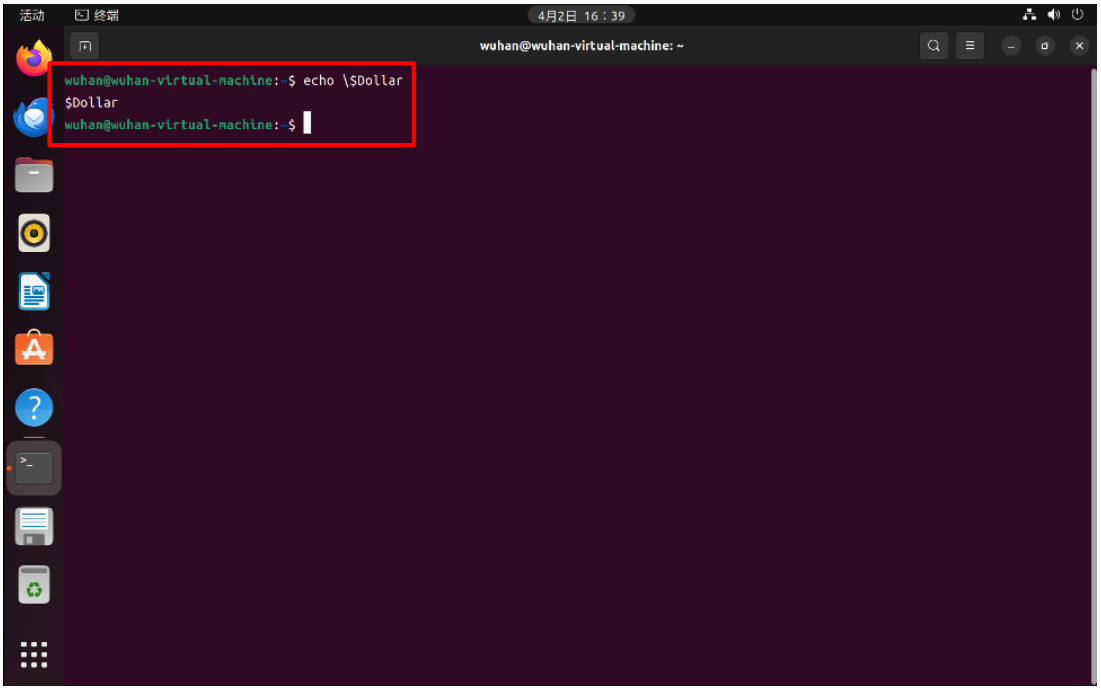
姓名：吴涵　　学号：202101000720　　成绩：

实验 13 批处理操作接口 3：引用与命令替换

1、输出字符串“$Dollar”

命令：echo \$Dollar

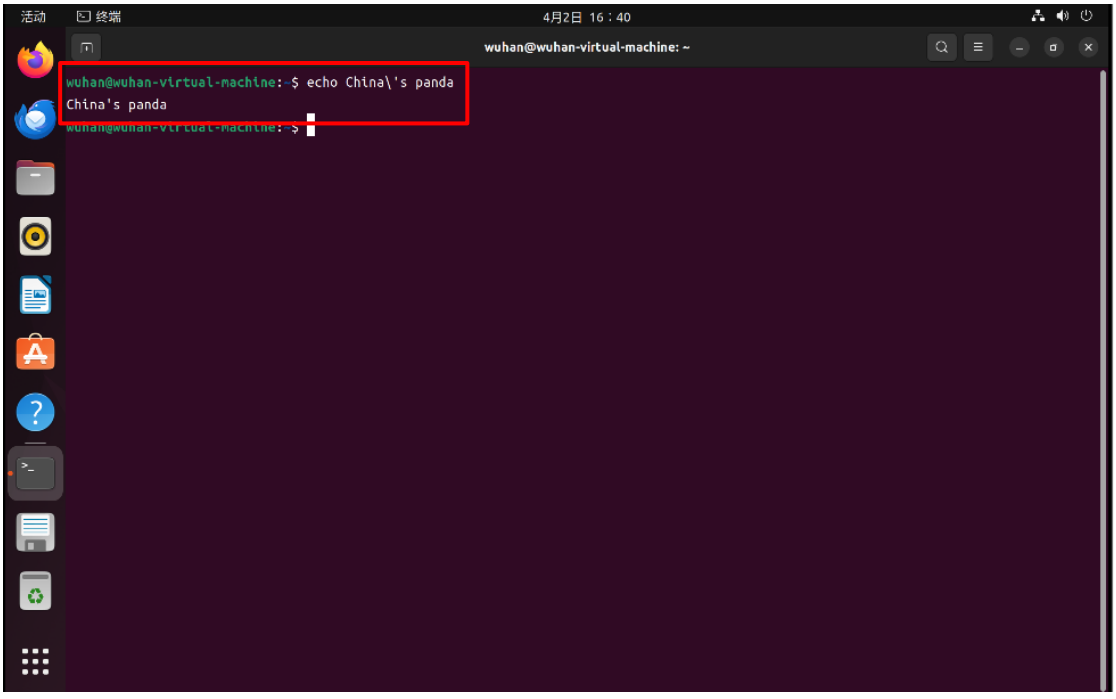
结果：



2、输出字符串“China's panda”

命令：echo China\'s panda

结果：



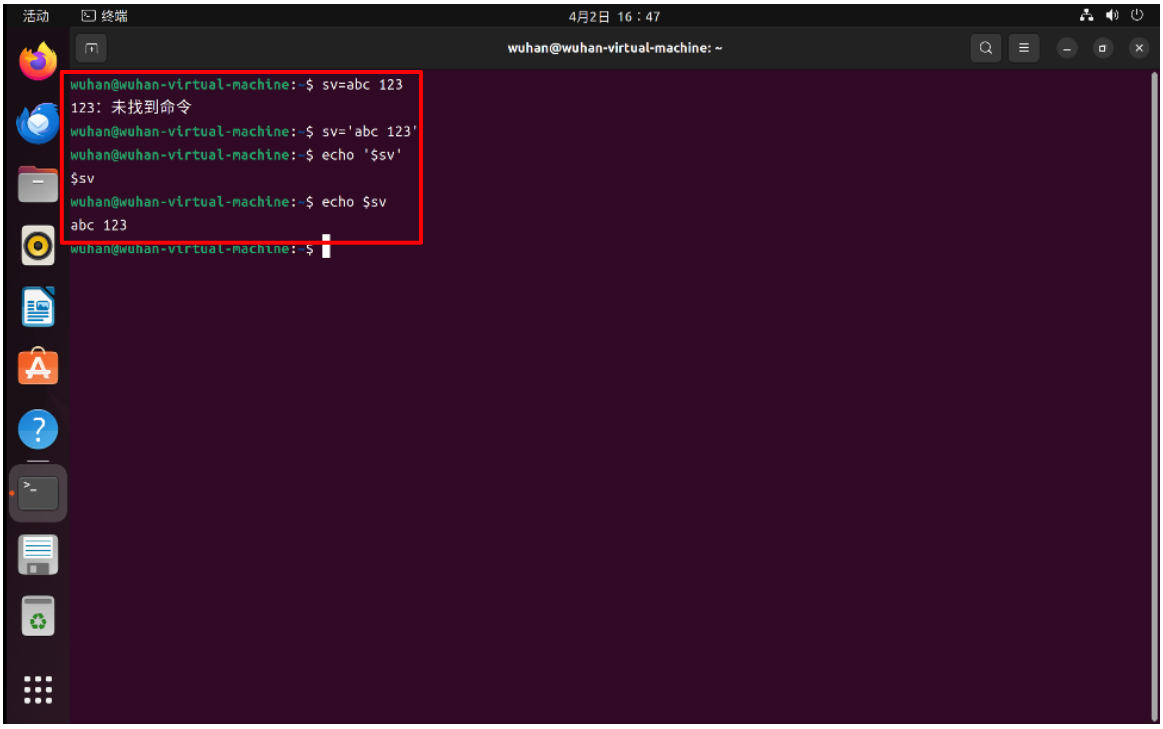
3、比较对错

sv=abc 123

sv='abc 123'

echo '$sv'

