EPAM University Programs

DevOps external course

Module 4 Linux & Bash Essentials

TASK 4.6

1. *User management*. Here we suppose there are at least two users, namely, root and guest.

(i) Create a new user *user*

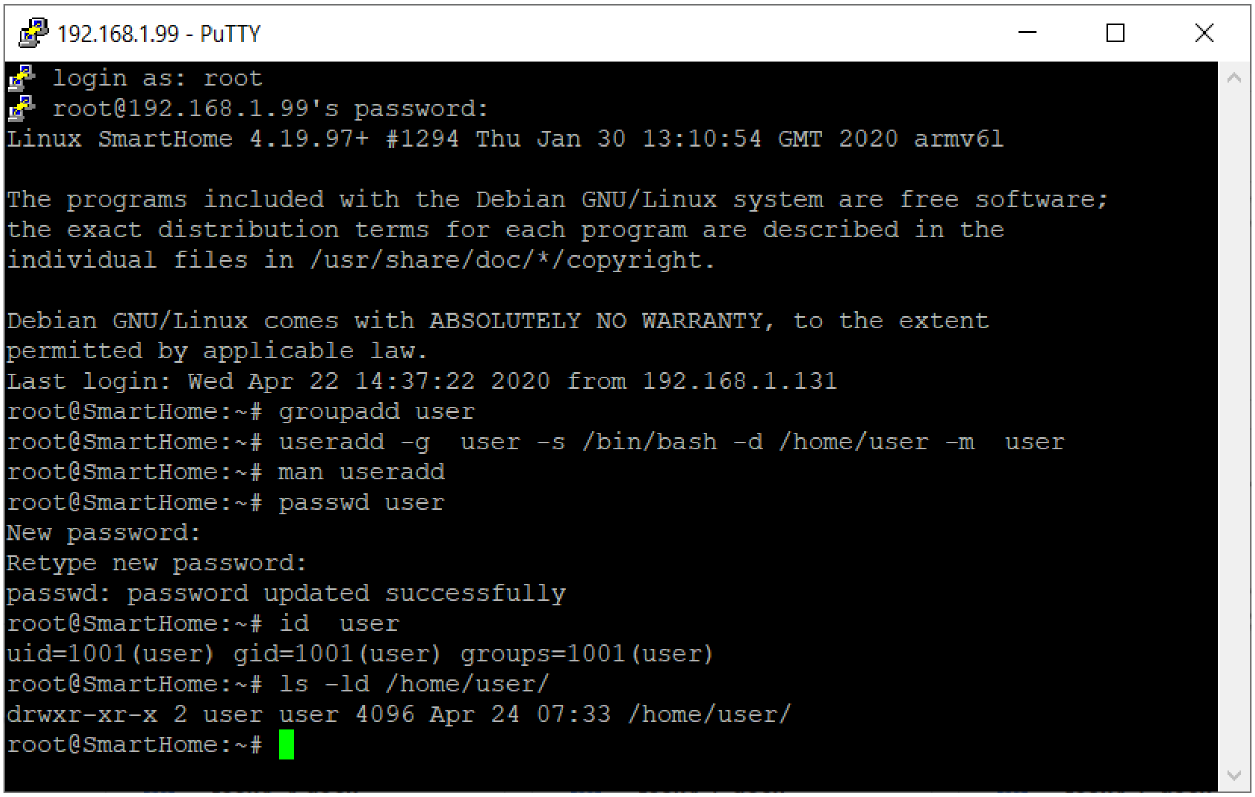
groupadd user (add group user)

useradd -g user -s /bin/bash -d /home/user -m user (added user "user with group user, with shell=bash, homedir /home/user will be created)

passwd user (change password for user)

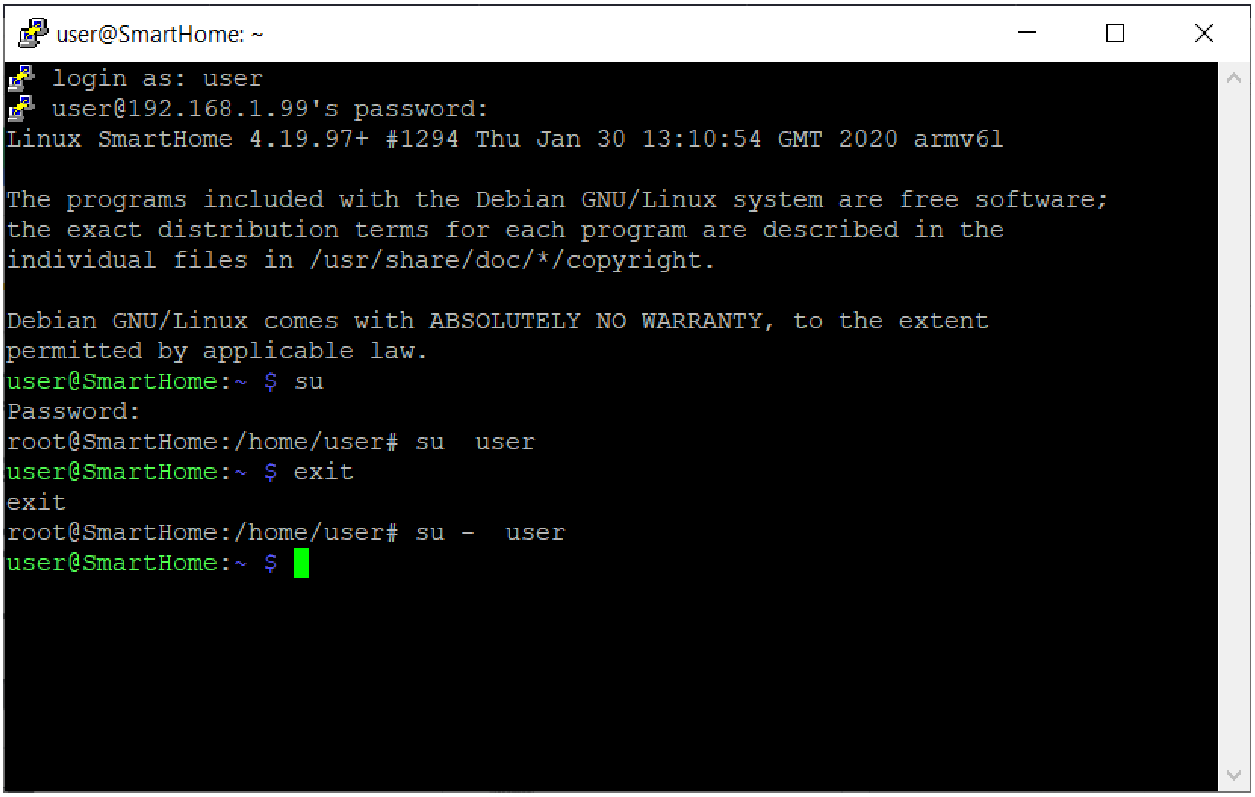
id user (show "user" ID)

ls -ld /home/user (show permission for the folder of user)



