电子科技大学

University of Electronic Science and Technology of China

Linux操作系统编程

实验一 | 开发环境实验

主讲老师：杨珊

实验一｜开发环境实验目的

目的一：熟悉基于Linux内核的操作系统

- ✓ ubuntu 11或以上（bourne again shell）

目的二：熟悉基于GNU工具链的开发环境

- ✓gcc 4.5.2
- ✓gdb 7.2

目的三：掌握Linux程序编译调试方法

实验一｜实验原理｜控制台初步操作

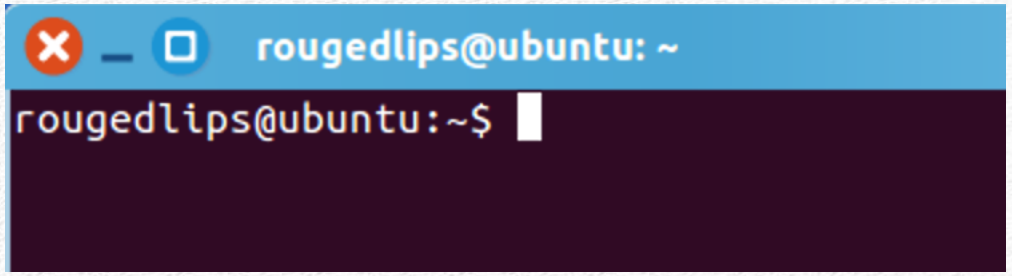
通过控制台+命令操作Linux操作系统（ubuntu 11或以上）

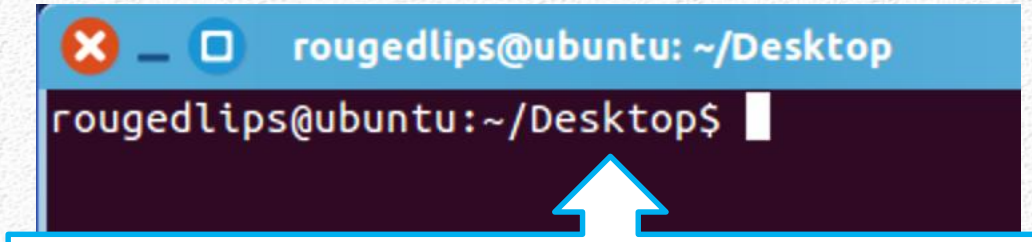
打开控制台：

- ✓ 点击搜索您的电脑和在线资源 ->下方图标选择应用程序 ->已安装 ->终端
- ✓ 快捷键：CTRL+ALT+T
- ✓ 图形界面下桌面左边的快捷方式栏的终端图标
- ✓ 右键点击后，会有“在终端中打开”选项