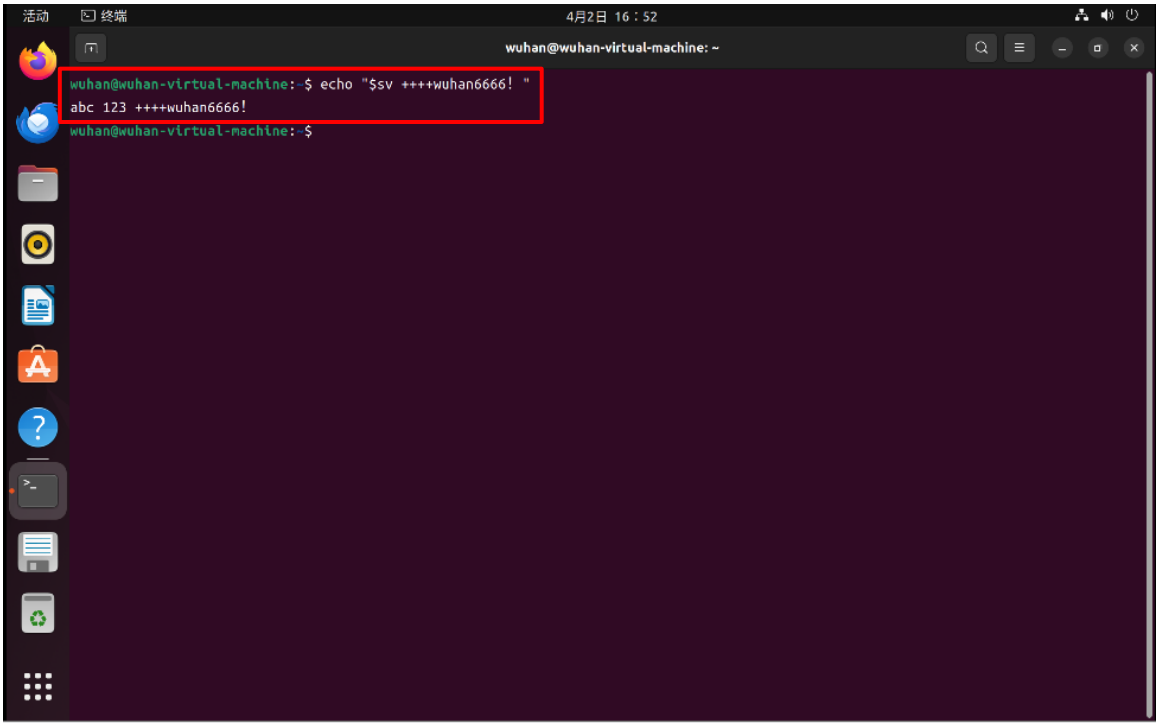
结果：



4、引用

命令：echo "$sv ++++加上另一些字符！"

结果：



5、命令比较

命令："ls"\

命令：'ls'

结果：

```
wuhan@wuhan-virtual-machine:~$ "ls"
c  Desktop  Documents  Downloads  Music  Pictures  Public  shel  snap  Templates  Videos
wuhan@wuhan-virtual-machine:~$ 'ls'
c  Desktop  Documents  Downloads  Music  Pictures  Public  shel  snap  Templates  Videos
```

命令：ab='ls'

ab

命令：$ab

命令："$ab"

结果：

```
wuhan@wuhan-virtual-machine:~$ ab='ls'
wuhan@wuhan-virtual-machine:~$ ab
找不到命令 “ab”，但可以通过以下软件包安装它：
sudo apt install apache2-utils
wuhan@wuhan-virtual-machine:~$ $ab
c  Desktop  Documents  Downloads  Music  Pictures  Public  shel  snap  Templates  Videos
wuhan@wuhan-virtual-machine:~$ "$ab"
c  Desktop  Documents  Downloads  Music  Pictures  Public  shel  snap  Templates  Videos
```

命令：echo "This is ${ab}"

命令：echo "This is $(ab)"

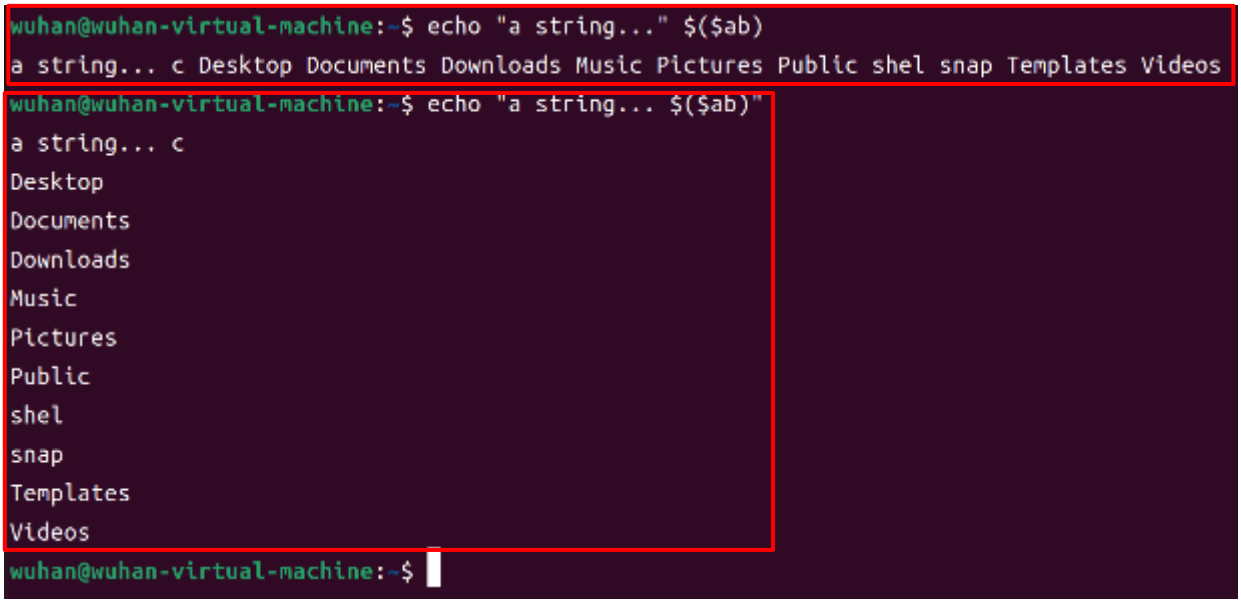
结果：

```
wuhan@wuhan-virtual-machine:~$ echo "This is ${ab}"
This is ls
wuhan@wuhan-virtual-machine:~$ echo "This is $(ab)"
找不到命令 “ab”，但可以通过以下软件包安装它：
sudo apt install apache2-utils
This is
```

命令：echo "a string..." $($ab)

命令：echo "a string... $($ab)"

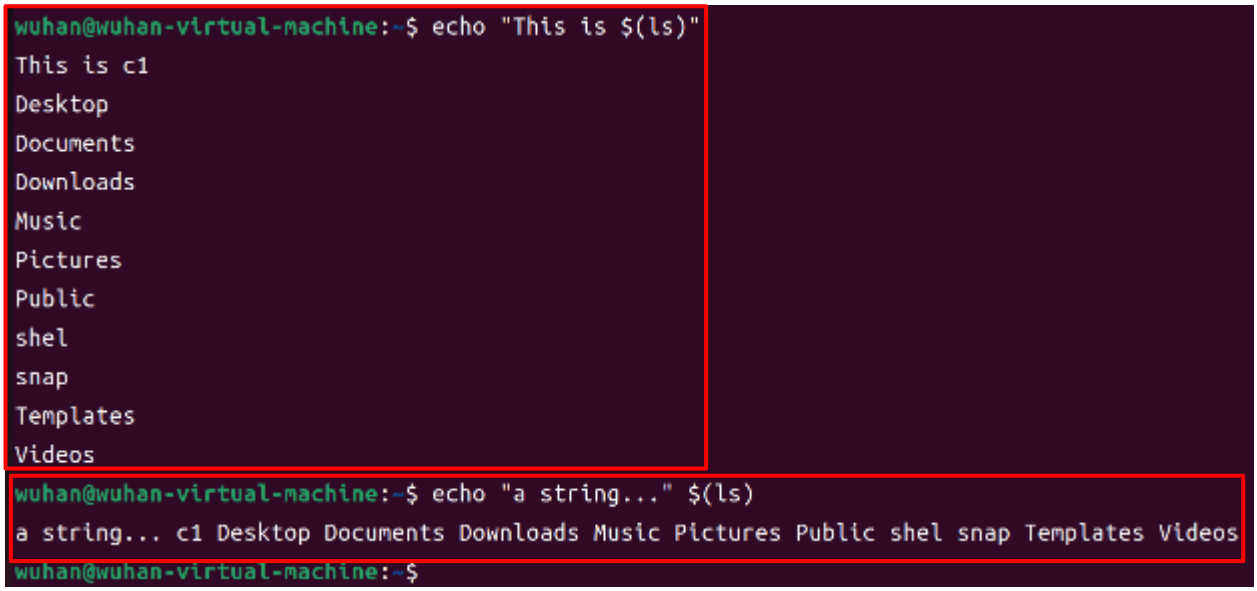
结果：



命令：echo "This is $(ls)"

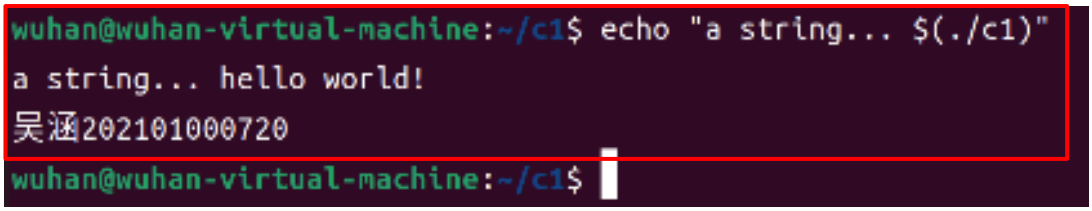
命令：echo "a string..." $(ls)

结果：



命令：echo "a string... $(./c1)"

