Linux Kernel

Compile Linux Kernel on CentOS7

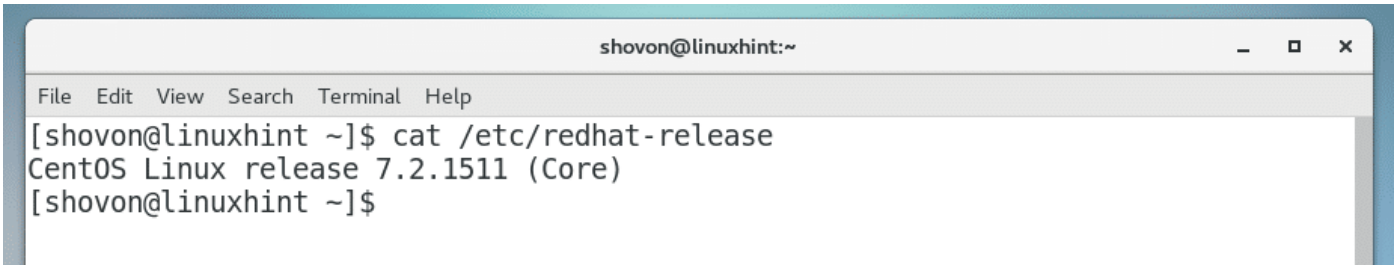
4 years agoby Shahriar Shovon

Compile the Latest Linux Kernel from Source on CentOS 7

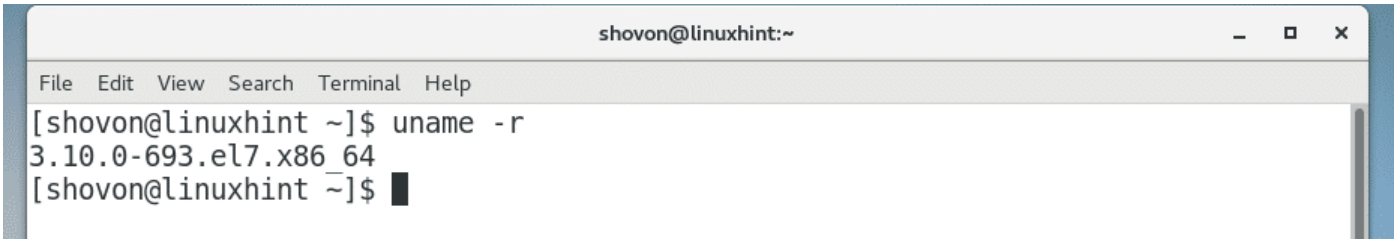
In this article I will show you how to download the latest Linux kernel source from the official website of Linux kernel, compile Linux kernel from source and use the compiled kernel on CentOS 7. Let's get started.

Checking Currently Used Kernel:

You can see from the following screenshot that I am using CentOS 7.



And the current kernel version is 3.10

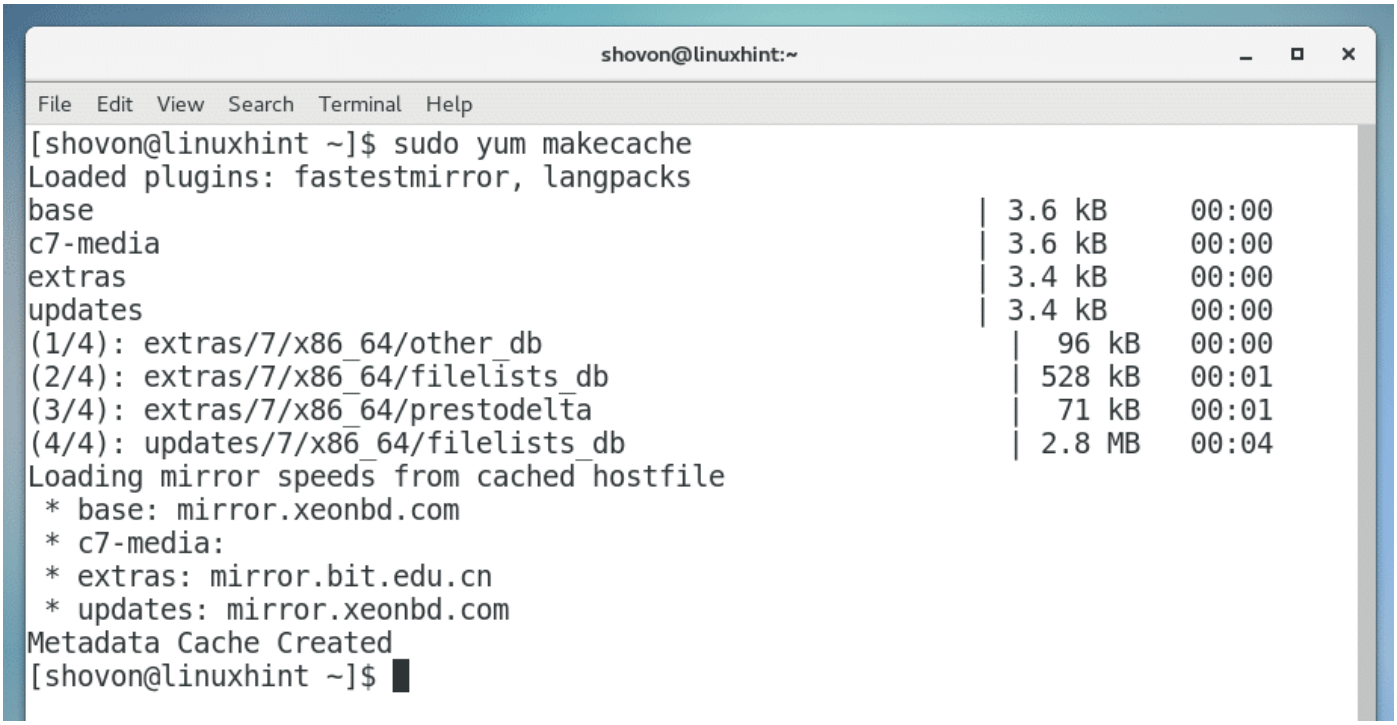


Installing the prerequisites:

To compile the latest Linux kernel from source on CentOS 7, you must have a build tool and some other packages installed on your CentOS 7 operating system.

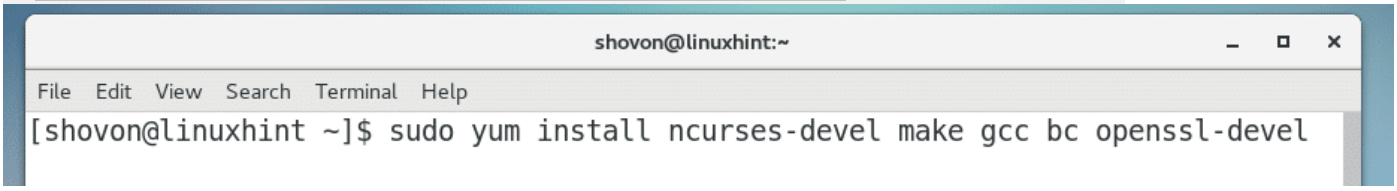
Before you install anything, run the following command to update package cache:

```
$ sudo yum makecache
```

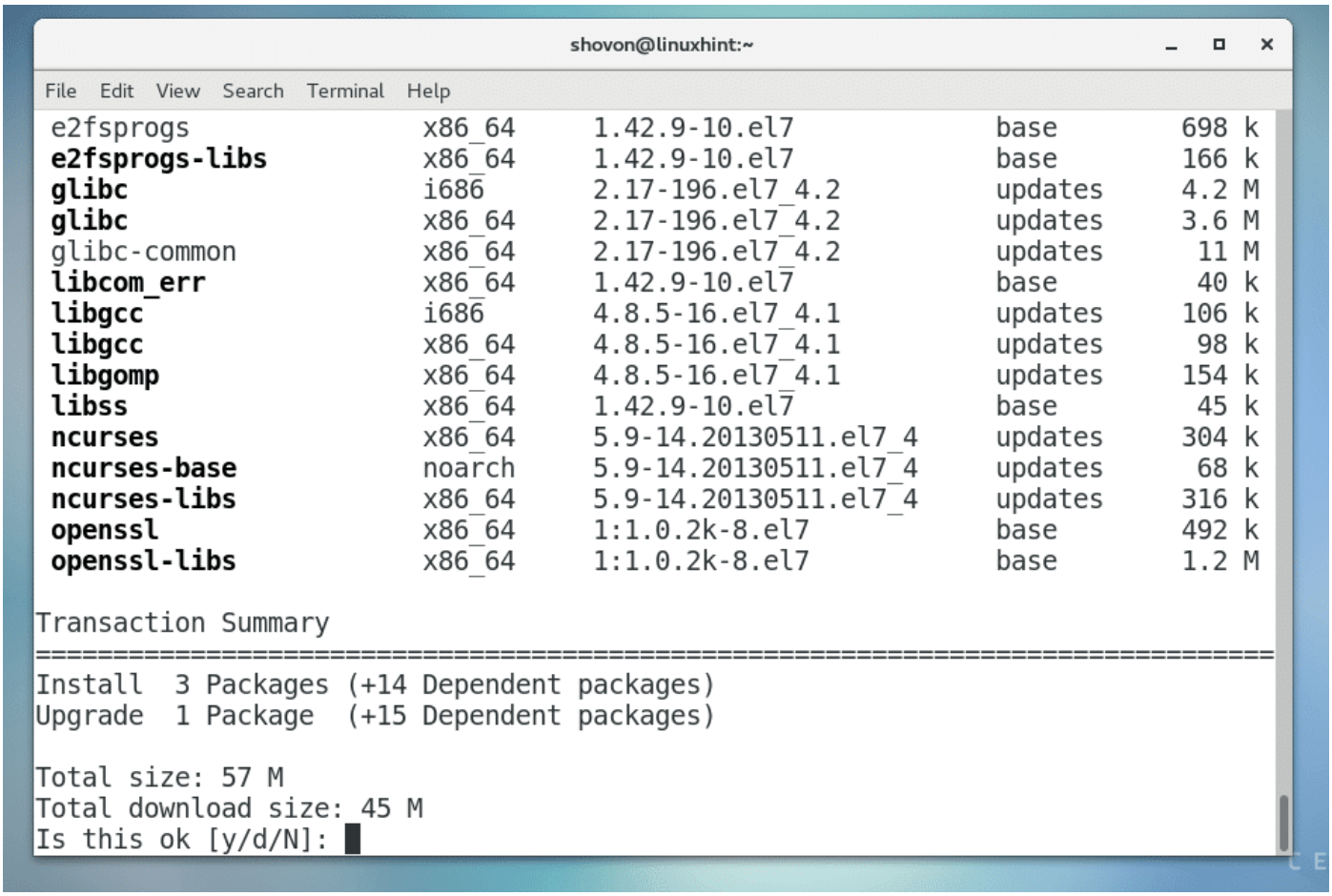


Now you can install the compilers and libraries required for compiling the kernel with the following command:

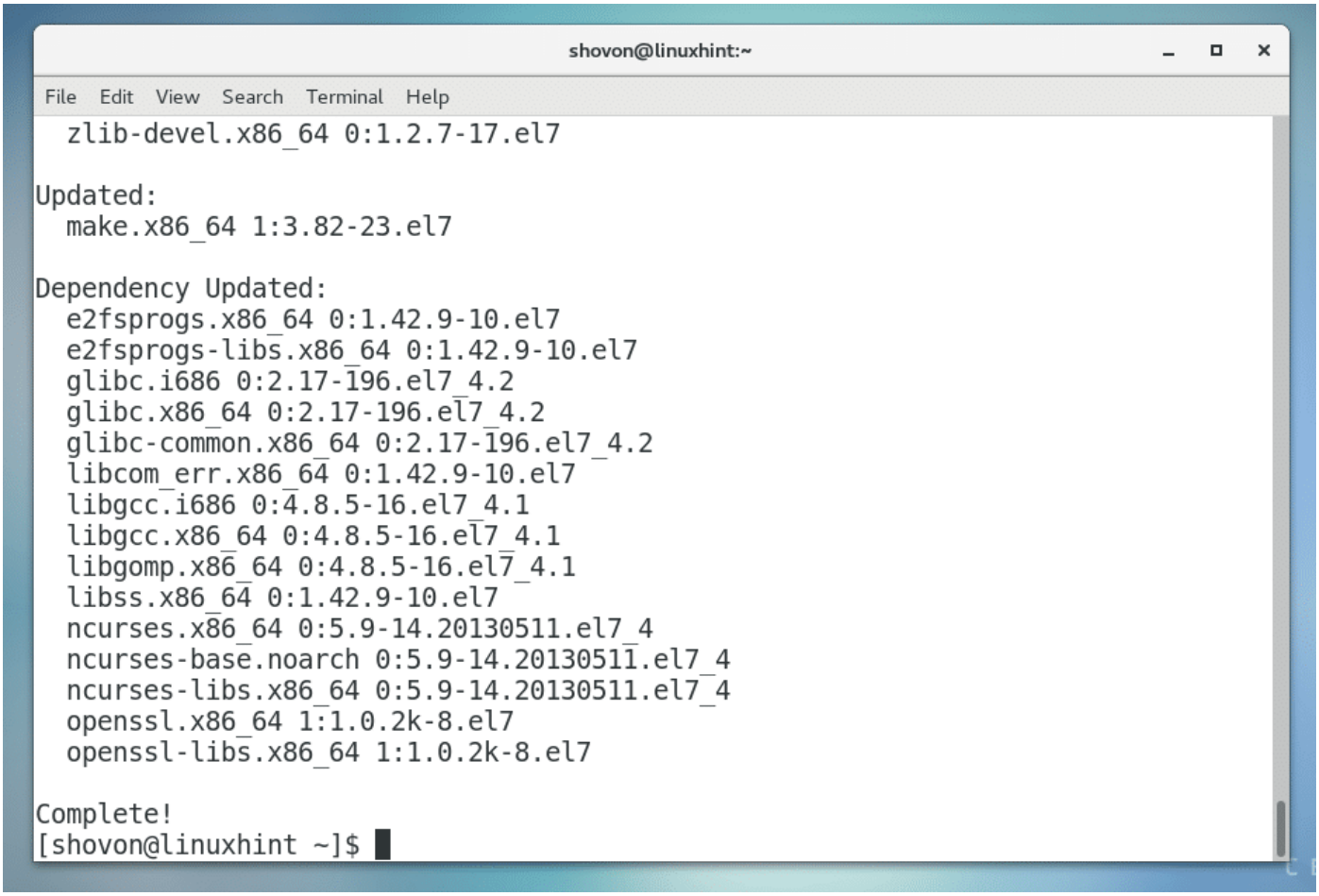
```
$ sudo yum install ncurses-devel make gcc bc
```



Press ‘y’ and then press <Enter> to continue.

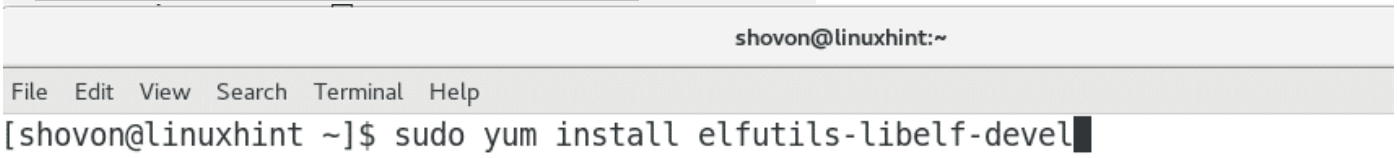


The build tools should be installed.

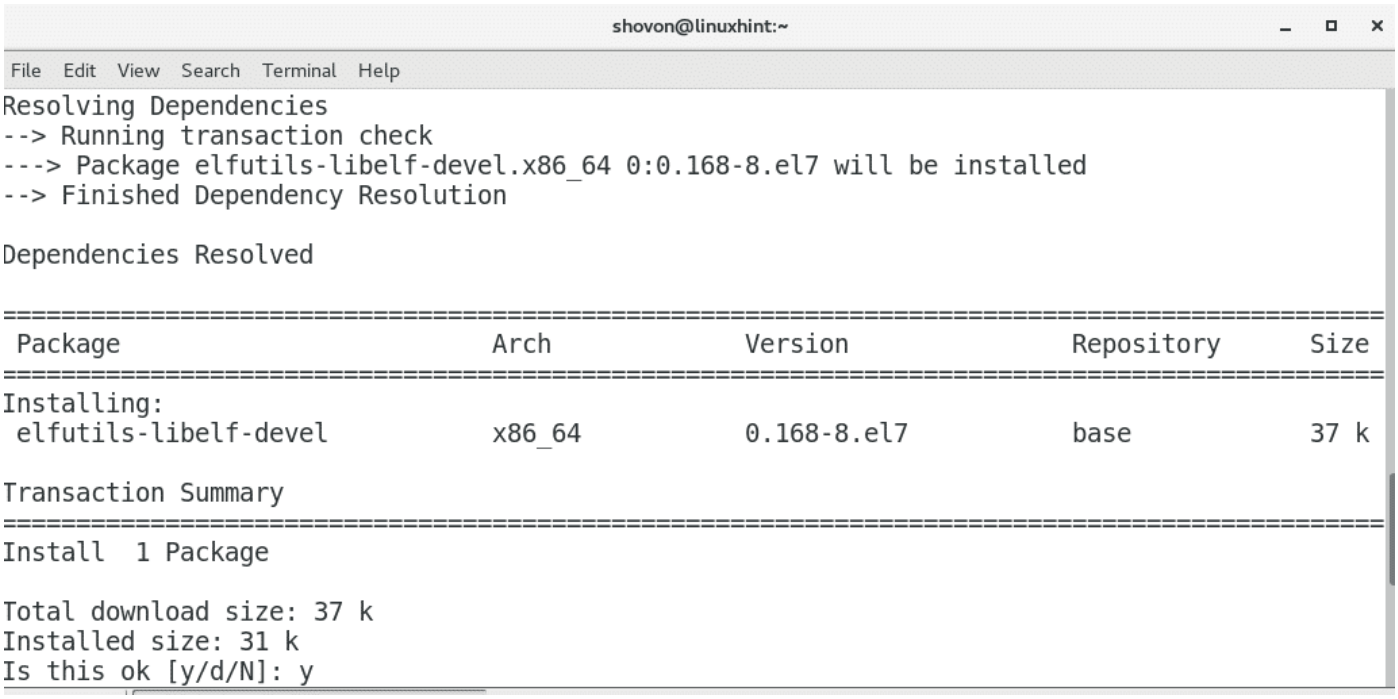


Now you have to install elfutils package. Run the following command to install it:

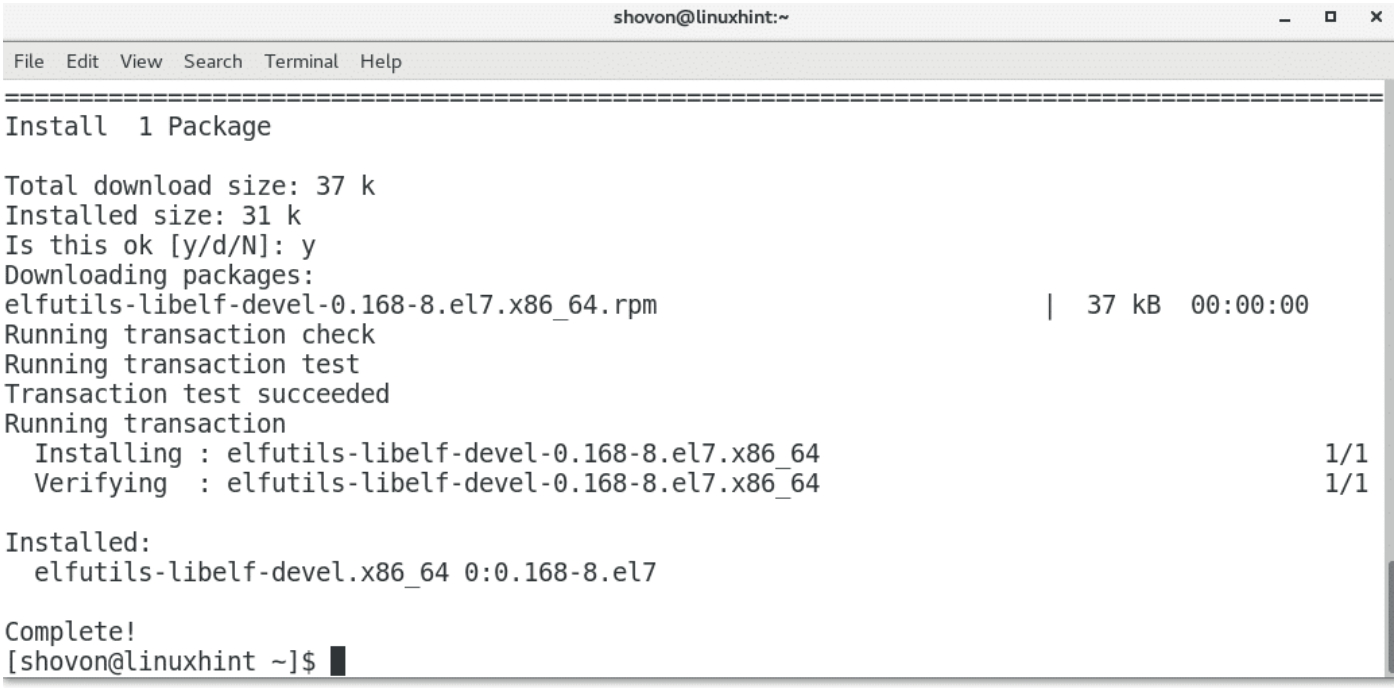
```
$ sudo yum install elfutils-libe
```



Press ‘y’ and press <Enter> to continue.

