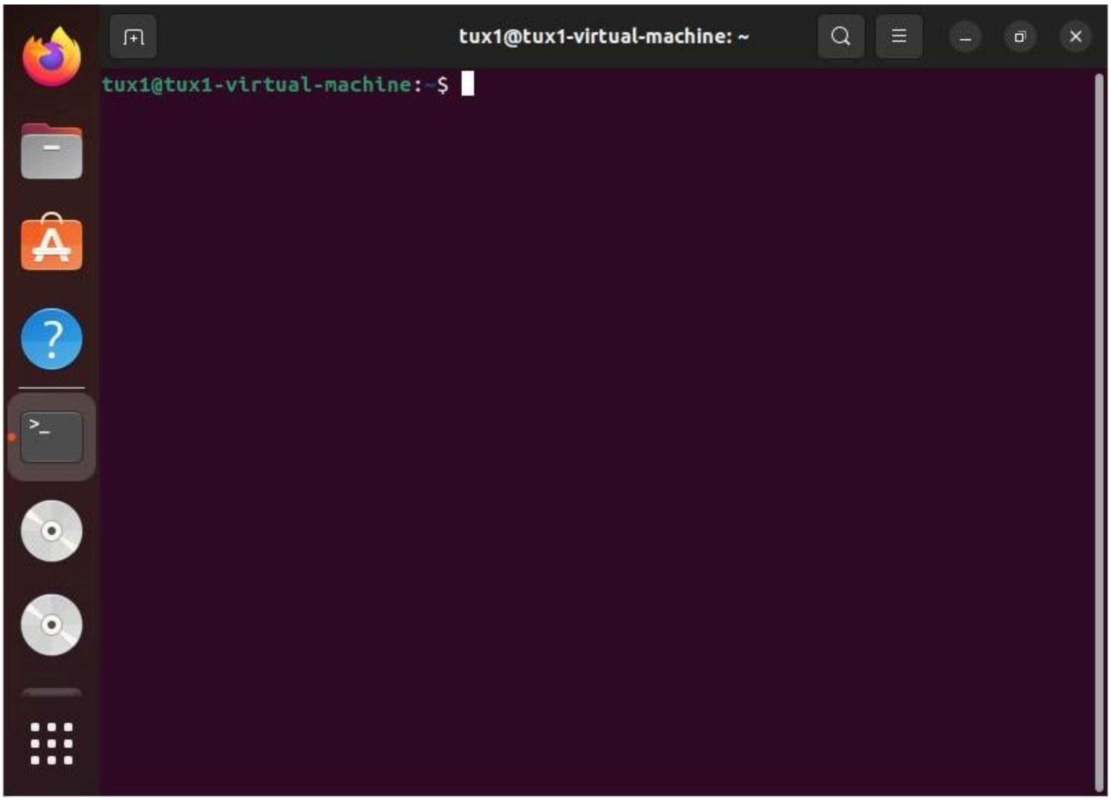
Exercise1

1.If you are not logged in as yourself at tty7, log in now. Make sure you've got a terminal window open.

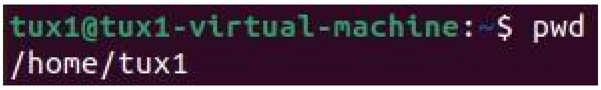
tty1-6 是文本型控制台，tty7 是图形显示管理器，在 Ubuntu 系统上以 tux1 身份登录即可进入 tty7，并打开终端窗口：



```
tux1@tux1-virtual-machine:~$ 
```

2.Check the directory you are placed in. What directory is this?

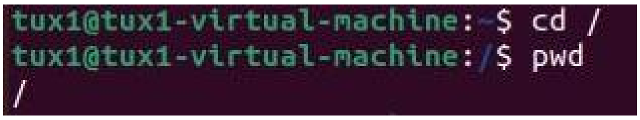
使用 pwd 命令：



```
tux1@tux1-virtual-machine:~$ pwd
/home/tux1
```

3.Change your current directory to the root directory (/).

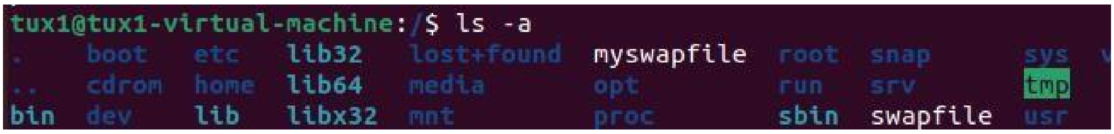
使用 cd 命令：



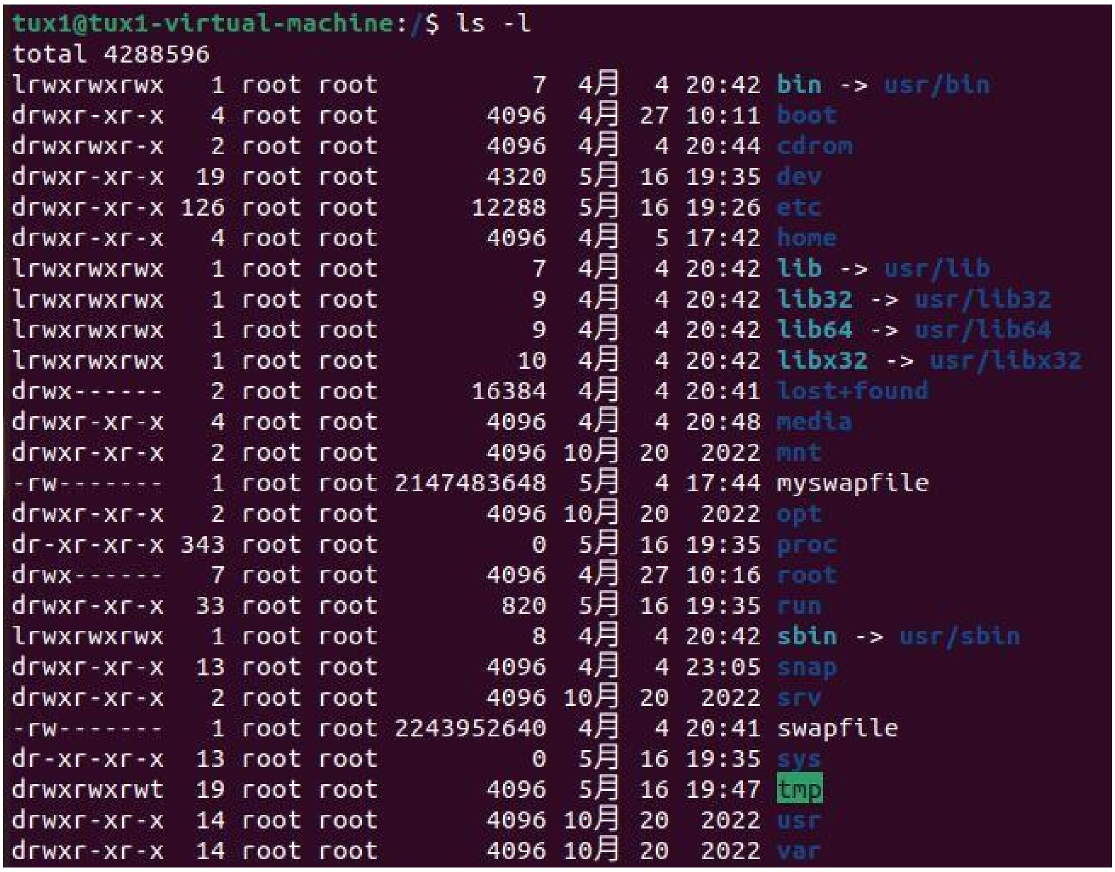
```
tux1@tux1-virtual-machine:~$ cd /
tux1@tux1-virtual-machine:/$ pwd
/
```

4.Verify that you are in the root directory and then execute both a simple and a long listing of the files in that directory.

使用 ls -a 命令：



```
tux1@tux1-virtual-machine:/$ ls -a
.    boot   etc   lib32   lost+found  myswapfile  root  snap      sys  v
..   cdrom  home  lib64   media       opt         run   srv       tmp
bin  dev    lib   libx32  mnt         proc        sbin  swapfile  usr
```



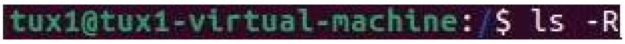
```
tux1@tux1-virtual-machine:/$ ls -l
total 4288596
lrwxrwxrwx   1 root root          7  4月   4 20:42 bin -> usr/bin
drwxr-xr-x   4 root root       4096  4月  27 10:11 boot
drwxrwxr-x   2 root root       4096  4月   4 20:44 cdrom
drwxr-xr-x  19 root root       4320  5月  16 19:35 dev
drwxr-xr-x 126 root root      12288  5月  16 19:26 etc
drwxr-xr-x   4 root root       4096  4月   5 17:42 home
lrwxrwxrwx   1 root root          7  4月   4 20:42 lib -> usr/lib
lrwxrwxrwx   1 root root          9  4月   4 20:42 lib32 -> usr/lib32
lrwxrwxrwx   1 root root          9  4月   4 20:42 lib64 -> usr/lib64
lrwxrwxrwx   1 root root         10  4月   4 20:42 libx32 -> usr/libx32
drwx------   2 root root      16384  4月   4 20:41 lost+found
drwxr-xr-x   4 root root       4096  4月   4 20:48 media
drwxr-xr-x   2 root root       4096 10月  20  2022 mnt
-rw-------   1 root root 2147483648  5月   4 17:44 myswapfile
drwxr-xr-x   2 root root       4096 10月  20  2022 opt
dr-xr-xr-x 343 root root          0  5月  16 19:35 proc
drwx------   7 root root       4096  4月  27 10:16 root
drwxr-xr-x  33 root root        820  5月  16 19:35 run
lrwxrwxrwx   1 root root          8  4月   4 20:42 sbin -> usr/sbin
drwxr-xr-x  13 root root       4096  4月   4 23:05 snap
drwxr-xr-x   2 root root       4096 10月  20  2022 srv
-rw-------   1 root root 2243952640  4月   4 20:41 swapfile
dr-xr-xr-x  13 root root          0  5月  16 19:35 sys
drwxrwxrwt  19 root root       4096  5月  16 19:47 tmp
drwxr-xr-x  14 root root       4096 10月  20  2022 usr
drwxr-xr-x  14 root root       4096 10月  20  2022 var
```

根目录中应包含的文件夹均可在 ls 命令所列出的文件/文件夹中找到，证明当前所在目录为根目录。

5.List all files in the current directory and list all files in the current directory and below.

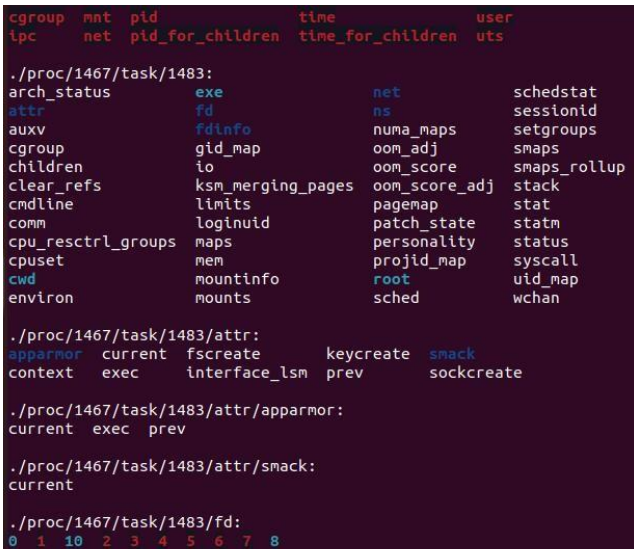
Note: This command provides extensive output. When you have seen enough, end the command with the correct sequence.

