

(1条消息)Linux内核---10.obj-y中将.c编译为.o流程 - wangcong02345的专栏 - CSDN博客

版权声明：本文为博主原创文章，遵循[CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 版权协议，转载请附上原文出处链接和本声明。

先去编译obj-y,以init下的obj-y为例：

obj-y=init/main.o init/version.o init/mounts.o init/noinitramfs.o init/calibrate.o

首先找到将.c编译为.o的规则：

点击(此处)折叠或打开

1. linux-2.6.30.4/scripts/Makefile.build
2. 226 # Built-in and composite module parts
3. 227 \$(obj)/%.o: \$(src)/%.c FORCE
4. 228 \$(call cmd,force_checksrc)
5. 229 \$(call if_changed_rule,cc_o_c)

将.c编译为.o要分两步

1. L228,如果KBUILD_CHECKSRC为空，不做任何操作;不为空，没看。
2. L229，编译的具体操作。

1.1 检查 L228 \$(call cmd,force_checksrc)

点击(此处)折叠或打开

1. linux-2.6.30.4/scripts/Kbuild.include
2. 154 # echo command.
3. 155 # Short version is used, if \$(quiet) equals `quiet_', otherwise full one.
4. 156 echo-cmd = \$(if \$(\$(quiet)cmd_\$(1)),\
5. 157 echo ' \$(call escsq,\$(\$(quiet)cmd_\$(1)))\$(echo-why)';)
6. 158
7. 159 # printing commands
8. 160 cmd = @\$ (echo-cmd) \$(cmd_\$(1))
9. cmd 定义在L160处，如果cmd_force_checksrc不为空，则把
cmd_force_checksrc这个命令打印出来，然后执行cmd_force_checksrc
10. 转到cmd_force_checksrc定义处:
- 11.
12. linux-2.6.30.4/scripts/Makefile.build
13. 98 # Linus' kernel sanity checking tool
14. 99 ifneq (\$(KBUILD_CHECKSRC),0)
15. 100 ifeq (\$(KBUILD_CHECKSRC),2)

```

16. 101 quiet_cmd_force_checksrc = CHECK $<
17. 102 cmd_force_checksrc = $(CHECK) $(CHECKFLAGS) $(c_flags) $<;
18. 103 else
19. 104 quiet_cmd_checksrc = CHECK $<
20. 105 cmd_checksrc = $(CHECK) $(CHECKFLAGS) $(c_flags) $<;
21. 106 endif
22. 107 endif
23. 由于此时 KBUILD_CHECKSRC=0,所以cmd_force_checksrc为空, L156
    处的if条件为false, echo不执行, L160处cmd_force_checksrc为空, 也不
    执行。

```

2. 下面开始执行L229 \$(call if_changed_rule,cc_o_c), 首先转到定义处:

点击(此处)折叠或打开

```

1. linux-2.6.30.4/scripts/Kbuild.include
2. 208 if_changed_rule = $(if $(strip $(any-prereq) $(arg-check) ), \
3. 209 @set -e; \
4. 210 $(rule_$(1)))
5. $(strip $(any-prereq) $(arg-check)),如果目标有更新则调用: rule_cc_o_c
6.
7. linux-2.6.30.4/scripts/Makefile.build
8. 215 define rule_cc_o_c
9. 216 $(call echo-cmd,checksrc) $(cmd_checksrc) \
10. 217 $(call echo-cmd,cc_o_c) $(cmd_cc_o_c); \
11. 218 $(cmd_modversions) \
12. 219 $(cmd_record_mcount) \
13. 220 scripts/basic/fixdep $(depfile) $@ '$(call make-cmd,cc_o_c)' > \
14. 221 $(dot-target).tmp; \
15. 222 rm -f $(depfile); \
16. 223 mv -f $(dot-target).tmp $(dot-target).cmd
17. 224 endef

```

rule_cc_o_c

2.1 echo-cmd && cmd_checksrc : 不执行

2.2 echo-cmd && cmd_cc_o_c :打印cmd_cc_o_c的内容, 并执行cmd_cc_o_c
将.c编译为.o

2.3 cmd_modversions : 不执行

2.4 cmd_record_mcount : 不执行

2.5 生成命令.cmd

下面逐条分析:

2.1-2.2 echo-cmd && cmd_checksrc 和 echo-cmd && cmd_cc_o_c

点击(此处)折叠或打开

1. linux-2.6.30.4/scripts/Kbuild.include
2. 154 # echo command.
3. 155 # Short version is used, if \$(quiet) equals `quiet_', otherwise full one.
4. 156 echo-cmd = \$(if \$(quiet)cmd_\$(1)),\
5. 157 echo ' \$(call escsq,\$(quiet)cmd_\$(1))\$(echo-why)';)
6. cmd_checkssrc为空，不执行
- 7.
8. linux-2.6.30.4/scripts/Makefile.build
9. 178 ifndef CONFIG_MODVERSIONS
10. 179 cmd_cc_o_c = \$(CC) \$(c_flags) -c -o \$@ \$<
- 11.
12. \$(call echo-cmd,cc_o_c)将执行的编译命令打印出来，\$(cmd_cc_o_c)执行具体的编译命令。这将会在命令行中看到一坨坨的打印。

2.3 cmd_modversions

点击(此处)折叠或打开

1. linux-2.6.30.4/scripts/Makefile.build
2. 178 ifndef CONFIG_MODVERSIONS
3. 179 cmd_cc_o_c = \$(CC) \$(c_flags) -c -o \$@ \$<
4. 180
5. 181 else
6. 194 cmd_cc_o_c = \$(CC) \$(c_flags) -c -o \$(@D)/tmp_\$(@F) \$<
7. 195 cmd_modversions = ...
8. 因为此处没有定义 CONFIG_MODVERSIONS，所以cmd_modversions为空。

2.4 cmd_record_mcount

点击(此处)折叠或打开

1. 208 ifdef CONFIG_FTRACE_MCOUNT_RECORD
2. 209 cmd_record_mcount = perl \$(srctree)/scripts/recordmcount.pl
"\$(ARCH)" \
3. 210 "\$(if \$(CONFIG_64BIT),64,32)" \
4. 211 "\$(OBJDUMP)" "\$(OBJCOPY)" "\$(CC)" "\$(LD)" "\$(NM)" "\$(RM)"
"\$(MV)" \
5. 212 "\$(if \$(part-of-module),1,0)" "\$@";
6. 213 endif
7. CONFIG_FTRACE_MCOUNT_RECORD 没有定义，所以为空

总结一下：

虽然写了这么多，但是有用处的只有两个echo-cmd && cmd_cc_o_c和 mv，打印cmd_cc_o_c的内容，并执行cmd_cc_o_c将.c编译为.o，把命令保存到.cmd文件中。假设让我自己写出来的Makefile可能也就是\$(CC) \$(c_flags)

-c -o \$@ \$<这么一句。

文章最后发布于: 2016-07-02 14:37:20