nixCraft

Linux and Unix tutorials for new and seasoned sysadmin.

How to Compile and Install Linux Kernel v4.5 Source On a Debian / Ubuntu Linux

by VIVEK GITE *on* SEPTEMBER 11, 2015 *last updated* MARCH 21, 2016
in DEBIAN / UBUNTU, LINUX, PACKAGE MANAGEMENT, UBUNTU LINUX

How do I download, compile and install the latest version of the Linux kernel on a Debian Linux v8.x or Ubuntu Linux LTS home server or my laptop? How do I build and install a custom Linux kernel on a Debian or Ubuntu Linux based system?

In order to create a custom kernel configuration file and build a custom kernel, the full Linux kernel source tree must first be downloaded and installed. The latest Linux kernel stable version is **4.5**. In this tutorial, you will **learn how to compile the Linux kernel version 4.5 on a Debian and Ubuntu Linux** operating system and build .deb file.

Why build a custom kernel?

Compiling a custom Linux kernel has its advantages and disadvantages. To change the kernel's behavior, one had to compile and then reboot into a new Linux. Most of the functionality in the Linux kernel contained in modules that can be dynamically loaded and unloaded from the kernel as necessary. Some benefits of a custom Linux kernel:

1. Support a wide range of hardware including the latest hardware.
2. Remove unwanted drivers from the kernel.

3. Faster boot time due to small kernel size.
4. Increased security due to additional or removed modules/drivers/features.
5. You will learn about the kernel and advanced usage.
6. Always run the cutting edge latest kernel.
7. Lower memory usage.

Note: The following instructions were tested on both Debian Linux v8.x and Ubuntu Linux v14.04.4 LTS.

Prerequisites

You need to install the following packages on a Debian or Ubuntu Linux to compiler the Linux kernel:

- **git** : Fast, scalable, distributed revision control system. You can grab the latest source code using the git command.
- **fakeroot** : Tool for simulating superuser privileges. Useful to build .deb files.
- **build-essential** : Tools for building the Linux kernel such as GCC compiler and related tools on a Debian or Ubuntu Linux based system.
- **ncurses-dev** : Developer's libraries for ncurses. This is used by menuconfig while configuring the kernel options.
- **kernel-package** : Utility for building Linux kernel related Debian packages.
- **xz-utils** : XZ-format compression utilities to decompress the Linux kernel tar ball.
- **Disk space** : 10 GB or more free disk space.
- **Time** : Kernel compilation may take quite a while, depending on the power of your machine.

Install required packages

Open the terminal application. Type the following apt-get command to install the required packages for building the Linux kernel:

```
$ sudo apt-get install git fakeroot build-essential
```

```
ncurses-dev xz-utils libssl-dev bc
```

Sample outputs:

```
veryv@instance-1:~$ sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libncurses5-dev' instead of 'ncurses-dev'
The following extra packages will be installed:
  binutils bzip2 cpp cpp-4.9 dpkg-dev g++ g++-4.9 gcc gcc-4.9 git-man
  libalgorithm-c3-perl libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libarchive-extract-perl libasan1 libatomic1
  libc-dev-bin libc6-dev libcgi-fast-perl libcgi-pm-perl libcilkrts5
  libclass-c3-perl libclass-c3-xs-perl libcloog-isl4 libcpan-meta-perl
  libcurl3-gnutls libdata-optlist-perl libdata-section-perl libdpkg-perl
  liberror-perl libfakeroot libfcgi-perl libfile-fcntllock-perl libgcc-4.9-dev
  libgomp1 libisl10 libitm1 liblog-message-perl liblog-message-simple-perl
  liblsan0 libmodule-build-perl libmodule-pluggable-perl
  libmodule-signature-perl libmpc3 libmpfr4 libmro-compat-perl
  libpackage-constants-perl libparams-util-perl libpod-latex-perl
  libpod-readme-perl libquadmath0 libregexp-common-perl
  libsoftware-license-perl libstdc++-4.9-dev libsub-exporter-perl
  libsub-install-perl libterm-ui-perl libtext-soundex-perl
  libtext-template-perl libtimedate-perl libtinfo-dev libtsan0 libubsan0
  linux-libc-dev make manpages-dev patch perl perl-modules rename
Suggested packages:
  binutils-doc bzip2-doc cpp-doc gcc-4.9-locales debian-keyring g++-multilib
  g++-4.9-multilib gcc-4.9-doc libstdc++6-4.9-dbg gcc-multilib autoconf
  automake libtool flex bison gdb gcc-doc gcc-4.9-multilib libgcc1-dbg
  libgomp1-dbg libitm1-dbg libatomic1-dbg libasan1-dbg liblsan0-dbg
  libtsan0-dbg libubsan0-dbg libcilkrts5-dbg libquadmath0-dbg git-daemon-run
  git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-arch
  git-cvs git-mediawiki git-svn glibc-doc ncurses-doc libstdc++-4.9-doc
  make-doc ed diffutils-doc perl-doc libterm-readline-gnu-perl
  libterm-readline-perl-perl libb-lint-perl libcpanplus-dist-build-perl
  libcpanplus-perl libfile-checktree-perl libobject-accessor-perl
Recommended packages:
  libarchive-tar-perl
The following NEW packages will be installed:
  binutils build-essential bzip2 cpp cpp-4.9 dpkg-dev fakeroot g++ g++-4.9 gcc
  gcc-4.9 git git-man libalgorithm-c3-perl libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libarchive-extract-perl
  libasan1 libatomic1 libc-dev-bin libc6-dev libcgi-fast-perl libcgi-pm-perl
  libcilkrts5 libclass-c3-perl libclass-c3-xs-perl libcloog-isl4
  libcpan-meta-perl libcurl3-gnutls libdata-optlist-perl libdata-section-perl
  libdpkg-perl liberror-perl libfakeroot libfcgi-perl libfile-fcntllock-perl
  libgcc-4.9-dev libgomp1 libisl10 libitm1 liblog-message-perl
```