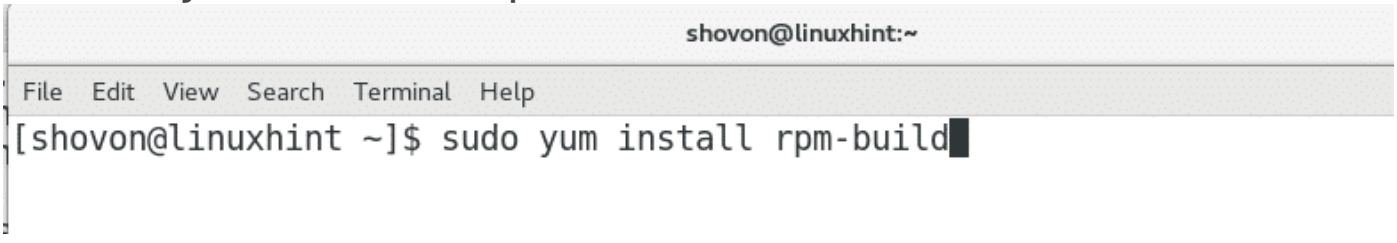


‘elfutils’ should be installed.

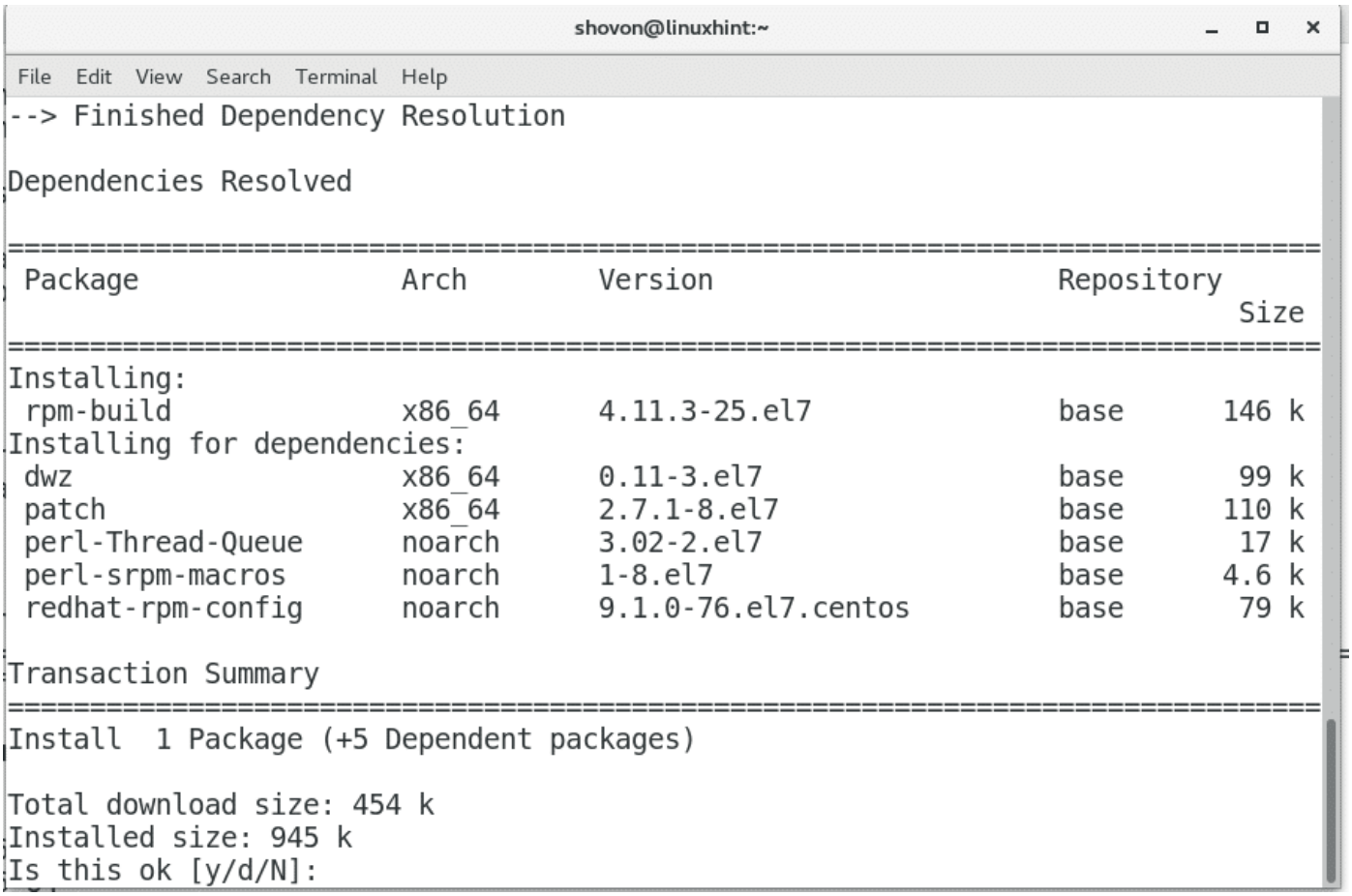


Now you have to install rpm-build with the following command:

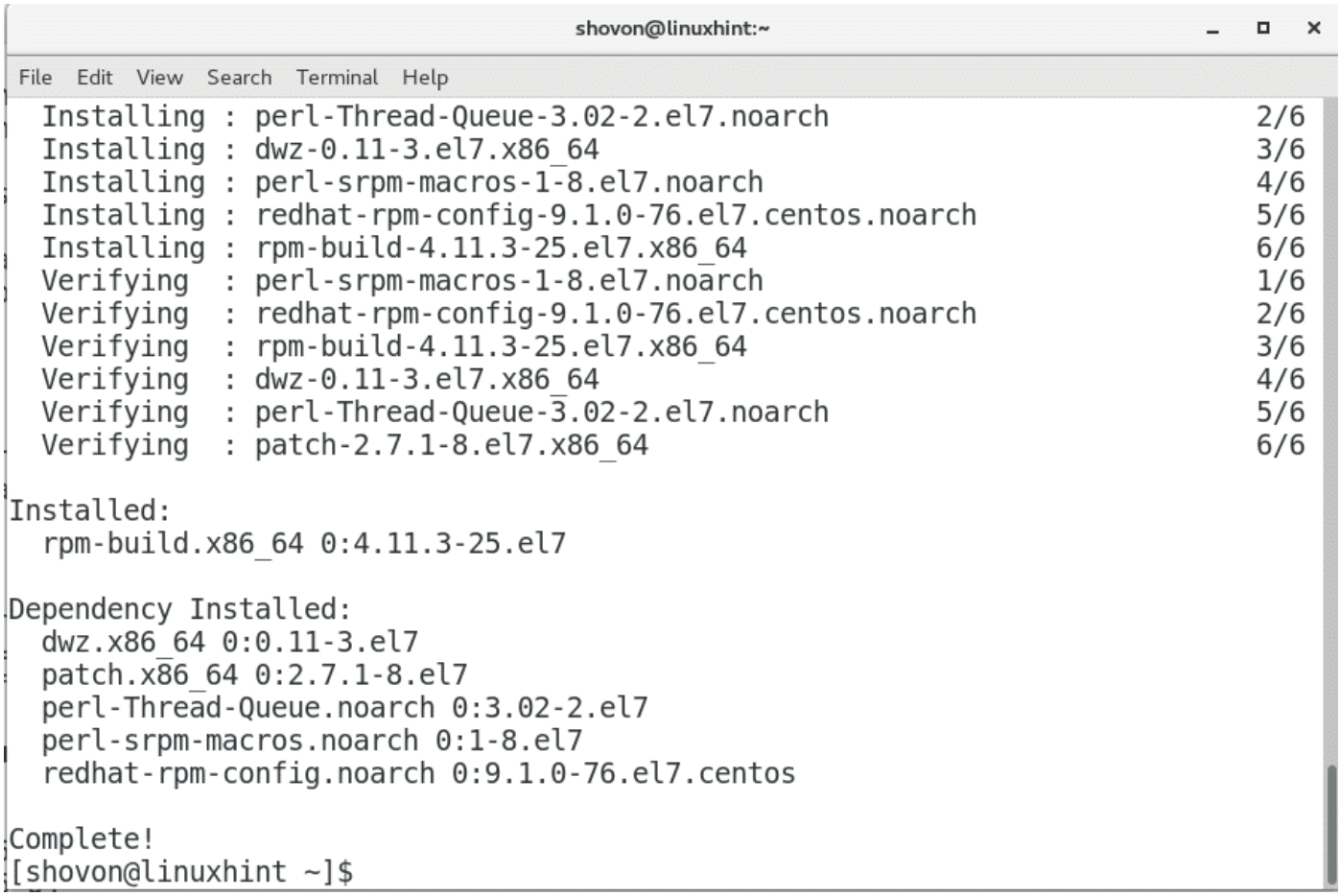
```
$ sudo yum install rpm-build
```



Press ‘y’ and then press <Enter> to continue.



