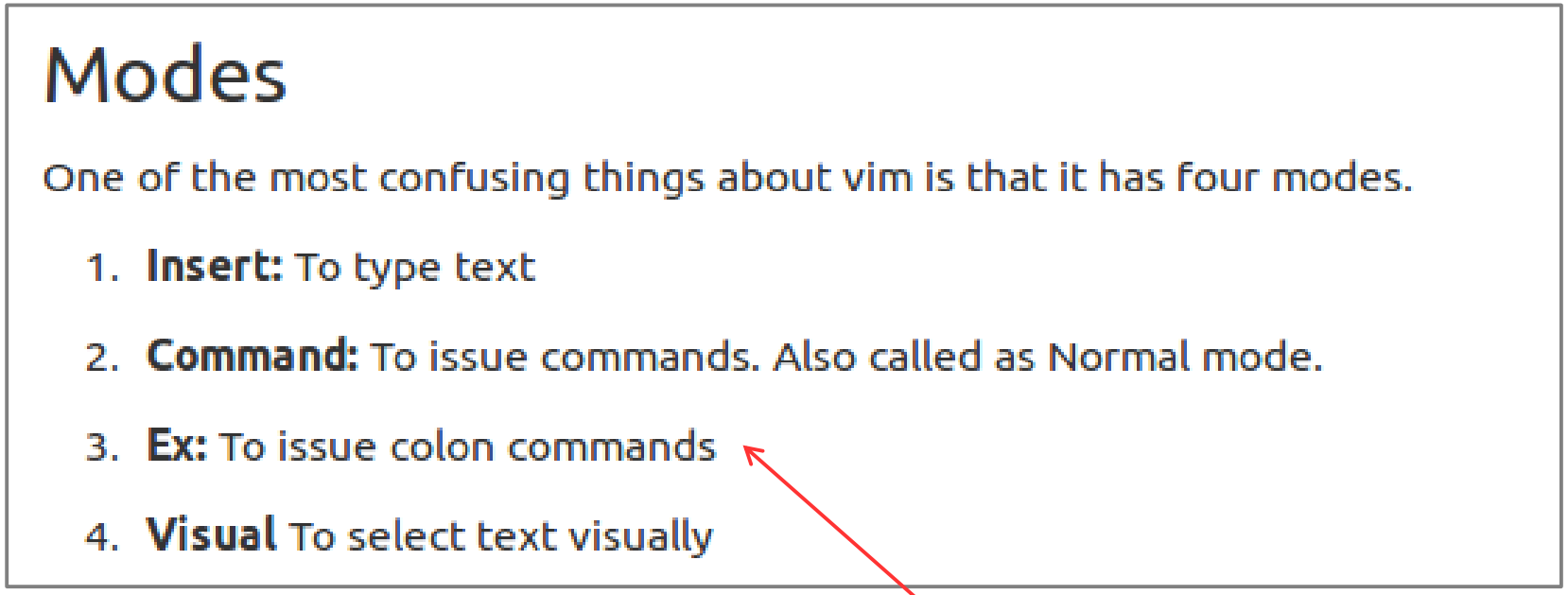
vi(m)

- vi exists on virtually every unix installation: pervasive & ubiquitous
- vi is the default editor for “most every” linux distro
- vim is simply “vi improved”
- `$ vimtutor`
  - Note: for Ubuntu 13.xx+, “they” (Canonical) decided to only install vim-tiny by default, so `vimtutor` is not installed.
- https://help.ubuntu.com/community/VimHowto



vi essentials!

- Must Memorize:
  - 4 modes
  - Cursor movement
  - Insert/delete text
  - Save / don't save and quit



Colon commands are file commands for inserting, writing, saving, exiting.