实验一 | 实验原理 | 编辑代码

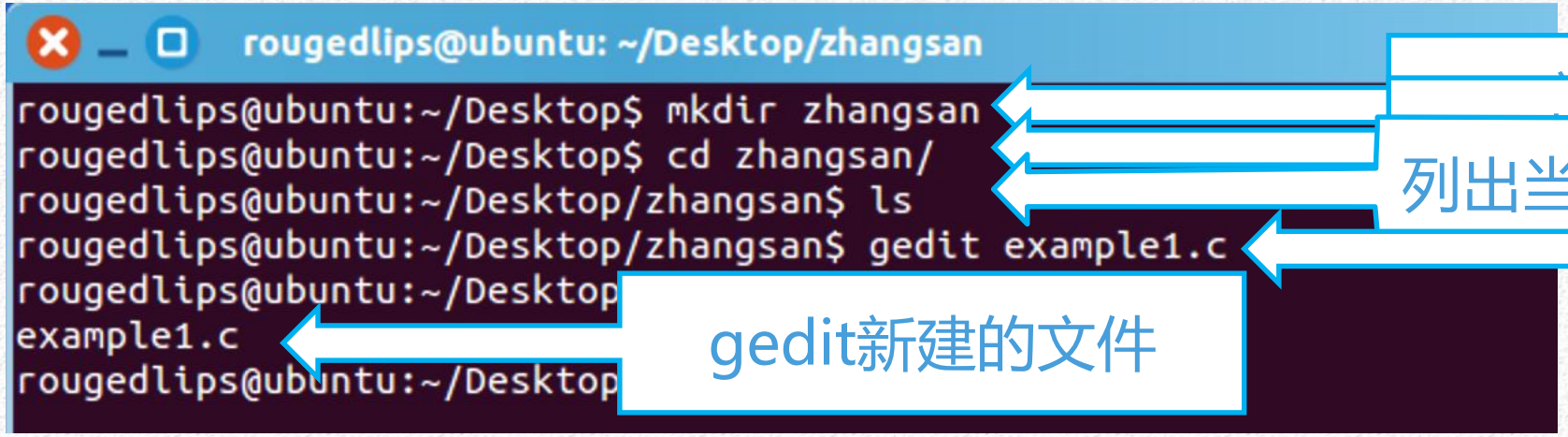
打开终端后，光标前的#或者$符号前会指示当前文件夹的路径





当前目录是/ home/<用户名>/**Desktop**

使用gedit程序编辑代码



列出当

gedit新建/打开文件

gedit新建的文件

GCC

✓是一个包含众多语言的编译器（C，C++，Ada，Object C，Java及Go等）->GNU Compiler Collection

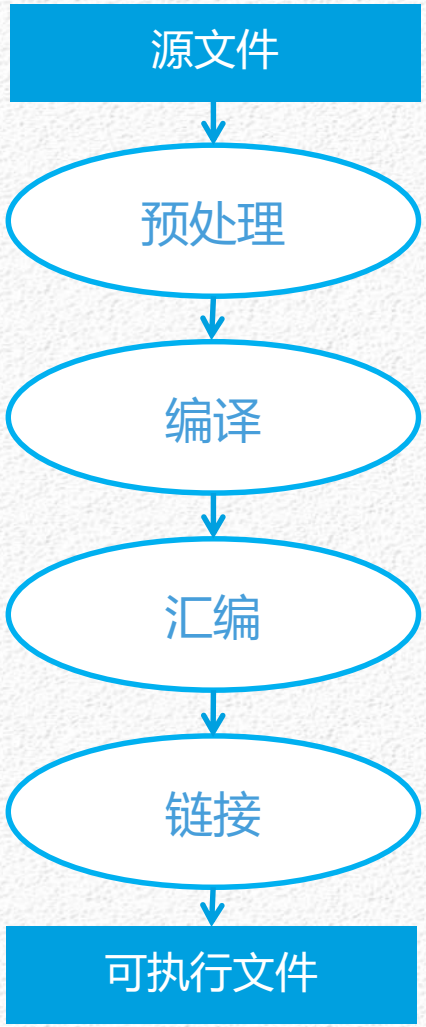
包括:

✓cpp(预处理器)

✓gcc(c 编译器)、g++(c++编译器)等编译器

✓binutils等二进制工具，含【as(汇编器)、 ld(链接器)等等】

实验一 | 实验原理 | GCC编译过程

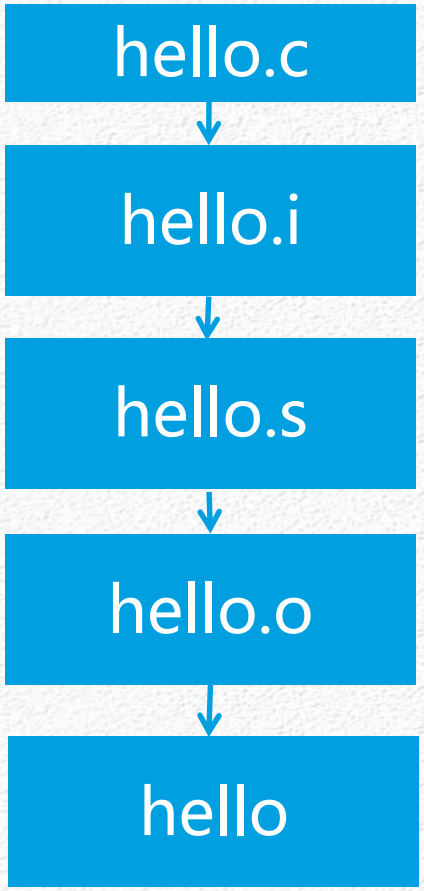


cpp –o hello.i hello.c

ccl –o hello.s hello.i

as –o hello.o hello.s

ld –o hello hello.o



实验一｜实验原理｜GCC命令格式

gcc命令格式：gcc [选项] <文件名>

例：gcc main.c –o main

(默认包含预处理、编译、[illegible]及链接)

