

[PATCH 1/2] kbuild: add deb-pkg target

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-06/4783.html>

From: Sam Ravnborg (sam_at_ravnborg.org)

Date: 06/20/04

Date: Sun, 20 Jun 2004 23:21:24 +0200

To: Andrew Morton <akpm@osdl.org>, Linus Torvalds <torvalds@osdl.org>, linux-kernel@vger.kernel.org

```
# This is a BitKeeper generated diff -Nru style patch.
#
# ChangeSet
# 2004/06/19 23:41:15+02:00 sam@mars.ravnborg.org
# kbuild: Add new package format: deb-pkg
#
# To prepare kbuild for more kernel packaging formats move all packaging
# support to scripts/package.
# In top-level Makefile introduce generic support for all package
# formats using target names *-pkg. Included the old rpm target for backward
# compatibility.
#
# A new variable KBUILD_IMAGE is used to specify what kernel image
# will be part of the final package, and is to be set by the arch specific
# makefile. KBUILD_IMAGE may be overridden from command line or environment.
# KBUILD_IMAGE will see wider usage later, mainly when installing
# kernel images.
# Introducing KBUILD_IMAGE allowed arch specific details to be deleted from the mkspec
# and builddeb scripts.
#
# While in the process added the deb packet format. Script is
# From: Wichert Akkerman <wichert@wiggy.net>
#
# To create a RPM packet use 'make rpm-pkg'.
# To create a deb packet use 'make deb-pkg'.
# Both targets are included in 'make help'
#
# Signed-off-by: Sam Ravnborg <sam@ravnborg.org>
#
# scripts/package/builddeb
# 2004/06/19 23:41:00+02:00 sam@mars.ravnborg.org +79 -0
#
# scripts/package/builddeb
# 2004/06/19 23:41:00+02:00 sam@mars.ravnborg.org +0 -0
# BitKeeper file /home/sam/bk/kbuild/scripts/package/builddeb
#
```

Linux-Kernel: [PATCH 1/2] kbuild: add deb-pkg target

```
# scripts/package/Makefile
# 2004/06/19 23:40:59+02:00 sam@mars.ravnborg.org +71 -0
#
# scripts/package/mkspec
# 2004/06/19 23:40:59+02:00 sam@mars.ravnborg.org +6 -15
# KBUILD_IMAGE tell what image is being built.
# Therefore remove hardcoded architecture specific info
#
# scripts/package/Makefile
# 2004/06/19 23:40:59+02:00 sam@mars.ravnborg.org +0 -0
# BitKeeper file /home/sam/bk/kbuild/scripts/package/Makefile
#
# scripts/Makefile
# 2004/06/19 23:40:59+02:00 sam@mars.ravnborg.org +1 -1
# Add package/ to dirs being visited during make mrproper
#
# arch/i386/Makefile
# 2004/06/19 23:40:59+02:00 sam@mars.ravnborg.org +7 -6
# Set KBUILD_IMAGE to kernel image being build.
# Since BOOTIMAGE and KBUILD_IMAGE contained the same info kill
# local version of BOOTIMAGE
#
# Makefile
# 2004/06/19 23:40:59+02:00 sam@mars.ravnborg.org +25 -37
# Move rpm target to scripts/package
# Added support for more generic package handling
#
# scripts/package/mkspec
# 2004/06/19 14:53:10+02:00 sam@mars.ravnborg.org +0 -0
# Rename: scripts/mkspec -> scripts/package/mkspec
#
diff -Nru a/Makefile b/Makefile
--- a/Makefile 2004-06-20 23:05:13 +02:00
+++ b/Makefile 2004-06-20 23:05:13 +02:00
@@ -290,8 +290,6 @@
 OBJCOPY = $(CROSS_COMPILE)objcopy
 OBJDUMP = $(CROSS_COMPILE)objdump
 AWK = awk
-RPM := $(shell if [ -x "/usr/bin/rpmbuild" ]; then echo rpmbuild; \
- else echo rpm; fi)
 GENKSYMS = scripts/genksyms/genksyms
 DEPMOD = /sbin/depmod
 KALLSYMS = scripts/kallsyms
@@ -409,13 +407,6 @@

scripts_basic: include/linux/autoconf.h

-
-# That's our default target when none is given on the command line
-# Note that 'modules' will be added as a prerequisite as well,
-# in the CONFIG_MODULES part below
```

```

-
-all: vmlinux
-
# Objects we will link into vmlinux / subdirs we need to visit
init-y := init/
drivers-y := drivers/ sound/
@@ -449,6 +440,19 @@

include $(srctree)/arch/$(ARCH)/Makefile

+# Default kernel image to build when no specific target is given.
+# KBUILD_IMAGE may be overruled on the commandline or
+# set in the environment
+# Also any assignments in arch/$(ARCH)/Makefile take precedence over
+# this default value
+export KBUILD_IMAGE ?= vmlinux
+
+# The all: target is the default when no target is given on the
+# command line.
+# This allow a user to issue only 'make' to build a kernel including modules
+# Defaults vmlinux but it is usually overridden in the arch makefile
+all: vmlinux
+
+ifdef CONFIG_CC_OPTIMIZE_FOR_SIZE
+CFLAGS += -Os
+else
@@ -795,7 +799,7 @@

# Directories & files removed with 'make clean'
CLEAN_DIRS += $(MODVERDIR)
-CLEAN_FILES += vmlinux System.map kernel.spec \
+CLEAN_FILES += vmlinux System.map \
    .tmp_kallsyms* .tmp_version .tmp_vmlinux*

# Directories & files removed with 'make mrproper'
@@ -848,37 +852,19 @@
    -o -name '%*' -o -name '*.cmd' -o -name 'core' ) \
    -type f -print | xargs rm -f

-# RPM target
-# -----
-
-.PHONY: rpm

-# Remove hyphens since they have special meaning in RPM filenames
-KERNELPATH=kernel-$(subst -,,$(KERNELRELEASE))
+# Packaging of the kernel to various formats
+# -----
+# rpm target kept for backward compatibility
+package-dir := $(srctree)/scripts/package

