

RHEL: Building a custom kernel on RHEL 6

Article Number: 124 | Rating: Unrated | Last Updated: Sat, Jun 2, 2018 8:41 AM

RHEL: Building a custom kernel on RHEL 6

```
# Tested on RHEL 6.4 i386
# Notes mainly from http://wiki.centos.org/HowTos/Custom\_Kernel

# WARNING: Please note that customizing kernel is not supported by
Red Hat Support Services

# NOTE: It is advisable to build packages as a non-root user as the
instructions for
#       building a given tarball may silently and invisibly change a
shared library, and
#       cause much damage to a system.

# Build preparations
# -----
# -----

# As a non-root user, create a build tree based on a ~/rpmbuild/
directory:

user@myhost:/home/user#> mkdir -p
~/rpmbuild/{BUILD,BUILDROOT,RPMS,SOURCES,SPECS,SRPMS}

user@myhost:/home/user#> echo '%_topdir %(echo $HOME)/rpmbuild' >
~/rpmmacros
```

```
# As root, install following needed packages:
```

```
root@myhost:/#> yum install rpm-build redhat-rpm-config asciidoc  
hmacalc
```

```
perl-ExtUtils-Embed xmlto audit-libs-devel binutils-devel  
elfutils-devel  
elfutils-libelf-devel newt-devel python-devel zlib-devel
```

```
# Obtain the kernel source rpm package and install it as a non-root  
user. For instance,
```

```
# for my RHEL 6 i386:
```

```
user@myhost:/home/user#> rpm -q kernel  
kernel-2.6.32-358.el6.i686
```

```
user@myhost:/home/user#> rpm -i kernel-2.6.32-431.20.3.el6.src.rpm
```

```
# Unpack and prepare the source files:
```

```
user@myhost:/home/user#> cd ~/rpmbuild/SPECS
```

```
user@myhost:/home/user/rpmbuild/SPECS#> rpmbuild -bp --target=$(uname  
-m) kernel.spec
```

```
Building target platforms: i686
```

```
Building for target i686
```

```
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.RsE9ro
```

```
+ umask 022
```

```
+ cd /home/user/rpmbuild/BUILD
```

```
+ LANG=C
```

```
+ export LANG
```

```
+ unset DISPLAY
```

```
+ patch_command='patch -p1 -F1 -s'
```

```
+ '[' '!' -d
```

```
kernel-2.6.32-431.20.3.el6/vanilla-2.6.32-431.20.3.el6/ ']'
```

```
+ rm -f pax_global_header
```

```
+ cd /home/user/rpmbuild/BUILD
[...]
```

```
+ gcc -o scripts/bin2c scripts/bin2c.c
+ scripts/bin2c ksign_def_public_key __initdata
+ cd ..
+ exit 0
```

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

```
# If you have any kernel patches to add, copy them now to
~/rpmbuild/SOURCES/ directory.
```

```
# Otherwise, if needed, you can create your own kernel patch
following this procedure:
```

```
# Copy the source tree to preserve the original tree while making
changes to the new one
```

```
user@myhost:/home/user/#> cd ~/rpmbuild/BUILD
```

```
user@myhost:/home/user/rpmbuild/BUILD#> cp -pr
kernel-2.6.32-431.20.3.el6 kernel-2.6.32-431.20.3.el6.new
```

```
# Make necessary modifications to your custom kernel. Me, for
instance, I'm just modifying
# a single commentary line in following file:
```

```
user@myhost:/home/user/rpmbuild/BUILD#> vi ./kernel-2.6.32-431.20.3.e
16.new/linux-2.6.32-431.20.3.el6.i686/kernel/time.c
```

```
# After the .new source tree is modified, generate the patch. To
generate the patch,
# run diff against the entire .new and original source trees with the
following command:
```

```
user@myhost:/home/user/rpmbuild/BUILD#> diff -uNrp
kernel-2.6.32-431.20.3.el6 kernel-2.6.32-431.20.3.el6.new >
../SOURCES/mypatch.patch