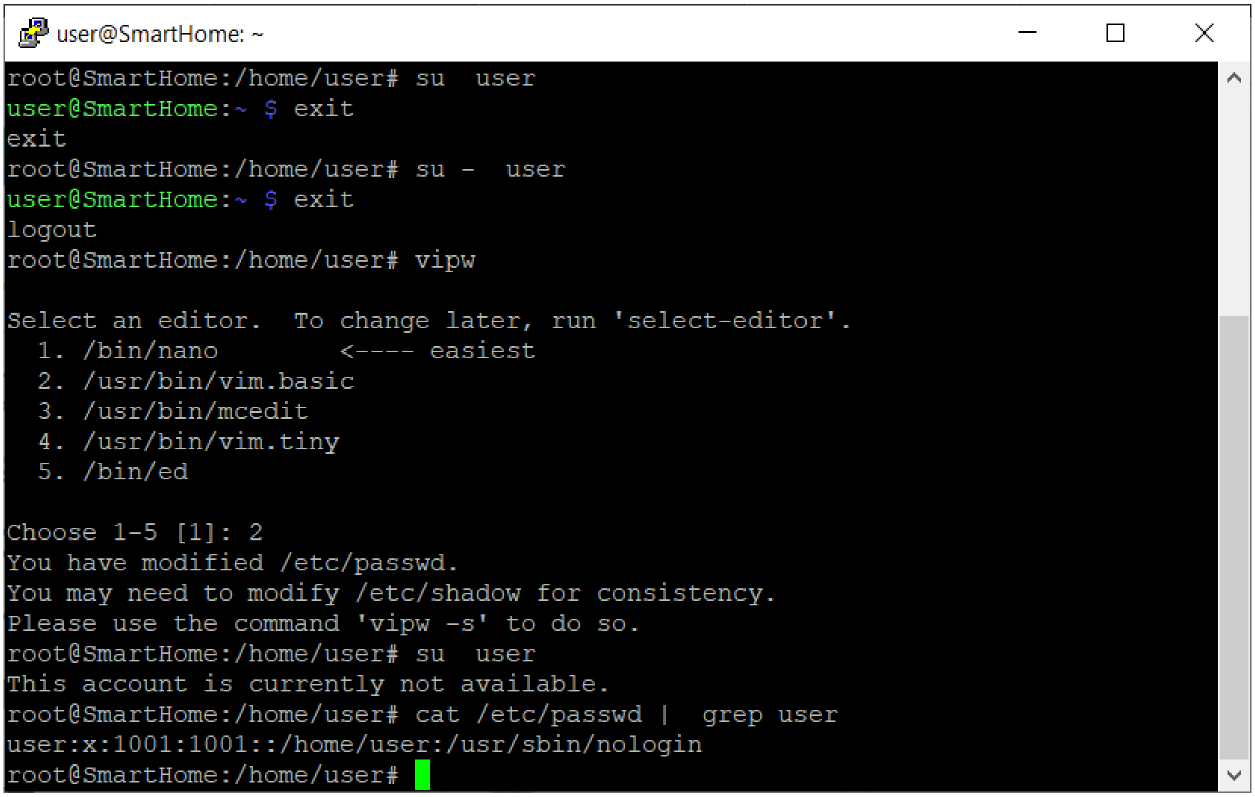
```
login as: root
root@192.168.1.99's password:
Linux SmartHome 4.19.97+ #1294 Thu Jan 30 13:10:54 GMT 2020 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 22 14:37:22 2020 from 192.168.1.131
root@SmartHome:~# groupadd user
root@SmartHome:~# useradd -g  user -s /bin/bash -d /home/user -m  user
root@SmartHome:~# man useradd
root@SmartHome:~# passwd user
New password:
Retype new password:
passwd: password updated successfully
root@SmartHome:~# id  user
uid=1001(user) gid=1001(user) groups=1001(user)
root@SmartHome:~# ls -ld /home/user/
drwxr-xr-x 2 user user 4096 Apr 24 07:33 /home/user/
root@SmartHome:~#
```

(ii) Log in to the system as "user" (hint use **su**). (I connected directly and via su -user)



```
login as: user
user@192.168.1.99's password:
Linux SmartHome 4.19.97+ #1294 Thu Jan 30 13:10:54 GMT 2020 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@SmartHome:~ $ su
Password:
root@SmartHome:/home/user# su  user
user@SmartHome:~ $ exit
exit
root@SmartHome:/home/user# su -  user
user@SmartHome:~ $
```

(ii) Edit **/etc/passwd** to prevent user *user* from logging in to the system. (I use vipw command to modify /etc/password and update /etc/shadow/in the same time and change shell to /usr/sbin/nologin)

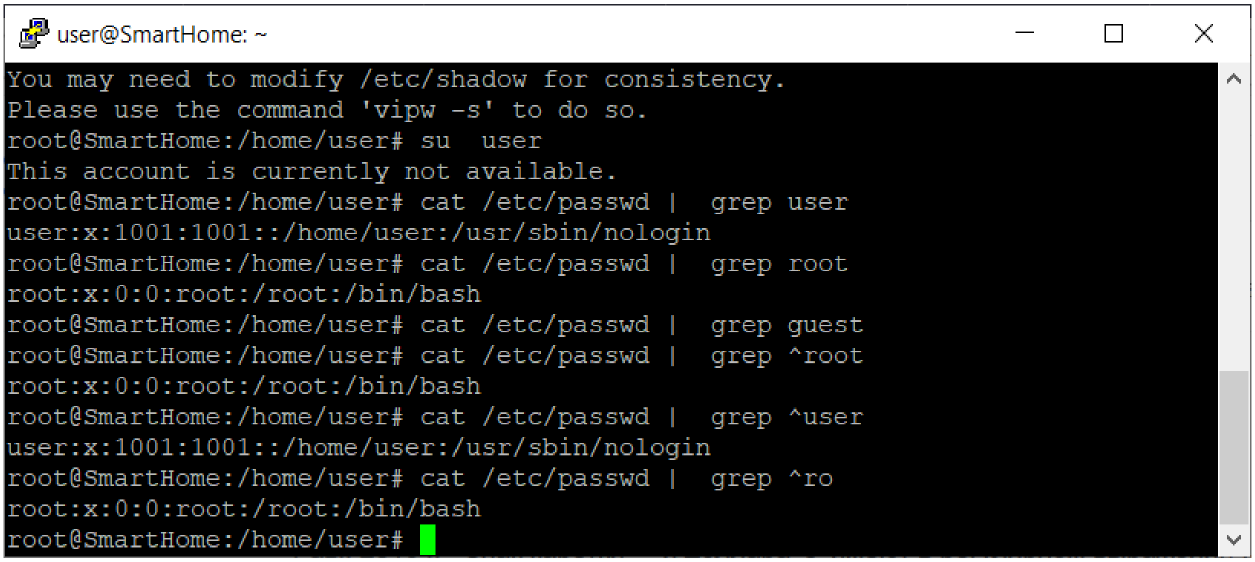


```
root@SmartHome:/home/user# su  user
user@SmartHome:~ $ exit
exit
root@SmartHome:/home/user# su -  user
user@SmartHome:~ $ exit
logout
root@SmartHome:/home/user# vipw

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/mcedit
  4. /usr/bin/vim.tiny
  5. /bin/ed

Choose 1-5 [1]: 2
You have modified /etc/passwd.
You may need to modify /etc/shadow for consistency.
Please use the command 'vipw -s' to do so.
root@SmartHome:/home/user# su  user
This account is currently not available.
root@SmartHome:/home/user# cat /etc/passwd |  grep user
user:x:1001:1001::/home/user:/usr/sbin/nologin
root@SmartHome:/home/user#
```

2. *Content of* **/etc/passwd** and **/etc/group**.

