

1 makefile 快速入门

简单的 makefile 如下所示：

```
ASM = nasm
ASMFLAGS = -I include/

TARGET = boot.bin loader.bin

.PHONY: everything clean all

everything: $(TARGET)

clean:
    rm -f $(TARGET)

all: clean everything

boot.bin: boot.asm include/load.inc include/fat12hdr.inc
    $(ASM) $(ASMFLAGS) -o $@ $<

loader.bin: loader.asm include/load.inc include/fat12hdr.inc include/pm.inc
    $(ASM) $(ASMFLAGS) -o $@ $<
```

夸张的说，明白以下两点，makefile 的内容就明白了一半：

1. “=” 用来定义变量。
2. ASM 和 ASMFLAGS 就是两个变量，使用 \$(ASM) 和 \$(ASMFLAGS) 调用这两个变量。

makefile 最重要的语法如下：

```
target: prerequisites
    command
```

这个语法的意思是：

1. 要想得到 target，需要执行命令 command。
2. target 依赖 prerequisites，当 prerequisites 中至少有一个文件比 target 文件新时，command 才会被执行。

再来看下面的语句：

```
$(ASM) $(ASMFLAGS) -o $@ $<
```

其中，\$@ 代表 target，\$< 代表 prerequisites 中的第一个文件，这条语句相当于：

```
nasm -I include/ -o loader.bin loader.asm
```

再来看 everything、clean 和 all，它们 3 个不是文件，只是一个动作名称。比如“make clean”将会执行“rm -f \$(TARGET)”。关键字.PHONY 用来声明这些动作名称。

all 后面跟着 clean 和 everything，这意味着如果执行“make all”，那么 clean 和 everything 下面的动作也会被执行。“make all”执行的结果如下：

```
rm -f boot.bin loader.bin
nasm -I include/ -o boot.bin boot.asm
nasm -I include/ -o loader.bin loader.asm
```