




Linux服务器开发

主讲人：李涛





- 工具篇
- Linux系统编程篇
- Linux网络编程篇
- 面试题



- 有C/C++语言基础
- Linux基础使用经验
- 在校大学生
- Linux运维人员想学Linux开发



什么是gcc?



C语言的编译过程

- 预处理
 - 将所有#define删除，并且展开
 - 处理所有条件预处理指令
 - 处理#include，将被包含的文件插入到该预编译指令的位置
 - 过滤所有的//、/**/
 - 保留所有#pragma编译指令
- 编译
 - 词法分析
 - 语法分析
 - 语义分析
 - 代码优化
 - 目标代码生成
- 汇编
- 链接
 - 静态链接
 - 动态链接



gcc的常用选项

- -E
- -O
- -S
- -c
- -Wall
- -g
- -I
- -L
- -l
- -MM



- /usr/include
- /usr/local/include



- ar - cr
- -L 路径
- -l 库名



- -fPIC
- -shared



编译有动态库的程序



运行有动态库的程序

- ldconfig /etc/ld.so.conf
- /usr/lib
- LD_LIBRARY_PATH



- gdb 程序名
- gdb 程序名 core



常用操作

- list 查看源代码
- b 行号或函数名，在对应的行或函数入口设置断点
- info break 查看断点信息
- run 执行命令
- next 单条语句执行，不进入函数
- step 单条语句执行，进入函数
- print 变量，打印变量的值
- whatis 变量，查看变量的类型
- finish 退出函数
- bt 查看堆栈
- quit 结束调试



gdb分析coredump文件

- ulimit -c unlimited
- gdb 程序名 core文件名



- makefile用于定义文件间的依赖关系

目标： 依赖文件
命令

```
main : main.o  
gcc main.o -o main
```

- 依赖规则
 - 目标对应的文件不存在，执行对应的命令
 - 依赖文件的时间比目标文件更新时，执行命令
 - 对比依赖链上的每个目标



- 默认情况下，make认为每个目标是一个文件
- .PHONY声明一个伪目标
 - 伪目标不对应实际的文件
 - 不管伪目标对应的依赖是否更新，命令总是执行



- 定义变量

- CC := gcc

- TARGET := main

- 使用变量

- \$(TARGET)

- := 和 = 的区别



$\$@$

$\$^$

$\$<$



变量值的替换

- 使用指定字符替换变量值的后缀

- $\$(var:a=b)$

```
src := add.c sub.c  
obj := $(src:.c=.o)
```

- 规则中的替换

— 目标: 目标中匹配子目标: 从子目标生成依赖

```
obj := add.o sub.o  
$(obj) : %.o : %.c  
$(CC) -o $@ -c $^  
↓  
add.o : add.c  
$(CC) -o $@ -c $^  
sub.o : sub.c  
$(CC) -o $@ -c $^
```



`$(wildcard *.c)`表示扫描当前目录下所有.c文件

`SRCS = $(wildcard *.c)`



谢谢观看

更多好课，请关注万门大学APP

