

Editors

CIS 2230 Linux System Administration

Lecture 10

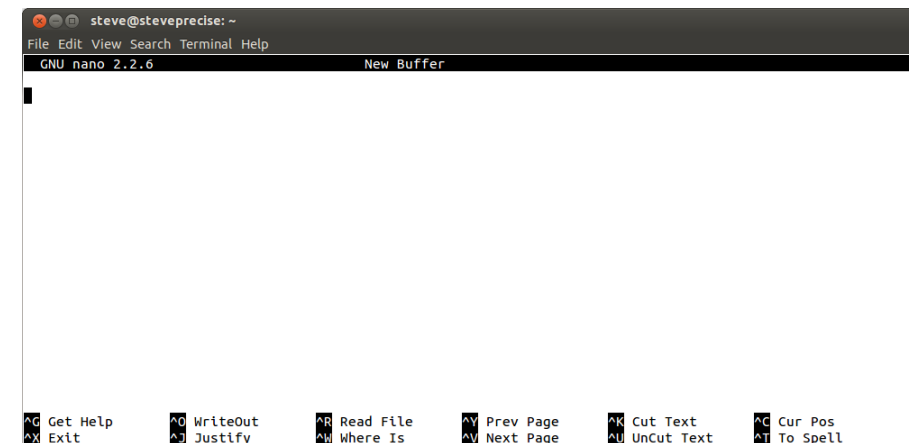
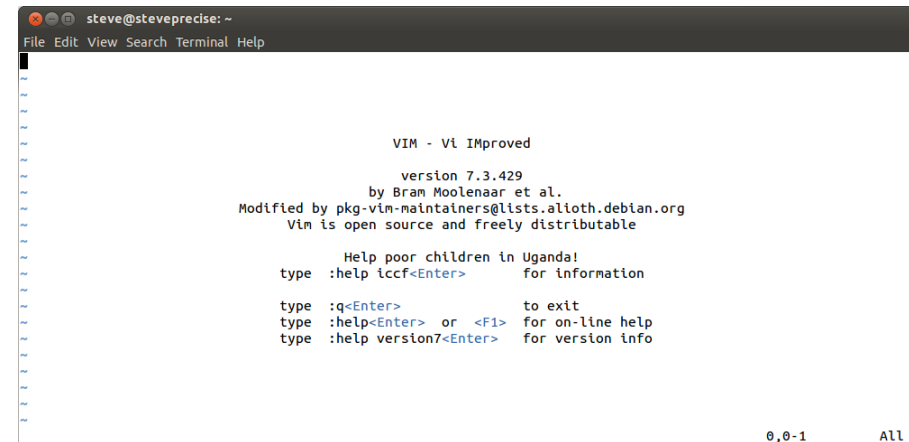
Steve Ruegsegger

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.

