

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. 工程管理器`make`的使用

3.2

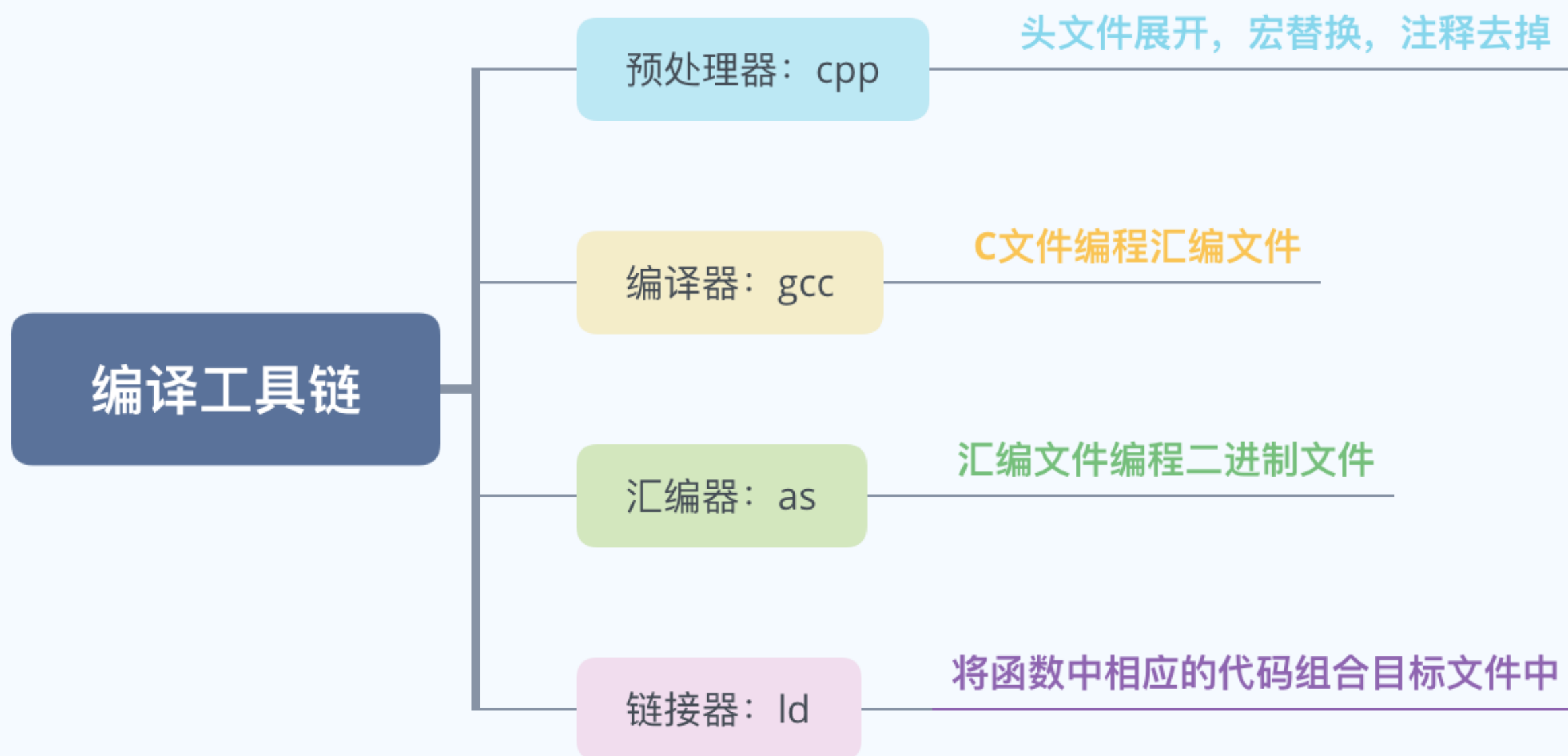
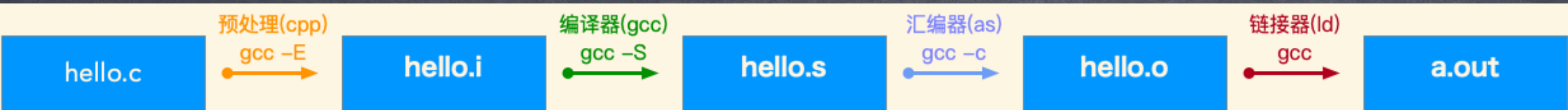
编译器gcc的使用