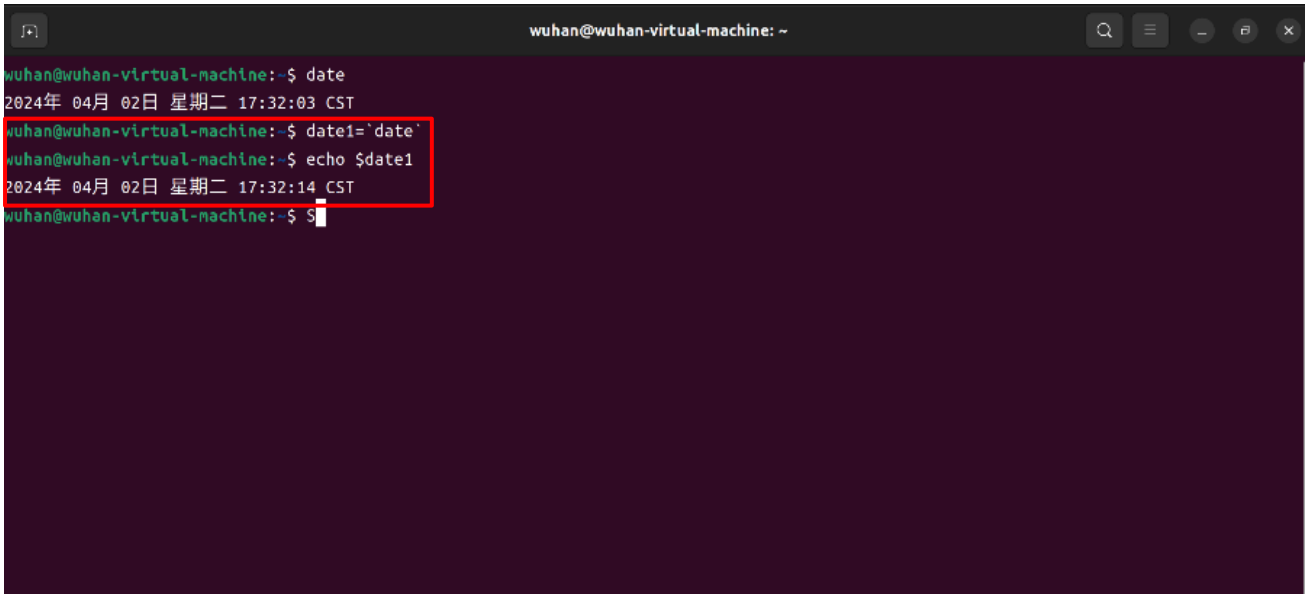
结果：



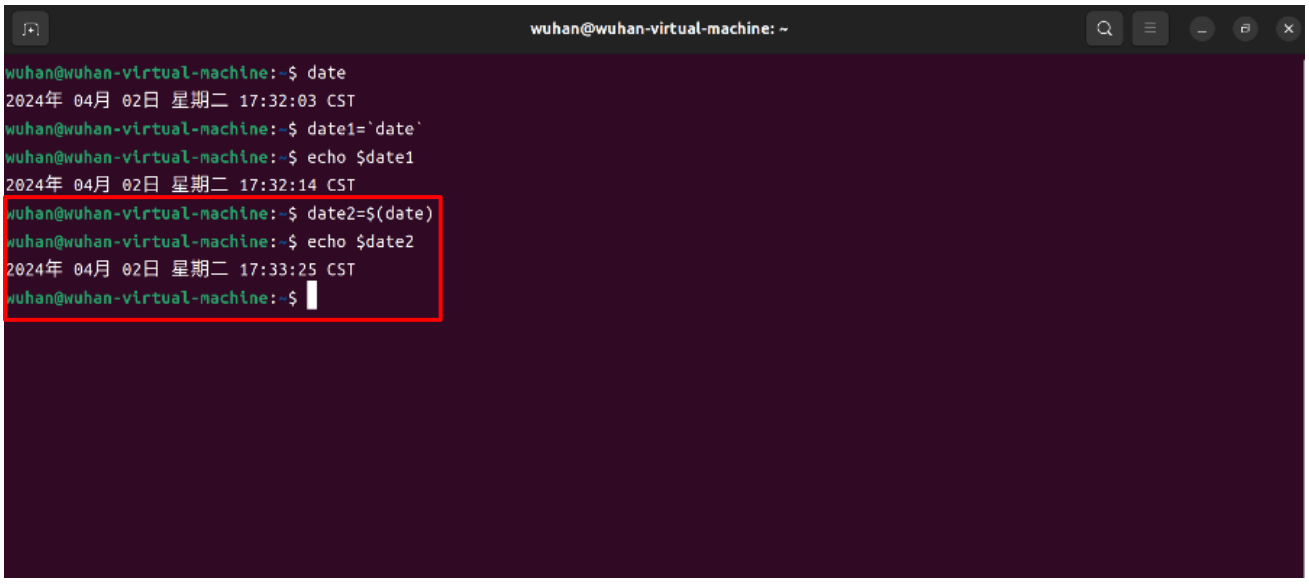
命令：echo 'It'\'' s a dog'
命令：echo "It's a dog"
结果：

```
wuhan@wuhan-virtual-machine:~/c1$ echo 'It'\'' s a dog'
It' s a dog
wuhan@wuhan-virtual-machine:~/c1$ echo "It's a dog"
It's a dog
wuhan@wuhan-virtual-machine:~/c1$
```

6、将命令 date 执行结果赋予变量 date1
命令：date1=’date’
结果：



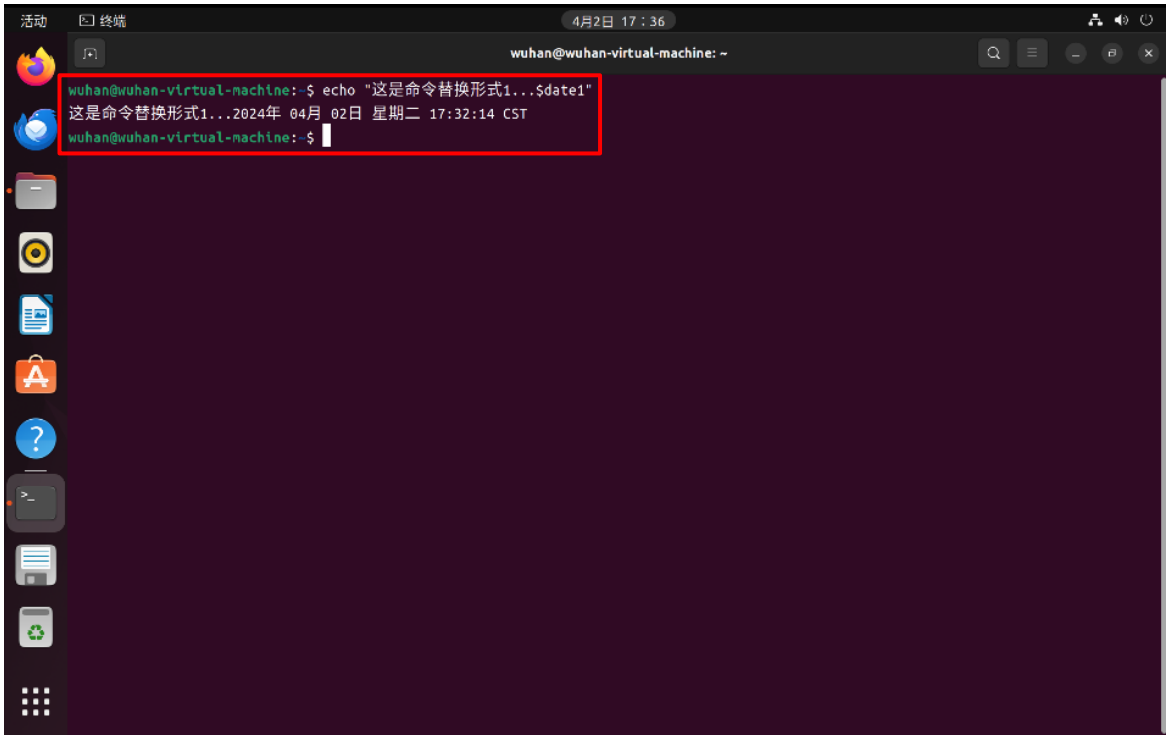
7、将命令 date 执行结果赋予变量 date2
命令：
结果：



8、命令替换

命令：echo "这是命令替换形式 1...$date1"

结果：



9、命令替换

命令：ls1=`ls -l`

echo $ls1

结果：

```
wuhan@wuhan-virtual-machine:~$ ls1=`ls -l`
wuhan@wuhan-virtual-machine:~$ echo $ls1
总计 44 drwxrwxr-x 2 wuhan wuhan 4096 3月 30 21:51 c1 drwxr-xr-x 2 wuhan wuhan 4096 3月 26 17:49 Desktop drwxr-xr-x 2 wuhan wuhan 40
96 3月 26 17:49 Documents drwxr-xr-x 2 wuhan wuhan 4096 3月 26 17:49 Downloads drwxr-xr-x 2 wuhan wuhan 4096 3月 26 17:49 Music drwx
r-xr-x 2 wuhan wuhan 4096 3月 26 17:49 Pictures drwxr-xr-x 2 wuhan wuhan 4096 3月 26 17:49 Public drwxrwxr-x 2 wuhan wuhan 4096 3月
30 22:29 shel drwx------ 4 wuhan wuhan 4096 4月 2 16:32 snap drwxr-xr-x 2 wuhan wuhan 4096 3月 26 17:49 Templates drwxr-xr-x 2 wuhan
 wuhan 4096 3月 26 17:49 Videos
wuhan@wuhan-virtual-machine:~$
```

命令：echo "$ls1"

