Managing Projects with GNU Make

第三版
完全修订版

GNU Make

项目管理

O'REILLY®
東南大學出版社

Robert Mecklenburg 著
*O'Reilly Taiwan*公司 编译

整个程序由100个源程序文件构成，分别是1SB.c，2SB.c，3SB.c，……100SB.c

gcc –o SB 1SB.c 2SB.c 3SB.c 4SB.c 5SB.c 6SB.c ……

我只修改了1SB.c，难道必须要重新编译2SB.c，3SB.c，……100SB.c?

gcc –c 1SB.c

gcc –o SB 1SB.o 2SB.o 3SB.o 4SB.o 5SB.o 6SB.o ……

可执行文件SB，是由1SB.o，2SB.o，……100SB.o连接而成的，

所以SB依赖于1SB.o，2SB.o，……100SB.o

1SB.o是由1SB.c编译而成的，

所以1SB.o依赖于1SB.c

2SB.o是由2SB.c编译而成的，

所以2SB.o依赖于2SB.c

.

.

.

100SB.o是由100SB.c编译而成的，

所以100SB.o依赖于100SB.c

SB:1SB.o 2SB.o 3SB.o 4SB.o 5SB.o … … 100SB.o

gcc –o SB 1SB.o 2SB.o 3SB.o 4SB.o 5SB.o … … 100SB.o

1SB.o:1SB.c

gcc –c 1SB.c

2SB.o:2SB.c

gcc –c 2SB.c

.

.

.

100SB.o:100SB.c

gcc –c 100SB.c

Makefile

make SB

make

```
CFLAGS=-g

SB:1SB.o 2SB.o 3SB.o 4SB.o 5SB.o   …  …  100SB.o
        gcc –o SB 1SB.o 2SB.o 3SB.o 4SB.o 5SB.o   …  …  100SB.o
1SB.o:1SB.c
        gcc –c $(CFLAGS)  1SB.c
            .
            .
            .
100SB.o:100SB.c
        gcc –c $(CFLAGS) 100SB.c
```

```
CFLAGS=-g
LDFLAGS=-L/usr/lib/gsl  -lgsl
SB:1SB.o 2SB.o 3SB.o 4SB.o 5SB.o   …   …  100SB.o
        gcc –o SB  $(LDFLAGS)  1SB.o 2SB.o  …  …  100SB.o
1SB.o:1SB.c
        gcc –c $(CFLAGS)  1SB.c
            .
            .
            .
100SB.o:100SB.c
        gcc –c $(CFLAGS) 100SB.c
```

```
CFLAGS=-g
LDFLAGS=-L/usr/lib/gsl  -lgsl
all:SB

SB:1SB.o 2SB.o 3SB.o 4SB.o 5SB.o   …  …  100SB.o
        gcc –o SB  $(LDFLAGS)  1SB.o 2SB.o  …  …  100SB.o
1SB.o:1SB.c
        gcc –c $(CFLAGS)  1SB.c
            .
            .
            .
100SB.o:100SB.c
        gcc –c $(CFLAGS) 100SB.c
```

make

```
CFLAGS=-g
LDFLAGS=-L/usr/lib/gsl  -lgsl
all:SB


clean:
        rm   *.o    SB
SB:1SB.o 2SB.o 3SB.o 4SB.o 5SB.o   …   …  100SB.o
        gcc –o SB  $(LDFLAGS)  1SB.o 2SB.o  …  …  100SB.o
1SB.o:1SB.c
        gcc –c $(CFLAGS)  1SB.c
100SB.o:100SB.c
        gcc –c $(CFLAGS) 100SB.c
```

make clean

目标:依赖项列表

（Tab缩进）命令

目标：欲生成的目标文件

依赖项：生成目标需要的文件

原理：

判断目标是否存在，如果不存在，或者虽然存在，但比依赖项要旧，则执行命令

过程：如果没在make命令中指明，则从第一个目标开始，递归地检查规则

```
CFLAGS=-g
LDFLAGS=-L/usr/lib/gsl  -lgsl
all:SB


clean:
        rm   *.o    SB
SB:1SB.o 2SB.o 3SB.o 4SB.o 5SB.o   …  …  100SB.o
        gcc –o SB  $(LDFLAGS)  1SB.o 2SB.o  …  …  100SB.o
1SB.o:1SB.c
        gcc –c $(CFLAGS)  1SB.c
100SB.o:100SB.c
        gcc –c $(CFLAGS) 100SB.c
```

```
COMPILE.c = $(CC) $(CFLAGS) $(CPPFLAGS)
$(TARGET_ARCH) -c

CC = gcc

.c.o:

        $(COMPILE.c) $(OUTPUT_OPTION) $<
```

隐含规则

变量名	缺省	意义
CC	gcc	C编译器名称
CXX	g++	C++编译器名称
CFLAGS		C编译器选项
CXXFLAGS		C++编译器选项
CPPFLAGS		编译预处理选项-I，-D，-U

make –p：显示规则（包括隐含规则）

make –f filename：指定用filename文件，而不是默认的Makefile文件来寻找规则

make –n：显示为了达到某个目标，make想执行的命令，但不实际执行