(i) Look through **/etc/passwd** and **/etc/group** (hint: use **less** or **cat**).

(ii) Get data from **/etc/passwd** and **/etc/group** about users: *root*, *guest*, *user* (hint: filter by **grep**). (guest user is not present in my sysem)

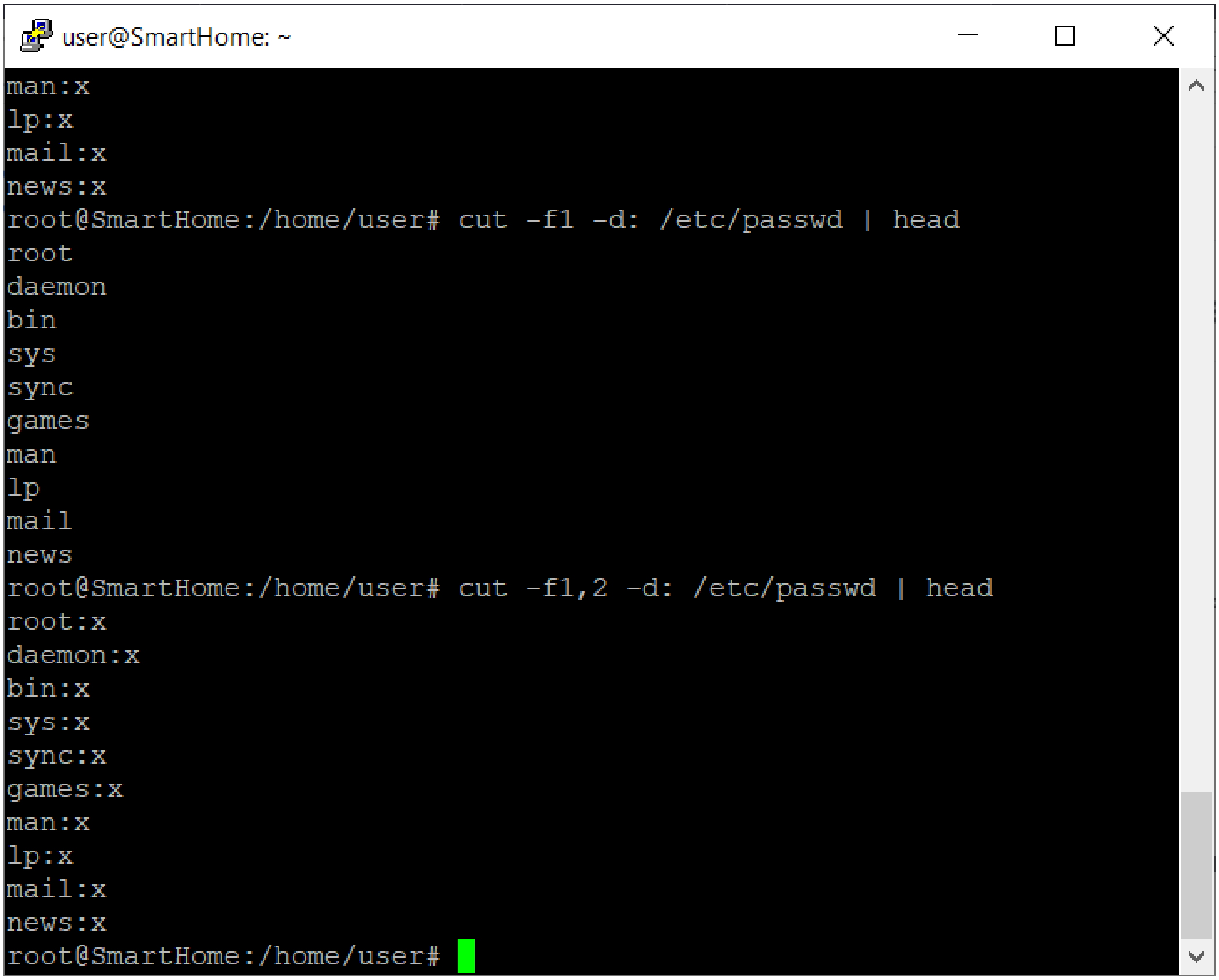


```
You may need to modify /etc/shadow for consistency.
Please use the command 'vipw -s' to do so.
root@SmartHome:/home/user# su  user
This account is currently not available.
root@SmartHome:/home/user# cat /etc/passwd |  grep user
user:x:1001:1001::/home/user:/usr/sbin/nologin
root@SmartHome:/home/user# cat /etc/passwd |  grep root
root:x:0:0:root:/root:/bin/bash
root@SmartHome:/home/user# cat /etc/passwd |  grep guest
root@SmartHome:/home/user# cat /etc/passwd |  grep ^root
root:x:0:0:root:/root:/bin/bash
root@SmartHome:/home/user# cat /etc/passwd |  grep ^user
user:x:1001:1001::/home/user:/usr/sbin/nologin
root@SmartHome:/home/user# cat /etc/passwd |  grep ^ro
root:x:0:0:root:/root:/bin/bash
root@SmartHome:/home/user# 
```

(iii)Parse **/etc/passwd** and **/etc/group** with cut. (I used pipe to head to display 10 string only to reduce output in screenshots)

cut -f1 -d: /etc/passwd (show column 1 (username)-d: means delimiter is “:”)

cut -f1,2 -d: /etc/passwd (show columns 1 and 2 (username and password (in shadow will be real hash, in passwd file – “x” only) -d: means delimiter is :)

