

# Linux系统及Shell编程

东北林业大学

卢洋



# 第三章

## Linux C 开发工具



1. 编辑器vi/vim的使用
2. 编译器gcc的使用
3. 调试器gdb的使用
4. 构建工具make的使用



3.4

工程管理器make的使用



## 2 Makefile的文件格式

- Makefile文件由一系列规则构成。每条规则的形式如下：

`<target>: <prerequisites>`

`[tab] <command>`

目标: 前置条件

`[必须有个tab键] <命令>`

- “目标”是必须的；“前置条件”和“命令”是可选的，但两者必存其一。
- 明确：构建目标的前置条件是什么，以及如何构建。



make a.txt

b.txt c.txt



# 目标(target)

- 一个目标 (**target**) 构成一个规则。目标通常是文件名，指明命令所要构建的对象，比如**a.txt**。
- 目标可以是一个文件名，也可以是多个文件名，之间用空格分隔。



# 目标(target)

- 除了文件名，目标还可以是某个操作的名字，这称之为“伪目标”（phony target）。

`clean:`

`rm *.o`

- 上面代码的目标是**clean**，不是一个文件名，是一个操作的名字，属于“伪目标”，作用是删除文件。

`$ make clean`



有没有可能，在当前路径下，刚好有一个文件叫做clean?



# 目标(target)

- 为了避免这种情况，可以声明**clean**是“伪目标”：

```
.PHONY: clean
```

```
clean:
```

```
    rm *.o
```

- 声明**clean**是“伪目标”之后，**make**不会去检查是否存在一个叫做**clean**的文件，而是每次运行都执行对应的命令。



# 目标(target)

- 如果make命令运行时没有指定目标，默认会执行Makefile文件的第一个目标。