How to Build Linux Kernel From Scratch {Step-By-Step Guide}

November 12, 2020

COMMANDS KERNEL LINUX

Contents

Introduction

The Linux Kernel is the foundation of the Unix-like operating systems. The kernel is responsible for communication between hardware and software and the allocation of available resources.

All Linux distributions are based on a predefined kernel. But, if you want to disable several options and drivers or try experimental patches, you need to build a Linux kernel.

In this step-by-step guide, you will learn how to build and compile a Linux kernel from scratch.

Prerequisites

- A system running Linux
- Access to the terminal/command line
- A user account with **sudo/root** privileges
- 12GB of available space on the hard drive

Building Linux Kernel

The process of building a Linux kernel takes seven easy steps to complete. However, the procedure requires a significant amount of time to complete, depending on the system speed.

Follow the steps below to build the latest Linux kernel at the time of writing this article.

Note: If the version on the kernel website does not match the one from the steps below, use these commands and replace the kernel version number.

Step 1: Download the Source Code

1. Visit the official kernel website and download the latest kernel version. The downloaded file contains a compressed source code.

2. Open the terminal and use the wget command to download the Linux kernel source code:

```
wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.9.6.tar.xz
```

The output shows the “saved” message when the download completes.

```
n@n-VirtualBox:~$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.9.6.tar.xz
--2020-11-06 16:10:11--  https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.9.6.tar.x
z
Resolving cdn.kernel.org (cdn.kernel.org)... 199.232.17.176, 2a04:4e42:41::432
Connecting to cdn.kernel.org (cdn.kernel.org)|199.232.17.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 115547768 (110M) [application/x-xz]
Saving to: ‘linux-5.9.6.tar.xz’

linux-5.9.6.tar.xz  100%[==================>] 110,19M  34,9MB/s    in 3,2s

2020-11-06 16:10:15 (34,9 MB/s) - ‘linux-5.9.6.tar.xz’ saved [115547768/115547768]

n@n-VirtualBox:~$
```

Step 2: Extract the Source Code

When the file is ready, run the tar command to extract the source code:

```
tar xvf linux-5.9.6.tar.xz
```

```
n@n-VirtualBox:~$ tar xvf linux-5.9.6.tar.xz
linux-5.9.5/virt/kvm/
linux-5.9.5/virt/kvm/Kconfig
linux-5.9.5/virt/kvm/async_pf.c
linux-5.9.5/virt/kvm/async_pf.h
linux-5.9.5/virt/kvm/coalesced_mmio.c
linux-5.9.5/virt/kvm/coalesced_mmio.h
linux-5.9.5/virt/kvm/eventfd.c
linux-5.9.5/virt/kvm/irqchip.c
linux-5.9.5/virt/kvm/kvm_main.c
linux-5.9.5/virt/kvm/vfio.c
linux-5.9.5/virt/kvm/vfio.h
linux-5.9.5/virt/lib/
linux-5.9.5/virt/lib/Kconfig
linux-5.9.5/virt/lib/Makefile
linux-5.9.5/virt/lib/irqbypass.c
n@n-VirtualBox:~$
```

Step 3: Install Required Packages

Install additional packages before building a kernel. To do so, run this command:

```
sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc flex libelf-dev bison
```

The command we used above installs the following packages:

Package	Package description
`git`	Tracks and makes a record of all changes during development in the source code. It also allows reverting the changes.
`fakeroot`	Packaging tool that makes the fake root environment.
`build-essential`	Installs development tools such as C, C++, gcc, and g++.
`ncurses-dev`	Programming library that provides API for the text-based terminals.
`xz-utils`	Provides fast file compression and decompression.
`libssl-dev`	Supports SSL and TSL that encrypt data and make the internet connection secure.
`bc` (Basic Calculator)	A mathematical scripting language that supports the interactive execution of statements.