```

Linux-Kernel: [PATCH 1/2] kbuild: add deb-pkg target

```

-# If you do a make spec before packing the tarball you can rpm -ta it
+.PHONY: %-pkg rpm

-spec:
- $(CONFIG_SHELL) $(srctree)/scripts/mkspec > $(objtree)/kernel.spec
-
-# a) Build a tar ball
-# b) generate an rpm from it
-# c) and pack the result
-# - Use /. to avoid tar packing just the symlink
-
-rpm: clean spec
- set -e; \
- cd .. ; \
- ln -sf $(srctree) $(KERNELPATH) ; \
- tar -cvz $(RCS_TAR_IGNORE) -f $(KERNELPATH).tar.gz $(KERNELPATH)/. ; \
- rm $(KERNELPATH)
-
- set -e; \
- $(CONFIG_SHELL) $(srctree)/scripts/mkversion > $(objtree)/.tmp_version; \
- mv -f $(objtree)/.tmp_version $(objtree)/.version;
+%pkg: FORCE
+ $(Q)$(MAKE) -f $(package-dir)/Makefile $@
+rpm: FORCE
+ $(Q)$(MAKE) -f $(package-dir)/Makefile $@

- $(RPM) --target $(UTS_MACHINE) -ta ../$(KERNELPATH).tar.gz
- rm ../$(KERNELPATH).tar.gz

# Brief documentation of the typical targets used
# -----
@@ -905,6 +891,8 @@
    @echo ' tags/TAGS - Generate tags file for editors'
    @echo ' cscope - Generate cscope index'
    @echo ' checkstack - Generate a list of stack hogs'
+ @echo 'Kernel packaging:'
+ @$(MAKE) -f $(package-dir)/Makefile help
    @echo "
    @echo 'Documentation targets:'
    @$(MAKE) -f $(srctree)/Documentation/DocBook/Makefile dochelp
diff -Nru a/arch/i386/Makefile b/arch/i386/Makefile
--- a/arch/i386/Makefile 2004-06-20 23:05:13 +02:00
+++ b/arch/i386/Makefile 2004-06-20 23:05:13 +02:00
@@ -121,22 +121,23 @@

all: bzImage

-BOOTIMAGE=arch/i386/boot/bzImage
-zImage zImage: BOOTIMAGE=arch/i386/boot/zImage
+# KBUILD_IMAGE specify target image being built
+ KBUILD_IMAGE := $(boot)/bzImage
```

