

PHYS 3340 Computational Physics

Text: Computational Physics

Landau, Paez, Bordeianu

Wiley, ISBN: 978-3-527-40626-5

buy online (e.g., amazon)

Thomas Coan, 47 Fondren Science

Office hrs: Mon 10:30 am - 12:30 pm, or just find me.

coan@mail.physics.smu.edu

TA: Alex Liang, 41 Fondren Science

Office hrs: TBD

zliang@mail.physics.smu.edu

Computer Access

Lab 26: M-F 7am -10pm & Su ~1-5pm

Rm 101: M-F 7am -10pm & Su 1-5pm

If locked: bug Alex !

If locked: bug me !

If locked: bug grad students (rm 41 & 49)

If locked: bug secretary (rm 9 & 102)

If locked: bug ... physicsbody !

Overview

Main Goal: Learn tools to solve physics problems best solved w/ computers.

- This is NOT a programming course.
- Much programming instruction (aka “code”) is provided.
- Concentrate on algorithms not language grammar.
- Learn by doing. No bench warmers, get in the game!

Tools:

- Linux “operating system” (os = sw that runs computer)
- C++ programming language (choice pieces, not the entirety)
- Canned sw + various utilities
eg, MATLAB, Root, gdb, gnuplot, makefiles, ...

Course Administtrivia

http://www.physics.smu.edu/~coan/3340_09.html

Homework via email. (hey baby, we're in the electronic era)

Grading: 70% hw, 25% longer project(s), 5% quizzes.

NO final examination.

Big Picture Syllabus



1st 5 weeks Crash course in Linux operating system and C++

- “Survival” Linux OS commands
- gedit text editor and C++ compilation
- Choice elements of C++
- Smattering of useful utilities: gnuplot, gmake, gdb, ...

Last 9 weeks The good stuff: Physics simulations.

- Realistic trajectories and orbits (intro to solving diffeqs)
- Radioactive decay (intro to random processes)
- Particles in a quantum well
- Chaos
- Waves in 1-Dim and Fourier analysis
- Data fitting
- ...

Getting Started (...the hardest part...)

Enter SMU ID and initial password.

Left-click terminal icon on top of screen. This produces a “terminal window”

Type `passwd` Enter old password and then new one when prompted.

Linux commands are terse, to minimize typing. Often hard to guess for novice.

Try each of the following commands. Notice what they do!

Beware! Commands are **case sensitive** (a general Linux feature)!!!!

<code>ls</code>	list directory contents (ie, list files)
<code>pwd</code>	print (to terminal window) “working” (current) directory (folder)
<code>cd</code>	change directory. Eg Usage: <code>cd bin</code> (bin is a directory name)
<code>cd ~</code>	change to your home directory (ie, where you start at login time)
<code>mkdir</code>	make a directory. Eg usage: <code>mkdir trash</code>
<code>rmdir</code>	remove (“delete”) directory. Eg usage: <code>rmdir trash</code>
<code>cat</code>	displays contents of a file: e.g., <code>cat .bashrc</code>
<code>rm</code>	remove (“delete”) a file. Eg usage: <code>rm junk1</code>
<code>man</code>	manual (help) page. Eg usage: <code>man ls</code>
<code>date</code>	displays on screen the current date and time

