

Lab1-Shell 编程

张配天-2018202180

2020 年 4 月 9 日

目录

1	实验目的	2
2	实验方法	2
3	程序清单	2
3.1	C 程序运行结果	3
3.2	Shell 脚本代码	3
4	实验结果	5
5	问题分析	6
5.1	出现的问题	6
5.2	解决办法	6

1 实验目的

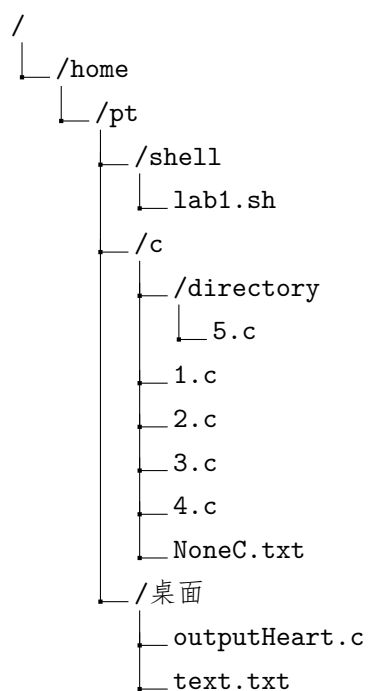
编写一个 shell 脚本程序，功能是取出指定目录中的所有 C 语言源程序文件，并对其编译和执行。

2 实验方法

注意到 ubuntu 的特性，在/home/目录下 vim 的新文件也不是全权限，因此先自己实现了一个小脚本 myvi 保存在/usr/bin/下，用来创造全权限的文件并转至 vim 编辑展示代码如下：

```
1  fileName="$1"
2  echo "FILE:${fileName}"
3  touch $fileName
4  chmod 777 $fileName
5  vim $fileName
```

3 程序清单



其中

- /桌面, /shell , /c , /directory 均为目录
- /shell 目录下保存着可执行的 Shell 脚本 lab1.sh
- /c 目录下保存着 4 个可编译.c 文件, 还保存着/directory 目录, 用来完成对遇到目录的特殊情况处理
- /c 目录下另有一个非 C 语言写成的 NoneC.txt 文件, 用来完成对编译错误的特殊情况处理
- /directory 目录下保存着最后一个可编译的.c 文件
- /桌面目录下保存着可编译的 outputHeart.c 和不可编译的 text.txt 文件, 将来可作为第二个参数传入脚本

3.1 C 程序运行结果

- 1.c 输出为“张”
- 2.c 输出为“配”
- 3.c 输出为“天”
- 4.c 输出为“晚安”
- 5.c 输出为由 * 组成的 V 字样
- outputHeart.c 输出心形图案

3.2 Shell 脚本代码

```
1 #定义编译链接的函数,将目录作为参数传递进入函数
2 compile_run_C(){
3     #遍历传入目录下所有文件和目录
4     for file in $(ls $1)
5     do
6         #完整绝对路径
7         path="$1$file"
8         #判断是否是普通文件,如果是则进行编译链接
9         if [ -f $path ];then
10             exe=${file%.*}
11             gcc -W -o $exe $path
12             result=$?
13             #获取gcc返回值,若为0则代表编译成功
14             if [ $result -eq 0 ];then
15                 ./$exe
16                 rm $exe
17             #否则输出错误,即非c程序
18             else
```

```
19         echo "$path is not a C program"
20     fi
21     #若为目录，输出信息，同时递归继续进行该函数
22     else
23         echo $path is a Directory
24         compile_run_C "$path/"
25     fi
26 done
27 return 0
28 }
29 #如果未传入参数，则输出需要更多参数
30 if [ $# -eq 0 ]
31 then
32     echo "This command requires more arguments"
33     fileDiction=$(pwd)
34     echo "FILE DIRECTION: $fileDiction"
35 fi
36 #获取全部传入参数的数组，针对每一个目录进行循环
37 for DIR in $@
38 do
39     #如果传入非目录，则输出错误信息
40     if [ ! -d "$DIR" ];then
41         echo "Sorry,directory $DIR does not exist"
42         exit 1
43     #否则进行编译链接
44     else
45         compile_run_C $DIR
46     fi
47 done
```

4 实验结果

```
pt@pt-virtual-machine:~/shell$ ./lab1.sh ~/c/ ~/桌面 /  
/home/pt/c/  
张  
配  
天  
晚安!  
/home/pt/c/directory is a Directory  
* *  
* *  
* *  
* *  
* *  
* *  
*/usr/bin/ld:/home/pt/c/NoneC.txt: file format not recognized; treating as linker script  
/usr/bin/ld:/home/pt/c/NoneC.txt:0: syntax error  
collect2: error: ld returned 1 exit status  
/home/pt/c/NoneC.txt is not a C program  
/home/pt/桌面/  
  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

图 1: ubuntu 终端运行结果

可见：

- 将/home/pt/c/和/home/pt/桌面/作为参数传入脚本，运行
- 成功运行 4 个 c 程序并输出
- 脚本检测到/home/pt/c/directory 不是程序
- 进入/home/pt/c/directory 后成功运行 5.c 并输出
- gcc 编译报错，因为/home/pt/c/NoneC.txt 不是 c 程序
- 脚本输出信息/home/pt/c/NoneC.txt is not a C program
- 进入第二个传入目录/home/pt/桌面/
- 成功运行 outputHeart.c 并输出
- gcc 编译报错，因为/home/pt/桌面/text.txt 不是 c 程序
- 脚本输出信息/home/pt/桌面/text.txt is not a C program

5 问题分析

5.1 出现的问题

- i. 对 shell 语言不熟悉导致各类语法错误, 尤其是 if else 语句中忘记加入 then、fi
- ii. 传入参数时 `$*` 和 `$@` 的区别?
- iii. 双引号什么时候必须加?
- iv. 如何禁止 gcc 内部输出编译错误的信息?

5.2 解决办法

- i. 在菜鸟教程上自学 shell 语法并练习
- ii. 目前认知是 `$*` 将参数作为一个字符串传入, `$@` 将参数作为数组传入
- iii. 暂时没有发现区别, 似乎 shell 语言默认类型就是字符串?
- iv. 尚不明确