

《Linux 系统应用与开发》

期末报告

学生姓名_____牟鑫一_____

学生学号_____20161001764_____

学生班级_____191174_____

日 期_____2019 年 10 月 24 日_____

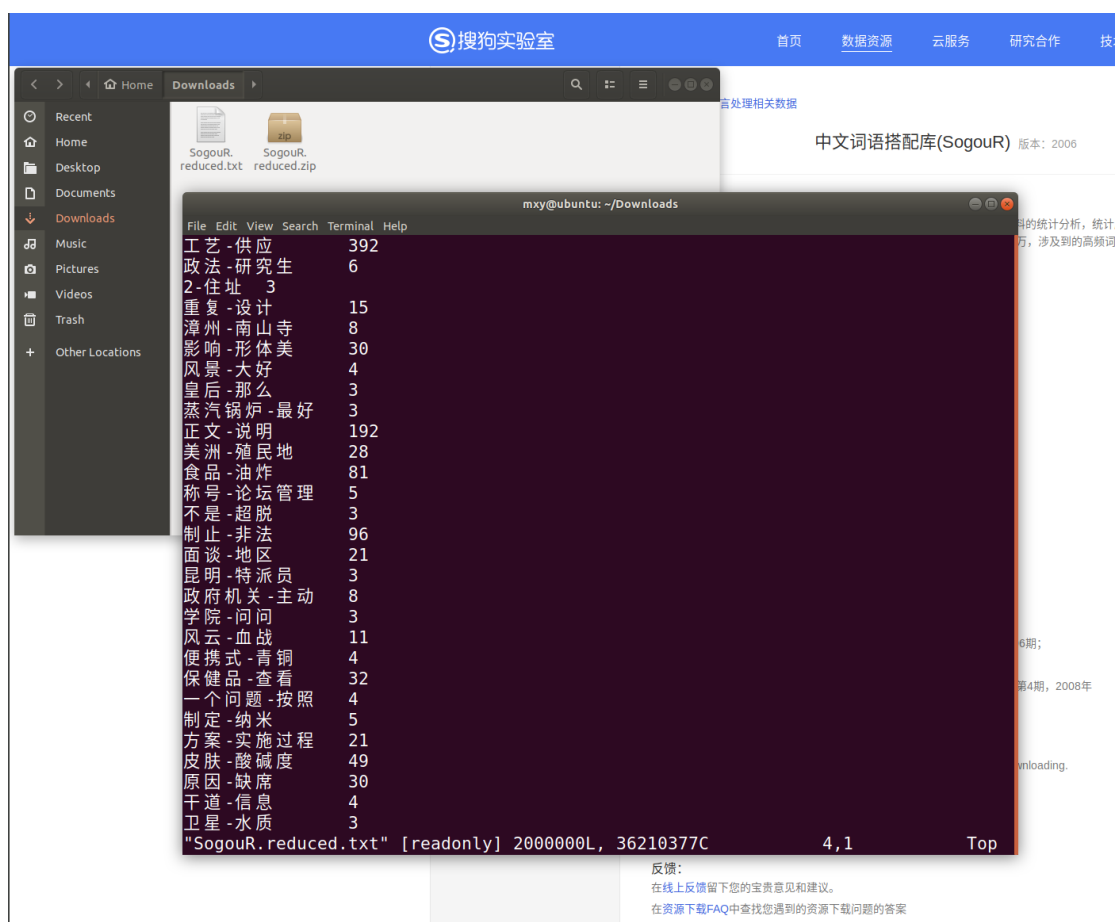
实验二

1. 要求

- 了解数据集作用及格式（本实验下载精简版即可）
- 在 linux 中解压缩并用相关文本命令进行数据查看、统计及分析。（包括但不限于：统计某关键词的词频、对文件进行排序（按照某列或某关键字），对大文件的快速浏览，管道命令、模糊查询（使用通配符）等）
- 按照 ppt 运行并熟悉相关命令。
- 熟悉 vi 或者 vim 的用法，掌握 PPT 中列表中的常用命令（如定位，复制，搜索等）

