Tools of the Trade

The Basics of Linux

Linux Operating System

- First released in 1991 by Linus Torvalds
- Based on free and open source software
- Written in C and assembly language
- Multi-user capability
- Everything accessible through graphical user interface (GUI) or terminal (command-line interface)
- AppleOS and Android both run on varying versions and architectures of Unix kernel

Linux at UND

- Dr. Delene is head of IT for Atmospheric Sciences Department
  - Clifford Room 420
  - delene@aero.und.edu
- Scientific Computing Center (SCC) is head of IT for JDO School of Aerospace Sciences
  - helpdesk@aero.und.edu
  - Monday-Friday, 8:00 am-4:30 pm

Linux at UND

- Advanced Scientific Visualization Laboratory
  - Clifford Hall Room 422
  - 9 Linux computers and 5 Windows computers with printer, scanner, and wireless projector
    - All Linux computers are Linux Mint except for radar2 (RedHat 5) and aitken (RedHat 6)
    - 4 Windows 7 and 1 Windows 10 computers
  - Login with Aerospace account
    - UN: first initial, last name; PW: w, student ID
  - Shared network drive
    - /home/username on Linux (U:/ drive on Windows)
    - 5 GB limit per student
  - Note: office computers aren’t connected to shared network, so need remote connection for access

Command-line Navigation: Essential Commands

Command	Function
cd	Change directory
ls	List contents of directory
pwd	Present working directory
man	Command help page (manual)
cp/scp	Copy across directories/servers
mv	Move/rename file/directory
rm	Remove file
mkdir/rmdir	Make/remove directory
ssh	Secure Shell Host (connect to remote Linux computer
history	List most recent commands
top	Display Linux processes

Command-Line Navigation: Essential Commands

Command	Function
grep	Find matching pattern in text file
find/locate	Search for files
du	Disk useage
df	See what drives are mounted and how much space is used on each

Command-line Navigation: Shortcuts

Keystroke/shortcut	Function
~	Home directory
Tab key	Complete file/program name
Up/Down arrow keys	Previous commands
!a	Execute most recent command that begins with "a"
!!	Execute most recent command
CTRL+C	Kill running command
./command > test.txt	Direct output from "command" to "test.txt" file

Command-line Navigation: No Special Characters

- A character with a meta-meaning (meaning beyond its literal meaning) is a special character
  - Space, #, “”, |, etc.
  - Need to be negated to use them for file name
    - “Cloud\ Physics\ Lab\ \#2.docx”

Linux Environment Variables

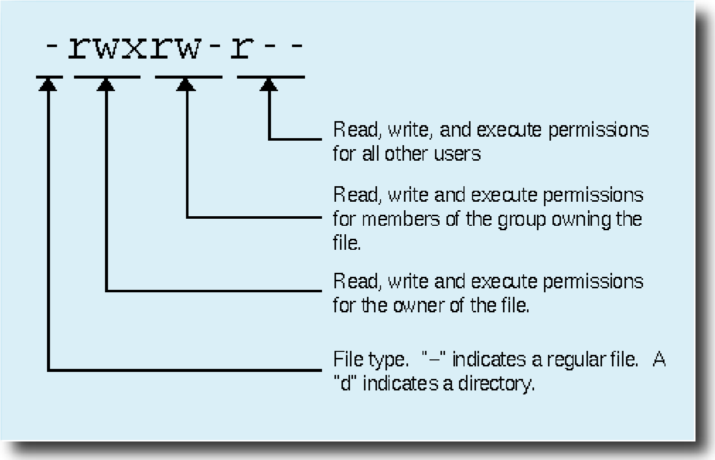
- Variable used by the OS for configuration purposes
  - $PATH is where all executable commands are located
  - $SHELL is what shell you’re using
  - $PYTHONPATH is where Python looks to run programs from
- Can be created for one-time use or set forever by both users and other programs
- Accessible from command-line
  - echo $PATH prints the value of the $PATH variable
  - export foo=2 sets the environment variable $foo to the value of 2

~/.bashrc File

- User-defined Linux configuration settings can be put in the ~/.bashrc file
    - File is executed by OS upon the opening of any new terminal (including logging into the system)
- Useful to keep configuration settings permanent
    - Aliases to commands
        - alias school=‘cd /home/ngapp/School’
        - alias derecho=‘ssh derecho.atmos.und.edu –Y’
        - alias latest=‘ls –ct | head -1’
    - Permanently set environment variables

Executing Programs from Command Line

- Commands need to be in the directories outlined in $PATH to be run from anywhere
- Programs need to be executable to be run from the OS
  - ls –l gives you a long listing
  - chmod (change mode) to make programs executable
  - which shows where command/ program is being run from



```
-rwxrw-r--
```

Read, write, and execute permissions for all other users
Read, write and execute permissions for members of the group owning the file.
Read, write and execute permissions for the owner of the file.
File type. "–" indicates a regular file. A "d" indicates a directory.

