Unix Essentials

Devin J. Pohly <djpohly@cse.psu.edu>
The Unix Philosophy

• Write programs that do one thing and do it well.
• Write programs to work together.
• Write programs to handle text streams, because that is a universal interface.

  ‣ Doug McIlroy, Unix patriarch
Why use the CLI?

• Efficient and powerful
• Easy to automate/script
• Simple and reliable
  › Works even when everything else is completely b0rked!
CLI and the filesystem

• Fairly typical structure
  ‣ Everything under one root directory, called “/”
  ‣ Separate directory names with additional slashes
  ‣ To back out to the parent directory, use “..”

• Examples:
  /usr/share/dict/words
  /usr/bin/./share/dict/words
  ‣ These are called paths.
Top-level directories

- Organized by the type of files they contain
  - `/usr`: installed software
    - `/usr/bin`, `/usr/lib`, ...
  - `/etc`: configuration
  - `/home`: users’ own files
  - `/dev`: devices
  - `/tmp`: temporary files
Navigating the filesystem

• Current *working directory*
  ‣ Default point of reference
  ‣ Shown in prompt

• Change directory: `cd`
  ‣ `cd /usr/share/dict`
  ‣ `cd` by itself takes you back home

• List files: `ls`

• Special directory names
  ‣ . means the working directory
  ‣ .. means the parent directory
    ‣ So `cd ..` takes you up a level
Practice

• Find the configuration files for the APT package manager.

• View the list of sources using the `less` program.
  › Press `q` to quit less.
Working with files

• Copying files:
  ‣ `cp src dest`

• Moving/renaming files:
  ‣ `mv src dest`

• Removing files:
  ‣ `rm file`

• Creating/removing empty directories:
  ‣ `mkdir newdir`
  ‣ `rmdir dir`
Getting help

• Manual pages: man
  ‣ Standard layout
  ‣ man operator
  ‣ man ascii
  ‣ man man!
  ‣ Man pages are full of info, you just have to know where to look.
• Keyword search: man -k
• (Also Google)
Common command syntax

Program:

    ls

Arguments tell the program what to operate on:

    ls /home /usr

Options modify program behavior:

    ls -l

Options and arguments can be combined:

    ls -l -a /home
    ls -l --all /home
    ls -la /home
• Find the command that prints a sequence of numbers.
• Tell it to print the numbers 8 through 12.
• Now tell it to pad the numbers with zeroes so everything is the same length (08, 09, 10, 11, 12).
  ‣ Remember: `man -k` to search, `man` to view.
Editing text files

• Unix philosophy again: text is important!
  ‣ Configuration? Text files.
  ‣ Scripting? Text files.
  ‣ Programming? Text files.
• Vim (and vi)
  ‣ Does one thing and does it well
  ‣ Integrates nicely with other CLI tools like Make
Vim setup

```
sudo apt-get install vim
wget tiny.cc/311setup
sh 311setup
```
Understanding Vim

• Several different modes
  ‣ Indicator in bottom-left corner

• Normal mode (default)
  ‣ Navigate the file
  ‣ Most operations except typing text
  ‣ When in doubt, Esc gets you to Normal mode.

• Insert mode
  ‣ Typing text
Getting around in Vim

- Avoid those arrow keys!
  - Moving: h, j, k, l
  - Paging: Ctrl-f/b, Ctrl-u/d
  - Other keys work too

- Searching: /
  - (Regular expressions)

- Practice: edit /etc/services

- To quit without saving, type :q! and press Enter in Normal mode.
Making changes

• Getting to Insert mode
  ‣ i: insert at cursor
  ‣ A: append to this line
  ‣ o: open a line below this
  ‣ O: open a line above this
• Esc: back to Normal mode
• Also worth knowing
  ‣ u: undo
  ‣ Ctrl-R: redo
• Write and quit: :wq
Salutations!

• Change to your home directory
• Open the file yo.c in Vim
• Fix the program by adding
  \#include <stdio.h>
  at the top (hint: open a line above)
• Write and quit: :wq
  ‣ Shortcuts: ZZ to save and quit, ZQ to quit without saving.
Compile and run

• Run the compiler:
  
  gcc -o yo yo.c

• Execute the program:
  
  ./yo

• Edit it again
  
  ▶ Delete a printf line by moving to it and pressing dd.
  
  ▶ Now add another printf by using the o command.

• Save, quit, compile, run!
Basic shell scripting

• Eventually you’ll get tired of repeatedly typing:

  gcc -o yo yo.c
  ./yo

• Shell script = list of commands to run

• Put these two lines in a file called go.

• Change yo.c, then run:
  sh go
Learning Vim

• Many, many features
  ‣ Try everything, and find the ones that work for you
  ‣ :help
    ◾ :help user-manual

• Use it on your own system!
  ‣ GVim for Windows, MacVim for Mac

• Run vimtutor at the CLI
  ‣ Simple, hands-on introduction
  ‣ Do lessons 1.1–3.4 and learn the commands and concepts