2. 调试与测试

下载数据集，使用命令 `unzip SogouR.reduced.zip` 解压数据集，使用 `vim` 查看数据集：



使用 `grep` 命令对关键词进行词频统计：

```
mxy@ubuntu: ~/Downloads
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
mxy@ubuntu:~/Downloads$ vim SogouR.reduced.txt
mxy@ubuntu:~/Downloads$ grep -o 不是 SogouR.reduced.txt|wc -l
4230
mxy@ubuntu:~/Downloads$ grep -o 家庭 SogouR.reduced.txt|wc -l
1391
mxy@ubuntu:~/Downloads$ grep -o 大学 SogouR.reduced.txt|wc -l
4797
mxy@ubuntu:~/Downloads$
```

使用 `sort` 命令排序：

打开(O) ▾	文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)	保存(S) ▮
工艺-供应 392	sort: cannot read: sorted.txt: 没有那个文件或目录	
政法-研究生 6	mxy@ubuntu:~/Downloads\$ sort SogouR.reduced.txt > sorted.txt	
2-住址 3	mxy@ubuntu:~/Downloads\$ more sorted.txt	
重复-设计 15	2-爱尔兰 4	
漳州-南山寺 8	2-安装 46	
影响-形体美 30	2-巴黎 4	
风景-大好 4	2-拜仁 3	
皇后-那么 3	2-斑竹 13	
蒸汽锅炉-最好 3	2-班级 34	
正文-说明 192	2-颁奖 4	
美洲-殖民地 28	2-保护 4	
食品-油炸 81	2-保证 7	
称号-论坛管理 5	2-宝马 3	
不是-超脱 3	2-报社 48	
制止-非法 96	2-爆炸 5	
面谈-地区 21	2-北京 126	
昆明-特派员 3	2-贝贝 5	
政府机关-主动 8	2-本机 3	
学院-问问 3	2-本留言 4	
风云-血战 11		
便携式-青铜 4		
保健品-查看 32		
一个问题-按照 4		
制定-纳米 5		
方案-实施过程 21		
皮肤-酸碱度 49		

纯文本 ▾ 制表符宽度: 8 ▾ 第 230709 行, 第 1 列 ▾ 插入

使用 `grep` 命令进行模糊查询：

```
mxy@ubuntu:~/Downloads$ grep 地质大学 SogouR.reduced.txt
科大-地质大学 3
络合物-地质大学 3
位于-地质大学 4
一个-地质大学 3
mxy@ubuntu:~/Downloads$
```

匹配以“大学生活”开头的字符串：

```
mxy@ubuntu:~/Downloads$ grep '^大学生活' SogouR.reduced.txt
大学生活-是不是 3
大学生活-为主题 8
大学生活-同学 5
大学生活-败类 7
大学生活-可谓 3
大学生活-丰富多彩 21
大学生活-每一天 5
大学生活-展望 4
大学生活-模式 5
大学生活-回忆 3
大学生活-怎么 8
大学生活-中文版 35
大学生活-好好珍惜 12
大学生活-体验 7
大学生活-充满 14
mxy@ubuntu:~/Downloads$
```

Vim 匹配并定位：

```
mxy@ubuntu: ~/Downloads
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
很快-膨胀 3
西瓜-竟能 53
商业机会-基本信息 5
小龙女-造型 3
身世-简介 7
高度-可行性 3
烘烤-喷涂 16
误会-尚且 13
位于-地质大学 4
政府-授予 467
找朋友-几个 3
首播-全新 8
租用-期为 3
类专业-考试时间 36
一族-地址 62
服务业-主要 15
周边-风景 40
遵守-和你 3
/地质大学
```

实验三

1. 要求

1) 设计如下一个菜单驱动程序。

Use one of the following options:

P: To display current directory

S: To display the name of running file

D: To display today's date and present time

L: To see the listing of files in your present working directory

W: To see who is logged in

Q: To quit this program

Enter your option and hit:

菜单程序将根据用户输入的选择项给出相应信息。要求对用户的输入忽略大小写，对于无效选项的输入给出相应提示。要求使用case 语句实现以上功能，输入相应的字母后应该执行相应的命令完成每项功能，如输入P 或p，就执行pwd 命令。

2) 编写一段bash Shell 程序，根据从键盘输入的学生成绩，显示相应的成绩等级，其中60分以下为“Failed!”，60~70 分为“Passed!”，70~80 分为“Medium!”，80~90 分为“Good!”，90~100 分为“Excellent!”。如果输入超过100 或负数的分数，则显示错误分数提示。按照ppt运行并熟悉相关命令。

3) 编写一个Shell 过程完成如下功能（必须在脚本中使用函数）。

① 程序接收3 个参数：\$1、\$2 和\$3，合并两个文件\$1、\$2 为\$3，并显示，三个文件均为文本文件。

② 如果文件\$3 不存在，那么先报告缺少\$3，然后将合并后的内容输出到mydoc.txt。如果有\$3，就合并到\$3。

③ 如果文件\$2 和文件\$3 都不存在，那么先报告缺少\$2、\$3，只显示\$1 的内容。

④ 如果文件\$1 不存在，则提示缺少\$1，要求重新运行程序。

4) 编写一个脚本，显示当天日期，查找给定的某用户是否在系统中工作。如果在系统中，就发一个问候给他。

5) 给shell增加一个新的命令mycommand

要求：

编程实现某shell命令的功能，如ls命令，显示当前目录下的目录及文件信息。同学们也可以编写不同的命令功能，如cp，rm等。必须编程实现，不能直接调用该命令的功能。

上述步骤实现后，为自己的命令增加至少两个选项（option），如ls命令可增加一个-l的选项，查看文件及目录的详细信息。将自定义的命令设置环境变量和权限，以root登录时在任何目录中均可正确执行该命令，同时退出登录再次登录时也可以在任何目录使用该命令。

2. 调试与测试

1) 题目一代码

```
mxy@MateBook14: ~/Shell
echo
echo 'Use one of the following options:'
echo ' P: To display current directory'
echo ' S: To display the name of running file'
echo ' D: To display today' s date and present time'
echo ' L: To see the listing of files in your present working directory'
echo ' W: To see who is logged in'
echo ' Q: To quit this program'
read -n 1 -p 'Enter your option and hit: ' option
echo
echo "Your option is $option"

case ${option} in
    P)
        pwd
        ;;
    S)
        $(basename %0)
        ;;
    D)
        date
        ;;
    L)
        ls -al
        ;;
    W)
        whoami
        ;;
    Q)
        exit
        ;;
    *)
        echo 'Please enter as required!'
        ;;
esac
```

测试:

```
mxy@MateBook14:~/Shell$ ./t1.sh

Use one of the following options:
P: To display current directory
S: To display the name of running file
D: To display today' s date and present time
L: To see the listing of files in your present working directory
W: To see who is logged in
Q: To quit this program
Enter your option and hit: p
Your option is p
/home/mxy/Shell

Enter your option and hit: s
Your option is s
t1.sh

Enter your option and hit: d
Your option is d
Fri Oct 18 17:32:37 CST 2019
```

```
Enter your option and hit: l
Your option is l
total 0
drwxrwxrwx 1 mxy mxy 4096 Oct 18 17:31 .
drwxr-xr-x 1 mxy mxy 4096 Oct 18 17:31 ..
-rwxrwxrwx 1 mxy mxy 603 Oct 18 17:31 t1.sh
```

```
Enter your option and hit: w
Your option is w
mxy
```

```
Enter your option and hit: q
Your option is q
mxy@MateBook14: ~/Shell$
```