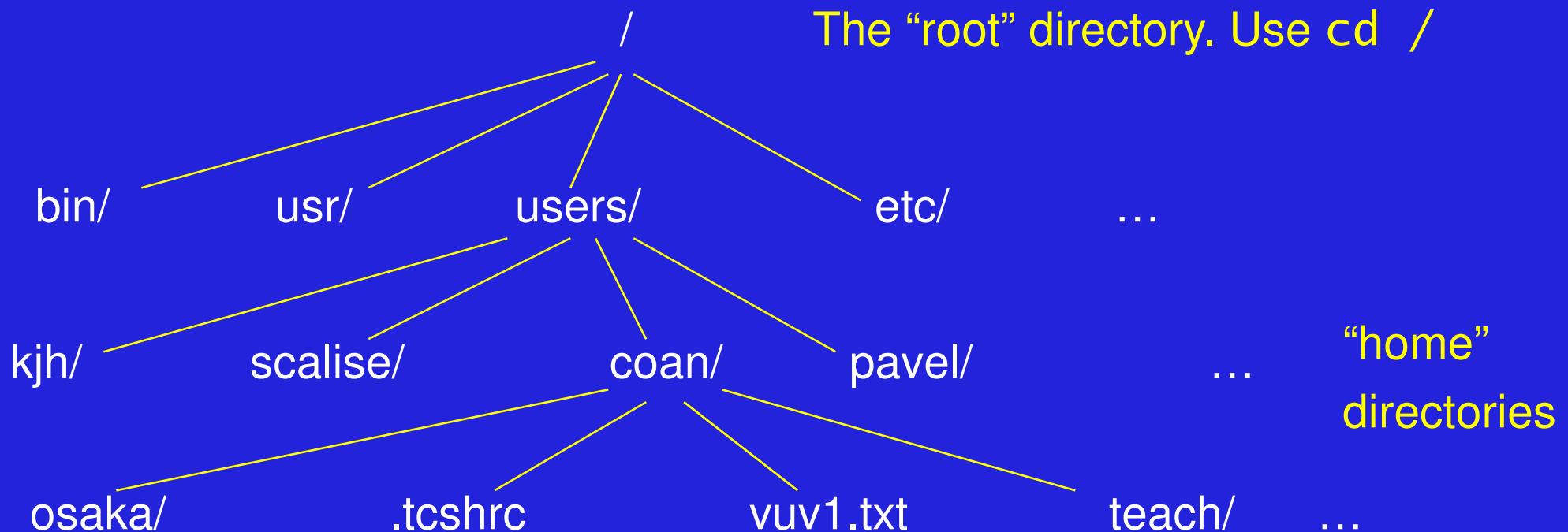
Linux File System Structure

It helps to know where files “live.”

Linux has a tree-like “directory” structure.

A directory is a file that contains other files (i.e., like a Windows folder).

Use `cd` to “drive” through structure.



“Driving around” in Linux

More on driving through file structure.

Distinguish between absolute “pathnames” and relative “pathnames.”

`cd ~/bin` Move to directory bin below your home directory (try it!)
Doesn't matter where you currently are in file system.

`cd ~` Move directly to your home directory (try it!)

`cd /usr/local/lib` move to directory /usr/local/lib (try it!)

`cd ..` move up to directory immediately above you (try it!)

`cd ./etc` you should be in /usr/local/etc (try it!)
. stands for the current directory. Useful shorthand!

`cd ~/..` What does this do? (try it!)

- Stands for the current directory. Useful shorthand.
- Stands for the parent of the current directory. Also useful.

Technical Interlude (Don't Panic)

Need to perform some Linux magic. The next steps will seem obscure.
We need to change your “working shell.”

The “shell” is the set of commands that you type at a screen to get the computer to do what you want. See Rubin for more detail.

Download and save to the Desktop the file `.tcshrc_3380`
Found on the downloads page of the course home page.

`cd ~` move to your home directory.

`cd Desktop` move to Desktop directory

`cp .tcshrc_3340 ~/.tcshrc` note the spelling !!

`tcsh` Technical mumbo jumbo for now.

`source .tcshrc` do NOT forget the .

Important Home Directory Files

Linux allows you to configure your “working environment.”

The working environment is roughly the look and feel of your login session.

Two important home directory files: `.login` and `.tcshrc`

`.login` contains commands and “aliases” (abbreviations). The `.` is crucial!
Executed once per login session.

`cd ~` move to your home directory.

`cat .login` spits out contents of `.login` onto screen/window.

`cat` command to list contents of non-directory file. Important!
Supposed to stand for catenate. (Seems a bit obscure.)

`source /users/std_usr/.login`



Yet another `.login` file. This is ok.

(Useful) command to execute commands/aliases in file that follows it.

Important Home Directory Files (2)

Linux allows you to configure your “working environment.”

The working environment is roughly the look/feel of your login session as well as values of important variables that affect command execution.

➤ Two Important home directory files: `.login` and `.tcshrc`

`.login` contains commands and “aliases” (abbreviations). The `.` is crucial! Executed once per login session.

`cd ~` move to your home directory

`cat .login` spits out contents of `.login` onto screen/window

`cat` command to list contents of non-directory file. Important!
Supposed to stand for catenate. (Seems a bit obscure.)

`source /users/std_usr/.login`

Yet another `.login` file. This is ok.

(Useful) command to execute commands/aliases in file that follows it.

~/.tcshrc file

~/.tcshrc is the other important file. Note the leading period . in the file name.

~/.tcshrc is executed every time you open a window/terminal.

Look at **your** .tcshrc file. (Try it!) What do you see? What is its structure?

```
set path = ( . ~/bin /usr/local/bin /bin ... )
```

Sets the important “shell” variable `path` to show what directories are searched for to execute a keyboard command **and** the search order.

```
set cdpath = ( . .. ~ )
```

Set the important “shell” variable `cdpath` to show what directories, and their order, searched in to execute, eg., `cd some_stupid_dir`
If `some_stupid_dir` not in `.` or `..` or `~` , an error msg is displayed.

```
set noclobber      shell variable acting as switch. More later.
```

Linux tutorial

One stop shopping for a decent Linux online tutorial:

<http://www.ee.surrey.ac.uk/Teaching/Unix/index.html>

Says “Unix” but ok for Linux.

Repeats/expands on what is said here.

Adds additional info.

Easily digestible (key features, non-exhaustive).

Read “.cshrc” as “.tchsrc” .

➤ Read tutorial at home (# 5 & 7 are a bit much for now)

➤ Useful to skim through glossary (app B) in Rubin et al.

Summary

Linux has 10 - 15 “must know” commands.

Linux has 10 - 15 “good to know” commands.

Linux has tree-like directory structure. / is the “root.”

.login and .tcshrc : important home directory files.

Distinguish quotation marks: “ v. ‘ v. `

Output redirection: > or >> or >&

➤ Read (and perform!) Linux tutorial

Don't suffer in silence. Scream for help!!!