# Now, edit the patch file to remove the leading directory, this is
required because
# the kernel spec file applies patches with '-p1' only.

user@myhost:/home/user/rpmbuild/BUILD#> vi
~/rpmbuild/SOURCES/mypatch.patch

# For example, change the following:

--- kernel-2.6.32-431.20.3.el6/linux-2.6.32-431.20.3.el6.i686/kernel/
time.c      2014-06-06 23:42:44.000000000 +0200
+++ kernel-2.6.32-431.20.3.el6.new/linux-2.6.32-431.20.3.el6.i686/ker
nel/time.c  2014-09-25 17:27:06.955461886 +0200

# To:

--- linux-2.6.32-431.20.3.el6.i686/kernel/time.c      2014-06-06
23:42:44.000000000 +0200
+++ linux-2.6.32-431.20.3.el6.i686/kernel/time.c      2014-09-25
17:27:06.955461886 +0200

# My new patch is ready

# Configure the kernel
# -----
-----

# Change directory to ~/rpmbuild/BUILD/kernel-*/linux-*/ and copy
into this directory,
# as .config, following file:
```

```
user@myhost:/#> cd ~/rpmbuild/BUILD/kernel-*/linux-*/

user@myhost:/#> cp configs/kernel-2.6.32-`uname -m`.config .config

# First run 'make oldconfig'

user@myhost:/#> make oldconfig
  scripts/kconfig/conf -o arch/x86/Kconfig
#
# configuration written to .config
#

# Now you should run 'make menuconfig' and do the necessary
selections. Once complete,
# remember to save your changes.

# make menuconfig is one of five similar tools that can configure the
Linux kernel source,
# a necessary early step needed to compile the source code. 'make
menuconfig', with a
# menu-driven user interface, allows the user to choose the features
of the Linux kernel
# (and other options) that will be compiled. It is normally invoked
using the command
# 'make menuconfig', 'menuconfig' is a target in the Linux kernel
Makefile.

# 'make menuconfig' was not in the first version of the Linux kernel.
The predecessor tool
# is a question-and-answer-based utility ('make config', 'make
oldconfig'). A third tool
# for Linux configuration is 'make xconfig', which requires Qt. There
is also
# 'make gconfig', which uses GTK+, and 'make nconfig', which is
similar to 'make menuconfig'.
```

```
user@myhost:/#> make menuconfig
scripts/kconfig/mconf arch/x86/Kconfig
#
# configuration written to .config
#

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

# If you have installed the full kernel source to build a kernel
module, you should do
# it at this point.

# Next, add a line that contains the commented out equivalent of the
hardware platform to
# the top of the configuration file (equivalent to the output
returned by a 'uname -i'
# command) just before you copy it back to the configs/ directory.
This will either be
# "i386" for the 32-bit architecture or "x86_64" for the 64-bit
architecture. It needs
# to be commented out with a "#" and must be the first line of the
file. Note that there
# must be a space between the hash symbol and the hardware platform
descriptor.

user@myhost:/#> vi .config

# Add, as the first line of the .config file, either:

# i386

# - or -

# x86_64
```

```
# Copy the .config file back to the configs/ directory:
```

```
user@myhost:/#> cp .config configs/kernel-2.6.32-`uname -m`.config
```

```
# The final step is to copy the entire contents of the configs/  
directory to the
```

```
# ~/rpmbuild/SOURCES/ directory.
```

```
user@myhost:/#> cp configs/* ~/rpmbuild/SOURCES/
```

```
# Modifying the kernel specification file
```

```
# -----  
-----
```

```
# The kernel specification file should now be modified.
```

```
user@myhost:/#> cd ~/rpmbuild/SPECS/
```

```
user@myhost:/#> cp kernel.spec kernel.spec.distro
```

```
user@myhost:/#> vi kernel.spec
```

```
# At line 18, the definition of buildid is commented out. This must  
be uncommented and
```

```
# given a value to avoid a conflict with your currently installed  
kernel. Change the line
```

```
# in a similar manner to the example below:
```

```
%define buildid .mykernel
```

```
# There should be no space between the "%" and the word "define".
```

```
# If you have any patches to apply, you need to make reference to  
them in two places:
```

```
# 1.- Just before line 613 (which reads "# empty final patch file to
facilitate testing of
# kernel patches"), add your declaration starting with the number
40000, so that your
# patch is not in any danger of conflicting with the RHEL kernel
patch space. Example:
```

```
Patch40000: mypatch.patch
```

```
# 2.- Just before line 935 (which reads, "ApplyOptionalPatch linux-
kernel-test.patch"),
# add a line to apply your patch. For example:
```

```
ApplyOptionalPatch mypatch.patch
```

```
# Replace line 929:
```

```
cp $RPM_SOURCE_DIR/config-* .
```

```
# with:
```

```
cp $RPM_SOURCE_DIR/kernel-*.config .
```

```
# Comment out line 933:
```

```
#make -f %SOURCE20 VERSION=%version configs
```

```
# Build the new kernel
```

```
# -----  
-----
```

```
# WARNING: Pay attention to the free space on the filesystem you are  
building the new
```

```
# kernel. For me it took about 7GB:
```

```
#
# user@myhost:/home/user/rpmbuild#> #> du -hs .
# 6.9G      .

# Start the build:

user@myhost:/#> cd ~/rpmbuild/SPECS/

user@myhost:/#> rpmbuild -bb --target=`uname -m` kernel.spec 2> build-
err.log | tee build-out.log
    Building target platforms: i686
    Building for target i686
    Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.Ua66Tz
    ###
[... ]
    Executing(%build): /bin/sh -e /var/tmp/rpm-tmp.8SaeUj
    BUILDING A KERNEL FOR i686...
