

姓名：吴涵

学号：202101000720

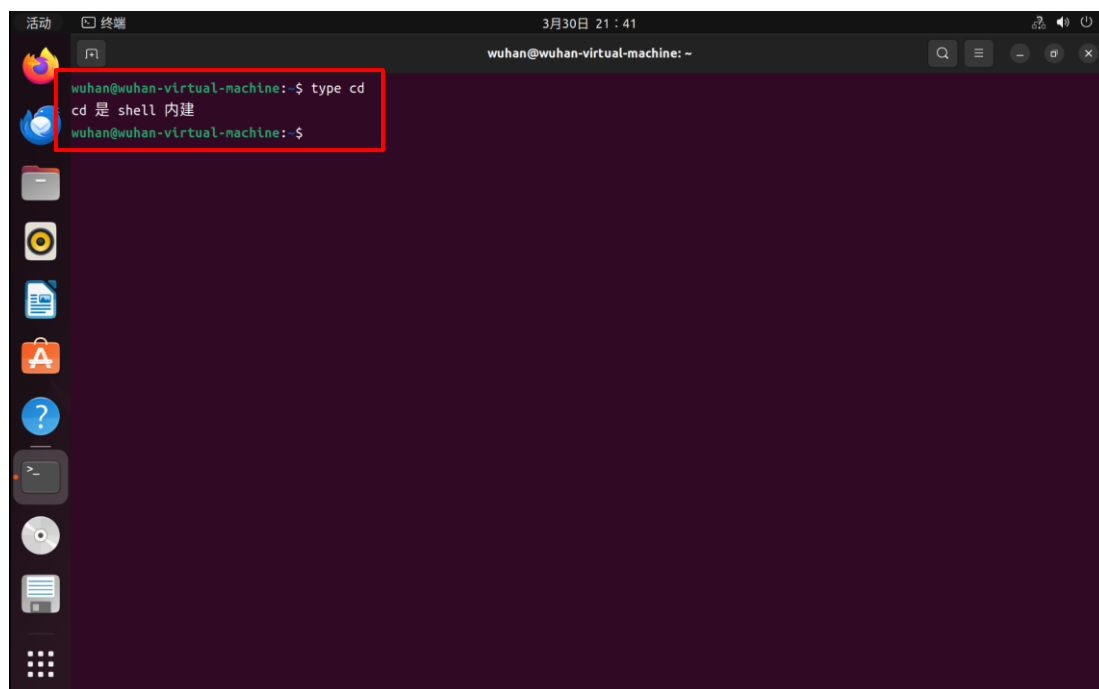
成绩：

## 实验 11 批处理操作接口 1：赋值与取值

1、判断 cd 为内部命令还是外部命令

命令：type cd

结果：

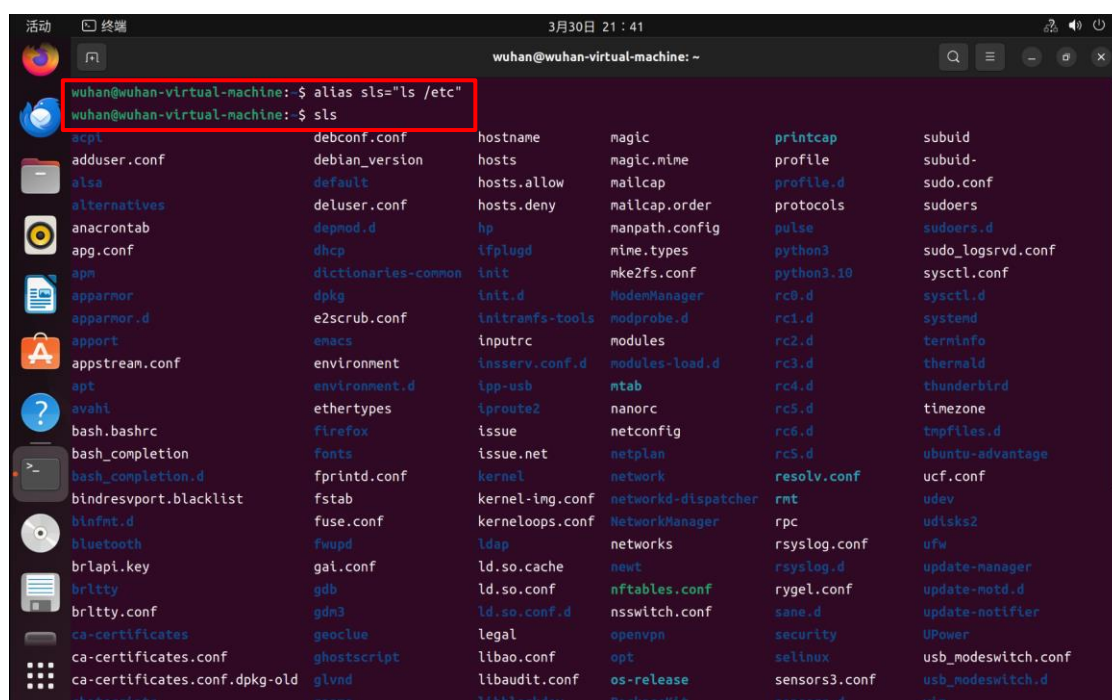


```
wuhan@wuhan-virtual-machine:~$ type cd
cd 是 shell 内建
wuhan@wuhan-virtual-machine:~$
```

2、为命令"ls /etc"取个别名 "sls"

命令：alias sls="ls /etc"

结果：

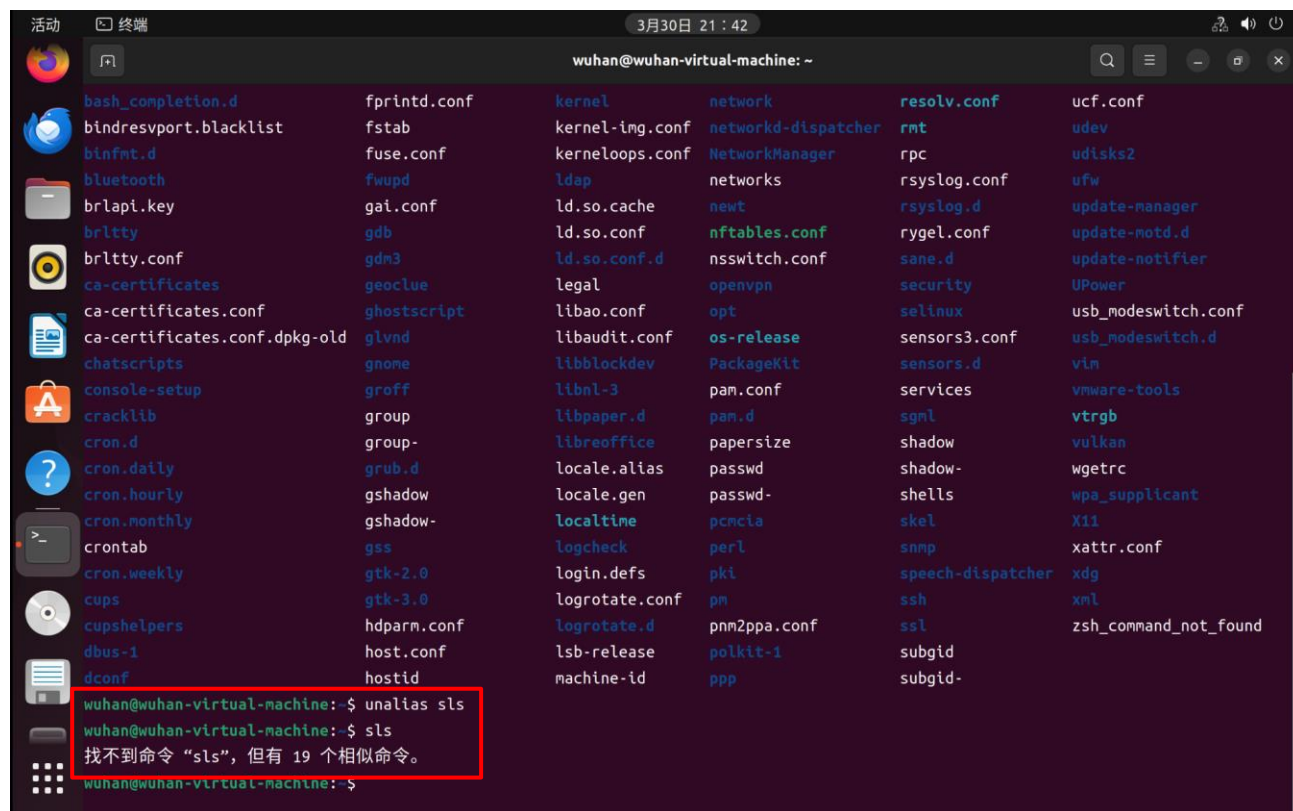


```
wuhan@wuhan-virtual-machine:~$ alias sls="ls /etc"
wuhan@wuhan-virtual-machine:~$ sls
acpt          debconf.conf  hostname      magic          printcap      subuid
adduser.conf  debian_version hosts          magic.mime     profile        subuid-
alsa          default       hosts.allow   mailcap        profile.d      sudo.conf
alternatives  deluser.conf  hosts.deny    mailcap.order  protocols     sudoers
anacrontab    depmod.d      hp            manpath.config pulse          sudoers.d
apg.conf      dhcp          ifplugd       mine.types     python3        sudo_logsrvd.conf
apn           dictionaries-common init           mke2fs.conf    python3.10     sysctl.conf
apparmor      dpkg          init.d        ModemManager   rc0.d          sysctl.d
apparmor.d    e2scrub.conf initramfs-tools modprobe.d      rc1.d          systemd
appport       emacs         inputrc       modules        rc2.d          terminfo
appstream.conf environment  insserv.conf.d modules-load.d  rc3.d          thernald
apt           environment.d iproute2      nanorc         rc4.d          thunderbird
avahi         ethertypes    issue         netconfig      rc5.d          timezone
bash.bashrc   firefox       issue.net     netplan        rc6.d          tmpfiles.d
bash_completion fonts          kernel        network        rc5.d          ubuntu-advantage
bash_completion.d fprintd.conf kernel-img.conf networkd-dispatcher resolv.conf    ucf.conf
bindresvport.blacklist fstab         kerneloops.conf NetworkManager rmt            udev
blfnt.d       fuse.conf     ldap          networks       rsyslog.conf  ufw
bluetooth     fwupd        ld.so.cache   newt           rsyslog.d     update-manager
brlapi.key    gai.conf     ld.so.conf   nftables.conf rygel.conf     update-motd.d
brlty.conf    gdb          ld.so.conf.d  nsswitch.conf sane.d         update-notifier
ca-certificates geoclue       legal         openvpn        security      UPower
ca-certificates.conf ghostscript   libao.conf    opt            selinux       usb_modeswitch.conf
ca-certificates.conf.dpkg-old glvnd        libaudit.conf os-release     sensors3.conf usb_modeswitch.d
chatarrate    gnome        libbluetoothd PackageKit     sensors.d     vin
```

### 3、取消命令别名“sls”

命令: unalias sls

结果:



A terminal window titled 'wuhan@wuhan-virtual-machine: ~' showing the command 'unalias sls' being executed. The output indicates that the alias 'sls' was successfully removed. The terminal also displays a list of system files and directories in the background.

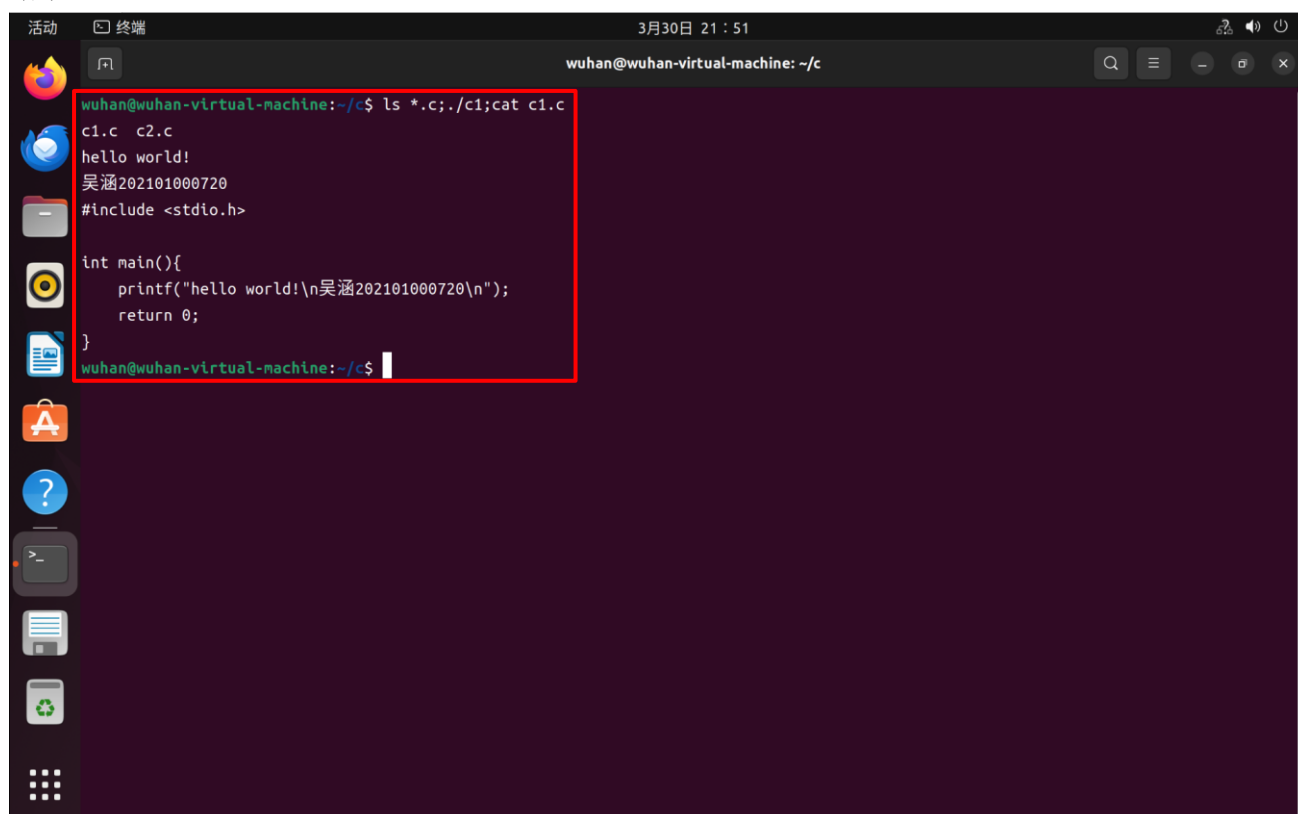
```
wuhan@wuhan-virtual-machine: ~  
bash_completion.d      fprintd.conf          kernel                network               resolv.conf           ucf.conf  
bindresvport.blacklist  fstab                 kernel-img.conf      networkd-dispatcher  rnt                  udev  
binfmt.d               fuse.conf             kernelloops.conf    NetworkManager       rpc                  udisks2  
bluetooth              fwupd                ld.so.cache         networks              rsyslog.conf         ufw  
brlapi.key             gai.conf             ld.so.conf          newt                  rsyslog.d            update-manager  
brltty                 gdb                  ld.so.conf.d        nftables.conf        rygel.conf           update-motd.d  
brltty.conf            gdm3                 ld.so.conf.d        nsswitch.conf        sane.d               update-notifier  
ca-certificates        geoclue              legal                openvpn               security              UPower  
ca-certificates.conf   ghostscript           libao.conf          opt                   selinux              usb_modeswitch.conf  
ca-certificates.conf.dpkg-old  glvnd                libaudit.conf       os-release            sensors3.conf        usb_modeswitch.d  
chatscripts            gnome                 libblockdev         PackageKit            sensors.d             vim  
console-setup          groff                libl-3              pam.conf              services             vmware-tools  
cracklib               group                libreoffice          pam.d                 sgm1                  vtrgb  
cron.d                 group-               libsecret            papersize              shadow                vulkan  
cron.daily             grub.d              locale.alias        passwd                 shadow-              wgetrc  
cron.hourly            gshadow              locale.gen          passwd-                shells                wpa_supplicant  
cron.monthly           gshadow-            localtime          pcmcia                 skel                  X11  
crontab                gss                 logcheck            perl                   snmp                  xattr.conf  
cron.weekly            gtk-2.0              logrotate.conf      pki                    speech-dispatcher    xdg  
cups                   gtk-3.0              logrotate.d         pm                     ssh                   xml  
cupshelpers            hdparm.conf          lsb-release         pnm2ppa.conf          ssl                   zsh_command_not_found  
dbus-1                 host.conf            machine-id           polkit-1              subgid  
dconf                  hostid              machine-id           ppp                    subgid-
```

```
wuhan@wuhan-virtual-machine:~$ unalias sls  
wuhan@wuhan-virtual-machine:~$ sls  
找不到命令“sls”，但有 19 个相似命令。  
wuhan@wuhan-virtual-machine:~$
```