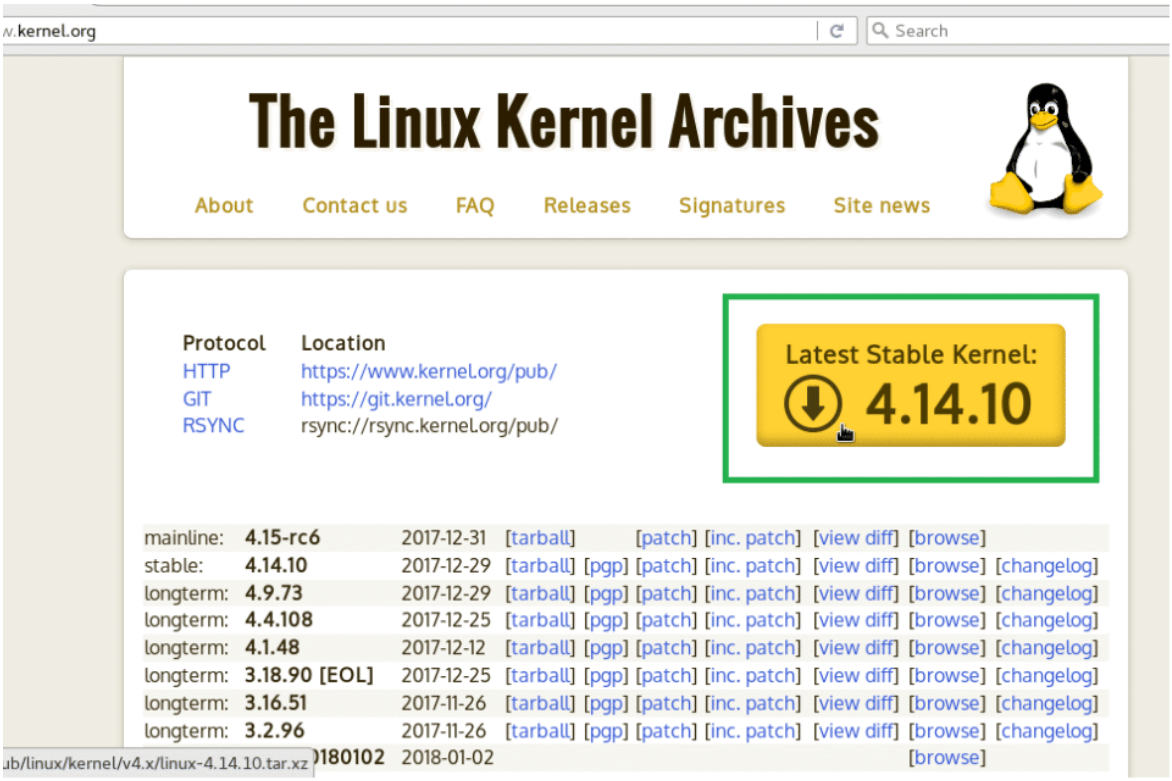
‘rpm-build’ package should be installed.



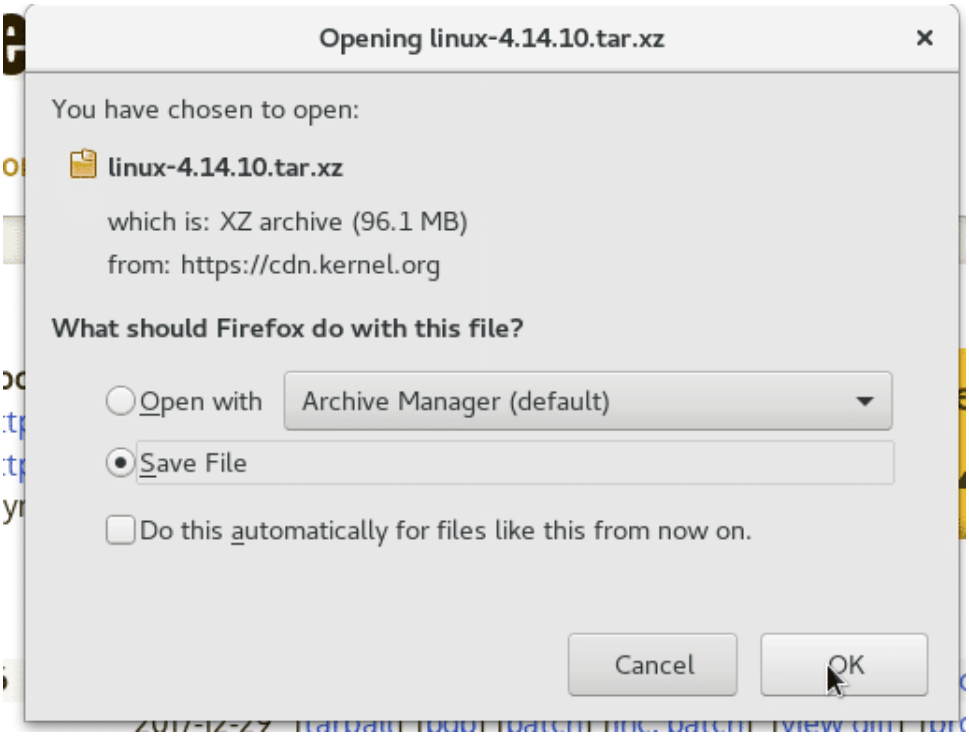
Downloading the Linux Kernel Source:

Go to the official website of Linux Kernel at https://www.kernel.org and you should be the following page.

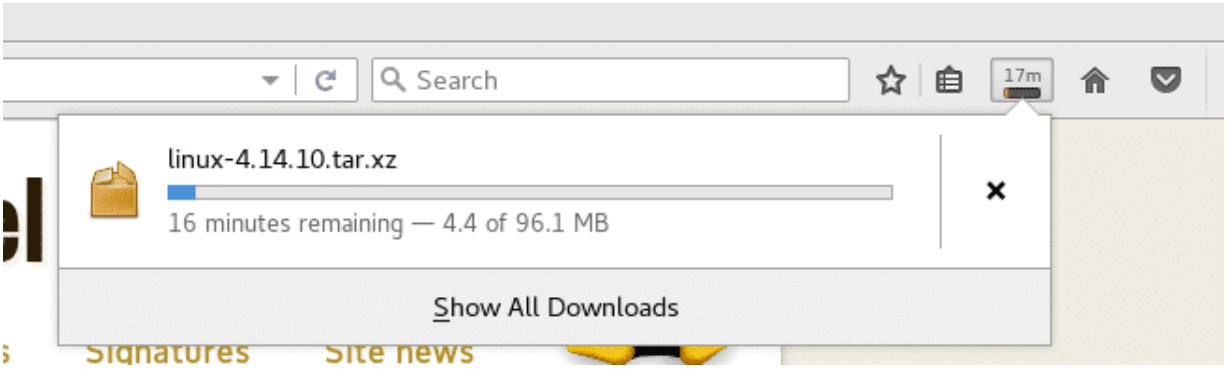
Click on the “Latest Stable Kernel” button as marked on the screenshot below.



Your browser should prompt you to save the file. Just click on “Save File” and click on “OK”.



Your download should begin.

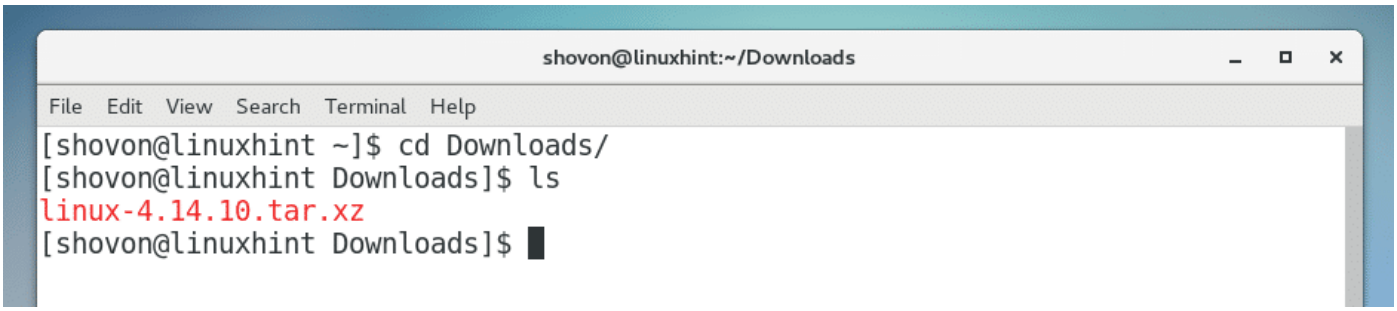


Compiling the Kernel:

Once the download is complete, navigate to the directory where you downloaded the file. In my case it is the Downloads directory in my user's HOME directory.

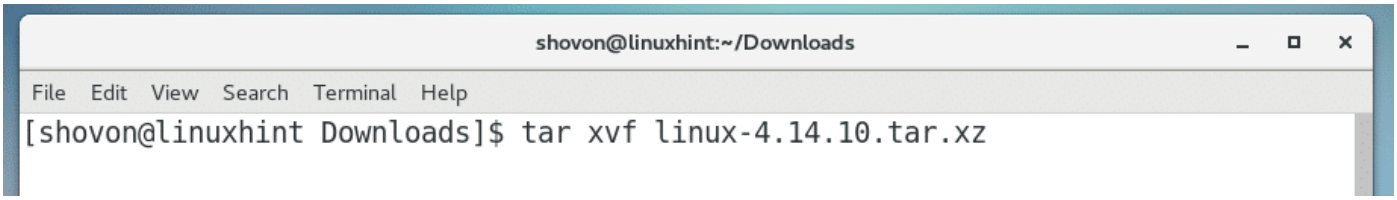
```
$ cd ~/Downloads
```

From the output of 'ls' you can see that the downloaded file is 'linux-4.14.10.tar.xz'. Which is a compressed tar file.