`libelf-dev`	Issues a shared library for managing ELF files (executable files, core dumps and object code)
`bison`	GNU parser generator that converts grammar description to a C program.

```
n@n-VirtualBox:~$ sudo apt-get install git fakeroot build-essential ncurses-dev xz-util
s libssl-dev bc flex libelf-dev bison
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libncurses-dev' instead of 'ncurses-dev'
bc is already the newest version (1.07.1-2build1).
bc set to manually installed.
11 upgraded, 45 newly installed, 0 to remove and 227 not upgraded.
Need to get 47,7 MB/56,9 MB of archives.
After this operation, 196 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Setting up build-essential (12.8ubuntu1.1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
n@n-VirtualBox:~$
```

Step 4: Configure Kernel

The Linux kernel source code comes with the default configuration. However, you can adjust it to your needs. To do so, follow the steps below:

1. Navigate to the linux-5.9.6. directory using the **cd** command:

```
cd linux-5.9.6
```

2. Copy the existing configuration file using the **cp** command:

```
cp -v /boot/config-$(uname -r) .config
```

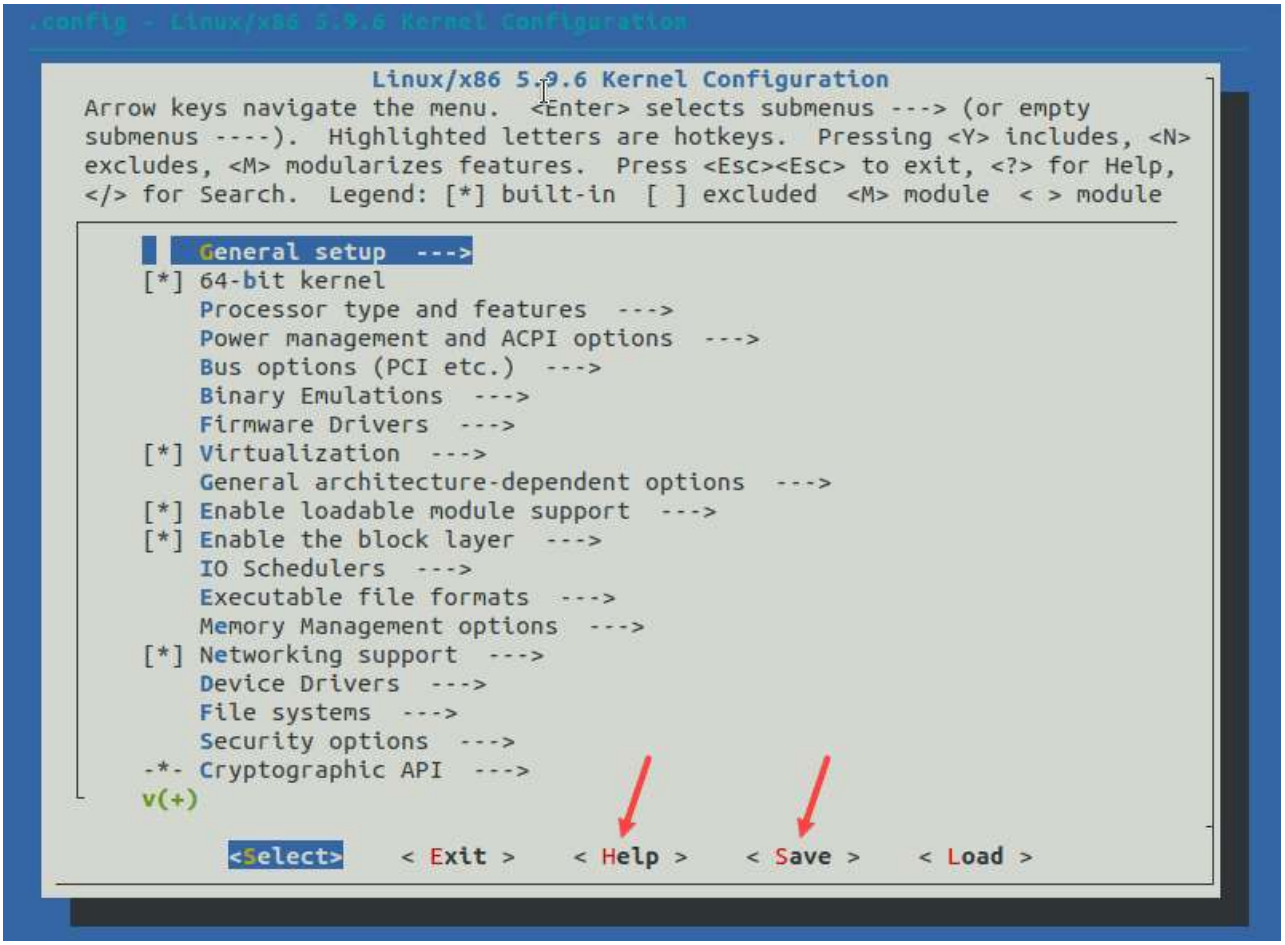
```
n@n-VirtualBox:~$ cd linux-5.9.6
n@n-VirtualBox:~/linux-5.9.6$ cp -v /boot/config-$(uname -r) .config
'/boot/config-5.9.6' -> '.config'
n@n-VirtualBox:~/linux-5.9.6$
```