执行 ls -R 命令：



```
tux1@tux1-virtual-machine:/$ ls -R
```

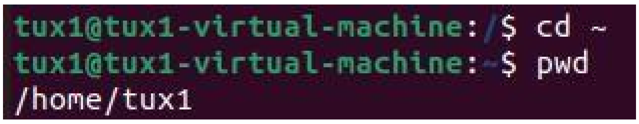
结果如下：



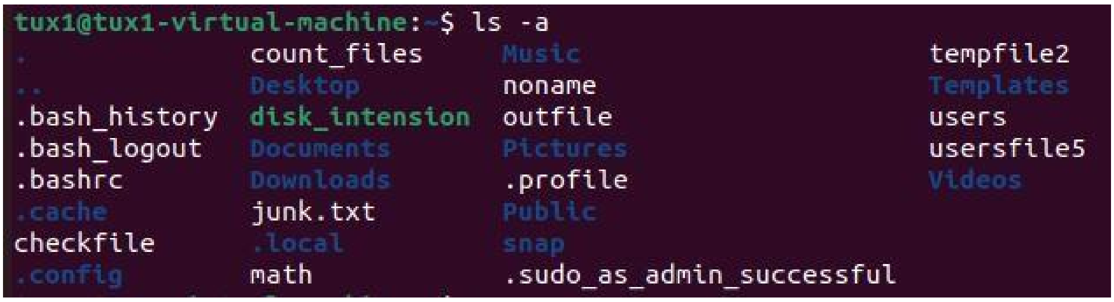
```
cgroup  mnt  pid               time               user
ipc     net  pid_for_children  time_for_children  uts

./proc/1467/task/1483:
arch_status         exe                 net             schedstat
attr                fd                  ns              sessionid
auxv                fdinfo              numa_maps       setgroups
cgroup              gid_map             oom_adj         smaps
children            io                  oom_score       smaps_rollup
clear_refs          ksm_merging_pages   oom_score_adj   stack
cmdline             limits              pagemap         stat
comm                loginuid            patch_state     statm
cpu_resctrl_groups  maps                personality     status
cpuset              mem                 projid_map      syscall
cwd                 mountinfo           root            uid_map
environ             mounts              sched           wchan

./proc/1467/task/1483/attr:
apparmor  current  fscreate       keycreate  smack
context   exec     interface_lsm  prev       sockcreate

./proc/1467/task/1483/attr/apparmor:
current  exec  prev

./proc/1467/task/1483/attr/smack:
current

./proc/1467/task/1483/fd:
0  1  10  2  3  4  5  6  7  8
```

6.Return to your home directory and list its contents including hidden files.

执行 cd~命令与 ls -a 命令：



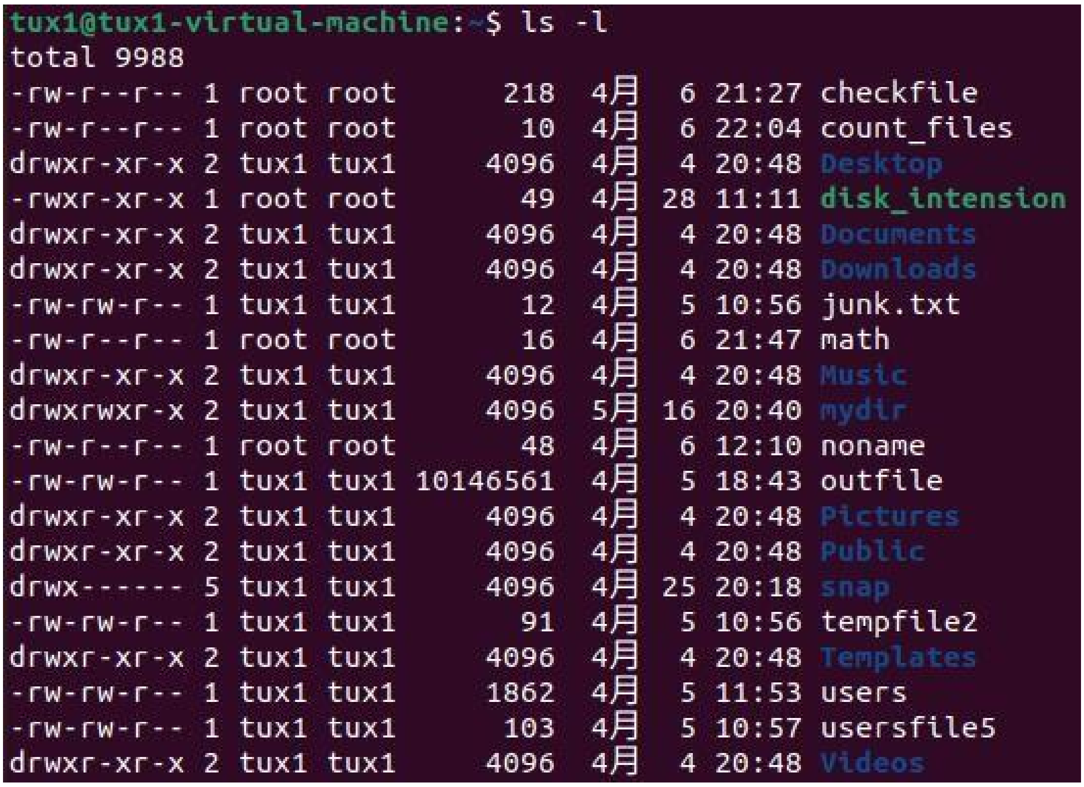
```
tux1@tux1-virtual-machine:/$ cd ~
tux1@tux1-virtual-machine:~$ pwd
/home/tux1
```



```
tux1@tux1-virtual-machine:~$ ls -a
.               count_files     Music                        tempfile2
..              Desktop         noname                       Templates
.bash_history   disk_intension  outfile                      users
.bash_logout    Documents       Pictures                     usersfile5
.bashrc         Downloads       .profile                     Videos
.cache          junk.txt        Public
checkfile       .local          snap
.config         math            .sudo_as_admin_successful
```

7.Create a directory in your home directory called mydir. Then, issue the command to view a long listing of your home directory and the ~/mydir directory. (Do not show the contents of the directories.) What is the size of each directory?

执行 mkdir 命令目录，执行 ls -l 命令分别查看主目录和~/mydir 目录的长列表：

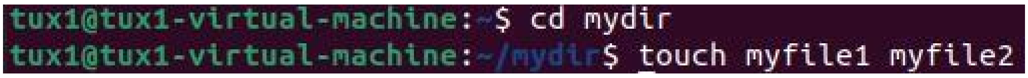


```
tux1@tux1-virtual-machine:~$ ls -l
total 9988
-rw-r--r-- 1 root root      218  4月  6 21:27 checkfile
-rw-r--r-- 1 root root       10  4月  6 22:04 count_files
drwxr-xr-x 2 tux1 tux1     4096  4月  4 20:48 Desktop
-rwxr-xr-x 1 root root       49  4月 28 11:11 disk_intension
drwxr-xr-x 2 tux1 tux1     4096  4月  4 20:48 Documents
drwxr-xr-x 2 tux1 tux1     4096  4月  4 20:48 Downloads
-rw-rw-r-- 1 tux1 tux1       12  4月  5 10:56 junk.txt
-rw-r--r-- 1 root root       16  4月  6 21:47 math
drwxr-xr-x 2 tux1 tux1     4096  4月  4 20:48 Music
drwxrwxr-x 2 tux1 tux1     4096  5月 16 20:40 mydir
-rw-r--r-- 1 root root       48  4月  6 12:10 noname
-rw-rw-r-- 1 tux1 tux1 10146561  4月  5 18:43 outfile
drwxr-xr-x 2 tux1 tux1     4096  4月  4 20:48 Pictures
drwxr-xr-x 2 tux1 tux1     4096  4月  4 20:48 Public
drwx------ 5 tux1 tux1     4096  4月 25 20:18 snap
-rw-rw-r-- 1 tux1 tux1       91  4月  5 10:56 tempfile2
drwxr-xr-x 2 tux1 tux1     4096  4月  4 20:48 Templates
-rw-rw-r-- 1 tux1 tux1     1862  4月  5 11:53 users
-rw-rw-r-- 1 tux1 tux1      103  4月  5 10:57 usersfile5
drwxr-xr-x 2 tux1 tux1     4096  4月  4 20:48 Videos
```

其中主目录大小为 9988KB，~/mydir 目录大小为 0。

8.Change to the mydir directory. Create two zero-length files called myfile1 and myfile2.

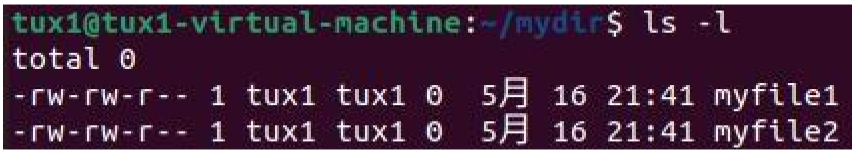
执行 touch 命令创建两个名为 myfile1 与 myfile2 的 0 长度文件：



```
tux1@tux1-virtual-machine:~$ cd mydir
tux1@tux1-virtual-machine:~/mydir$ touch myfile1 myfile2
```

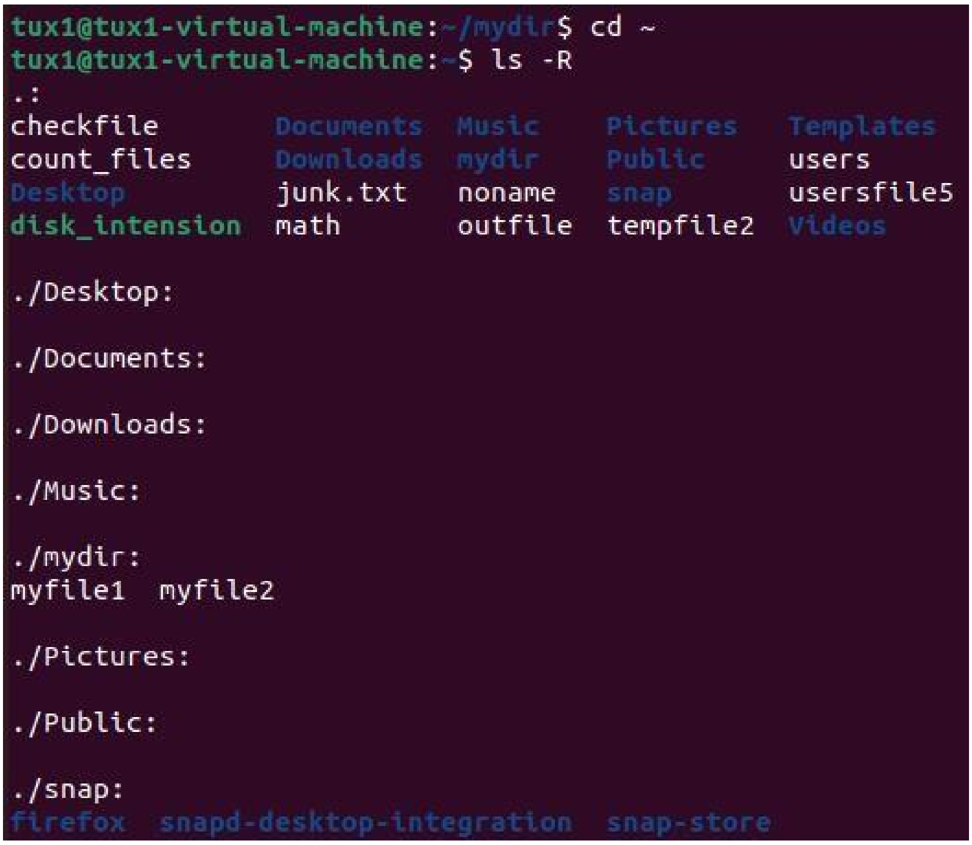
9.Issue the command to view a long listing of the contents of the mydir directory. What are the sizes of myfile1 and myfile2?_