Fig.01: Install gcc and friends

Finally, install the kernel-package package too:

```
$ sudo apt-get install kernel-package
```

OR

```
$ sudo apt-get --no-install-recommends install kernel-
package
```

Sample outputs:

```
veryv@instance-1:~$ sudo apt-get --no-install-recommends install kernel-package
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  bc docbook-xml docbook-xsl gettext intltool-debian libcroco3 libglib2.0-0 libunistring0
  libxml2-utils po-debconf sgml-base sgml-data xml-core xmlto xsltproc
Suggested packages:
  docbook docbook-dsssl docbook-defguide dbtoepub docbook-xsl-doc-html docbook-xsl-doc-pdf
  docbook-xsl-doc-text docbook-xsl-doc docbook-xsl-saxon fop libsaxon-java libxalan2-java
  libxslthl-java xalan gettext-doc linux-source libmail-box-perl sgml-base-doc perlsgml
  w3-recs opensp debhelper w3m lynx-cur links xmltex
Recommended packages:
  autopoint libasprintf-dev libgettextpo-dev docbook-utils uboot-mkimage kernel-common
  libglib2.0-data shared-mime-info xdg-user-dirs libmail-sendmail-perl libpaper-utils zip
The following NEW packages will be installed:
  bc docbook-xml docbook-xsl gettext intltool-debian kernel-package libcroco3 libglib2.0-0
  libunistring0 libxml2-utils po-debconf sgml-base sgml-data xml-core xmlto xsltproc
0 upgraded, 16 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,918 kB of archives.
After this operation, 34.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Fig.02: Install utility for building Linux kernel

Download the Linux kernel source code

Type the following wget command:

```
$ wget https://cdn.kernel.org/pub/linux/kernel
/v4.x/linux-4.5.tar.xz
```

Sample outputs:

```
vivek@dev.cyberciti.biz:~$ wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.2.tar.xz
--2015-09-09 10:25:29--  https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.2.tar.xz
Resolving www.kernel.org (www.kernel.org)... 198.145.20.140, 149.20.4.69, 199.204.44.194, ...
Connecting to www.kernel.org (www.kernel.org)|198.145.20.140|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85507784 (82M) [application/x-xz]
Saving to: 'linux-4.2.tar.xz'

linux-4.2.tar.xz    100%[===================>]  81.55M  36.6MB/s   in 2.2s

2015-09-09 10:25:32 (36.6 MB/s) - 'linux-4.2.tar.xz' saved [85507784/85507784]

vivek@dev.cyberciti.biz:~$
```

Fig.03: Use the wget to grab the latest source code from kernel.org

Use GnuPG to verify kernel signatures:

```
$ unxz linux-4.5.tar.xz
$ gpg --verify linux-4.5.tar.sign
```

Sample outputs:

```
gpg: assuming signed data in `linux-4.5.tar'
gpg: Signature made Mon 14 Mar 2016 04:35:48 AM UTC using R
gpg: Can't check signature: public key not found
```

Get the public key from the PGP keyserver in order to verify the signature i.e. RSA key ID **00411886** (from the above outputs):

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys
00411886
```

Sample outputs:

```
gpg: requesting key 00411886 from hkp server keys.gnupg.net
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 00411886: public key "Linus Torvalds " imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
```

Now verify again:

```
$ gpg --verify linux-4.5.tar.sign
```

Sample outputs:

```
gpg: assuming signed data in `linux-4.5.tar'
gpg: Signature made Mon 14 Mar 2016 04:35:48 AM UTC using R
gpg: Good signature from "Linus Torvalds "
gpg: WARNING: This key is not certified with a trusted sign
gpg:          There is no indication that the signature bel
Primary key fingerprint: ABAF 11C6 5A29 70B1 30AB  E3C4 79B
```

If you do not "**BAD signature**" output from "`gpg --verify`" command, untar the Linux kernel tar ball using the tar command enter:

```
$ tar xvf linux-4.5.tar
$ ls
$ cd linux-4.5
$ ls
```

Sample outputs

```
linux-4.5/
linux-4.5/.get_maintainer.ignore
linux-4.5/.gitignore
linux-4.5/.mailmap
linux-4.5/COPYING
linux-4.5/CREDITS
linux-4.5/Documentation/
linux-4.5/Documentation/00-INDEX
linux-4.5/Documentation/ABI/
linux-4.5/Documentation/ABI/README
linux-4.5/Documentation/ABI/obsolete/
linux-4.5/Documentation/ABI/obsolete/proc-sys-vm-nr_pdflush
....
```

```
...
linux-4.5/virt/kvm/async_pf.h
linux-4.5/virt/kvm/coalesced_mmio.c
linux-4.5/virt/kvm/coalesced_mmio.h
linux-4.5/virt/kvm/eventfd.c
linux-4.5/virt/kvm/irqchip.c
linux-4.5/virt/kvm/kvm_main.c
linux-4.5/virt/kvm/vfio.c
linux-4.5/virt/kvm/vfio.h
linux-4.5/virt/lib/
linux-4.5/virt/lib/Kconfig
linux-4.5/virt/lib/Makefile
linux-4.5/virt/lib/irqbypass.c

linux-4.5

arch     crypto     include  kernel       net          secur
block    Documentation  init     lib          README
certs    drivers    ipc  MAINTAINERS  REPORTING-BUGS  tools
COPYING  firmware   Kbuild   Makefile     samples         u
CREDITS  fs     Kconfig  mm       scripts         virt
```