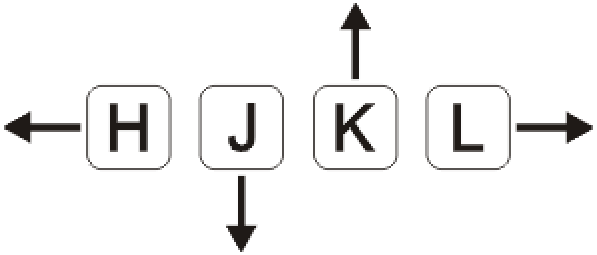
Vi IMproved (VIM) Text Editor

- Text editor built into Linux OS
- Hands never leave the keyboard—never touch your mouse again
- Syntax highlighting for many file extensions
- Steep learning curve, but it pays off
- Help:
  - http://yannesposito.com/Scratch/en/blog/Learn-Vim-Progressively
    /
  - http://vim.wikia.com/wiki/All_the_right_moves
  - https://vim.sourceforge.io/docs.php

VIM: Four Modes

- Normal Mode
  - Move cursor, copy/move/delete lines, etc.
- Command Mode
  - Execute VIM commands (save, quit, split windows, search/replace, etc.)
- Insert Mode
  - Type characters like a regular text document
- Visual Modes
  - Highlight characters, lines, or blocks of characters and lines to move, copy, delete, search, etc. within highlighted area

VIM: Normal Mode—Scrolling and Moving



Keystroke	Function
H	Cursor to top of window
L	Cursor to bottom of window
zz	Cursor to middle of screen
$	Move cursor to end of line
0	Move cursor to beginning of line
W	Move cursor to beginning of next word (left to right)
B	Move cursor to beginning of previous word (right to left)
G	Bottom of document
gg	Top of document
CTRL+b	Page up
CTRL+f	Pade down

VIM: Normal Mode—Moving Text

Keystroke	Function
y	Copy character under cursor
Y	Copy line under cursor
d	Cut (delete) character under cursor
dd	Cut (delete) line under cursor (including new line character)
D	Cut (delete) from cursor to end of line (excluding new line character)
P	Paste buffer before cursor
p	Paste buffer after cursor
x	Cut (delete) like DELETE key
u	Undo (can undo until file was last opened)

VIM: Command Mode

- Accessed from Normal Mode

Keystroke	Function
:w	Save (add filename to Save As)
:q/:q!	Quit/quit without saving
:wq	Save and quit
:e	Open (add filename to open new document)
:split/:vsplit	Split window horizontally/vertically
:tabe	Open new tab (add filename to open separate document)
:s/:%s	Find and replace in line/document
/	Search
:sh	Return to shell without losing current VIM configuration

VIM: Insert Mode

- Accessed from Normal Mode by pressing “i”
- To insert symbol (Δ, µ, ±, etc.):
  - From Normal Mode, type :digraph to see all symbols VIM has to offer
    - 1st column is VIM’s symbol code
    - 2nd column is the symbol
    - 3rd column is normal Unicode number of symbol
  - Take note of VIM’s two-character symbol code of desired symbol
  - In Insert Mode, type CTRL+K and type symbol code

VIM: Visual Modes

- Accessed from Normal Mode using keystrokes below
- After highlighting, can cut/delete/copy/paste using commands from Normal Mode and can find/replace within highlighted section using commands from Command Mode

Keystroke	Function
v	Highlight characters by moving cursor any direction (VISUAL mode)
V	Highlight lines by moving cursor up or down (VISUAL LINE mode)
CTRL+V	Highlight blocks of characters by moving cursor any direction (VISUAL BLOCK mode)

VIM: Keystrokes that Combine Modes

Keystroke	Starting Mode	Function and Ending Mode
o	Normal	Add new line below and enter Insert Mode
O	Normal	Add new line above and enter Insert Mode
A	Normal	Go to end of line and enter Insert Mode
C	Normal	Cut (delete) from cursor to end of line and enter Insert Mode
S	Normal	Cut (delete) entire line and enter Insert Mode
I	Visual Block	Insert text before highlighted block
:s	Any Visual Mode	Find and replace within highlighted area

VIM: Tips and Tricks

- Save VIM session for later with :mksession session_name.vim (need .vim extension)
  - Access session from command-line using command vim –S session_name.vim
- Use numbers to repeat command several times
  - 3Y will copy current line and next 2 lines (3 lines total)
  - 3P will paste current buffer 3 times
  - 10gg will go to the 10th line of the document
- Scroll split windows at same time with :set scrollbind
  - Need to set scroll bind on each window in need of simultaneous scrolling