执行 ls -l 命令，文件大小均为 0：



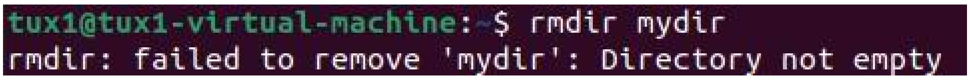
```
tux1@tux1-virtual-machine:~/mydir$ ls -l
total 0
-rw-rw-r-- 1 tux1 tux1 0  5月 16 21:41 myfile1
-rw-rw-r-- 1 tux1 tux1 0  5月 16 21:41 myfile2
```

10.Return to your home directory and use the ls -R command to view your directory tree.



```
tux1@tux1-virtual-machine:~/mydir$ cd ~
tux1@tux1-virtual-machine:~$ ls -R
.:
checkfile       Documents  Music    Pictures   Templates
count_files     Downloads  mydir    Public     users
Desktop         junk.txt   noname   snap       usersfile5
disk_intension  math       outfile  tempfile2  Videos

./Desktop:

./Documents:

./Downloads:

./Music:

./mydir:
myfile1  myfile2

./Pictures:

./Public:

./snap:
firefox  snapd-desktop-integration  snap-store
```

11.Try to remove the mydir directory. Does it work?

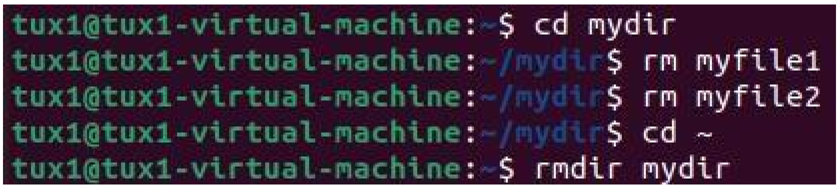


```
tux1@tux1-virtual-machine:~$ rmdir mydir
rmdir: failed to remove 'mydir': Directory not empty
```

终端报告删除 mydir 目录失败，原因是 mydir 目录非空，其中有其他文件或目录。

12.Go to the mydir directory once more and delete the two files in that directory; then go back up to your home directory and delete the mydir directory.

此时 mydir 目录成功删除：

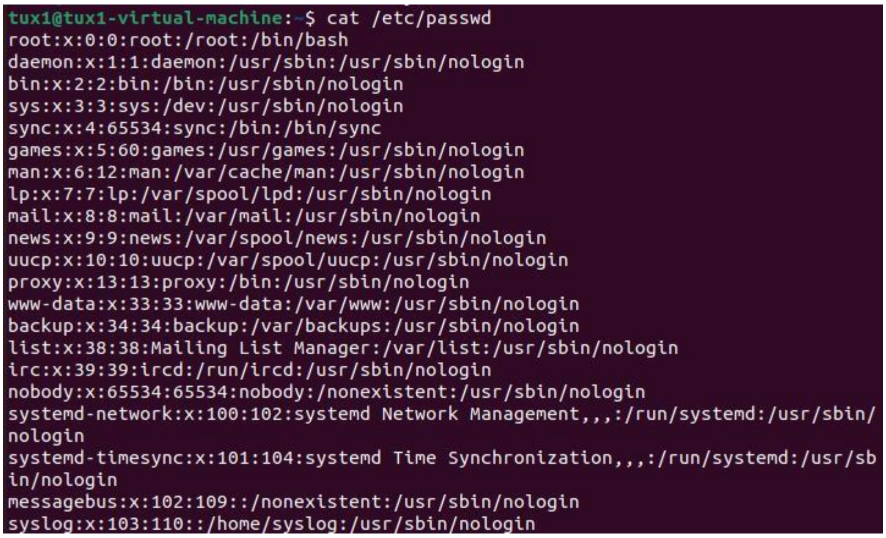


```
tux1@tux1-virtual-machine:~$ cd mydir
tux1@tux1-virtual-machine:~/mydir$ rm myfile1
tux1@tux1-virtual-machine:~/mydir$ rm myfile2
tux1@tux1-virtual-machine:~/mydir$ cd ~
tux1@tux1-virtual-machine:~$ rmdir mydir
```

Working with files

13.Look at the contents of the /etc/passwd file. The /etc/passwd file contains a list of all the users authorized to use the system.

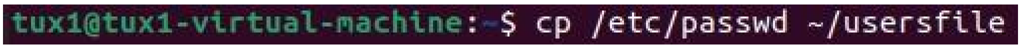
使用 cat 命令：



```
tux1@tux1-virtual-machine:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/
nologin
systemd-timesync:x:101:104:systemd Time Synchronization,,,:/run/systemd:/usr/sb
in/nologin
messagebus:x:102:109::/nonexistent:/usr/sbin/nologin
syslog:x:103:110::/home/syslog:/usr/sbin/nologin
```

14.Copy the /etc/passwd file to your home directory, and rename it to usersfile.

使用 cp 命令：

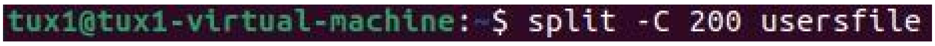


```
tux1@tux1-virtual-machine:~$ cp /etc/passwd ~/usersfile
```

15.Split the usersfile into a number of smaller files of 200 bytes each.

执行 split 命令将 userfile 拆分成多个小文件，该命令格式如下：

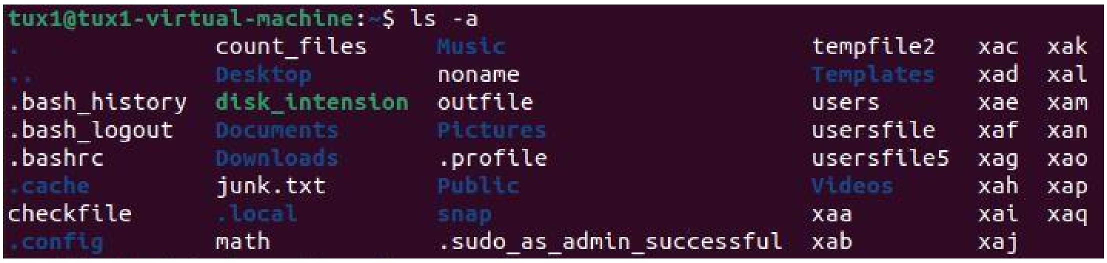
split[操作大文件的参数][大文件][输出文件的可选参数][输出文件名]



```
tux1@tux1-virtual-machine:~$ split -C 200 usersfile
```

16.Make a long listing of all files in your home directory.

执行 ls -a 命令：

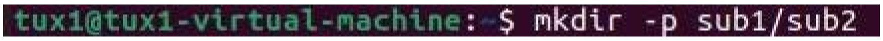


```
tux1@tux1-virtual-machine:~$ ls -a
.               count_files     Music                    tempfile2   xac  xak
..              Desktop         noname                   Templates   xad  xal
.bash_history   disk_intension  outfile                  users       xae  xam
.bash_logout    Documents       Pictures                 usersfile   xaf  xan
.bashrc         Downloads       .profile                 usersfile5  xag  xao
.cache          junk.txt        Public                   Videos      xah  xap
checkfile       .local          snap                     xaa         xai  xaq
.config         math            .sudo_as_admin_successful xab        xaj
```

Working with files and directories recursively

17.Create a directory sub1 and create a directory sub2 in sub1. Do this all with one command.

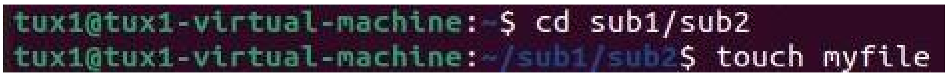
执行 mkdir 命令创建目录 sub1，并在 sub1 中创建目录 sub2：



```
tux1@tux1-virtual-machine:~$ mkdir -p sub1/sub2
```

18.Go to the sub2 directory and create a file called myfile.

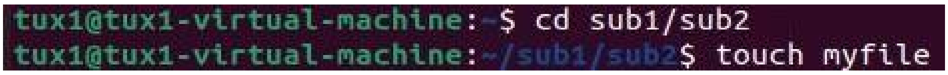
执行 cd 命令进入 sub2 目录，随后使用 touch 命令创建一个名为 myfile 的文件：



```
tux1@tux1-virtual-machine:~$ cd sub1/sub2
tux1@tux1-virtual-machine:~/sub1/sub2$ touch myfile
```

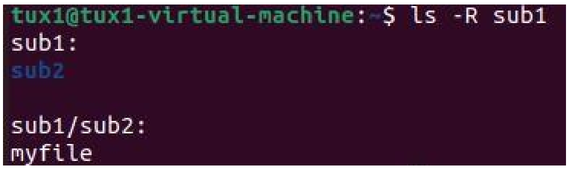
19.Go back to your home directory. Then make a copy of the whole sub1 directory tree by the name of tree1. Make a recursive listing of all files and directories in sub1 and tree1.

执行 cd~命令回到主目录，然后使用 cp 命令将 sub1 目录树复制到 tree1 中，选项-r 表示递归处理，将指定目录下的文件与子目录一并处理：

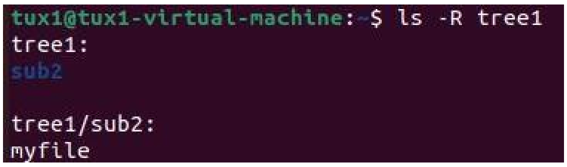


```
tux1@tux1-virtual-machine:~$ cd sub1/sub2
tux1@tux1-virtual-machine:~/sub1/sub2$ touch myfile
```

随后执行 ls -R 命令分别生成 sub1 与 tree1 中所有文件和目录的递归列表：



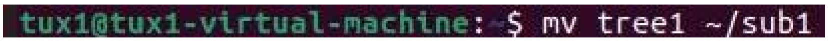
```
tux1@tux1-virtual-machine:~$ ls -R sub1
sub1:
sub2

sub1/sub2:
myfile
```



```
tux1@tux1-virtual-machine:~$ ls -R tree1
tree1:
sub2

tree1/sub2:
myfile
```

20.You now have two directory trees, named sub1 and tree1. Move the directory tree tree1 into the sub1 subdirectory.