Configure the Linux kernel

First, copy your existing Linux kernel config file

```
$ cd linux-4.5
$ cp /boot/config-$(uname -r) .config
```

To configure the kernel, run:

```
$ make menuconfig
```

Sample outputs:

```
veryv@instance-1:~/linux-4.2$ make menuconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/mconf.o
  SHIPPED scripts/kconfig/zconf.tab.c
  SHIPPED scripts/kconfig/zconf.lex.c
  SHIPPED scripts/kconfig/zconf.hash.c
  HOSTCC  scripts/kconfig/zconf.tab.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
```

Fig.04: Starting menuconfig

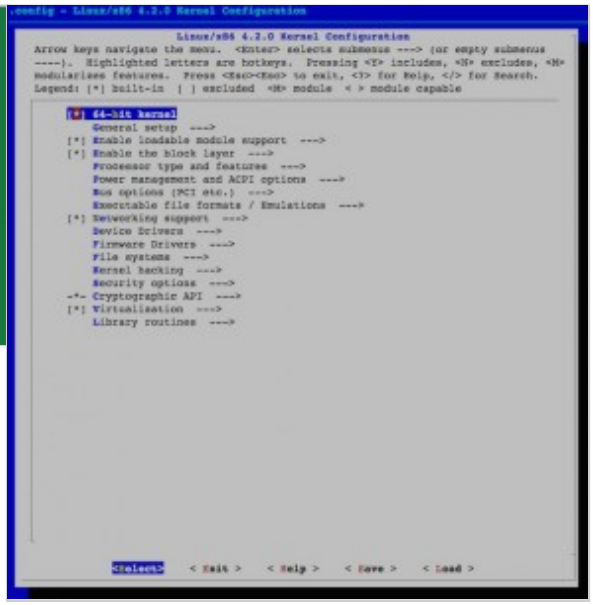

Fig.05: Select Linux kernel config options and drivers to build

Gallery 01: Click to enlarge

WARNING: It is easy to remove support for a device driver or option and end up with a broken kernel. For example, if the ext4 driver is removed from the kernel configuration file, a system may not boot. When in doubt, just leave support in the kernel.

Make sure you save the changes before exit from menuconfig.

Compile the Linux kernel

You need to clean the source tree and reset the kernel-package parameters, type:

```
$ make-kpkg clean
```

Sample outputs:

```
veryv@instance-1:~/linux-4.2$ make-kpkg clean
exec make kpkg_version=13.014+nmu1 -f /usr/share/kernel-package/ruleset/minimal.mk clean
====== making target minimal_clean [new prereqs: ]======
This is kernel package version 13.014+nmu1.
test ! -f .config || cp -pf .config config.precious
test ! -e stamp-building || rm -f stamp-building
test ! -f Makefile || \
        make    ARCH=x86_64 distclean
make[1]: Entering directory '/home/veryv/linux-4.2'
  CLEAN   scripts/basic
  CLEAN   scripts/kconfig
  CLEAN   include/config include/generated
  CLEAN   .config .config.old
make[1]: Leaving directory '/home/veryv/linux-4.2'
test ! -f config.precious || mv -f config.precious .config
rm -f modules/modversions.h modules/ksyms.ver scripts/cramfs/cramfsck scripts/cramfs/mkcramf
s
```

Fig.06: Run make-kpkg command

Now, you can compile the kernel, run:

```
$ fakeroot make-kpkg --initrd --revision=1.0.NAS
kernel_image kernel_headers
```

To speed up the compile process pass the `-j` option (-j 16 means you are using all 16 cores to compile the Linux kernel):

```
$ fakeroot make-kpkg --initrd --revision=1.0.NAS
kernel_image kernel_headers -j 16
```

Sample outputs:

```
veryv@instance-1:~/linux-4.2$ fakeroot make-kpkg --initrd --revision=1.0.NAS kernel_image kernel_headers
exec make kpkg_version=13.014+nmu1 -f /usr/share/kernel-package/ruleset/minimal.mk debian DEBIAN_REVISION=1
.0.NAS  INITRD=YES
====== making target debian/stamp/conf/minimal_debian [new prereqs: ]======
This is kernel package version 13.014+nmu1.
test -d debian              || mkdir debian
test ! -e stamp-building || rm -f stamp-building
install -p -m 755 /usr/share/kernel-package/rules debian/rules
for file in ChangeLog  Control  Control.bin86 config templates.in rules; do
     \
          cp -f  /usr/share/kernel-package/$file ./debian/;                          \
        done
cp: cannot stat '/usr/share/kernel-package/ChangeLog': No such file or directory
for dir  in Config docs examples ruleset scripts pkg po;  do                              \
         cp -af /usr/share/kernel-package/$dir  ./debian/;                          \
        done
test -f debian/control || sed         -e 's/=V/4.2.0/g'  \
              -e 's/=D/1.0.NAS/g'         -e 's/=A/amd64/g'  \
       -e 's/=SA//g'  \
       -e 's/=I//g'                                \
       -e 's/=CV/4.2/g'                            \
       -e 's/=M/Unknown Kernel Package Maintainer <unknown@unconfigured.in.etc.kernel-pkg.conf>/g'
   \
```

Fig.07: Start compiling the kernel

The fakeroot runs a command called make-kpkg in an environment wherein it appears to have root privileges for file manipulation. This is useful for allowing users to create archives (tar, ar, .deb etc.) with files in them with root permissions/ownership. The make-kpkg command build Debian/Ubuntu kernel packages from Linux kernel sources and options are:

- `--initrd` : Create an initrd image.
- `--revision=1.0.NAS` : Set custom revision for your kernel such as 1.0.NAS or -1.0-custom-kernel etc.
- `kernel_image` : This target produces a Debian package of the Linux kernel source image, and any modules configured in the kernel configuration file .config.
- `kernel_headers` : This target produces a Debian package of the Linux kernel header image.

