

[3/5][POWERPC] boot: use more Kbuild rules

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-07/msg00297.html>

- *From:* Milton Miller <miltonm@xxxxxxx>
 - *Date:* Sat, 1 Jul 2006 18:46:44 -0500 (CDT)
-

Switch to `if_changed_dep` for `bootcc`, `cmd` for known rebuilds without deps, and `if_changed` for other commands. Add `FORCE` for all `if_changed` rule uses. Make sure targets includes all generated files.

Move clean-files to targets for `if_changed` files.

Rename `OBJCOPYFLAGS` to `OBJCOPY_SEC_FLAGS` so we can use `$(objcopy)` rule.

Rename `zliblinuxheader` to `linuxheader`

Change from explicit dependency of main and `zlib` files and `zlib` headers to generate all headers before compiling the source. `fixdep` will give us the exact headers, but not on a clean compile.

Note: the `obj-sec .c` files are created/touched when the content file changes.

Signed-off-by: Milton Miller <miltonm@xxxxxxx>

Index: kernel/arch/powerpc/boot/Makefile

```
=====
--- kernel.orig/arch/powerpc/boot/Makefile 2006-07-01 00:03:41.940189322 -0500
+++ kernel/arch/powerpc/boot/Makefile 2006-07-01 01:33:55.937370129 -0500
@@ -25,16 +25,13 @@ HOSTCC := gcc
BOOTCFLAGS := $(HOSTCFLAGS) -fno-builtin -nostdinc -isystem \
$(shell $(CROSS32CC) -print-file-name=include) -fPIC
BOOTAFLAGS := -D__ASSEMBLY__ $(BOOTCFLAGS) -traditional -nostdinc
- OBJCOPYFLAGS := contents,alloc,load,readonly,data
+ OBJCOPY_SEC_FLAGS := contents,alloc,load,readonly,data
OBJCOPY_COFF_ARGS := -O aixcoff-rs6000 --set-start 0x500000
OBJCOPY_MIB_ARGS := -O aixcoff-rs6000 -R .stab -R .stabstr -R .comment

zlib := infblock.c infcodes.c inffast.c inflate.c inftrees.c infutil.c
zlibheader := infblock.h infcodes.h inffast.h inftrees.h infutil.h
-zliblinuxheader := zlib.h zconf.h zutil.h
-
-$(addprefix $(obj)/,$(zlib) main.o): $(addprefix $(obj)/,$(zliblinuxheader)) $(addprefix
$(obj)/,$(zlibheader))
-#$(addprefix $(obj)/,main.o): $(addprefix $(obj)/,zlib.h)
+linuxheader := zlib.h zconf.h zutil.h
```

[3/5][POWERPC] boot: use more Kbuild rules

```
src-boot := crt0.S string.S prom.c stdio.c main.c div64.S
src-boot += $(zlib)
@@ -48,20 +45,23 @@ quiet_cmd_copy_zlib = COPY $@

quiet_cmd_copy_zlibheader = COPY $@
cmd_copy_zlibheader = sed "s@<linux^\([^\>]\|+\).*@\"1\"@" $< > $@
+
# stddef.h for NULL
-quiet_cmd_copy_zliblinuxheader = COPY $@
-cmd_copy_zliblinuxheader = sed
"s@<linux/string.h>@\"string.h\"@";s@<linux/kernel.h>@<stddef.h>@;s@<linux^\([^\>]\|+\).*@\"1\"@" $<
> $@
+quiet_cmd_copy_linuxheader = COPY $@
+cmd_copy_linuxheader = sed
"s@<linux/string.h>@\"string.h\"@";s@<linux/kernel.h>@<stddef.h>@;s@<linux^\([^\>]\|+\).*@\"1\"@" $<
> $@

-$(addprefix $(obj)/,$(zlib)): $(obj)/%: $(srctree)/lib/zlib_inflate/%
-$call cmd,copy_zlib
+$(addprefix $(obj)/,$(zlib)): $(obj)/%: $(srctree)/lib/zlib_inflate/% FORCE
+$call if_changed,copy_zlib

-$(addprefix $(obj)/,$(zlibheader)): $(obj)/%: $(srctree)/lib/zlib_inflate/%
-$call cmd,copy_zlibheader
+$(addprefix $(obj)/,$(zlibheader)): $(obj)/%: $(srctree)/lib/zlib_inflate/% \
+ FORCE
+$call if_changed,copy_zlibheader

-$(addprefix $(obj)/,$(zliblinuxheader)): $(obj)/%: $(srctree)/include/linux/%
-$call cmd,copy_zliblinuxheader
+$(addprefix $(obj)/,$(linuxheader)): $(obj)/%: $(srctree)/include/linux/% FORCE
+$call if_changed,copy_linuxheader

-clean-files := $(zlib) $(zlibheader) $(zliblinuxheader)
+targets += $(zlib) $(zlibheader) $(linuxheader)
+targets += $(patsubst $(obj)/%,%, $(obj)-boot))

quiet_cmd_bootcc = BOOTCC $@
@@ -73,11 +73,16 @@ quiet_cmd_bootas = BOOTAS $@
quiet_cmd_bootld = BOOTLD $@
cmd_bootld = $(CROSS32LD) -T $(srctree)/$(src)/$(3) -o $@ $(2)

-$(patsubst %.c,%.o, $(filter %.c, $(src-boot))): %.o: %.c
+$(patsubst %.c,%.o, $(filter %.c, $(src-boot))): %.o: %.c FORCE
$(call if_changed_dep,bootcc)
-$(patsubst %.S,%.o, $(filter %.S, $(src-boot))): %.o: %.S
+
+$(patsubst %.S,%.o, $(filter %.S, $(src-boot))): %.o: %.S FORCE
$(call if_changed_dep,bootas)
```