-o filename ： 指定输出文件为filename

如无指定，则默认的输出结果是： a.out

编译方式： 首先生成目标文件 example.o add.o modify.o delete.o

例如：gcc –c example.c

–c选项代表生成.o文件

然后链接4个目标文件

gcc example.o add.o modify.o delete.o –o example

实验一 | 实验原理 | GCC命令选项

gcc [编译选项] <文件名>

-E: 只激活预处理

```
[root@localhost linuxkc]# gcc -E example.c >result.txt
[root@localhost linuxkc]# ls
a.out  example1_1  example.c  example.o  example.s  result.txt
[root@localhost linuxkc]#
```

-E 参数选项

出结果重定向到result.txt

-S: 激活预处理和编译

```
[root@localhost linuxkc]# gcc -S example.c
[root@localhost linuxkc]# ls
a.out  example1_1  example.c  example.o  example.s
[root@localhost linuxkc]# _
```

生成汇编代码

-C: 激活预处理、编译、汇编

```
[root@localhost linuxkc]# gcc -c example.c
[root@localhost linuxkc]# ls
a.out  example1_1  example.c  example.o
[root@localhost linuxkc]# _
```

生成目标（obj文件）

实验一 | 实验原理 | GCC命令选项

gcc [调试选项] <文件名>

- ✓ -g：以操作系统的本地格式(stabs, COFF, XCOFF,或DWARF). 产生调试信息. GDB能够使用这些调试信息.
- ✓ -ggdb：ggdb选项可以插入的更为丰富的调试信息，但是生成的可执行文件有可能无法用其它调试器调试

调试选项的启动会让二进制文件的大小急剧增长

gcc [优化选项] <文件名>

- ✓ -O 或者 –O1 优化选项：编译器会试图减少目标码的大小和执行时间
- ✓ -O2：除了涉及空间和速度交换的优化选项,执行几乎所有的优化工作，但不进行循环展开和函数内嵌
- ✓ -O3：乱序执行，循环展开的优化
- ✓ 调试时不能用优化选项，否则变量值和源代码无法对应

实验一 | 实验原理 | GDB介绍

GDB

- ✓ 是GNU计划开发的**程序调试工具**

包括：

- ✓ 启动程序，可以按照自定义的要求运行程序
- ✓ 指定**断点**停住（断点可以是条件表达式）
- ✓ 程序停住时，可以检查当前程序中的情况
- ✓ 动态的改变程序的执行环境

启动方式：

- ✓ 终端（shell）界面运行gdb命令，进入后file program命令装载程序
- ✓ 终端界面启动gbd并加载程序可执行文件 gdb <program>
- ✓ 用gdb同时调试一个运行程序和core文件 gdb program core
- ✓ 调试正在运行的进程 gdb prog[illegible]m <processid> 或者是进入gdb

core是程序非法执行后core dump后产生的文件

实验一｜实验原理｜GDB常用命令

查询某条命令的用法：

- ✓ help [command]

退出gdb

- ✓ quit
- ✓ ctrl+d

```
rougedlips@ubuntu: ~/Desktop/zhangsan
(gdb) help breakpoint
Making program stop at certain points.

List of commands:

awatch -- Set a watchpoint for an expression
break -- Set breakpoint at specified line or function
break-range -- Set a breakpoint for an address range
catch -- Set catchpoints to catch events
catch assert -- Catch failed Ada assertions
catch catch -- Catch an exception
catch exception -- Catch Ada exceptions
catch exec -- Catch calls to exec
catch fork -- Catch calls to fork
catch load -- Catch loads of shared libraries
catch rethrow -- Catch an exception
catch signal -- Catch signals by their names and/or numbers
catch syscall -- Catch system calls by their names and/or numbers
catch throw -- Catch an exception
catch unload -- Catch unloads of shared libraries
catch vfork -- Catch calls to vfork
clear -- Clear breakpoint at specified line or function
commands -- Set commands to be executed when a breakpoint is hit
condition -- Specify breakpoint number N to break only if COND is true
---Type <return> to continue, or q <return> to quit---
```

break命令（缩写b，设置程序暂停运行的地方）：

格式：break [LOCATION] [thr

- ✓ [LOCATION]
- ✓ [thread THREADN
- ✓ [if CONDITION]

