

# Running a non-stock Linux Kernel

July 02, 2024

By Aaron Grothe  
NEbraskaCERT

# Introduction (Continued)

If you have questions/comments please feel free to ask them anytime. You don't have to hold them until the end of the talk.

If there are other resources similar to these that you think might be useful to people please let the group know.

Hopefully this will be an interactive and productive session.

# Introduction

Non-stock Linux Kernel???

First off you might not want to do this

- The default kernels assembled by a distribution goes through LOTS of testing
- Attempt to strike a balance between performance/resource usage/responsiveness, etc
- You can learn a lot doing this as well :-)

# Disclaimer

Here there be Dragons

You may/will have issues with running non-stock kernels

Things like Nvidia drivers, Wifi cards, Virtualbox extensions, etc. may or may not work.

That is why it is always a good idea to keep a stock kernel on the system as well.

Your mileage may vary.



# Non-Stock Kernels

We're going to talk about the following types of Kernels tonight

- Kernels available with Distros (Arch, Ubuntu, etc...)
- Libre Kernel
- Xanmod/Liquorix, etc
- Linux-TKG (easily build your own kernel)
- Doing it the hard way

# Tip: Change the Default Settings

Update the `/etc/default/grub` File with the following settings

```
# vi /etc/default/grub
```

```
GRUB_DEFAULT=saved
```

```
GRUB_SAVEDEFAULT=true
```

```
GRUB_TIMEOUT_STYLE=menu
```

```
GRUB_TIMEOUT=30
```

# Tip: Change the Default Settings

This will save whatever you set the kernel to during boot  
It will show you a menu without having to hold shift

This will improve your quality of life

After modifying the file rerun the update-grub

```
# update-grub
```

# Different Types of Kernels???

We'll be talking about the following types of Kernels

- Pre-emptive Kernels/Real Time Kernels
- Various patches (different schedulers, allocators, etc)
- Libre kernels (Kernels with binary blobs and drivers removed)
- Security Kernels (Hide process ids, system traces, parent process ids, etc)
- Cloud kernels (kernels optimized for the cloud, amazon, azure, etc)
- Vendor kernels (Oracle Unbreakable Linux)
- Roll your own kernel (just what you need, etc)



# Preemptive/Real Time Kernels

Preemptive kernels have patches that are designed to help them respond faster to requests in interactive systems

Examples of uses for this are Audio, Video editing, Games etc.

These changes may increase the overall system load

Some examples of this are the Liquorix Kernel, the Xanmod kernel, and others

# We'll use Liquorix for an Example

The Liquorix kernel <https://www.liquorix.net> is one example of this

To install on Ubuntu do the following

```
% curl -s 'https://liquorix.net/install-liquorix.sh' | sudo  
bash
```

Already have it installed on Kubuntu image - let's go ahead and boot into it

# Another example Xanmod

XanMod Kernel - <https://xanmod.org/> - another general purpose kernel

How to install

1. Register the PGP key:

```
wget -qO - https://dl.xanmod.org/archive.key | sudo gpg  
--dearmor -vo  
/usr/share/keyrings/xanmod-archive-keyring.gpg
```

# Another example Xanmod

XanMod Kernel - <https://xanmod.org/> - another general purpose kernel

How to install - continued

2. Add the repository:

```
echo 'deb  
[signed-by=/usr/share/keyrings/xanmod-archive-keyring.gp  
g] http://deb.xanmod.org releases main' | sudo tee  
/etc/apt/sources.list.d/xanmod-release.list
```



# Another example Xanmod

XanMod Kernel - <https://xanmod.org/> - another general purpose kernel

How to install - continued

3. Then update and install

```
sudo apt update && sudo apt install linux-xanmod-x64v3
```

4. Reboot and Enjoy - Or not (spoilers)

# Xanmod go boom???

Yep. No Joy.

Why???

There are some issues with the modified kernels and Virtualbox and some hardware systems.

Your mileage will vary with these.

# Difference between Preemptive/Real Time

Preemptive kernels attempt to make sure that every process gets processed as quickly as possible

Real Time Kernels are similar except they attempt to make sure that every process gets processed within a certain time frame

Real Time Kernels make take longer to actually give a process CPU time, but works to make sure every process gets cpu time

Real Time kernels are mostly used in things like IOT, embedded systems

# Libre Kernel

The Libre Kernel is a modified Linux kernel that has had the proprietary bits removed

Binary blobs are removed - lot of wifi adapters, graphics drivers, quit working :-)

Brings your Linux Kernel into compliance with the GNU Free Software Standard

Your kernel might not do what you want it to anymore.