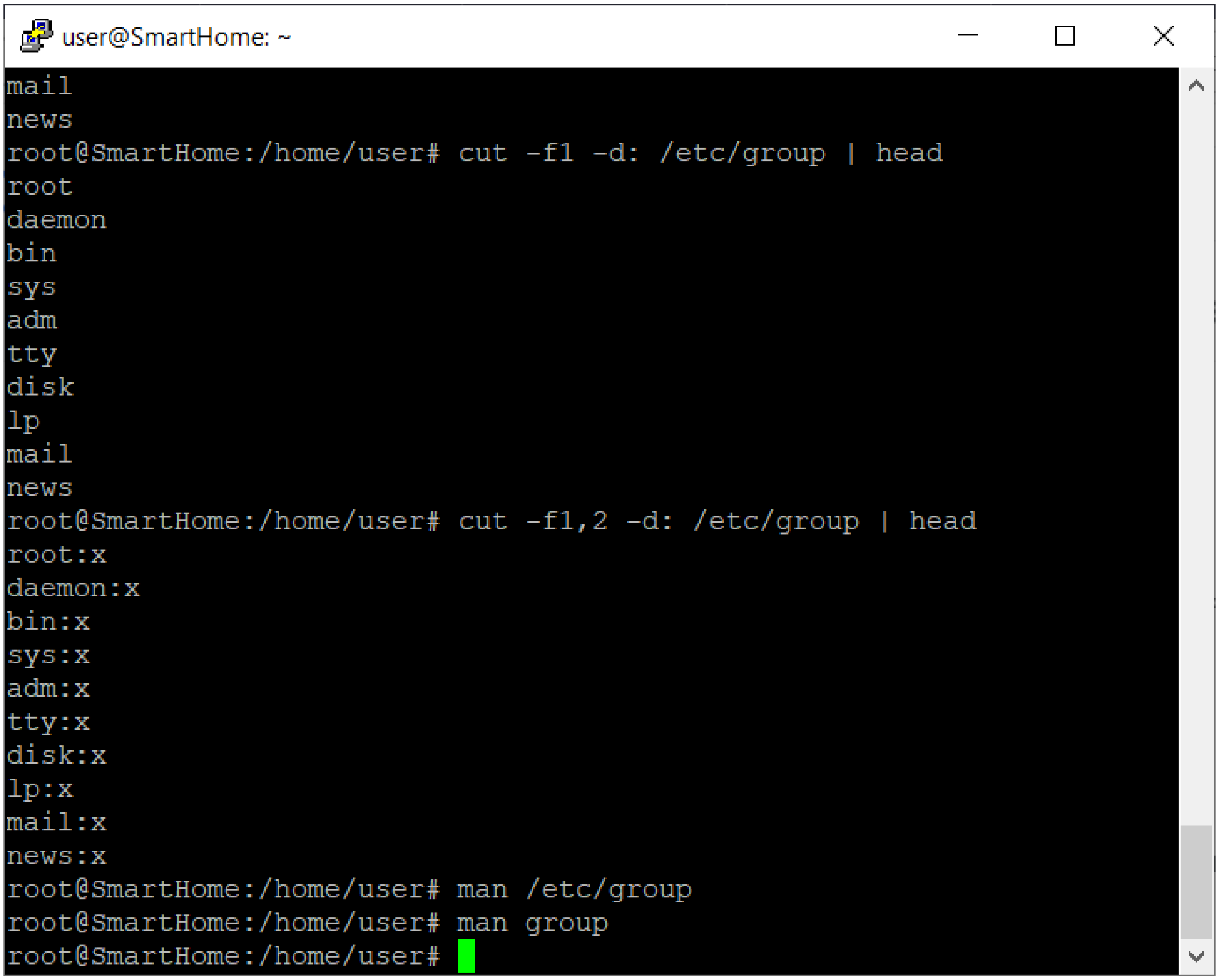


```
man:x
lp:x
mail:x
news:x
root@SmartHome:/home/user# cut -f1 -d: /etc/passwd | head
root
daemon
bin
sys
sync
games
man
lp
mail
news
root@SmartHome:/home/user# cut -f1,2 -d: /etc/passwd | head
root:x
daemon:x
bin:x
sys:x
sync:x
games:x
man:x
lp:x
mail:x
news:x
root@SmartHome:/home/user#
```

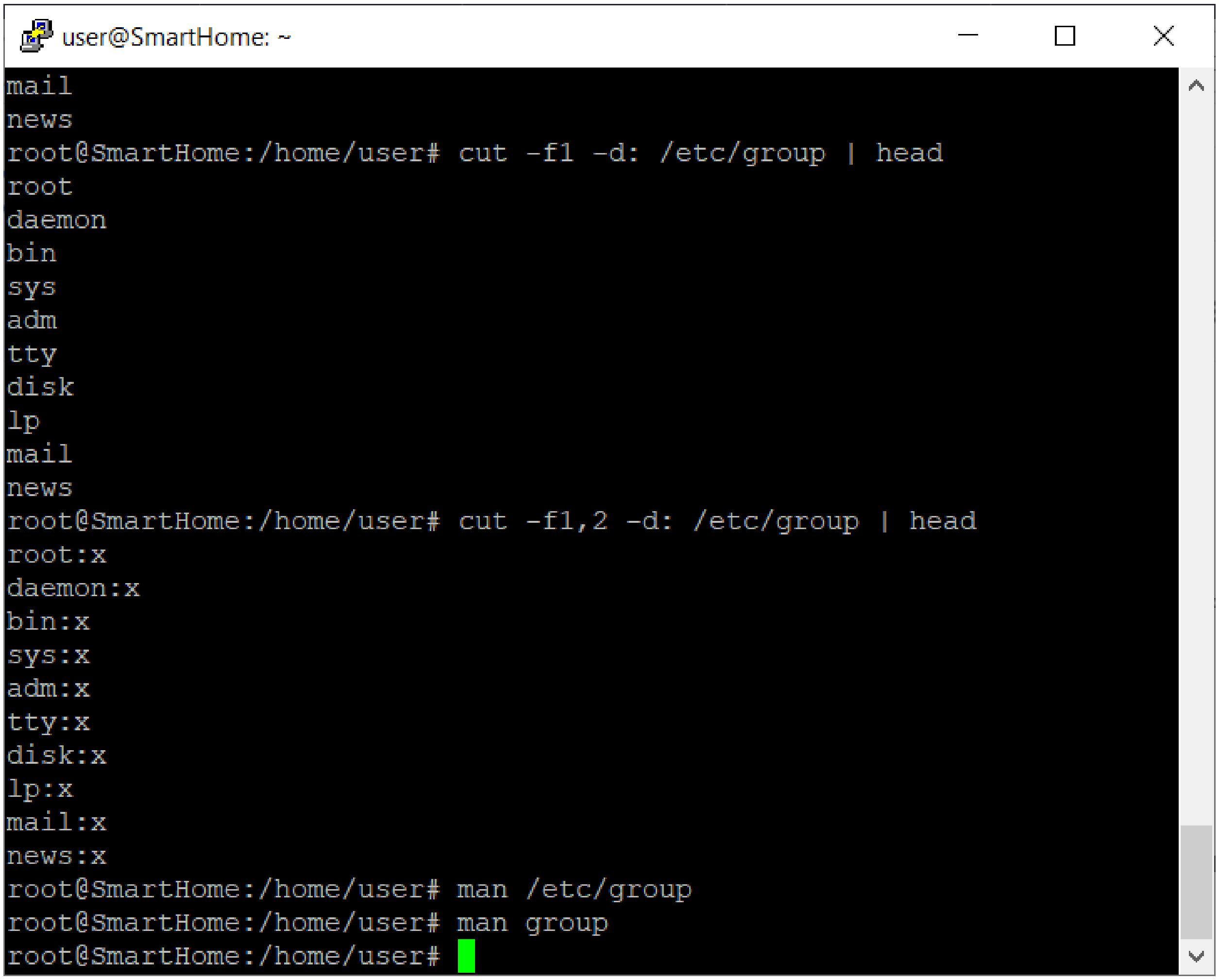
cut -f1,7 -d: /etc/passwd (show username and shell of user)

