



r/debian



Search in r/debian



Create



3



r/debian • 1 yr. ago

bionich

## I compiled my first Linux kernel

Back in the 90's I compiled a lot of UNIX kernels, but I've never felt the need to compile a Linux kernel even though it has been my daily OS for about 20 years. Yesterday I decided it was time. There are different ways to do this depending on your distro. I'm a Debian guy so that's where I started. Debian lets you compile the kernel and create kernel .deb packages at the same time. These packages can then be used to update the kernel on other Debian installations.

If anyone would like to give it a try, here are the commands I used to upgrade the kernel from source. I used kernel 6.10.6 from kernel.org. Have fun!

### # Update and Install Dependencies

```
sudo apt update
sudo apt install build-essential linux-headers-$(uname -r) libncurses5-dev bc bison flex
```

### # Setup a build directory

```
mkdir build
cd build
wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.10.6.tar.xz
tar xvf linux-6.10.6.tar.xz
```

### # Configure the kernel . You can make changes or leave it at the default. I left it as default. Save and exit

```
cd linux-6.10.6
cp /boot/config-`uname -r` .config
make menuconfig
```

# Compile the kernel . You can set the jobs flag (j) to whatever you want depending on your processor and RAM, or leave it out entirely to compile one job at a time. Also, you can name your kernel something other than bionich, it's your kernel after all.

```
make -j4 bindeb-pkg LOCALVERSION=-bionich
```

### # Install the new kernel from the .deb packages



r/debian



Create



3



# Reboot your system, login, and check your kernel

```
uname -a
```

Linux debian-kernel-test 6.10.6-bionich #3 SMP PREEMPT\_DYNAMIC Thu Aug 22 12:09:15 PDT 2024  
x86\_64 GNU/Linux

88

31

Share

Join the conversation

Sort by: **Best**

Search Comments



suprjami • 1y ago

Good on you, this is a valuable skill to learn.

Some tips to extend your knowlege:

You can copy a Debian kconfig from a newer package (eg: testing/sid) to `.config` and `make olddefconfig` to apply any new upstream defaults on top of that, this gives you a more Debian-like kernel instead of all upstream defaults.

You can generalise CPU core selection with `make -j $(nproc)`



bionich OP • 1y ago

Thanks. I'm familiar with the nproc util but knew nothing about kconfig. Much appreciated.



suprjami • 1y ago

You might find the book *Linux Kernel in a Nutshell* interesting, it goes through the kconfig and compiling a kernel in more depth.

It's free and available online: <http://www.kroah.com/lkn/>

You will already know a lot of it. Some will be new to you as well.

The way you are compiling the Debian package with `make bindeb-pkg` is preferable for Debian systems. That automates all the module nonsense and is cleaner than manual install/remove.



**c64z86** • 1y ago

Cool! Thanks for the walkthrough! What was compiling and using Unix like back in the 90s? Was it more difficult or really just the same as Linux?

 **bionich** **OP** • 1y ago

It was just different. Installations were definitely much more involved. At that time I installed/managed SCO and BSD UNIX mostly, but I also worked on an AT&T UNIX system in the late 80s. It was my intro to UNIX which solidified my fondness of later OS's like FreeBSD and Linux. To me Linux feels a lot like UNIX, just grown up and heavily evolved.

**c64z86** • 1y ago

Did you have to install programs from source back then? Or were there some early package formats that made it a lot easier?

**HCharlesB** • 1y ago

Back in the day and when I was running SCO UNIX V/386 I installed a lot of code from comp.sources.unix (Netnews) When the Internet became more popular it was often easier to download a tarball (AKA .tgz), unpack it and then run `./configure` and `make` in the top level directory. I believe that is how packages are installed on Slackware today.

**c64z86** • 1y ago • Edited 1y ago

Cool! Did you also use the X11 Window system back then? Did you play any games on SCO UNIX? When did you stop using Unix and switched over to Linux? Did you jump straight into Debian?

**HCharlesB** • 1y ago

Did you also use the X11 Window system back then?

No. The SCO X11 server was \$\$\$\$. There was Xfree86 but building that was beyond my ability.

I stopped using UNIX when I wanted to learn C++ and SCO wanted nearly \$600 for their C++ tool chain. I don't recall why I didn't look for an open source solution. At about that time IBM offered OS/2 with their C++ tool chain for \$289 and I went for that. It turned out that IBM had a decent C++ compiler (better than MSVC at the time) and I



r/debian



Create



3



IIRC my first distro was Slackware but I used Ubuntu, Mandrake, Enlightenment and a few I don't even recall before settling on Debian.



bionich OP • 1y ago

Wow good memory! I've setup Xfree86 on SCO OpenServer a few times but I had completely forgotten that it even existed.