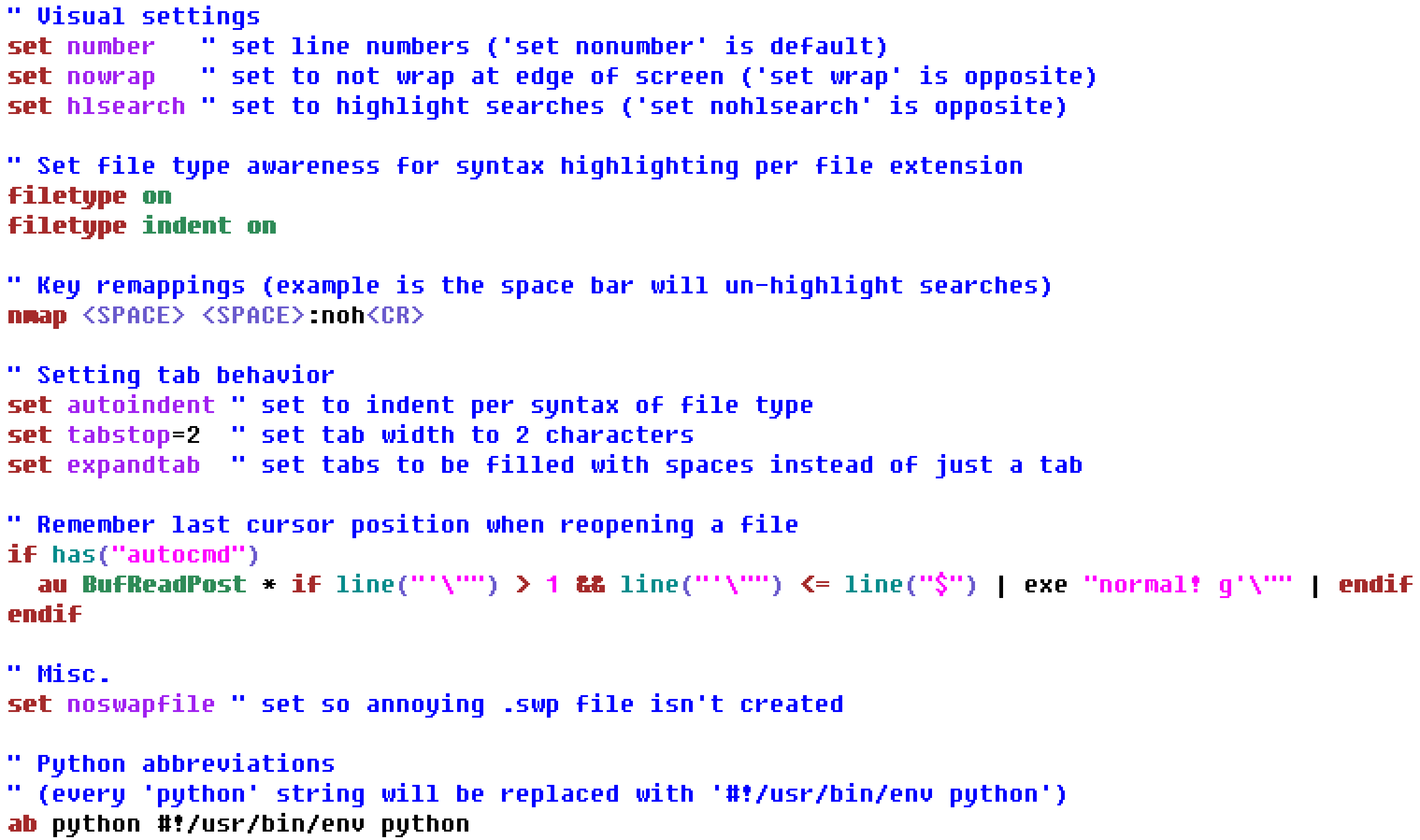
VIM: Tips and Tricks

- Run OS command in VIM with :r
  - :r !ls will paste the directory listing into VIM starting at cursor position
- Easily capitalize/un-capitalize text in any Visual Mode
  - Highlight text and press U/u to capitalize/un-capitalize
- Easily indent lines
  - From Normal or Visual Line Mode, type >> to indent once or << to un-indent once
  - Use with numbers to indent more than once (3>> indents 3 times)

VIM: Tips and Tricks

- Search for multiple words at once
  - /first_word\|second word
- Make search case-insensitive
  - /word\c
- Scroll with cursor always in middle of screen
  - From Normal mode, :set so=999
- ~/.vimrc
  - Like ~/.bashrc but for VIM
  - See next slide for useful commands

VIM: ~/.vimrc



```
" Visual settings
set number   " set line numbers ('set nonumber' is default)
set nowrap   " set to not wrap at edge of screen ('set wrap' is opposite)
set hlsearch " set to highlight searches ('set nohlsearch' is opposite)

" Set file type awareness for syntax highlighting per file extension
filetype on
filetype indent on

" Key remappings (example is the space bar will un-highlight searches)
nmap <SPACE> <SPACE>:noh<CR>

" Setting tab behavior
set autoindent " set to indent per syntax of file type
set tabstop=2  " set tab width to 2 characters
set expandtab  " set tabs to be filled with spaces instead of just a tab

" Remember last cursor position when reopening a file
if has("autocmd")
  au BufReadPost * if line("'\"") > 1 && line("'\"") <= line("$") | exe "normal! g'\"" | endif
endif

" Misc.
set noswapfile " set so annoying .swp file isn't created

" Python abbreviations
" (every 'python' string will be replaced with '#!/usr/bin/env python')
ab python #!/usr/bin/env python
```