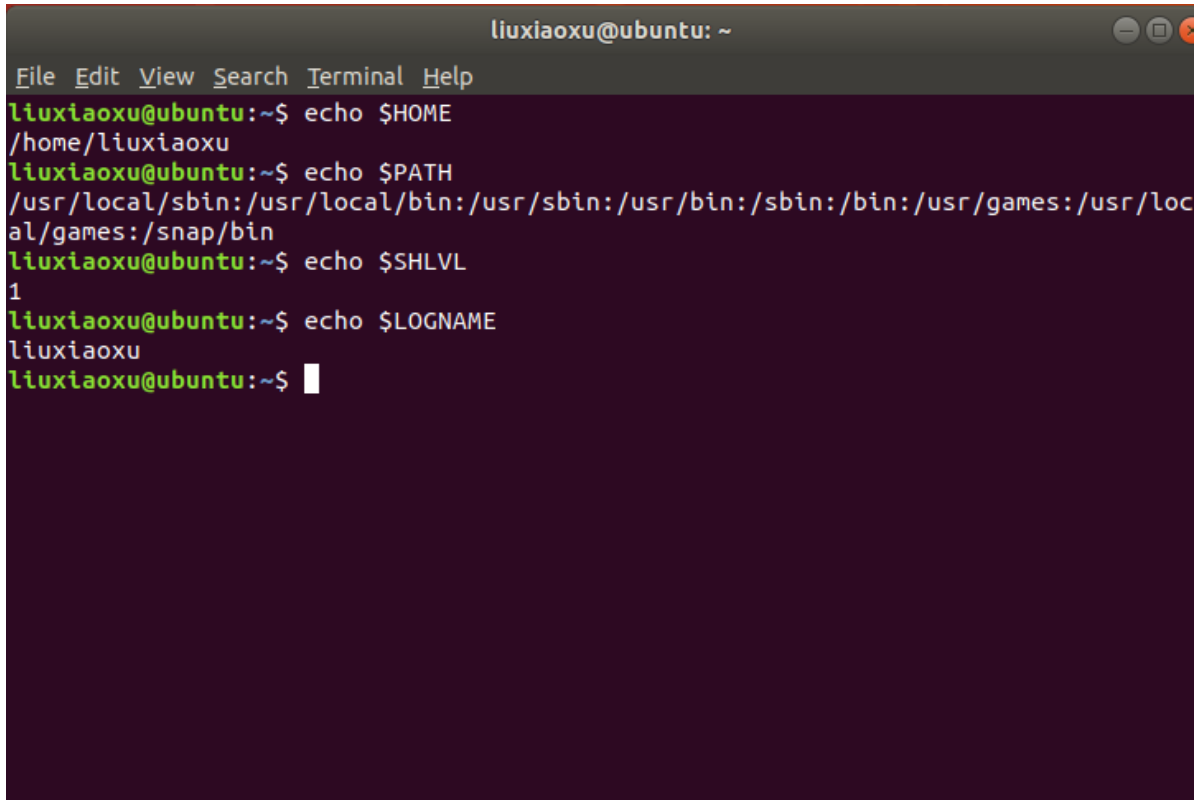


201250123 刘晓旭 Linux程序设计第二次作业

一、用命令行打印HOME、PATH、SHLVL、LOGNAME变量的值

命令行

A screenshot of a Linux terminal window titled 'liuxiaoxu@ubuntu: ~'. The terminal has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The user 'liuxiaoxu' is at the 'ubuntu' machine in the home directory. They execute four 'echo' commands to display the values of environment variables: \$HOME, \$PATH, \$SHLVL, and \$LOGNAME. The output shows the home directory, the PATH variable with multiple directories, the shell level '1', and the username 'liuxiaoxu'.

```
liuxiaoxu@ubuntu: ~  
File Edit View Search Terminal Help  
liuxiaoxu@ubuntu:~$ echo $HOME  
/home/liuxiaoxu  
liuxiaoxu@ubuntu:~$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin  
liuxiaoxu@ubuntu:~$ echo $SHLVL  
1  
liuxiaoxu@ubuntu:~$ echo $LOGNAME  
liuxiaoxu  
liuxiaoxu@ubuntu:~$
```

Shell脚本

Open

print.sh
~/

Save

```
#!/bin.sh
echo $HOME
echo $PATH
echo $SHLVL
echo $LOGNAME
```

Loading file "/home/liuxiaoxu/print.sh"...shTab Width: 8Ln 6, Col 1INS

print.sh Properties

Basic

Permissions

Open With

Owner:

Me

Access:

Read and write

Group:

liuxiaoxu

Access:

Read and write

Others

Access:

Read-only

Execute:

☒ Allow executing file as program

Security context:

unknown

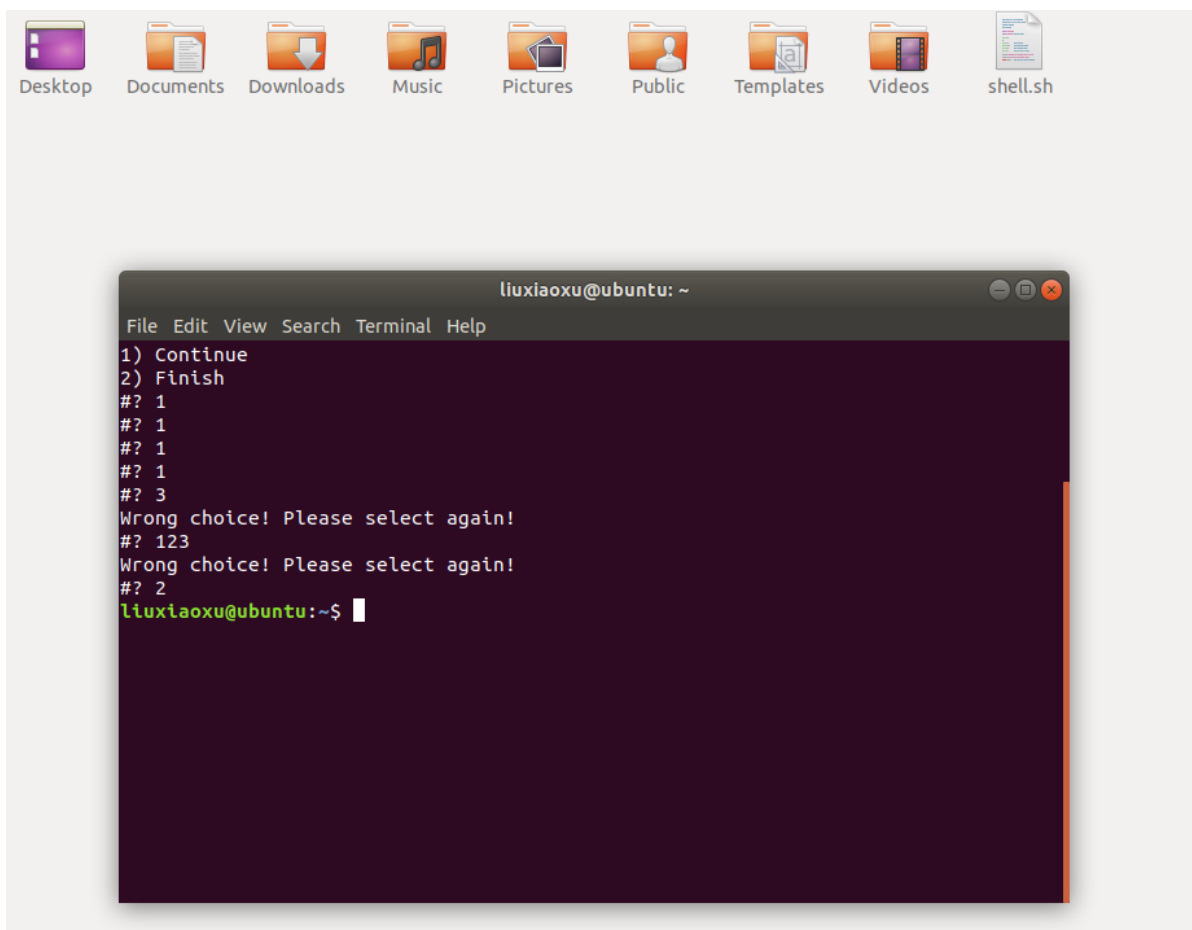
```
liuxiaoxu@ubuntu: ~  
File Edit View Search Terminal Help  
liuxiaoxu@ubuntu:~$ sudo bash print.sh  
[sudo] password for liuxiaoxu:  
/home/liuxiaoxu  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin  
1  
root  
liuxiaoxu@ubuntu:~$
```

二、请用中文解释Shell脚本程序，并说明运行结果

解释

```
# #!/bin/sh指定了脚本解释器，为bash shell。  
#!/bin/sh  
# clear命令清空了屏幕，以便显示菜单。  
clear  
# select是一种显示菜单的循环应用，会将Continue和Finish自动用1)和2)列出，并且能够自动读入选择的值赋给下面的item变量  
select item in Continue Finish  
do  
# 输入1继续，输入2退出，输入其它显示输入错误。  
case "$item" in  
    Continue) ;;  
    Finish) break ;;  
    *) echo "wrong choice! Please select again!" ;;  
esac  
# 循环结束，脚本执行完毕。  
done
```

运行结果



三、阅读下列Makefile并用中文说明其含义。

```
=====Makefile1=====
#引入变量TOP: 当前目录
export Top:=${shell pwd}
#引入变量SRC: 源代码目录
export Src:=$(Top)/src/
#引入变量INCLUDE: 头文件目录
export Include:=$(Top)/include/
#引入变量BUILD: 构建目录
export Build:=$(Top)/build/
#all目标 @表示不回显 $(MAKE)表示执行指定位置的Makefile -C $(Src)表示执行$(Src)中的
Makefile
all:
    @$(MAKE) -C $(Src)
#install目标 表示将$(Build)/test复制到$(Top)目录中。
install:
    @cp $(Build)/test $(Top)
#clean目标 表示删除$(Build)和$(Top)/test目录。
clean:
    @-rm -rf $(Build) $(Top)/test
===== Makefile2=====
#main.o和test4.o是执行目标all的先决条件
all:main.o test4.o
    @mkdir -p $(Build) #递归创建$(Build)目录
    @mv *.o $(Build) #将.o文件移动到刚才的$(Build)目录
    $(MAKE) -C $(Src)/dir1 #对$(Src)/dir1和$(Src)/dir2执行make指令
    $(MAKE) -C $(Src)/dir2
```

#使用环境变量中的编译器对\$(Build)/*.o \$(Build)/dir1/*.o \$(Build)/dir2/*.o的所有文件进行静态链接，生成的可执行文件存储于\$(Build)/test。

```
$(CC) -o $(Build)/test $(Build)/*.o $(Build)/dir1/*.o $(Build)/dir2/*.o
```

#main.o依赖于\$(Include)/func.h文件

```
main.o : $(Include)/func.h
```

```
$(CC) -c main.c -I$(Include) #将main.c进行编译但是不链接。并且指定头文件路径为$(Include)。
```