### 4、顺序执行命令“ls \*.c”、“./c1”和“cat c1.c”

命令: ls \*.c;./c1;cat c1.c

结果:



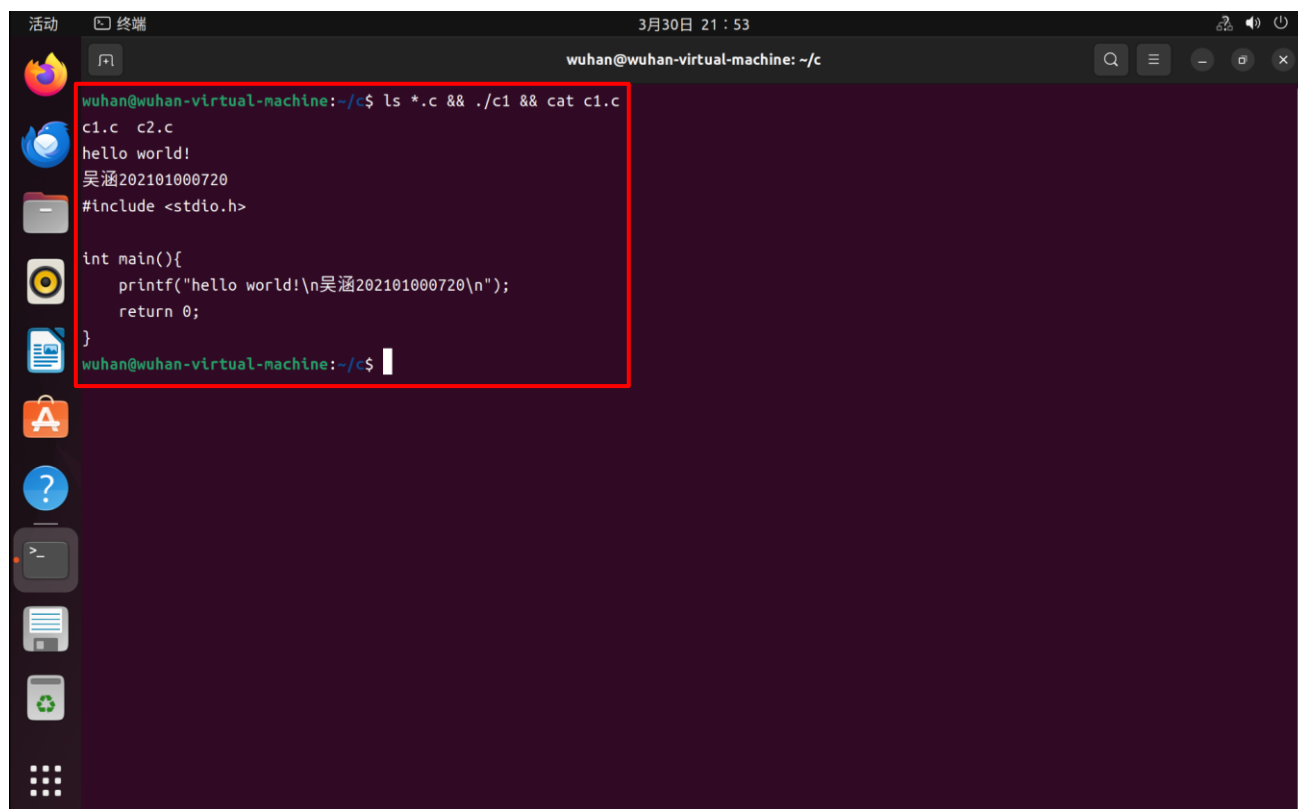
A terminal window titled 'wuhan@wuhan-virtual-machine: ~/c' showing the execution of the commands 'ls \*.c', './c1', and 'cat c1.c'. The output shows the files 'c1.c' and 'c2.c', the execution of the program 'c1.c', and the contents of 'c1.c' which is a C program that prints 'hello world!' followed by the user's name.

```
wuhan@wuhan-virtual-machine:~/c$ ls *.c;./c1;cat c1.c  
c1.c  c2.c  
hello world!  
吴涵202101000720  
#include <stdio.h>  
  
int main(){  
    printf("hello world!\n吴涵202101000720\n");  
    return 0;  
}  
wuhan@wuhan-virtual-machine:~/c$
```

5、执行命令“ls \*.c”，若成功，则继续执行命令“./c1”，若又成功，则继续执行命令“cat c1.c”

命令：ls \*.c && ./c1 && cat c1.c

结果：



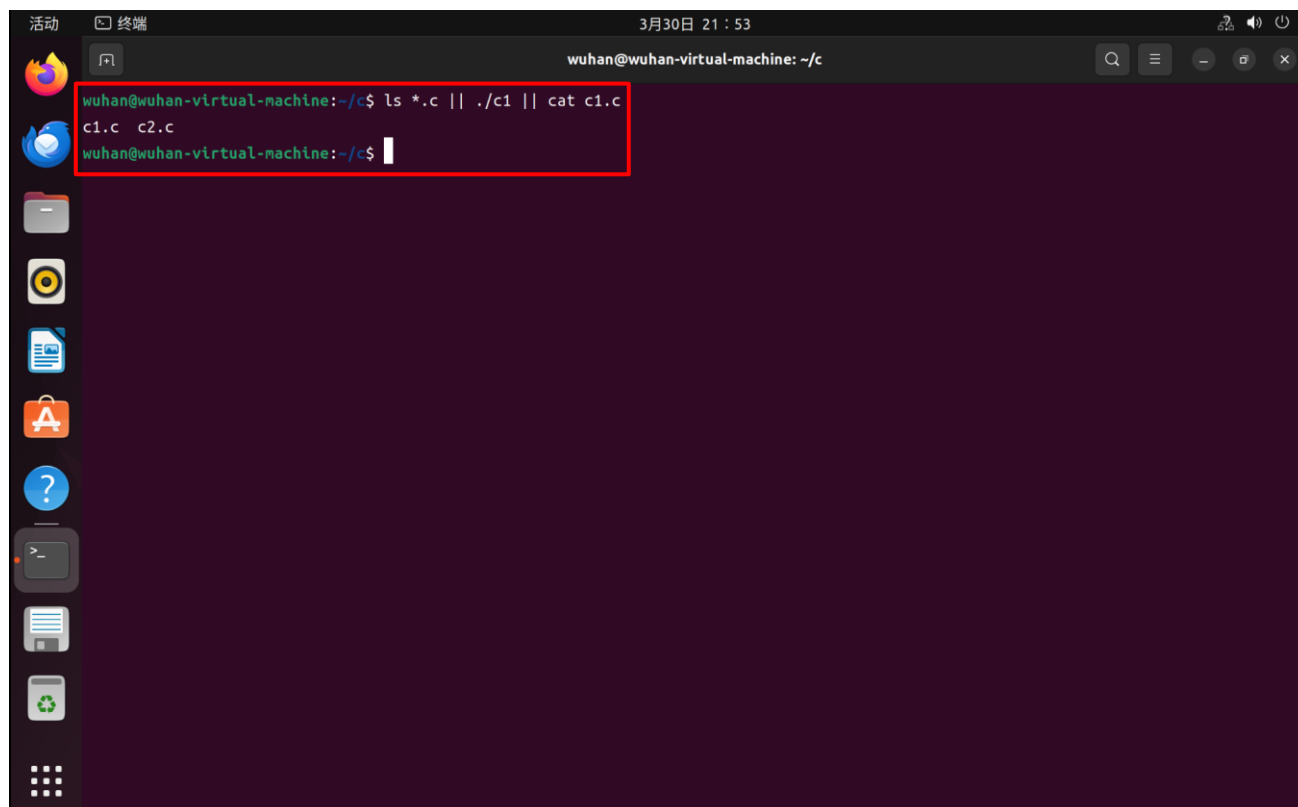
```
wuhan@wuhan-virtual-machine: ~/c
wuhan@wuhan-virtual-machine:~/c$ ls *.c && ./c1 && cat c1.c
c1.c  c2.c
hello world!
吴涵202101000720
#include <stdio.h>

int main(){
    printf("hello world!\n吴涵202101000720\n");
    return 0;
}
wuhan@wuhan-virtual-machine:~/c$
```

6、执行命令“ls \*.c”，若不成功，则继续执行命令“./c1”，若又不成功，则继续执行命令“cat c1.c”

命令：ls \*.c || ./c1 || cat c1.c

结果：

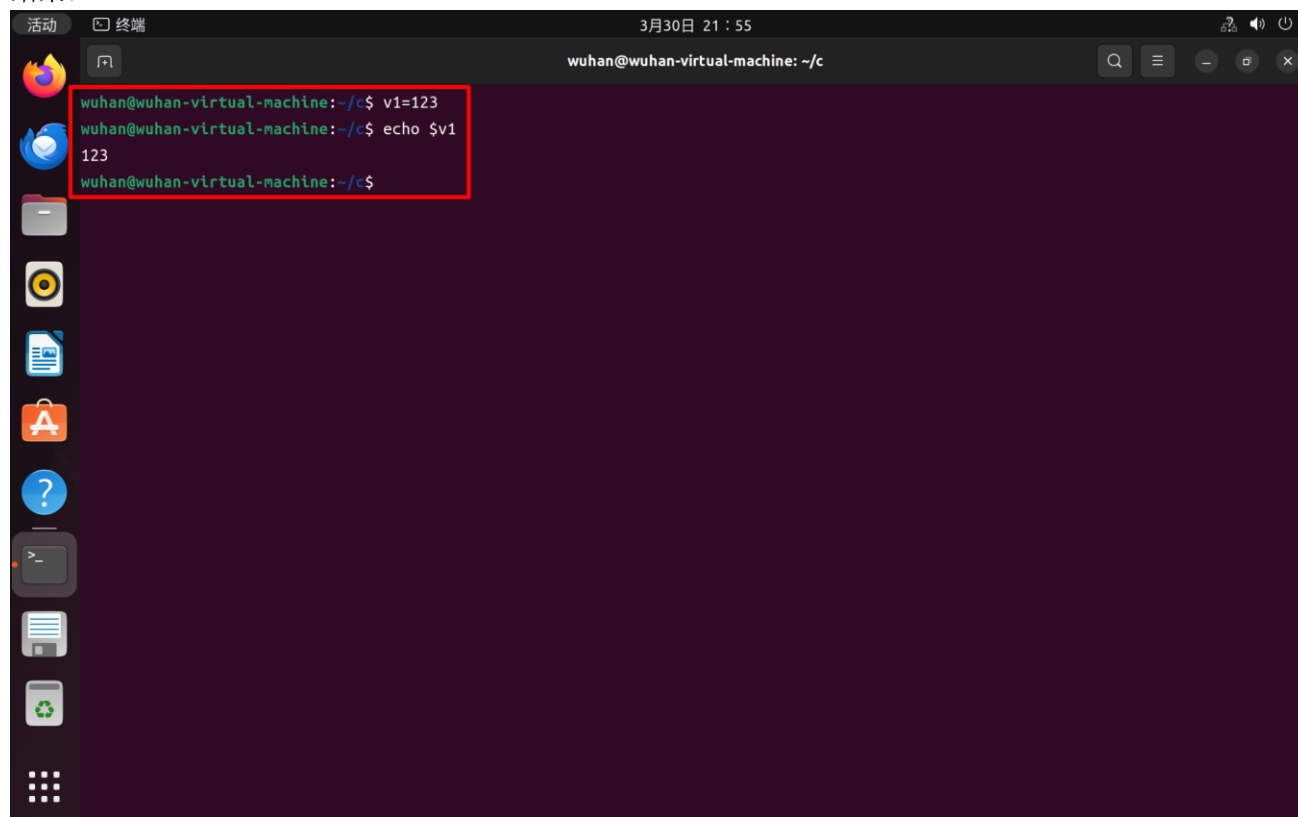


```
wuhan@wuhan-virtual-machine: ~/c
wuhan@wuhan-virtual-machine:~/c$ ls *.c || ./c1 || cat c1.c
c1.c  c2.c
wuhan@wuhan-virtual-machine:~/c$
```

7、为变量 v1 赋值 123

命令: v1=123

结果:



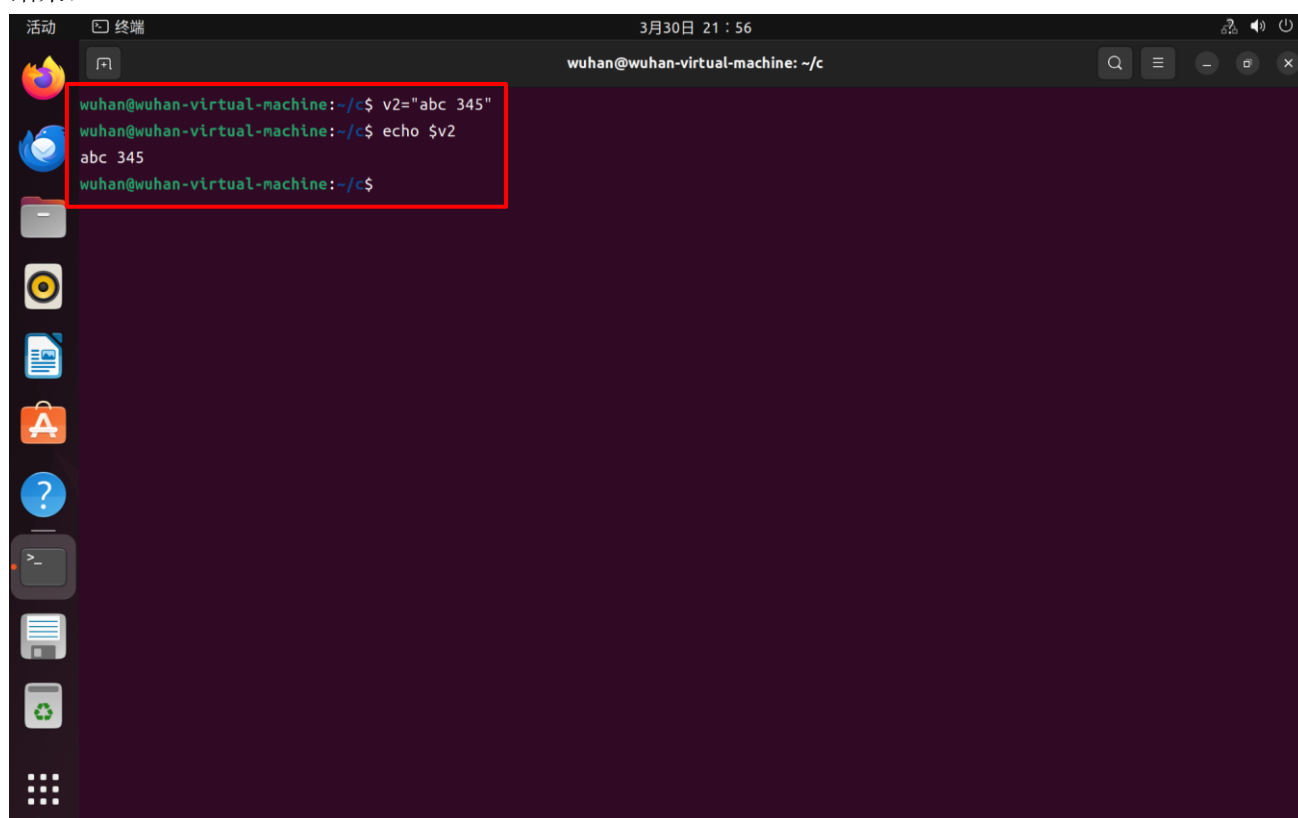
A terminal window titled 'wuhan@wuhan-virtual-machine: ~/c' with a timestamp of '3月30日 21:55'. The terminal shows three lines of text: 'wuhan@wuhan-virtual-machine:~/c\$ v1=123', 'wuhan@wuhan-virtual-machine:~/c\$ echo \$v1', and '123'. The first two lines are enclosed in a red rectangular box. The terminal has a dark purple background and a sidebar on the left with various application icons.

```
wuhan@wuhan-virtual-machine:~/c$ v1=123
wuhan@wuhan-virtual-machine:~/c$ echo $v1
123
wuhan@wuhan-virtual-machine:~/c$
```

8、为变量 v2 赋值"abc 345"

命令: v2="abc 345"

结果:



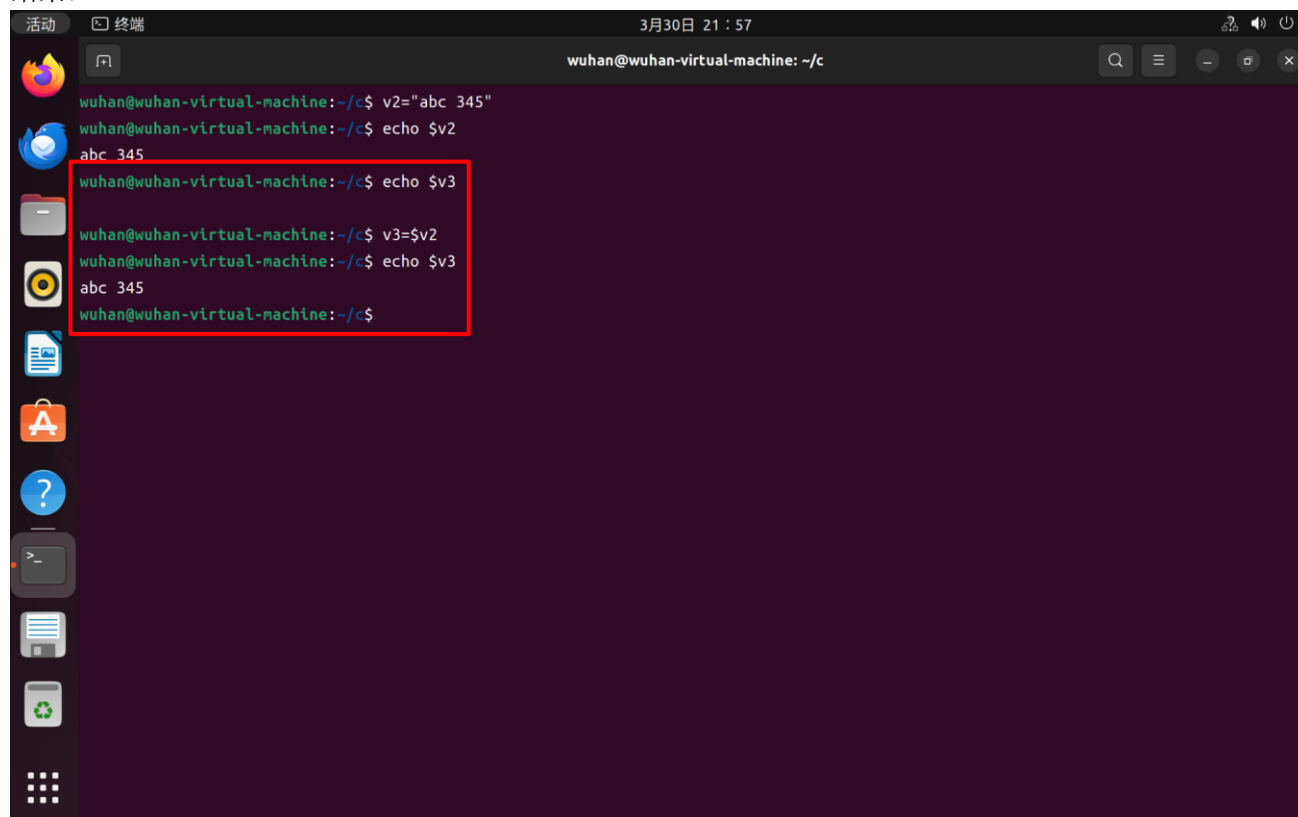
A terminal window titled 'wuhan@wuhan-virtual-machine: ~/c' with a timestamp of '3月30日 21:56'. The terminal shows three lines of text: 'wuhan@wuhan-virtual-machine:~/c\$ v2="abc 345"', 'wuhan@wuhan-virtual-machine:~/c\$ echo \$v2', and 'abc 345'. The first two lines are enclosed in a red rectangular box. The terminal has a dark purple background and a sidebar on the left with various application icons.

```
wuhan@wuhan-virtual-machine:~/c$ v2="abc 345"
wuhan@wuhan-virtual-machine:~/c$ echo $v2
abc 345
wuhan@wuhan-virtual-machine:~/c$
```

9、将变量 v2 的值赋给 v3

命令: `v3=$v2`

结果:



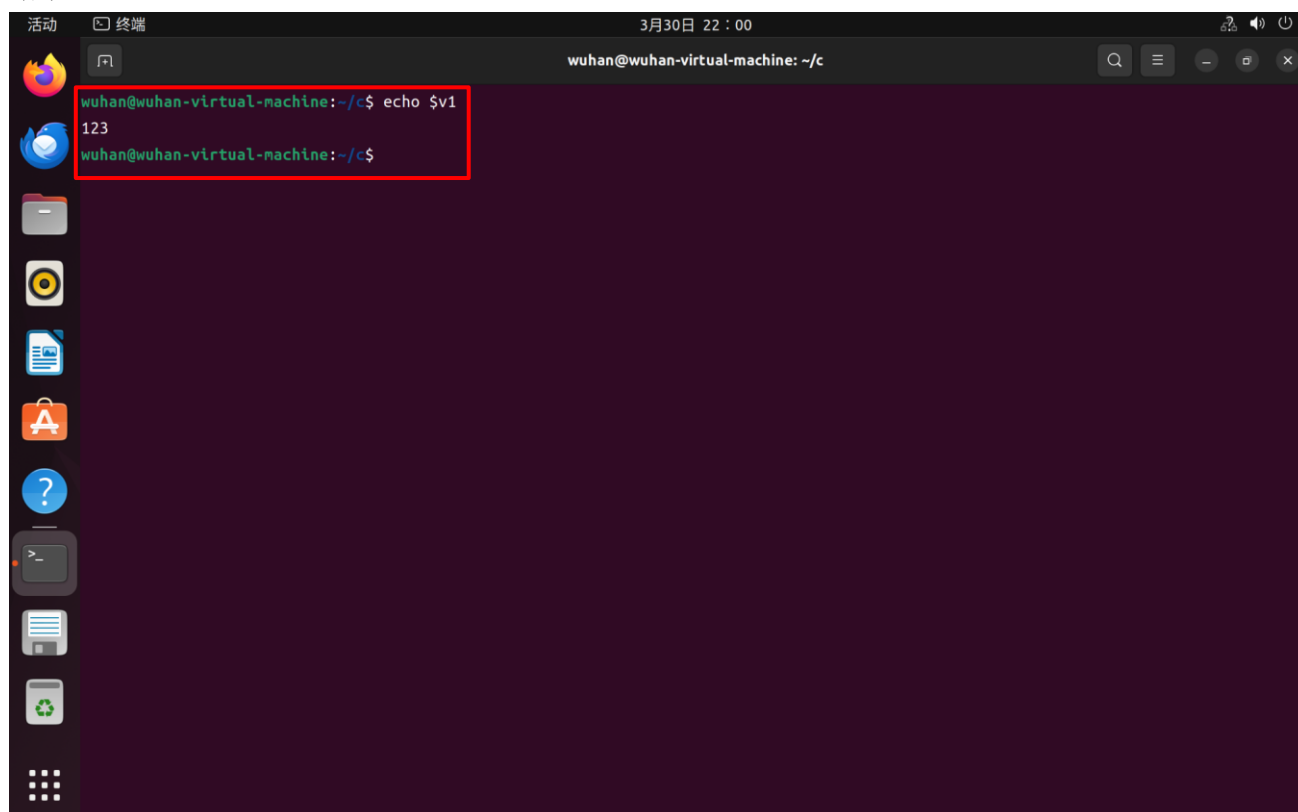
A terminal window titled '终端' (Terminal) showing a series of commands and their outputs. The prompt is 'wuhan@wuhan-virtual-machine: ~/c'. The commands and outputs are: `wuhan@wuhan-virtual-machine:~/c$ v2="abc 345"`, `wuhan@wuhan-virtual-machine:~/c$ echo $v2` (output: `abc 345`), `wuhan@wuhan-virtual-machine:~/c$ echo $v3` (output: `abc 345`), `wuhan@wuhan-virtual-machine:~/c$ v3=$v2`, `wuhan@wuhan-virtual-machine:~/c$ echo $v3` (output: `abc 345`), and `wuhan@wuhan-virtual-machine:~/c$`. A red box highlights the last three lines of the terminal output.

```
wuhan@wuhan-virtual-machine:~/c$ v2="abc 345"
wuhan@wuhan-virtual-machine:~/c$ echo $v2
abc 345
wuhan@wuhan-virtual-machine:~/c$ echo $v3
abc 345
wuhan@wuhan-virtual-machine:~/c$ v3=$v2
wuhan@wuhan-virtual-machine:~/c$ echo $v3
abc 345
wuhan@wuhan-virtual-machine:~/c$
```

10、显示变量 v1 的值

命令: `echo $v1`

结果:



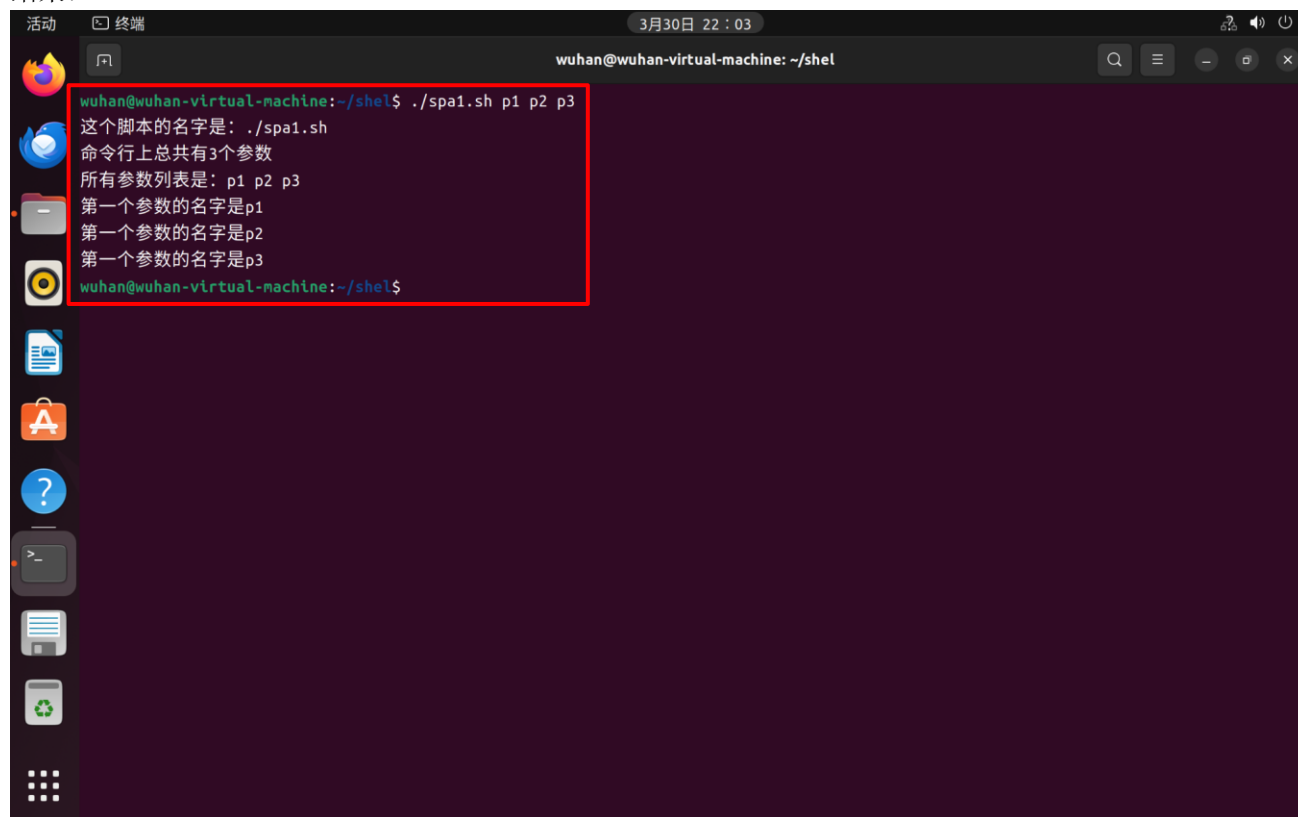
A terminal window titled '终端' (Terminal) showing a single command and its output. The prompt is 'wuhan@wuhan-virtual-machine: ~/c'. The command and output are: `wuhan@wuhan-virtual-machine:~/c$ echo $v1` (output: `123`). A red box highlights the command and its output.

```
wuhan@wuhan-virtual-machine:~/c$ echo $v1
123
wuhan@wuhan-virtual-machine:~/c$
```

## 11、Shell 脚本程序命令行参数访问

命令: `./spa1.sh p1 p2 p3`

结果:



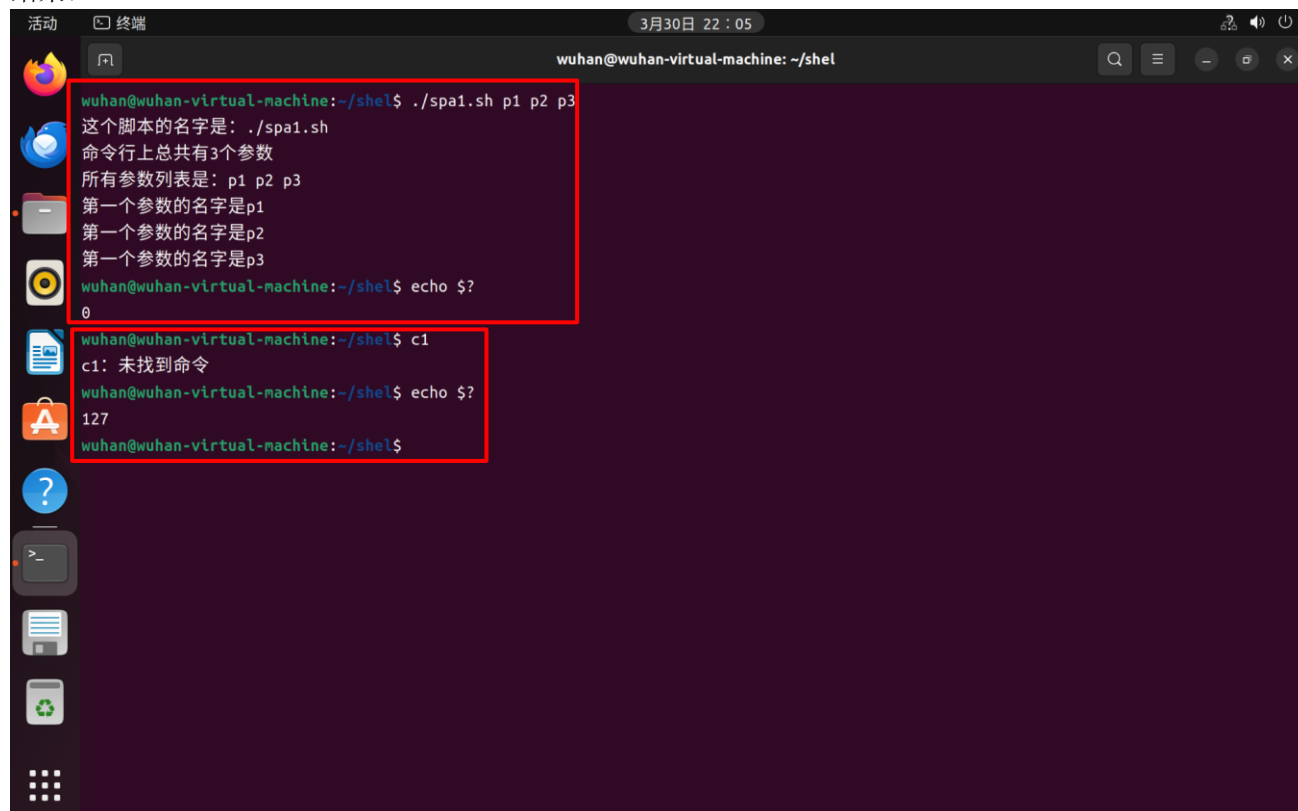
A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' showing the execution of a script. The command `./spa1.sh p1 p2 p3` is entered. The script outputs: '这个脚本的名字是: ./spa1.sh', '命令行上总共有3个参数', '所有参数列表是: p1 p2 p3', '第一个参数的名字是p1', '第一个参数的名字是p2', and '第一个参数的名字是p3'. The prompt returns to `wuhan@wuhan-virtual-machine:~/shel$`.

```
wuhan@wuhan-virtual-machine:~/shel$ ./spa1.sh p1 p2 p3
这个脚本的名字是: ./spa1.sh
命令行上总共有3个参数
所有参数列表是: p1 p2 p3
第一个参数的名字是p1
第一个参数的名字是p2
第一个参数的名字是p3
wuhan@wuhan-virtual-machine:~/shel$
```

## 12、查看命令返回值

命令: `c1`、`echo $?`

结果:



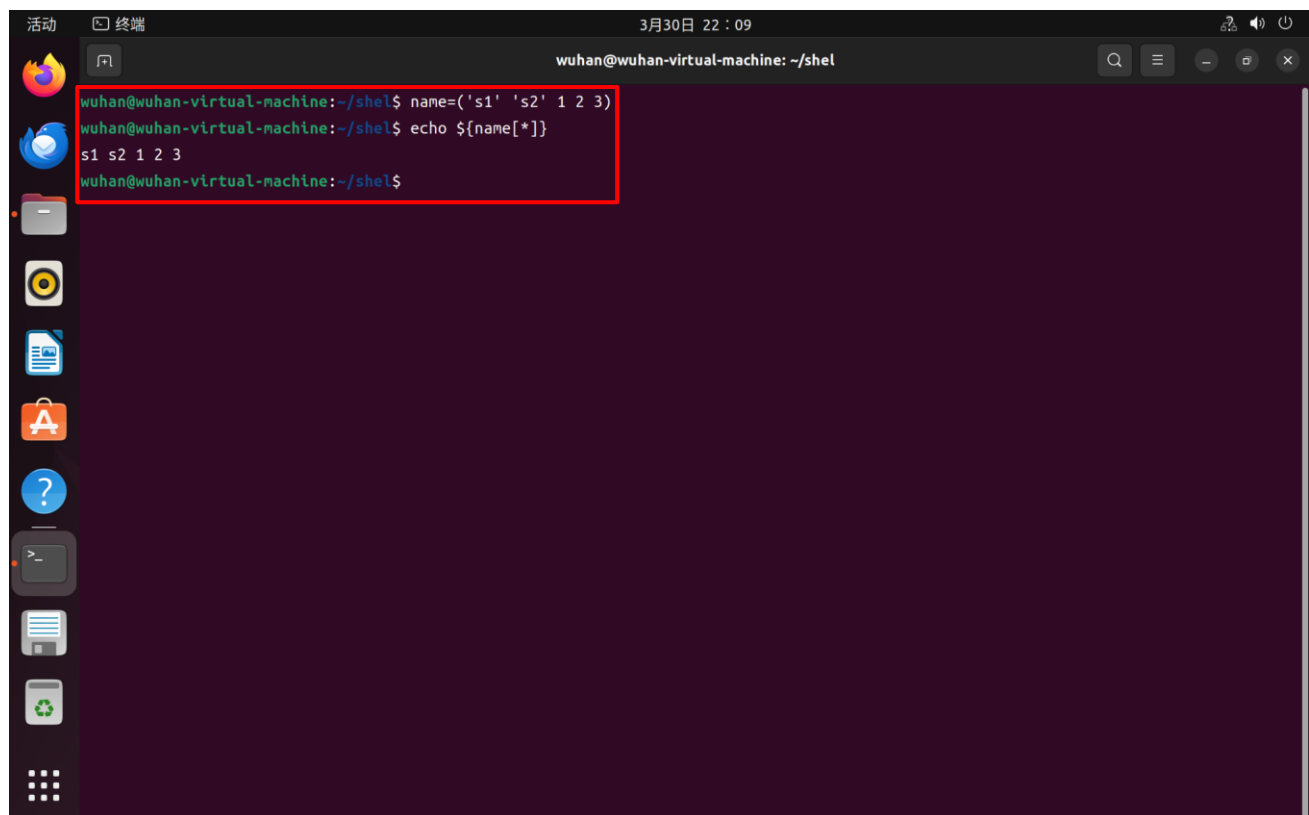
A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' showing the execution of a script and subsequent commands. The script `./spa1.sh p1 p2 p3` is run, outputting the same parameter information as in the previous screenshot. Then, the command `echo $?` is entered, returning `0`. Next, the command `c1` is entered, resulting in the error `c1: 未找到命令`. Finally, `echo $?` is entered again, returning `127`. The prompt returns to `wuhan@wuhan-virtual-machine:~/shel$`.

```
wuhan@wuhan-virtual-machine:~/shel$ ./spa1.sh p1 p2 p3
这个脚本的名字是: ./spa1.sh
命令行上总共有3个参数
所有参数列表是: p1 p2 p3
第一个参数的名字是p1
第一个参数的名字是p2
第一个参数的名字是p3
wuhan@wuhan-virtual-machine:~/shel$ echo $?
0
wuhan@wuhan-virtual-machine:~/shel$ c1
c1: 未找到命令
wuhan@wuhan-virtual-machine:~/shel$ echo $?
127
wuhan@wuhan-virtual-machine:~/shel$
```

### 13、数组整体全部赋值

命令: `name=('s1' 's2' 1 2 3)`

结果:



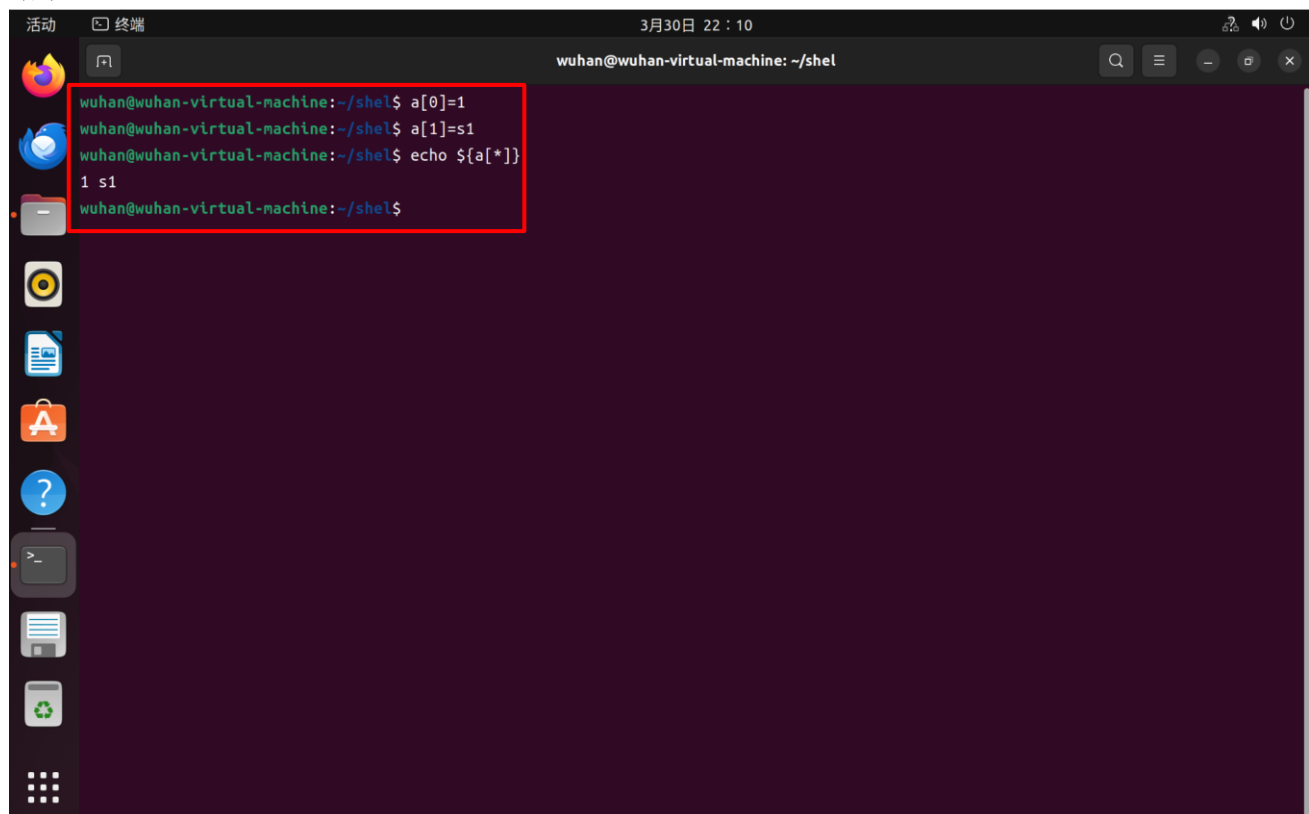
A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a date and time of '3月30日 22:09'. The terminal shows the following commands and output:

```
wuhan@wuhan-virtual-machine:~/shel$ name=('s1' 's2' 1 2 3)
wuhan@wuhan-virtual-machine:~/shel$ echo ${name[*]}
s1 s2 1 2 3
wuhan@wuhan-virtual-machine:~/shel$
```

### 14、数组元素单个赋值

命令: `a[0]=1`、`a[1]=s1`

结果:



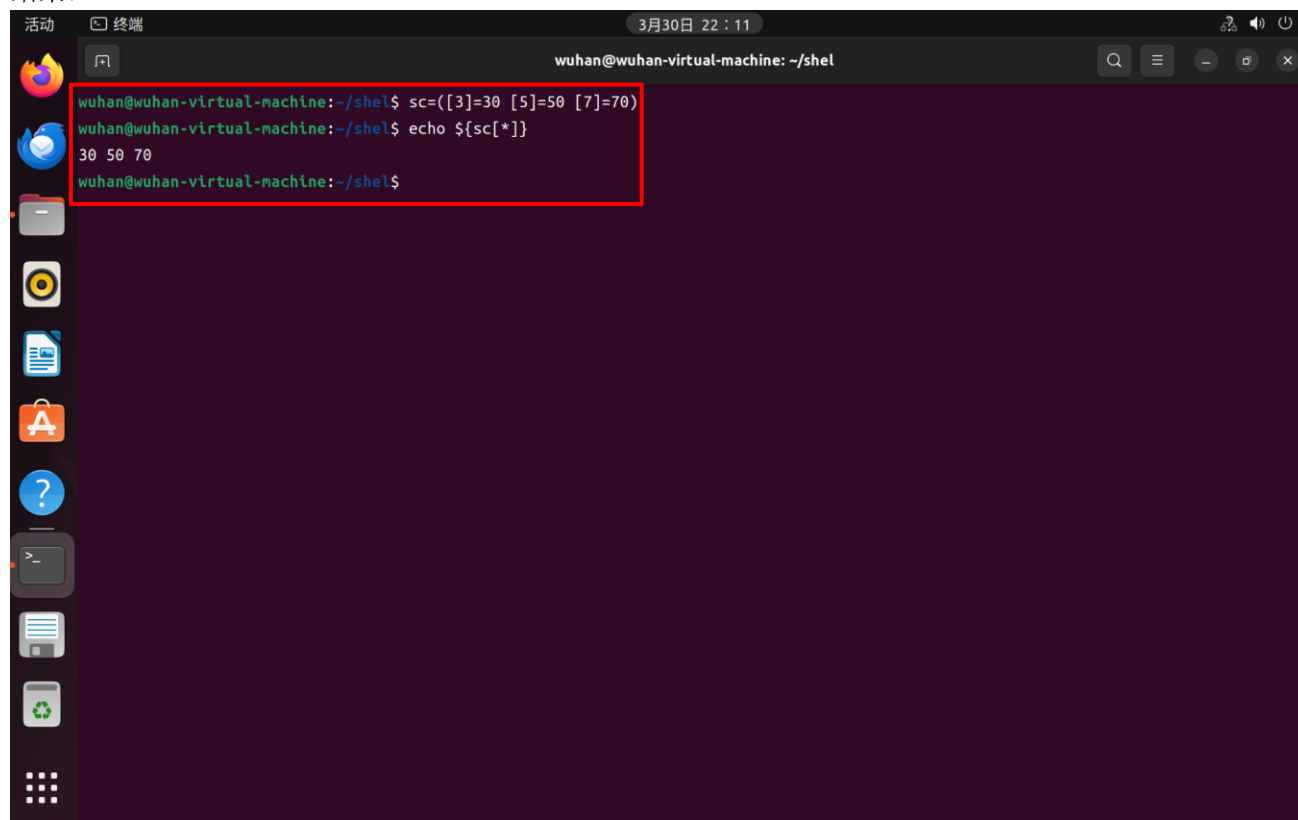
A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a date and time of '3月30日 22:10'. The terminal shows the following commands and output:

```
wuhan@wuhan-virtual-machine:~/shel$ a[0]=1
wuhan@wuhan-virtual-machine:~/shel$ a[1]=s1
wuhan@wuhan-virtual-machine:~/shel$ echo ${a[*]}
1 s1
wuhan@wuhan-virtual-machine:~/shel$
```