2) 题目二

Shell 代码:

```
mxy@MateBook14: ~/Shell
echo
read -p "Input your score: " score
if (( $score >= 0 && $score < 60 ))
then
    echo 'Failed!'
elif (( $score >= 60 && $score < 70 ))
then
    echo 'Passed!'
elif (( $score >= 70 && $score < 80 ))
then
    echo 'Medium!'
elif (( $score >= 80 && $score < 90 ))
then
    echo 'Good!'
elif (( $score >= 90 && $score <= 100 ))
then
    echo 'Excellent!'
else
    echo 'Input error!'
fi
```

测试:

```
mxy@MateBook14: ~/Shell
mxy@MateBook14: ~/Shell$ ./t2.sh
Input your score: 0
Failed!
mxy@MateBook14: ~/Shell$ ./t2.sh
Input your score: 20
Failed!
mxy@MateBook14: ~/Shell$ ./t2.sh
Input your score: 60
Passed!
mxy@MateBook14: ~/Shell$ ./t2.sh
Input your score: 69
Passed!
mxy@MateBook14: ~/Shell$ ./t2.sh
Input your score: 70
Medium!
mxy@MateBook14: ~/Shell$ ./t2.sh
Input your score: 79
Medium!
mxy@MateBook14: ~/Shell$ ./t2.sh
Input your score: 80
Good!
mxy@MateBook14: ~/Shell$ ./t2.sh
Input your score: 89
Good!
mxy@MateBook14: ~/Shell$ ./t2.sh
Input your score: 90
Excellent!
mxy@MateBook14: ~/Shell$ ./t2.sh
Input your score: 100
Excellent!
```

错误分数测试:

```
mxy@MateBook14: ~/Shell
mxy@MateBook14:~/Shell$ ./t2.sh
Input your score: -1
Input error!
mxy@MateBook14:~/Shell$ ./t2.sh
Input your score: 111
Input error!
```

3) 题目三

```
mxy@MateBook14: ~/Shell
#!/bin/bash
function merge()
{
    cat $1 $2 > $3
    cat $3
}

if [ $# -eq 3 ]
then
    if [ ! -e $1 ]
    then
        echo "$1 does not exist, please rerun the program "
    elif [ ! -e $2 -a ! -e $3 ]
    then
        echo "$2 and $3 do not exist"
        cat $1
    elif [ -e $2 -a ! -e $3 ]
    then
        echo "$3 does not exist"
        if [ ! -e mydoc.txt ]
        then
            touch mydoc.txt
        fi
        cat $1 $2 > mydoc.txt
    else
        merge $1 $2 $3
    fi
else
    echo "please input three parameter"
fi

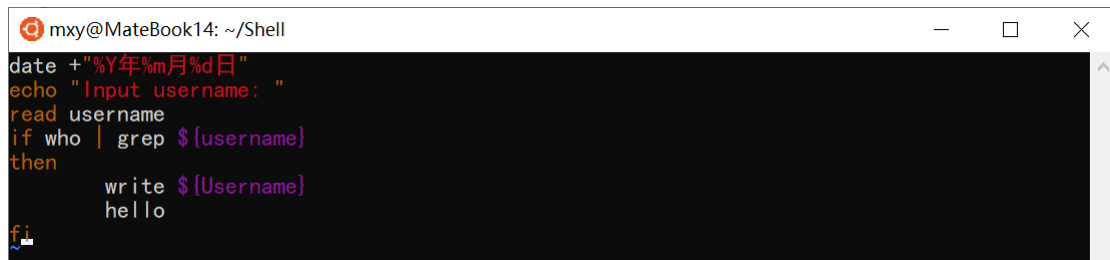
exit 0
```

16, 2-16 All

测试:

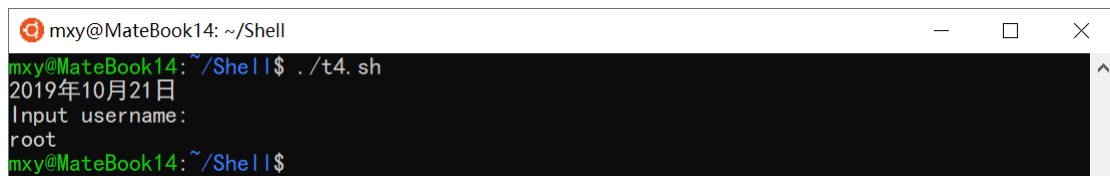
```
mxy@MateBook14: ~/Shell
mxy@MateBook14:~/Shell$ cat 1
mxy
hello
mxy@MateBook14:~/Shell$ cat 2
world
mxy@MateBook14:~/Shell$ cat 3
mxy@MateBook14:~/Shell$ ./t3.sh 1 2 3
mxy
hello
world
mxy@MateBook14:~/Shell$ cat 3
mxy
hello
world
mxy@MateBook14:~/Shell$ _
```


4) 题目四

A terminal window titled 'mxy@MateBook14: ~/Shell' with standard window controls. The terminal contains a shell script with the following content:

```
date +%Y年%m月%d日
echo "Input username: "
read username
if who | grep ${username}
then
    write ${Username}
    hello
fi
```

测试:

A terminal window titled 'mxy@MateBook14: ~/Shell' with standard window controls. The terminal shows the execution of the script: the user runs './t4. sh', the script prints the date '2019年10月21日', prompts for 'Input username:', the user enters 'root', and the script outputs 'root' and 'mxy@MateBook14:~/Shell\$'.