Please note that kernel compilation may take quite a while, depending on the power of your machine. On my shared 4 CORE CPU and 4GB ram it took 60 mins to build the Linux kernel. In the end you should see something as follows on screen:

```
test ! -e debian/control~ || rm -f debian/control~
dpkg-gencontrol -isp -DArchitecture=amd64 -plinux-headers-4
                                          -P/root/linux-4.5
dpkg-gencontrol: warning: -isp is deprecated; it is without
create_md5sums_fn () { cd $1 ; find . -type f ! -regex './D
chown -R root:root                  /root/linux-4.5/debian/
chmod -R og=rX                      /root/linux-4.5/debian/
dpkg --build                        /root/linux-4.5/debian/
dpkg-deb: building package `linux-headers-4.5.0' in `../lin
cp -pf debian/control.dist          debian/control
```

```
make[2]: Leaving directory '/root/linux-4.5'
make[1]: Leaving directory '/root/linux-4.5'
```

Verify kernel deb files:

```
$ ls  ../*.deb
../linux-headers-4.5.0_1.0.NAS_amd64.deb  ../linux-image-4.
```

Installing a custom kernel

Type the following dpkg command to install a custom kernel on your system:

```
$ sudo dpkg -i linux-headers-4.5.0_1.0.NAS_amd64.deb
$ sudo dpkg -i linux-image-4.5.0_1.0.NAS_amd64.deb
```

Sample outputs:

```
Selecting previously unselected package linux-image-4.5.0.
(Reading database ... 251501 files and directories currently installed.)
Preparing to unpack linux-image-4.5.0_1.0.NAS_amd64.deb ...
Examining /etc/kernel/preinst.d/
Done.
Unpacking linux-image-4.5.0 (1.0.NAS) ...
Setting up linux-image-4.5.0 (1.0.NAS) ...

 Hmm. There is a symbolic link /lib/modules/4.5.0/build
 However, I can not read it: No such file or directory
 Therefore, I am deleting /lib/modules/4.5.0/build


 Hmm. The package shipped with a symbolic link /lib/modules/4.5.0/source
 However, I can not read the target: No such file or directory
 Therefore, I am deleting /lib/modules/4.5.0/source

Running depmod.
Examining /etc/kernel/postinst.d.
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.5.0 /boot/vmlinu
run-parts: executing /etc/kernel/postinst.d/dkms 4.5.0 /boot/vmlinuz-4.5.0
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.5.0 /boot/vmlinuz
update-initramfs: Generating /boot/initrd.img-4.5.0
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.5.0 /boot/vmlinuz-
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.5.0
Found initrd image: /boot/initrd.img-4.5.0
```

```
Found linux image: /boot/vmlinuz-4.4.4
Found initrd image: /boot/initrd.img-4.4.4
Found linux image: /boot/vmlinuz-4.4.2
Found initrd image: /boot/initrd.img-4.4.2
Found linux image: /boot/vmlinuz-4.4.0
Found initrd image: /boot/initrd.img-4.4.0
Found linux image: /boot/vmlinuz-4.3.0
Found initrd image: /boot/initrd.img-4.3.0
Found linux image: /boot/vmlinuz-4.2.5
Found initrd image: /boot/initrd.img-4.2.5
Found linux image: /boot/vmlinuz-4.2.0
Found initrd image: /boot/initrd.img-4.2.0
done
```

Reboot the box/server/laptop

Type the following command:

```
$ reboot
```

OR

```
$ shutdown -r now
```

Verify that everything is working

Type the following command to verify your new kernel and everything is working fine:

```
$ uname -a
$ uname -r
$ uname -mrs
$ dmesg | more
$ dmesg | egrep -i --color 'error|critical|failed'
```

Sample outputs:

```
Linux nas02.nixcraft.net.in 4.5.0 #1 SMP Mon Mar 21 05:25:2
```

And, there you have it, the Linux kernel version 4.4.4 installed and working correctly.

See also

- The Linux kernel project site.
- Man pages – apt-get(8), make-kpkg(1), fakeroot(1)

Share this tutorial on:

Twitter Facebook Google+ Download PDF version Found an error/typo on this page?

About the author: Vivek Gite is a seasoned sysadmin and a trainer for the Linux/Unix & shell scripting. Follow him on **Twitter**. OR **read more like this**:

- How to: Linux delete or remove kernel
- Debian / Ubuntu Linux Delete Old Kernel Images Command
- Steps for compiling Linux kernel on SMP Itanium IA64 system
- How To Patch and Protect Linux Kernel Zero Day Vulnerability CVE-2016-0728…
- Linux rebuild the initial ramdisk image
- CentOS Linux install kernel headers to build vmware / 3rd party modules
- How do I upgrade the kernel in Linux without compiling from source code?
- Debian Linux boot disk creation
- Debian / Ubuntu Linux Install Kernel Headers Package
- Ubuntu Linux LTS: Force Kernel Package Updates Which Have Been Kept Back

{ 46 comments… add one }

AppleFanBoy September 13, 2015, 8:37 pm

New kernels will automatically come with newer updates/distro, but the learning experience of compiling kernel is pricelessï»¿.