nano

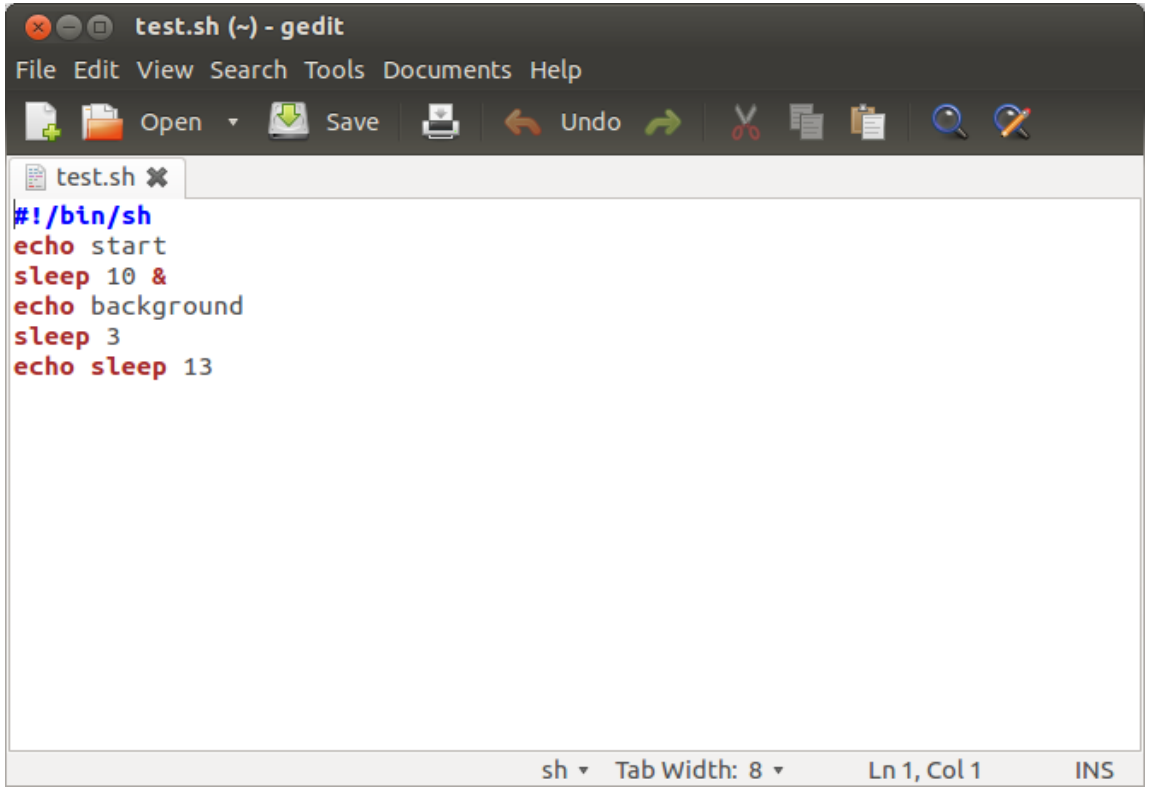
- GNU nano is a simple terminal-based text editor. Though not as powerful as Emacs or Vim, it is easy to learn and use.
- Nano is a gnu (i.e. “free”) replacement of non-free pico editor.
- https://help.ubuntu.com/community/Nano
- Note: line-wrapping *on* by default, use “-w” option to turn off.
- The bottom 2 lines gives a “shortcut list” of most used commands.

emacs

- My *favorite* editor
- More complicated, yet more powerful
- Need to install in Ubuntu – not in default install
- Commands are often 2 “Cntrl-<key>” commands
- Emacs opens a new GUI window, $ emacs -nw for text mode.
- Emacs keystrokes used in other places, including linux shell terminals
- https://help.ubuntu.com/community/EmacsHowto
  - Good summary of key commands
  - Movement commands: ^a, ^e, ^p, ^n, ^f, ^b
  - Delete: ^d, ^k
  - File commands: ^xf, ^xs
  - Important: ^g, ^xu, ^x^c

gedit

- The default, simple graphical editor for Ubuntu
- Commands in buttons and pulldown menus
- Many nice “context modes” and highlighting
- Full mouse integration



Simple scripting

2 methods run a script

1. Method #1
   - Just put commands in a file
   - The file is argument to a shell

```
$ bash backup.sh
```

2. Method #2
   - Add the "*shebang*" – `#!/bin/bash`
   - Add commands to file (save it)
   - Change file permissions to be executable by user

```
$ chmod u+x <file>
```

   - Run your script file from the command prompt
     - You *must* understand: Explicit vs Implicit :)