## 15、数组个别元素赋值

命令： `sc=([3]=30 [5]=50 [7]=70)`

结果：



A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a timestamp of '3月30日 22:11'. The terminal shows the following commands and output:

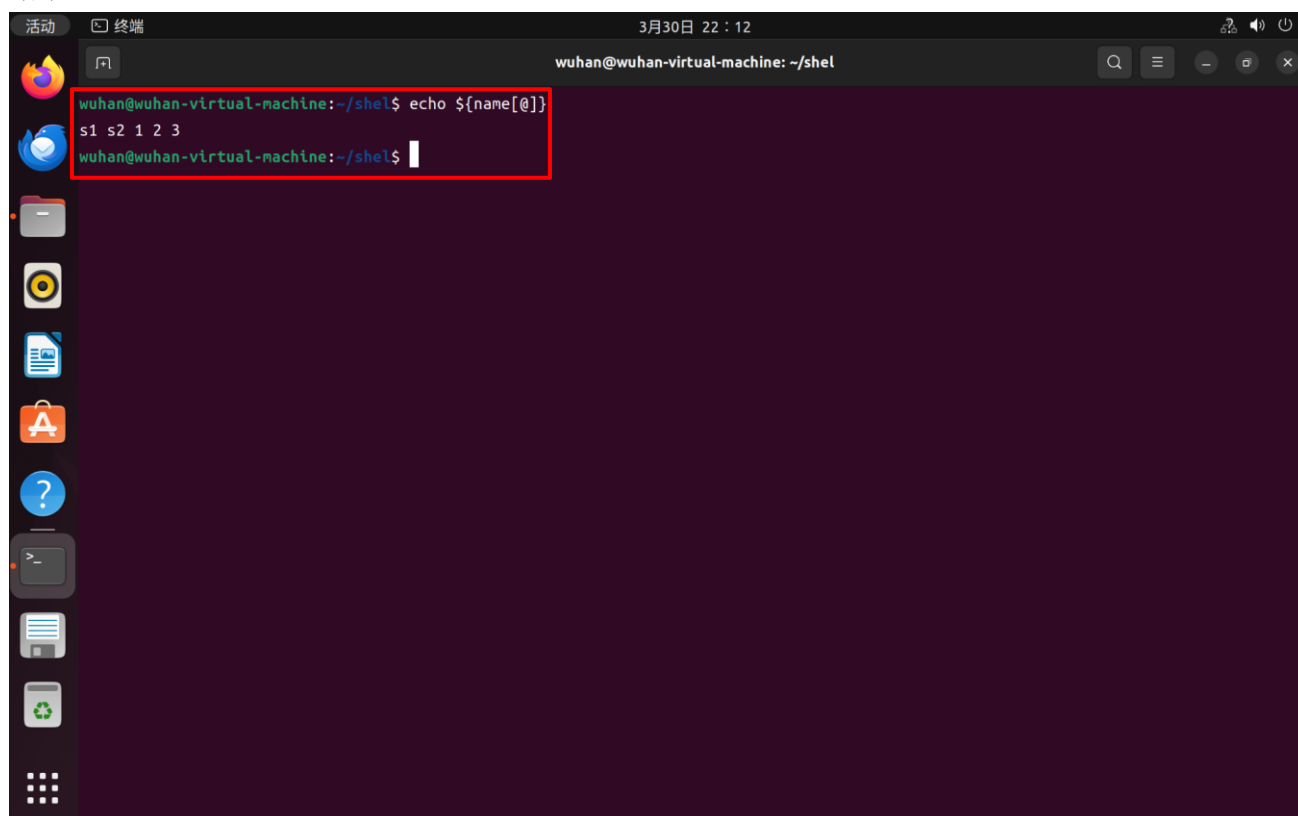
```
wuhan@wuhan-virtual-machine:~/shel$ sc=([3]=30 [5]=50 [7]=70)
wuhan@wuhan-virtual-machine:~/shel$ echo ${sc[*]}
30 50 70
wuhan@wuhan-virtual-machine:~/shel$
```

The output '30 50 70' is highlighted with a red box.

## 16、显示数组 name 全部元素值

命令： `echo ${name[@]}`

结果：



A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a timestamp of '3月30日 22:12'. The terminal shows the following command and output:

```
wuhan@wuhan-virtual-machine:~/shel$ echo ${name[@]}
s1 s2 1 2 3
wuhan@wuhan-virtual-machine:~/shel$
```

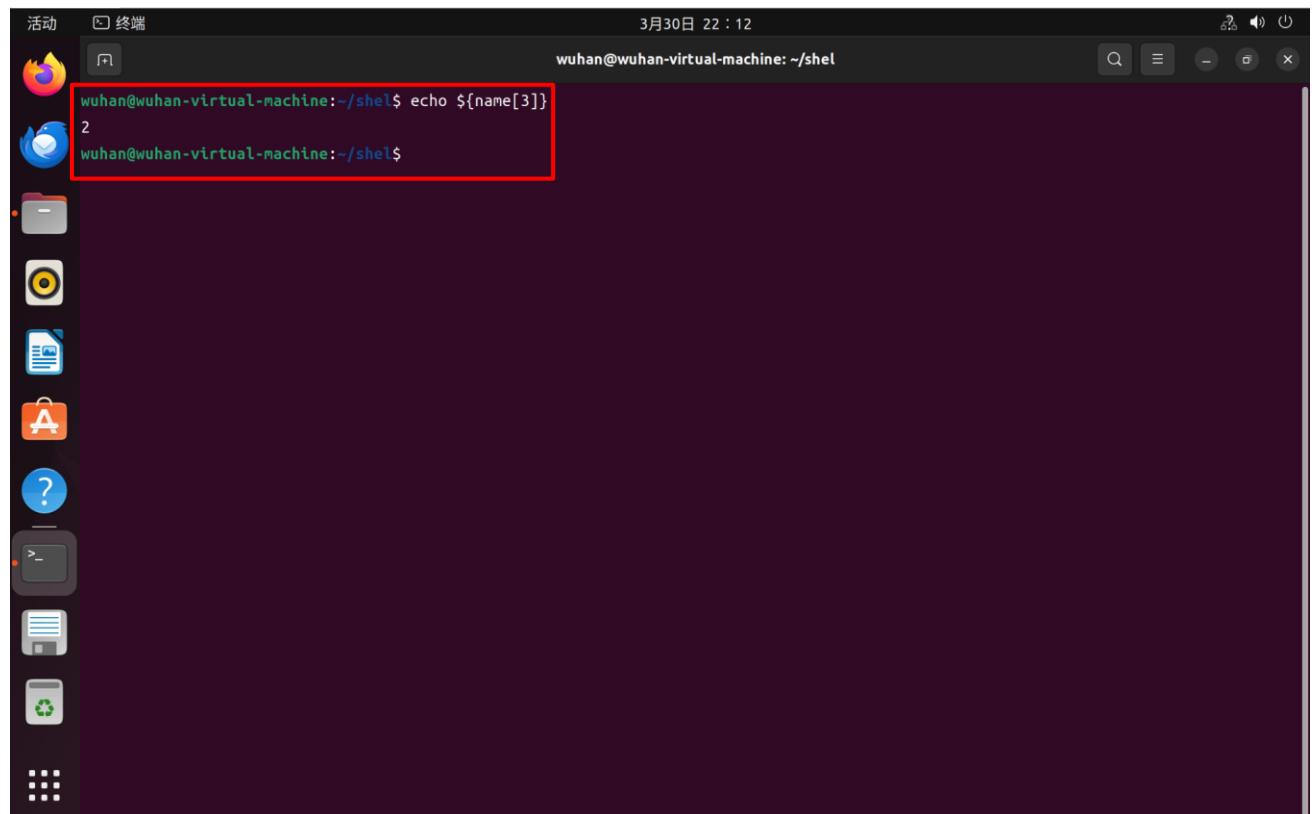
The output 's1 s2 1 2 3' is highlighted with a red box.



17、显示下标为 3 的数组 name 元素值

命令: `echo ${name[3]}`

结果:

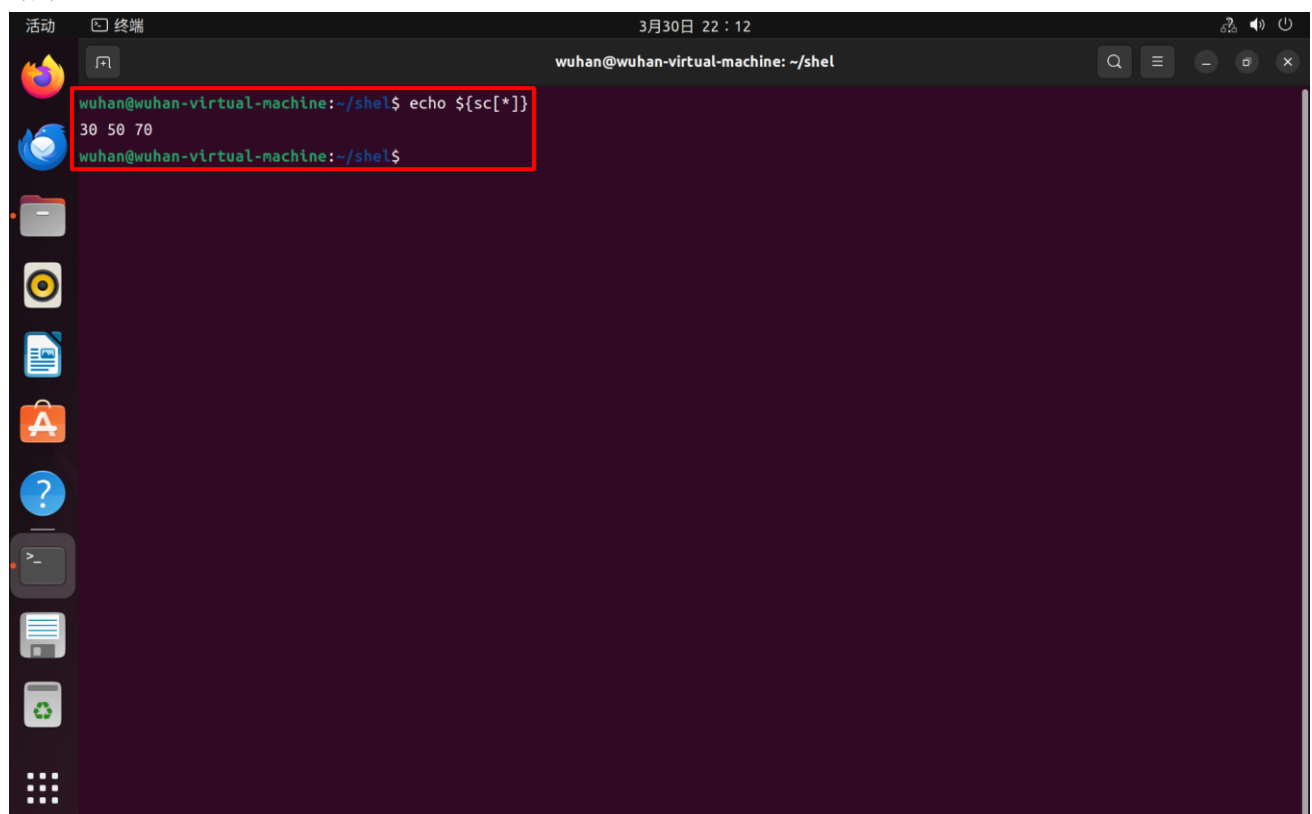


A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a timestamp of '3月30日 22:12'. The terminal shows the command `wuhan@wuhan-virtual-machine:~/shel$ echo ${name[3]}` and its output `2`. The command and output are highlighted with a red box. The terminal interface includes a top bar with '活动' and '终端' tabs, and a left sidebar with various application icons.

18、显示数组 sc 全部元素值

命令: `echo ${sc[*]}`

结果:



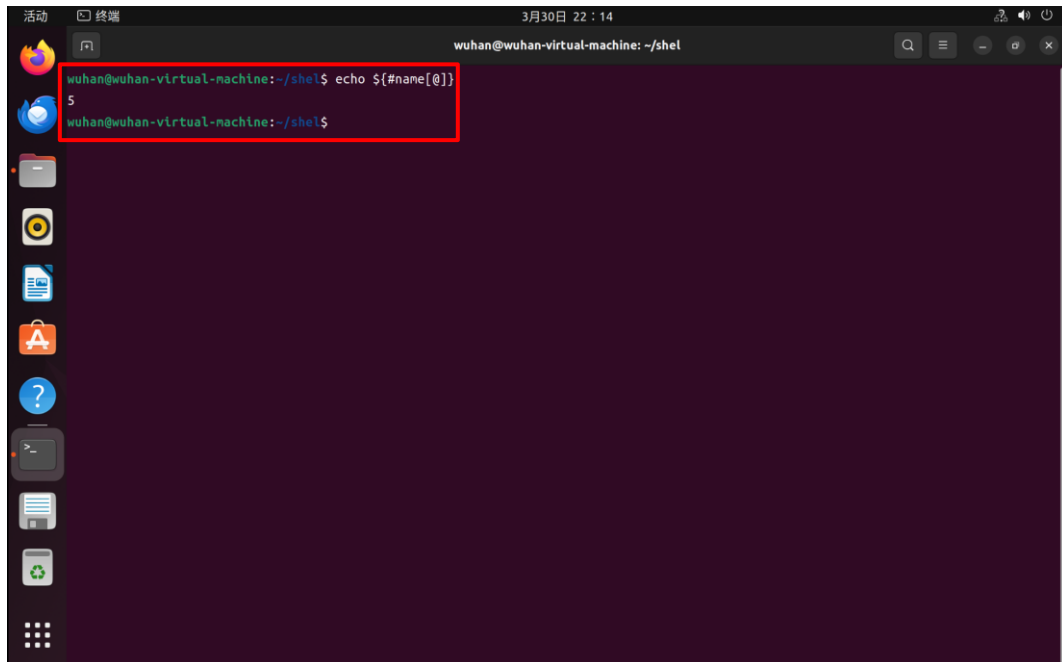
A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a timestamp of '3月30日 22:12'. The terminal shows the command `wuhan@wuhan-virtual-machine:~/shel$ echo ${sc[*]}` and its output `30 50 70`. The command and output are highlighted with a red box. The terminal interface includes a top bar with '活动' and '终端' tabs, and a left sidebar with various application icons.