[PATCH 1/2] kbuild: add deb-pkg target

Linux-Kernel: [PATCH 1/2] kbuild: add deb-pkg target

```
+zImage zImage zdisk: KBUILD_IMAGE := arch/i386/boot/zImage
```

```
zImage bzImage: vmlinux
```

```
- $(Q)$MAKE $(build)=$(boot) $(BOOTIMAGE)
+ $(Q)$MAKE $(build)=$(boot) $(KBUILD_IMAGE)
```

```
compressed: zImage
```

```
zImage bzImage: vmlinux
```

```
- $(Q)$MAKE $(build)=$(boot) BOOTIMAGE=$(BOOTIMAGE) zImage
+ $(Q)$MAKE $(build)=$(boot) BOOTIMAGE=$(KBUILD_IMAGE) zImage
```

```
zImage bzImage: vmlinux
```

```
- $(Q)$MAKE $(build)=$(boot) BOOTIMAGE=$(BOOTIMAGE) zImage
+ $(Q)$MAKE $(build)=$(boot) BOOTIMAGE=$(KBUILD_IMAGE) zImage
```

```
install fdimage fdimage144 fdimage288: vmlinux
```

```
- $(Q)$MAKE $(build)=$(boot) BOOTIMAGE=$(BOOTIMAGE) $@
+ $(Q)$MAKE $(build)=$(boot) BOOTIMAGE=$(KBUILD_IMAGE) $@
```

```
prepare: include/asm-$(ARCH)/asm_offsets.h
```

```
CLEAN_FILES += include/asm-$(ARCH)/asm_offsets.h
```

```
diff -Nru a/scripts/Makefile b/scripts/Makefile
```

```
--- a/scripts/Makefile 2004-06-20 23:05:13 +02:00
```

```
+++ b/scripts/Makefile 2004-06-20 23:05:13 +02:00
```

```
@@ -13,7 +13,7 @@
```

```
subdir-$(CONFIG_MODVERSIONS) += genksyms
```

```
# Let clean descend into subdirs
```

```
-subdir- += basic lxdialog kconfig
```

```
+subdir- += basic lxdialog kconfig package
```

```
# dependencies on generated files need to be listed explicitly
```

```
diff -Nru a/scripts/mkspec b/scripts/mkspec
```

```
--- a/scripts/mkspec 2004-06-20 23:05:13 +02:00
```

```
+++ /dev/null Wed Dec 31 16:00:00 196900
```

```
@@ -1,72 +0,0 @@
```

```
#!/bin/sh
```

```
##
```

```
## Output a simple RPM spec file that uses no fancy features requiring
```

```
## RPM v4. This is intended to work with any RPM distro.
```

```
##
```

```
## The only gothic bit here is redefining install_post to avoid
```

```
## stripping the symbols from files in the kernel which we want
```

```
##
```

```
## Patched for non-x86 by Opencon (L) 2002 <opencon@rio.skydome.net>
```

```
##
```

```
## That's the voodoo to see if it's a x86.
```

```
-ISX86=`echo ${ARCH:=`arch`} | grep -ie i.86`
```

```
-if [ ! -z $ISX86 ]; then
```