REPLY LINK

Richard September 15, 2015, 4:07 am

Thanks for the tutorial… Took a couple of hours to build on my machine.
But it worked. Well sort of. I get missing or strange characters in text. Doesn't matter if an icon label, in a terminal, LibreOffice, Firefox, etc. I went back to original kernel and all is well again. Using LUbuntu 3.19 kernel 64 bit.

REPLY LINK

kxmx September 16, 2015, 8:18 am

Is there any difference between "raw" kernel and "ubuntu" kernel?
Ubuntu compilated kernel 4.2 is marked like unstable, but kernel.org says 4.2 is stable. What does it mean?

REPLY LINK

> **Vivek Gite** September 16, 2015, 8:45 am
> 4.2 is stable. Ubuntu built packages are behind the schedule and most likely built when 4.2 was in beta. Also, some propitiatory NVIDIA/ATI & others driver may not work on the latest version 4.2. YMMV.
>
> REPLY LINK

cage September 22, 2015, 9:24 am

Hi Vivik,

Thanks for the tutorial. I followed the same steps you mentioned but I found one file (abi) is missing on my macbook (parallels + ubuntu 14.04) , here is /boot folder. Do you know which step is wrong ?

```
parallels@ubuntu:/boot$ ls -al
total 115040
drwxr-xr-x  3 root root     4096 Sep 22 16:42 .
drwxr-xr-x 23 root root     4096 Sep 21 19:25 ..
-rw-r--r--  1 root root  1158016 May  3  2014 abi-3.13.0-24-g
-rw-r--r--  1 root root  1162712 Aug 14  2014 abi-3.13.0-34-g
-rw-r--r--  1 root root  1165204 Aug 15 07:07 abi-3.13.0-63-g
-rw-r--r--  1 root root   165510 May  3  2014 config-3.13.0-2
-rw-r--r--  1 root root   165611 Aug 14  2014 config-3.13.0-3
-rw-r--r--  1 root root   165763 Aug 15 07:07 config-3.13.0-6
-rw-r--r--  1 root root   182704 Sep 22 16:35 config-4.2.0-he
-rw-r--r--  1 root root   182704 Sep 22 15:23 config-4.2.0-he
drwxr-xr-x  5 root root     4096 Sep 22 16:42 grub
-rw-r--r--  1 root root 19096859 Aug 18  2014 initrd.img-3.13
-rw-r--r--  1 root root 19145542 Sep  8 10:29 initrd.img-3.13
-rw-r--r--  1 root root 19219306 Sep 18 00:24 initrd.img-3.13
-rw-r--r--  1 root root 18069958 Sep 22 16:42 initrd.img-4.2
-rw-r--r--  1 root root   176500 Mar 12  2014 memtest86+.bin
-rw-r--r--  1 root root   178176 Mar 12  2014 memtest86+.elf
-rw-r--r--  1 root root   178680 Mar 12  2014 memtest86+_mul
-rw-------  1 root root  3372643 May  3  2014 System.map-3.13
-rw-------  1 root root  3381262 Aug 14  2014 System.map-3.13
-rw-------  1 root root  3392068 Aug 15 07:07 System.map-3.13
-rw-r--r--  1 root root  3663717 Sep 22 16:35 System.map-4.2
-rw-------  1 root root  5776416 May  3  2014 vmlinuz-3.13.0
-rw-------  1 root root  5797728 Aug 14  2014 vmlinuz-3.13.0
-rw-------  1 root root  5821152 Aug 15 07:07 vmlinuz-3.13.0
-rw-r--r--  1 root root  6118160 Sep 22 16:35 vmlinuz-4.2.0-l
```

REPLY LINK

Walt September 23, 2015, 2:49 pm

If you want to only compile the loaded modules with the current kernel

use ‘make localmodconfig’. This will make the compile time shorter and include all of the currently loaded modules of you current kernel.

REPLY LINK

M October 8, 2015, 1:11 pm

Great information.

Thanks!

REPLY LINK

Luigi November 12, 2015, 10:09 am

What does this——> ubuntu@mate:~$ dmesg | egrep -i –color ‘error|critical|failed’
[4.718735] systemd[1]: Failed to insert module ‘kdbus’: Function not implemented
[9.308305] EXT4-fs (sdc1): re-mounted. Opts: errors=remount-ro
[22.411061] vboxdrv: module verification failed: signature and/or required key missing – tainting kernel <————————mean?

REPLY LINK

Andy December 31, 2015, 4:00 pm

Try adding -j for a significant speed increase:
i.e.

```
fakeroot make-kpkg --initrd --revision=1.0.NAS -j 16
```

This sped up the process to about 10 minutes on my 8 core system. By default

only a single core is used.

REPLY LINK

frank paulsen January 11, 2016, 12:04 pm

newer kernels directly support “make deb-pkg” instead of using”make-kpkg”

REPLY LINK

wayno January 12, 2016, 10:58 am

great tute, though with the unpacking “Installing custom kernel” .deb part, had to do image first then headers.. naturally.. thanx…

REPLY LINK

Green January 17, 2016, 11:50 am

I have problem after :
“fakeroot make-kpkg –initrd –revision=1.0.NAS kernel_image kernel_headers”

it’s showing:
“scripts/sign-file.c:23:30: fatal error: openssl/opensslv.h: No such file or directory
compilation terminated.
scripts/Makefile.host:91: recipe for target ‘scripts/sign-file’ failed
make[2]: *** [scripts/sign-file] Error 1
Makefile:545: recipe for target ‘scripts’ failed
make[1]: *** [scripts] Error 2
make[1]: Leaving directory ‘/home/x/linux-4.4’
debian/ruleset/targets/common.mk:295: recipe for target ‘debian/stamp/build/kernel’ failed

make: *** [debian/stamp/build/kernel] Error 2”

What I’m doing wrong?