3. To make changes to the configuration file, run the **make** command:

```
make menuconfig
```

```
n@n-VirtualBox:~/linux-5.9.6$ make menuconfig
  HOSTCC  scripts/basic/fixdep
  UPD     scripts/kconfig/mconf-cfg
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/confdata.o
  HOSTCC  scripts/kconfig/expr.o
  LEX     scripts/kconfig/lexer.lex.c
  YACC    scripts/kconfig/parser.tab.[ch]
  HOSTCC  scripts/kconfig/lexer.lex.o
  HOSTCC  scripts/kconfig/parser.tab.o
  HOSTCC  scripts/kconfig/preprocess.o
  HOSTCC  scripts/kconfig/symbol.o
  HOSTCC  scripts/kconfig/util.o
  HOSTLD  scripts/kconfig/mconf
```

4. The configuration menu includes options such as firmware, file system, network, and memory settings. Use the arrows to make a selection or choose **HELP** to learn more about the options. When you finish making the changes, select **SAVE,** and then exit the menu.



Note: Changing settings for some options can lead to a non-functional kernel. If you are unsure what to change, leave the default settings.

1. Start building the kernel by running the following command:

```
make
```

The process of building and compiling the Linux kernel takes some time to complete.

The terminal lists all Linux kernel components: memory management, hardware device drivers, filesystem drivers, network drivers, and process management.

```
n@n-VirtualBox:~/linux-5.9.5$ make
  SYSTBL  arch/x86/include/generated/asm/syscalls_32.h
  SYSHDR  arch/x86/include/generated/asm/unistd_32_ia32.h
  SYSHDR  arch/x86/include/generated/asm/unistd_64_x32.h
  SYSTBL  arch/x86/include/generated/asm/syscalls_64.h
  HYPERCALLS arch/x86/include/generated/asm/xen-hypercalls.h
  SYSHDR  arch/x86/include/generated/uapi/asm/unistd_32.h
  SYSHDR  arch/x86/include/generated/uapi/asm/unistd_64.h
  SYSHDR  arch/x86/include/generated/uapi/asm/unistd_x32.h
  HOSTCC  arch/x86/tools/relocs_32.o
  HOSTCC  scripts/selinux/mdp/mdp
  LD [M]  sound/usb/usx2y/snd-usb-usx2y.ko
  CC [M]  sound/x86/snd-hdmi-lpe-audio.mod.o
  LD [M]  sound/x86/snd-hdmi-lpe-audio.ko
  CC [M]  sound/xen/snd_xen_front.mod.o
  LD [M]  sound/xen/snd_xen_front.ko
  GEN     scripts/gdb/linux/constants.py
n@n-VirtualBox:~/linux-5.9.6$
```

2. Install the required modules with this command:

```
sudo make modules_install
```

```
n@n-VirtualBox:~/linux-5.9.6$ sudo make modules_install
  INSTALL sound/usb/line6/snd-usb-line6.ko
  INSTALL sound/usb/line6/snd-usb-pod.ko
  INSTALL sound/usb/line6/snd-usb-podhd.ko
  INSTALL sound/usb/line6/snd-usb-toneport.ko
  INSTALL sound/usb/line6/snd-usb-variax.ko
  INSTALL sound/usb/misc/snd-ua101.ko
  INSTALL sound/usb/snd-usb-audio.ko
  INSTALL sound/usb/snd-usbmidi-lib.ko
  INSTALL sound/usb/usx2y/snd-usb-us122l.ko
  INSTALL sound/usb/usx2y/snd-usb-usx2y.ko
  INSTALL sound/x86/snd-hdmi-lpe-audio.ko
  INSTALL sound/xen/snd_xen_front.ko
  DEPMOD  5.9.6
n@n-VirtualBox:~/linux-5.9.6$
```