断点语句	功能
b 123	停在第123行
b main	停在main函数处
b increas	
b in	

break frik.c:13 thread 28

置断点

b 123 if index==2

当index为2时，程序在123行停下

watchpoint命令（观察对象值变化，则程序立即停止）：

语句格式	功能
watch <expr>	为表达式（变量）expr设置一个观察点。一旦表达式值有变化时，马上停住程序
rwatch <expr>	当表达式（变量）expr被读时，停住程序
awatch <expr>	当表达式（变量）的值被读或被写时，停住程序
info watchpoints	列出当前所设置了的所有观察点

清除禁止断点或者观察点：

语句格式	功能
clear [linenum] [function name]	清除所有断点,不会清除watchpoints。
delete <num>	清除编号为num的断点或者watchpoint。
disable <num>	禁止某个断点。
enable <num>	开启某个断点。

其他常用命令：

语句格式	功能
Info [**]	查看设置的断点，函数名称，类名，例：info b查看已经设置的断点名称
dir 源代码路径	gdb默认有$cdir和$cwd两个源代码搜索路径，如果你要调试程序的源代码不在这两个路径中，可用dir命令增加
r [[参数1]...[参数n]]	在gdb中运行已经装载的可执行文件，参数为程序所需的参数

实验一｜实验原理｜GDB查看运行时数据

print命令（缩写p，查看当前程序运行数据，同义命令inspect）：

格式：print <expr> 或 print /<f> <expr>

- ✓ <expr>当
- ✓ /<f>设置输出格式，例：

断点语句	功能
p i	查看变量i的值
p i*5	查看变量i乘5后的值
p s.name	查看结构体s的成员变量name的值

list命令（简写l，调试过程中打印源码）：

语句格式	功能
list	当前行的上5行和下5行，默认10行
list <linenum>	打印Linenum行的代码
list <+offset>| list <-offset>	当前行号的正偏移量|当前行号的负偏移量
list <filename:linenum>	Filename文件的linenum行
list <filename:function>	Filename文件的function函数
list <*address>	运行时语句在内存中的地址

搜索代码命令

语句格式	功能
forward-search <regexp>	向前搜索
search <regexp> ：	向后搜索
reverse-search <regexp>	全文搜索

注：<regexp>：正则表达式，匹配字符串的模式

实验一｜实验原理｜GDB调试命令

调试命令	功能
step	单步调试命令，一次执行一行程序
next	单步调试命令，但跳过函数调用
finish	单步调试时直接从一个函数中返回
disassemble	显示汇编代码

调试命令	功能
Backtrace 或者bt	查看目前程序的堆栈情况
whre	查看目前程序的堆栈情况
up/down	向上或者向下移动一个堆栈。
frame<num> 或者f	移动到第num个堆栈

注：当移动到某个堆栈时，可以用gdb命令查看在此堆栈中的局部变量。

实验一 | 实验内容

内容一：在Linux下通过C语言程序设计链表应用程序

- ✓ 定义单向链表的数据结构
- ✓ 创建链表
- ✓ 插入结点
- ✓ 遍历结点等

内容二：Linux下通过GNU工具链编译、调试、运行该程序

实验一 | 参考代码片段

```
//预定义数据结构
typedef struct stuInfo {
  char stuName[10]; /*学生姓名*/
  int  Age  /*年龄*/
  } ElemType

typedef struct node {
  Elemtype  data;
  struct  node  *next;
  } ListNode, *ListPtr;
```

```
int main(int argc,char argv[])
{
        while(1)
        {
                printf("1 Create List\n");
                printf("2 Printf List\n");
                printf("3 Insert List\n");
                printf("4 Quit\n");
                char command =getchar();
                switch(command)
                {
                        case '1':ListHead = CreateList();
                        break;
                        case '2':PrintList(ListHead);
                        break;
                        case '3':InsertList(ListHead);
                        break;
                        case '4':
                        return 0;
                }
        }
        return 0;
}
```

电子科技大学

University of Electronic Science and Technology of China

Linux操作系统编程

感谢观看

主讲老师：杨珊