Thanks for help

REPLY LINK

Vivek Gite January 17, 2016, 12:13 pm

Did you installed build-essential package? openssl.h means you need to install libssl-dev? Try

```
sudo apt-get install libssl-dev build-essential
ncurses-dev xz-utils kernel-package
```

Edited by Vivek Gite. Reason: typo.

REPLY LINK

Green January 17, 2016, 1:03 pm

Reading package lists… Done
Building dependency tree
Reading state information… Done
E: Unable to locate package curses-dev

REPLY LINK

Vivek Gite January 17, 2016, 2:20 pm

My bad. It should be `ncurses-dev` .

REPLY LINK

Green January 18, 2016, 9:34 am

Thanks for help

After I type :
ls ../*.deb

it' showing on red :
../linux-headers-4.4.0_1.0.NAS_amd64.deb
../linux-image-4.4.0_1.0.NAS_amd64.deb

sudo dpkg -i linux-headers-4.4.0_1.0.NAS_amd64.deb:
dpkg: error processing archive linux-headers-
4.4.0_1.0.NAS_amd64.deb (–install):
cannot access archive: No such file or directory
Errors were encountered while processing:
linux-headers-4.4.0_1.0.NAS_amd64.deb

REPLY LINK

Vivek Gite January 18, 2016, 7:19 pm

cd to .. and try:

```
sudo dpkg -i linux-headers-
4.4.0_1.0.NAS_amd64.deb
sudo dpkg -i linux-image-
4.4.0_1.0.NAS_amd64.deb
```

OR

```
sudo dpkg -i ../linux-headers-
4.4.0_1.0.NAS_amd64.deb
sudo dpkg -i ../linux-image-
4.4.0_1.0.NAS_amd64.deb
```

REPLY LINK

Green January 19, 2016, 9:34 am

I have enough :) I tried to run ubuntu on Yoga
900 – Wifi / touchscreen and track-pad doesn't work.

I manage to run Wifi but with rest I have huge problem (nothing works).

Anyway BIG BIG thanks for help and all replies.

REPLY LINK

lange June 30, 2016, 12:14 am

you need to compile the linux-image, and after the linux-header

REPLY LINK

Mikel March 21, 2016, 12:35 pm

Thaks so much Vivek Gite, worked for me!

REPLY LINK

Alex January 18, 2016, 4:22 pm

Your article needs updating. All of the required packages do not exist in the cache. If someone comes to this site hoping for a guide to properly compiling and installing a new Linux kernel – not having the packages prior to coming here – they will spend quite some time trying to track down a fistful of packages that no longer exist, have been updated and/or replaced.

Please fix this ASAP. Thanks. :)

REPLY LINK

Vivek Gite January 18, 2016, 6:26 pm

Can you provide your Linux distro name and other info like what packages are missing? I tested this on Debian 8 and Ubuntu Linux 14.04.3 LTS only. Everything worked perfectly.

REPLY LINK

Alex January 18, 2016, 8:09 pm

Debian 8.1

Checked /etc/apt/sources.list all required sources are in place. Searching apt-cache returns no entries for any of the listed packages.

REPLY LINK

Jur January 19, 2016, 8:59 am

```
user@PLUS:~$ dmesg | egrep -i --color
'error|critical|failed'
[ 7.474368] systemd[1]: Cannot add dependency job for
unit display-manager.service, ignoring: Unit display-
manager.service failed to load: No such file or
directory.
[ 7.474377] systemd[1]: Cannot add dependency job for
unit display-manager.service, ignoring: Unit display-
manager.service failed to load: No such file or
directory.
[ 10.513727] EXT4-fs (sda7): re-mounted. Opts:
errors=remount-ro
user@PLUS:~$
```

REPLY LINK

IT GUY January 19, 2016, 11:20 am

Much easier way to compile your kernel:

linux-kernel-utilities

REPLY LINK

TheRiddick July 2, 2016, 2:38 pm

Thanks for the thought but your install script causes dkms errors due to elaborate versioning. I get some extremely long build and version numbers which makes it useless. You need to allow for the user to enter EXPLICIT kernel versions and not do some crazy automate based on date/time etc……

REPLY LINK

TheRiddick July 2, 2016, 2:40 pm

I would suggest a prompt that shows the user WHAT you will be calling the kernel version and deb files, and allow the user to modify it at this point.

REPLY LINK

alimp5 January 28, 2016, 1:02 pm

tnx a lot:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

REPLY LINK

Luigi February 7, 2016, 4:36 am

After I select Linux kernel config options and drivers to build and I click on save and then exit, it says to “make” in the terminal but your tutorial says that I should “make-kpkg clean” so which one should I do first after I save and exit the menuconfig?

REPLY LINK

Luigi February 7, 2016, 4:40 am

End of the configuration.

