

Slide 1

Administrivia

- About the reading: What may work well is to wait until after class, and then focus on things mentioned in class while still at least skimming other material (might be something that would interest you!).
- (Homework 1 coming soon — Wednesday?)

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.
 - And of course the Web, via your favorite search engine!

RTFM — man pages

Slide 3

- Reference documentation (as opposed to tutorials).
- Organized into “sections” (user commands, sysadmin commands, library functions, etc.). Can have entry with same name in multiple sections. `-a` option or section number gives access to non-default.
- Of particular interest is the section `SEE ALSO` — sometimes lists other related commands.
- `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...`

Other Useful Info-Gathering Commands

- `locate` (fast file-search by name, Linux but maybe not other UNIX).
- `whereis` (finds files associated with a command).
- `file` (tries to identify file contents).
- `which` (looks for command name in search path).

Slide 5

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.
- 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” (incorporated into the hierarchy) and “unmounted”.
- We won’t review basic commands for navigating and manipulating the filesystem, but you should if you’ve forgotten (`cd`, `ls`, `cp`, `mv`, `rm`, `mkdir`, `rmdir`).

Slide 6

A Little About Files – File Permissions

Slide 7

- Security model is simple but fairly flexible — rights (read, write, execute) for owner, group, others; “sticky bit” provides a few more things.
- `r` and `w` have obvious meanings; `x` means “can execute” for files, “can `cd` to” for directories. Normally creating or removing files requires write access to *directory*.
- “Sticky” bit is — interesting? For directories (`t`), means files can be removed only by owner (example `/tmp`). For files (`s`), means program executes with the permissions of the file owner (example `/usr/bin/passwd`).

A Little About Files – Links

Slide 8

- “Links” (hard or soft) allow non-tree directory structure.
- “Soft” (symbolic) link (`ln -s`) is just a special type of file pointing to another file. Allows access through either name, but can “break” if pointed-to file isn’t there.
- “Hard” (non-symbolic?) link (`ln`) only works within a filesystem but creates a second directory entry to the same underlying file. File itself exists until all (hard) links to it are gone.

A Little About Processes

Slide 9

- Another key concept — process as one of a set of “concurrently executing” entities (users, applications, etc.)
- Processes can spawn “child” processes. (This happens, e.g., every time the shell runs a command.) Child process cannot change anything in parent (so, e.g., if you `cd` in a script, it only affects the script, not the caller).
- Processes can have “environment variables”, inherited by child processes. Examples — `USER`, `PATH`.

Minute Essay

Slide 10

- Anything today particularly unclear, or that you want to hear more about?