cut -f1 -d: /etc/group (show group names)

cut -f1,2 -d: /etc/group (show group names and x like group password)



```
mail
news
root@SmartHome:/home/user# cut -f1 -d: /etc/group | head
root
daemon
bin
sys
adm
tty
disk
lp
mail
news
root@SmartHome:/home/user# cut -f1,2 -d: /etc/group | head
root:x
daemon:x
bin:x
sys:x
adm:x
tty:x
disk:x
lp:x
mail:x
news:x
root@SmartHome:/home/user# man /etc/group
root@SmartHome:/home/user# man group
root@SmartHome:/home/user# 
```



```
mail
news
root@SmartHome:/home/user# cut -f1 -d: /etc/group | head
root
daemon
bin
sys
adm
tty
disk
lp
mail
news
root@SmartHome:/home/user# cut -f1,2 -d: /etc/group | head
root:x
daemon:x
bin:x
sys:x
adm:x
tty:x
disk:x
lp:x
mail:x
news:x
root@SmartHome:/home/user# man /etc/group
root@SmartHome:/home/user# man group
root@SmartHome:/home/user#
```

(iv) Try to call **less** on **/etc/shadow** and invoke

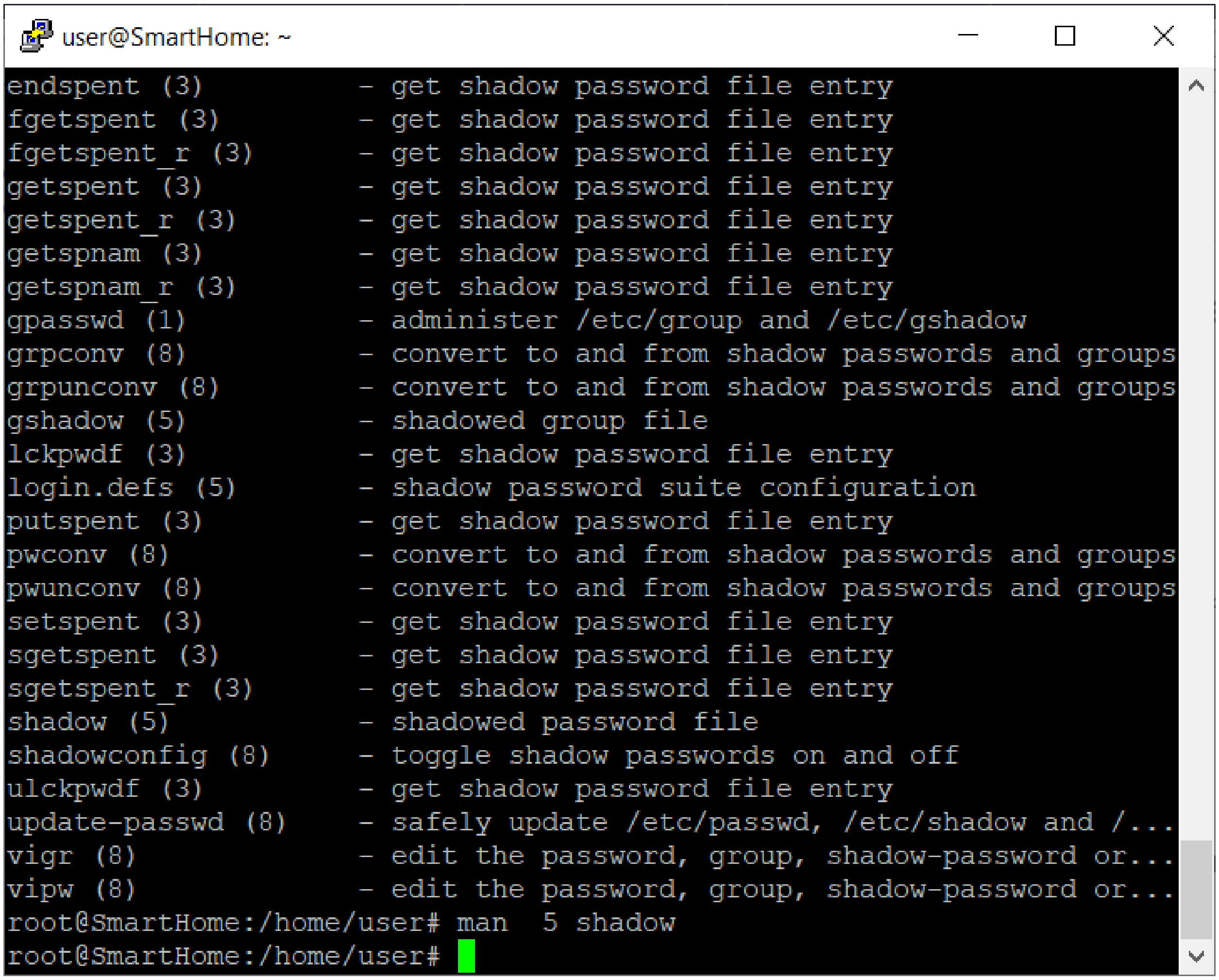
sudo less /etc/shadow

man -k shadow (display all command connected with shadow functionality)

man 5 shadow (5 means description and structure of the file shadow)

Analyze content of **/etc/shadow** based on what you've found in **man** 5 shadow.

(second field is real hash of user password)



```
endspent (3)         - get shadow password file entry
fgetspent (3)        - get shadow password file entry
fgetspent_r (3)      - get shadow password file entry
getspent (3)         - get shadow password file entry
getspent_r (3)       - get shadow password file entry
getspnam (3)         - get shadow password file entry
getspnam_r (3)       - get shadow password file entry
gpasswd (1)          - administer /etc/group and /etc/gshadow
grpconv (8)          - convert to and from shadow passwords and groups
grpunconv (8)        - convert to and from shadow passwords and groups
gshadow (5)          - shadowed group file
lckpwdf (3)          - get shadow password file entry
login.defs (5)       - shadow password suite configuration
putspent (3)         - get shadow password file entry
pwconv (8)           - convert to and from shadow passwords and groups
pwunconv (8)         - convert to and from shadow passwords and groups
setspent (3)         - get shadow password file entry
sgetspent (3)        - get shadow password file entry
sgetspent_r (3)      - get shadow password file entry
shadow (5)           - shadowed password file
shadowconfig (8)     - toggle shadow passwords on and off
ulckpwdf (3)         - get shadow password file entry
update-passwd (8)    - safely update /etc/passwd, /etc/shadow and /...
vigr (8)             - edit the password, group, shadow-password or...
vipw (8)             - edit the password, group, shadow-password or...
root@SmartHome:/home/user# man  5 shadow
root@SmartHome:/home/user#
```