结果：

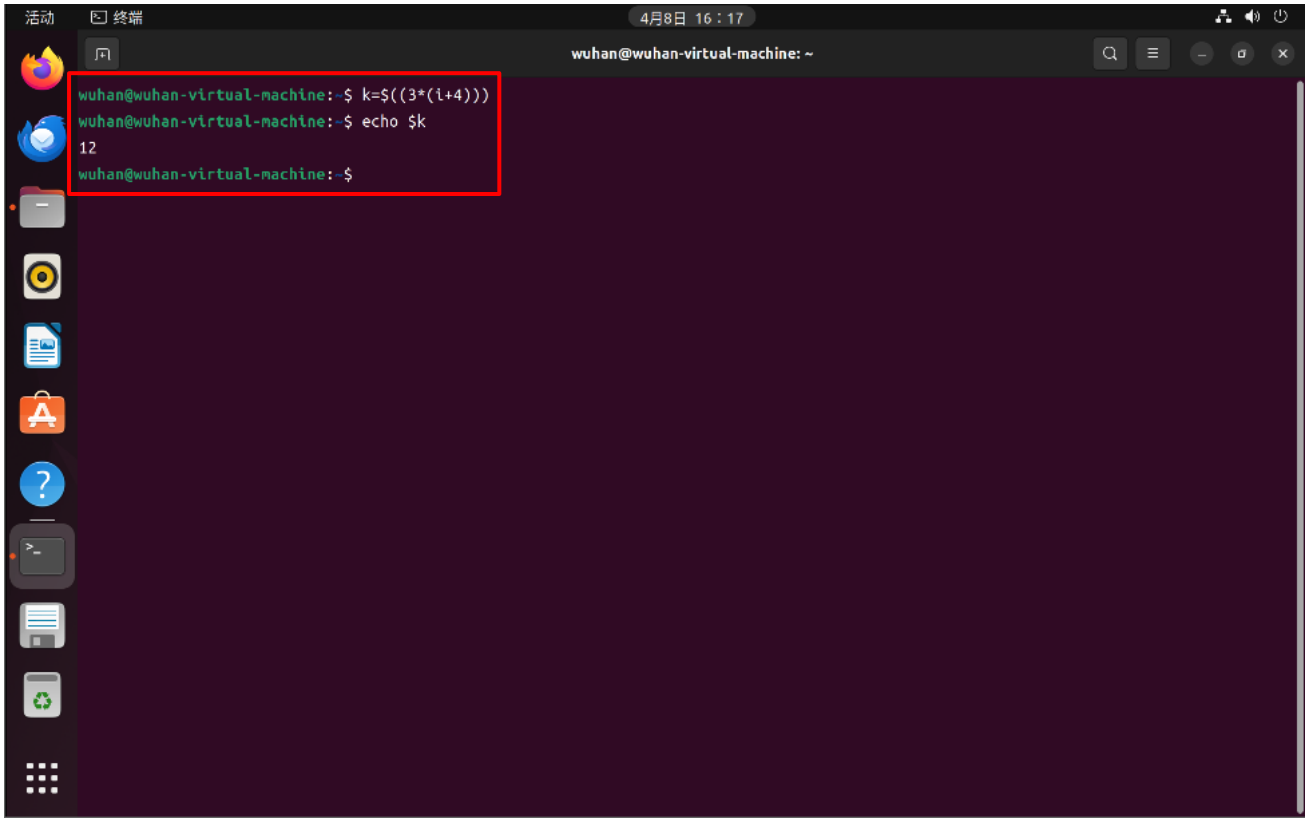
```
wuhan@wuhan-virtual-machine:~$ echo "$ls1"
总计 44
drwxrwxr-x 2 wuhan wuhan 4096  3月 30 21:51 c1
drwxr-xr-x 2 wuhan wuhan 4096  3月 26 17:49 Desktop
drwxr-xr-x 2 wuhan wuhan 4096  3月 26 17:49 Documents
drwxr-xr-x 2 wuhan wuhan 4096  3月 26 17:49 Downloads
drwxr-xr-x 2 wuhan wuhan 4096  3月 26 17:49 Music
drwxr-xr-x 2 wuhan wuhan 4096  3月 26 17:49 Pictures
drwxr-xr-x 2 wuhan wuhan 4096  3月 26 17:49 Public
drwxrwxr-x 2 wuhan wuhan 4096  3月 30 22:29 shel
drwx------ 4 wuhan wuhan 4096  4月  2 16:32 snap
drwxr-xr-x 2 wuhan wuhan 4096  3月 26 17:49 Templates
drwxr-xr-x 2 wuhan wuhan 4096  3月 26 17:49 Videos
wuhan@wuhan-virtual-machine:~$
```

10、命令替换

命令：k=$((3*(i+4)))