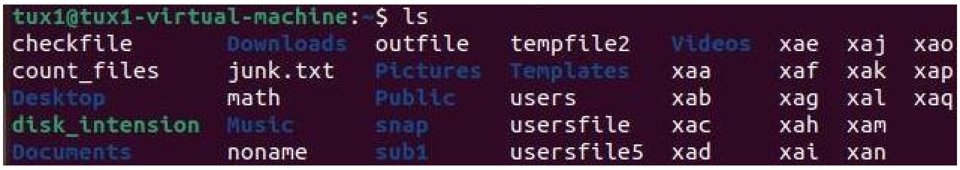
执行 mv 命令将目录树 tree1 移动到 sub1 子目录中：



```
tux1@tux1-virtual-machine:~$ mv tree1 ~/sub1
```

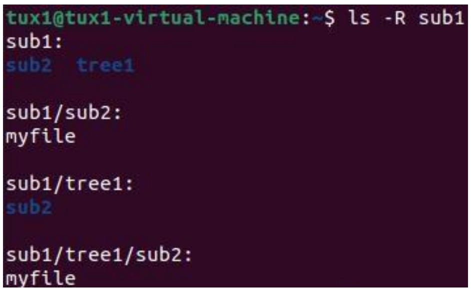
21.List the contents of your home directory. Make a recursive listing of all files and directories in the sub1 directory.

首先执行 ls 命令列出主目录内容，输出结果如下：



```
tux1@tux1-virtual-machine:~$ ls
checkfile       Downloads  outfile   tempfile2   Videos  xae  xaj  xao
count_files     junk.txt   Pictures  Templates   xaa     xaf  xak  xap
Desktop         math       Public    users       xab     xag  xal  xaq
disk_intension  Music      snap      usersfile   xac     xah  xam
Documents       noname     sub1      usersfile5  xad     xai  xan
```

其中只新加了一个目录 sub1，执行 ls -R 命令生成 sub1 目录中所有文件与目录的递归列表，输出结果如下：



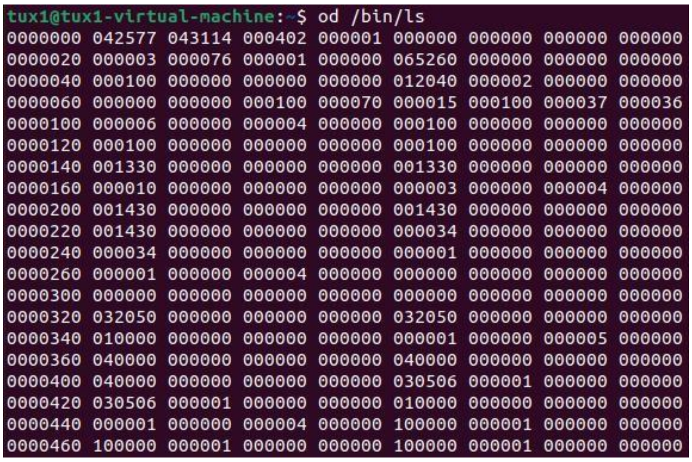
```
tux1@tux1-virtual-machine:~$ ls -R sub1
sub1:
sub2  tree1

sub1/sub2:
myfile

sub1/tree1:
sub2

sub1/tree1/sub2:
myfile
```

sub1 中有两个目录 sub1 与 tree1，且 sub2 与 tree1 的递归列表输出正确，说明成功将目录树 tree1 移动到 sub1 子目录中。

Working with binary files

22.List the content of the file /bin/ls using od or hexdump.

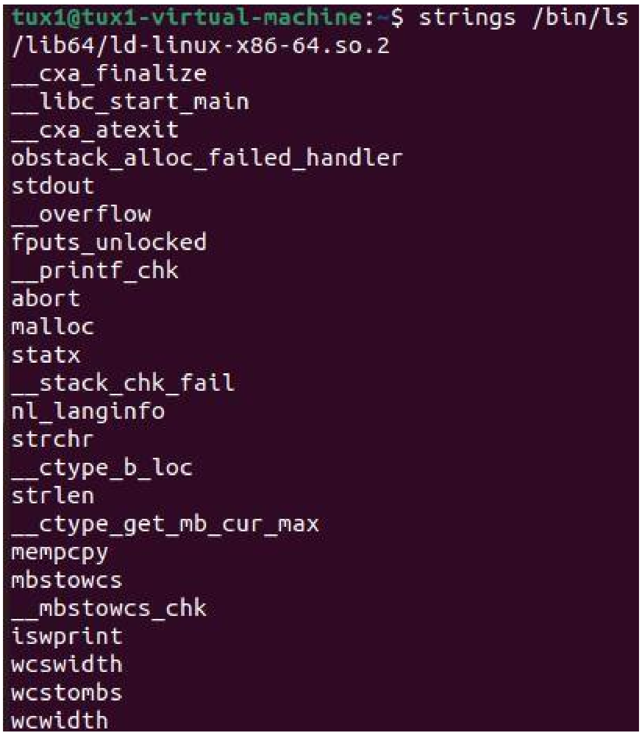
执行 od 命令列出/bin/ls 文件的内容，od 命令用于将指定文件内容以八进制、十进制、十六进制、浮点格式或 ASCII 编码字符方式显示，通常用于显示或查看文件中不能直接显示在终端的字符：



```
tux1@tux1-virtual-machine:~$ od /bin/ls
0000000 042577 043114 000402 000001 000000 000000 000000 000000
0000020 000003 000076 000001 000000 065260 000000 000000 000000
0000040 000100 000000 000000 000000 012040 000002 000000 000000
0000060 000000 000000 000100 000070 000015 000100 000037 000036
0000100 000006 000000 000004 000000 000100 000000 000000 000000
0000120 000100 000000 000000 000000 000100 000000 000000 000000
0000140 001330 000000 000000 000000 001330 000000 000000 000000
0000160 000010 000000 000000 000000 000003 000000 000004 000000
0000200 001430 000000 000000 000000 001430 000000 000000 000000
0000220 001430 000000 000000 000000 000034 000000 000000 000000
0000240 000034 000000 000000 000000 000001 000000 000000 000000
0000260 000001 000000 000004 000000 000000 000000 000000 000000
0000300 000000 000000 000000 000000 000000 000000 000000 000000
0000320 032050 000000 000000 000000 032050 000000 000000 000000
0000340 010000 000000 000000 000000 000001 000000 000005 000000
0000360 040000 000000 000000 000000 040000 000000 000000 000000
0000400 040000 000000 000000 000000 030506 000001 000000 000000
0000420 030506 000001 000000 000000 010000 000000 000000 000000
0000440 000001 000000 000004 000000 100000 000001 000000 000000
0000460 100000 000001 000000 000000 100000 000001 000000 000000
```

23.List all strings in the /bin/ls program.

strings 命令是二进制工具集 GNU Binutils 的一员，用于打印文件中可打印字符串，strings 命令在对象文件或二进制文件中查找可打印的字符串。执行 strings 命令列出/bin/ls 程序中的所有字符串，输出结果如下图所示（此处仅展示部分结果）：

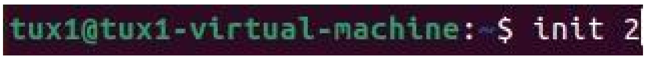


```
tux1@tux1-virtual-machine:~$ strings /bin/ls
/lib64/ld-linux-x86-64.so.2
__cxa_finalize
__libc_start_main
__cxa_atexit
obstack_alloc_failed_handler
stdout
__overflow
fputs_unlocked
__printf_chk
abort
malloc
statx
__stack_chk_fail
nl_langinfo
strchr
__ctype_b_loc
strlen
__ctype_get_mb_cur_max
mempcpy
mbstowcs
__mbstowcs_chk
iswprint
wcswidth
wcstombs
wcwidth
```

Exercise2

1.On tty3, log in as root.

执行 init2 命令进入多用户模式:



```
tux1@tux1-virtual-machine:~$ init 2
```

2.Execute the following series of commands:

groupadd penguins

useradd -m -g penguins -c “Tux the Penguin (1)” tux1

useradd -m -g penguins -c “Tux the Penguin (2)” tux2

passwd tux1

New password: penguin1

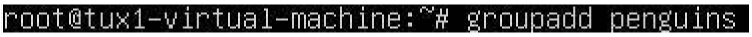
Retype new password: penguin1

passwd tux2

New password: penguin2

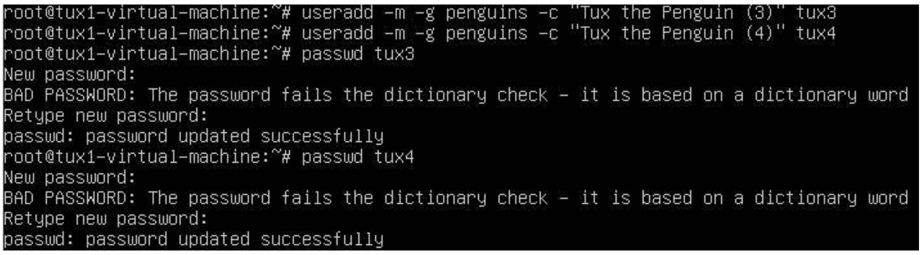
Retype new password: penguin2

执行过程如下：



```
root@tux1-virtual-machine:~# groupadd penguins
```

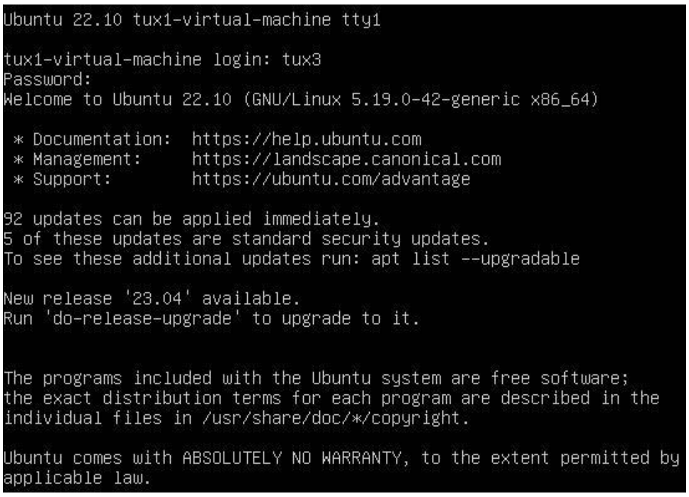
由于用户 tux1 与 tux2 已存在，后续操作改为在用户 tux3 与 tux4 上进行：



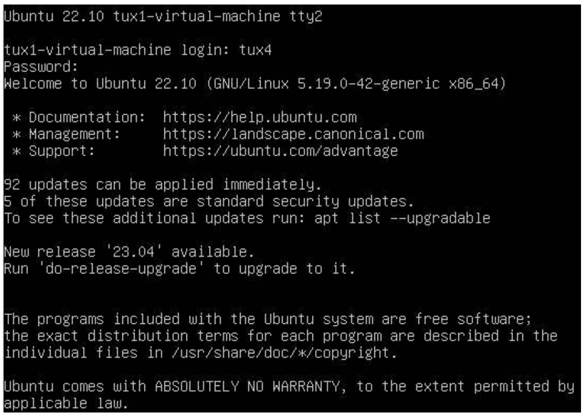
```
root@tux1-virtual-machine:~# useradd -m -g penguins -c "Tux the Penguin (3)" tux3
root@tux1-virtual-machine:~# useradd -m -g penguins -c "Tux the Penguin (4)" tux4
root@tux1-virtual-machine:~# passwd tux3
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: password updated successfully
root@tux1-virtual-machine:~# passwd tux4
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: password updated successfully
```

3.On tty1, log in as tux1 with password penguin1, and on tty2, log in as tux2 with password penguin2.

