



四川大学
国家示范性软件学院
SCU Software collage.



Operating System

Lab04 Introduction



四川大学

国家示范性软件学院

SCU Software College

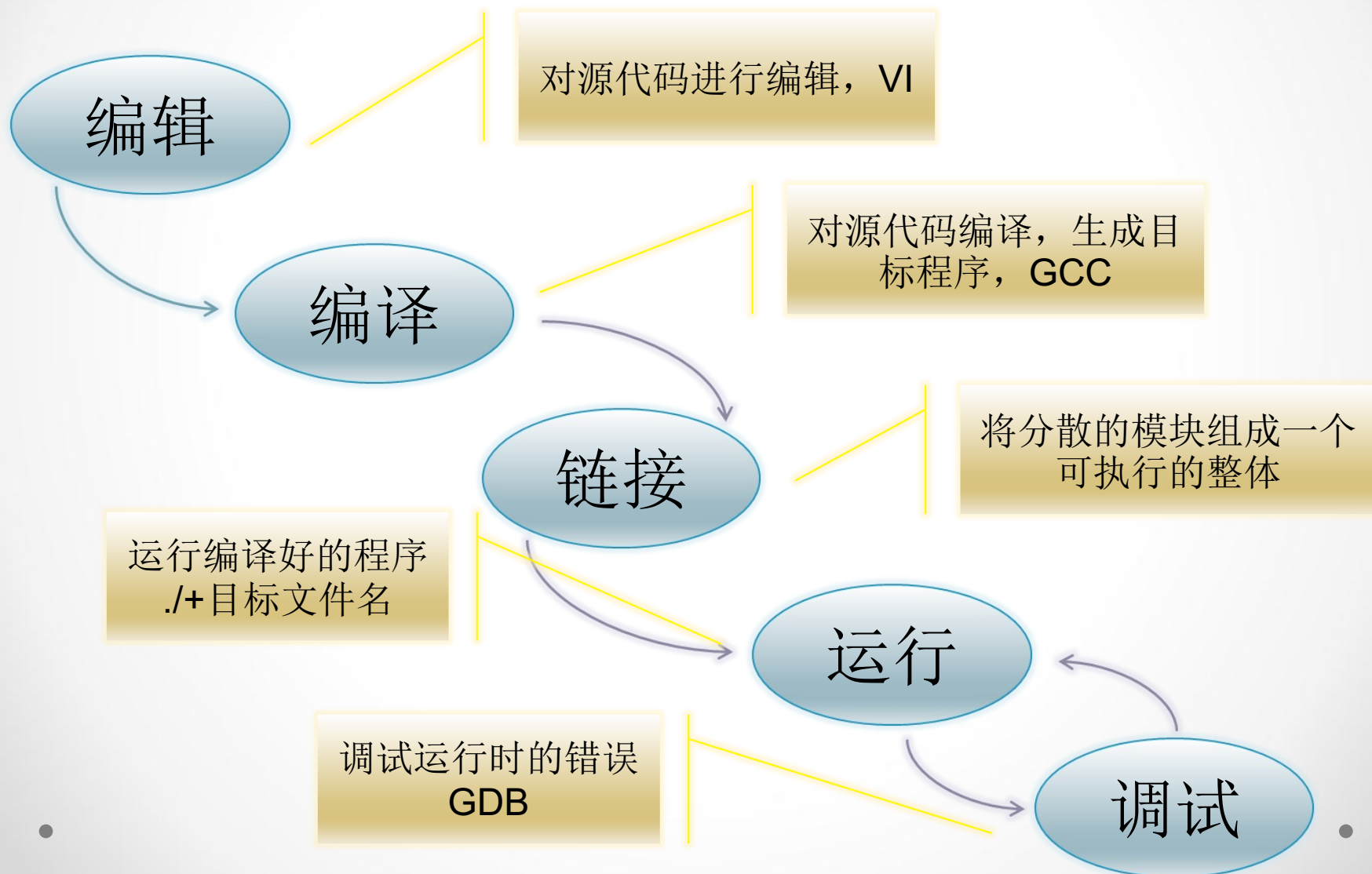


目录

1. Linux编程概述
2. Makefile文件简介
3. GDB调试程序



1、Linux编程概述_{1/2}





四川大学

国家示范性软件学院

SCU Software College



1、Linux编程概述_{2/2}

Linux开发工具

◎ 编辑工具

- VI

◎ 编译工具

- GCC

◎ 链接工具

- Make

◎ 调试工具

- GDB



四川大学

国家示范性软件学院

SCU Software College



目录

1. Linux编程概述
2. Makefile文件简介
3. GDB调试程序



Makefile文件简介_{1/7}

◎ What is Make?

- Make 是用于自动编译、链接程序的实用工具。

◎ Make的优点

- “自动化编译”，一旦写好，只需要一个make命令，整个工程完全自动编译，极大的提高了软件开发的效率。
- “自动”是指它能够根据文件时间戳自动发现更新过的文件而减少编译的工作量。



Makefile文件简介_{2/7}

□ **Make**在执行时，会默认的在当前目录下寻找名为**Makefile**或**makefile**的文件，当名字不为这两者之一时，可以使用如下方法指定：

make -f 文件名

□ **Makefile**是一个文本形式的数据库文件

- 描述了整个工程的编译、链接等规则
- 工程中的哪些文件需要编译，具体怎么编译
- 需要创建哪些库文件以及如何创建这些库文件
- 如何产生我们最后想要得到的可执行文件



Makefile文件简介_{3/7}

□ Makefile编写规则

目标(target): 依赖(prerequisites)...