> *** Execute ‘make’ to start the build or try ‘make help’. is what it says so what should I do next?
>
> REPLY LINK

nir February 18, 2016, 9:02 pm

this is my output anyone can help me?

root@debian:/home/habernir/linux-4.1.18# dpkg -i linux-headers-4.1.18_1.0.NAS_amd64.deb
(Reading database … 214910 files and directories currently installed.)
Preparing to unpack linux-headers-4.1.18_1.0.NAS_amd64.deb …
Unpacking linux-headers-4.1.18 (1.0.NAS) over (1.0.NAS) …
Setting up linux-headers-4.1.18 (1.0.NAS) …
Examining /etc/kernel/header_postinst.d.
run-parts: executing /etc/kernel/header_postinst.d/dkms 4.1.18 /boot/vmlinuz-4.1.18
Error! Your kernel headers for kernel 4.1.18 cannot be found.
Please install the linux-headers-4.1.18 package,
or use the –kernelsourcedir option to tell DKMS where it’s located
root@debian:/home/habernir/linux-4.1.18#

REPLY LINK

> **nir** February 18, 2016, 9:16 pm
> amd debian 8.3 was installed in uefi
>
> REPLY LINK

Mohan March 15, 2016, 3:50 am

Hi I am trying to build linux-socfpga-3.10-ltsi kernel from rocketboards

and facing the following error. Please help:
make[4]: *** No rule to make target `drivers/acpi/acpica/uteval.o', needed by `drivers/acpi/acpica/acpi.o'. Stop.
make[3]: *** [drivers/acpi/acpica] Error 2
make[2]: *** [drivers/acpi] Error 2
make[1]: *** [drivers] Error 2
make[1]: Leaving directory `/home/mohanubuntu/Mohan/linux-socfpga-socfpga-3.10-ltsi'
make: *** [debian/stamp/build/kernel] Error 2

REPLY LINK

Dawei Li March 15, 2016, 4:22 am

I ran into error when executing the following command:
$ fakeroot make-kpkg –initrd –revision=1.0.NAS kernel_image kernel_headers -j 4

Can anyone help me? The last lines of the following are the error information.

objcopy –add-gnu-debuglink=/home/daweili/linux-4.4.4/debian/linux-image-4.4.4/usr/lib/debug/lib/modules/4.4.4/kernel/virt/lib/irqbypass.ko /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib/modules/4.4.4/kernel/virt/lib/irqbypass.ko
rm -rf /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/usr/lib/debug
test ! -f tools/lguest/lguest || \
install -p -o root -g root -m 644 tools/lguest/lguest /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib/modules/4.4.4/lguest
test ! -f /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib/modules/4.4.4/lguest || \
chmod 755 /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib/modules/4.4.4/lguest
test ! -e /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib/modules/4.4.4/source || \
mv /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib/modules/4.4.4/source

```
./debian/source-link
test ! -e /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib/modules/4.4.4/build
|| \
mv /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib/modules/4.4.4/build
./debian/build-link
test ! -e ./debian/source-link || \
mv ./debian/source-link /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib
/modules/4.4.4/source
test ! -e ./debian/build-link || \
mv ./debian/build-link /home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib
/modules/4.4.4/build
/sbin/depmod -q -FSystem.map -b /home/daweili/linux-4.4.4/debian/linux-
image-4.4.4 4.4.4;
restore_upstream_debianization
test ! -f scripts/package/builddeb.kpkg-dist || mv -f scripts/package
/builddeb.kpkg-dist scripts/package/builddeb
test ! -f scripts/package/Makefile.kpkg-dist || mv -f scripts/package
/Makefile.kpkg-dist scripts/package/Makefile
/usr/bin/make INSTALL_MOD_PATH=/home/daweili/linux-4.4.4/debian/linux-
image-4.4.4 \
INSTALL_FW_PATH=/home/daweili/linux-4.4.4/debian/linux-image-4.4.4/lib
/firmware/4.4.4 \
INSTALL_PATH=/home/daweili/linux-4.4.4/debian/linux-image-4.4.4//boot install
make[2]: Entering directory `/home/daweili/linux-4.4.4′
scripts/kconfig/conf –silentoldconfig Kconfig
make[2]: Leaving directory `/home/daweili/linux-4.4.4′
make[2]: Entering directory `/home/daweili/linux-4.4.4′
sh ./arch/x86/boot/install.sh 4.4.4 arch/x86/boot/bzImage \
System.map “/home/daweili/linux-4.4.4/debian/linux-image-4.4.4//boot”
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.4.4 /home/daweili
/linux-4.4.4/debian/linux-image-4.4.4//boot/vmlinuz-4.4.4
/etc/kernel/postinst.d/apt-auto-removal: 84: /etc/kernel/postinst.d/apt-auto-
removal: cannot create /etc/apt/apt.conf.d//01autoremove-kernels.dpkg-new:
Permission denied
run-parts: /etc/kernel/postinst.d/apt-auto-removal exited with return code 2
```

make[3]: *** [install] Error 1
make[2]: *** [install] Error 2
make[2]: Leaving directory `/home/daweili/linux-4.4.4′
make[1]: *** [debian/stamp/install/linux-image-4.4.4] Error 2
make[1]: Leaving directory `/home/daweili/linux-4.4.4′
make: *** [kernel_image] Error 2

REPLY LINK

Mikel March 21, 2016, 12:54 pm

Hi, is it normal to get Warnings like this during compilation?
e.g. ” drivers/isdn/hardware/eicon/capifunc.c:1094:1: warning: the frame size of 1152 bytes is larger than 1024 bytes [-Wframe-larger-than=] “

REPLY LINK

> **Vivek Gite** March 21, 2016, 1:40 pm
> Yes.
>
> REPLY LINK

rick March 27, 2016, 2:53 pm

Hi,
This worked for me. A further question: the needed hardware config selections / modules were not compiled into the kernel, so I did this a second time with a different revision, enabling the relevant options in menuconfig, but grub (nor uname, proc/release) does not show different revisions, only the kernel version, so not sure which I am using… is there a way to tell? The hardware (touchpad) does not work- it is not identified by the system either. Is it possible to compile it separately as a module?