[PATCH 1/2] kbuild: add deb-pkg target

Linux-Kernel: [PATCH 1/2] kbuild: add deb-pkg target

```

- PC=1
-else
- PC=0
-fi
-# starting to output the spec
-if [ "`grep CONFIG_DRM=y .config | cut -f2 -d\"=\" = \"y\" ` ]; then
- PROVIDES=kernel-drm
-fi
-
-PROVIDES=\"$PROVIDES kernel-$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION\"
-
-echo \"Name: kernel\"
-echo \"Summary: The Linux Kernel\"
-echo \"Version: \"$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION | sed -e \"s/-//g\"
-# we need to determine the NEXT version number so that uname and
-# rpm -q will agree
-echo \"Release: ` . $srctree/scripts/mkversion`\"
-echo \"License: GPL\"
-echo \"Group: System Environment/Kernel\"
-echo \"Vendor: The Linux Community\"
-echo \"URL: http://www.kernel.org\"
-echo -n \"Source: kernel-$VERSION.$PATCHLEVEL.$SUBLEVEL\"
-echo \"$EXTRAVERSION.tar.gz\" | sed -e \"s/-//g\"
-echo \"BuildRoot: /var/tmp/% {name}-% {PACKAGE_VERSION}-root\"
-echo \"Provides: $PROVIDES\"
-echo \"%define __spec_install_post /usr/lib/rpm/brp-compress || :\"
-echo \"%define debug_package % {nil}\"
-echo \"\"
-echo \"%description\"
-echo \"The Linux Kernel, the operating system core itself\"
-echo \"\"
-echo \"%prep\"
-echo \"%setup -q\"
-echo \"\"
-echo \"%build\"
-echo \"make clean all\"
-echo \"\"
-echo \"%install\"
-echo 'mkdir -p $RPM_BUILD_ROOT/boot $RPM_BUILD_ROOT/lib $RPM_BUILD_ROOT/lib/modules'
-echo 'INSTALL_MOD_PATH=$RPM_BUILD_ROOT make modules_install'
-# This is the first disagreement between i386 and most others
-if [ $PC = 1 ]; then
- echo 'cp arch/i386/boot/bzImage
-$RPM_BUILD_ROOT\"/boot/vmlinuz-$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION\"
-else
- echo 'cp vmlinux
-$RPM_BUILD_ROOT\"/boot/vmlinuz-$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION\"
-fi
-# Back on track
-echo 'cp System.map
-$RPM_BUILD_ROOT\"/boot/System.map-$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION\"

```

Linux-Kernel: [PATCH 1/2] kbuild: add deb-pkg target

```
-echo 'cp .config
$RPM_BUILD_ROOT"/boot/config-$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION"
-echo ""
-echo "%clean"
-echo '#echo -rf $RPM_BUILD_ROOT'
-echo ""
-echo "%files"
-echo '%defattr (-, root, root)'
-echo "%dir /lib/modules"
-echo "/lib/modules/$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION"
-echo "/boot/*"
-echo ""
diff -Nru a/scripts/package/Makefile b/scripts/package/Makefile
--- /dev/null Wed Dec 31 16:00:00 196900
+++ b/scripts/package/Makefile 2004-06-20 23:05:13 +02:00
@@ -0,0 +1,71 @@
+# Makefile for the different targets used to generate full packages of a kernel
+# It uses the generic clean infrastructure of kbuild
+
+# Ignore the following files/directories during tar operation
+TAR_IGNORE := --exclude SCCS --exclude BitKeeper --exclude .svn --exclude CVS
+
+
+# RPM target
+# -----
+# The rpm target generates two rpm files:
+# /usr/src/packages/SRPMS/kernel-2.6.7rc2-1.src.rpm
+# /usr/src/packages/RPMS/i386/kernel-2.6.7rc2-1.<arch>.rpm
+# The src.rpm files includes all source for the kernel being built
+# The <arch>.rpm includes kernel configuration, modules etc.
+#
+# Process to create the rpm files
+# a) clean the kernel
+# b) Generate .spec file
+# c) Build a tar ball, using symlink to make kernel version
+# first entry in the path
+# d) and pack the result to a tar.gz file
+# e) generate the rpm files, based on kernel.spec
+# - Use /. to avoid tar packing just the symlink
+
+# Do we have rpmbuild, otherwise fall back to the older rpm
+RPM := $(shell if [ -x "/usr/bin/rpmbuild" ]; then echo rpmbuild; \
+ else echo rpm; fi)
+
+# Remove hyphens since they have special meaning in RPM filenames
+KERNELPATH := kernel-$(subst -,,$(KERNELRELEASE))
+MKSPEC := $(srctree)/scripts/package/mkspec
+PREV := set -e; cd ..;
+
+.PHONY: rpm-pkg rpm
```

