Different Distros, Different Approaches

January 02, 2023

By Aaron Grothe

Introduction

If anybody has any questions or comments at any time please let me know.

If I start to mumble please let me know as well :-)

There are a lot of other options to consider. These are just a couple of them. Some of them are just representative of other options.

I'm not saying any one of these is the future of linux, but a lot of the ideas of these will apply to the Linux of 5-10 years from now

Different Distros, Different Approaches?

The classic Linux distribution approach is as follows

- Basic system - confirming to the Linux Filesystem Hierarchy
- Package manager that handles dependencies, installed applications, etc.
- Throw the installed applications into /usr/sbin or /usr/bin, /libs and so on
- Packages are updated over time, replaced or upgraded as needed

Examples of this are Fedora, Ubuntu, Debian, etc…

Different Distros, Different Approaches?

Some drawbacks of this are as follows:

- Package installation requires admin privs
- Systems have drift over time - does your workstation line up with other peoples
- If an upgrade goes wrong, you might be in trouble
  - Actual upgrade for me: Debian upgrade broke synaptic, went to apt, then to dpkg and then to alien to extract files from .deb files to restore system functionality

Options We're going to Talk about

- Fedora Silverblue
  - https://silverblue.fedoraproject.org/
- Gobo Linux
  - https://gobolinux.org/index.html#content
- Nix OS
  - https://nixos.org/
- Gnu Guix OS
  - https://guix.gnu.org/

Fedora Silverblue - Fedora, but its not

At first glance Silverblue looks just like a regular Fedora installation

Several major differences between Silverblue and regular Fedora

- Immutable operating system / Read Only Filesystem
- Atomic updates
- No Yum/DNF - Flatpak
- Toolbox

Fedora Silverblue - Immutable

Cannot modify the Operating system

We'll try and remove ls from the system

```
# sudo bash
# rm /usr/bin/ls
```

Fails read-only operating system

This applies not only to the user but to ANYTHING attempting to modify the operating system.

Fedora Silverblue - Immutable

So how do you change /etc/hosts?

sudo vi /etc/hosts

Wait!? That's possible.

How? Because the OS is split between configuration and O/S files.

Fedora Silverblue - Atomic Updates

So how do you update the system. Use the rpm-ostree commands

rpm-ostree upgrade –check
rpm-ostree upgrade

How does it work: creates a checkpoint, updates the o/s, and then you reboot into the new o/s version

Changes don't take effect until you reboot.

Fedora Silverblue - Atomic Installs

So how do you install an application

rpm-ostree install gimp

This results in versioning the system so you can rollback to before it is installed as well.

Fedora Silverblue - Atomic Updates

Say an update doesn't go well. How do you rollback?

rpm-ostree rollback

Will let you rollback to any version of SB you've installed on your system, provided you haven't purged it.

So the OS is read-only, atomically upgradeable, rollbackable.

Fedora Silverblue - Rebase to Rawhide

You can use the rebase command to rebase to another version of silverblue

E.g. what does it take to upgrade from F37 to Rawhide (testing)

ostree remote refs fedora

rpm-ostree rebase fedora/rawhide/x86_64/silverblue

And you're living on the edge. Didn't boot last time I tried this. So rolled back to a working system.

Fedora Silverblue - No DNF/Yum

The system doesn't have yum

yum; dnf

Flatpak is the main choice to install additional applications

flatpak install org.gimp.GIMP
flatpak run org.gimp.GIMP

Fedora Silverblue - Toolbox

Toolbox provides the ability to create a transparent container you can treat more like a regular distribution

% toolbox create # will create fedora 37 container
% toolbox enter

You're in a container with fedora 37, you can do regular stuff rm /usr/bin/ls, install gimp, etc...

Note: you're doing this all as a regular user to the system and still can't muck with the underlying OS.

Fedora Silverblue - Summary

Read Only Filesystem
Upgrades are atomic - success or failure
Upgrades can be rolled back, forward, purged, etc
No questions during install about Packages
No yum/dnf - rpm-ostree installs possible, flatpak
Toolbox - user containers
Looks like regular Fedora, but it is not

Some other distros such as Suse's tumbleweed and Ubuntu's Core distribution bring the idea of immutable O/Ses as well

GoboLinux - Fight the Hierarchy

GoboLinux doesn't use a database for package management

The filesystem is the package management system

E.g. All installed programs are installed under /Programs

```
% ls /Programs
% ls /Programs/GCC
% ls /Programs/GCC/9.2.0
```

GoboLinux - Fight the Hierarchy

So if you have multiple versions of GCC on a system they'll be under that directory structure

/Programs/GCC/9.2.8
/Programs/GCC/13.0.0, etc…

Dynamically figures out paths to include everything in the relevant directories

Builds everything from Source using repositories

GoboLinux - Bit of Housekeeping

Had real issues with GoboLinux

First time I've really had to read the release notes to try and get a distro up and running in a long time

- Loopback lo0 isn't active by default, so installs doesn't work
- Trying to install packages from had certificate issues and tried to install for the older version of GoboLinux
- Doesn't look to have that much active development going on right now

GoboLinux - Summary

Interesting distribution/concept

Goes back to 2003 - 19 years old
Last release - 2 years ago

Best described as an attempt to bring the FreeBSD ports system to Linux

Interesting concept, but requires a bit of a commitment to make a go of it.