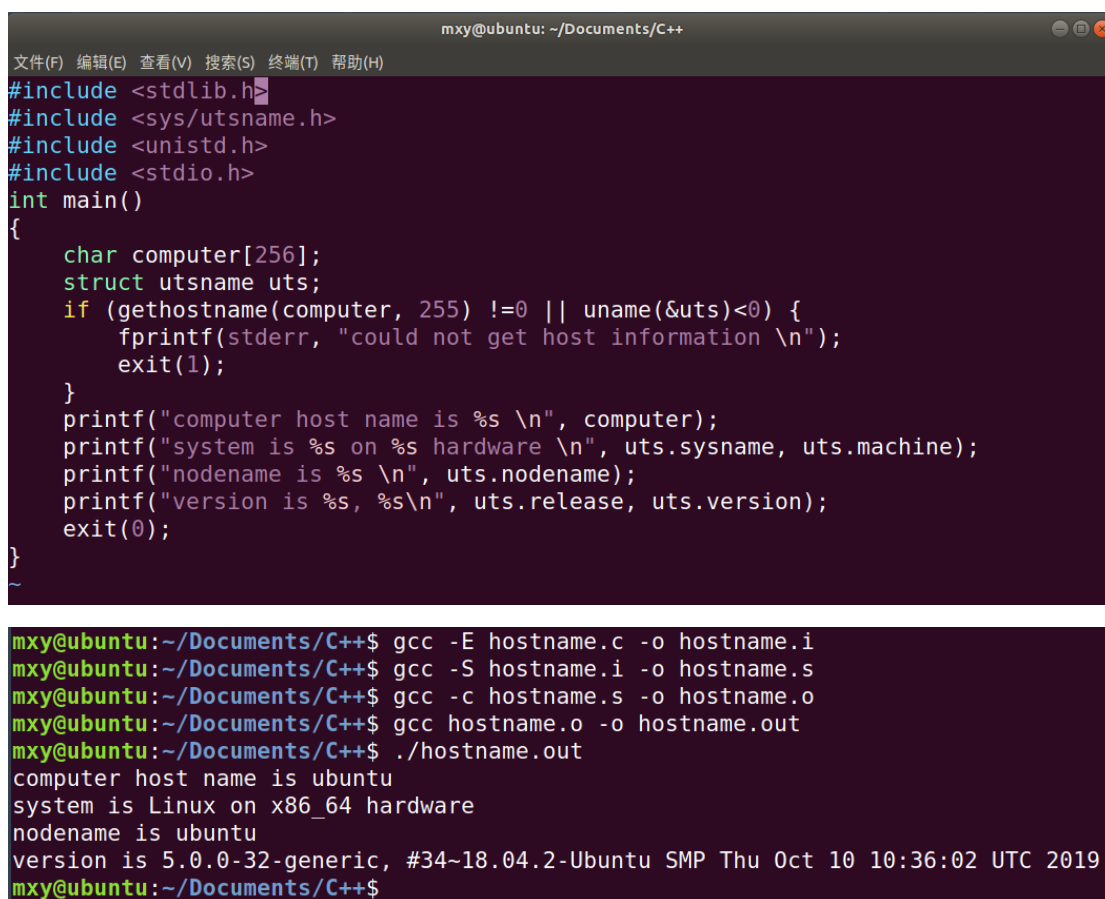
实验四

1. 要求

- 了解 Linux 下 C 语言程序编译和调试的环境。
- 掌握 GCC 的各种编译参数。
- 掌握 makefile 的使用方法。
- 掌握 gdb 调试工具的使用。

2. 调试与测试

1) 题目一



```
mxy@ubuntu: ~/Documents/C++
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
#include <stdlib.h>
#include <sys/utsname.h>
#include <unistd.h>
#include <stdio.h>
int main()
{
    char computer[256];
    struct utsname uts;
    if (gethostname(computer, 255) !=0 || uname(&uts)<0) {
        fprintf(stderr, "could not get host information \n");
        exit(1);
    }
    printf("computer host name is %s \n", computer);
    printf("system is %s on %s hardware \n", uts.sysname, uts.machine);
    printf("nodename is %s \n", uts.nodename);
    printf("version is %s, %s\n", uts.release, uts.version);
    exit(0);
}

mxy@ubuntu:~/Documents/C++$ gcc -E hostname.c -o hostname.i
mxy@ubuntu:~/Documents/C++$ gcc -S hostname.i -o hostname.s
mxy@ubuntu:~/Documents/C++$ gcc -c hostname.s -o hostname.o
mxy@ubuntu:~/Documents/C++$ gcc hostname.o -o hostname.out
mxy@ubuntu:~/Documents/C++$ ./hostname.out
computer host name is ubuntu
system is Linux on x86_64 hardware
nodename is ubuntu
version is 5.0.0-32-generic, #34~18.04.2-Ubuntu SMP Thu Oct 10 10:36:02 UTC 2019
mxy@ubuntu:~/Documents/C++$
```

2) 题目二

程序源码:

```
mxy@ubuntu:~/Documents/C++/mytool$ ll
总用量 20
drwxr-xr-x 3 mxy mxy 4096 10月 24 18:12 ./
drwxr-xr-x 3 mxy mxy 4096 10月 24 00:20 ../
drwxr-xr-x 2 mxy mxy 4096 10月 24 18:10 functions/
-rw-r--r-- 1 mxy mxy 246 10月 24 17:46 main.c
-rw-r--r-- 1 mxy mxy 427 10月 24 18:09 Makefile
mxy@ubuntu:~/Documents/C++/mytool$ ll functions/
总用量 32
drwxr-xr-x 2 mxy mxy 4096 10月 24 18:10 ./
drwxr-xr-x 3 mxy mxy 4096 10月 24 18:12 ../
-rw-r--r-- 1 mxy mxy 134 10月 24 18:10 mytool1.c
-rw-r--r-- 1 mxy mxy 85 10月 24 00:25 mytool1.h
-rw-r--r-- 1 mxy mxy 148 10月 24 18:07 mytool2.c
-rw-r--r-- 1 mxy mxy 99 10月 24 00:26 mytool2.h
-rw-r--r-- 1 mxy mxy 147 10月 24 18:08 mytool3.c
-rw-r--r-- 1 mxy mxy 99 10月 24 00:27 mytool3.h
mxy@ubuntu:~/Documents/C++/mytool$ z
```

```
mxy@ubuntu: ~/Documents/C++/mytool
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
/*main.c*/
#include "functions/mytool1.h"
#include "functions/mytool2.h"
#include "functions/mytool3.h"

int main()
{
    mytool1_print("hello mytool1!");
    mytool2_print("hello mytool2!");
    mytool3_print("hello mytool3!");
    return 0;
}
```

```
#include "mytool1.h"
#include <stdio.h>

void mytool1_print(char *print_str)
{
    printf("This is mytool1 print: %s ", print_str);
}
~
```

```
#ifndef _MYTOOL_1_H
#define _MYTOOL_1_H
void mytool1_print(char *print_str);
#endif
```

