# Makefile学习—变量

## 1.变量基础

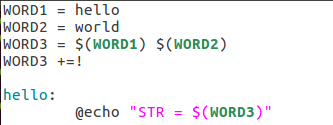
### 1.1变量定义

CC = gcc

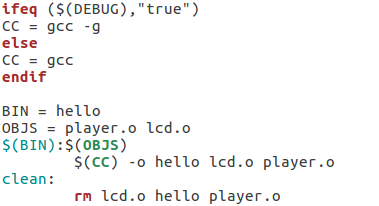
### 1.2变量赋值

追加赋值：+=

条件赋值：？=



### 1.3变量引用

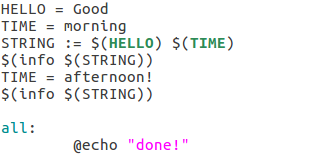


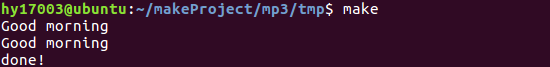
## 2.变量分类

### 2.1立即展开变量

使用:=操作符赋值

在解析阶段直接赋值常量字符串

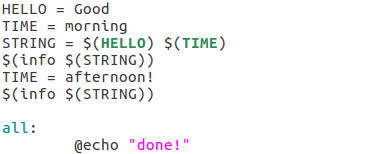


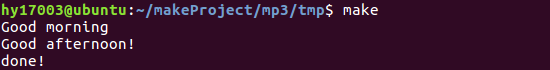


### 2.2延迟展开变量

使用=操作符赋值

在运行阶段，实际使用变量时变量时再进行求值





### 2.3注意事项

一般在目标、目标依赖中使用立即展开变量

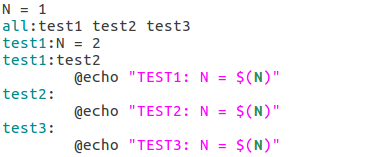
在命令中一般使用延迟展开变量

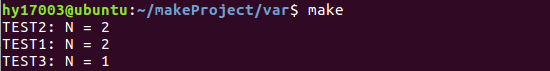
## 3.目标变量

一般变量：默认为全局变量

目标变量：指针对某些目标的变量，是局部变量，该目标所依赖的规则中都可以使用

使用目标变量的用途：做到文件级的编译选项

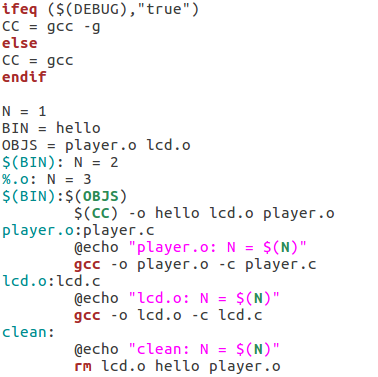


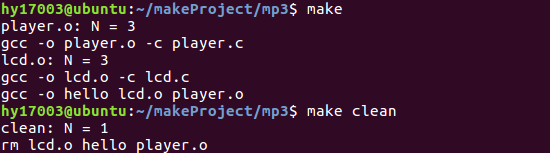


N = 1是全局变量，N = 2是局部变量，test1依赖于test2，因此N=2在test1和test2中都可以使用。

## 4.模式变量

变量可以定义在符合某种模式的目标上，以下%.o: N = 3





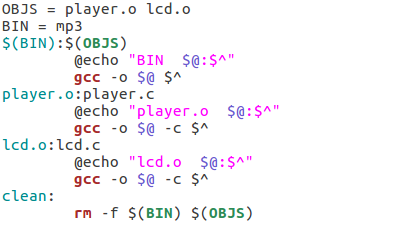
## 5.自动变量

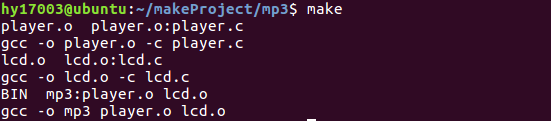
自动变量是局部变量

目标：$@

所有依赖：$^

第一个依赖：$<





## 6.系统环境变量

### 6.1作用范围

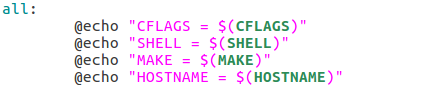
变量在make开始运行时被载入到Makefile文件中，对所有的Makefile都有效，若Makefile中定义同名变量，系统环境变量将被覆盖，命令行中传递同名变量，系统环境变量将被覆盖

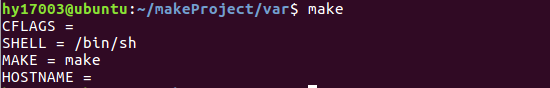
### 6.2常见的系统环境变量

CFLAGS

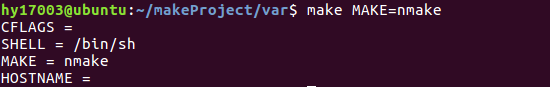
SHELL

MAKE





传递参数：



## 7 变量的传递

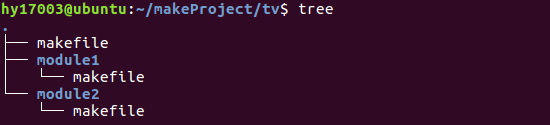
### 7.1Makefile在多目录下递归执行

$(MAKE) –C subdir

Cd subdir && $(MAKE)

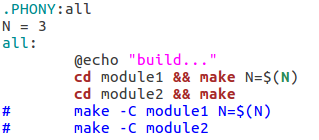
### 7.2通过命令行传递变量

例如，有以下文件结构：



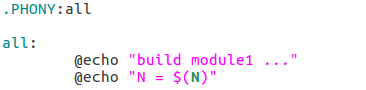
其中各makefile的内容如下：

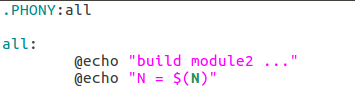
主makefile，使用命令行的方式对module1中的makefile传递一个N值，但对moudle2中的makefile并没有传递值。

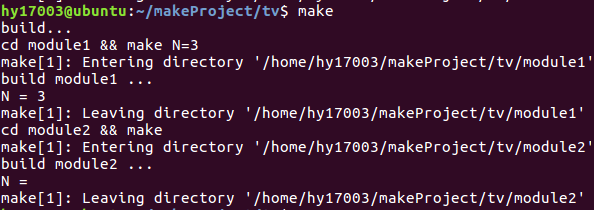


以上注释部分与上两条代码等价

其余两个makefile:







从结果可以看出，主makefile将N = 3以命令行的方式传递给module1中的makefile，从而可以打印出N = 3，而module2中的makefile并没有打印出N的值信息，即其值为空，因此，主makefile中定义的变量只在自身makefile文件中有效，并不能影响到module2中去。

### 7.3通过export传递变量

变量的传递还可以使用export的形式，使用export的形式，变量将变成一个全局变量，其作用范围是所有的makefile，如：