[3/5][POWERPC] boot: use more Kbuild rules

```
+$ (obj)-boot): COPYHEADERS
+COPYHEADERS: $(addprefix $(obj)/,$(linuxheader) $(zlibheader))
+PHONY += COPYHEADERS
+
#-----
# ELF sections within the zImage bootloader/wrapper
#-----
@@ -104,18 +109,20 @@ quiet_cmd_ramdisk = RAMDISK $@
quiet_cmd_stripvm = STRIP $@
cmd_stripvm = $(STRIP) -s -R .comment $< -o $@

-vmlinux.strip: vmlinux
+vmlinux.strip: vmlinux FORCE
$(call if_changed,stripvm)
-$(obj)/vmlinux.initrd: vmlinux.strip $(obj)/addRamDisk $(obj)/ramdisk.image.gz
+$(obj)/vmlinux.initrd: vmlinux.strip $(obj)/addRamDisk \
+ $(obj)/ramdisk.image.gz FORCE
$(call if_changed,ramdisk)

quiet_cmd_addsection = ADDSEC $@
cmd_addsection = $(CROSS32OBJCOPY) $@ \
--add-section=.kernel:$(strip $(patsubst $(obj)/kernel-%.o,%,$@))=$(patsubst %.o,% .gz,$@) \
--set-section-flags=.kernel:$(strip $(patsubst $(obj)/kernel-%.o,%,$@))=$(OBJCOPYFLAGS)
+ --set-section-flags=.kernel:$(strip $(patsubst $(obj)/kernel-%.o,%,$@))=$(OBJCOPY_SEC_FLAGS)

quiet_cmd_addnote = ADDNOTE $@
- cmd_addnote = $(obj)/addnote $@
+ cmd_addnote = @cp -f $< $@ && \
+ $(obj)/addnote $@

quiet_cmd_gen-mibboot = GEN $@
cmd_gen-mibboot = $(OBJCOPY) $(OBJCOPY_MIB_ARGS) \
@@ -125,7 +132,19 @@ quiet_cmd_gencoff = COFF $@
cmd_gencoff = $(OBJCOPY) $(OBJCOPY_COFF_ARGS) $@ && \
$(obj)/hack-coff $@

-$(call gz-sec, $(required)): $(obj)/kernel-%.gz: %
+# a single rule so that we can do if_changed correctly
+define rule_ldcoff
+ $(call echo-cmd,bootld) $(cmd_bootld); \
+ $(call echo-cmd,gencoff) $(cmd_gencoff)
+endef
+
+# $(required) is built in $(objtree) not $(obj) so read cmd files manually
+cmd_files := $(wildcard $(foreach f,$(required),$(dir $f)).$(notdir $(f)).cmd)
+ifneq ($(cmd_files),)
+ include $(cmd_files)
+endif
+
+$(call gz-sec, $(required)): $(obj)/kernel-%.gz: % FORCE
```