HCharlesB • 1y ago

Wow good memory!

Bits and pieces. I have to take copious notes these days to keep things straight.

TheUruz • 1y ago

this is the kind of posts that deserves to be up the ladder. i'm a linux enthusiast and i'd really like to try this. it's pretty straightforward when it comes to commands but i'd like to delve deeper on what these commands do. any chance there's a youtube video explaining those?



bionich OP • 1y ago

TheUruz,

At the end of the day it's pretty straight forward to understand, *conceptually*.

- Download the tools used for compiling
- Download the kernel source code
- Configure the options (choose what you want in your kernel)
- Compile (make) your kernel; that is transform the source code into binaries
- Install your new kernel

I hope this helps. You can search YouTube and I'm sure you'll find something useful, although it may be out of date.



r/debian



Create



3



Tricky\_Reporter8809 • 1y ago

What are the main benefits to compiling the kernel yourself? I know that this can easily be searched for, but I would like to hear your and others opinion.



bionich OP • 1y ago

I think the most common reason is to get the latest drivers. That's why I'm interested. I have a Framework laptop. It's completely user serviceable. I have an Intel motherboard now, but I may want to switch it out for an AMD Ryzen 7040. I would need an up-to-date kernel for Linux to work on that hardware.

There are other reasons, like trying to reduce the size of your kernel to make your system run a little quicker and use less memory or to get some new features, but I think the main reason is the one I already stated, "drivers."



Tricky\_Reporter8809 • 1y ago

Interesting. On your current setup with the settings/flags you showcased, how long does the process of compiling usually take?



bionich OP • 1y ago

This is entirely hardware dependent. I have a laptop with an Intel 13th GEN i5 and 32GB of RAM. What I found from benchmarking was that the kernel compiled in the about the same amount of time whether I set the j flag to 8 or 16. I believe it's because of thermal throttling. At 16 concurrent jobs the CPU throttled down from 4.5 Ghz to about 2.7 Ghz. It took about 30 minutes in either case. If you have older hardware it could take hours.



CPlushPlus • 1y ago

dang.. and i thought the dell xps 9570 was good for modularity.



bionich OP • 1y ago

I'm definitely a [Framework Laptop](#) fanboy. I even bought one for my wife to replace her old Mac laptop, and she runs Debian now too.



CPlushPlus • 1y ago

[r/wholesome](#) 😊



r/debian



Create



3



Most people stuck on Windows are either stuck there for games, or for audio visual software..

(The pro audio situation has vastly improved. Thanks to pipewire and yabridge for vsts, not to mention Bitwig, providing a native experience similar to Ableton, (for a price))

HidekiAI • 1y ago

I too am curious, though only reasons I can think of was when I needed to test legacy (very old) obscure hardware (i.e. hardware back in Win95 days) in which the driver was not supported on latest kernel, so I had to rollback on the kernel that matched the driver for that peripheral... so mainly, did it for the sake of KO file(s) to insmod but nothing more. (Coming from Gentoo for more than a decade prior to switching to Debian, I've got tired of compiling kernels and diagnosing why grub won't load newly built kernel, etc).



GregorDeLaMuerte • 1y ago

nice, did you do it just for fun or did you need to customise something?



bionich OP • 1y ago • Edited 1y ago

Just for fun, really. I may need an up-to-date kernel down the road if I decide to upgrade the mobo in my Framework laptop from Intel to AMD. Otherwise I have no need, but that's what inspired me to figure it out.



GregorDeLaMuerte • 1y ago

Cool!

falunosama • 1y ago

I would love to do this but for my raspberry pi, but i'm not sure how to cross compile :(

birds\_swim • 1y ago

Hey, is this the thread you were talking about that might fix my weird AMD microcode issues?



r/debian



Create



3



If you have newer hardware then a newer kernel may be the ticket to fixing weird issues, because the updated drivers may better support that new hardware.

**birds\_swim** • 1y ago • Edited 1y ago

What's the difference between this solution and the microcode package available in the Debian repos?

Secondly, do I need to remove that package before doing this?

Relevant searches:

- The package in question <https://packages.debian.org/search?keywords=amd64-microcode>
- Debian Wiki <https://wiki.debian.org/Microcode>



**bionich** OP • 1y ago

I think\*\*\*, but I could be completely wrong,\*\*\* that the microcode only addresses issues with the CPU and not other things like your AMD GPU for example. Whereas the new kernel has all the latest drivers for your new hardware.

Perhaps someone with more experience can chime in here?

**Diligent-Thing-1944** • 1y ago

Thanks for the valuable inputs. This is for the keeps.



**TuxedoUser** • 2mo ago

Worked perfectly for me! Thanks!