Slide 1

Administrivia

- (Review collaboration policy on syllabus.)
- Homework 1 status — on Web, due next Monday.

Slide 2

Reading The Fine Manuals

- One of the most useful things you can learn is how to learn more. Documentation on UNIX systems is not always perfect, and it's not particularly novice-friendly, but usually it's thorough.
- Places to look:
  - `man` pages.
  - `info` pages.
  - Elsewhere on the system. `locate` on Linux may help.
  - The Web, via your favorite search engine.
  - Usenet, including Google's archives (click "Groups" from Google's main page).

RTFM — `man` pages

Slide 3

- Reference documentation (as opposed to tutorials).
- Organized into “sections” (user commands, sysadmin commands, library functions, etc.).
- Of particular interest is the section `SEE ALSO`.
- `man -k` (or `apropos`) to search for command names.
- Try `man man`...
- Now you might want to know about `more`, or `less`.

RTFM — `info` pages

Slide 4

- Also reference documentation, sometimes more current / complete than `man` pages. (Why are there are two systems? Probably historical reasons!)
- Organized in a way somewhat similar to hypertext.
- Try `info info`...

Slide 5

Other Useful Info-Gathering Commands

- `whereis`.
- `type`.
- `file`.
- `which`.

Slide 6

A Little About Files

- A key underlying concept — “everything’s a file” (sequence of bytes). Directories are files. Devices are represented as “special files”. Many files are text.
- Things to note:
  - Windows/DOS “extensions” idea doesn’t really apply.
  - Also no notion of “drive letters” — all paths form a single hierarchy. Removable media can be “mounted”.
  - Security model is simple but fairly flexible — rights (read, write, execute) for owner, group, others.
  - “Links” (hard or soft) allow non-tree directory structure.
- Be familiar with basic commands to manipulate/navigate filesystem.

Slide 7

A Little About Processes

- Another key concept — process as one of a set of “concurrently executing” entities (users, applications, etc.)
- Things to note:
  - Processes can spawn “child” processes. (This happens, e.g., every time the shell runs a command.)
  - Processes can have “environment variables”, inherited by child processes. Examples — `USER`, `PATH`.

Slide 8

Minute Essay

- Something of a trick question: `man printf` doesn’t give you the `man` page for the C library function `printf`. What are two ways you *can* get that `man` page?

Slide 9

Minute Essay Answer

- `man -a printf` or `man 3 printf`. (You could also use `man fprintf`.)