3. Dealing with **chmod**.

(i) An executable script. Open your favorite editor and put these lines into a file

#!/bin/bash

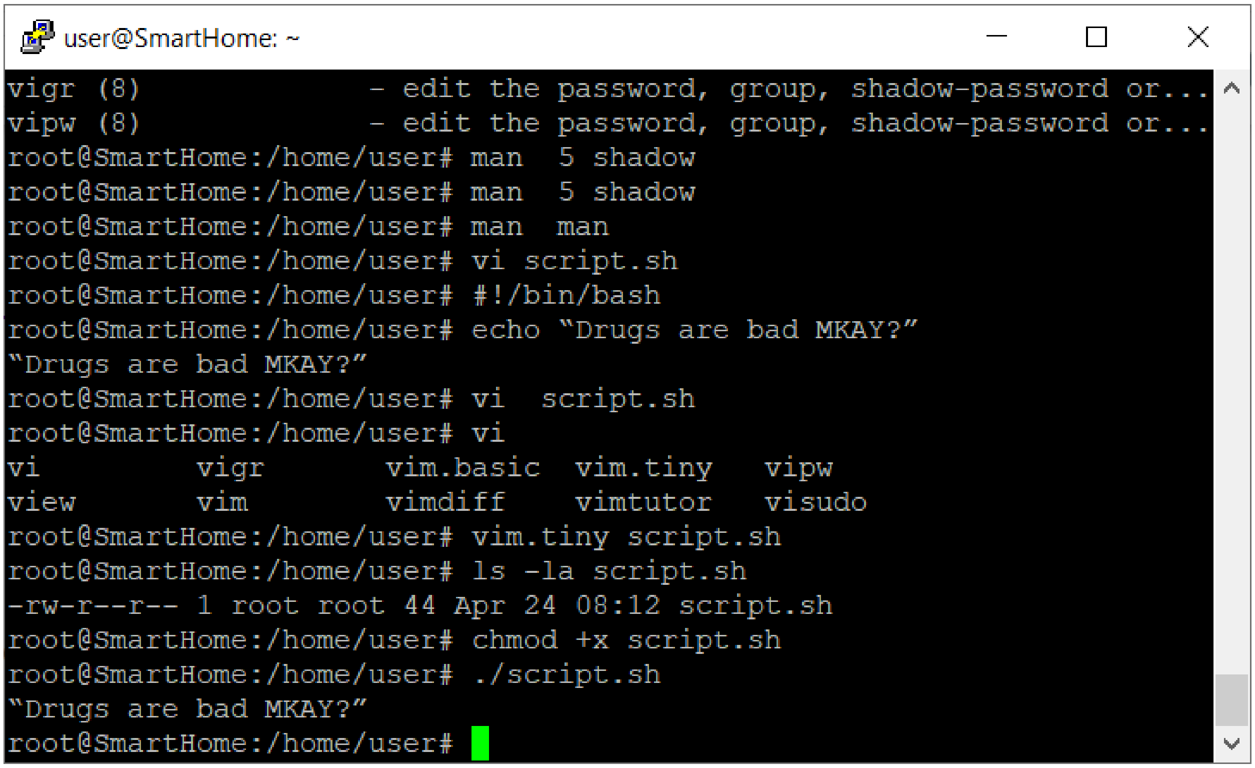
echo “Drugs are bad MKAY?”

Give name “script.sh” to the script and call to

chmod +x script.sh (add X “exec” perm (user group and other) to file)

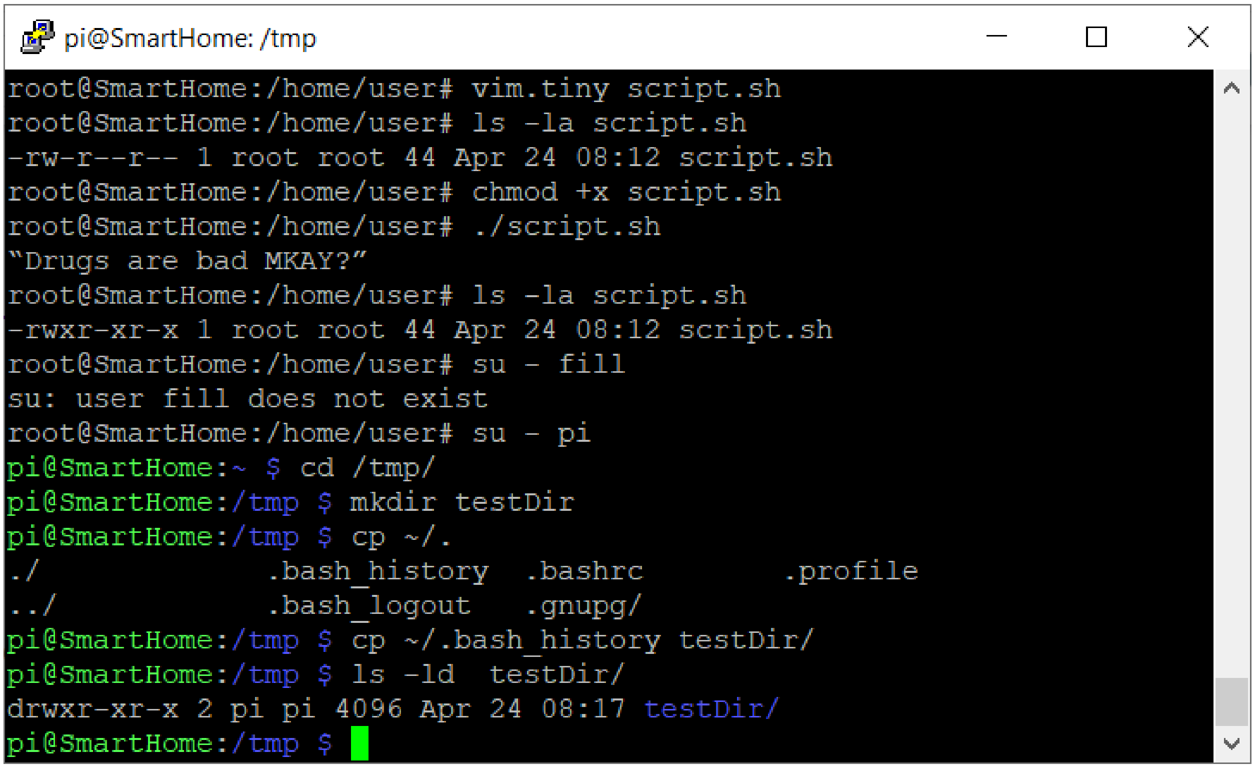
Then you are ready to execute the script:

./**script.sh**



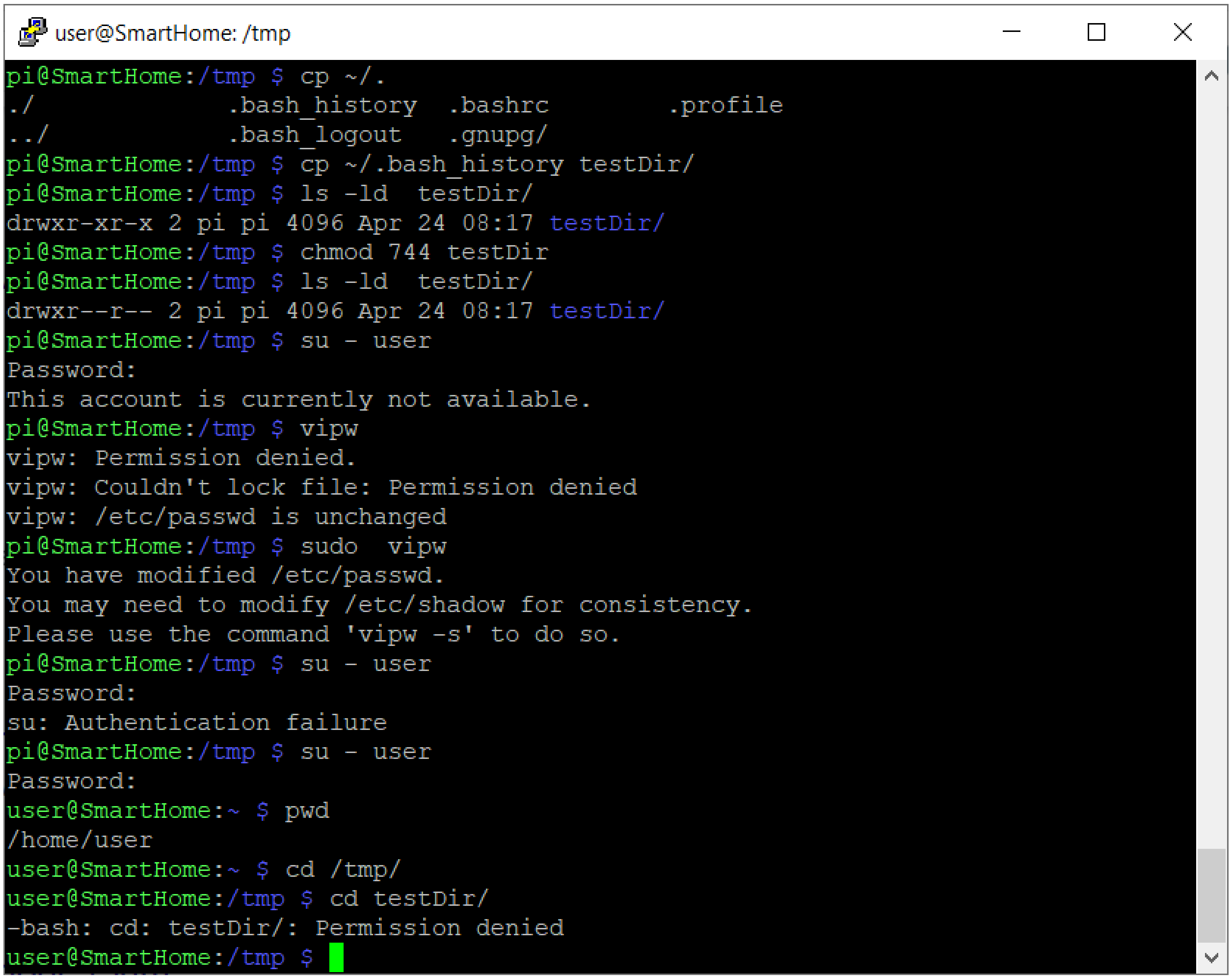
```
vigr (8)             - edit the password, group, shadow-password or...
vipw (8)             - edit the password, group, shadow-password or...
root@SmartHome:/home/user# man  5 shadow
root@SmartHome:/home/user# man  5 shadow
root@SmartHome:/home/user# man  man
root@SmartHome:/home/user# vi script.sh
root@SmartHome:/home/user# #!/bin/bash
root@SmartHome:/home/user# echo “Drugs are bad MKAY?”
“Drugs are bad MKAY?”
root@SmartHome:/home/user# vi  script.sh
root@SmartHome:/home/user# vi
vi          vigr        vim.basic   vim.tiny    vipw
view        vim         vimdiff     vimtutor    visudo
root@SmartHome:/home/user# vim.tiny script.sh
root@SmartHome:/home/user# ls -la script.sh
-rw-r--r-- 1 root root 44 Apr 24 08:12 script.sh
root@SmartHome:/home/user# chmod +x script.sh
root@SmartHome:/home/user# ./script.sh
“Drugs are bad MKAY?”
root@SmartHome:/home/user#
```

(ii) Suppose, you have logged in to the system as *guest*. (I used user pi) Create directory “testDir” in the **/tmp**; put some file into testDir and prohibit user *user* from visiting this directory (i.e. “testDir”).



```
root@SmartHome:/home/user# vim.tiny script.sh
root@SmartHome:/home/user# ls -la script.sh
-rw-r--r-- 1 root root 44 Apr 24 08:12 script.sh
root@SmartHome:/home/user# chmod +x script.sh
root@SmartHome:/home/user# ./script.sh
“Drugs are bad MKAY?”
root@SmartHome:/home/user# ls -la script.sh
-rwxr-xr-x 1 root root 44 Apr 24 08:12 script.sh
root@SmartHome:/home/user# su - fill
su: user fill does not exist
root@SmartHome:/home/user# su - pi
pi@SmartHome:~ $ cd /tmp/
pi@SmartHome:/tmp $ mkdir testDir
pi@SmartHome:/tmp $ cp ~/.
./             .bash_history  .bashrc        .profile
../            .bash_logout   .gnupg/
pi@SmartHome:/tmp $ cp ~/.bash_history testDir/
pi@SmartHome:/tmp $ ls -ld  testDir/
drwxr-xr-x 2 pi pi 4096 Apr 24 08:17 testDir/
pi@SmartHome:/tmp $
```

