

# 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



- ❶ 代码编程可执行文件，叫做编译 (**compile**) ；
- ❷ 在一个项目中，先编译这个，还是先编译那个（即编译的安排），叫做构建 (**build**) 。
- ❸ **make**是最常用的构建工具，诞生于1977年，主要用于C语言想项目。
- ❹ 实际上，任何只要某个文件发生变化，就要重现构建的项目都可以用**make**构建。



# 1 make的概念

- **make**: 制作。比如，要做出文件**a.txt**，就可以执行：

```
$ make a.txt
```

- **make**本身不知道如何做出**a.txt**，需要告知**make**如何调用其它命令完成这个目标。



- 比如，假设文件**a.txt**依赖于**b.txt**和**c.txt**，是后面两个文件连接（**cat**命令）的产物。那么，**make**需要知道下面的规则：

**a.txt: b.txt c.txt**

**cat b.txt c.txt > a.txt**

- 也就是，**make a.txt**这条命令的背后实际分为两步：
  - (1) 确认**b.txt**和**c.txt**必须已经存在；
  - (2) 使用**cat**命令将两个文件进行合并，输出为新的文件。



- 像这样的规则，需要写在一个名为**Makefile**的文件中，**make**命令依赖这个文件进行构建。
- **Makefile**也可以写成**makefile**，或者用命令行参数指定为其它文件名：

```
$ make -f rules.txt
```

```
$ make --file=rules.txt
```

- **make**只是一个根据指定的**Shell**命令进行构建的工具。