## 实验 12 批处理操作接口 2：获取数组长度、变量作用域

1、获取数组 name 的长度

命令：echo \${#name[@]}

结果：

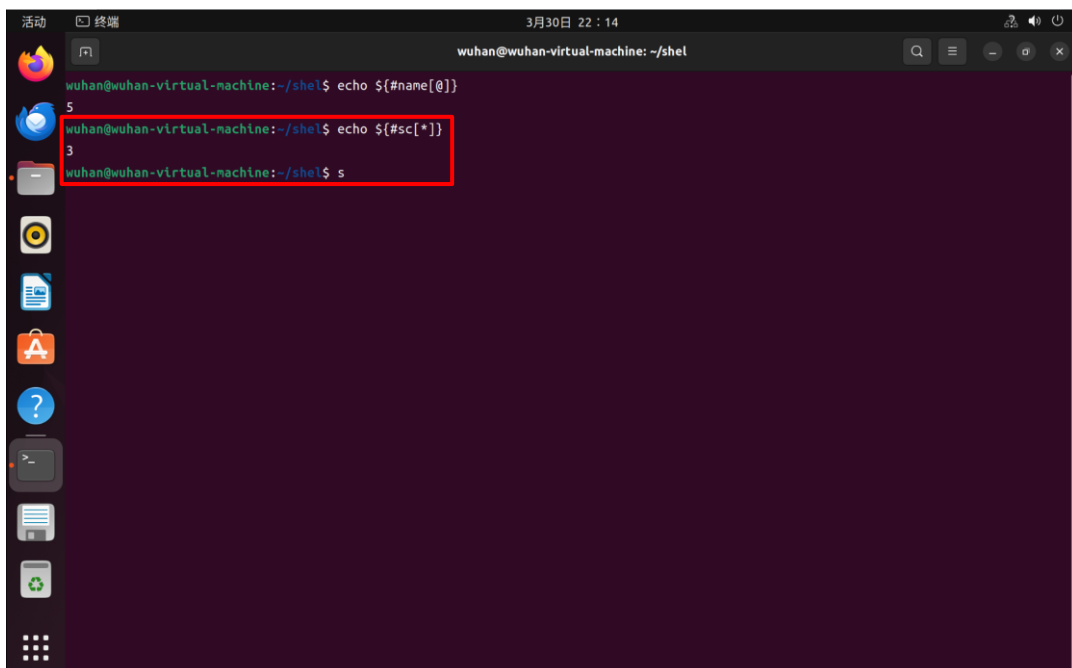


```
wuhan@wuhan-virtual-machine: ~/shel
wuhan@wuhan-virtual-machine:~/shel$ echo ${#name[@]}
5
wuhan@wuhan-virtual-machine:~/shel$
```

2、获取数组 sc 的长度

命令：echo \${#sc[\*]}

结果：

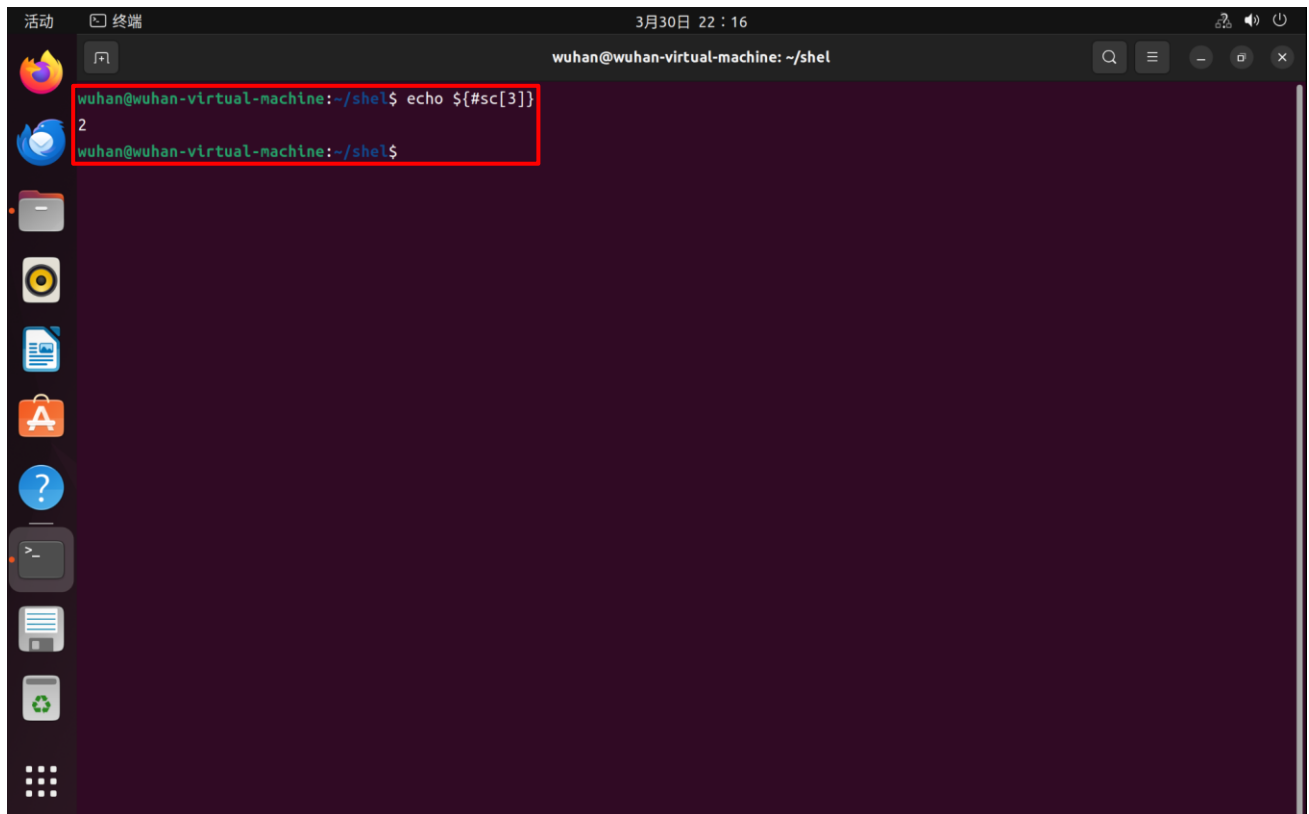


```
wuhan@wuhan-virtual-machine: ~/shel
wuhan@wuhan-virtual-machine:~/shel$ echo ${#name[@]}
5
wuhan@wuhan-virtual-machine:~/shel$ echo ${#sc[*]}
3
wuhan@wuhan-virtual-machine:~/shel$ s
```

### 3、获取数组元素 sc[3]长度

命令：echo \${#sc[3]}

结果：

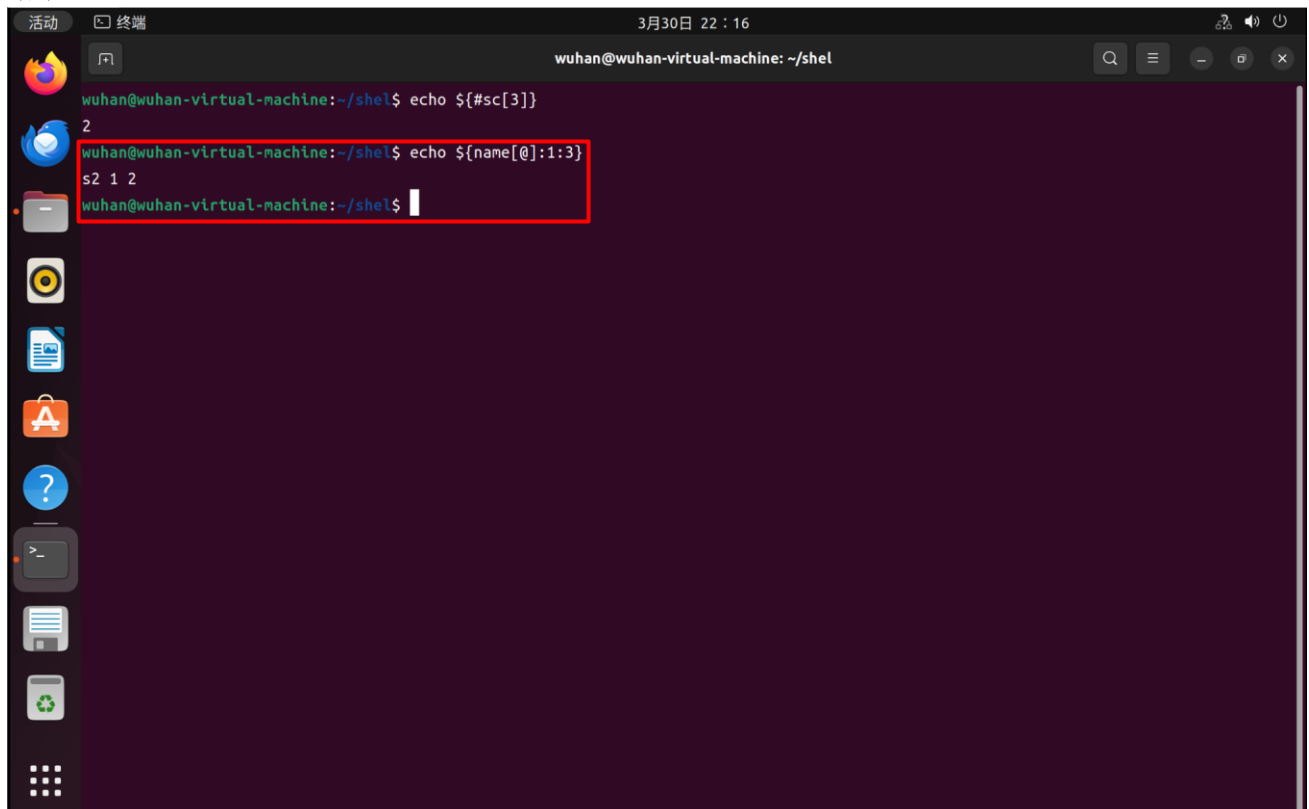


```
wuhan@wuhan-virtual-machine: ~/shel
wuhan@wuhan-virtual-machine:~/shel$ echo ${#sc[3]}
2
wuhan@wuhan-virtual-machine:~/shel$
```

### 4、获取数组 name 自下标 1 起的连续三个元素值

命令：echo \${name[@]:1:3}

结果：

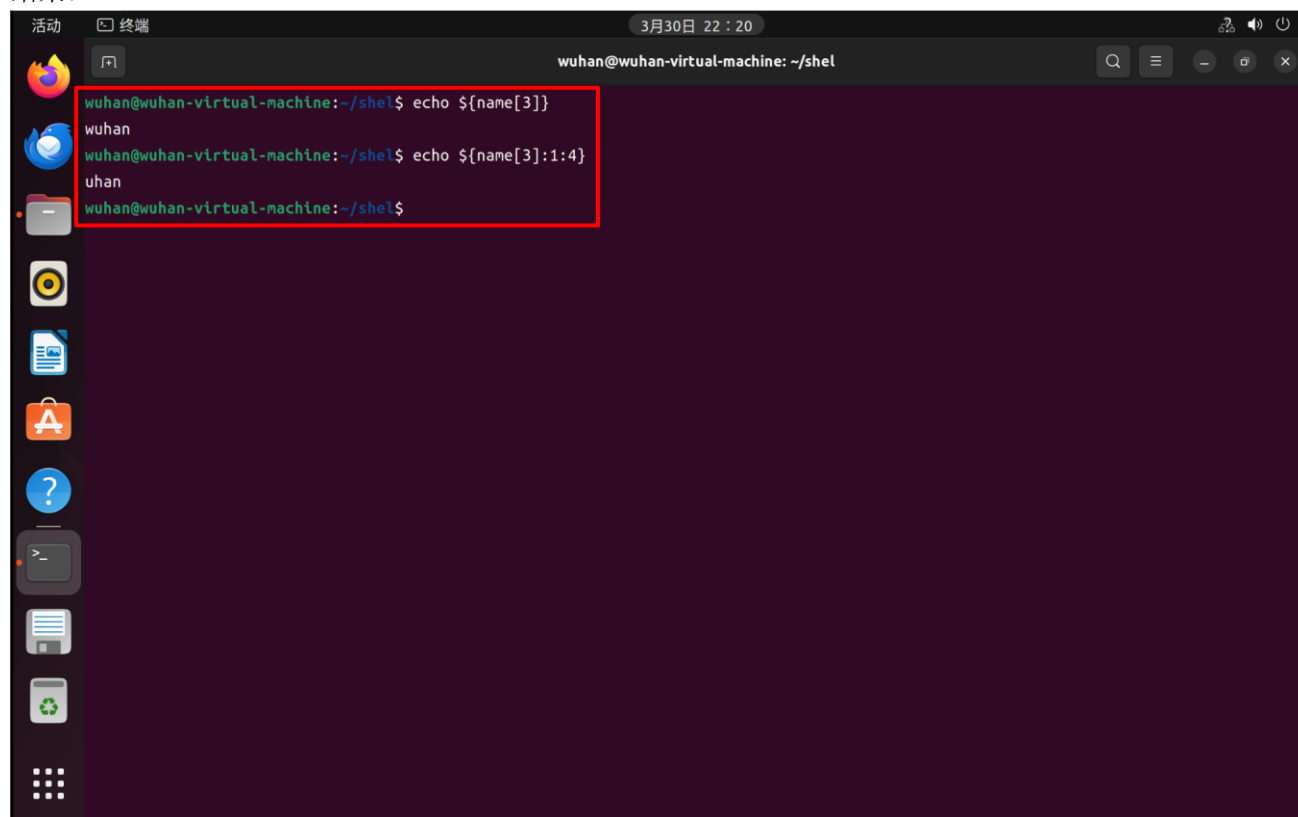


```
wuhan@wuhan-virtual-machine: ~/shel
wuhan@wuhan-virtual-machine:~/shel$ echo ${#sc[3]}
2
wuhan@wuhan-virtual-machine:~/shel$ echo ${name[@]:1:3}
s2 1 2
wuhan@wuhan-virtual-machine:~/shel$
```

5、显示 name 数组元素 name[3]自下标 1 起的连续 4 个字符

命令: `echo ${name[3]:1:4}`

结果:

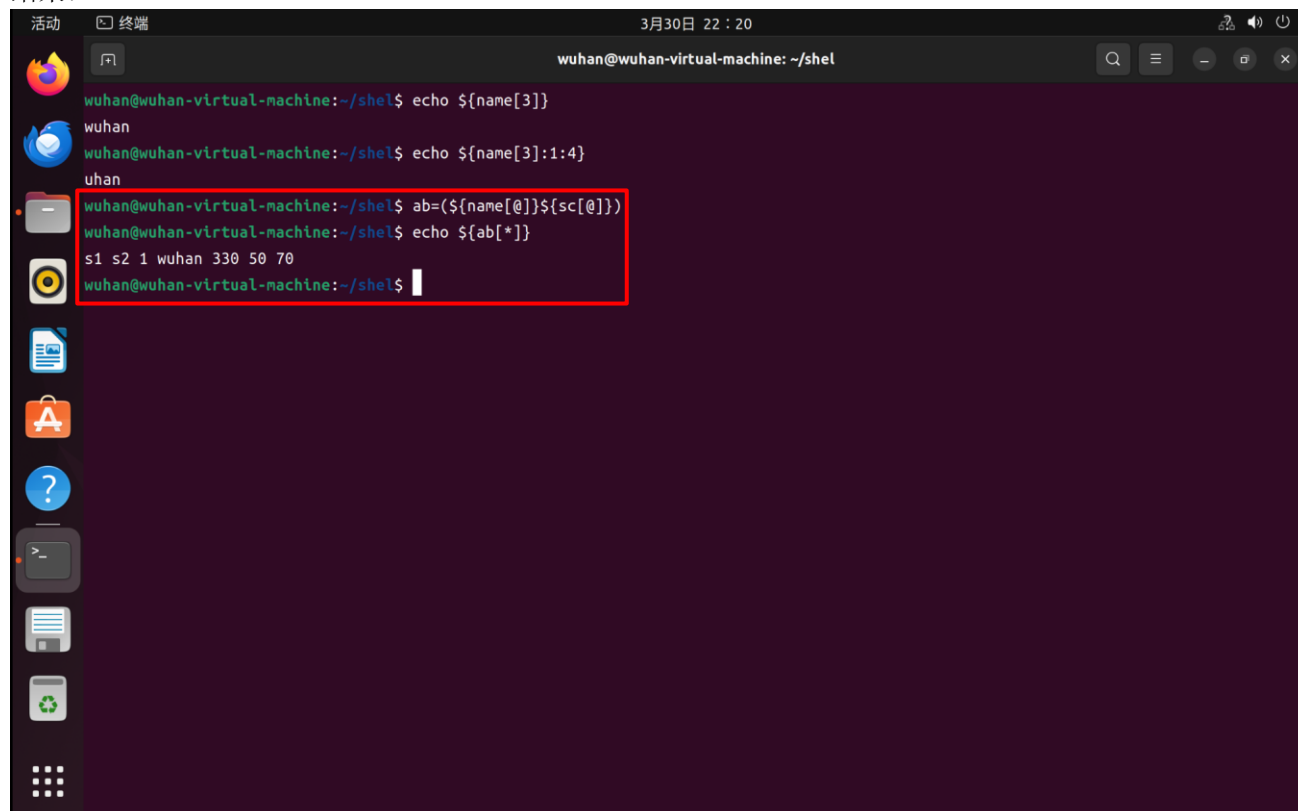


```
wuhan@wuhan-virtual-machine: ~/shel
wuhan@wuhan-virtual-machine:~/shel$ echo ${name[3]}
wuhan
wuhan@wuhan-virtual-machine:~/shel$ echo ${name[3]:1:4}
uhan
wuhan@wuhan-virtual-machine:~/shel$
```