3. Finally, install the kernel by typing:

```
sudo make install
```

```
n@n-VirtualBox:~/linux-5.9.6$ sudo make install
sh ./arch/x86/boot/install.sh 5.9.6 arch/x86/boot/bzImage \
        System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.9.6 /boot/vmlinuz-5.9.6
run-parts: executing /etc/kernel/postinst.d/dkms 5.9.6 /boot/vmlinuz-5.9.6
 * dkms: running auto installation service for kernel 5.9.6                  [ OK ]
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.9.6 /boot/vmlinuz-5.9.6
update-initramfs: Generating /boot/initrd.img-5.9.6
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.9.6 /boot/vmlinuz-5.9
.6
run-parts: executing /etc/kernel/postinst.d/update-notifier 5.9.6 /boot/vmlinuz-5.9.6
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.9.6 /boot/vmlinuz-5.9.6
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.9.6
Found initrd image: /boot/initrd.img-5.9.6
Found linux image: /boot/vmlinuz-5.4.0-52-generic
Found initrd image: /boot/initrd.img-5.4.0-52-generic
Found linux image: /boot/vmlinuz-5.4.0-42-generic
Found initrd image: /boot/initrd.img-5.4.0-42-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
n@n-VirtualBox:~/linux-5.9.6$
```

Step 6: Update the Bootloader (Optional)

The GRUB bootloader is the first program that runs when the system powers on.

The **`make install`** command performs this process automatically, but you can also do it manually.

1. Update the initramfs to the installed kernel version:

```
sudo update-initramfs -c -k 5.9.6
```

2. Update the GRUB bootloader with this command:

```
sudo update-grub
```

The terminal prints out the process and confirmation message:

```
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.9.6
Found initrd image: /boot/initrd.img-5.9.6
Found linux image: /boot/vmlinuz-5.4.0-52-generic
Found initrd image: /boot/initrd.img-5.4.0-52-generic
Found linux image: /boot/vmlinuz-5.4.0-42-generic
Found initrd image: /boot/initrd.img-5.4.0-42-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
```

Step 7: Reboot and Verify Kernel Version

When you complete the steps above, reboot the machine.

When the system boots up, verify the kernel version using the **uname** command:

```
uname -mrs
```

The terminal prints out the current Linux kernel version.

```
n@n-VirtualBox:~$ uname -mrs
Linux 5.9.6 x86_64
n@n-VirtualBox:~$
```

Conclusion

In this step-by-step guide, you learned how to build a Linux kernel from scratch and install the required packages.

If you follow the instructions carefully, the process will complete successfully on your Linux machine.

The Linux kernel has a modular design. Functionality is extendible with modules or drivers. Learn how to use the modprobe command to add or remove modules on Linux.

Goran Jevtic

Goran combines his leadership skills and passion for research, writing, and technology as a Technical Writing Team Lead at phoenixNAP. Working with multiple departments and on various projects, he has developed an extraordinary understanding of cloud and virtualization technology trends and best practices.

Next you should read

SysAdmin, Web Servers

How to Remove Old Kernels on Ubuntu 16.04, 18.04, & 19.04

March 11, 2020

There are several methods to remove old or unused kernels. It's also considered good system hygiene practice...

READ MORE

SysAdmin, Web Servers

How to Upgrade Linux Kernel in CentOS 7

October 8, 2019

architecture of each Linux distribution. It provides and defines...

READ MORE

DevOps and Development, SysAdmin

How to Uninstall or Remove Software Packages From Ubuntu

September 4, 2019

This guide will walk you through several methods for removing old software from Ubuntu...

READ MORE

Security, SysAdmin

How to Check Kernel Version in Linux in Command Line

June 25, 2019

The Linux kernel is much like the central brain of the operating system. Although it is open-source, meaning...