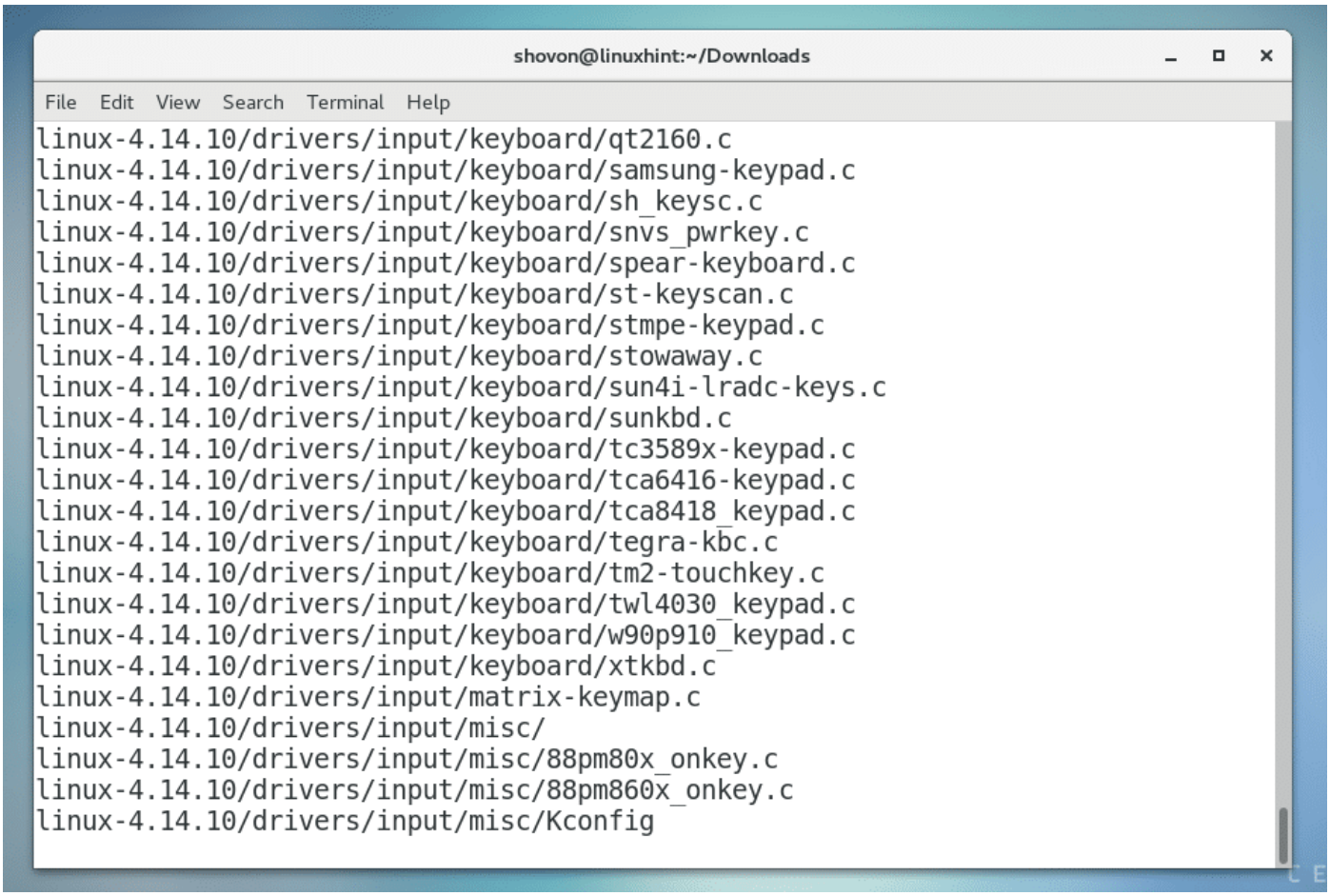


Now extract the compressed tar file with the following command:

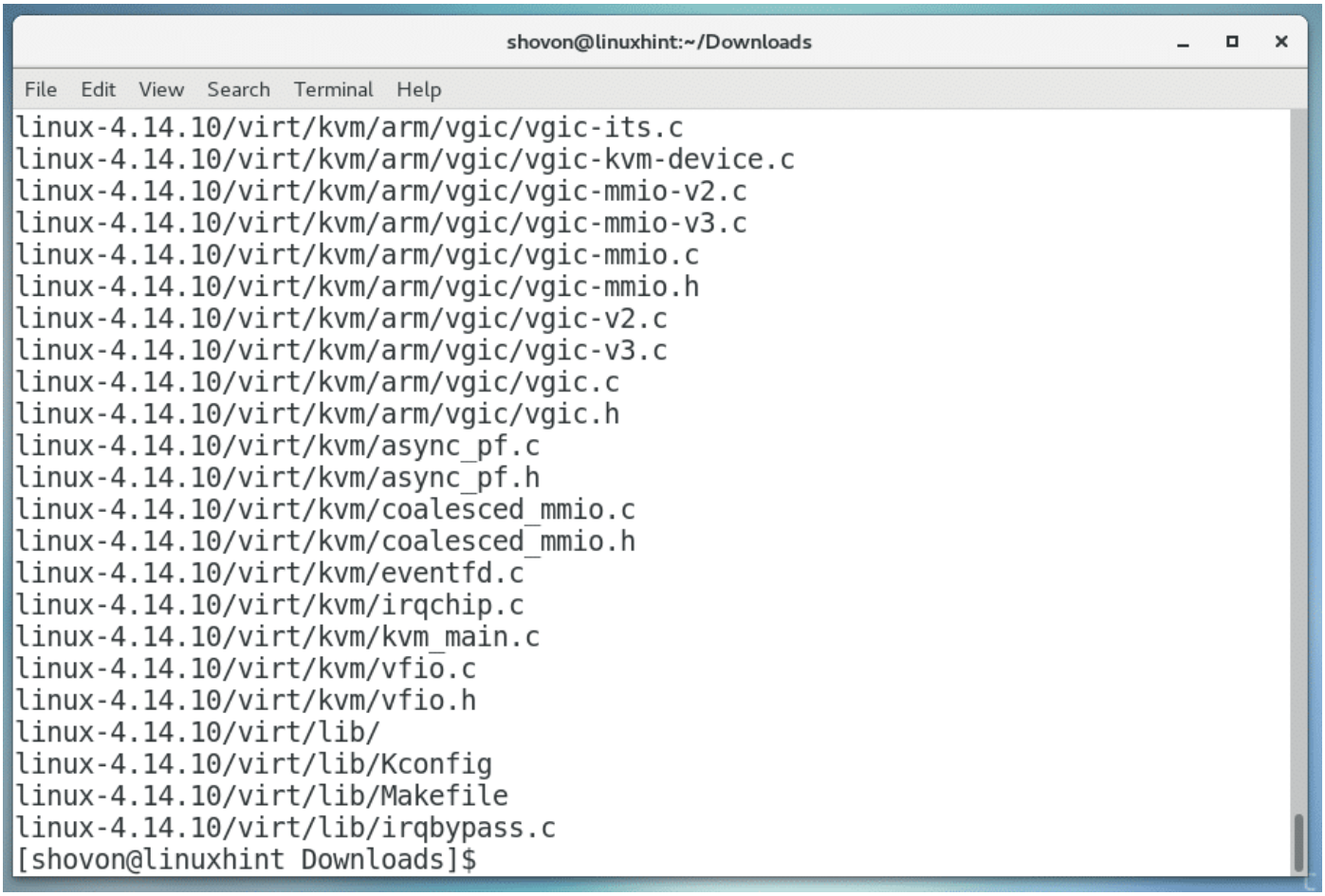
```
$ tar xvf linux-4.14.10.tar.xz
```



tar is extracting the compressed file.

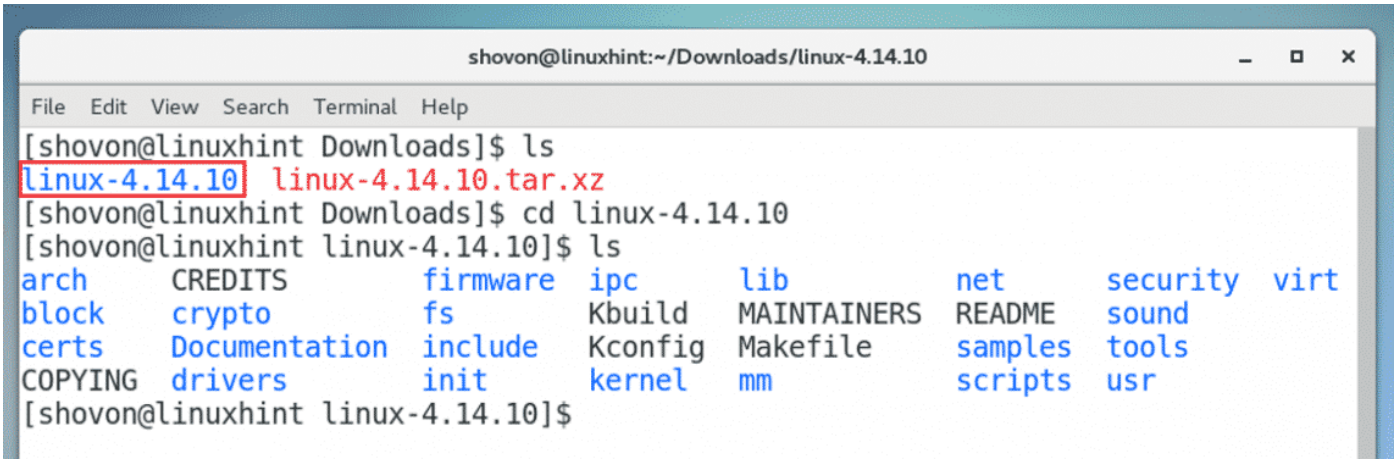


Once the file is extracted, you should see the following window.



After extraction, you should see a new directory as marked red in the screenshot below. Navigate to the directory with the following command.

```
$ cd linux-4.14.10
```

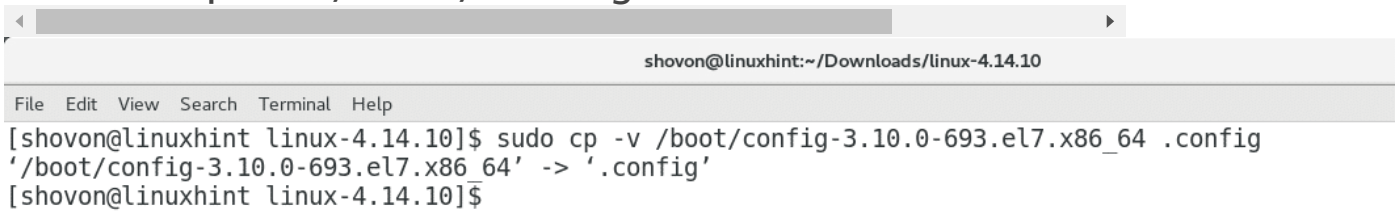


If you run the following command, you should see a list of config files used by the kernels that are installed on your system. You can run ‘uname -r’ command to find the one that you need. The filename should match with the output of the ‘uname -r’ command.

```
[shovon@linuxhint linux-4.14.10]$ ls /boot/config*
/boot/config-3.10.0-327.el7.x86_64  /boot/config-3.10.0-693.el7.x86_64
[shovon@linuxhint linux-4.14.10]$
```