echo $k

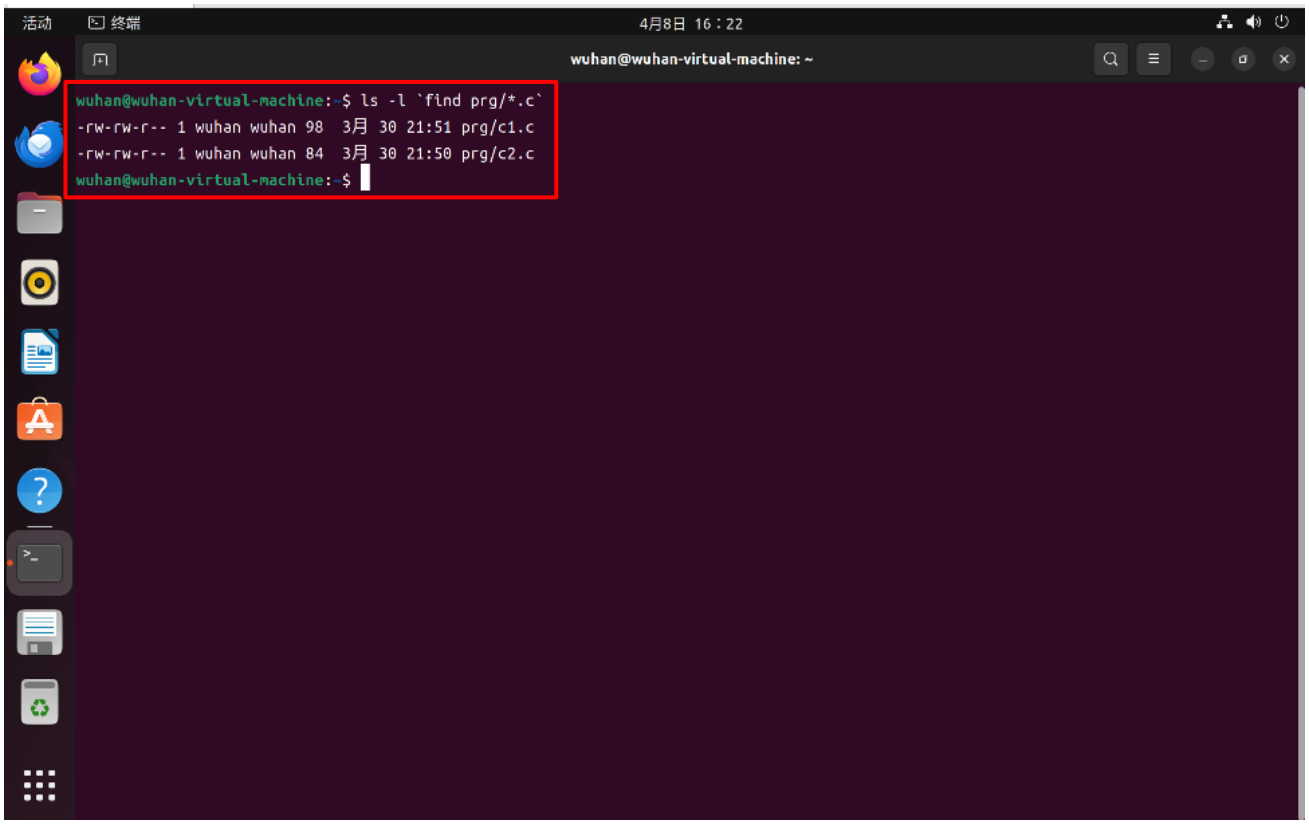
结果：



11、命令替换

命令：ls -l `find prg/*.c`

结果：



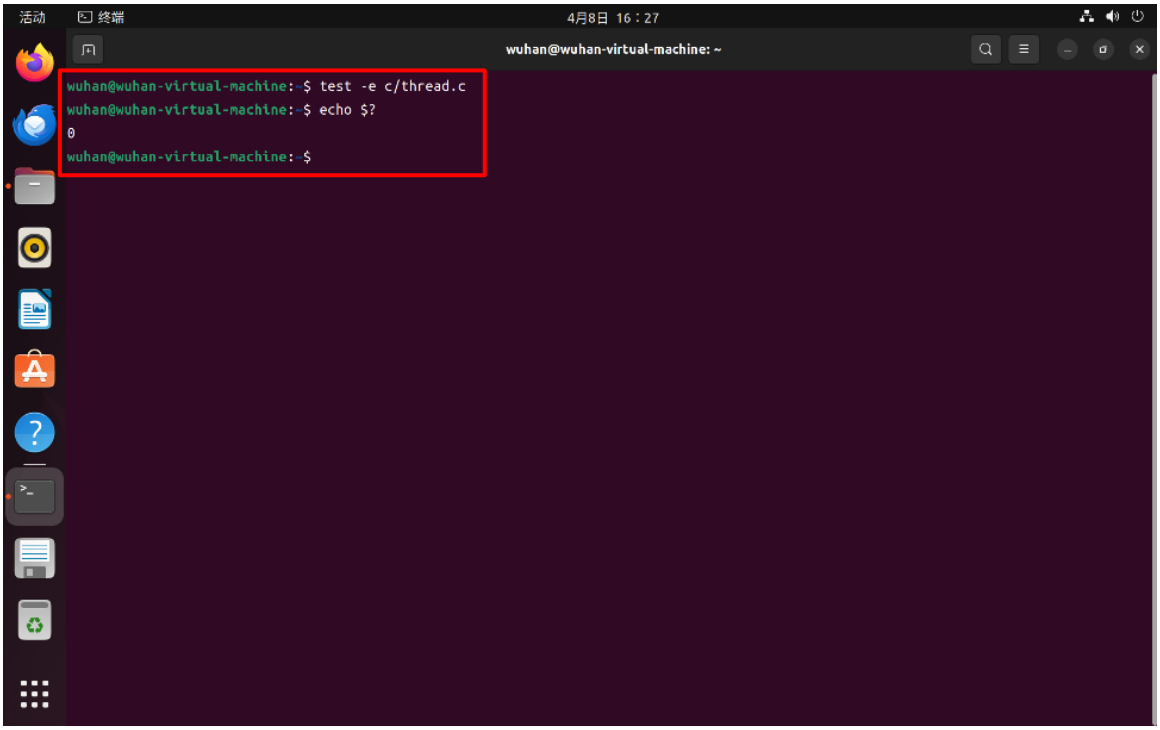
实验 14 批处理操作接口 4：测试、if 判断

1、测试文件 c/thread.c 是否存在

命令：test -e c/thread.c

echo $?

结果：



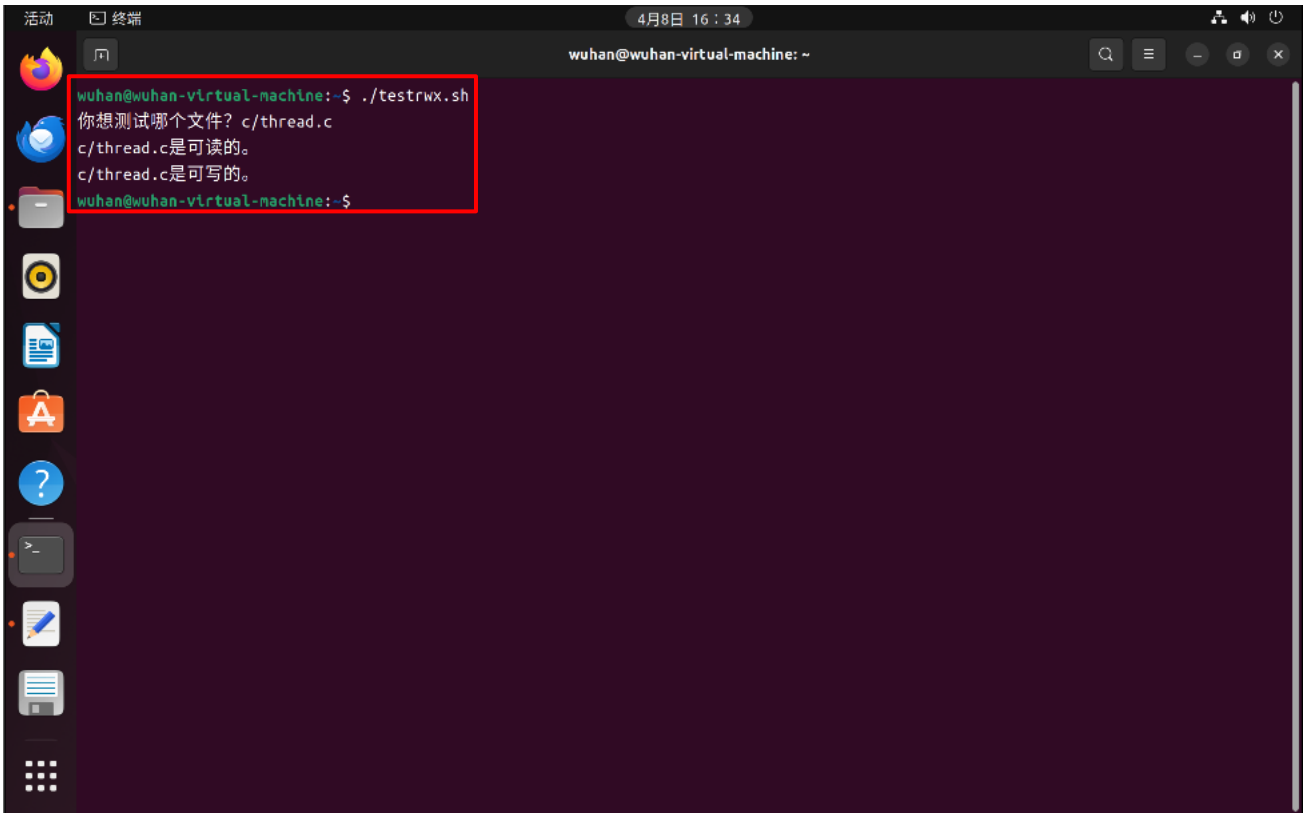
2、编写 Shell 脚本文件 testrwx.sh，对文件拥有的属性进行判断

脚本：

```
#!/bin/bash
read -p "你想测试哪个文件？" filename
if [ ! -e "$filename" ]; then
    echo "这个文件不存在。"
    #exit 1
fi
if [ -r "$filename" ]; then
    echo "$filename 是可读的。"
    #exit 1
fi
if [ -w "$filename" ]; then
    echo "$filename 是可写的。"
    #exit 1
fi
if [ -x "$filename" ]; then
    echo "$filename 是可执行的。"
    #exit 1
fi
```

命令：. ./testrwx.sh

结果：



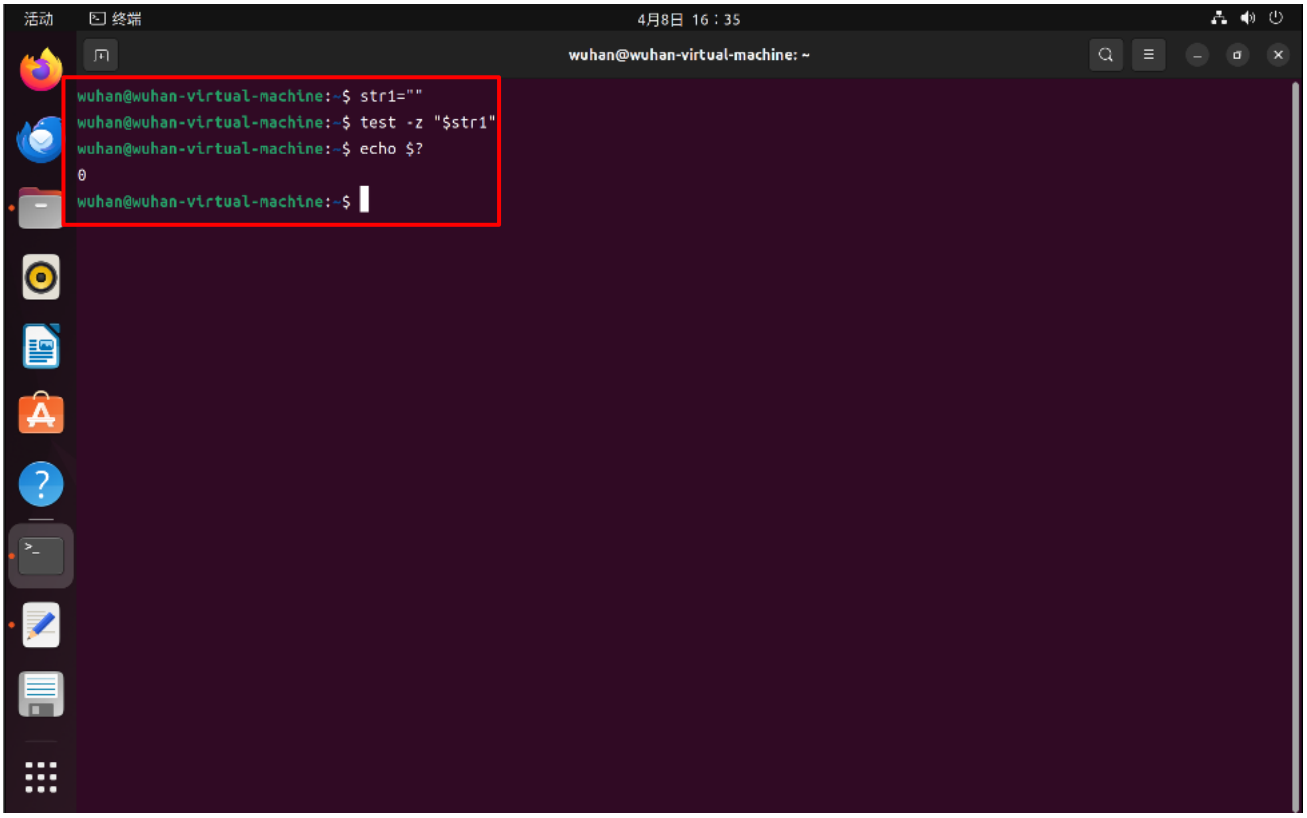
3、测试字符串 str1 是否为空，为空则返回 0

命令：str1=""

test -z "$str1"

echo $?

