

Slide 1

Administrivia

- Homework 1 on the Web, linked from the “lecture topics and assignments” page. Due next Wednesday (11:59pm).
- Notice that usually you will turn in homework by e-mail. Please send from your Trinity e-mail and put in the Subject line which assignment *and which course*, to be sure I put it in the right folder for grading.
If you have *questions* about the homework, be sure to say that in the Subject line — and it doesn’t hurt to add “urgent”.

Slide 2

Getting Started With the Command Line

- What you get when you start a terminal window is a “command shell”, similar to Windows’ “MS-DOS prompt”.
Rather than pointing and clicking, you type the name of the program you want to run, plus whatever arguments (parameters) it needs.
- (Why would you want to use a command line? because for some things it’s arguably more efficient, and it’s “scriptable” in ways that GUIs typically aren’t.)
- Let’s try some commands . . . (Don’t worry if this goes by quickly — you should plan anyway to spend some time outside class trying out what we do in class and what’s in the “reading”.)

Slide 3

Some Commands

- `pwd` shows the current directory. (When you give a filename, it's relative to this directory unless you give a full pathname.)
- `ls` lists the current directory. Add `-l` to get more information.
- `cd foo` changes to directory `foo`. Just `cd` goes back to your home directory. Try `cd Local` and then `ls`.
- `mkdir foo` creates a director `foo`. Might be useful to create one for your files for this class (call it `csci1312`, maybe, or `CS1`).
- `passwd` changes your password. (Not a command you'll want often, but probably now!)

Slide 4

Useful Command-Line Tips

- The shell (the application that's processing what you type) keeps a history of commands you've recently typed. Up and down arrows let you cycle through this history and reuse commands.
(Pedantic aside: "The shell" here means the one you're most likely to be using. There are other programs with similar functionality you could use instead.)
- The shell offers "tab completion" for filenames — if you type part of a filename and press the tab key, it will try to complete it.
- To learn more about command `foo`, type `man foo`. This is reference information rather than a tutorial, but usually very complete. `man -k foo` will give you a list of commands having something to do with `foo`.

Remote Access

Slide 5

- One of the strengths of a command-line environment is that it works well for “remote access” (using the computer when you aren’t sitting in front of it).
- To do this from another UNIX-like computer, use `ssh`. `scp` and `sftp` can be used to copy files.
- From a Windows computer, install either Cygwin (UNIX-like toolkit) or PuTTY (terminal emulator).
- E-mail sent Friday saying more about PuTTY (available via ITS’s Windows VDI) and machine names to use. DiasNN may be best for this class — not up-to-date right now, but good enough for this class, and Linux-only.

Text Editors

Slide 6

- Many, many text editors, and people have favorites. Notepad is an example from the Windows world.
- I use and will teach in this class `vi`: It’s found on every UNIX/Linux system I know of, and is very powerful, though it takes some getting used to. (`vi` on our Linux machines is actually `vim`, a more featureful “clone” of the original `vi`.)
- Other popular Linux text editors include `emacs`, `pico`, and `gedit`.
Advice: Give `vi` a real try first, but if using it is just too painful, use something else!

vi Basics

Slide 7

- `vi` has two *modes* — insert mode (where what you type goes into the file) and command mode (where you can type commands to copy, move, delete, save, etc.).
- You start an editing session by typing, e.g., `vi hello.txt`. It starts in command mode. Enter insert mode by typing `i`. Exit by pressing `ESC`. Move around with the arrow keys. (Try entering some text.) Delete a single character with `x` (in command mode).
- Save by typing `:w`; exit by typing `:q`.
- *Highly recommended:* `vimtutor` brings up an interactive tutorial. (Homework 1 asks you to try it.)

vi Tips — Errors/Mistakes

Slide 8

- `u` means “undo” the previous action (insertion, deletion, paste). Repeat to undo multiple actions.
- `:q!` exits without saving. Useful if you make a complete mess of things.

More Commands

Slide 9

- Now that we have a way of creating files, we can try out some other basic commands.
- `cat` to show contents of a file. `more` or `less` to show it a screenful at a time.
- `cp` to copy one file to another. `mv` to move or rename a file. For both, `-i` to warn about overwrites.
- `rm` to delete a file. (Note — no recycle bin, so use with caution! or `-i` to prompt.) For this and `cp` and `mv`, `-v` shows what's being done.
- Other useful/interesting commands in video lectures and next time.

UNIX Filesystem Basics

Slide 10

- Unlike in Windows (and Mac, usually), UNIX filesystems are case-sensitive (so `hello` and `Hello` are different files).
- Files have two levels of ownership — “owner” (user) and “group”. Groups allow sharing files with some but not all users.
- File access is controlled by “permissions”. Three levels (owner, group, and everyone else), three types of access (read, write, execute).
- `ls -l` shows permissions. `chmod` changes them.

Minute Essay

- If you've used a command-line environment, how does this one compare to what you've used before?
- Anything surprising or startling today?

Slide 11