[3/5][POWERPC] boot: use more Kbuild rules

```
$(call if_changed,gzip)
```

```
$(obj)/kernel-initrd.gz: $(obj)/ramdisk.image.gz  
@@ -135,16 +154,16 @@ $(call src-sec, $(required) $(initrd)):  
@touch $@
```

```
$(call obj-sec, $(required) $(initrd)): $(obj)/kernel-%.o: $(obj)/kernel-%.c  
- $(call if_changed_dep,bootcc)  
+ $(call cmd,bootcc)  
$(call cmd,addsection)
```

```
$(obj)/zImage.vmode $(obj)/zImage.coff: obj-boot += $(call obj-sec, $(required))  
-$(obj)/zImage.vmode: $(call obj-sec, $(required)) $(obj-boot) $(srctree)/$(src)/zImage.lds  
- $(call cmd,bootld,$(obj-boot),zImage.lds)  
+$(obj)/zImage.vmode: $(call obj-sec, $(required)) $(obj-boot) $(srctree)/$(src)/zImage.lds FORCE  
+ $(call if_changed,bootld,$(obj-boot),zImage.lds)
```

```
$(obj)/zImage.initrd.vmode $(obj)/zImage.initrd.coff: obj-boot += $(call obj-sec, $(required) $(initrd))  
-$(obj)/zImage.initrd.vmode: $(call obj-sec, $(required) $(initrd)) $(obj-boot) $(srctree)/$(src)/zImage.lds  
- $(call cmd,bootld,$(obj-boot),zImage.lds)  
+$(obj)/zImage.initrd.vmode: $(call obj-sec, $(required) $(initrd)) $(obj-boot) $(srctree)/$(src)/zImage.lds  
FORCE  
+ $(call if_changed,bootld,$(obj-boot),zImage.lds)
```

```
# For 32-bit powermacs, build the COFF and miboot images
```

```
# as well as the ELF images.
```

```
@@ -154,30 +173,29 @@ mibootimg-$(CONFIG_PPC_PMAC)-$(CONFIG_PP  
mibrdimg-$(CONFIG_PPC_PMAC)-$(CONFIG_PPC32) := $(obj)/miboot.initrd.image
```

```
$(obj)/zImage: $(obj)/zImage.vmode $(obj)/addnote $(coffimage-y-y) \  
- $(mibootimg-y-y)  
- @cp -f $< $@  
+ $(mibootimg-y-y) FORCE  
$(call if_changed,addnote)  
+ @true
```

```
$(obj)/zImage.initrd: $(obj)/zImage.initrd.vmode $(obj)/addnote \  
- $(coffrdimg-y-y) $(mibrdimg-y-y)  
- @cp -f $< $@  
+ $(coffrdimg-y-y) $(mibrdimg-y-y) FORCE  
$(call if_changed,addnote)  
+ @true
```

```
$(obj)/zImage.coff: $(call obj-sec, $(required)) $(obj-boot) \  
- $(srctree)/$(src)/zImage.coff.lds $(obj)/hack-coff  
- $(call cmd,bootld,$(obj-boot),zImage.coff.lds)  
- $(call cmd,gencoff)  
+ $(srctree)/$(src)/zImage.coff.lds $(obj)/hack-coff FORCE  
+ $(call if_changed_rule,ldcoff,$(obj-boot),zImage.coff.lds)
```

```
$(obj)/zImage.initrd.coff: $(call obj-sec, $(required) $(initrd)) $(obj-boot) \  

```

[3/5][POWERPC] boot: use more Kbuild rules

```
- $(src)/$(src)/zImage.coff.lds $(obj)/hack-coff
- $(call cmd,bootld,$(obj)-boot,zImage.coff.lds)
- $(call cmd,gencoff)
+ $(src)/$(src)/zImage.coff.lds $(obj)/hack-coff \
+ FORCE
+ $(call if_changed_rule,ldcoff,$(obj)-boot,zImage.coff.lds)

-$(obj)/miboot.image: $(obj)/dummy.o $(obj)/vmlinux.gz
- $(call cmd,gen-miboot,image)
+$(obj)/miboot.image: $(obj)/dummy.o $(obj)/vmlinux.gz FORCE
+ $(call if_changed,gen-miboot,image)

-$(obj)/miboot.initrd.image: $(obj)/miboot.image $(images)/ramdisk.image.gz
- $(call cmd,gen-miboot,initrd)
+$(obj)/miboot.initrd.image: $(obj)/miboot.image $(images)/ramdisk.image.gz FORCE
+ $(call if_changed,gen-miboot,initrd)

# -----
# build u-boot images
@@ -199,6 +217,7 @@ extra-y += vmlinux.bin vmlinux.gz

$(obj)/vmlinux.bin: vmlinux FORCE
$(call if_changed,objbin)
+ @true

$(obj)/vmlinux.gz: $(obj)/vmlinux.bin FORCE
$(call if_changed,mygzip)
-
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
More majordomo info at http://vger.kernel.org/majordomo-info.html
Please read the FAQ at http://www.tux.org/lkml/
```