结果：

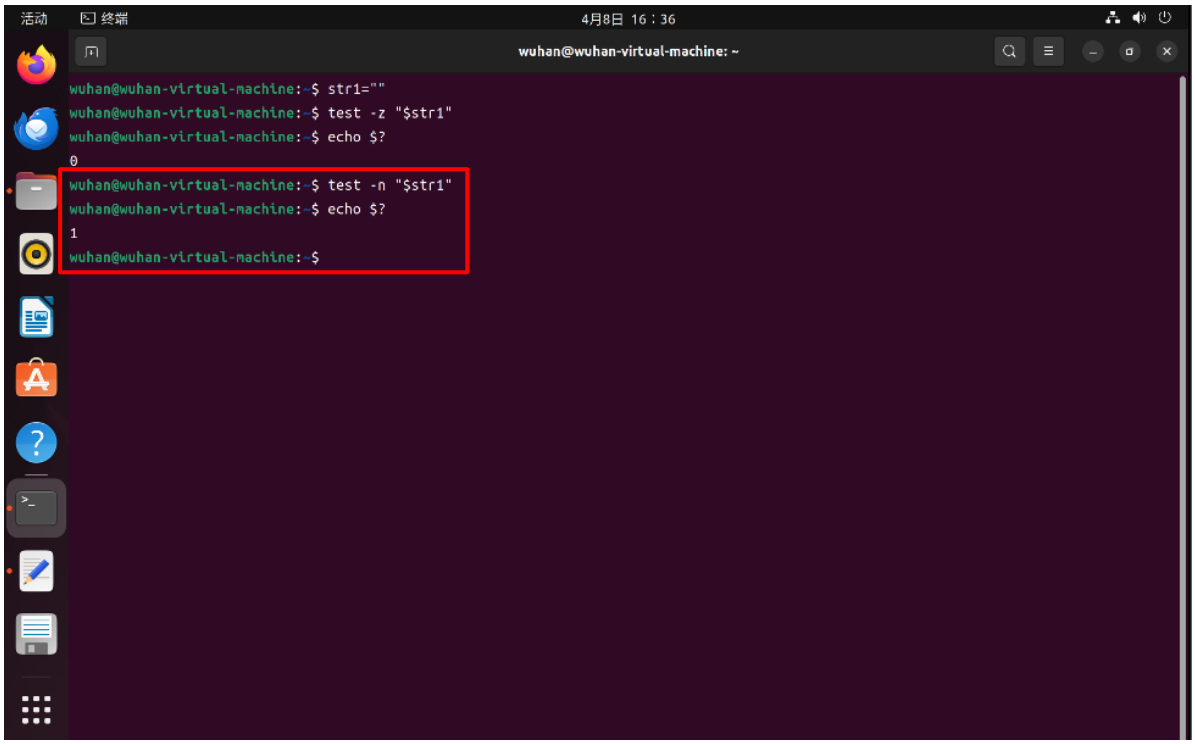


4、测试字符串 str1 是否为空，非空则返回 0，为空返回非 0

命令：test -n "$str1"

echo $?

结果：

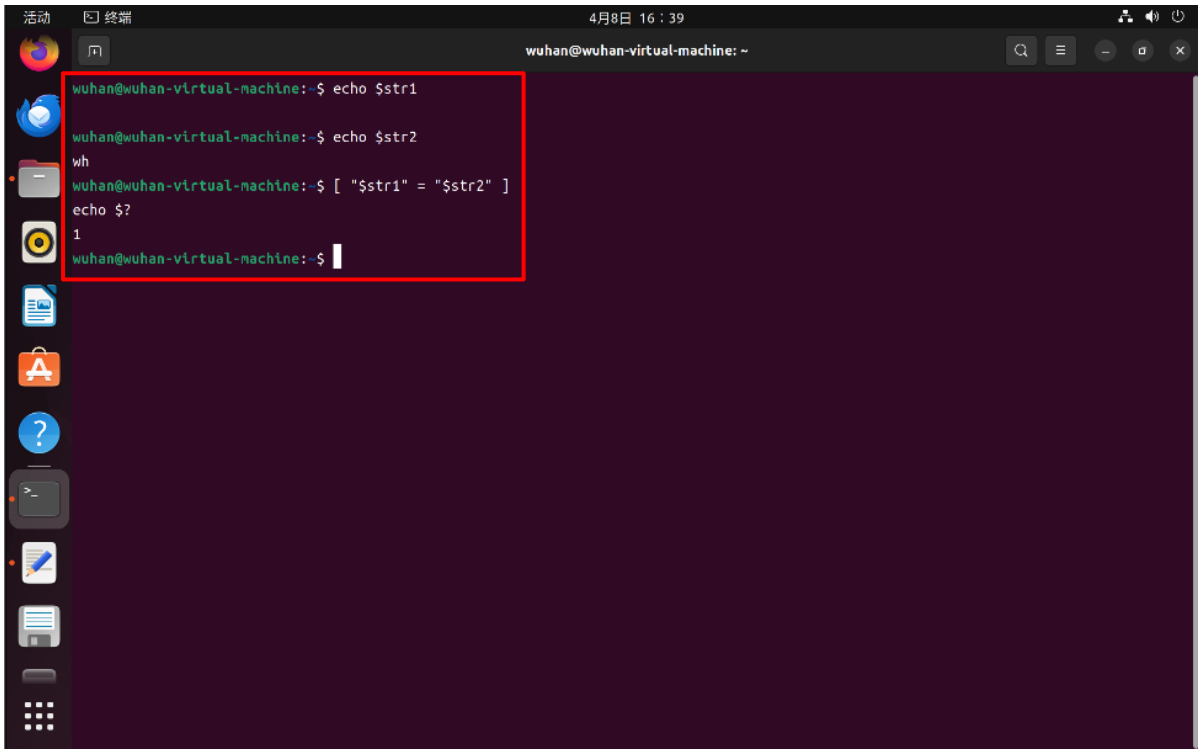


5、比较 str1 与 str2 是否相同，相同则返回 0，否则返回非 0

命令：["$str1" = "$str2"]

echo $?

结果：

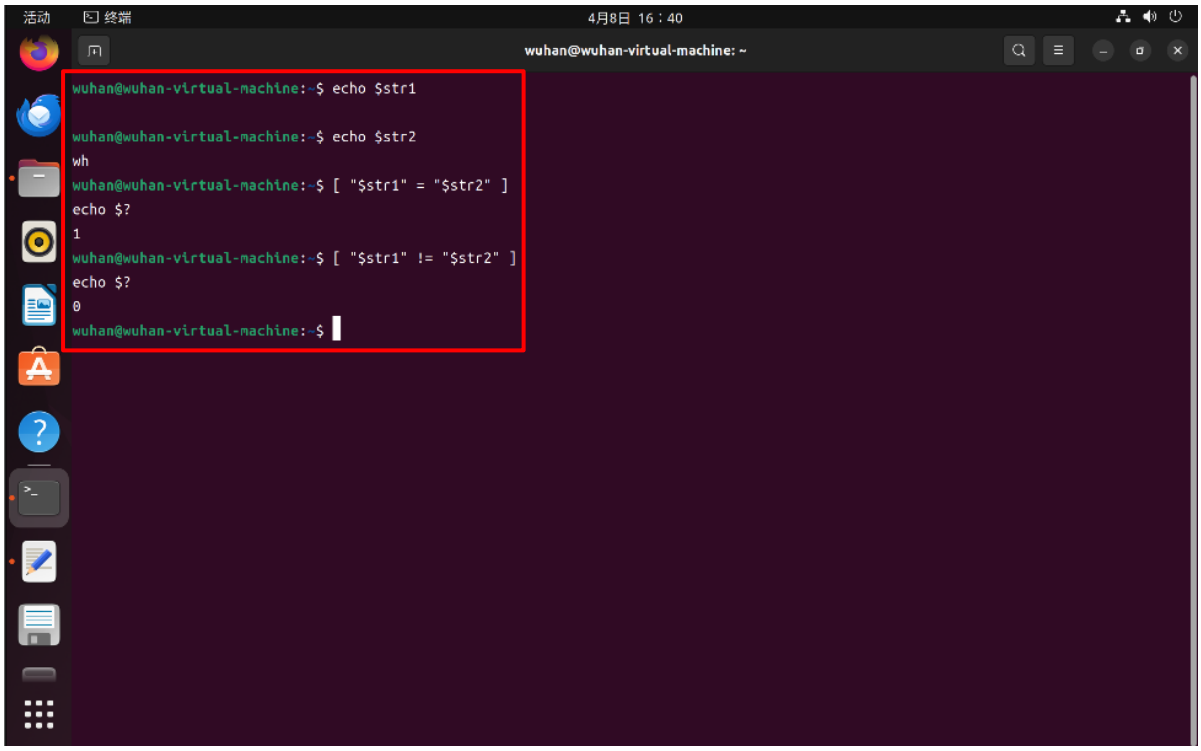


6、比较 str1 与 str2 是否不同，不同则返回 0

命令：["$str1" != "$str2"]

echo $?