# Libre Kernel

## Installing Libre Kernel

1. Add the freesh keyring

wget

[https://linux-libre.fsfla.org/pub/linux-libre/freesh/pool/main/f/freesh-archive-keyring/freesh-archive-keyring\\_1.1\\_all.deb](https://linux-libre.fsfla.org/pub/linux-libre/freesh/pool/main/f/freesh-archive-keyring/freesh-archive-keyring_1.1_all.deb)

2. Install the keyring

```
sudo dpkg -i freesh-archive-keyring_1.1_all.deb
```

# Libre Kernel

## Installing Libre Kernel

### 3. Update the system

```
% sudo apt update
```

### 4. Install the kernel

```
% sudo apt install linux-libre or linux-libre-lts
```

### 5. Reboot and enjoy

# Other Kernel Options (Arch)

Let's take a look at some of the Kernel Options for Arch Linux

<https://wiki.archlinux.org/title/kernel>

Officially Supported Kernels from distro

Optional kernels from distro

Kernel.org Kernels

Unofficial Kernels

Whole lot of options

# Other Kernel Options (Ubuntu)

Let's take a look at some of the Kernel Options for Ubuntu Linux

% synaptic

Search for linux-headers-

Examples: aws, ibm, google, azure, realtime, etc

These are optimized for various cloud providers



# Other Kernel Options (Ubuntu)

Let's take a look at some of the Kernel Options for Ubuntu Linux

% synaptic

Low-latency - kernel with pre-emptive patches, designed for a/v, gaming etc

HWE - HardWare Enhanced - Newer kernel with some additional drivers

# Other Kernel Options

Oracle ships the same kernels as RHEL/CentOS - also has an additional kernel called Unbreakable Linux Kernel - more up to date kernel with some of their patches

Kernels exist out there to give additional capabilities: btrfs, xen, zfs, and a lot more...

Grsecurity has a kernel patch service of security patches to improve kernel security

# Linux-TKG

Linux-TKG - <https://github.com/Frogging-Family/linux-tkg> - is a Linux build system

1. Start with cloning the linux-gtk environment

```
git clone https://github.com/Frogging-Family/linux-tkg.git
```

2. Change to the linux-gtk directory

```
% cd linux-tkg
```

# Linux-TKG

3. Start building the Kernel

```
% ./install.sh install
```

4. Step through a lot of options



# Linux-TKG - Waiting

Linux-TKG takes a while to build your kernel. On my HP Z620 it takes about 10-15 minutes on average, on my laptop I left it running for 2 days and it didn't finish

Eventually - you'll get a kernel in the RPMS folder or DEBS folder

You can then install it with your default package manager and give it a shot.

# Gentoo Genkernel

Gentoo's Genkernel is a very nice system for automatically building and installing a kernel onto a Gentoo system

I used this a lot back on my Ultrasparc 5 back in the day when doing some Kernel development

# The Hard Way

This is the classic way of doing it

1. Grab the kernel from [kernel.org](http://kernel.org)
2. Untar it
3. Configure it with menuconfig
4. Make bzimage; make modules
5. Install

# The Hard Way

Why?

- Total control over the kernel
- Easier to patch/modify

Why not?

- Can be fussy
- You'll get a lot of broken kernels over time



# Benefits of a non-stock Kernel

If you're doing kernel development work it is really nice to be able to build your patch into a kernel for testing

You can run a kernel with additional security patches in it if you want

You might get better performance for things like Gaming, A/V editing and so on

You'll learn a lot

# You might need a non-stock kernel

E.g. A company in town had several Sun thumpers - very nice machine with 48 drive bays. They had quit being supported by Sun.

Decision was given to keep them running until they quit working.

Due to the use of the ZFS kernel, and being a RHEL shop the decision was made to install a custom kernel on RHEL with ZFS support

ZFS isn't included with many distributions due to concerns over the CDDL license that is used with it

# Things I wish I had known Before

- Change the grub settings for the default menu - I spent a lot of time trying to hit the shift key at the right time
- Try some of the kernels that come with your distro example. The ubuntu low-latency will get you most of the benefits of a pre-emptive kernel like xanmod, liquorix, or zen
- Anybody claiming benchmarks is doing it for a specific task, not necessarily what you're doing it for. Your mileage will vary
- Don't worry about screwing up, 99.95% of the time you're a reboot away from getting everything working again
- Try the Linux Libre kernel will give you an idea how free you can be

# Summary & QA

That's all I've got for tonight

Thanks for listening