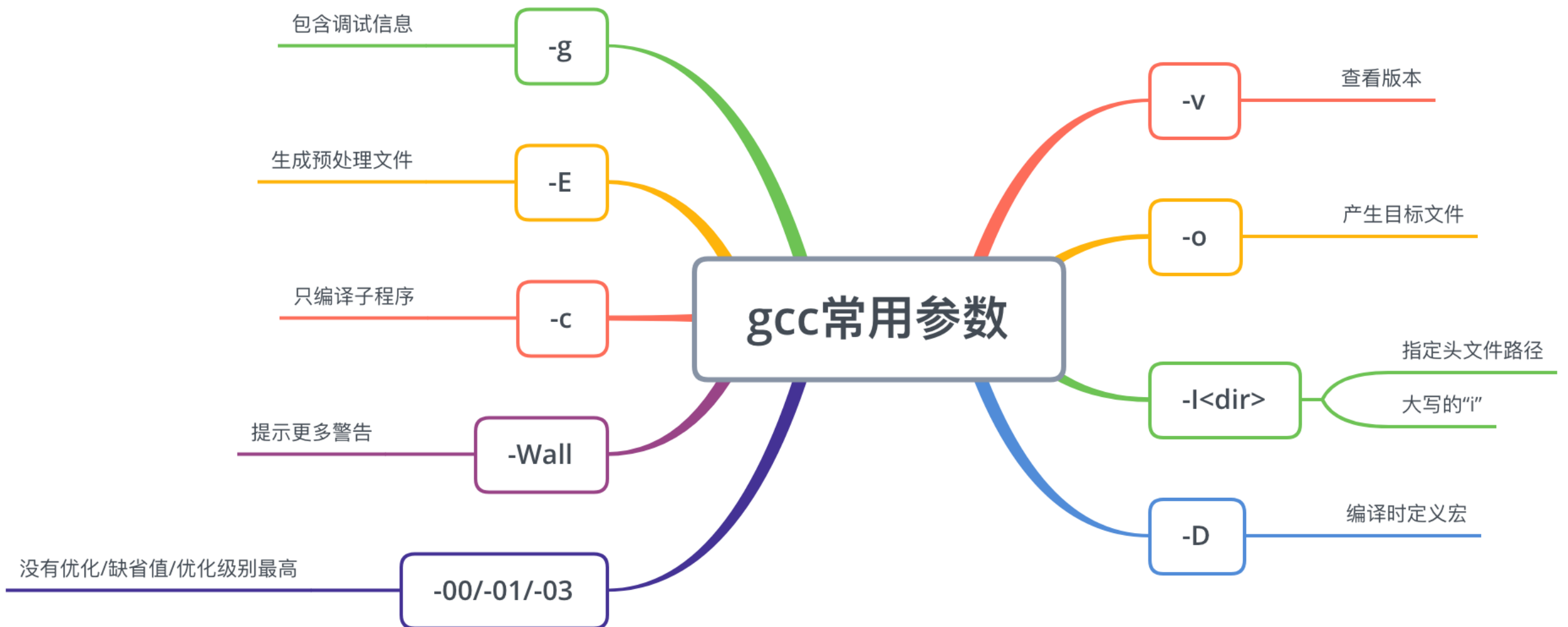
gcc概述

- GCC: GNU Compiler Collection,
GNU编译套件
- GNU: GNU's Not Unix
- GCC是由GNU开发的编程语言编译器, 包括
C、C++、Objective-C、Fortran、Java、
Ada、Golang。

gcc编译流程分析

- 基本语法
- `gcc [option | filename]`





gcc编译流程分析

准备

- 编写代码
- 创建`test.c`，并写入（此时代码包含错误）：

```
#include <stdio.h>
```

```
void main(){
```

```
    scanf("%d", a);
```

```
    printf("a = %d\n", a);
```

```
}
```


预处理

- 输入：编写的文件*.c

- 输出：中间文件*.i

- 命令：

`gcc -E test.c -o test.i`

- 提示错误

gcc编译流程

① 预处理

`gcc -E [编译文件] -o [目标文件]`

② 编译阶段

`gcc -S [编译文件] -o [目标文件]`

③ 汇编阶段

`gcc -c [编译文件] -o [目标文件]`

④ 链接阶段

`gcc [编译文件] -o [目标文件]`

编译阶段

- 输入：中间文件*.i
- 输出：汇编语言文件*.s
- 功能：检查语法错误
- 命令：

`gcc -S test.i -o test.s`

- 提示错误

gcc编译流程

① 预处理

`gcc -E [编译文件] -o [目标文件]`

② 编译阶段

`gcc -S [编译文件] -o [目标文件]`

③ 汇编阶段

`gcc -c [编译文件] -o [目标文件]`

④ 链接阶段

`gcc [编译文件] -o [目标文件]`

汇编阶段

- 输入：汇编语言文件*.s

- 输出：二进制机器码*.o

- 命令：

`gcc -c test.s -o test.o`

- 选项-c：使编译器完成汇编阶段就停止

gcc编译流程

① 预处理

`gcc -E [编译文件] -o [目标文件]`

② 编译阶段

`gcc -S [编译文件] -o [目标文件]`

③ 汇编阶段

`gcc -c [编译文件] -o [目标文件]`

④ 链接阶段

`gcc [编译文件] -o [目标文件]`

链接阶段

- 输入：二进制机器代码文件*.o

- 输出：可执行的二进制文件

- 命令：

`gcc test.o -o test`

- 运行./test即可执行程序

gcc编译流程

① 预处理

`gcc -E [编译文件] -o [目标文件]`

② 编译阶段

`gcc -S [编译文件] -o [目标文件]`

③ 汇编阶段

`gcc -c [编译文件] -o [目标文件]`

④ 链接阶段

`gcc [编译文件] -o [目标文件]`

一次性编译

- `gcc [编译文件] -o [目标文件]`
- `gcc test.c -o test`