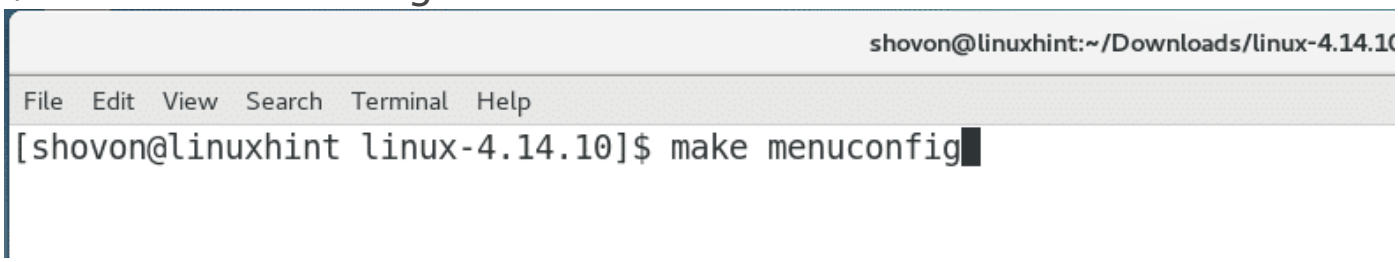
Then copy the config file to the linux-4.14.10 directory with the following command:

```
$ sudo cp -v /boot/config-3.10.0-693.el7.x86_
```

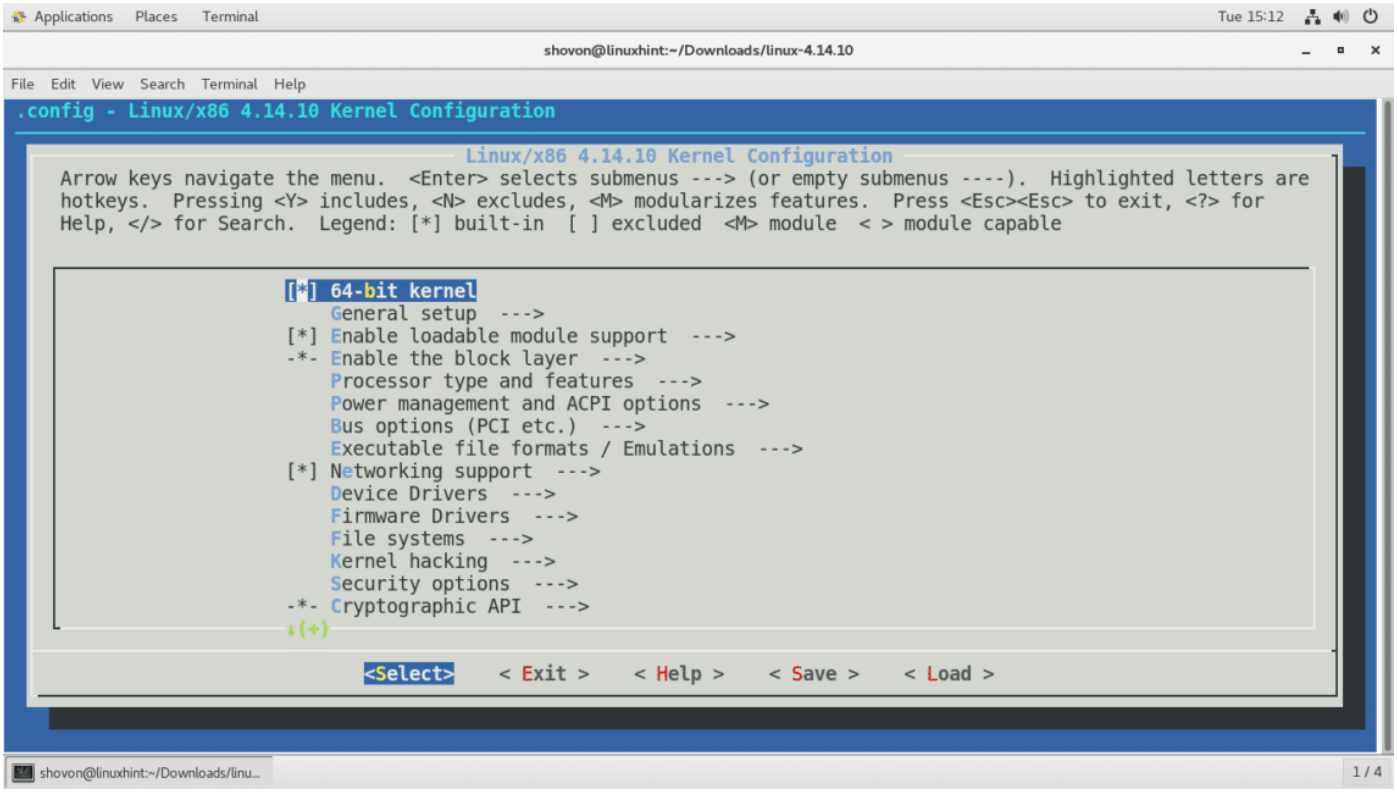


Now run the following command:

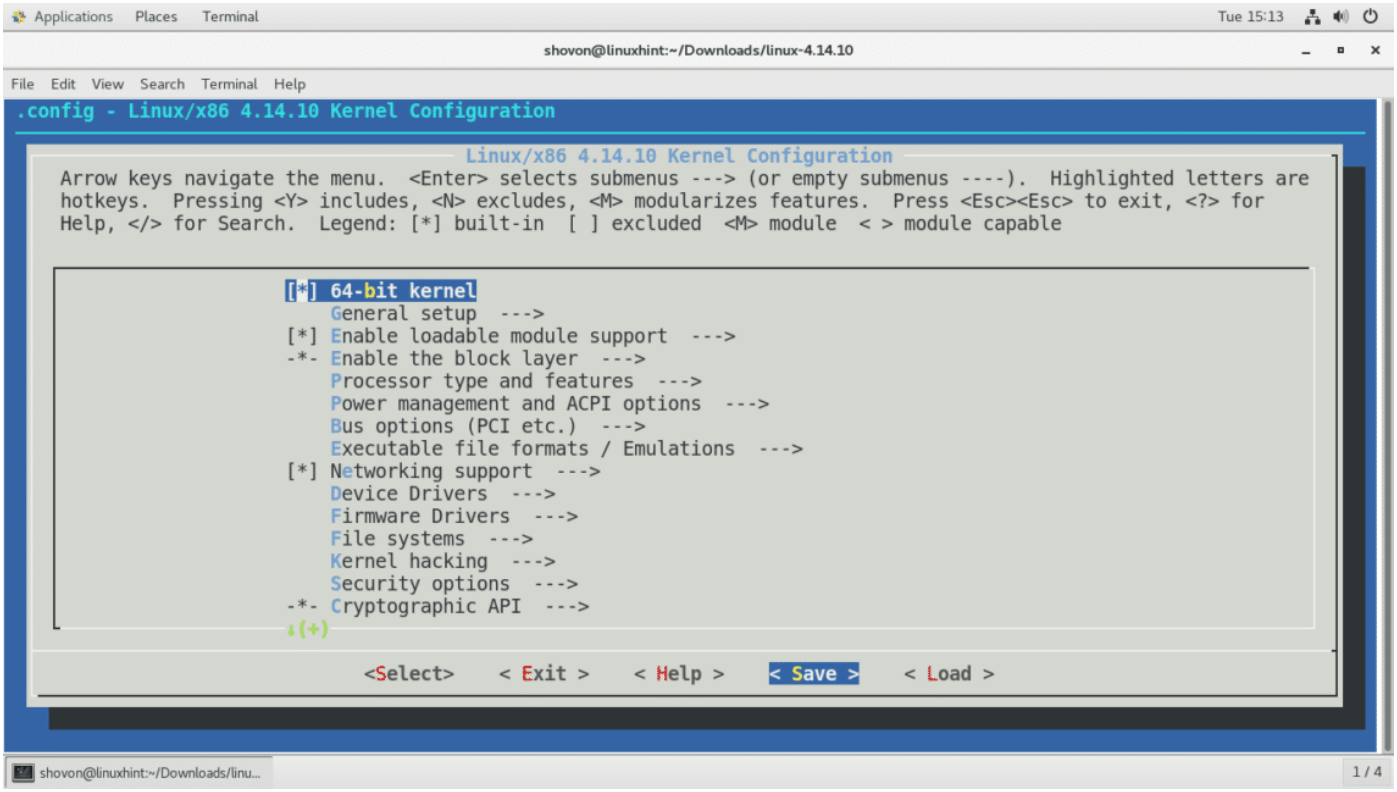
```
$ make menuconfig
```



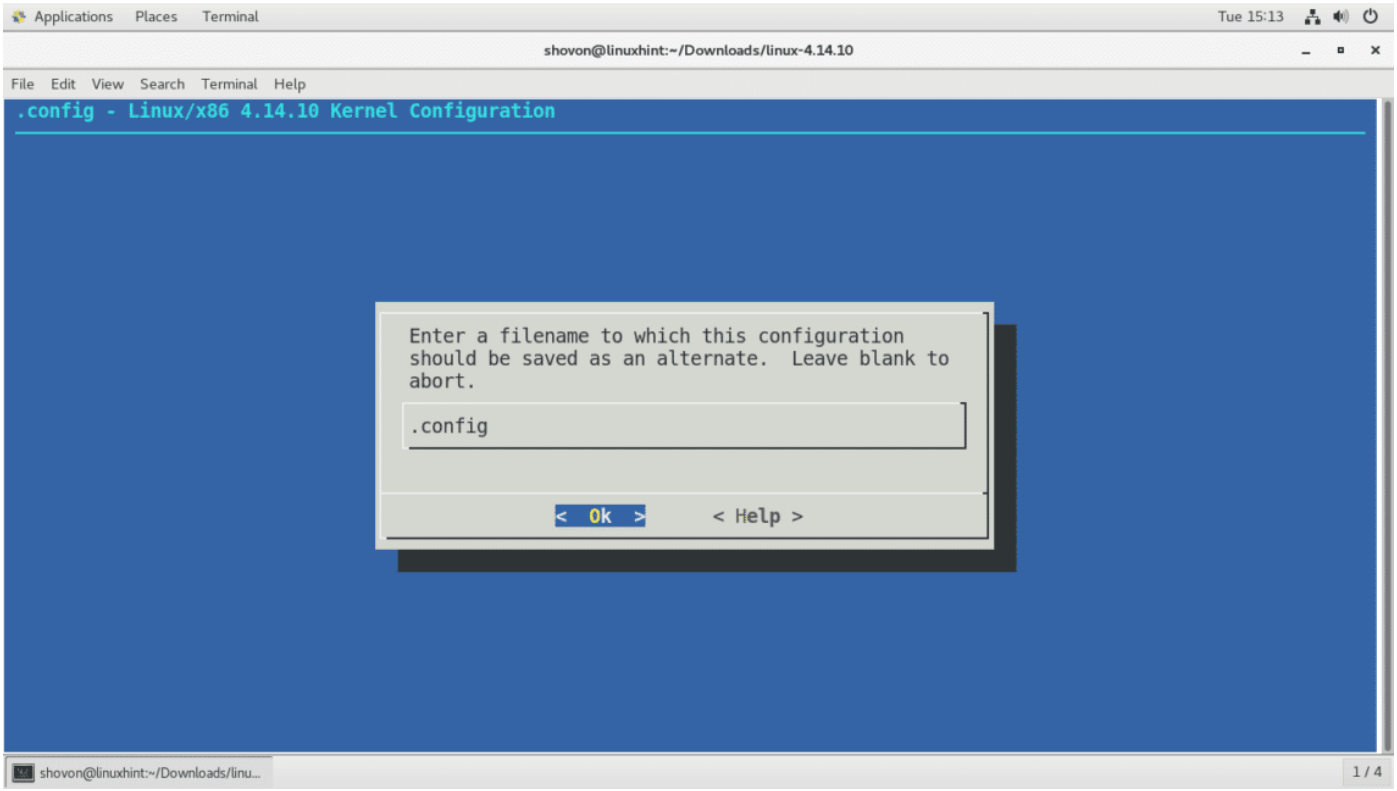
You should see the following window. This is where you enable or disable certain kernel features. If you don't know what to do here, then just leave the defaults.