6、连接数组 name 和 sc，将结果赋给 ab

命令: `ab=(${name[@]}${sc[@]})`

结果:

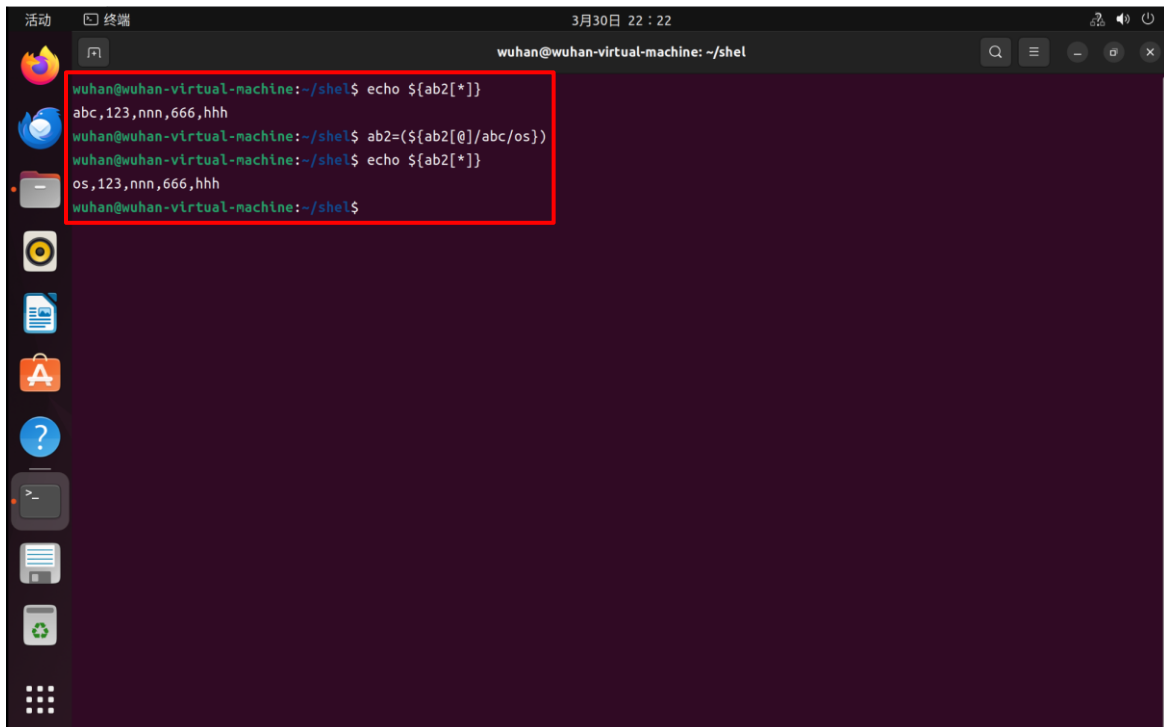


```
wuhan@wuhan-virtual-machine: ~/shel
wuhan@wuhan-virtual-machine:~/shel$ echo ${name[3]}
wuhan
wuhan@wuhan-virtual-machine:~/shel$ echo ${name[3]:1:4}
uhan
wuhan@wuhan-virtual-machine:~/shel$ ab=(${name[@]}${sc[@]})
wuhan@wuhan-virtual-machine:~/shel$ echo ${ab[*]}
s1 s2 1 wuhan 330 50 70
wuhan@wuhan-virtual-machine:~/shel$
```

7、将数组 ab2 中的元素值 abc 替换为 os，并将替换后的数组赋给 ab2

命令：ab2=(\${ab2[@]/abc/os})

结果：

A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a timestamp of '3月30日 22:22'. The terminal shows a sequence of commands and their outputs. The first command is 'echo \${ab2[\*]}', which outputs 'abc,123,nnn,666,hhh'. The second command is 'ab2=(\${ab2[@]/abc/os})'. The third command is 'echo \${ab2[\*]}', which outputs 'os,123,nnn,666,hhh'. The terminal interface includes a sidebar with application icons and window controls at the top.

```
wuhan@wuhan-virtual-machine:~/shel$ echo ${ab2[*]}
abc,123,nnn,666,hhh
wuhan@wuhan-virtual-machine:~/shel$ ab2=(${ab2[@]/abc/os})
wuhan@wuhan-virtual-machine:~/shel$ echo ${ab2[*]}
os,123,nnn,666,hhh
wuhan@wuhan-virtual-machine:~/shel$
```

8、建立文件 Namespace1.sh，内容为：

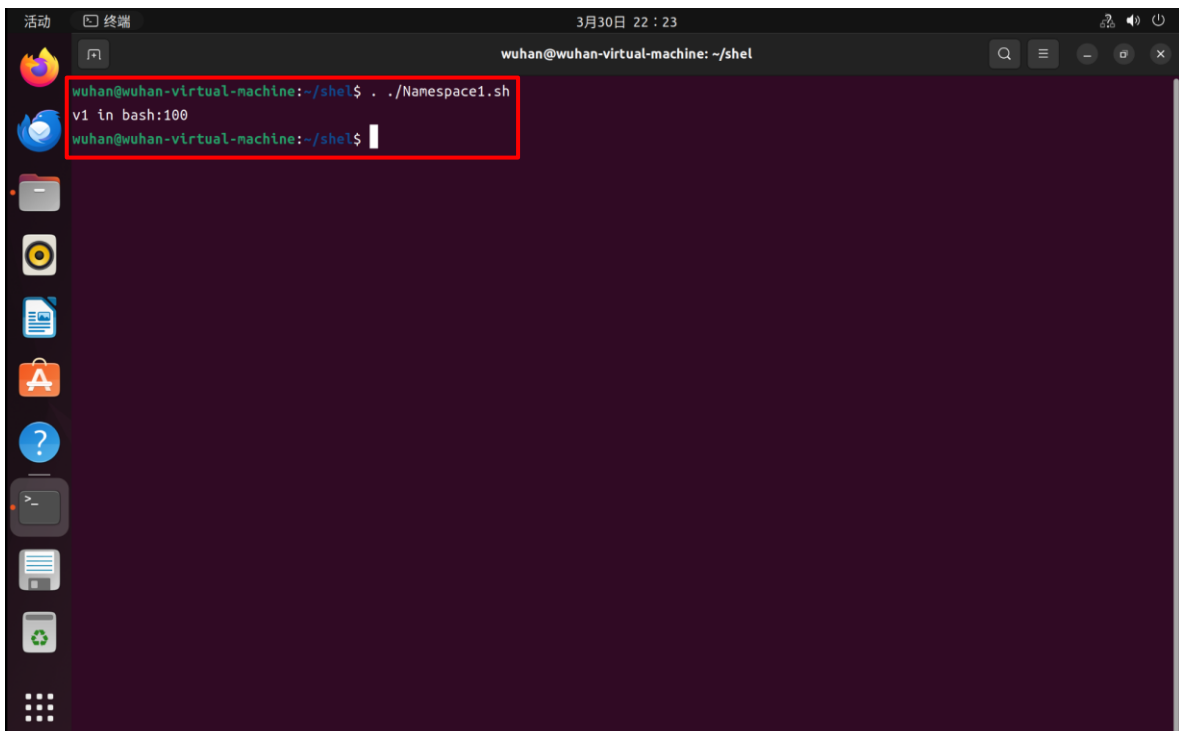
#!/bin/bash

v1=100

echo v1 in \${0}:\${v1}

运行：../Namespace1.sh

结果：

A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a timestamp of '3月30日 22:23'. The terminal shows the command './Namespace1.sh' being executed, which results in the output 'v1 in bash:100'. The terminal interface includes a sidebar with application icons and window controls at the top.

```
wuhan@wuhan-virtual-machine:~/shel$ ./Namespace1.sh
v1 in bash:100
wuhan@wuhan-virtual-machine:~/shel$
```

9、建立文件 Namespace3.sh，其内容为：

```
#!/bin/bash
```

```
function ch_var(){
```

```
    echo 在函数中修改前 v1 的值:$v1
```

```
    v1=200
```

```
}
```

```
v1=100
```

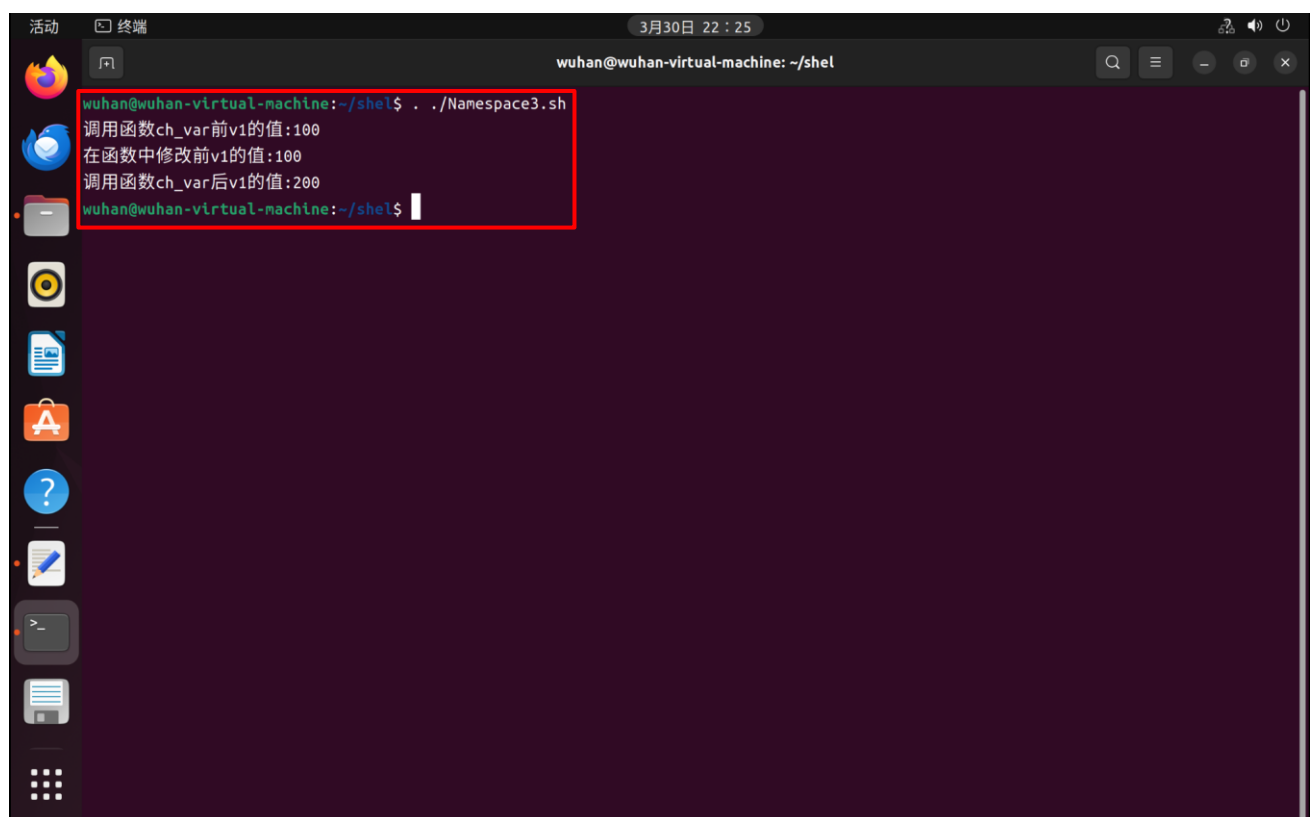
```
echo 调用函数 ch_var 前 v1 的值:$v1
```

```
ch_var
```

```
echo 调用函数 ch_var 后 v1 的值:$v1
```

```
运行: ./Namespace3.sh
```

结果：



The image shows a terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a timestamp of '3月30日 22:25'. The terminal displays the output of running the script './Namespace3.sh'. The output consists of four lines: '调用函数ch\_var前v1的值:100', '在函数中修改前v1的值:100', '调用函数ch\_var后v1的值:200', and a prompt 'wuhan@wuhan-virtual-machine: ~/shel\$'. The first three lines are enclosed in a red rectangular box.

```
wuhan@wuhan-virtual-machine: ~/shel$ ./Namespace3.sh
调用函数ch_var前v1的值:100
在函数中修改前v1的值:100
调用函数ch_var后v1的值:200
wuhan@wuhan-virtual-machine: ~/shel$
```

10、建立文件 Namespace4.sh，其内容为：

```
#!/bin/bash
```

```
function ch_var(){
```

```
    v2=200  #v2 默认是 global 类型
```

```
}
```

```
echo 尚未调用函数时 v2 的值:$v2
```

```
ch_var
```

