

Slide 1

### Administrivia

- Homework 3 to be available on Web soon.

Slide 2

### A Few Words About Backups — Policy

- Why do backups at all? because sometimes things go wrong, either hardware/software failure or user error. Up to you to decide whether the potential payoff justifies the hassle. (Maybe you don't really need a copy of last semester's homework. But how about the information you need to bill people for your consulting services?)
- What to back up? Typically, only data that can't easily be restored/recreated.
- How often to do backups? Depends on how current the backup data needs to be — is it okay to lose a day's work? a week's work?
- Where to store backups? If the data's really important, at least one copy needs to be in a different location. Also keep in mind security considerations.
- Also a good idea to consider doing backup before making changes.

### A Few Words About Backups — Mechanism

Slide 3

- Many choices for backup medium — floppy disk, removable disk, tape, CD. In deciding which one, consider storage requirements, cost, reusability, permanence.
- Traditional commands for doing backups/restores are `dump` and `restore`. `dump` allows dumping only selected files/directories, only recently-changed files, multivolume output.
- Other commands that could be used — `tar` (also good for bundling up a bunch of files), `cpio`, `dd`.
- There is also free/commercial software to automate backups.

### Just a Few More Words About Backups

Slide 4

- A useful command for keeping versions of file "in synch" — `rsync`.

## Logging and Log Files

Slide 5

- Log files record activity by system background processes (daemons) and also record some startup activity. Useful for resolving problems, including security issues. Typically use `syslogd` facility, configured in `/etc/syslogd.conf`.
- Where they're kept varies from system to system. Linux keeps them in `/var/log`. Many use `syslogd` to manage logging.
- To keep these files from eventually taking over all available space, useful to "rotate" log files. Some systems provide facilities for doing this (e.g., `logrotate` on Linux); otherwise you can write scripts to be run as `cron` jobs.
- To get the most use out of these files, probably a good idea to run something that scans them regularly and reports potential problems, usually by e-mail to `root`. Some suggestions in book; also `logwatch`

## Installing and Updating Software — "Tarballs"

Slide 6

- Traditionally, Unix software distributed in the form of a "tarball" (archive created by `tar`, possibly compressed, usually containing source). Still often available and useful — e.g., to install in your home directory.
- What do you do with a tarball? Typical installation goes like this:
  - "Untar" the file (`tar xf`). Usually creates a directory, often containing `README` and/or `INSTALL` files — which you should review.
  - Run `configure` script to set system-specific options. Usually figures most things out for itself, but may need/allow user input, either via command-line options or standard input.
  - Run `make` to compile, etc. Normally puts created files in the same directory.
  - Run `make install` to move/copy executables, etc., to system directories. Notice that this is the only step that requires root privileges.

Slide 7

### Installing and Updating Software — Package Managers

- Installing from source (provided as a tarball) can be a pain, and doesn't help manage dependencies among different s/w packages.
- So, most Linux distributions include a “package manager” that automates installation (usually from precompiled binaries) and manages dependencies:
  - Red Hat package manager — `rpm` command.
  - Debian package manager — `dpkg` and `apt-get` commands.
- Similar functionality available for other Unix systems, though less standardized. (E.g., Google search on “Solaris” and “package manager” turned up a page at University of Florida listing several.)
- Probably a good idea to use these if possible — package manager's database then keeps track of state of system, can manage dependencies.

Slide 8

### Installing and Updating Software — Beyond Package Managers

- Many Linux distributions now also come with a way to (semi-)automatically find updates to all packages you have installed.
- `yum` and `up2date` build on `rpm`.
- The Debian package manager apparently includes this via `apt-get`.

## Minute Essay

- None — sign in.

Slide 9