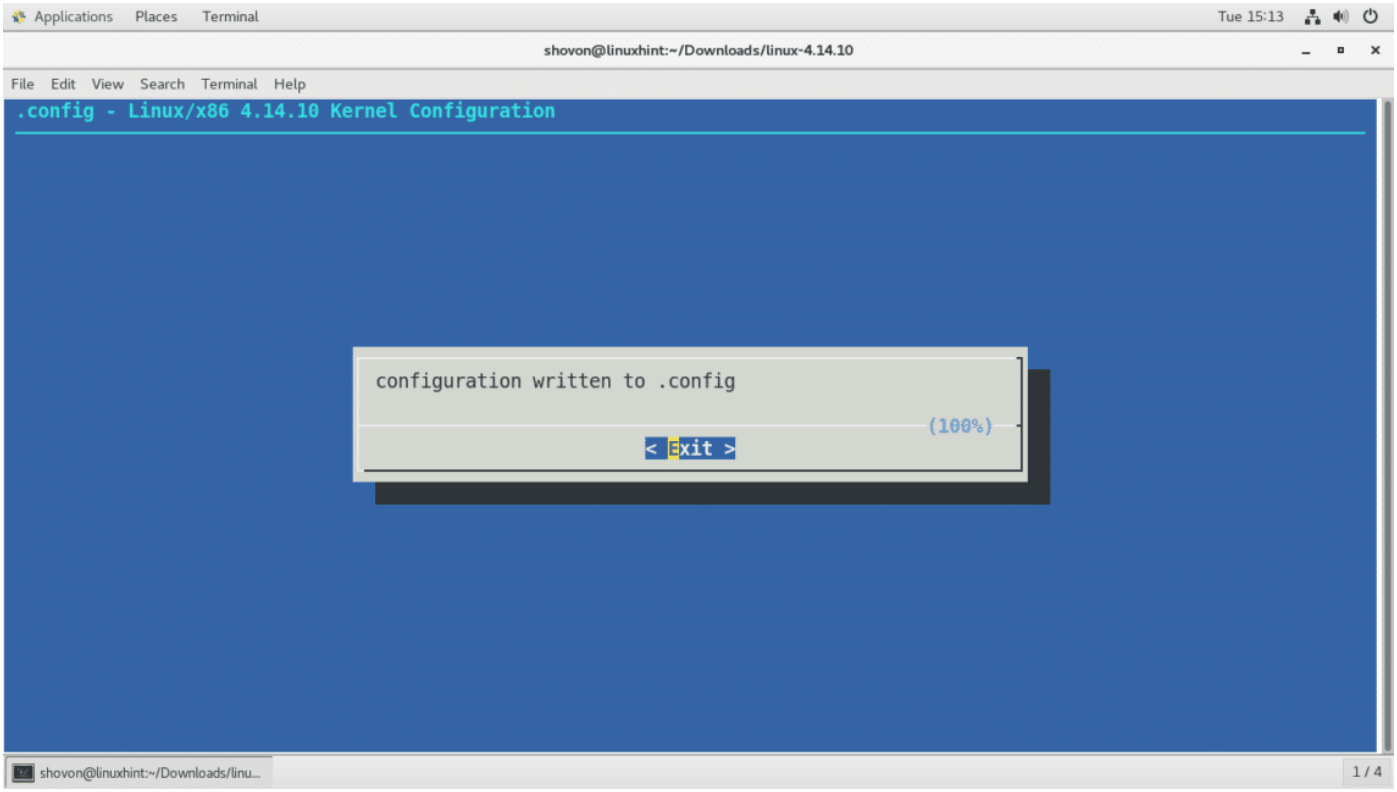
Once you're done, press <Left Arrow> button several times and go to "< Save >" as shown in the screenshot below. Then press <Enter>.



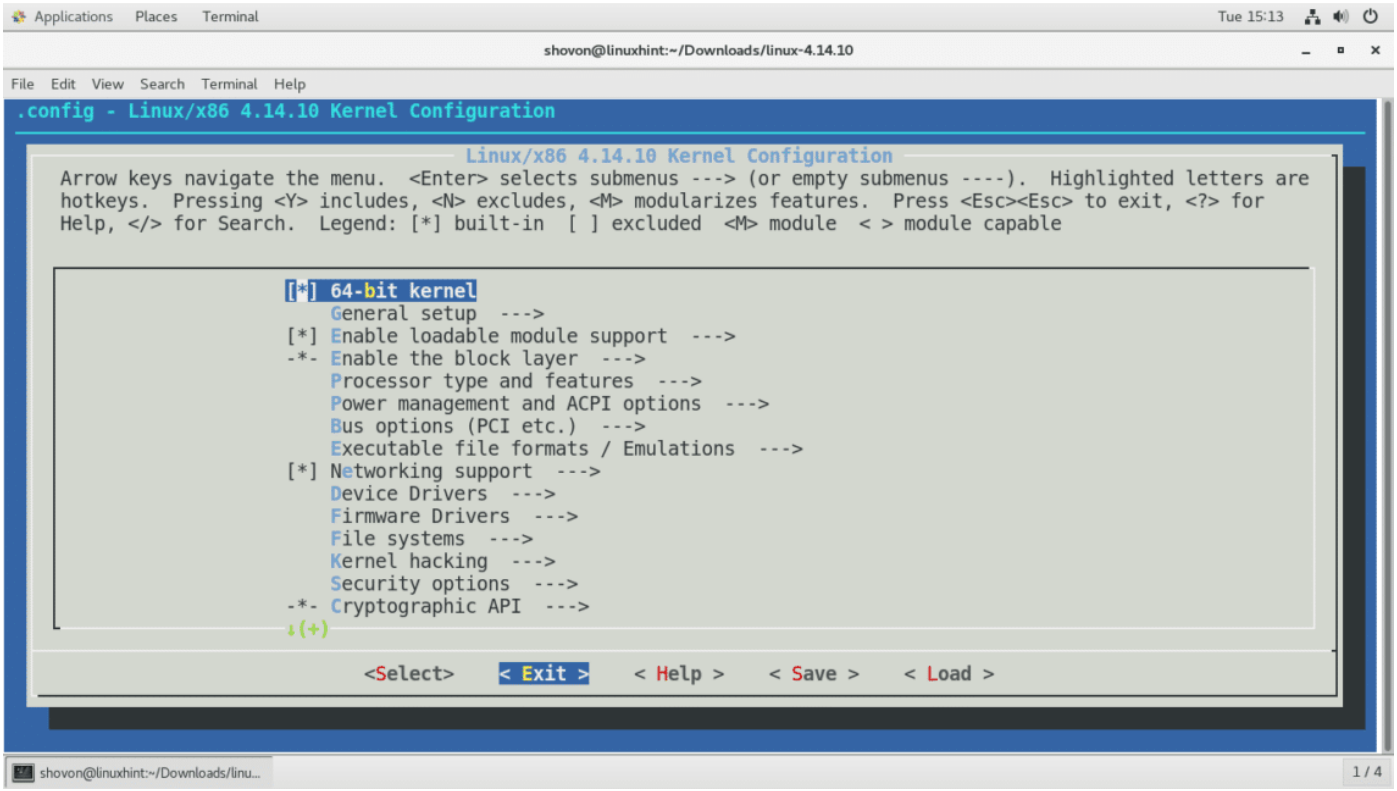
Then press <Enter> again.



Press <Enter> again.



Now navigate to “< Exit >” and press <Enter>



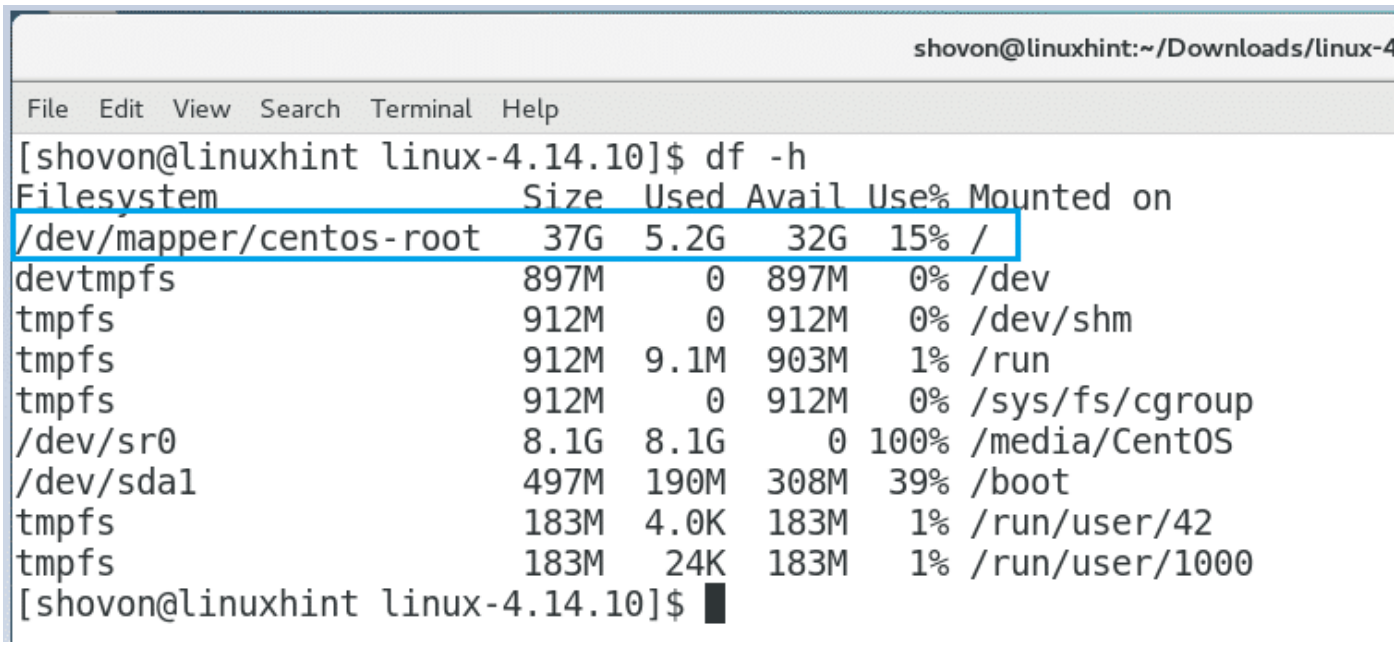
The .config file is updated for the new kernel.

```
[shovon@linuxhint linux-4.14.10]$ make menuconfig
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
.config:655:warning: symbol value 'm' invalid for CPU_FREQ_STAT


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

[shovon@linuxhint linux-4.14.10]$
```

Before you start compiling the new kernels, make sure you have more than 20GB of free space on the filesystem where you're compiling the kernel.