Linux-Kernel: [PATCH 1/2] kbuild: add deb-pkg target

```
+$ (objtree)/kernel.spec: $(MKSPEC)
+ $(CONFIG_SHELL) $(MKSPEC) > $@
+
+rpm-pkg rpm: $(objtree)/kernel.spec
+ $(MAKE) clean
+ $(PREV) ln -sf $(srctree) $(KERNELPATH)
+ $(PREV) tar -cz $(RCS_TAR_IGNORE) -f $(KERNELPATH).tar.gz $(KERNELPATH)/
+ $(PREV) rm $(KERNELPATH)
+
+ set -e; \
+ $(CONFIG_SHELL) $(srctree)/scripts/mkversion > $(objtree)/.tmp_version
+ set -e; \
+ mv -f $(objtree)/.tmp_version $(objtree)/.version
+
+ $(RPM) --target $(UTS_MACHINE) -ta ..$(KERNELPATH).tar.gz
+ rm ..$(KERNELPATH).tar.gz
+
+clean-rule += rm -f $(objtree)/kernel.spec
+
+# Deb target
+# -----
+#
+.PHONY: deb-pkg
+deb-pkg:
+ $(MAKE)
+ $(CONFIG_SHELL) $(srctree)/scripts/package/builddeb
+
+clean-rule += && rm -rf $(objtree)/debian/
+
+
+# Help text displayed when executing 'make help'
+# -----
+help:
+ @echo ' rpm-pkg - Build the kernel as an RPM package'
+ @echo ' deb-pkg - Build the kernel as an deb package'
+
diff -Nru a/scripts/package/builddeb b/scripts/package/builddeb
--- /dev/null Wed Dec 31 16:00:00 196900
+++ b/scripts/package/builddeb 2004-06-20 23:05:13 +02:00
@@ -0,0 +1,79 @@
+#!/bin/sh
+#
+# builddeb 1.2
+# Copyright 2003 Wichert Akkerman <wichert@wiggy.net>
+#
+# Simple script to generate a deb package for a Linux kernel. All the
+# complexity of what to do with a kernel after it is installed or removed
+# is left to other scripts and packages: they can install scripts in the
+# /etc/kernel/{pre,post}{inst,rm}.d/ directories that will be called on
+# package install and removal.
+
```

Linux–Kernel: [PATCH 1/2] kbuild: add deb–pkg target

```
+set -e
+
+# Some variables and settings used throughout the script
+version="$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION"
+tmpdir="$objtree/debian/tmp"
+
+# Setup the directory structure
+rm -rf "$tmpdir"
+mkdir -p "$tmpdir/DEBIAN" "$tmpdir/lib" "$tmpdir/boot"
+
+# Build and install the kernel
+cp System.map "$tmpdir/boot/System.map-$version"
+cp .config "$tmpdir/boot/config-$version"
+cp $KBUILD_IMAGE "$tmpdir/boot/vmlinuz-$version"
+
+if grep -q '^CONFIG_MODULES=y' .config ; then
+INSTALL_MOD_PATH="$tmpdir" make modules_install
+fi
+
+# Install the maintainer scripts
+for script in postinst postrm preinst prerm ; do
+mkdir -p "$tmpdir/etc/kernel/$script.d"
+cat <<EOF > "$tmpdir/DEBIAN/$script"
+#!/bin/sh
+
+set -e
+
+test -d /etc/kernel/$script.d && run-parts --arg="$version" /etc/kernel/$script.d
+exit 0
+EOF
+chmod 755 "$tmpdir/DEBIAN/$script"
+done
+
+name="Kernel Compiler <$(id -nu)@$(hostname -f)>"
+# Generate a simple changelog template
+cat <<EOF > debian/changelog
+linux ($version) unstable; urgency=low
+
+ * A standard release
+
+ -- $name $(date -R)
+EOF
+
+# Generate a control file
+cat <<EOF > debian/control
+Source: linux
+Section: base
+Priority: optional
+Maintainer: $name
+Standards-Version: 3.6.1
+
```