按下 Ctrl+Alt+F1 进入 tty1，以 tux3 身份登录：



```
Ubuntu 22.10 tux1-virtual-machine tty1

tux1-virtual-machine login: tux3
Password:
Welcome to Ubuntu 22.10 (GNU/Linux 5.19.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

92 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '23.04' available.
Run 'do-release-upgrade' to upgrade to it.


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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

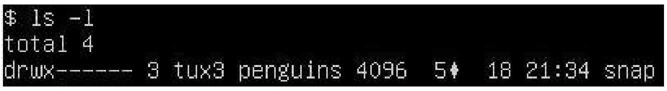


```
Ubuntu 22.10 tux1-virtual-machine tty2

tux1-virtual-machine login: tux4
Password:
Welcome to Ubuntu 22.10 (GNU/Linux 5.19.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

92 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '23.04' available.
Run 'do-release-upgrade' to upgrade to it.


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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

File and directory permissions

4.Switch to VT 1, where you are logged in as tux1, and look at the permissions on your home directory.

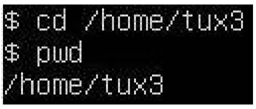
使用 ls -l 命令可以查看在主目录上的权限，有读、写与执行的权限：



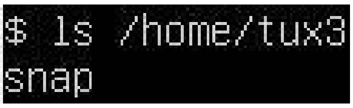
```
$ ls -l
total 4
drwx------ 3 tux3 penguins 4096  5♦  18 21:34 snap
```

5.Switch to VT2, where you are logged in as tux2. Try to change to the home directory of tux1, or read the contents of the home directory of tux1. Does this work?

进入 tty2，执行 cd 命令尝试进入目录/home/tux3，并执行 ls 命令，结果如下：



```
$ cd /home/tux3
$ pwd
/home/tux3
```



```
$ ls /home/tux3
snap
```

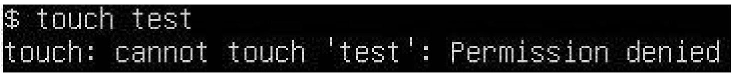
On a Fedora or Red Hat system, both commands fail, because the default permissions on a user's home directory are set to rwx------. On a SuSE system, both commands succeed, because the default permissions are set to rwxr-xr-x.

6.Fedora/Red Hat only: Switch to tty1. Change the permissions on the home directory of tux1 so that other users are allowed to read and access it. Then try to access the directory again as tux2. Does this work now?

由于本次实验使用 Ubuntu 操作系统，本题无法完成。

7.As tux2, try to create and delete files in tux1s home directory. Does this work?

使用 touch 命令在 tux3 的主目录中创建一个名为 test 的文件，执行结果如下：

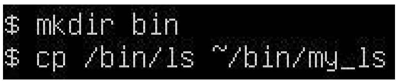


```
$ touch test
touch: cannot touch 'test': Permission denied
```

操作请求被拒绝，命令执行失败，tux4 不具备此权限。

8.Switch once again to tty1. Create a bin directory (Fedora/Red Hat only) and copy the file /bin/ls in there, renaming it to my_ls in the process.

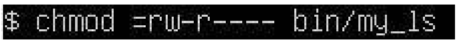
切换到 tty1，执行 mkdir 命令创建一个 bin 目录，随后执行 cp 命令将文件 /bin/ls 复制到其中，并将其重命名为 my_ls：



```
$ mkdir bin
$ cp /bin/ls ~/bin/my_ls
```

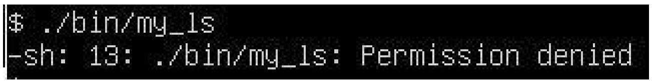
9.Set the permissions on my_ls to rw-r-----, and then try to execute it both as tux1 and tux2. Does this work? Why not?

执行 chmod 命令将 my_ls 命令的权限设置为 rw-r---：



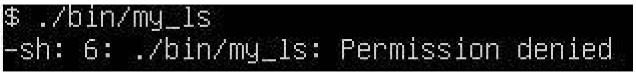
```
$ chmod =rw-r---- bin/my_ls
```

在 tty1 中以 tux3 身份执行 my_ls 文件，执行结果如下：



```
$ ./bin/my_ls
-sh: 13: ./bin/my_ls: Permission denied
```

转到 tty2 以 tux4 身份执行 my_ls 文件，执行结果如下：

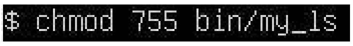


```
$ ./bin/my_ls
-sh: 6: ./bin/my_ls: Permission denied
```

my_ls 执行请求被拒绝，均无法顺利执行，其原因是 tux3 与 tux4 均不具有文件 my_ls 的执行权限。

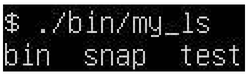
10.Now set the permissions to rwxr-xr-x, then try to execute it once more, both as tux1 and tux2. Does this work now?

执行 chmod 命令将 my_ls 命令的权限设置为 rwxr-xr-x：



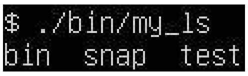
```
$ chmod 755 bin/my_ls
```

在 tty1 中以 tux3 身份执行 my_ls 文件，执行结果如下：



```
$ ./bin/my_ls
bin  snap  test
```

转到 tty2 以 tux4 身份执行 my_ls 文件，执行结果如下：



```
$ ./bin/my_ls
bin  snap  test
```

my_ls 程序能够顺利执行，此时 tux3 与 tux4 均具有文件 my_ls 的执行权限。

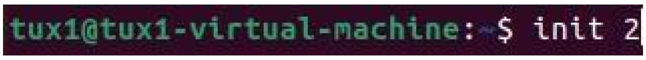
11.Try to execute my_ls as tux1, as tux2, and as yourself, but now with permissions rw-------, rw-rw----, rwx------, rwx--x---, and rwx--x--x as well. What permissions are required, at a minimum, for tux1 to execute my_ls? What permissions are required for tux2? What permissions does your own user account require?

通过实验可以发现：tux3 要想运行文件 my_ls，至少需要权限--x，即文件所有者的可执行权限；tux4 要想运行文件 my_ls，至少需要权限-----x--，即与文件所有者同属一个用户组的其他用户的可执行权限；tux1 要想运行文件 my_ls，至少需要权限-------x，即其他用户组的可执行权限。

Exercise3

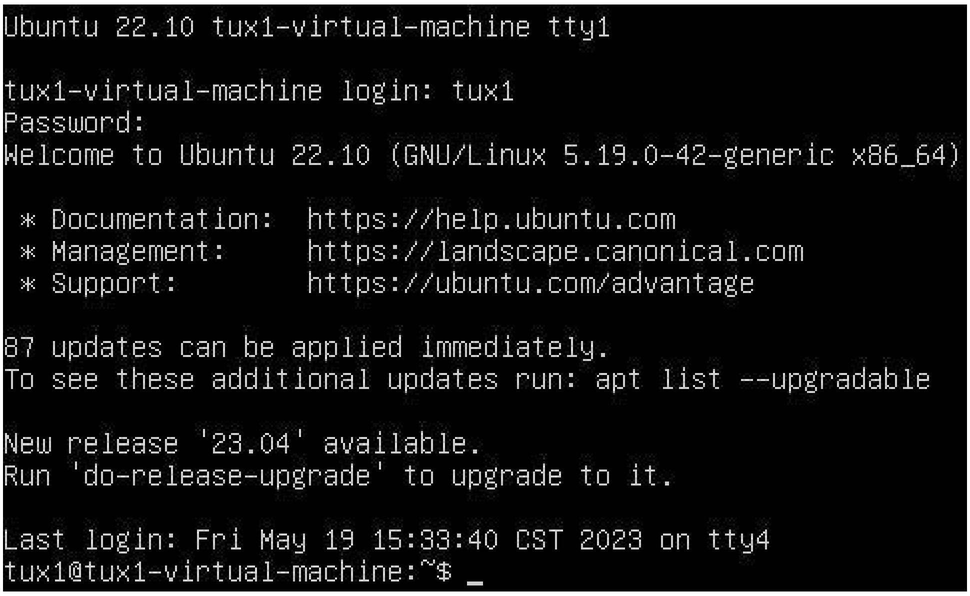
1.If you aren't already logged in as tux1 at tty1, log in now.

执行 init2 命令进入多用户模式:



```
tux1@tux1-virtual-machine:~$ init 2
```

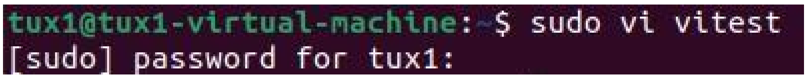
随后以 tux1 身份登录 tty1：



```
Ubuntu 22.10 tux1-virtual-machine tty1

tux1-virtual-machine login: tux1
Password:
Welcome to Ubuntu 22.10 (GNU/Linux 5.19.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

87 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '23.04' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri May 19 15:33:40 CST 2023 on tty4
tux1@tux1-virtual-machine:~$ _
```

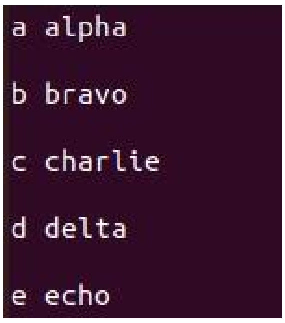
2.Ensure that you are in your home directory. Create a file in your home directory named vitest using Vi.

执行 pwd 命令确定当前所在目录为主目录，使用 vi 编辑器创建一个名为 vitest 的文件：



```
tux1@tux1-virtual-machine:~$ sudo vi vitest
[sudo] password for tux1:
```

Vi 编辑器中输入内容如下（此处仅截取部分）：



```
a alpha

b bravo

c charlie

d delta

e echo
```

Type the following text and the marine alphabet into the vitest file. Adding the alphabet is an easy way to fill a couple of screens of information needed for later use. This is a training session about the usage of the Vi editor. We need some more lines to learn the most common commands of the editor.

a alpha

b bravo

c charlie

...

(the rest of the marine alphabet)

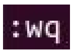
x x-ray

y yankee

z zulu

3.Return to command mode. Write and quit the file. Notice that as soon as you press the colon (:), it appears below the last line of your input area. When the buffer is empty and the file is closed, you see a message giving the number of lines and characters in the file.

按下 esc 键返回命令模式，输入“：wq”，可将文件保存并退出：



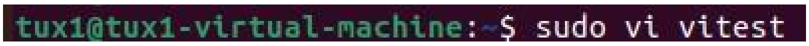
```
:wq
```

在命令模式下，只要按下冒号（：），就会出现在输入区最后一行下面。当缓冲区为空并关闭文件时，将看到一条消息，显示文件中的行数与字符数。

Cursor movement keys

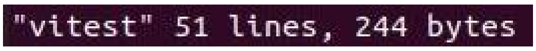
4.Open vitest file again. Notice that the bottom line of the screen indicates the name of the file and number of characters.

再次打开 vitest 文件：



```
tux1@tux1-virtual-machine:~$ sudo vi vitest
```

此时屏幕的最后一行表示文件名与字符数：

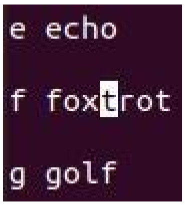


```
"vitest" 51 lines, 244 bytes
```

Vitest 文件共有 51 行，244 比特。

5.Using the H, J, K, and L keys, practice moving the through the file.

分别以光标所在位置为初始点，使用 h，j，k，l 键练习在文件中移动，光标初始位置如下：



```
e echo

f foxtrot

g golf
```

练习后发现：h 键表示光标向左移动一个字符；j 键表示光标向下移动一行；k 键表示光标向上移动一行；l 键表示光标向右移动一个字符。

6. Use the appropriate Vi commands to move through the text.

a.Move forward one page.