You can check how much space you have available with the following command:

```
$ df -h
```



Now run the following command to start the compilation process:

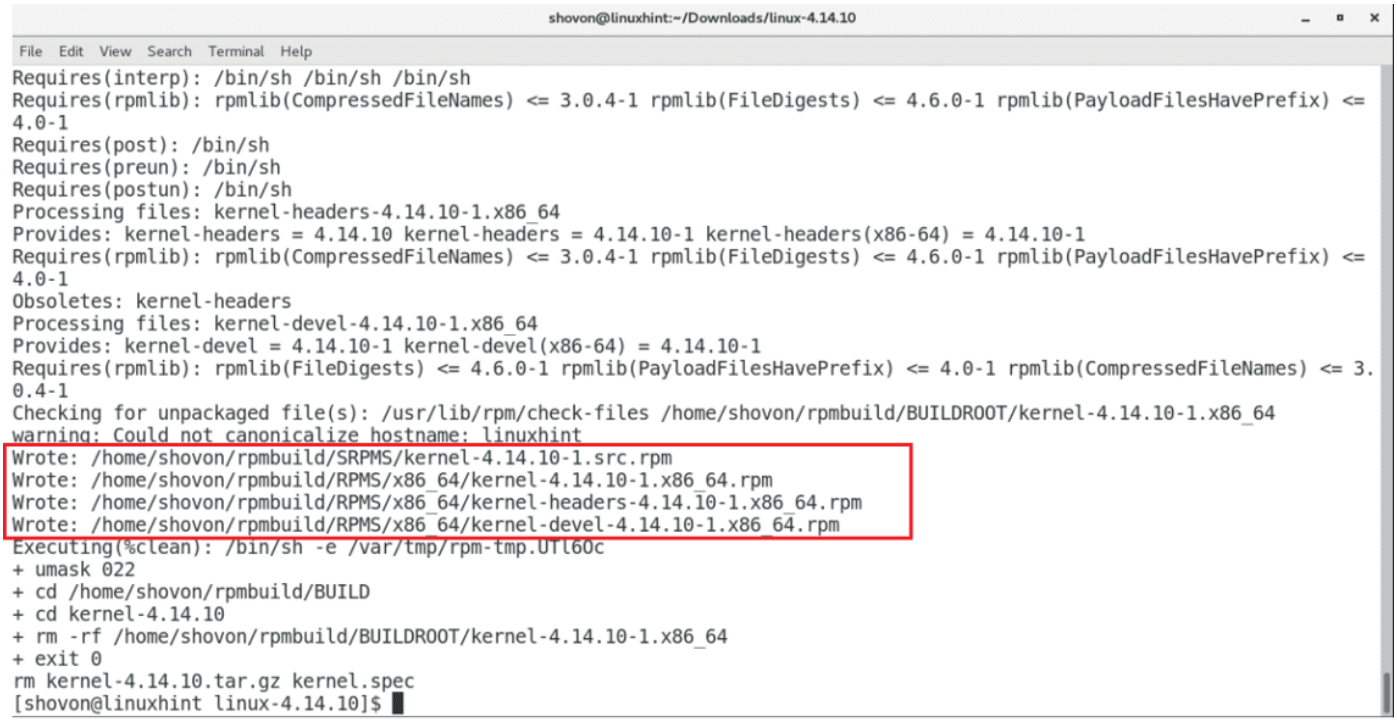
```
$ make rpm-pkg
```

```
shovon@linuxhint:~/Downr
File  Edit  View  Search  Terminal  Help
[shovon@linuxhint linux-4.14.10]$ make rpm-pkg
  CHK     include/config/kernel.release
  UPD     include/config/kernel.release
make clean
/bin/sh ./scripts/package/mkspec >./kernel.spec
  TAR     kernel-4.14.10.tar.gz
```

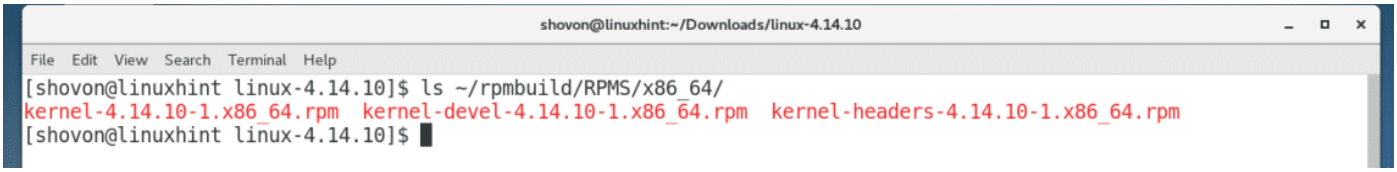
The kernel should be compiling just fine. It should take a long time.

```
shovon@linuxhint:~/Downloads/linux-4.14.10
File  Edit  View  Search  Terminal  Help
[shovon@linuxhint linux-4.14.10]$ make rpm-pkg
  CHK     include/config/kernel.release
make clean
/bin/sh ./scripts/package/mkspec >./kernel.spec
  TAR     kernel-4.14.10.tar.gz
rpmbuild  --target x86_64 -ta kernel-4.14.10.tar.gz
Building target platforms: x86_64
Building for target x86_64
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.jMN33Y
+ umask 022
+ cd /home/shovon/rpmbuild/BUILD
+ cd /home/shovon/rpmbuild/BUILD
+ rm -rf kernel-4.14.10
+ /usr/bin/gzip -dc /home/shovon/Downloads/linux-4.14.10/kernel-4.14.10.tar.gz
+ /usr/bin/tar -xf -
+ STATUS=0
+ '[' 0 -ne 0 ']'
+ cd kernel-4.14.10
+ /usr/bin/chmod -Rf a+rX,u+w,g-w,o-w .
+ exit 0
Executing(%build): /bin/sh -e /var/tmp/rpm-tmp.Ry7S1Z
+ umask 022
+ cd /home/shovon/rpmbuild/BUILD
+ cd kernel-4.14.10
+ make clean
```

On completion, you should see the following window. Some rpm package file was created on the user’s home directory as you can see from the screenshot.



The generated rpm package files.



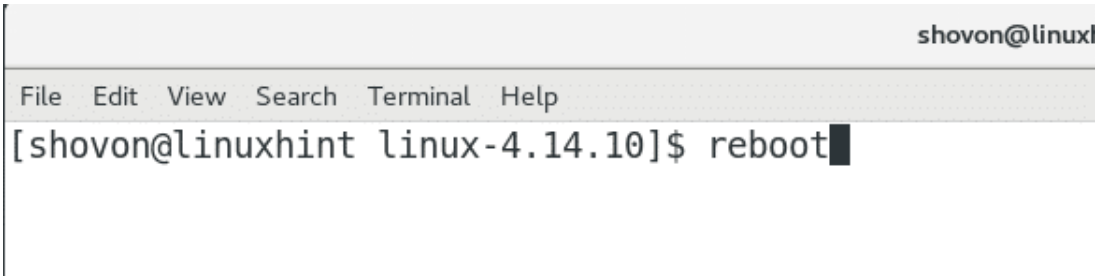
Now you can run the following command to install the rpm packages:

```
$ sudo rpm -iUv ~/rpmbuild/RPMS/x
```

```
[shovon@linuxhint linux-4.14.10]$ sudo rpm -iUv ~/rpmbuild/RPMS/x86_64/*.rpm
Preparing packages...
kernel-headers-4.14.10-1.x86_64
kernel-devel-4.14.10-1.x86_64
kernel-4.14.10-1.x86_64
kernel-headers-3.10.0-693.11.1.el7.x86_64
kernel-3.10.0-693.el7.x86_64
kernel-3.10.0-327.el7.x86_64
[shovon@linuxhint linux-4.14.10]$
```

Once the installation is complete, run the following command to restart your computer.

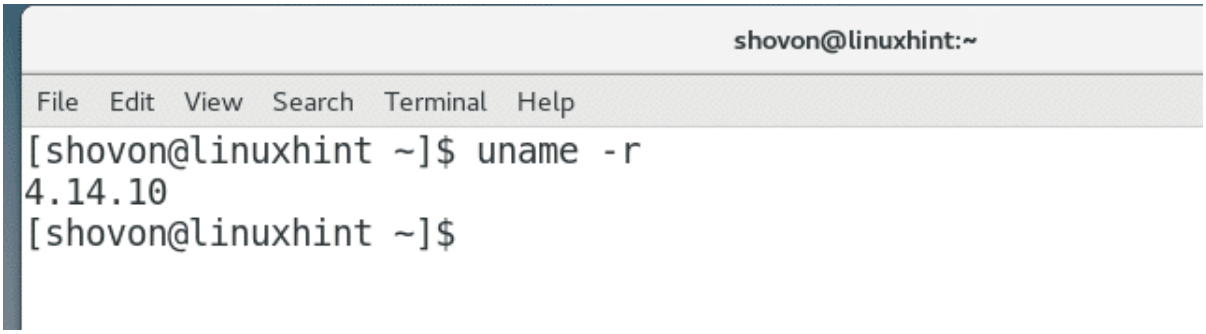
```
$ reboot
```



Once your computer starts, you can run the following command to check the version of the kernel that you're currently using.

```
$ uname -r
```

You should see that it's the version that you just installed. For me, it is '4.14.10'.



So that's how you compile the latest kernel from source and use it on CentOS 7. Thanks for reading this article.