结果：



7、比较 str1 与 str2 的大小

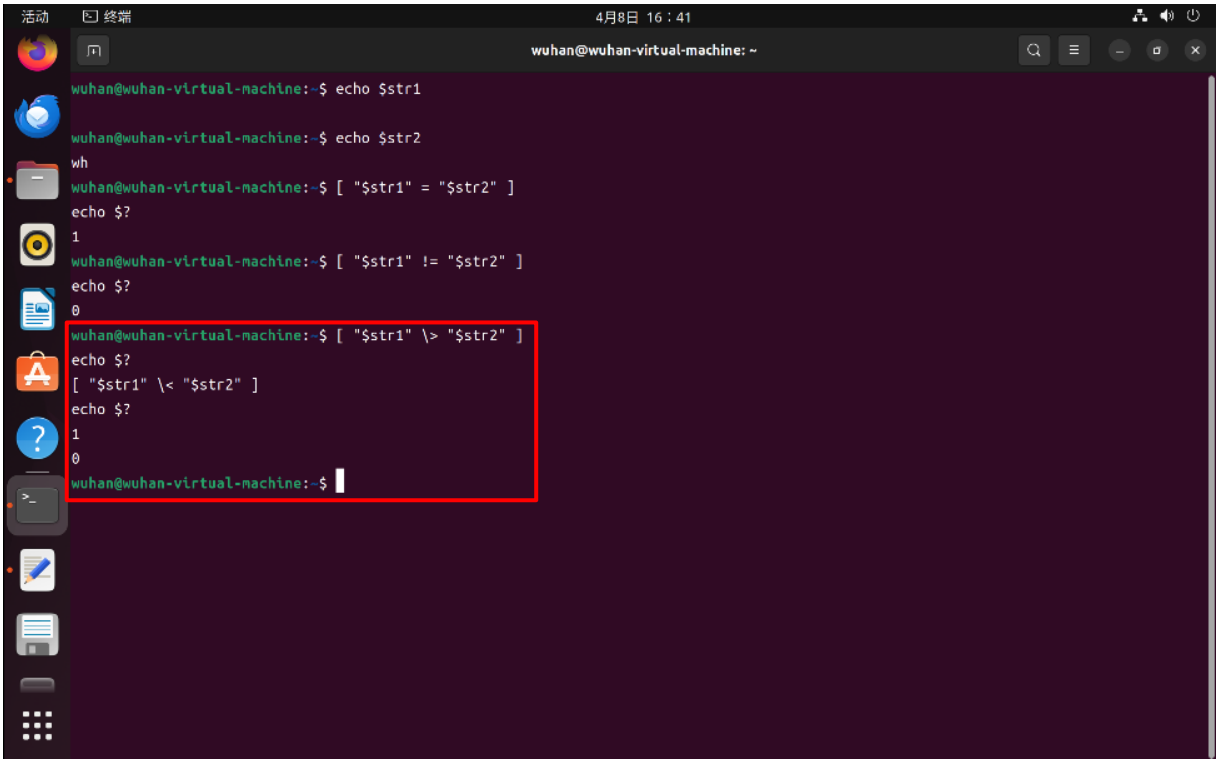
命令：["$str1" \> "$str2"]

echo $?

["$str1" \< "$str2"]

echo $?

结果：



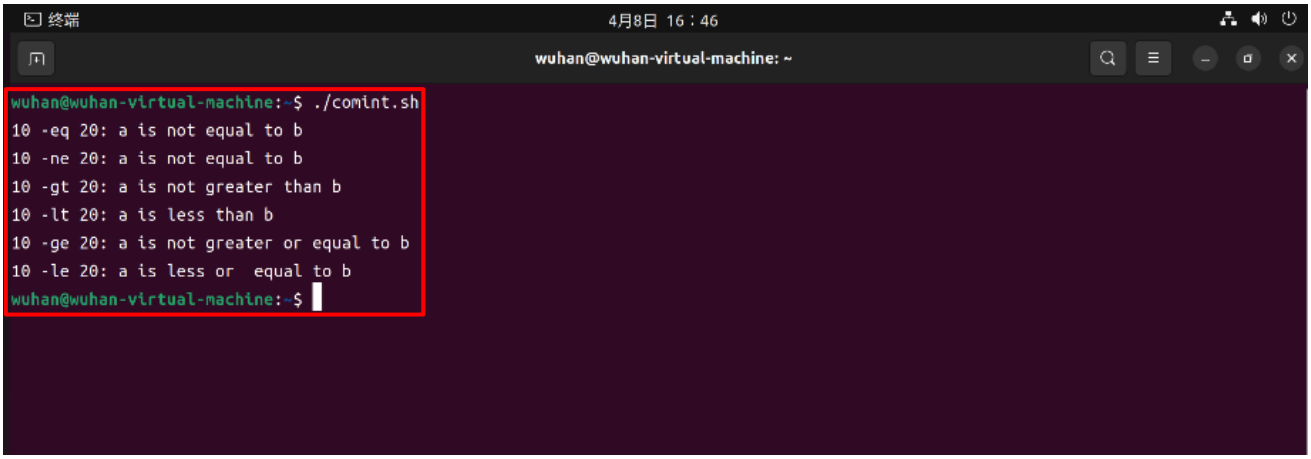
8、建立文件 comint.sh，进行整数比较

脚本：

```
#!/bin/sh
a=10
b=20
if [ $a -eq $b ]
then
   echo "$a -eq $b : a is equal to b"
else
   echo "$a -eq $b: a is not equal to b"
fi
if [ $a -ne $b ]
then
   echo "$a -ne $b: a is not equal to b"
else
   echo "$a -ne $b : a is equal to b"
fi
if [ $a -gt $b ]
then
   echo "$a -gt $b: a is greater than b"
else
   echo "$a -gt $b: a is not greater than b"
fi
if [ $a -lt $b ]
then
   echo "$a -lt $b: a is less than b"
else
   echo "$a -lt $b: a is not less than b"
fi
if [ $a -ge $b ]
then
   echo "$a -ge $b: a is greater or  equal to b"
else
   echo "$a -ge $b: a is not greater or equal to b"
fi
if [ $a -le $b ]
then
   echo "$a -le $b: a is less or  equal to b"
else
   echo "$a -le $b: a is not less or equal to b"
fi
```

命令：. ./ comint.sh

结果：



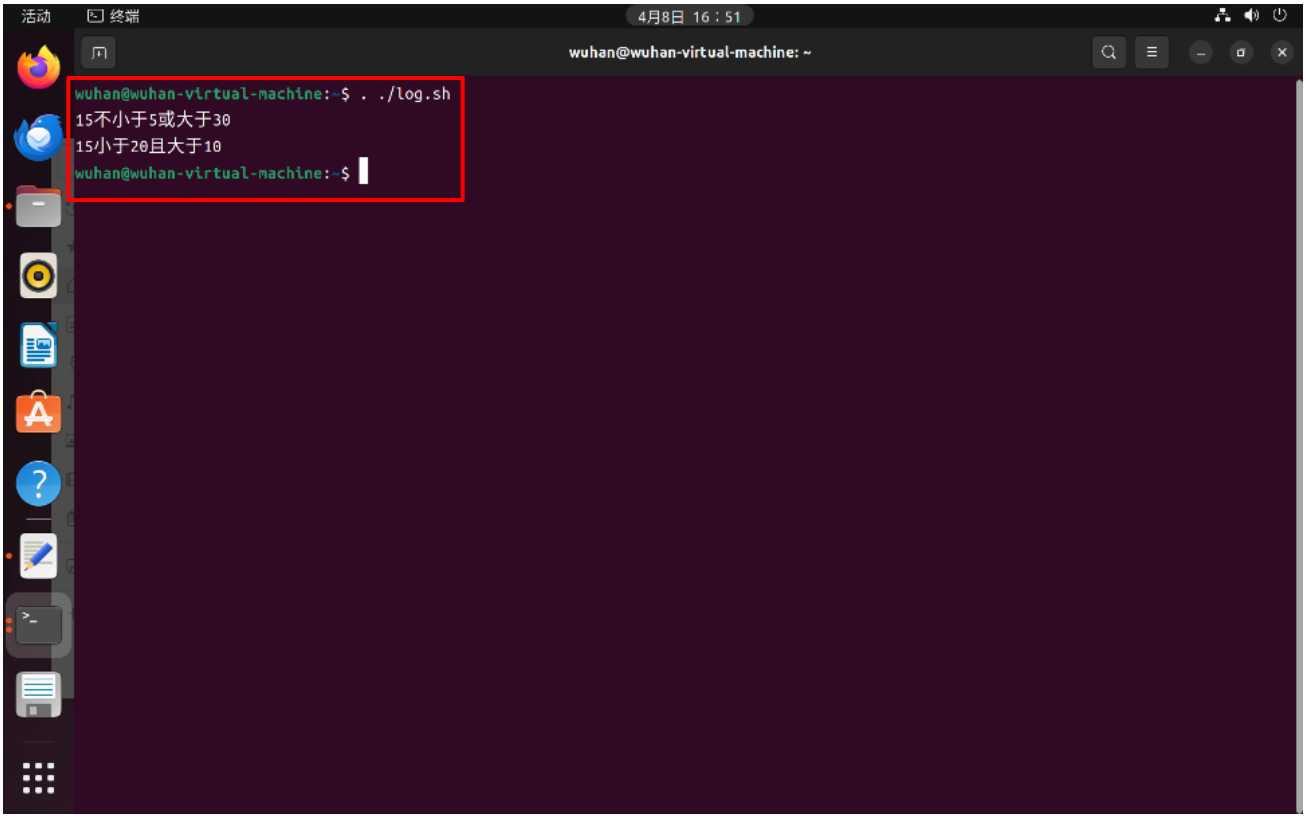
9、建立文件 log.sh，进行逻辑运算

脚本：

```
#!bin/sh
iv=15
if [ ! "$iv" -lt 5 -o "$iv" -gt 30 ];then
   echo "$iv 不小于 5 或大于 30"
fi
if [ "$iv" -lt 20 -a "$iv" -gt 10 ];then
   echo "$iv 小于 20 且大于 10"
fi
```

