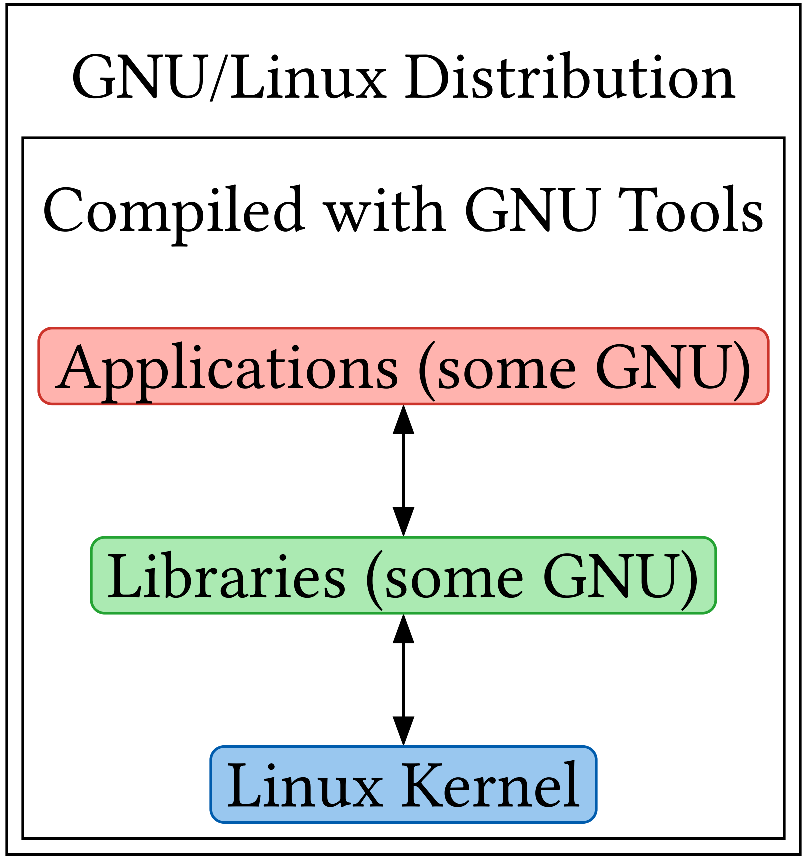
Package Management in Linux

Ryan Tolboom

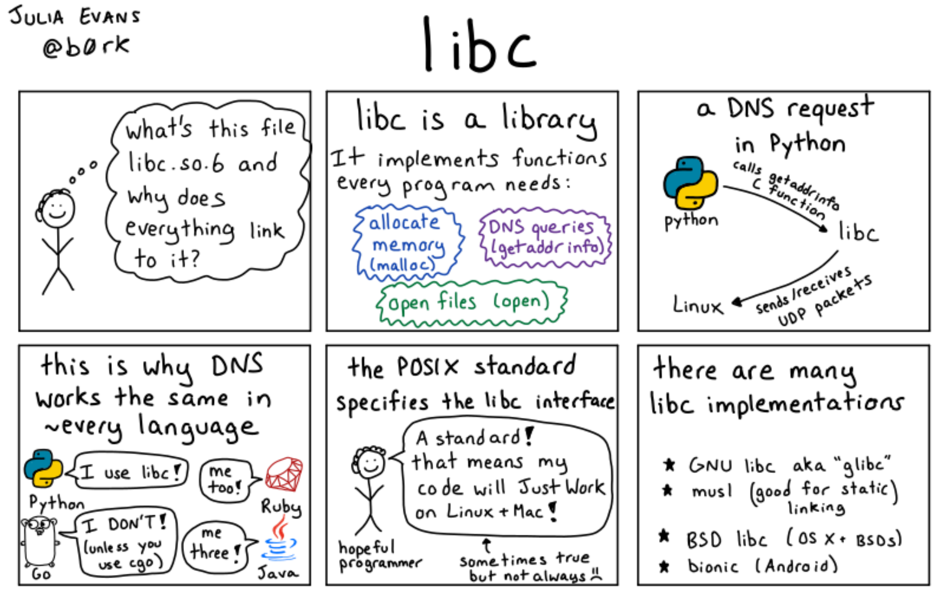
New Jersey Institute of Technology

What is Linux?

What is Linux?



libc



JULIA EVANS @b0rk
libc
what's this file libc.so.6 and why does everything link to it?
libc is a library
It implements functions every program needs:
allocate memory (malloc)
DNS queries (getaddrinfo)
open files (open)
a DNS request in Python
python
calls getaddrinfo C function
libc
Linux
sends/receives UDP packets
this is why DNS works the same in ~every language
Python
I use libc!
me too!
Ruby
I DON'T! (unless you use cgo)
me three!
Go
Java
the POSIX standard specifies the libc interface
A standard! that means my code will Just Work on Linux + Mac!
hopeful programmer
sometimes true but not always!!
there are many libc implementations
★ GNU libc aka "glibc"
★ musl (good for static linking)
★ BSD libc (OS X + BSDs)
★ bionic (Android)

- Standard C Libraries on Linux
- Used by all common UNIX tools
- Most common is glibc (GNU libc), but there is also uClibc, dietlibc, and musl libc
- Small libc libraries are increasingly popular in containers

Purpose

- Linux systems are made up of a large collection of software
- Package management makes it easy to upgrade/install/remove individual pieces of software
- What kind of software are we talking about?
- Dependency tracking is the largest hurdle and different distributions handle it in different ways

Parcel Parcels Packages is in the public domain

There are many different Linux distributions available. Let's look at how some of them handle package management.