使用“Crtl+f”可以实现向前翻页。

b.Move back one page.

使用“Crtl+b”可以实现向后翻页。

c.Move the cursor to the first line on the screen.

使用“H”可以将光标移动到屏幕第一行。

d.Move the cursor to the last line in the file.

使用“G”可以将光标移动到屏幕最后一行。

e.Move the cursor to the first line in the file.

使用“gg”可以将光标移动到文件第一行。

f.Move the cursor to line 5 of the file.

使用“5gg”可以将光标移动到文件第五行。

g.Move the cursor to the end of the line.

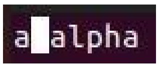
使用“$”可以将光标移动到行尾。

h.Move the cursor to the beginning of the line.

使用“0”可以将光标移动到行首。

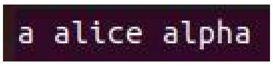
7.Change the file vitest so that after each letter of the alphabet a common first name is added that starts with that letter. Make sure you use different methods for switching from command mode to insert mode.

按下 i 键可以从命令模式切换到插入模式，i 表示在光标当前位置进行插入，可以在下图所示的光标位置处按 i 键进行编辑：



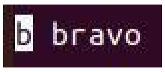
```
a alpha
```

编辑后的内容如下：



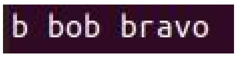
```
a alice alpha
```

按下 a 键可以从命令模式切换到插入模式，a 表示在光标后进行插入，可以在下图所示的光标位置处按 a 键进行编辑：



```
b bravo
```

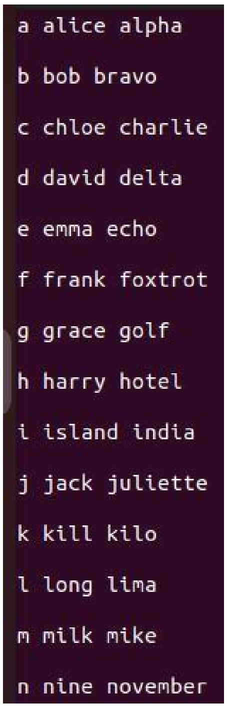
编辑后的内容如下：



```
b bob bravo
```

8.Practice some more with all the commands that are listed on your cheat sheet.

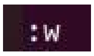
基于上述光标移动命令与进入编辑模式等命令，对字母表中其他内容进行填充以练习（此处仅展示部分结果）：



```
a alice alpha

b bob bravo

c chloe charlie

d david delta

e emma echo

f frank foxtrot

g grace golf

h harry hotel

i island india

j jack juliette

k kill kilo

l long lima

m milk mike

n nine november
```

9.Save the file but do not exit Vi.

按下 esc 键进入命令模式，按下“：”进入末行模式，输入命令 w 可以保存文件，但不退出 vi：

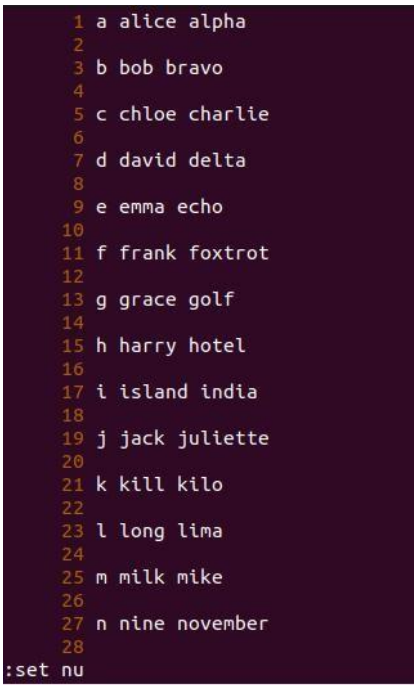


```
:w
```

Using set to customize the editing session

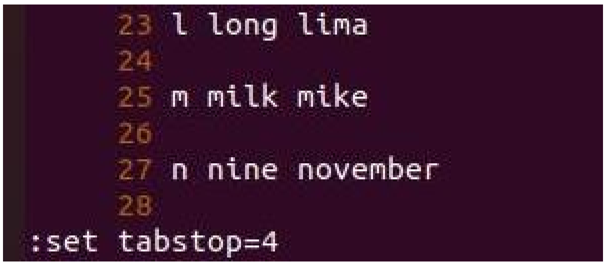
10.Turn online numbering and set your tab stop to 4.

vi 编辑器里包括定制 vi 会话的操作，在命令模式下按“：”进入末行模式，在末行模式下输入 set nu，可显示行号：



```
      1 a alice alpha
      2
      3 b bob bravo
      4
      5 c chloe charlie
      6
      7 d david delta
      8
      9 e emma echo
     10
     11 f frank foxtrot
     12
     13 g grace golf
     14
     15 h harry hotel
     16
     17 i island india
     18
     19 j jack juliette
     20
     21 k kill kilo
     22
     23 l long lima
     24
     25 m milk mike
     26
     27 n nine november
     28
:set nu
```

在末行模式下输入 set tabstop = 4，可将制表符宽度设置为 4：

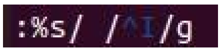


```
23 l long lima
24
25 m milk mike
26
27 n nine november
28
:set tabstop=4
```

Global search and replace

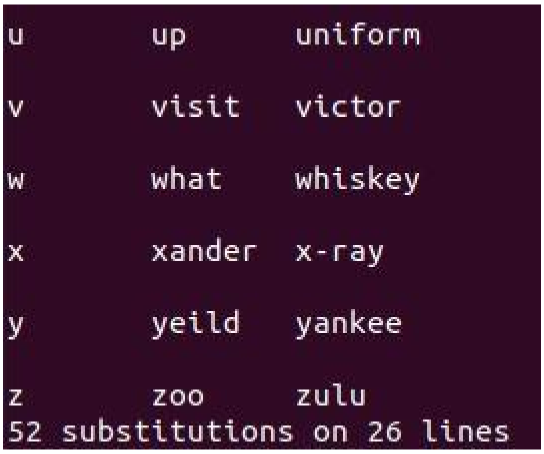
11.Replace all spaces in your file with tabs.

在命令模式下输入%s//<TAB 键>/g 可以用制表符替换文件中的所有空格：



```
:%s/ /^I/g
```

执行结果如下（此处仅截取部分）：



```
u       up      uniform

v       visit   victor

w       what    whiskey

x       xander  x-ray

y       yeild   yankee

z       zoo     zulu
52 substitutions on 26 lines
```

用制表符替换文件中所有空格操作成功。

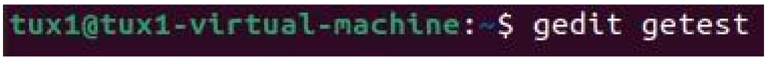
12.Save your file.

输入“：wq”即可保存文件。

Working with other editors

13.Your system has various other text mode and graphical editors available as well. Start some of these to get acquainted with them.

文本模式下编辑器除 vi 和 vim，还有 emacs、nano、ed 等，图形模式下的编辑器有 gedit、kwrite、Kate、xedit 等，下面进行一些使用 gedit 图形编辑器的操作，使用 gedit 编辑器创建一个 getest 文件并使用 gedit 编辑器进行编辑：



```
tux1@tux1-virtual-machine:~$ gedit getest
```

All editors listed in the course material might not be available or installed on your distribution.

Exercise4

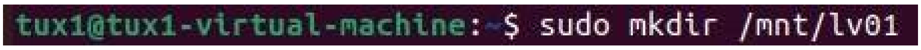
You now create various file systems in the logical volumes we made earlier. Note that /dev/vg00/lv00 already contains an ext2 file system.

1.Create an ext3 file system in /dev/vg00/lv01.

由于本次实验环境选择 Ubuntu 操作系统，并非实验要求的 Fedora，RHEL 或 SLES 操作系统，因此该命令无法正常执行。（不存在此磁盘分区）

2.Create a mount point, /mnt/lv01, for this file system.

首先创建一个目录用于挂载点，使用 mkdir 命令：

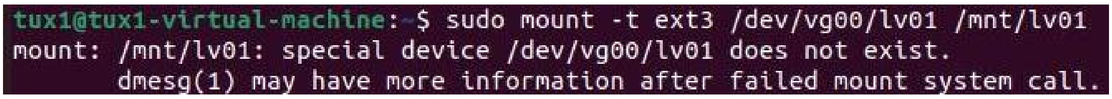


```
tux1@tux1-virtual-machine:~$ sudo mkdir /mnt/lv01
```

3.Mount the file system on this mount point.

mount 命令用于挂载文件系统，其基本命令格式为：

mount -t type [-o options] device dir

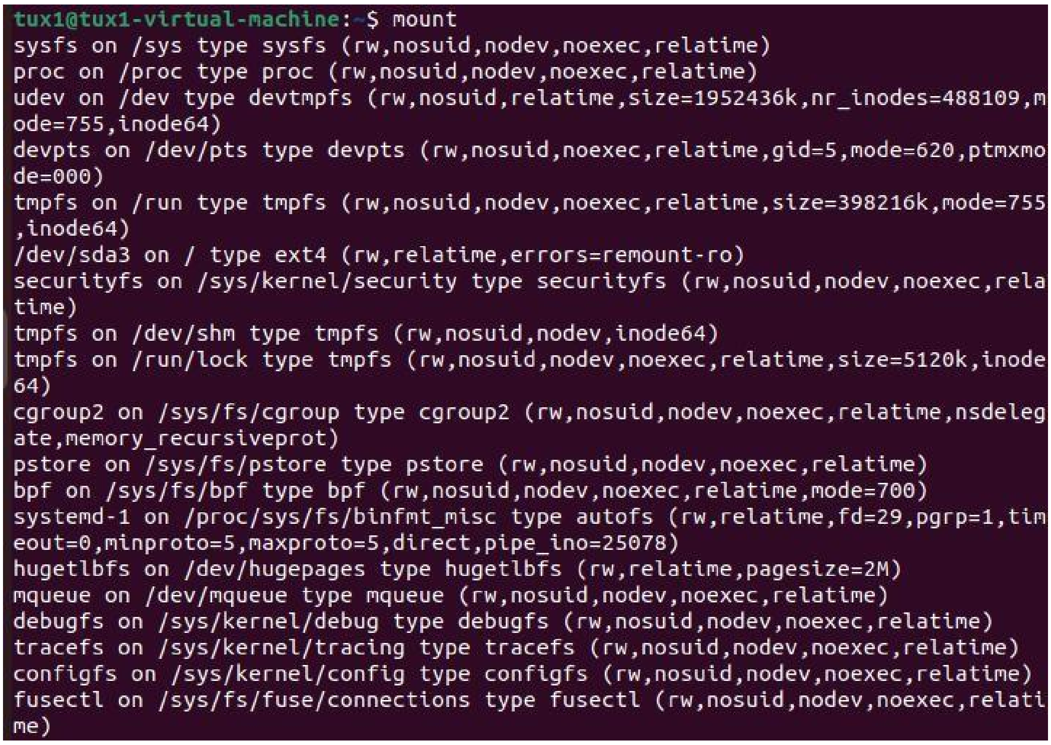


```
tux1@tux1-virtual-machine:~$ sudo mount -t ext3 /dev/vg00/lv01 /mnt/lv01
mount: /mnt/lv01: special device /dev/vg00/lv01 does not exist.
       dmesg(1) may have more information after failed mount system call.
```

由于本次实验环境选择 Ubuntu 操作系统，并非实验要求的 Fedora，RHEL 或 SLES 操作系统，因此该命令无法正常执行。

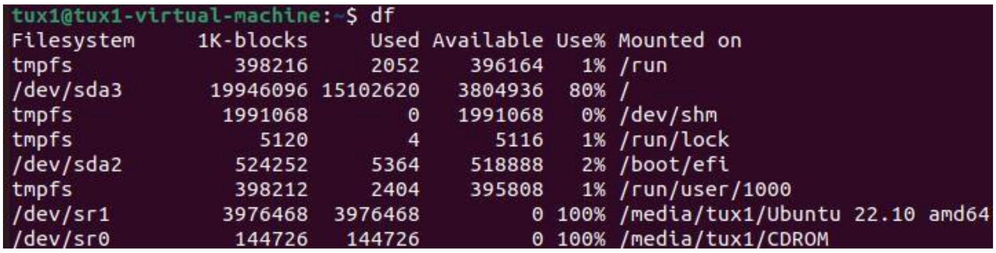
4.Issue the mount and df commands to see the mounted file systems. Can you see how much space the journals are taking? Also, list the root directory of the mounted file systems. Are the journals visible?