命令：`. ./log.sh`

结果：



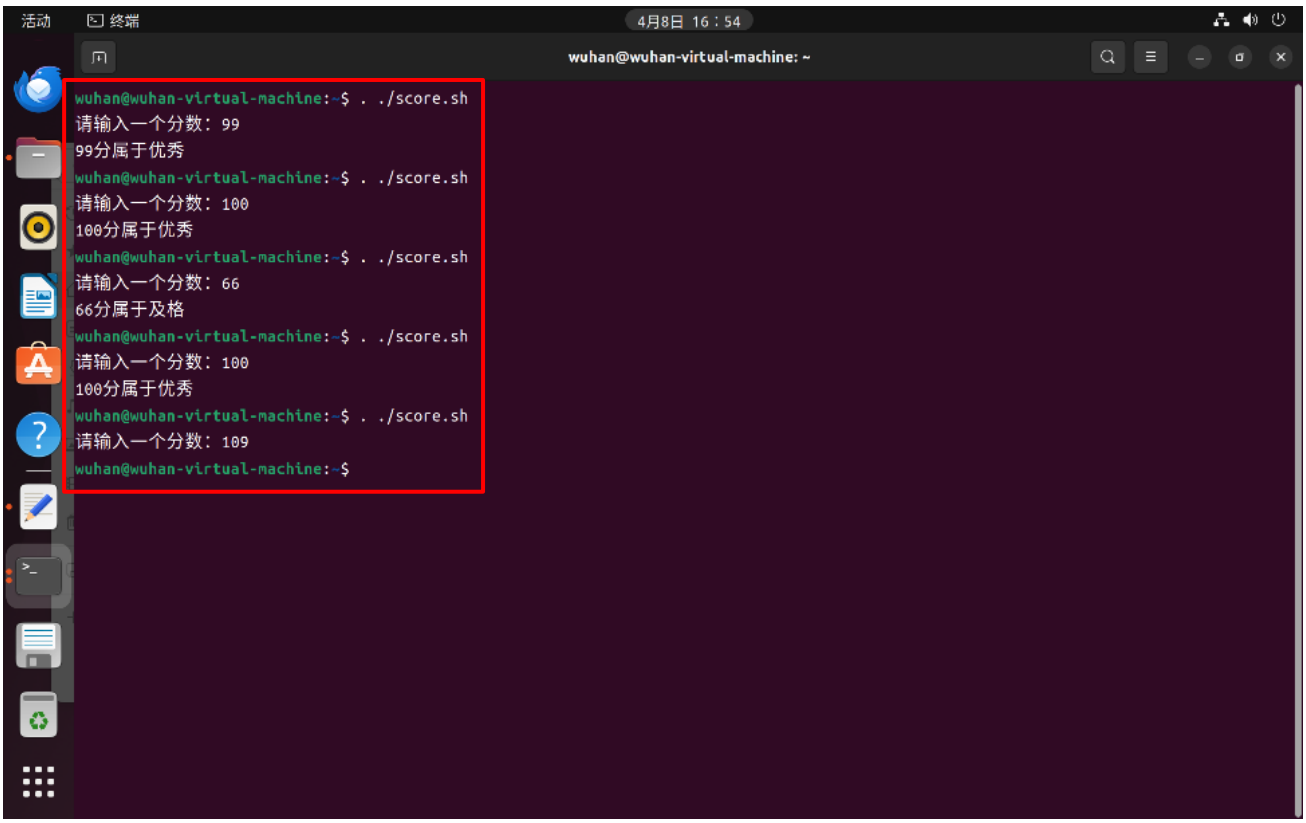
10、建立文件 score.sh，使用 if 进行成绩等级的判断

脚本：

```
#!/bin/bash
echo -n "请输入一个分数："
read score
if [ "$score" -lt 60 ];then
        echo "$score 分属于不及格"
fi
if [ "$score" -lt 70 -a "$score" -ge 60 ];then
        echo "$score 分属于及格"
fi
if [ "$score" -lt 80 -a "$score" -ge 70 ];then
        echo "$score 分属于中等"
fi
if [ "$score" -lt 90 -a "$score" -ge 80 ];then
        echo "$score 分属于良好"
fi
if [ "$score" -le 100 -a "$score" -ge 90 ];then
        echo "$score 分属于优秀"
fi
```

命令：../score.sh

结果：



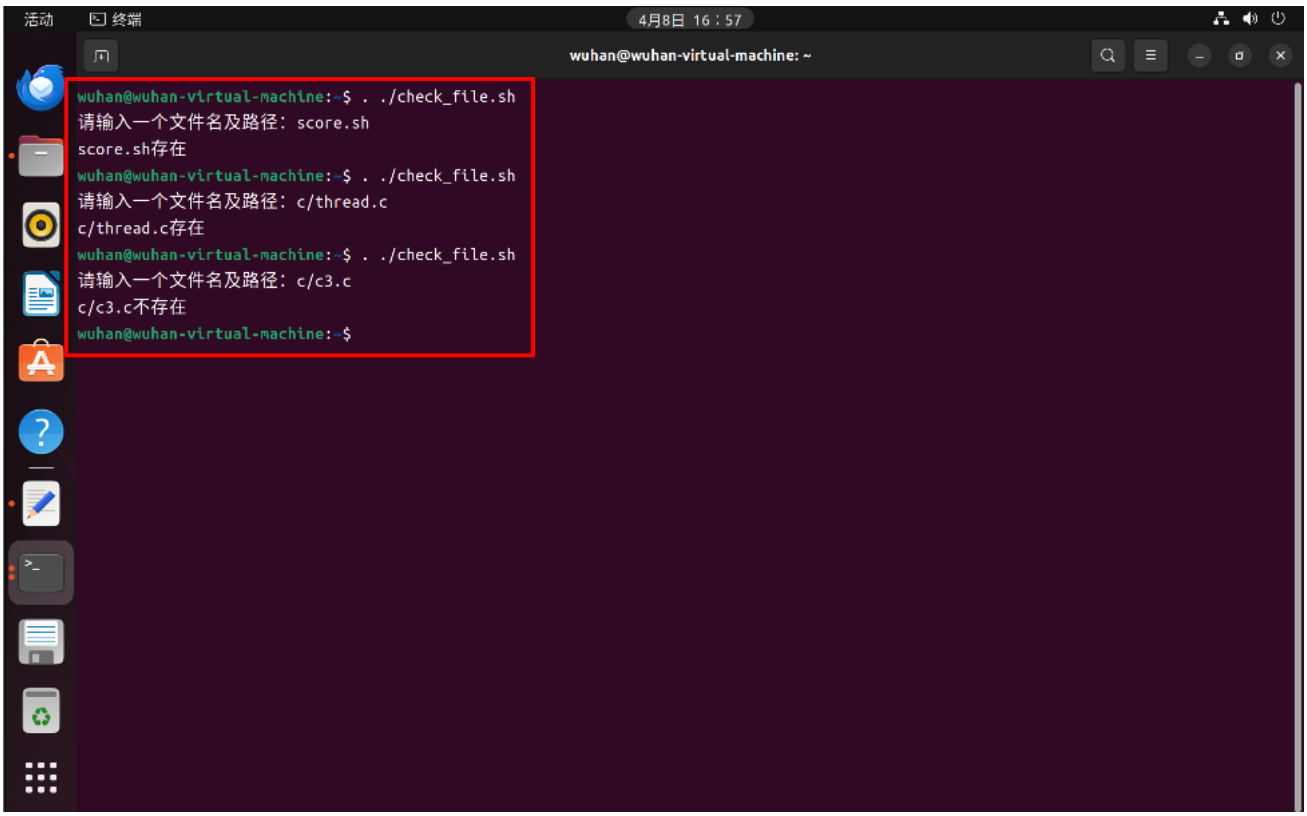
11、建立文件 check_file.sh，判断某个文件是否存在

脚本：

```
#!/bin/bash
echo -n "请输入一个文件名及路径："
read FILE
if [ -e $FILE ];then
        echo "$FILE 存在"
else
        echo "$FILE 不存在"
fi
```

命令：. ./check_file.sh

结果：



12、建立文件 ifelif.sh，对成绩等级进行判断

脚本：

```
#!/bin/bash
echo -n "请输入一个分数："
read score
if [ "$score" -lt 60 ];then
        echo "$score 分属于不及格"
elif [ "$score" -lt 70 -a "$score" -ge 60 ];then
        echo "$score 分属于及格"
elif [ "$score" -lt 80 -a "$score" -ge 70 ];then
        echo "$score 分属于中等"
elif [ "$score" -lt 90 -a "$score" -ge 80 ];then
        echo "$score 分属于良好"
else    echo "$score 分属于优秀"
fi
```

命令：./ifelif.sh

结果：