    USING ARCH=i386
    3f47dc2f8a3f1f9fc4205205cdf132ebb09545ce
    2b3af23cd31109682bf94574bed4f13ff25b4c35
    abc7b98eb5ff6aaa3cafa8d3c37e7a12428ff496
    5b1d6b49a129f51f9db7f96c5ce0160ef02dd17e
    cb36c6d75b023a9ad75d5475b2960dd2cbd09a4f
    cb36c6d75b023a9ad75d5475b2960dd2cbd09a4f
[... ]
    Checking for unpackaged file(s): /usr/lib/rpm/check-files /home/us
er/rpmbuild/BUILDROOT/kernel-2.6.32-431.20.3.el6.mykernel.i386
    Wrote: /home/user/rpmbuild/RPMS/i686/kernel-2.6.32-431.20.3.el6.my
kernel.i686.rpm
    Wrote: /home/user/rpmbuild/RPMS/i686/kernel-
headers-2.6.32-431.20.3.el6.mykernel.i686.rpm
    Wrote: /home/user/rpmbuild/RPMS/i686/kernel-debuginfo-common-
i686-2.6.32-431.20.3.el6.mykernel.i686.rpm
    Wrote: /home/user/rpmbuild/RPMS/i686/perf-2.6.32-431.20.3.el6.myke
rnel.i686.rpm
    Wrote: /home/user/rpmbuild/RPMS/i686/perf-
```

```
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
  Wrote: /home/user/rpmbuild/RPMS/i686/python-
perf-2.6.32-431.20.3.el6.mykernel.i686.rpm
  Wrote: /home/user/rpmbuild/RPMS/i686/python-perf-
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
  Wrote: /home/user/rpmbuild/RPMS/i686/kernel-
devel-2.6.32-431.20.3.el6.mykernel.i686.rpm
  Wrote: /home/user/rpmbuild/RPMS/i686/kernel-
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
  Wrote: /home/user/rpmbuild/RPMS/i686/kernel-
debug-2.6.32-431.20.3.el6.mykernel.i686.rpm
  Wrote: /home/user/rpmbuild/RPMS/i686/kernel-debug-
devel-2.6.32-431.20.3.el6.mykernel.i686.rpm
  Wrote: /home/user/rpmbuild/RPMS/i686/kernel-debug-
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
  Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.R5KFyB
```

```
# For kernels >= 2.6.18-53.el5, you can add some useful options to
the rpmbuild command by
# using the --with and/or --without flags and associated arguments.
The main options are:
```

```
--with baseonly
--with xenonly
--without up
--without xen
--without debug
--without debuginfo
--without fips
--without kabichk
```

```
# When the build completes, your custom kernel rpm files will be
found in the
# ~/rpmbuild/RPMS/`uname -m`/ directory. Make sure that you install
those files, as root,
```

```
# using an 'rpm -ivh kernel-*.rpm' command.

# Note: If you have built a kernel version that is older than a
currently installed
# version you will also have to use the --oldpackage flag with the
rpm command.

# UNDER NO CIRCUMSTANCES use an 'rpm -Uvh' command to install your
kernel as this will
# update (overwrite) the currently installed version. Hence if you
have a problem with
# your custom kernel, you will not be able to revert to the previous,
working, version.
