

Clean Infrastructure

clean 为什么可以称 Infrastructure？因为 Kbuild 系统的 clean 有三种层次的实现：

- i. make clean: Delete most generated files, Leave enough to build external modules
- ii. make mrproper: Delete the current configuration, and all generated files
- iii. make distclean: Remove editor backup files, patch leftover files and the like

clean, mrproper, distclean, 递进的进行 clean，所以称之为 infrastructure。相关的代码都定义在 top Makefile 中。但它们的定义在空间上跨越很大,不易厘清，用伪代码来展示一下：

```
ifeq ($(KBUILD_EXTMOD),)
    ifdef CONFIG_MODULES
    else
    endif #CONFIG_MODULES

    clean: bluhbluh
    mrproper: bluhbluh
    distclean: bluhbluh

else
endif # KBUILD_EXTMOD

clean: bluhbluh
```

clean 完成大部分的清理动作，mrproper 在 clean 的基础上继续清理，distclean 又在 mrproper 的基础上继续清理。一般情况下 mrproper 是你的清理首选。

clean

clean 的流程图：

```
clean<--|# 定义在 arch/x86/Makefile 中
|--archclean
|       $(Q)rm -rf $(objtree)/arch/i386
|       $(Q)rm -rf $(objtree)/arch/x86_64
|       $(Q)$(MAKE) $(clean)=$(boot)
|       $(Q)$(MAKE) $(clean)=arch/x86/tools
|       $(Q)$(MAKE) $(clean)=arch/x86/purgatory
|--vmlinuxclean
|       $(Q)$(CONFIG_SHELL) $(srctree)/scripts/link-vmlinux.sh clean
|       $(Q)$(if $(ARCH_POSTLINK), $(MAKE) -f $(ARCH_POSTLINK) clean)
|# clean-dirs := $(addprefix _clean_, . $(vmlinux-alldirs) Documentation samples)
|--$(clean-dirs)
|       $(Q)$(MAKE) $(clean)=$(patsubst _clean_%,%, $@)
|
$(call cmd,rmdirs)
$(call cmd,rmfiles)
@find $(if $(KBUILD_EXTMOD), $(KBUILD_EXTMOD), .) $(RCS_FIND_IGNORE) \
    \( -name '*.oas' -o -name '*.ko' -o -name '*.cmd' \
    -o -name '*.ko.*' \
    -o -name '*.dwo' \
    -o -name '*.su' \
    -o -name '*.d' -o -name '*.tmp' -o -name '*.mod.c' \
    -o -name '*.symtypes' -o -name 'modules.order' \
    -o -name modules.builtin -o -name '.tmp_*.o.*' \
    -o -name '*.c.[012]*.*' \
    -o -name '*.gcno' \) -type f -print | xargs rm -f
```

可以看出很多清理动作都和变量 "clean" 有关，定义在 scripts/Kbuild.include：

```
clean := -f $(srctree)/scripts/Makefile.clean obj
```

如果看懂了前面的两篇文章，会发现 Makefile.clean 的内容在结构上和其他 scripts/ 目录下的 Makefile 大同小异，递归式的进

入每一个子文件夹下清理，需要清理的内容都定义在 `__clean-files` 和 `__clean-dirs`
不过，通过 `Makefile.clean` 清理的文件一般都是最终目标的文件，也就是说，那些中间文件，比如 `.o` 不在其列，在 `clean` 的 recipe 中清理。

Target **clean** 的 recipe 的前两条的全景 code，一点点繁琐，但很容易理解：

```
# MODVERDIR 是 .tmp_versions 目录
CLEAN_DIRS += $(MODVERDIR)

# CLEAN_FILES 一般定义在 arch Makefile 中。x86 没有定义它
clean: rm-dirs := $(CLEAN_DIRS)
clean: rm-files := $(CLEAN_FILES)

cmd_rmdirs = rm -rf $(rm-dirs)
cmd_rmfiles = rm -f $(rm-files)

$(call cmd,rmdirs)
$(call cmd,rmfiles)
```

第三条 `find` 起始的那行应该就不用解释了。

mrproper

在 target **clean** 的基础上，`mrproper` 继续清理：

```
# Directories & files removed with 'make mrproper'
MRPROPER_DIRS += include/config usr/include include/generated \
                arch/*/include/generated .tmp_objdiff
MRPROPER_FILES += .config .config.old .version \
                  Module.symvers tags TAGS cscope* GPATH GTAGS GRTAGS GSYMS \
                  signing_key.pem signing_key.priv signing_key.x509 \
                  x509.genkey extra_certificates signing_key.x509.keyid \
                  signing_key.x509.signer vmlinux-gdb.py

mrproper: rm-dirs := $(wildcard $(MRPROPER_DIRS))
mrproper: rm-files := $(wildcard $(MRPROPER_FILES))
mrproper-dirs := $(addprefix _mrproper_,scripts)

$(mrproper-dirs):
    $(Q)$(MAKE) $(clean)=$(patsubst _mrproper_%,%, $@)

# 顾名思义，archmrproper 即使有，也是定义在 arch Makefile 中。x86 没有定义它
mrproper: clean archmrproper $(mrproper-dirs)
    $(call cmd,rmdirs)
    $(call cmd,rmfiles)
```

附上一条趣味故事，`mrproper` 名字的来历：

<https://0x657573.wordpress.com/2011/01/03/what-does-mrproper-in-make-mrproper-stand-for>
<http://www.neatorama.com/2007/09/15/the-many-names-of-mr-clean/>
<https://lists.gt.net/linux/kernel/178306>

简言之：`mrproper` 是 P&G 的一款洗洁剂产品。没想到吧:)

distclean

Target **distclean** 做的事情更简单了，直接看代码：

```
distclean: mrproper
    @find $(srcdir) $(RCS_FIND_IGNORE) \
        \( -name '*.orig' -o -name '*.rej' -o -name '*~' \
        -o -name '*.bak' -o -name '##*' -o -name '%*' \
        -o -name 'core' \) \
        -type f -print | xargs rm -f
```

仅在 `mrproper` 的基础上清理一堆杂物。

This is what so-called Clean Infrastructure.

参考：[5 Kbuild clean infrastructure](#) of Documentation/kbuild/makefiles.txt