NixOS - Declarative OS

NixOS is a Linux distribution that uses the Nix package manager

The Nix package manager is a package management system that can be installed onto Linux and MacOS systems

NixOS is a system totally controlled by the Nix package manager, including the kernel.

NixOS - Configuring a System

First off every NixOS system has a config file that exists to control the system.

The file is /etc/nixos/configuration.nix

Time to take a bit of a look at it

We'll put wget on the system

NixOS - Rebuilding a System

We've modified /etc/nixos/configuration.nix

Now need to build and install it

nixos-rebuild switch

The above command will reset the system to what the /etc/nixos/configuration.nix file says and switch to it without any additional input from us. See as wget is now on the system

wget

NixOS - Installing Software

Not the only way to do it

Lets go ahead and put python3 on the system

% nix-env -iA nixos.python2

Note: Didn't do that as sudo. Did that as just a regular user.

NixOS - How it does it?

Time to see how the donuts are made

% which ls
% which python2

Note: the python2 is under my account and is not a system file

This means I as a regular user can install whatever packages are available through the nix package manager without needing sudo privs

Drawbacks: old versions, duplicated disk space, etc...

NixOS - Packages

How to see what packages I have installed as a user

% nix-env –query –installed

How to uninstall a package

% nix-env –uninstall python-2.7.18.5

See all available packages

% nix-env -qa

NixOS - Why it is Cool

So what makes NixOS cool

Reproducible system configuration - If I pull the configuration.nix file and take it to another system and do a rebuild. The two machines will have the same system configuration

The entire system - from the kernel up is defined by declarative files in the style of the configuration.nix file

Rollbacks - lets boot the system and take a look at the available boot options for past configurations

NixOS - Shells

You can use shells to create temporary environments, make a few changes and then remove the shell

Want to do a git clone and don't have git installed

% nix-shell -p git

Now we have git, we can use it leave the shell and no harm no fowl

NixOS - Reproducible Builds

This is a very cool option that uses the shells to create an executable in a repeatable process.

```
#! /usr/bin/env nix-shell
#! nix-shell --pure -i python -p "python38.withPackages (ps: [ ps.django ])"
#! nix-shell -I nixpkgs=https://github.com/NixOS/nixpkgs/archive/2a601aafdc5605a5133a2ca506a34a3a73377247.tar.gz

import django

print(django)
```

NixOS - Reproducible Builds

Anyone using the recipe on the previous slide can create the same django environment.

The ability to recreate software in a reproducible matter has been on a lot of distros wish list

NixOS - Summary

The Nix Package manager can be used without using the Nix OS distribution

The Nix Package manager is very cool

NixOS has been described as a technology demonstrator of the Nix package manager

GuixOS - Hurd is Coming :-)

FSF has endorsed several “free” distributions

- Dragora
- Dynebolic
- Guix
- Hyperbola
- Parabola
- PureOS
- Trisquel
- Ututu S

GuixOS - Hurd is Coming :-)

The GNU Guix System is a distribution built by the GNU group

The Guix package manager is based on the Nix Package manager

GuixOS has several major changes to NixOS

Guix is written in Guile - a Scheme based language

GuixOS - Configuring System

Changing configuration of system:

Similar to NixOS. File to modify is /etc/config.scm - one cool thing this file is created during the installation process.

Provides similar options for rollbacks to NixOS based around generations.

To update system

guix pull; guix package -u

GuixOS - Installing a Package (User)

% guix search firefox

No firefox, guess you'll settle for icecat

To install package

% guix install icecat

To remove package

% guix remove icecat

GuixOS - Summary

What makes GuixOS interesting

- Guix Package manager is written in Guile - a variant of Scheme
- It only supports “Free” software
- It uses the Linux-libre kernel (only “Free” kernel modules)
- It uses the GNU shepherd “init” system
- Will support the GNU Hurd kernel as well as the Linux kernel in the future

Is the FSF attempt to create a Linux distribution

Two Other Interesting Distros

Just want to include a couple of mentions to some other distros as well

- Endless OS: An immutable O/S, built on top of Debian with Flatpak. Has a 17gb download which has almost everything you'll need along with the regular downloads. Meant largely as a beginner O/S. Have played with it a bit in a VM. Interesting.
- PopOS!: The System76 guys are working on a new window manager using Rust instead of C/C++. Very much a work in progress, but could be very interesting

Distros - Summary

Each of these distributions brings something new to the party

Silverblue - Atomic updates, Rollbacks, toolbox
GoboLinux - Goodbye LFHS, easy package management
NixOS - Shells, Configuration scripts, reproducible builds
GuixOS - "Free" O/S, shepherd init system, Guix package manager

Each of them is worth firing up in a VM and giving a Shot.

Thanks and Q&A

That's all I've got for tonight.

Thanks for Listening.

Any Questions???