Slackware

Slackware logo is in the public domain

- tarballs of files
- simplest and oldest
- no dependency tracking

Debian/Ubuntu

- dpkgs usually installed with apt or another front end
- allows scripting
- has *extensive* dependency management

Redhat/Rockylinux/Fedora

Red Hat logo is in the public domain

- RPMs usually installed with yum or another front end
- allows scripting
- has extensive dependency management

Gentoo

- source based ebuilds (BSD lineage)
- allows scripting
- has dependency management

Arch (BTW...)

"Arch Linux "Crystal" icon" by Jude Vinet, Aaron Griffin, and Levente Polyák is licensed under GPLv2

- PKGBUILD source system that creates binary packages
- packages can be installed with pacman
- allows scripting
- has dependency management

Rolling Releases vs. Point Releases

- Most distributions have point releases where MAJOR changes will be a new version and the old versions will continue to be supported for a certain amount of time (LTS)
- Arch and Gentoo use the rolling release model where each package is updated to the latest version as it becomes available

Why should I update?

- Security patches
- Mature software in a point release actually doesn't change that much
- What if something breaks?
  - Backups and filesystem snapshots
  - "Rollback" apt
  - Triage your updates
  - What if something *is going* to break?

Software Update is in the public domain

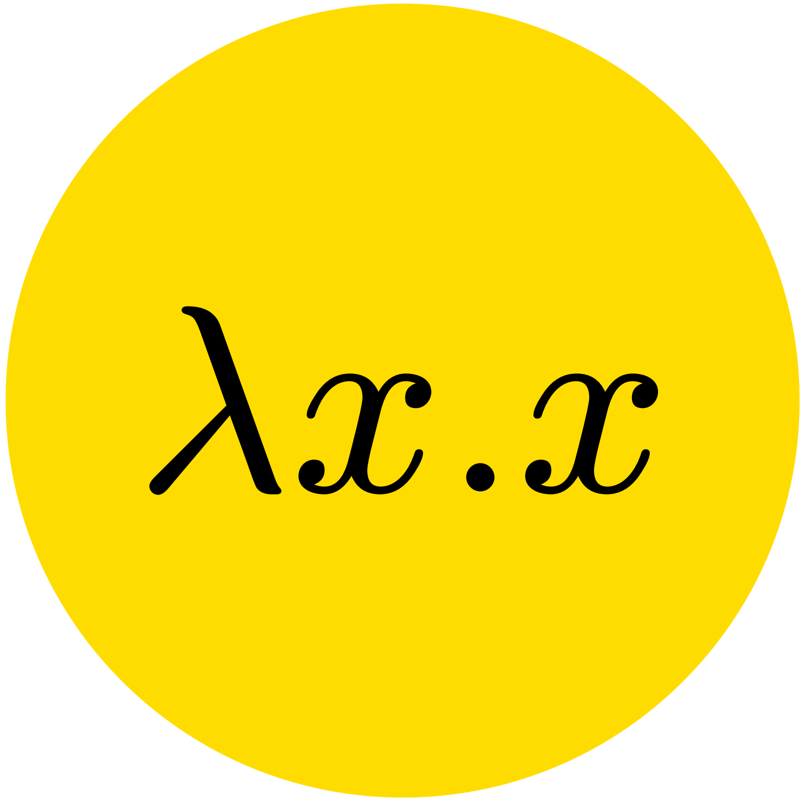
apt

Advanced **P**ackage **T**ool

- Used in Debian based systems to allow updating and installing packages from repositories
- Partly a front end for dpkg
- Most of the things you used to use apt-get for can now be done with the apt binary
- `apt update`: Download package lists from repositories
- `apt upgrade`: Upgrade packages from repositories
- `apt full-upgrade`: Remove packages if needed to make dependencies work (used to be `apt-get dist-upgrade`)

Functional Package Management

- A relatively new way of doing things where you can have *multiple* versions of packages in isolation. Each package has *exactly* what it needs.
- Our example provide stand-alone package managers since the packages are installed in an independent store. This has helped with adoption.



NixOS

- Declarative system structure
- Atomic upgrades
- Rollbacks
- Reproducible system configurations (container replacers?)
- Lots of symlinks!

GUIX (pronounced GEEKS)

- Based on Nix, so everything mentioned previously
- Guile scheme as base language instead of a DSL
- Emphasis on free (as in freedom) software

“Guix logo” by Luis Felipe López Acevedo is licensed under CC BY-SA 4.0

Environment Specific Package Managers

- pip - Python package manager
- npm - NodeJS package manager
- cargo - Rust package manager

(this isn’t intended to replace system packages)

Why talk about package management?

Question Mark on Chalk Board is in the public domain under CC0

- Many problems solved by complex virtualization solutions are actually package management issues
- Applications may rely on environment package managers which can cause headaches
- Knowing how to package things makes deployment much easier