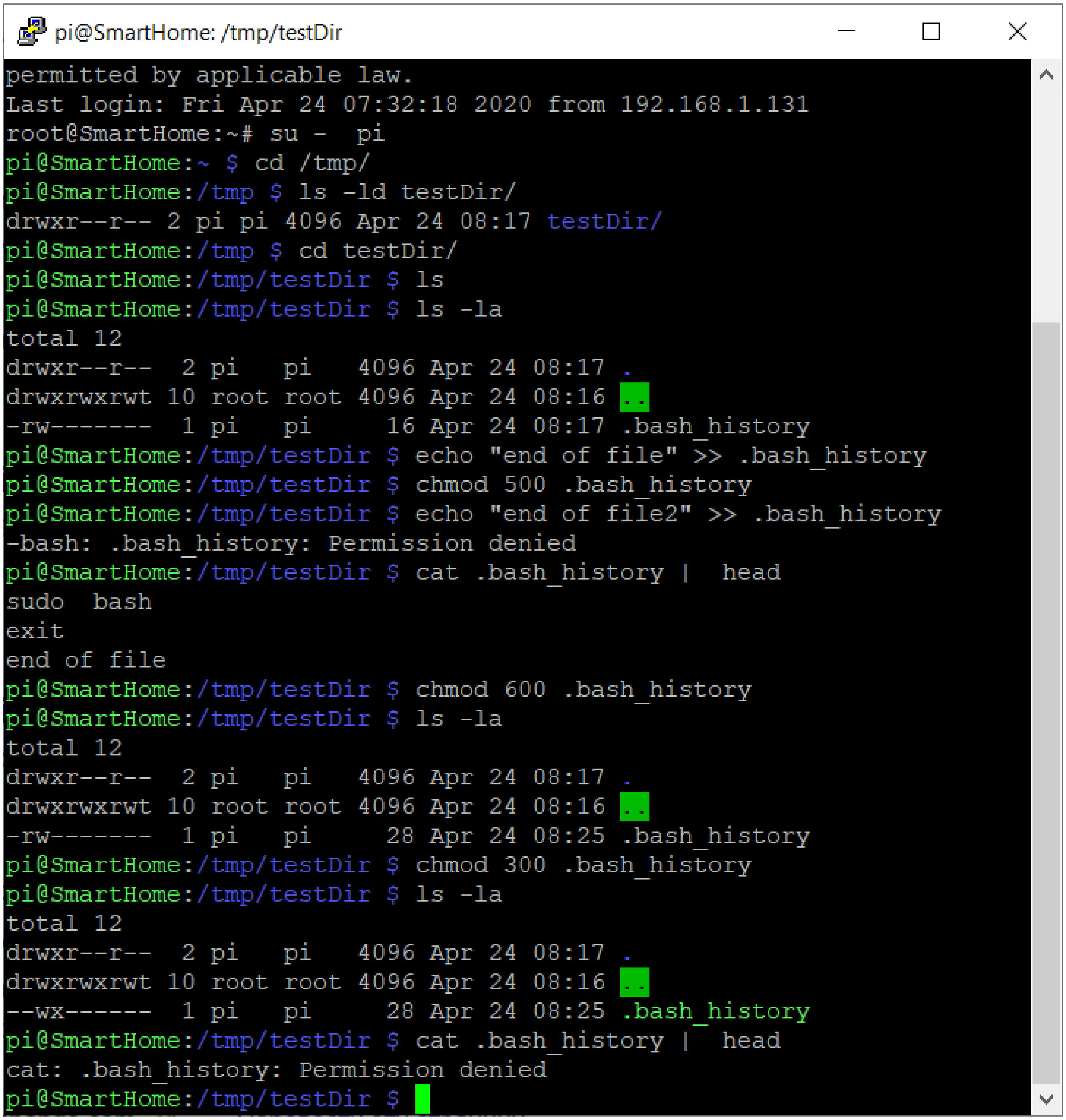
It is needed to set permission of directory=disable x bit for Other field (I used command chmod 744 testDir)



```
pi@SmartHome:/tmp $ cp ~/.
./              .bash_history  .bashrc        .profile
../             .bash_logout   .gnupg/
pi@SmartHome:/tmp $ cp ~/.bash_history testDir/
pi@SmartHome:/tmp $ ls -ld  testDir/
drwxr-xr-x 2 pi pi 4096 Apr 24 08:17 testDir/
pi@SmartHome:/tmp $ chmod 744 testDir
pi@SmartHome:/tmp $ ls -ld  testDir/
drwxr--r-- 2 pi pi 4096 Apr 24 08:17 testDir/
pi@SmartHome:/tmp $ su - user
Password:
This account is currently not available.
pi@SmartHome:/tmp $ vipw
vipw: Permission denied.
vipw: Couldn't lock file: Permission denied
vipw: /etc/passwd is unchanged
pi@SmartHome:/tmp $ sudo  vipw
You have modified /etc/passwd.
You may need to modify /etc/shadow for consistency.
Please use the command 'vipw -s' to do so.
pi@SmartHome:/tmp $ su - user
Password:
su: Authentication failure
pi@SmartHome:/tmp $ su - user
Password:
user@SmartHome:~ $ pwd
/home/user
user@SmartHome:~ $ cd /tmp/
user@SmartHome:/tmp $ cd testDir/
-bash: cd: testDir/: Permission denied
user@SmartHome:/tmp $
```

(iii) Test, if it possible to forbid an owner of some file to read to or write from this file.

Yes, it is possible read and write see screenshot with actions for details:



```
permitted by applicable law.
Last login: Fri Apr 24 07:32:18 2020 from 192.168.1.131
root@SmartHome:~# su -  pi
pi@SmartHome:~ $ cd /tmp/
pi@SmartHome:/tmp $ ls -ld testDir/
drwxr--r-- 2 pi pi 4096 Apr 24 08:17 testDir/
pi@SmartHome:/tmp $ cd testDir/
pi@SmartHome:/tmp/testDir $ ls
pi@SmartHome:/tmp/testDir $ ls -la
total 12
drwxr--r--  2 pi   pi   4096 Apr 24 08:17 .
drwxrwxrwt 10 root root 4096 Apr 24 08:16 ..
-rw-------  1 pi   pi     16 Apr 24 08:17 .bash_history
pi@SmartHome:/tmp/testDir $ echo "end of file" >> .bash_history
pi@SmartHome:/tmp/testDir $ chmod 500 .bash_history
pi@SmartHome:/tmp/testDir $ echo "end of file2" >> .bash_history
-bash: .bash_history: Permission denied
pi@SmartHome:/tmp/testDir $ cat .bash_history |  head
sudo  bash
exit
end of file
pi@SmartHome:/tmp/testDir $ chmod 600 .bash_history
pi@SmartHome:/tmp/testDir $ ls -la
total 12
drwxr--r--  2 pi   pi   4096 Apr 24 08:17 .
drwxrwxrwt 10 root root 4096 Apr 24 08:16 ..
-rw-------  1 pi   pi     28 Apr 24 08:25 .bash_history
pi@SmartHome:/tmp/testDir $ chmod 300 .bash_history
pi@SmartHome:/tmp/testDir $ ls -la
total 12
drwxr--r--  2 pi   pi   4096 Apr 24 08:17 .
drwxrwxrwt 10 root root 4096 Apr 24 08:16 ..
--wx------  1 pi   pi     28 Apr 24 08:25 .bash_history
pi@SmartHome:/tmp/testDir $ cat .bash_history |  head
cat: .bash_history: Permission denied
pi@SmartHome:/tmp/testDir $
```