Kbuild

Output files

modules.order

This file records the order in which modules appear in Makefiles. This is used by modprobe to deterministically resolve aliases that match multiple modules.

modules.builtin

This file lists all modules that are built into the kernel. This is used by modprobe to not fail when trying to load something builtin.

modules.builtin.modinfo

This file contains modinfo from all modules that are built into the kernel. Unlike modinfo of a separate module, all fields are prefixed with module name.

Environment variables

KCPPFLAGS

Additional options to pass when preprocessing. The preprocessing options will be used in all cases where kbuild does preprocessing including building C files and assembler files.

KAFLAGS

Additional options to the assembler (for built-in and modules).

AFLAGS_MODULE

Additional assembler options for modules.

AFLAGS_KERNEL

Additional assembler options for built-in.

KCFLAGS

Additional options to the C compiler (for built-in and modules).

CFLAGS_KERNEL

Additional options for $(CC) when used to compile code that is compiled as built-in.

CFLAGS_MODULE

Additional module specific options to use for $(CC).

LDFLAGS_MODULE

Additional options used for $(LD) when linking modules.

HOSTCFLAGS

Additional flags to be passed to $(HOSTCC) when building host programs.

HOSTCXXFLAGS

Additional flags to be passed to $(HOSTCXX) when building host programs.

HOSTLDFLAGS

Additional flags to be passed when linking host programs.

HOSTLDLIBS

Additional libraries to link against when building host programs.

USERCFLAGS

Additional options used for $(CC) when compiling userprogs.

USERLDFLAGS

Additional options used for $(LD) when linking userprogs. userprogs are linked with CC, so $(USERLDFLAGS) should include "-Wl," prefix as applicable.

KBUILD_KCONFIG

Set the top-level Kconfig file to the value of this environment variable. The default name is "Kconfig".

KBUILD_VERBOSE

Set the kbuild verbosity. Can be assigned same values as "V=...".

See make help for the full list.

Setting "V=..." takes precedence over KBUILD_VERBOSE.

KBUILD_EXTMOD

Set the directory to look for the kernel source when building external modules.

Setting "M=..." takes precedence over KBUILD_EXTMOD.

KBUILD_OUTPUT

Specify the output directory when building the kernel.

The output directory can also be specified using "O=...".

Setting "O=..." takes precedence over KBUILD_OUTPUT.

KBUILD_EXTRA_WARN

Specify the extra build checks. The same value can be assigned by passing W=... from the command line.

See *make help* for the list of the supported values.

Setting "W=..." takes precedence over KBUILD_EXTRA_WARN.

KBUILD_DEBARCH

For the deb-pkg target, allows overriding the normal heuristics deployed by deb-pkg. Normally deb-pkg attempts to guess the right architecture based on the UTS_MACHINE variable, and on some architectures also the kernel config. The value of KBUILD_DEBARCH is assumed (not checked) to be a valid Debian architecture.

ARCH

Set ARCH to the architecture to be built.

In most cases the name of the architecture is the same as the directory name found in the arch/ directory.

But some architectures such as x86 and sparc have aliases.

- x86: i386 for 32 bit, x86_64 for 64 bit
- sh: sh for 32 bit, sh64 for 64 bit
- sparc: sparc32 for 32 bit, sparc64 for 64 bit

CROSS_COMPILE

Specify an optional fixed part of the binutils filename. CROSS_COMPILE can be a part of the filename or the full path.

CROSS_COMPILE is also used for ccache in some setups.

CF

Additional options for sparse.

CF is often used on the command-line like this:

```
make CF=-Wbitwise C=2
```

INSTALL_PATH

INSTALL_PATH specifies where to place the updated kernel and system map images. Default is /boot, but you can set it to other values.

INSTALLKERNEL

Install script called when using "make install". The default name is "installkernel".

The script will be called with the following arguments:

- $1 - kernel version
- $2 - kernel image file
- $3 - kernel map file
- $4 - default install path (use root directory if blank)

The implementation of "make install" is architecture specific and it may differ from the above.

INSTALLKERNEL is provided to enable the possibility to specify a custom installer when cross compiling a kernel.

MODLIB

Specify where to install modules. The default value is:

```
$(INSTALL_MOD_PATH)/lib/modules/$(KERNELRELEASE)
```

The value can be overridden in which case the default value is ignored.

INSTALL_MOD_PATH

INSTALL_MOD_PATH specifies a prefix to MODLIB for module directory relocations required by build roots. This is not defined in the makefile but the argument can be passed to make if needed.

INSTALL_MOD_STRIP

INSTALL_MOD_STRIP, if defined, will cause modules to be stripped after they are installed. If INSTALL_MOD_STRIP is '1', then the default option --strip-debug will be used. Otherwise, INSTALL_MOD_STRIP value will be used as the options to the strip command.

INSTALL_HDR_PATH

INSTALL_HDR_PATH specifies where to install user space headers when executing "make headers_*".

The default value is:

```
$(objtree)/usr
```

$(objtree) is the directory where output files are saved. The output directory is often set using "O=..." on the commandline.

The value can be overridden in which case the default value is ignored.

KBUILD_ABS_SRCTREE

Kbuild uses a relative path to point to the tree when possible. For instance, when building in the source tree, the source tree path is '.'

Setting this flag requests Kbuild to use absolute path to the source tree. There are some useful cases to do so, like when generating tag files with absolute path entries etc.

KBUILD_SIGN_PIN

This variable allows a passphrase or PIN to be passed to the sign-file utility when signing kernel modules, if the private key requires such.

KBUILD_MODPOST_WARN

KBUILD_MODPOST_WARN can be set to avoid errors in case of undefined symbols in the final module linking stage. It changes such errors into warnings.

KBUILD_MODPOST_NOFINAL

KBUILD_MODPOST_NOFINAL can be set to skip the final link of modules. This is solely useful to speed up test compiles.

KBUILD_EXTRA_SYMBOLS

For modules that use symbols from other modules. See more details in modules.rst.

ALLSOURCE_ARCHS

For tags/TAGS/cscope targets, you can specify more than one arch to be included in the databases, separated by blank space. E.g.:

```
$ make ALLSOURCE_ARCHS="x86 mips arm" tags
```

To get all available archs you can also specify all. E.g.:

```
$ make ALLSOURCE_ARCHS=all tags
```

KBUILD_BUILD_TIMESTAMP

Setting this to a date string overrides the timestamp used in the UTS_VERSION definition (uname -v in the running kernel). The value has to be a string that can be passed to date -d. The default value is the output of the date command at one point during build.

KBUILD_BUILD_USER, KBUILD_BUILD_HOST

These two variables allow to override the user@host string displayed during boot and in /proc/version. The default value is the output of the commands whoami and host, respectively.

LLVM

If this variable is set to 1, Kbuild will use Clang and LLVM utilities instead of GCC and GNU binutils to build the kernel.