I Need the Kernel Source

Created by JohnnyHughes. Currently maintained by AlanBartlett and AkemiYagi.

Note: the font used for commands, file names, URL's and paths has been changed to "DéjàVu Sans Mono" which can be downloaded free if you don't already have it.
IN that font there are:

- The letter "eye": **`I i`**
- The letter "ell": **`L l`**
- The digit "one": **`1`**

Contents

The following 2 links are connected to the ORIGINAL versions of these steps. You can refer to them for additional information if needed.

1. If you do not need the full kernel source (for CS350 and CS311, you DO need it, so skip to Step 2)

If you need to compile a kernel driver (module), the chances are you **do not** really need to install the full kernel source tree. You might just need to install the kernel-devel package. (If, however, you are certain that the full source tree is required, please follow the instructions in Section 2.)

In CentOS-7, there is just one kernel-devel package available:

- kernel-devel (64-bit architecture)

If you are running the standard kernel (for example), you can install the kernel-devel package by:

```
[student@host] su
enter the root password when prompted. Now you are the root user.
[root@host]# yum install kernel-devel
Now return to being the regular user
[root@host] exit
```

You can use this command to determine the version of your running kernel:

```
[user@host]# uname -r
```

The result will look similar to this:

```
For TJW: 2.6.32-504.23.4.el6.s390x


For VMWARE: 3.10.0-514.26.2.el7.x86_64
```

Please ensure that you install the kernel-devel package that **matches** your running kernel. See this FAQ for details.

If your kernel is not listed by yum because it is in an older tree, you can download it manually from the CentOS Vault.

CentOS-7: Look in either the **7.N.YYMM/os/x86_64/Packages/** or the **7.N.YYMM/updates/x86_64/Packages/** directories for the **kernel-devel-***version***.x86_64.rpm**

Prep for rebuild Centos 7

Once you have the proper **kernel**[*-type*]**-devel-***version.arch***.rpm** package installed, try to compile your module. It should work that way. If it does not, please provide feedback to the module's developer as this is the way all new kernel modules should be designed to be built.

2. If you really need the full kernel source (you do)

If you really must have the kernel source tree, for whatever reason, it is obtainable.

For all the following commands, replace ”user” with the “student” user id. Where you see “host”, you will see your TJW userid such as **TJW101** or for VMWare system **CS311-**{your BU userid})

As an ordinary user, **not root**, create a build tree based on a ~/rpmbuild/ directory. First, issue the “ls” command to see if you have an rpmbuild folder. If you do, remove it with: **rm -rf rpmbuild**.

```
Be careful of % and $ signs in the second command
[user@host ~]$ mkdir -p ~/rpmbuild/{BUILD,BUILDROOT,RPMS,SOURCES,SPECS,SRPMS}
[user@host ~]$ echo '%_topdir %(echo $HOME)/rpmbuild' > ~/.rpmmacros
```

You are **strongly** advised **against** package building as root. (See: Building Source RPM as non-root under CentOS)

To install the source package and tools for CentOS-7:

ALL USERS: do this ONE command (as student):

sudo yum install rpm-build

You will have to enter the student password. If it responds “nothing to do” you may SKIP step “a” and go to step “b”.

If it does the install, it means that step a must be done. You can either prefix each command with “sudo” as you just did, or you can issue the “su” command to switch to root to do them. **DON’T FORGET TO type the “exit” command to get back to the student account.**

a. As root, install the rpm-build, redhat-rpm-config, asciidoc, hmaccalc, perl-ExtUtils-Embed, ~~pesign~~, xmlto, audit-libs-devel, binutils-devel, elfutils-devel, elfutils-libelf-devel, ncurses-devel, newt-devel, ~~numactl-devel~~, pciutils-dedvel, python-devel and zlib-devel packages:

```
use the “su” command as you did before, then do these commands, then exit from the root
[root@host]# yum install redhat-rpm-config
[root@host]# yum install asciidoc hmaccalc perl-ExtUtils-Embed pesign xmlto
[root@host]# yum install audit-libs-devel binutils-devel elfutils-devel elfutils-
libelf-devel
[root@host]# yum install ncurses-devel newt-devel numactl-devel pciutils-devel
python-devel zlib-devel
```

b. find the kernel package you need (I did that for you, but the “how to” is here)

Normally, you would look for the kernel source rpm package in the *vault* and use its URL in the “*rpm -i”* command below. I have found it and the examples show you the command with the proper values inserted.

For VMware:

Replace the "N.YYMM" with the relevant sub-version, year and month numbers. For fall 2017, it’s 7.3.1611 and use the SECOND URL of these 2, in the command below, to pick up the original and all the updates

1. http://vault.centos.org/7.N.YYMM/os/Source/SPackages/ (original without patches)
2. http://vault.centos.org/7.N.YYMM/updates/Source/SPackages/ (original + patches)

For TJW, there is only 1 package usable (6.6) for our hardware:

- http://vault.centos.org/6.6/updates/Source/SPackages/

c. As an ordinary user, **not root**, install the source package by executing ONE of the following commands (whichever matches your system). EACH COMMAND shown here is really JUST ONE LINE. Be sure to enter it as ONE LINE. BE sure to use the package names/locations AS SHOWN.

For VMWare:

```
[user@host]$ rpm -i
http://vault.centos.org/7.3.1611/updates/Source/SPackages/kernel-3.10.0-
514.26.2.el7.src.rpm 2>&1 | grep -v exist
```

For TJW:

```
[user@host]$ rpm -i
http://vault.centos.org/6.6/updates/Source/SPackages/kernel-2.6.32-
504.30.3.el6.src.rpm 2>&1 | grep -v exist
```

d. **Now that the source package and tools are installed, unpack and prepare the source files**:

```
[user@host]$ cd ~/rpmbuild/SPECS

# now make some random numbers available to speed up processes that need them. It WILL make
a difference in how fast your build works. When prompted, enter your

# student password for this system. Don't forget the "&" which makes it run in the background.
This command may run for 45 minutes. Be patient

[user@host]$ sudo rngd -r /dev/urandom &

# Note: there are TWO dashes before the word target!!

[user@host SPECS]$ rpmbuild -bp --target=$(uname -m) kernel.spec
```

The value of $(uname -m) in the last command sets the target architecture for the build process to the same architecture as your current kernel. For the TJW machine, this value will be s390x. For VMWare, it will be x86_64. For safety, just use the *uname* as shown.

The kernel source tree will now be found under the ~/rpmbuild/BUILD/kernel*/linux*/ directory.

Now you are ready to make modifications to your system, rebuild it and make it bootable. Go to the W17-2 file.