Linux-Kernel: [PATCH 1/2] kbuild: add deb-pkg target

```
+Package: linux-$version
+Architecture: any
+Description: Linux kernel, version $version
+ This package contains the Linux kernel, modules and corresponding other
+ files version $version.
+EOF
+
+# Fix some ownership and permissions
+chown -R root:root "$tmpdir"
+chmod -R go-w "$tmpdir"
+
+# Perform the final magic
+dpkg-gencontrol -isp
+dpkg --build "$tmpdir" ..
+
+exit 0
+
diff -Nru a/scripts/package/mkspec b/scripts/package/mkspec
--- /dev/null Wed Dec 31 16:00:00 196900
+++ b/scripts/package/mkspec 2004-06-20 23:05:13 +02:00
@@ -0,0 +1,63 @@
+#!/bin/sh
+#
+# Output a simple RPM spec file that uses no fancy features requiring
+# RPM v4. This is intended to work with any RPM distro.
+#
+# The only gothic bit here is redefining install_post to avoid
+# stripping the symbols from files in the kernel which we want
+#
+# Patched for non-x86 by Opencon (L) 2002 <opencon@rio.skydome.net>
+#
+
+# starting to output the spec
+if [ "`grep CONFIG_DRM=y .config | cut -f2 -d\"=\" = \"y\" ]"; then
+ PROVIDES=kernel-drm
+fi
+
+PROVIDES="$PROVIDES kernel-$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION"
+
+echo "Name: kernel"
+echo "Summary: The Linux Kernel"
+echo "Version: "$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION | sed -e "s/--/g"
+# we need to determine the NEXT version number so that uname and
+# rpm -q will agree
+echo "Release: `.` $srcdir/scripts/mkversion`"
+echo "License: GPL"
+echo "Group: System Environment/Kernel"
+echo "Vendor: The Linux Community"
+echo "URL: http://www.kernel.org"
+echo -n "Source: kernel-$VERSION.$PATCHLEVEL.$SUBLEVEL"
+echo "$EXTRAVERSION.tar.gz" | sed -e "s/--/g"
```

Linux-Kernel: [PATCH 1/2] kbuild: add deb-pkg target

```
+echo "BuildRoot: /var/tmp/%{name}-%{PACKAGE_VERSION}-root"
+echo "Provides: $PROVIDES"
+echo "%define __spec_install_post /usr/lib/rpm/brp-compress || :"
+echo "%define debug_package %{nil}"
+echo ""
+echo "%description"
+echo "The Linux Kernel, the operating system core itself"
+echo ""
+echo "%prep"
+echo "%setup -q"
+echo ""
+echo "%build"
+echo "make clean && make"
+echo ""
+echo "%install"
+echo 'mkdir -p $RPM_BUILD_ROOT/boot $RPM_BUILD_ROOT/lib $RPM_BUILD_ROOT/lib/modules'
+
+echo 'INSTALL_MOD_PATH=$RPM_BUILD_ROOT make modules_install'
+echo 'cp $KBUILD_IMAGE
$RPM_BUILD_ROOT"/boot/vmlinuz-$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION"
+
+echo 'cp System.map
$RPM_BUILD_ROOT"/boot/System.map-$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION"
+
+echo 'cp .config
$RPM_BUILD_ROOT"/boot/config-$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION"
+echo ""
+echo "%clean"
+echo '#echo -rf $RPM_BUILD_ROOT'
+echo ""
+echo "%files"
+echo '%defattr (-, root, root)'
+echo "%dir /lib/modules"
+echo "/lib/modules/$VERSION.$PATCHLEVEL.$SUBLEVEL$EXTRAVERSION"
+echo "/boot/*"
+echo ""
-
```

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>