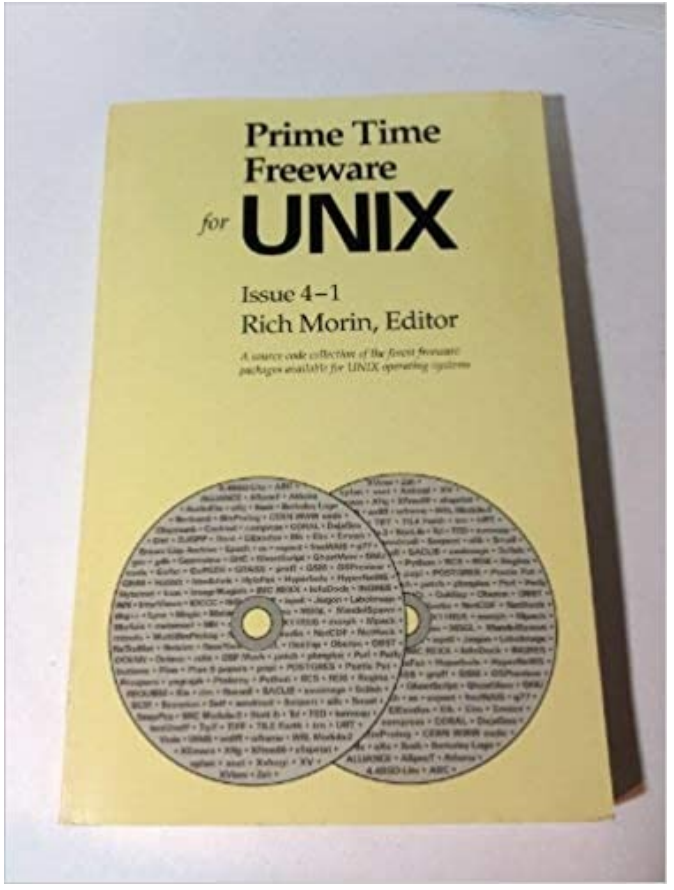
Package Management

Linux System Administration
Spring 2023

In the beginning...

- Software on CD
- **autoconf** to deal with OS & compiler peculiarities
- Build from source

```
tar xzf apache_1.3.6.tar.gz
cd apache_1.3.6
./configure --prefix=/usr
make
sudo make install
```



Then came package managers

- Pre-compiled binaries and configuration files
- Checks shared library dependencies
- Specialized archive to include config scripts
  - Redhat rpm (.rpm)
  - Debian dpkg (.deb)
- CD or download



Package manager systems

- With the internet came repositories
  - Download on demand
- Package management systems download the package as well as its dependencies
  - Red Hat **yum** (now called **dnf**)
  - Debian **apt-get** (also **apt**)
  - Arch **pacman**
  - OSX **homebrew**
  - BSD **pkg**

Advantages

- Simplifies downloading and installing packages
- Facilitates management of dependencies
- Automate the process of updating packages
- Supports cleanly removing packages
  - but not necessarily dependencies

Package Repositories

- Collection of packages for a distribution
- Accessed via HTTP/HTTPS
- Mirrors spreads bandwidth load
  - Update mirrors with rsync
  - Many sites run a local mirror
- Repositories may have multiple distros and versions

RedHat EPEL

- Extra Packages for Enterprize Linux
  - Additional packages not included in RHEL
  - Adds more bleeding edge packages from Fedora
- **dnf install epel-release**
  - Add EPEL as an additional repository

Repository Configuration

- RedHat
  - `/etc/yum.repos.d/*`
    - dnf still uses yum.repos.d
  - Rocky
    - `/etc/yum.repos.d/rocky.repo`
  - EPEL
    - `/etc/yum.repos.d/epel.repo`
- debian
  - `/etc/apt/sources.list`
  - deb *URL release* main

Package naming

- RedHat
  - package-version.os.arch.rpm
    - `openssh-8.7p1-8.el9.x86_64.rpm`
    - `perl-base-2.27-479.el9.noarch.rpm`
- debian
  - package_version+os_arch.deb
    - `openssh-client_8.4p1-5+deb11u1_amd64.deb`

Making sure all packages are current

- RedHat
  - `dnf update`
    - check metadata
    - check for updates
  - Confirmation defaults to NO
- debian
  - `apt-get update`
    - check metadata
  - `apt-get upgrade`
    - check for updates
  - Confirmation defaults to YES

Installing **dig**

- Redhat

  `dnf provides dig`

  - bind-utils

  `dnf install bind-utils`

  - Dependencies installed

    bind-libs
    bind-license
    fstrm
    libmaxminddb
    libuv
    protobuc-c

- debian

  `apt-file search /usr/bin/dig`

  - bind9-dnsutils

  `apt-get install bind9-dnsutils`

  - May already be installed

Installing a downloaded package

- `rpm -i foo.rpm`
- `dpkg -i foo.deb`
  - Will fail and show dependencies not met
- `dnf install ./foo.rpm`
- `apt-get install ./foo.deb`
  - Will fetch dependencies as needed

Finding out about packages

- Redhat
  - Find string in package name or description
    - `dnf search` *`string`*
  - Package metadata
    - `dnf info` *`package`*
- debian
  - Find string in package name or decsription
    - `apt search string`
  - Package metadata
    - `apt show package`

What package installed **dig**?

- RedHat

  ```
  rpm -qf /usr/bin/dig
  ```
  bind-utils-9.16.23-1.el9.x86_64

- debian

  ```
  dpkg -S /usr/bin/dig
  ```
  bind9-dnsutils: /usr/bin/dig

What files were installed by **bind-utils**?

- RedHat

```
$ rpm -ql bind-utils
...
/usr/bin/arpaname
/usr/bin/delv
/usr/bin/dig
/usr/bin/dnstap-read
/usr/bin/host
/usr/bin/nslookup
/usr/bin/nsupdate
...
```

- debian

```
$ dpkg-query -L bind9-dnsutils
...
/usr/bin/delv
/usr/bin/dig
/usr/bin/dnstap-read
/usr/bin/mdig
/usr/bin/nslookup
/usr/bin/nsupdate
...
```

What packages are installed

- RedHat
  - `rpm -qa`
- debian
  - `dpkg -l`
    - ii installed
    - un uninstalled

Removing packages

- Redhat
  - `dnf remove` *`package`*
  - `rpm -e` *`package`*
  - `dnf autoremove`
    - Removes obsolete
- debian
  - `apt-get remove` *`package`*
    - --purge removes configs
  - `dpkg -r` *`package`*
    - dpkg -P to purge
  - `apt autoremove`
    - Removes obsolete

Configuration files in packages

- Packages may contain configuration files
  - What happens if a new release changes the configuration file, but you changed the config?
- Redhat
  - Saves new file with .rpmsave or .rpmnew appended
- debian
  - Prompts to keep, replace, modify, etc.
- Generally you will need to re-apply your edits

Some sage advice

- When changing a configuration file, save a copy of the original
  - `cp -a foo.conf foo.conf.0`
  - diff shows changes
  - Helps to recover from failures
  - Reboot if appropriate
- RedHat does not enable packages automatically

Installing groups of packages

- Redhat
  - `dnf groupinstall “Development Tools”`
- debian
  - `apt install build-essential`
- Installs package collections
  - gcc, g++, make, binutuls, header files, and lots more

Automated installation

- Bare metal installation from a script
  - Used in data centers or compute clusters
- RedHat/Anaconda kickstart
- debian FAI (Fully Automatic Installation)