```

```
root@myhost:/#> cd /home/user/rpmbuild/RPMS/`uname -m`
```

```
root@myhost:/home/user/rpmbuild/RPMS/i686#> ll
```

```
total 628924
-rw-rw-r-- 1 user group 26844728 Sep 19 04:49
kernel-2.6.32-431.20.3.el6.mykernel.i686.rpm
-rw-rw-r-- 1 user group 27460356 Sep 19 05:02 kernel-
debug-2.6.32-431.20.3.el6.mykernel.i686.rpm
-rw-rw-r-- 1 user group 261491476 Sep 19 05:11 kernel-debug-
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
-rw-rw-r-- 1 user group 9221168 Sep 19 05:02 kernel-debug-
devel-2.6.32-431.20.3.el6.mykernel.i686.rpm
-rw-rw-r-- 1 user group 255223848 Sep 19 05:01 kernel-
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
-rw-rw-r-- 1 user group 40208252 Sep 19 04:51 kernel-debuginfo-
common-i686-2.6.32-431.20.3.el6.mykernel.i686.rpm
-rw-rw-r-- 1 user group 9181032 Sep 19 04:52 kernel-
devel-2.6.32-431.20.3.el6.mykernel.i686.rpm
-rw-rw-r-- 1 user group 2996324 Sep 19 04:49 kernel-
headers-2.6.32-431.20.3.el6.mykernel.i686.rpm
-rw-rw-r-- 1 user group 3067580 Sep 19 04:51
perf-2.6.32-431.20.3.el6.mykernel.i686.rpm
-rw-rw-r-- 1 user group 3673100 Sep 19 04:51 perf-
```

```
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
  -rw-rw-r-- 1 user group 2292360 Sep 19 04:51 python-
perf-2.6.32-431.20.3.el6.mykernel.i686.rpm
  -rw-rw-r-- 1 user group 2338372 Sep 19 04:51 python-perf-
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
```

```
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv <rpms>
```

```
# Note: Starting from kernel-2.6.32-358.el6, if "bfa-firmware"
package is already installed on the system,
# then it has to be above a certain version, otherwise the
installation of kernel will cause a conflict:
```

```
#
#          bfa-firmware < 3.2.21.1-2 conflicts with
kernel-2.6.32-431.20.3.el6.mykernel.i686
```

```
#
# If bfa-firmware is not required for other packages, we can remove
it from our system
```

```
root@myhost:/#> rpm -q --whatrequires bfa-firmware
no package requires bfa-firmware
```

```
root@myhost:/#> rpm -e bfa-firmware
```

```
# Apart from that I'm running into problems because kernel requires a
newer version of kernel-firmware
```

```
# This is the message I'm receiving:
```

```
#
#          kernel-firmware >= 2.6.32-431.20.3.el6.mykernel is needed by
kernel-2.6.32-431.20.3.el6.mykernel.i686
```

```
#
# so I decided to rebuild the kernel adding the option --with
firmware in order to generate the
# kernel-firmware package too
```

```
# Connected as our non-root user:
```

```
user@myhost:/#> cd ~/rpmbuild/SPECS/
```

```
user@myhost:/home/user/rpmbuild/SPECS#> rpmbuild -bb --target=`uname  
-m` --with firmware kernel.spec 2> build-err2.log | tee build-  
out2.log
```

```
# As root
```

```
root@myhost:/#> cd /home/user/rpmbuild/RPMS/`uname -m`
```

```
root@myhost:/home/user/rpmbuild/RPMS/i686#> ll kernel-firmware*  
-rw-rw-r-- 1 user group 13627368 Sep 25 03:47 kernel-  
firmware-2.6.32-431.20.3.el6.mykernel.i686.rpm
```

```
# Now, we can start installing packages normally
```

```
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv kernel-  
firmware-2.6.32-431.20.3.el6.mykernel.i686.rpm
```

```
Preparing...
```

```
##### [100%]
```

```
file /usr/share/doc/kernel-firmware-2.6.32/WHENCE from  
install of kernel-firmware-2.6.32-431.20.3.el6.mykernel.i686  
conflicts with file from package kernel-  
firmware-2.6.32-358.el6.noarch
```

```
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm --replacefiles -ihv  
kernel-firmware-2.6.32-431.20.3.el6.mykernel.i686.rpm
```

```
Preparing...
```

```
##### [100%]
```

```
1:kernel-firmware
```

```
##### [100%]
```

```
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv
kernel-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv kernel-
debug-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv kernel-debuginfo-
common-i686-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv kernel-debug-
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv kernel-debug-
devel-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv kernel-
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv kernel-
devel-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm --replacefiles -ihv
kernel-headers-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv
perf-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv perf-
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv python-
perf-2.6.32-431.20.3.el6.mykernel.i686.rpm
root@myhost:/home/user/rpmbuild/RPMS/i686#> rpm -ihv python-perf-
debuginfo-2.6.32-431.20.3.el6.mykernel.i686.rpm

# Verify/modify grub's grub.conf file to add new custom kernel:

title CentOS (2.6.32-431.20.3.el6.mykernel.i686)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-431.20.3.el6.mykernel.i686 ro
root=/dev/mapper/rootvg-lv_root rd_NO_LUKS KEYBOARDTYPE=pc
KEYTABLE=fr LANG=en_US.UTF-8 rd_LVM_LV=rootvg/lv_swap rd_NO_MD
SYSFONT=lataarcyheb-sun16 crashkernel=auto rd_LVM_LV=rootvg/lv_root
rd_NO_DM quiet
    initrd /initramfs-2.6.32-431.20.3.el6.mykernel.i686.img
```

Posted - Sat, Jun 2, 2018 8:41 AM. This article has been viewed 5151 times.

Online URL: <http://kb.ictbanking.net/article.php?id=124>