scds:~$ command line intro

who needs a mouse?

today's plan

command line computing briefly explained
shell/terminal client familiarity
logging in
basic syntax
moving around
creating directories and files
text editing
changing permissions
logging out

command line computing

not programming or scripting

Linux = thousands of tiny programs

computing minus the GUI - no mouse, all text, although some mousing can be useful

fast, lean, powerful

shell or terminal client

Windows - PuTTY (many options)

Mac OS - Terminal

Linux - xterm (many options)

host

PuTTY
[redacted - contact Dale for address]

Terminal, xterm, et al.
ssh [redacted - contact Dale for address]

must be using MacSecure!

basic syntax

sometext:~$ grep -i “textstring” filename

command prompt

program

flag/option

arguments

where am i? who am i? what's here?

pwd - print working directory

id - user and group memberships

cd - change directory

ls - directory listing

try out pwd and id

moving around

cd go home

cd .. go up

cd ../../.. go up that many levels

cd - go to the most recent directory

directory listing

ls shows files in the current directory

ls -al shows all files (including hidden) and puts them in a vertical list with details

ls -al |more is useful if the directory has many items

help

For any Linux program, type:

man <name of program>

e.g.-

man ls

and be dazzled by the array of options!

making directories

mkdir <name of directory>

grabbing files

curl

curl http://someurl.com displays it to the screen (be careful)

curl http://someurl.com > somefilename saves the contents to that file

wget

wget http://someurl.com/... saves target to a file

curl is a scalpel, wget is a hammer

making files, aka text editing holy wars

vi vs. emacs vs. pico vs. ...

Pick your own poison, for today vi

vi <desired filename>

i to insert

:w

:wq

:q!

permissions

ls -al will reveal permissions, e.g.-

-rwxrwxr-x 1 owner group 4096 Nov 20 15:39 filename

-rwxrwxr-x = user group world

r = read

w = write

x = execute

changing permissions

chmod

Can use symbolic or octal arguments

Symbolic

chmod u=rwx assigns user read/write/execute

Octal

chmod 775 assigns rwxrwxr-x

permissions tips and caveats

no write for world, unless you mean it

scripts, programs require executable bit

can change permissions recursively

can change permissions using wildcards such as *.* and *.html

miscellaneous tips

arrow up to see previous commands

use tab to complete file or directory names

fancy: edit shell profile to create shortcuts

taking out the garbage

rm and **rmdir**

use with great caution, and rtfm first

no undo, although files can be recovered

logging out

type **exit** or **logout** at prompt