Slide 1

Administrivia

- Project hint: Think of something you do often that seems repetitive and tedious and automatable.
- Be advised that I'm teaching "Unix system administration" next fall.

Slide 2

"What Command Do I Use To . . . "

- You know about `apropos` as a way to discover new commands. You probably also know that it's not perfect.
- So today, a tour of some commands I have found useful . . .

Slide 3

Commands for Working With Text and Other Data

- `script` to capture all terminal input/output. (`exit` to stop capturing.) Includes any control characters, all keystrokes (including backspaces and tabs for completion), so output looks strange viewed with `vim` or even `less`. `cat` displays it correctly. `export TERM=dumb` (after `script`) helps some.

  Can also use as a crude "view what another user is doing" by combining `script -f` and `tail -f`.
- `strings` to search a file for printable strings.
- `grep` to search a file or files. Can search on regular expressions too.
- `diff` to compare two files. Also try `vimdiff`. (I like `vimdiff -o`.)
- `ispell` or `aspell` to check/correct spelling. (No `man` page on our machines. `locate` to find documentation.)

Slide 4

- `od` to show data in various forms (binary, hexadecimal, etc.).

Slide 5

Commands for Working with Filenames

- `basename` to get file's "base name" (without directory and (optionally) extension).
- `dirname` to get file's directory name.
- Both can be useful in writing scripts — recall that you can "recycle" the output of one command in another by using backquotes.

  Example — if you don't have a `rename` command, can write your own.

Slide 6

Commands for Printing

- `lpr` to print PostScript or text. `lpq` to check print queue; `lprm` to cancel a print job. (May not work if `lpr` sends output to printer managed by another computer.)
- `enscript` or `pr` to pretty-print text.

Slide 7

Commands to Compress and Archive Data

- `gzip` and `gunzip` to compress/uncompress data. Or use `compress` and `uncompress`.
- `tar` to create Unix-standard-format "archive" file. (Conceptually similar to ZIP archive files — which you can generate, using `zip`.)

  A sometimes useful sequence (copies a directory, preserving any symbolic links):

```
(cd sourceDir; tar cf - . ) | ( cd target; tar xf - )
```

Slide 8

Web-Related Commands

- `wget` or `curl` to download a Web page or pages.
- `lynx`, `links`, or `w3m` to browse in text mode.

Slide 9

Commands for Working with Numbers

- `bc` and `dc` calculators. `dc` uses RPN (somewhat strange) but is arbitrary-precision, which allows working with very large integers.
- `gnuplot` to generate plots.

Slide 10

Commands for "Batch" Programs

- `batch` and `at` to run something "in batch mode" / at specified time. If output is not redirected, it's sent to you by e-mail.
- `crontab` to set up "cron job" to execute periodically.

Slide 11

Commands for Working with Programs

- `-E` (show preprocessor output) and `-S` (generate assembly-language output) flags on most compilers.
- `gdb` source-level debugger. Semi-graphical version available from `xemacs`.

Slide 12

Mail on Unix Systems

- Full discussion beyond the scope of this course. Let's talk about some basics / things of interest to users.
- But first, a bit of terminology:
  - MTA ("mail transport agent") — program that delivers mail. Choice made by sysadmin. A well-known one is `sendmail`.
  - MUA ("mail user agent") — program users use to read mail, send mail, etc. Many choices.

Slide 13

Mail Delivery

- Normally, mail gets delivered to the system "mail spool". (For `@cs.trinity.edu` addresses, on Sol.)
- To forward mail elsewhere, create a text file `.forward` in your home directory. In it put the forwarding address(es). If one of them could create forward-to-the-same-machine loops, specify address with backslash at start, e.g., `\username@sol`.

Slide 14

Reading Mail

- There are lots of programs you can use to read mail (MUAs). By default, many read from local mail spool. For us, this means you would have to run them on Sol — no longer allowed for performance reasons.
- Many MUAs, though, also allow reading from server (for us, Sol) using POP3 or IMAP. Also, `fetchmail` can be used to fetch mail using one of these protocols. Probably works best if run from your own machine.
- Or you can use `.forward` file or `procmail` (more later) to put mail in a file in your home directory and use any MUA to read it from there.
- Worth looking at text-mode MUAs — often very configurable/scriptable. `mutt` and `pine` both seem good.