Anyway, the guide worked as described, even if it did not accomplish what I wanted, so thanks for that.

REPLY LINK

FEDERICO March 29, 2016, 9:48 am

Hello

Great article, but i am having problems with this

Verify kernel deb files:

No debs files are there

fede@fede-Lenovo-YOGA-900-13ISK:~/linux-4.5$ ls ../*.deb
ls: cannot access ../*.deb: No such file or directory
fede@fede-Lenovo-YOGA-900-13ISK:~/linux-4.5$

REPLY LINK

FEDERICO March 29, 2016, 9:51 am

oval: cannot create /etc/apt/apt.conf.d//01autoremove-
kernels.dpkg-new: Permission denied
run-parts: /etc/kernel/postinst.d/apt-auto-removal exited with return code 2
make[3]: *** [install] Error 1
make[2]: *** [install] Error 2
make[2]: Leaving directory `/home/fede/linux-4.5′
make[1]: *** [debian/stamp/install/linux-image-4.5.0] Error 2
make[1]: Leaving directory `/home/fede/linux-4.5′
make: *** [kernel_image] Error 2

REPLY LINK

SeaDog April 1, 2016, 12:50 pm

I use Asus ux305 on the which I installed Ubuntu 15.10 with default kernetl 4.2. This kernel not supports Intel M Core the same did not work my TouchPad. Thanks to this guide I installed kernel 4.5 and all works fine. Thanks alot.

REPLY LINK

> **CodeSlayer** April 27, 2016, 6:06 am
> It worked on my ASUS UX305 which is also having Intel Core M processor, and it worked perfectly
>
> REPLY LINK

degomos April 29, 2016, 2:53 am

Hi i have debian installed on my asus zenbook ux32vd, and i have this error and failure do you know how to fix this:
[1.470054] ata1.00: READ LOG DMA EXT failed, trying unqueued
[1.470115] ata1.00: failed to get NCQ Send/Recv Log Emask 0x1
[1.471198] ata1.00: failed to get NCQ Send/Recv Log Emask 0x1
[11.204433] ata2: COMRESET failed (errno=-16)
[21.260609] ata2: COMRESET failed (errno=-16)
[56.293242] ata2: COMRESET failed (errno=-16)
[61.309330] ata2: COMRESET failed (errno=-16)
[61.309411] ata2: reset failed, giving up
[61.662470] EXT4-fs (sda1): re-mounted. Opts: errors=remount-ro
[61.715798] Error: Driver ‘pcspkr’ is already registered, aborting…
[64.039613] acpi_call: Cannot get handle: Error: AE_NOT_FOUND
[64.050550] acpi_call: Cannot get handle: Error: AE_NOT_FOUND

REPLY LINK

james April 30, 2016, 10:24 am

which I need to select in Fig.05, which is rightest for install kernel. you not clear, I donâ€™t understand anything.

REPLY LINK

Fahad June 4, 2016, 11:35 am

Amazing guide, used this to compile a perfect kernel 4.6.1.
Bookmarked!!!!

REPLY LINK

Akash July 3, 2016, 11:57 am

Thanks much for this very informative article.
I have been using the following 5 commands to build the 4.x kernel with a particular
.config file taken from http://kernel.ubuntu.com/~kernel-ppa/mainline/
Steps on Development machine
Copy the config file inside the kernel source directory as .config
yes ” | make oldconfig
make -j4 deb-pkg LOCALVERSION=-my_kernel
Steps on target machine
sudo dpkg -i linux-headers-x.xxâ€¦_amd64.deb
sudo dpkg -i linux-image-x.xxâ€¦_amd64.deb

After this even if I have to do a minor modification in kernel source (like add some new printk statements) then also I had to repeat the above steps which takes lot of time (as everything is done from scratch, nothing is reused from previous

compilation)
How can I expedite the subsequent compilation of kernel ?
On debian package installation, I see the following files inside the /boot directory on the target machine
/boot/vmlinuz*KERNEL-VERSION*
/boot/initrd*KERNEL-VERSION*
/boot/System-map*KERNEL-VERSION*
/boot/config-*KERNEL-VERSION*

Can I just replace the vmlinuz*KERNEL-VERSION* file with that of modified kernel & reboot ?
Please can somebody kindly tell me a shortcut.

For modules, I can use a shortcut, like for example for i915 module, I follow these 2 steps.
make DRM_MODULES=”i915″ -j4
cp drivers/gpu/drm/i915/i915.ko to /lib/modules/VERSION/kernel/drivers/gpu/drm/i915/ on the target.

REPLY LINK

tito September 2, 2016, 3:24 pm

It worked for me!
debian 3.16 -> debian 4.5 Wow

REPLY LINK

Security: Are you a robot or human?

Leave a Comment

Name

Email

Comment

Receive Email Notifications?

yes, replies to my comment | instantly

Or, you can subscribe without commenting.

Submit

You can use these HTML tags and attributes: <pre> <code> <a href="" title="">

Tagged with: Advanced

Next FAQ: Btrfs Filesystem /etc/fstab Entry To Mount It Automatically At Linux System Startup

Previous FAQ: How To Install ncurses Library on a Linux

To search, type and hit enter