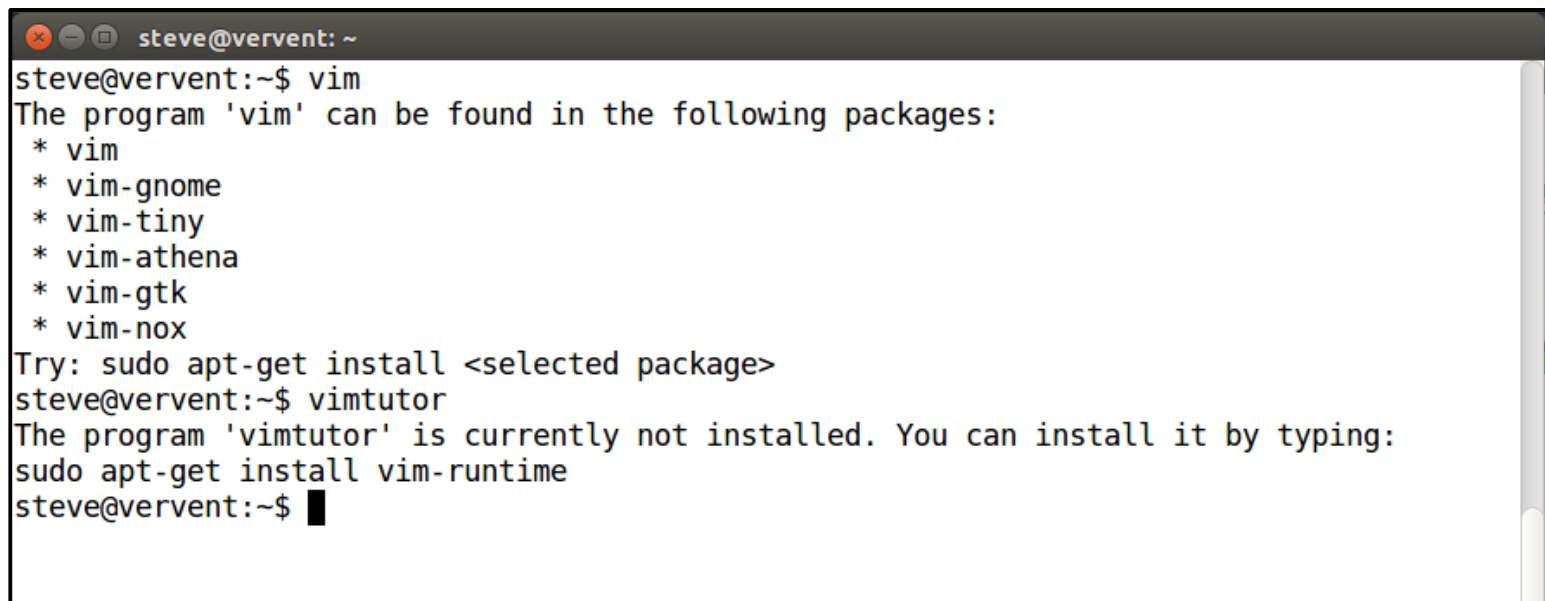
Editors

- 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>



```
steve@vervent: ~  
steve@vervent:~$ vim  
The program 'vim' can be found in the following packages:  
* vim  
* vim-gnome  
* vim-tiny  
* vim-athena  
* vim-gtk  
* vim-nox  
Try: sudo apt-get install <selected package>  
steve@vervent:~$ vimtutor  
The program 'vimtutor' is currently not installed. You can install it by typing:  
sudo apt-get install vim-runtime  
steve@vervent:~$
```

vi essentials!

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

Modes

One of the most confusing things about vim is that it has four modes.

1. **Insert:** To type text
2. **Command:** To issue commands. Also called as Normal mode.
3. **Ex:** To issue colon commands
4. **Visual** To select text visually



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.

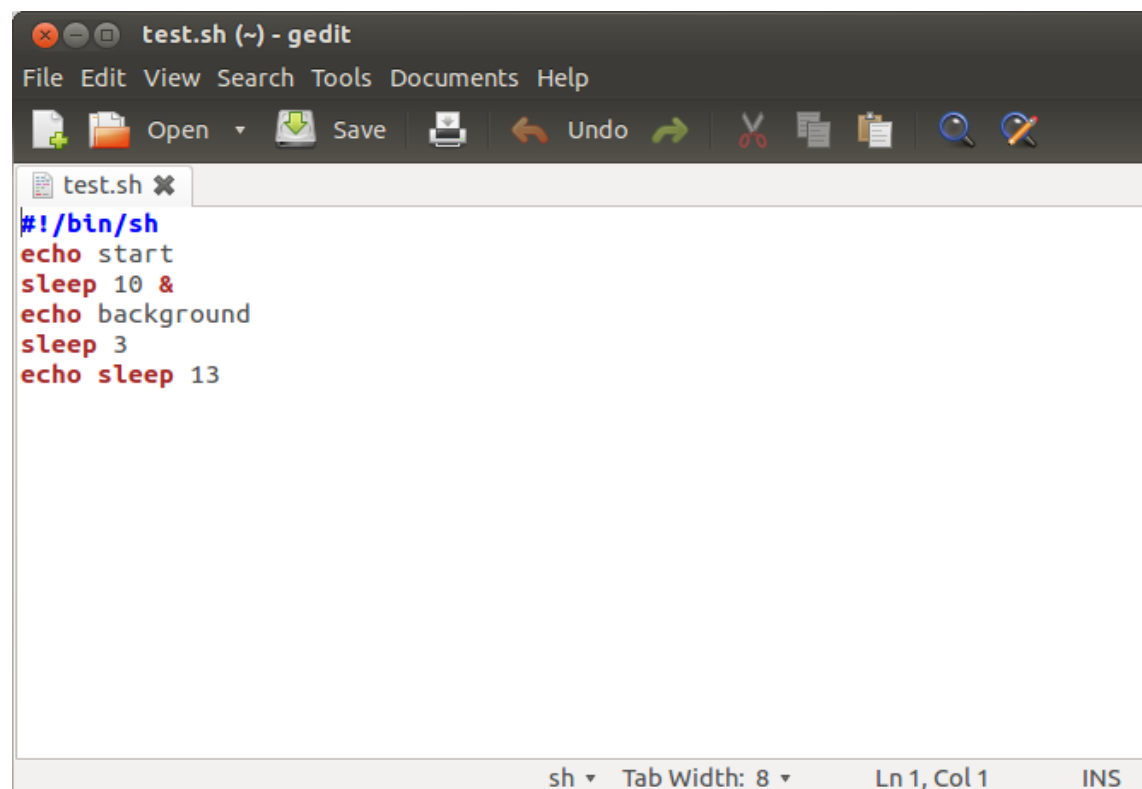
^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell

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



The screenshot shows the gedit graphical text editor window titled "test.sh (~) - gedit". The window has a menu bar with "File", "Edit", "View", "Search", "Tools", "Documents", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, printing, undo, redo, cut, copy, paste, search, and zoom. The main editing area shows a shell script file named "test.sh" with the following content:

```
#!/bin/sh
echo start
sleep 10 &
echo background
sleep 3
echo sleep 13
```

The status bar at the bottom of the window displays "sh", "Tab Width: 8", "Ln 1, Col 1", and "INS".

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 :)