Slide 15

Sending Mail from the Command Line

- Simplest / most primitive program for sending (and reading) mail is `mail`. Pretty reasonable for sending pre-composed text-only messages.

  Example: `echo "this is a test" | mail -s "test" bmassing@cs.trinity.edu`
- What about attachments? `mail` doesn't really "do" MIME. Workarounds:
  - Encode files to attach with `shar`. Recipient pipes message body through `unshar`.
  - Encode files to attach with `uuencode`. Recipient pipes message body through `uudecode`.
- Other text-mode MUAs (e.g., `mutt` and `pine`) are also "scriptable" and understand MIME.

  Example: `echo "here is my file" | mutt -a somefile -s "my file" bmassing@cs.trinity.edu`.

Slide 16

Filtering Mail with `procmail`

- `procmail` can be used for many kinds of "filtering" operations on mail. Selected messages can be saved (to files), forwarded, automatically replied to, or passed to other programs.
- On many Unix systems, you make this happen via a `.forward` file. On RH/FC Linux systems, it happens automatically if you have a file `.procmailrc` in your home directory.
- Syntax for `.procmailrc` can be intimidating, but `man` pages for `procmail`, `procmailrc`, and `procmailex` have examples that can help.
- One use of `procmail` is to run all incoming mail through a spam-filtering program, such as `spamassassin` (installed on Sol).

Slide 17

A Little About X ("The X Window System") — Basic Ideas

- Some operating systems include GUI support in the "kernel". Unix takes a different, layered approach. Slower, but safer and more flexible.
- Basic idea — separate processing from GUI and allow them to be on the same computer or different computers — "X client" versus "X server".

Slide 18

A Little About X — Basic Ideas, Continued

- X "client(s)" are programs that want to do GUI input/output — e.g., Mozilla, gv, etc.
- X "server" manages display, accepts input. Can be a process running alongside clients, or a whole operating system (for an "X terminal"), or an application running on a different operating system ("X server/emulator for Windows").
- X defines protocol for client/server communication.

Slide 19

A Little About X — Application Programs

- How to write a program with a GUI? Can make calls to X library functions directly — set up window(s), main processing loop to handle "events". (Example.)
- Or can use a higher-level "widget set" (buttons, menus, etc.): Motif, GTK, etc. (Why several? Well, this is Unix.)

Slide 20

A Little About X — User Interface

- Separate "window manager" controls how user interacts with windows — how they're arranged on the screen, how the user moves them around, etc. Examples include `twm`, `fvwm`, Window Maker. (Why several? You know.) `switchdesk` provides limited ability to change window manager. To tweak further, edit appropriate dot-something files in home directory.
- In addition, can have a "desktop environment" that provides additional features. Examples include CDE (Sun), KDE, Gnome. Desktop environments provide something that looks more like Mac/Windows interface, but at a performance cost.
- A somewhat (but not very) extreme view: "A window manager is a mechanism for managing xterms."

Slide 21

A Little About X — Tips and Tricks

- To copy and paste text — highlight with left mouse button, paste with middle mouse button. Works with all "standard" X applications.
- Can start a second X server on Linux machines via:

  `X :1 -query machine -once`

  Switch back and forth with control-alt-F7/F8.

  (Warning: There are rumors that this doesn't work right on some lab machines. Supposedly a problem with driver for monitor.)
- Lab machines set up now so that when you `ssh` to another machine and run an X-using application, it automagically displays on your screen. Formerly, you had to set the `DISPLAY` environment variable and export it.

Slide 22

Minute Essay

- How are you doing with the homework? that is, as of 5pm today, what will you have left to do?
- Reminder: Homework 6, 7 due today. (But if you're swamped, okay to turn in later.)