使用 mount 命令查看已挂载的文件系统：



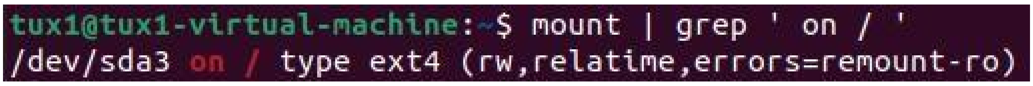
```
tux1@tux1-virtual-machine:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1952436k,nr_inodes=488109,m
ode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmo
de=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=398216k,mode=755
,inode64)
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,rela
time)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,inode
64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdeleg
ate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,tim
eout=0,minproto=5,maxproto=5,direct,pipe_ino=25078)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relati
me)
```

从中无法看到文件系统日志占用多少空间，使用 df 指令查看已挂载的文件系统：



```
tux1@tux1-virtual-machine:~$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
tmpfs             398216     2052    396164   1% /run
/dev/sda3       19946096 15102620   3804936  80% /
tmpfs            1991068        0   1991068   0% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
/dev/sda2         524252     5364    518888   2% /boot/efi
tmpfs             398212     2404    395808   1% /run/user/1000
/dev/sr1         3976468  3976468         0 100% /media/tux1/Ubuntu 22.10 amd64
/dev/sr0          144726   144726         0 100% /media/tux1/CDROM
```

挂载在/run 上的 tmpfs 文件系统占用空间大小为 398216KB。执行命令 mount | grep ‘ on/ ‘查看挂载在根目录的文件系统信息：

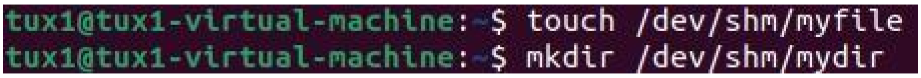


```
tux1@tux1-virtual-machine:~$ mount | grep ' on / '
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro)
```

挂载在根目录上的文件系统是/dev/sda3。

5.Create some files and/or directories on these mounted file systems.

选择挂载在目录/dev/shm 上的 tmpfs 文件系统，在其上创建文件和目录。使用 touch 命令创建一个 myfile 文件，使用 mkdir 命令创建一个 mydir 目录：

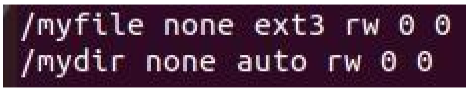


```
tux1@tux1-virtual-machine:~$ touch /dev/shm/myfile
tux1@tux1-virtual-machine:~$ mkdir /dev/shm/mydir
```

Automatically mounting file systems at system startup

File systems that you want to mount automatically at system startup should be listed in /etc/fstab.

6.Edit the /etc/fstab file to include all the previously created file systems.



```
/myfile none ext3 rw 0 0
/mydir none auto rw 0 0
```

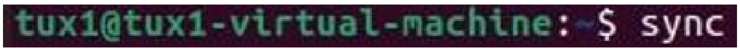
/myfie 无挂载点，文件系统类型为 ext3，权限为 rw，dump 为 0，pass 为 0。

/myfile 无挂载点，文件系统类型为 auto，权限为 rw，dump 为 0，pass 为 0。

<dump>是 dump 工具何时备份的决定指标，<pass>是 fsck 需要检查的文件系统的检查顺序的决定数值。

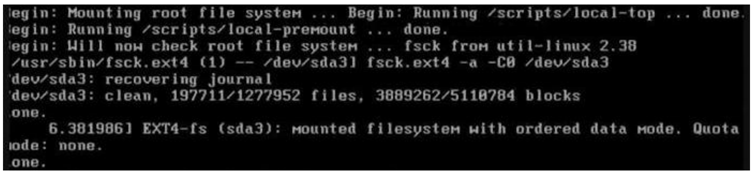
7.Issue the sync command to ensure that all data is written to disk. Wait five seconds, and then turn off your computer with the power off button. (Do not do a proper shutdown!) Turn on your computer and watch the boot messages. Which file system was checked the fastest?

执行 sync 命令，迫使缓存块数据立即写盘并更新超级快：



```
tux1@tux1-virtual-machine:~$ sync
```

等待 5 秒钟后关闭电脑，随后启动电脑，观察引导信息：



```
egin: Mounting root file system ... Begin: Running /scripts/local-top ... done.
egin: Running /scripts/local-premount ... done.
egin: Will now check root file system ... fsck from util-linux 2.38
/usr/sbin/fsck.ext4 (1) -- /dev/sda3] fsck.ext4 -a -C0 /dev/sda3
dev/sda3: recovering journal
dev/sda3: clean, 197711/1277952 files, 3889262/5110784 blocks
one.
    6.381986] EXT4-fs (sda3): mounted filesystem with ordered data mode. Quota
ode: none.
one.
```

Changing file system size

8.Using lvextend and resize2fs, increase the size of each file system in vg00 to 100 MB. When complete, verify the new file system size using df.

由于本次实验环境选择 Ubuntu 操作系统，并非实验要求的 Fedora，RHEL 或 SLES 操作系统，因此该命令无法正常执行。（不存在此磁盘分区）

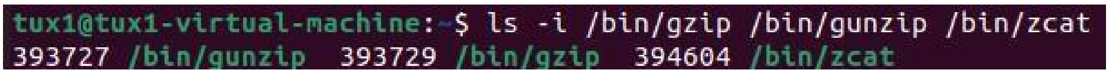
Note: The ext2 file system does not support online resizing, so it will have to be unmounted. The ext3 file system can be resized online (while mounted).

Working with files with multiple names

As seen in the lecture, a file can have multiple names, as long as each name refers to the same i-node. An example of this situation is /bin/gzip, /bin/gunzip, and /bin/zcat.

9.Verify that /bin/gzip, /bin/gunzip, and /bin/zcat indeed have the same i-node number.

通过 ls 命令的-i 选项可以查看文件的 i 节点号，-i 选项输出文件的 i 节点的索引信息：

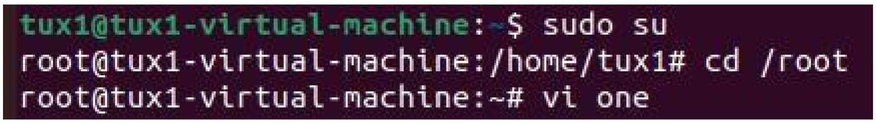


```
tux1@tux1-virtual-machine:~$ ls -i /bin/gzip /bin/gunzip /bin/zcat
393727 /bin/gunzip  393729 /bin/gzip  394604 /bin/zcat
```

如上图所示，/bin/gzip 的 i 节点号为 393727，/bin/gunzip 的 i 节点号为 393729，/bin/zcat 的 i 节点号为 394604。

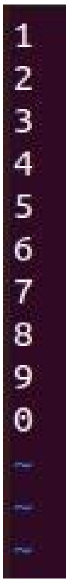
10.Go to /root and create a file named one, using vi. Put some text in the file, so that you can identify that file for yourself.

执行命令 sudo su 切换至 root 身份，使用 cd 命令进入/root 目录，随后使用 vi 编辑器创建一个名为 one 的文件：



```
tux1@tux1-virtual-machine:~$ sudo su
root@tux1-virtual-machine:/home/tux1# cd /root
root@tux1-virtual-machine:~# vi one
```

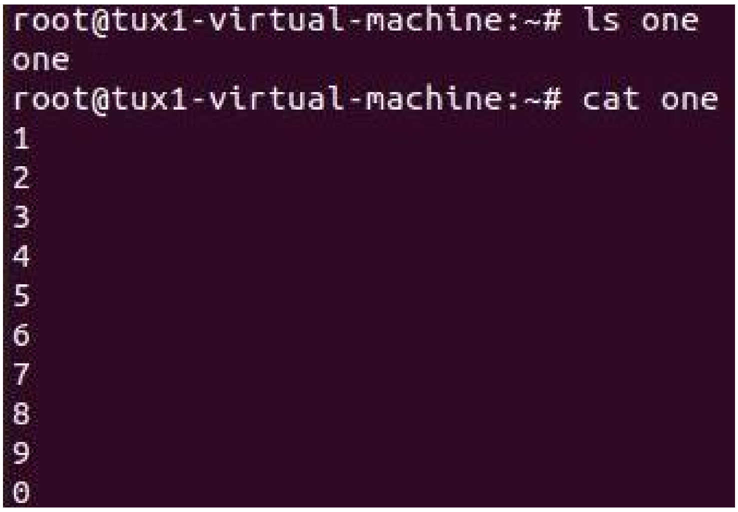
one 文件中的内容如下所示：



```
1
2
3
4
5
6
7
8
9
0
~
~
~
```

11.Verify the file name and the contents.

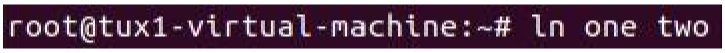
使用 ls 命令可以验证创建文件的文件名，使用 cat 命令可以验证创建文件的内容：



```
root@tux1-virtual-machine:~# ls one
one
root@tux1-virtual-machine:~# cat one
1
2
3
4
5
6
7
8
9
0
```

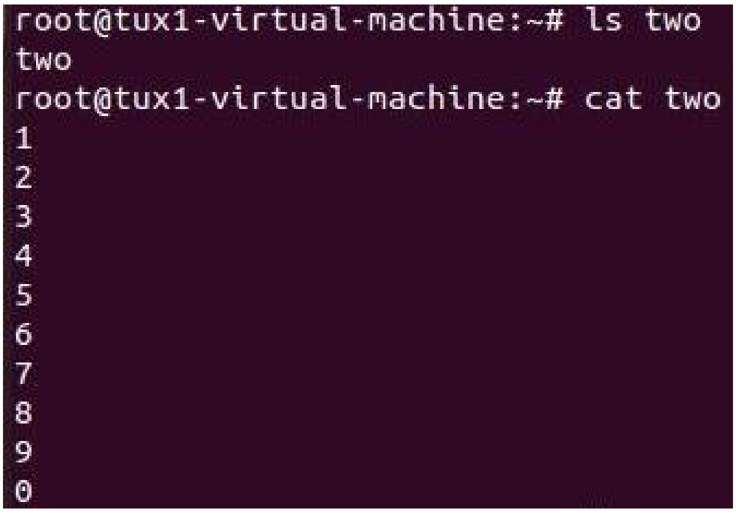
12.Now create a second link to the file (a second file name) named two. Verify the file name and the contents.