<tab>命令(command)

- **目标(target)**通常是要产生的文件的名称，目标的例子是可执行文件或OBJ文件。目标也可是一个执行的动作名称，这样的目标通常称为伪目标（PHONY）诸如 ‘clean’。
- **依赖**是用来输入从而产生目标的文件，一个目标经常有几个依赖。
- **命令**是Make执行的动作，一个规则可以含有几个命令，每个命令占一行。
- 每个命令行前面必须是一个**Tab**字符，即命令行第一个字符是**Tab**。这是不小心容易出错的地方。



Makefile文件简介_{4/7}

□例：编译两个源文件的makefile文件

```
hello:string.o main.o
```

```
gcc string.o main.o -o hello
```

```
string.o:string.c
```

```
gcc -c string.c
```

```
main.o:main.c
```

```
gcc -c main.c
```

```
.PHONY:clean
```

```
clean:
```

```
rm -f hello string.o main.o
```



Makefile文件简介_{5/7}

□ Makefile的高级写法

```
CC = gcc
```

```
OBJECTS = string.o main.o
```

```
SOURCES = string.c main.c
```

```
TARGET = hello
```

```
$(TARGET): $(OBJECTS)
```

```
    $(CC) -o $(TARGET) $(OBJECTS)
```

```
$(OBJECTS): $(SOURCES)
```

```
    $(CC) -c $(SOURCES)
```

```
all:
```

```
    make $(TARGET)
```

```
clean:
```

```
    rm -f *.o
```

- ```
rm -f $(TARGET)
```

```
hello:string.o main.o
```

```
 gcc string.o main.o -o hello
```

```
string.o:string.c
```

```
 gcc -c string.c
```

```
main.o:main.c
```

```
 gcc -c main.c
```

```
.PHONY:clean
```

```
clean:
```

```
 rm -f hello string.o main.o
```



# Makefile文件简介<sub>6/7</sub>

## □ Makefile注释

makefile文件中只有行注释，其注释用“**#**”字符，就像C语言中的“**//**”，如果要在makefile文件中使用“**#**”字符，可以使用“**\**”进行转义，即“**\#**”。



# Makefile文件简介<sub>7/7</sub>

## □ Make的语法及常用参数

make的使用

- `make [options] [target]...`
- 常见的选项：
  - `-f FILE` 以指定的FILE 文件作为makefile。
  - `-n` 只打印要执行的命令，但不执行这些命令。
  - `-s` 在执行命令时不显示命令。
  - `-d` 显示调试信息



四川大学

国家示范性软件学院

SCU Software College



# 目录

---

1. Linux编程概述
2. Makefile文件简介
3. GDB调试程序



# GDB调试程序<sub>1/5</sub>

- ◎ Gdb是一款GNU开发组织并发布的UNIX/Linux下的程序调试工具。它使你能在程序运行时观察程序的内部结构和内存的使用情况. 以下是gdb 所提供的一些功能:
  - 它使你能监视你程序中变量的值.
  - 它使你能设置断点以使程序在指定的代码行上停止执行.
  - 它使你能一行行的执行你的代码.



# GDB调试程序<sub>2/5</sub>

## □ GDB的使用方法

- ① 1. 使用gcc命令进行编译（务必需要-g选项）

gcc -g -o buffer buffer.c

- ② 2. 启动gdb进行调试

gdb buffer

- ③ 3. 使用run命令运行程序

r (or run)

- ④ 4. 根据逻辑找出问题，可附加使用list查看代码

l (or list)



# GDB调试程序<sub>3/5</sub>

- ⑤ 使用break命令设置断点，c继续  
b (or break) lineNumber or functionName
- ⑥ 使用run命令运行程序  
r (or run)
- ⑦ 使用watch命令指定需要跟踪的变量  
watch Buffer
- ⑧ 按照终端提示输入input字符串，其他  
gdb 模式下输入n (or next)或者回车继续执行
- ⑨ 重复8，找出症结所在





四川大学

国家示范性软件学院

SCU Software College



# GDB调试程序<sub>4/5</sub>

- ◎ 10. 使用bt命令查看栈信息
- ◎ 11. 使用info b (or break)查看断点信息
- ◎ 12. 使用delete或者clear命令清除断点
- ◎ 13. kill杀死当前正在调试的程序进程
- ◎ 14. 使用quit命令退出gdb调试环境



四川大学

国家示范性软件学院

SCU Software College



# GDB调试程序<sub>5/5</sub>

## □ 程序的控制指令

- **run**: 程序开始执行，一直运行到断点才终止；
- **continue**: 运行到下一个断点；
- **next**: 运行下一条指令；
- **step**: 运行下一条指令；



四川大学

国家示范性软件学院

SCU Software College



---

THE END...

THANK YOU~