Universal Packages

Sumner Evans and Robby Zampino

February 1, 2018

Mines Linux Users Group

Introduction

What are packages?

A **package** is an archive containing a collection of executable files or source code, along with metadata, which represent a computer program.

What is a package format?

A **package format** is an organizational structure for delivering packages to users.

Why do we need package formats?

- They provide a common way to bundle executables, libraries, assets, etc. for deployment on user machines.
- They provide metadata about programs for use in package managers.
- It would suck if we had to go find the source code for every single program we want to use and compile from source.[1]

[1] Actually, some package formats do require compilation from source (for example some AUR packages) but at least it helps automate this process.

A bit of history

1994 `dpkg` — the package format behind `apt` and `apt-get`. Used by Debian-based systems.

1997 `RPM` — the package format behind `yum` and `dnf`. Used by RHEL-like systems.

2002 `pacman` — the package manager for Arch Linux. It just uses `tar` files.

2004 `klik`/`PortableLinuxApps` (2011)/`AppImage` (2013) — a package format built to be Linux-distro agnostic.

2006 `nix` — a purely functional package format. Primarily used by NixOS.

June 2016 `snapd` — the Canonical-backed universal package format is ported to a wide range of Linux distros.

June 2016 `Flatpak` — the Red Hat-backed universal package format becomes generally available.

Universal Package Formats

Common objectives

- Linux distro agnosticism
- Solve the “dependency hell”
- Create a “single” deployment target for all of Linux

AppImage

Why is AppImage cool?

- **AppImage does not require installation.** The AppImage file is just its compressed image that is mounted with FUSE when it runs.
- **AppImage does not require root permission.** The application is run as the user and the base system is left untouched.
- **The AppImage itself is executable.** Just `chmod +x` the `.AppImage` file and run.
- **Linus says so**

  "This is just very cool."

  ~ Linus Torvalds

Live Demo:
Running an AppImage

snapd

flatpak

nix

Love to Hate Them

Proprietary enterprise applications are coming to Linux

Currently, when enterprises want to make a cross-platform application, they see this:

macOS	`.dmg`
Windows	`.exe`
Linux	`.deb` and `.rpm` and `PKGBUILD` and …, then deal with the dependency hell[2]

However, when companies like Canonical come in and say "just target snaps", all of a sudden, it may tip the scale at enterprises for them to start targeting Linux. If they create a snap, then they capture all of the Linux market, not just the subset that uses a particular format.

[2] Yes, you have to deal with dependency hell on other platforms too, but every platform has a different type of dependency hell. Coming to Linux is an expensive prospect for many enterprises.

Pros and cons

Pros

- More application availability.

Cons

- The applications are going to be crap. Bloated, Electron, enterprise crap.

Live Demo

Questions?

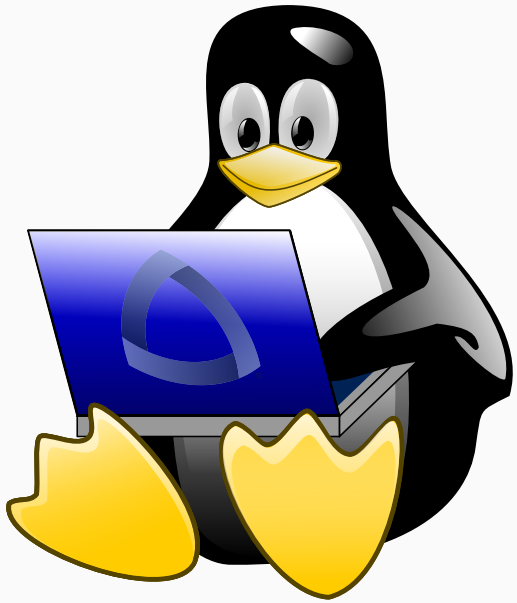
Resources

- https://

Copyright Notice

This presentation was from the **Mines Linux Users Group**. A mostly-complete archive of our presentations can be found online at `https://lug.mines.edu`.

Individual authors may have certain copyright or licensing restrictions on their presentations. Please be certain to contact the original author to obtain permission to reuse or distribute these slides.



Colorado School of Mines
Linux Users Group