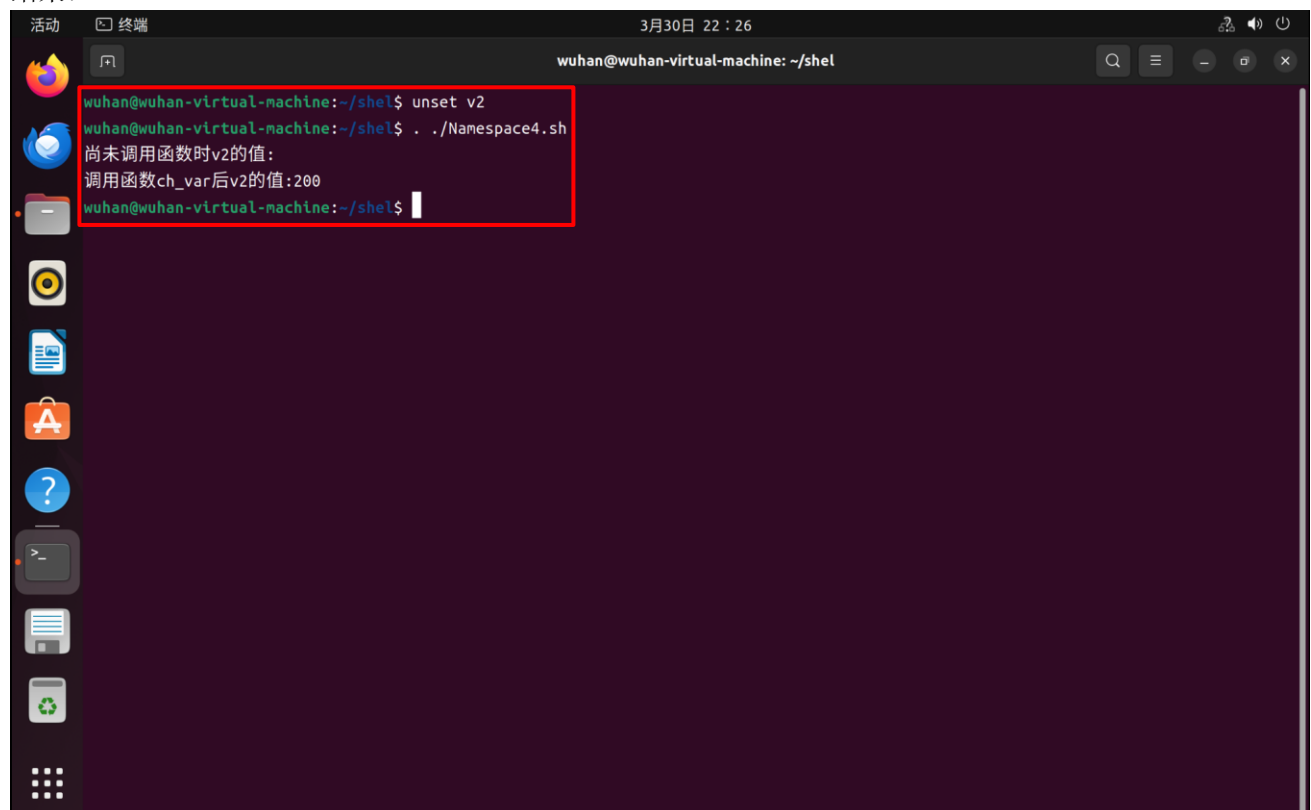
```
echo 调用函数 ch_var 后 v2 的值:$v2
```

运行：

```
unset v2
```

```
. ./Namespace4.sh
```

结果：



The image shows a terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a timestamp of '3月30日 22:26'. The terminal output is as follows:

```
wuhan@wuhan-virtual-machine:~/shel$ unset v2
wuhan@wuhan-virtual-machine:~/shel$ . ./Namespace4.sh
尚未调用函数时v2的值:
调用函数ch_var后v2的值:200
wuhan@wuhan-virtual-machine:~/shel$
```

The first two lines of output are highlighted with a red box. The terminal window has a sidebar on the left with various application icons and a top bar with window controls.

11、建立文件 Namespace5.sh:

```
#!/bin/bash
```

```
function ch_var(){
```

```
    local v3=200  #v3 定义为 local 类型
```

```
    echo 在函数中执行时 v3 的值:$v3
```

```
}
```

```
echo 尚未调用函数时 v3 的值:$v3
```

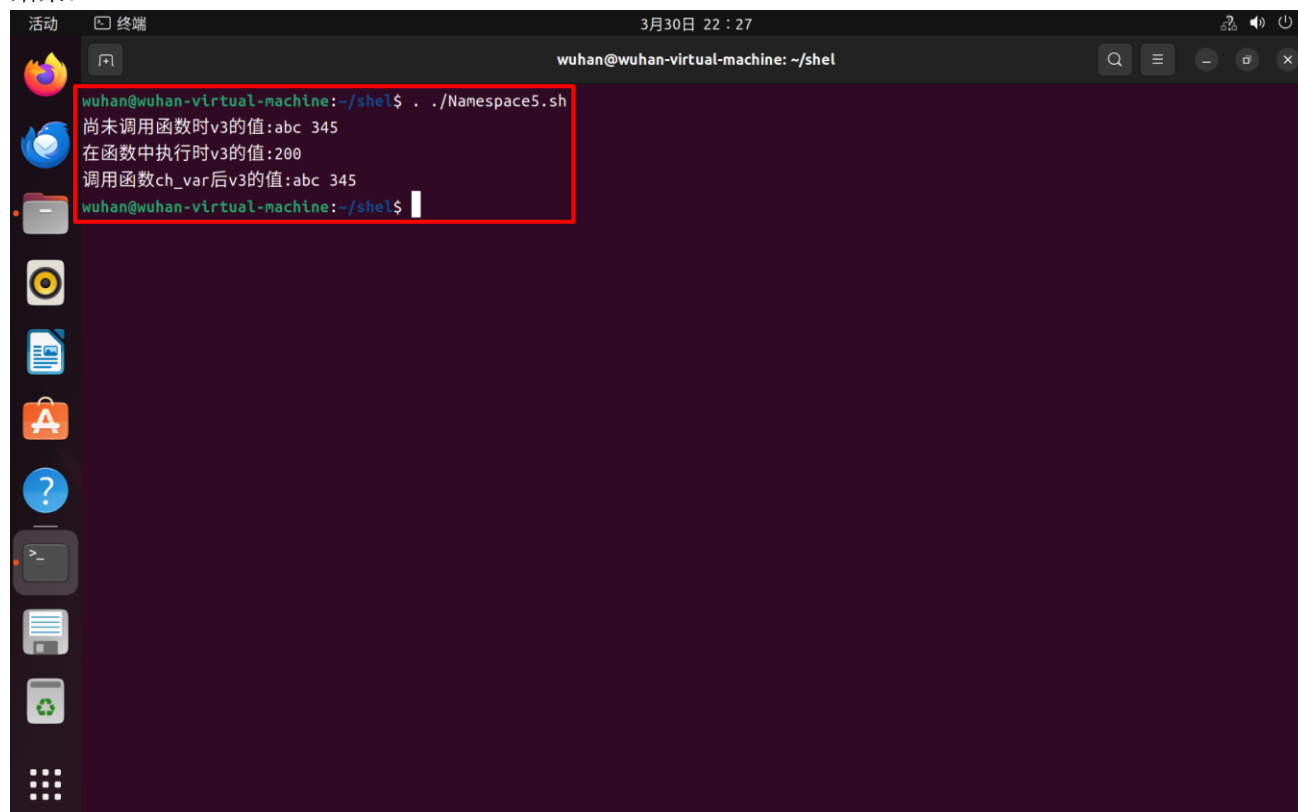
```
ch_var
```

```
echo 调用函数 ch_var 后 v3 的值:$v3
```

运行:

```
./Namespace5.sh
```

结果:



The image shows a terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel'. The terminal output is as follows:

```
wuhan@wuhan-virtual-machine:~/shel$ ./Namespace5.sh
尚未调用函数时v3的值:abc 345
在函数中执行时v3的值:200
调用函数ch_var后v3的值:abc 345
wuhan@wuhan-virtual-machine:~/shel$
```

The output matches the script's logic: it prints the initial value of v3 (abc 345), then the value inside the function (200), and finally the value after the function call (abc 345).



12、建立文件 Namespace6.sh:

```
#!/bin/bash
```

```
sfunc()
```

```
{
```

```
    echo "参数: $@"
```

```
}
```

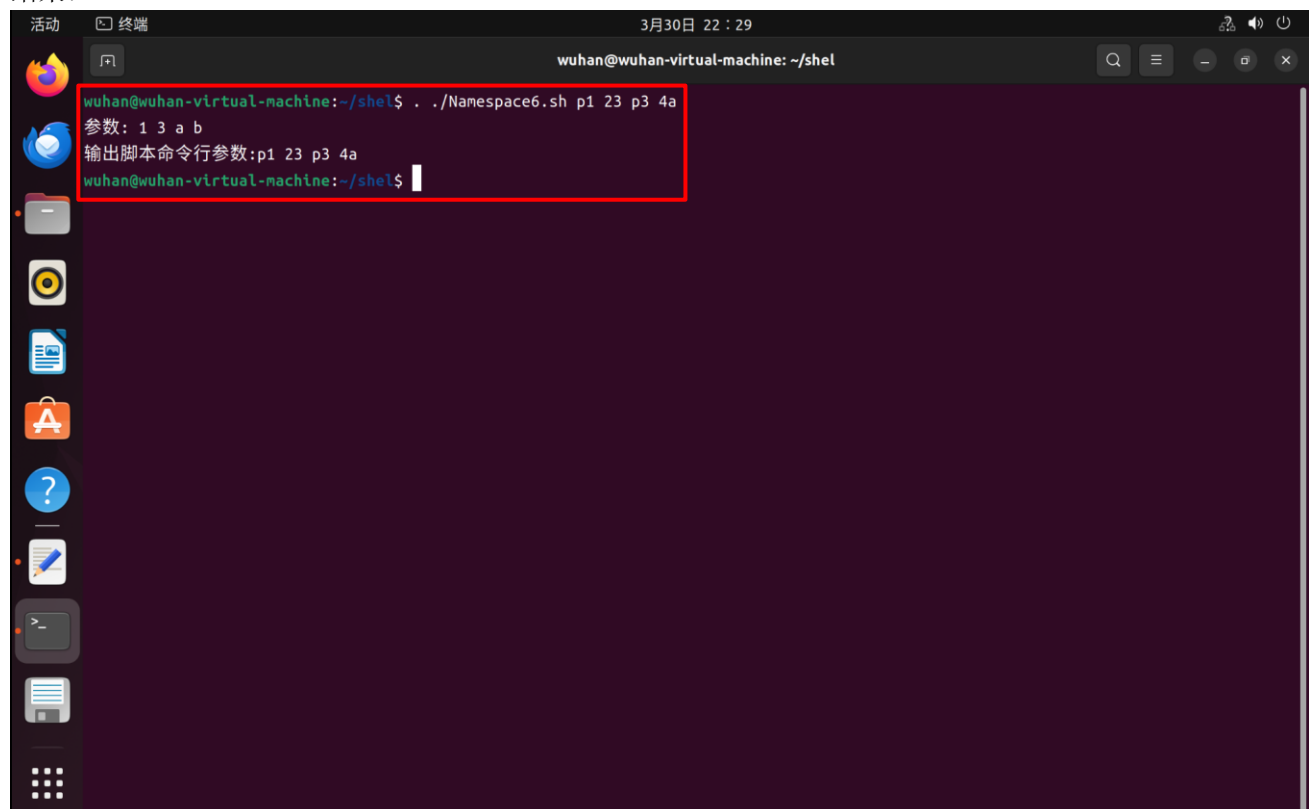
```
sfunc 1 3 a b
```

```
echo 输出脚本命令行参数:$@
```

运行:

```
./Namespace6.sh p1 23 p3 4a
```

结果:



A terminal window titled "wuhan@wuhan-virtual-machine: ~/shel" with a timestamp of "3月30日 22:29". The terminal shows the execution of the script `./Namespace6.sh p1 23 p3 4a`. The output is displayed in two lines: "参数: 1 3 a b" and "输出脚本命令行参数:p1 23 p3 4a". The prompt `wuhan@wuhan-virtual-machine:~/shel$` is visible at the bottom of the terminal window.

### 13、建立文件 Namespace7.sh:

```
#!/bin/bash
```

```
sfunc()
```

```
{  
    echo 在函数中输出全局变量 v4 的值: $v4    #行 1  
    local v4=200abc                            #行 2  
    echo 在函数中输出局部变量 v4 的值: $v4    #行 3  
}
```

```
v4=100                                          #行 4
```

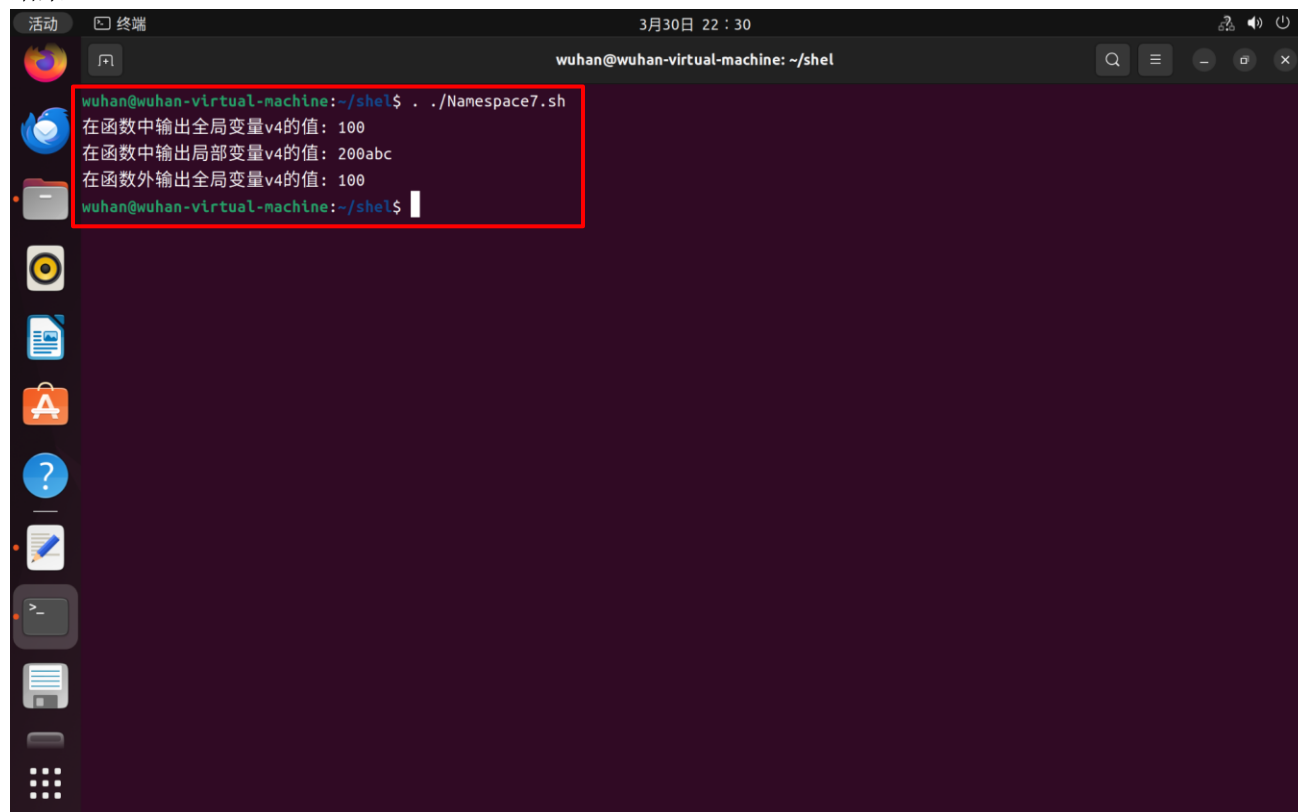
```
sfunc
```

```
echo 在函数外输出全局变量 v4 的值: $v4      #行 5
```

运行:

```
./Namespace7.sh
```

结果:



A terminal window titled 'wuhan@wuhan-virtual-machine: ~/shel' with a timestamp of '3月30日 22:30'. The terminal shows the execution of the script './Namespace7.sh'. The output is as follows:

```
wuhan@wuhan-virtual-machine:~/shel$ ./Namespace7.sh  
在函数中输出全局变量v4的值: 100  
在函数中输出局部变量v4的值: 200abc  
在函数外输出全局变量v4的值: 100  
wuhan@wuhan-virtual-machine:~/shel$
```

The output lines are highlighted with a red box in the original image.