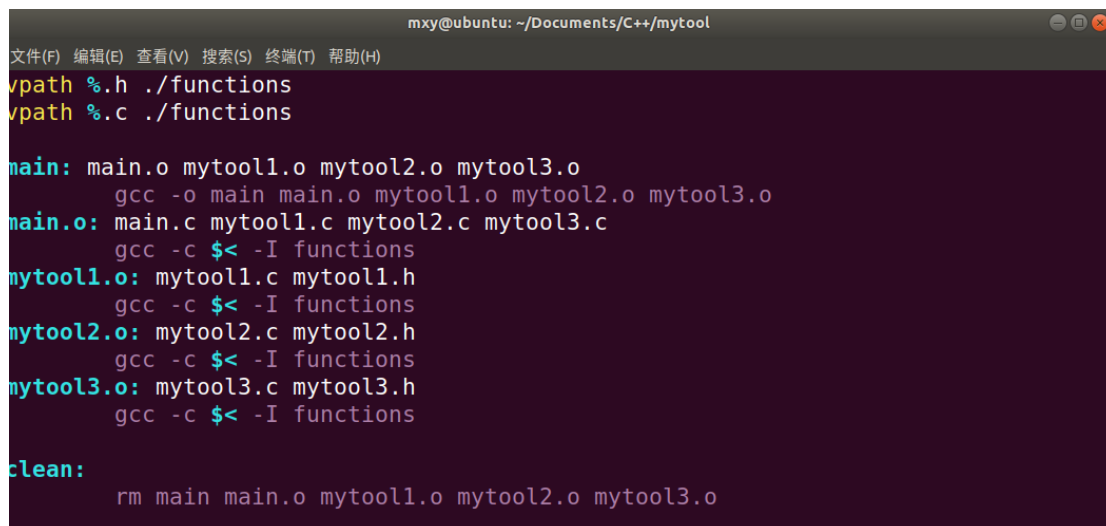
```
/*mytool2.c*/
#include "mytool2.h"
#include <stdio.h>
void mytool2_print(char *print_str)
{
    printf("This is mytool2 print: %s ", print_str);
}
~
```

```
/*mytool2.h*/
#ifndef _MYT00L_2_H
#define _MYT00L_2_H
void mytool2_print(char *print_str);
#endif

/*mytool3.c*/
#include "mytool3.h"
#include <stdio.h>
void mytool3_print(char *print_str)
{
    printf("This is mytool3 print: %s ", print_str);
}

/*mytool3.h*/
#ifndef _MYT00L_3_H
#define _MYT00L_3_H
void mytool3_print(char *print_str);
#endif
```

Makefile 源码:

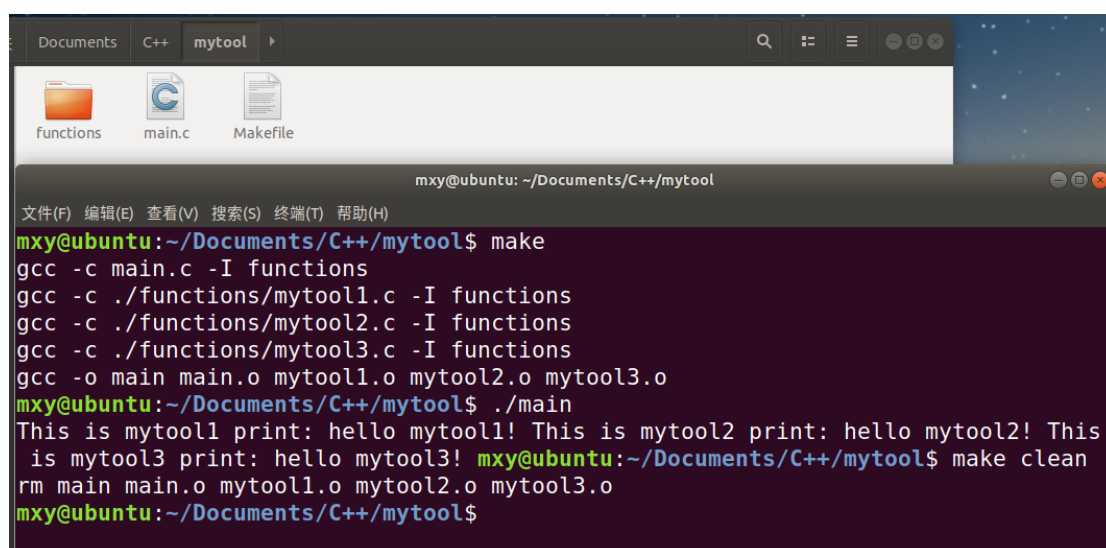


```
mxy@ubuntu: ~/Documents/C++/mytool
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
vpath %.h ./functions
vpath %.c ./functions

main: main.o mytool1.o mytool2.o mytool3.o
    gcc -o main main.o mytool1.o mytool2.o mytool3.o
main.o: main.c mytool1.c mytool2.c mytool3.c
    gcc -c $< -I functions
mytool1.o: mytool1.c mytool1.h
    gcc -c $< -I functions
mytool2.o: mytool2.c mytool2.h
    gcc -c $< -I functions
mytool3.o: mytool3.c mytool3.h
    gcc -c $< -I functions

clean:
    rm main main.o mytool1.o mytool2.o mytool3.o
```

测试:



```
Documents C++ mytool
functions main.c Makefile

mxy@ubuntu: ~/Documents/C++/mytool
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
mxy@ubuntu:~/Documents/C++/mytool$ make
gcc -c main.c -I functions
gcc -c ./functions/mytool1.c -I functions
gcc -c ./functions/mytool2.c -I functions
gcc -c ./functions/mytool3.c -I functions
gcc -o main main.o mytool1.o mytool2.o mytool3.o
mxy@ubuntu:~/Documents/C++/mytool$ ./main
This is mytool1 print: hello mytool1! This is mytool2 print: hello mytool2! This
is mytool3 print: hello mytool3! mxy@ubuntu:~/Documents/C++/mytool$ make clean
rm main main.o mytool1.o mytool2.o mytool3.o
mxy@ubuntu:~/Documents/C++/mytool$
```

课程总结

通过对 Linux 的学习,首先了解到了在我们熟知的 Windows 和 MacOS 之外还有一系列可以统称为 Linux 的操作系统,而 Linux 的各种发行版本又是数不胜数。Linux 相对于我们常用的 Windows,其简洁性、安全性以及稳定性使得 Linux 系列的操作系统在开发编程中占据了重要的角色,作为计算机专业的一名学生,也可见其重要性。

一学期的学习,我学会了 Linux 操作系统的一系列命令,学会了编写 shell 脚本,也学会了在 Linux 操作系统上使用 vim、gcc 等工具进行编程开发。学会了很多,但显然也还有许多的内容没有学习到,所以,今后的学习中,也还要努力学习更多的 Linux 知识,增强自身技能。