

Kbuilder

kbuilder is a bash script that allows you to build one of the four officially supported kernels locally with your own optional patches and configuration, under a different package name.

For example, you could fetch the sources of the [linux-lts](https://archlinux.org/packages/?name=linux-lts) (<https://archlinux.org/packages/?name=linux-lts>) kernel, apply a patch that optimizes the kernel for intel processors, and install that kernel as **linux-lts-intel-optimized**.

Contents

- [Installation](#)
- [Building a kernel](#)
- [Build variables](#)
- [Hooks](#)
- [See also](#)

Installation

The following methods can be used to get kbuilder:

- **Install** the [kbuilder](https://aur.archlinux.org/packages/kbuilder/) (<https://aur.archlinux.org/packages/kbuilder/>)^{AUR} package.
- Install the latest development version from the [github repository](https://github.com/AnnikaV9/kbuilder) (<https://github.com/AnnikaV9/kbuilder>).

Building a kernel

To fetch the default vanilla [linux](https://archlinux.org/packages/?name=linux) (<https://archlinux.org/packages/?name=linux>) kernel sources and build the package with the default name **linux-kbuilder-custom**:

```
$ kbuilder build
```

Note: The build directory is `~/.kbuilder/cache`.

During the build process, kbuilder will inject a bash shell into the PKGBUILD. The shell will be run right before `build()` is called, allowing you to add any patches or change the kernel config interactively. The build will continue once you exit from the shell.

To install the built packages:

```
$ kbuilder install
```

Build variables

Although kbuilder does not have a configuration file, you can control the build process with [environment variables](#).

- `KBUILDER_SOURCE_PKG` should be set as 'linux', 'linux-lts', 'linux-zen' or 'linux-hardened'. kbuilder will fetch and build the kernel specified with this variable. Defaults to 'linux'. For more information about the official kernels, see [kernel](#).

Note: Patches that work on one kernel may not necessarily work on another.

- `KBUILDER_PKG_NAME_APPEND` can be set as any string. It will be appended to the end of the kernel version and the kernel package. Defaults to 'kbuilder-custom'.
- `KBUILDER_BUILD_JOBS` should be set as an integer. kbuilder will replace the `make all` line in the PKGBUILD with `make -jX all`, where X is the integer specified.

Example usage:

```
$ KBUILDER_BUILD_JOBS=4 KBUILDER_PKG_NAME_APPEND="my-custom-patch" KBUILDER_SOURCE_PKG="linux-zen" kbuilder build
```

Hooks

kbuilder looks for hooks in `/etc/kbuilder/hooks/`, and uses run-parts to execute them right before `build()` is called. The run-parts command is executed with the flag `--regex "kbuilder-hook-*`.

An example hook:

```
/etc/kbuilder/hooks/05-kbuilder-hook-cpu-optimize.sh

#!/bin/bash
#
# kbuilder hook for applying graysky2's kernel compiler patch:
# https://github.com/graysky2/kernel\_compiler\_patch
#

SOURCE="https://raw.githubusercontent.com/graysky2/kernel\_compiler\_patch/master"
PATCH="more-uarches-for-kernel-5.17+.patch"
HOOK_NAME="cpu-optimize-hook"

curl -o $PATCH $SOURCE/$PATCH || { echo "$HOOK_NAME failed when fetching $PATCH from $SOURCE" ; exit 1 ; }
patch -Np1 -i "$PATCH" || { echo "$HOOK_NAME failed when applying $PATCH" ; exit 1 ; }
```

To create a hook, [create](#) and make [executable](#) a file, with a name following the `xx-kbuilder-hook-hookname` form: they will be run in order according to the xx value.

See also

- [Kernel](#)
- [The Linux Kernel Documentation \(https://docs.kernel.org/\)](https://docs.kernel.org/)
- [Applying Patches To The Linux Kernel \(https://docs.kernel.org/process/applying-patches.html\)](https://docs.kernel.org/process/applying-patches.html)

Retrieved from "<https://wiki.archlinux.org/index.php?title=Kbuilder&oldid=733998>"

This page was last edited on 25 June 2022, at 12:18.

Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.

- [Privacy policy](#)
- [About ArchWiki](#)
- [Disclaimers](#)