

零基础学写 Makefile

- [Makefile 简易教程](#)

目录

- [一. Makefile 基础知识](#)
 - [1.1 从一个简单的例子开始认识 Makefile](#)
 - [1.2 Makefile 的基本结构](#)
 - [1.3 Makefile 的基础语法](#)
- [二. Makefile 实例](#)
 - [2.1 编写简单的 Makefile](#)
 - [2.2 Makefile 中自定义类的链接](#)
 - [2.3 如何在 Makefile 里面链接 ROOT](#)

一. Makefile 基础知识

- make 主要用来编译源代码,生成结果代码,然后把结果代码连接起来生成可执行文件或库文件.
- makefile 实质是用来管理 make 命令的特殊文件.

1.1 从一个简单的例子开始认识 Makefile

- **例子: 主程序代码(main.C), 源代码(getop.C,stack.C,getch.C), 头文件(calc.h)**

编译命令: `gcc -o calc main.C getch.C getop.C stack.C`

每当重新编译时,都需要手动输入上述命令

- **一个最简单的 Makefile**

第一部分: 目标 calc. 在冒号之前,为所要编译的目标程序.

第二部分: 依赖关系表. 冒号后的部分(main.C getch.C getop.C stack.C)

第三部分: 编译命令. `gcc -o calc main.C getch.C getop.C stack.C`

- 依赖关系中的文件,只要有一个发生了变化,就会触发编译命令.
- 语法规定 Makefile 中的任何命令之前都必须要有有一个tab缩进.

```
calc: main.C getch.C getop.C stack.C
    gcc -o calc main.C getch.C getop.C stack.C
```

- **Makefile 改进 -- 效率提高**

将编译器、目标文件、源文件定义为宏, 即字符串替换.

```
cc      = gcc
prom    = calc
source  = main.C getch.C getop.C stack.C

$(prom): $(source)
        $(cc) -o $(prom) $(source)
```

- **Makefile 改进 -- 添加头文件依赖**

```
cc      = gcc
prom    = calc
deps    = calc.h
obj     = main.o getch.o getop.o stack.o

$(prom): $(obj)
        $(cc) -o $(prom) $(obj)

main.o: main.C $(deps)
        $(cc) -c main.C

getch.o: getch.C $(deps)
        $(cc) -c getch.C

getop.o: getop.C $(deps)
        $(cc) -c getop.C

stack.o: stack.C $(deps)
        $(cc) -c stack.C
```

- **Makefile 改进 -- 效率提高**

使用特殊的宏命令, 使 Makefile 更加简洁

`%.o: %.C $(deps)` 表示所有的.o 目标文件都依赖于与它同名的.C文件, 以及deps中的头文件

```
cc      = gcc
prom    = calc
deps    = calc.h
obj     = main.o getch.o getop.o stack.o

$(prom): $(obj)
        $(cc) -o $(prom) $(obj)

%.o: %.C $(deps)
        $(cc) -c $< -o $@
```

1.2 Makefile 的基本结构

1.3 Makefile 的基础语法

二. Makefile 实例

2.1 编写简单的 Makefile

2.2 Makefile 中自定义类的链接

2.3 如何在 Makefile 里面链接 ROOT

- [Linking ROOT Libraries to a MakeFile install](#)

```
// Makefile
CC          = g++
CFLAGS      = -c -g -Wall `root-config --cflags`
LDFLAGS     = `root-config --glibs`
SOURCES     = test.C
OBJECTS     = $(SOURCES:.C=.o)
EXECUTABLE  = test

all: $(SOURCES) $(EXECUTABLE)

$(EXECUTABLE): $(OBJECTS)
    $(CC) $(OBJECTS) $(LDFLAGS) -o $@
//## 注意: $(CC) $(LDFLAGS) $(OBJECTS) -o $@ 可能会报以下错误:
//## Undefined reference to `TVersionCheck::TVersionCheck(int)

.C.o:
    $(CC) $(CFLAGS) $< -o $@

clean:
    rm ./~ ./*.o ./test
```