Compile and boot a Linux Kernel

Anthony McGlone

October 19, 2022

Table of Contents

Chapter 1 Introduction

This guide will demonstrate the basic process of compiling and booting a Linux kernel. More advanced kernel configuration (such as enabling a device driver or new kernel features) will not be covered.

If you're already using a Linux operating system (such as Ubuntu, Red Hat or CentOS), you can skip ahead to the kernel installation instructions for your OS.

If you're on Windows or Mac, you can use Oracle's VirtualBox to test an installation.

1.1 Installing Oracle VirtualBox

1. Read the installation instructions for your OS.
2. Download the VirtualBox binary / package for your OS.
3. Download an ISO of a Linux OS (e.g. Ubuntu, CentOS, Fedora).
4. Create a virtual machine using the ISO / installation media.

Chapter 2 Ubuntu

2.1 Download the latest Linux kernel

In your Linux OS or virtual machine, open the kernel home page by navigating to this website. Download the latest source code by clicking on the large yellow button.

The `tar.xz` file should be downloaded to the `Downloads` folder.

2.2 Extract the source code

Open a terminal and navigate to your `Downloads` folder. Extract the `tar` file using the `unxz` command:

```
unxz -v linux-6.0.2.tar.xz
```

Start to verify the PGP signature of the `tar` file that you extracted. On kernel home page, copy the link address from the `[pgp]` link for the latest kernel version. Then add this link to the `wget` command and run it:

```
wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.0.2.tar.sign
```

Retrieve the RSA key needed to complete verification of the PGP signature. Run the following command:

```
gpg --verify linux-6.0.2.tar.sign
```

The output from the command contains the RSA key, which will be used in the verification:

```
gpg:  assuming signed data in 'linux-6.0.2.tar'
gpg:  Signature made Sat 15 Oct 2022 07:04:02 IST
gpg:  using RSA key 647F28654894E3BD457199BE38DBBDC86092693E
gpg:  Can't check signature:  No public key
```