使用 ln 命令可以创建一个硬链接，硬链接是指不同的文件名指向同一个文件的数据块：



```
root@tux1-virtual-machine:~# ln one two
```

使用 ls 命令可以验证文件的文件名，使用 cat 命令可以验证创建文件的内容：

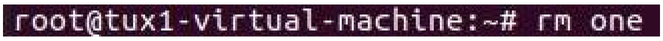


```
root@tux1-virtual-machine:~# ls two
two
root@tux1-virtual-machine:~# cat two
1
2
3
4
5
6
7
8
9
0
```

文件 two 存在且文件内容与文件 one 相同，说明链接创建成功。

13.Delete the file one.

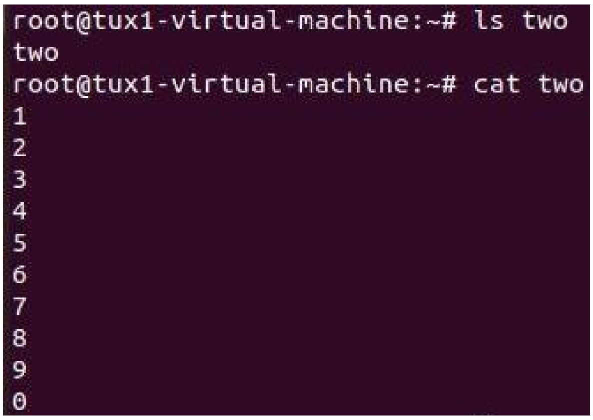
使用 rm 命令删除文件 one：



```
root@tux1-virtual-machine:~# rm one
```

14.Verify the file name and contents of two. Note that the link count has dropped.

使用 ls 命令验证文件 two 的文件名，使用 cat 命令验证文件 two 的内容：

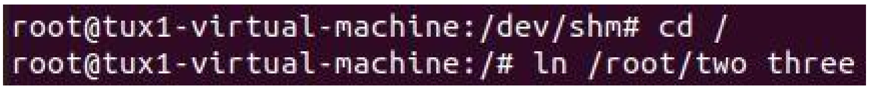


```
root@tux1-virtual-machine:~# ls two
two
root@tux1-virtual-machine:~# cat two
1
2
3
4
5
6
7
8
9
0
```

文件 two 仍然存在，其文件内容仍可查询到且不变，说明文件 one 删除后，硬链接创建的文件仍然存在，只是文件的 link count 从 2 下降至 1。

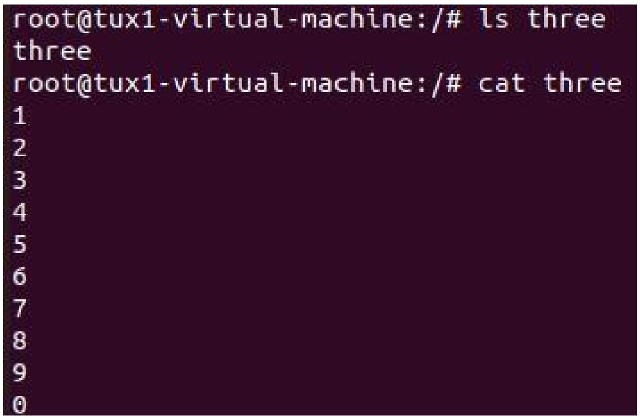
15.Create a third link, called three, in the / directory. It should again point to the original file. Verify the file name and contents of this file.

执行 cd 命令进入根目录，使用 ln 命令创建第三个链接。由于文件 one 已经删除，因此若要再次指向原始文件，应当与文件 two 进行链接：



```
root@tux1-virtual-machine:/dev/shm# cd /
root@tux1-virtual-machine:/# ln /root/two three
```

使用 ls 命令验证文件 three 的文件名，使用 cat 命令验证文件 three 的内容：

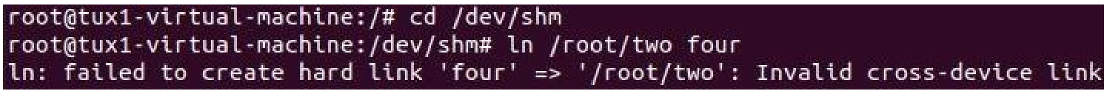


```
root@tux1-virtual-machine:/# ls three
three
root@tux1-virtual-machine:/# cat three
1
2
3
4
5
6
7
8
9
0
```

文件 three 存在且文件内容与文件 two 相同，说明链接创建成功。

16.Now try to create a fourth link, named four, on one of the file systems you just created.

执行 cd 命令进入/dev/shm 目录，使用 ln 命令创建第四个链接。由于文件 one 已删除，因此若要再次指向原始文件，应当与文件 two 进行链接：

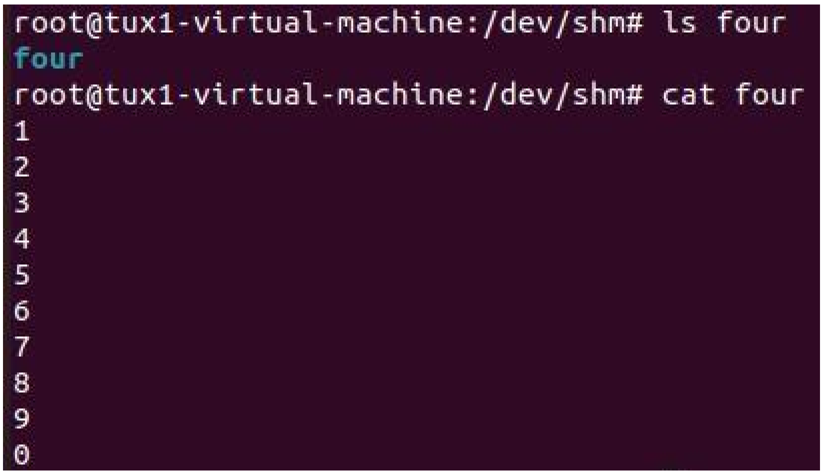


```
root@tux1-virtual-machine:/# cd /dev/shm
root@tux1-virtual-machine:/dev/shm# ln /root/two four
ln: failed to create hard link 'four' => '/root/two': Invalid cross-device link
```

命令执行出错（硬链接创建失败），说明硬链接只能在同一个文件系统中创建。

17.Create a symbolic link from /root/two to /mnt/lv00/four. Read the manual page for ln if necessary. Verify the name and contents of the file.

使用 ln 命令创建一个从/root/two 到/dev/shm/four 的符号链接，选项-s 指定此链接为软链接；使用 ls 命令验证文件 four 的文件名，使用 cat 命令验证文件 four 的内容：

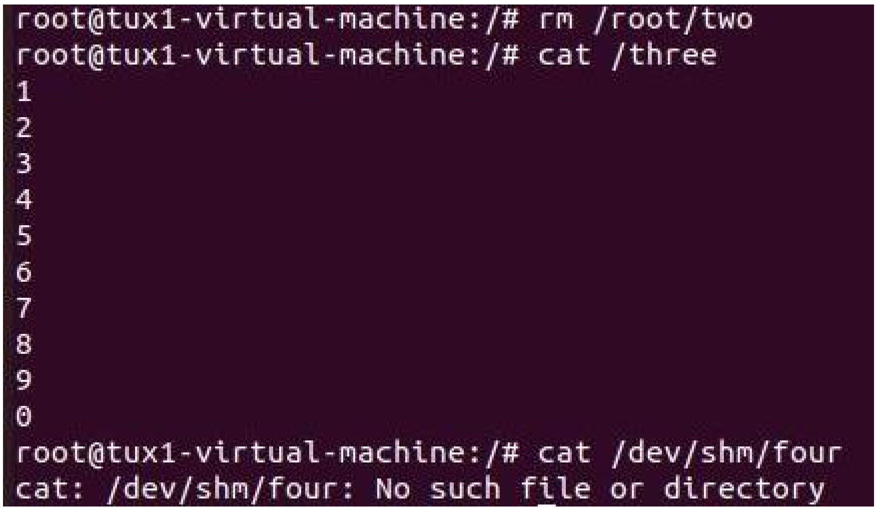


```
root@tux1-virtual-machine:/dev/shm# ls four
four
root@tux1-virtual-machine:/dev/shm# cat four
1
2
3
4
5
6
7
8
9
0
```

文件 four 存在且文件内容与文件 two 相同，说明链接创建成功。

18.Now delete /root/two, and verify the contents of both /tmp/three and /mnt/lv00/four.

使用 rm 命令删除/root/two，随后利用 cat 命令分别查看/three 和/dev/shm/four 中的内容：



```
root@tux1-virtual-machine:/# rm /root/two
root@tux1-virtual-machine:/# cat /three
1
2
3
4
5
6
7
8
9
0
root@tux1-virtual-machine:/# cat /dev/shm/four
cat: /dev/shm/four: No such file or directory
```

当文件 two 删除后，硬链接仍然存在，但符号链接失效。

Configuring user quota

19.Remount the /home file system so that quota are enabled, and ensure that quotas are enabled when the system reboots.

由于本次实验环境选择 Ubuntu 操作系统，并非实验要求的 Fedora，RHEL 或 SLES 操作系统，因此该命令无法正常执行。（/home 文件系统未挂载）

20.Perform a recalculation of current quotas and turn quota checking on.

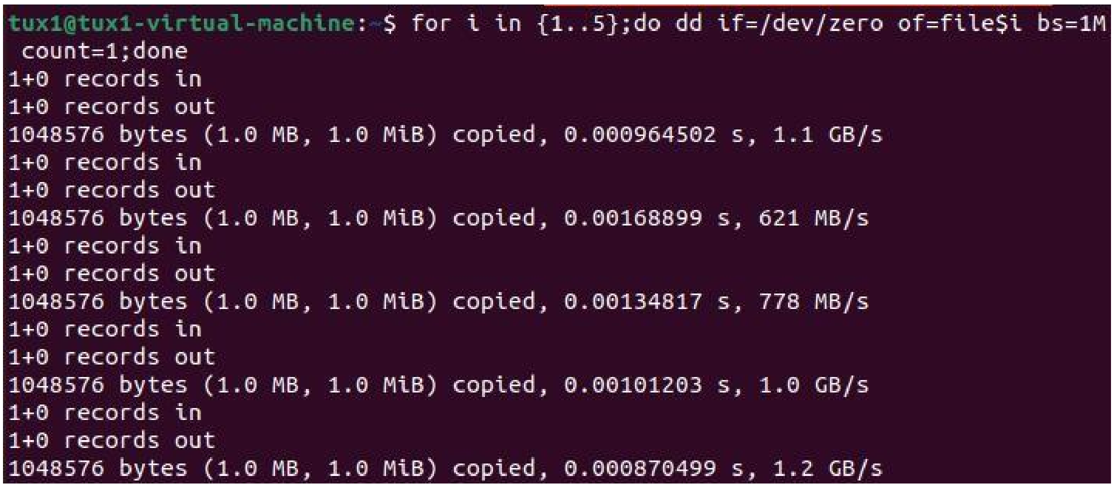
由于本次实验环境选择 Ubuntu 操作系统，并非实验要求的 Fedora，RHEL 或 SLES 操作系统，因此该命令无法正常执行。(/home 文件系统未挂载)

21.Set the soft limit of users tux1 and tux2 to 4 MB and the hard limit to 5 MB.

由于本次实验环境选择 Ubuntu 操作系统，并非实验要求的 Fedora，RHEL 或 SLES 操作系统，因此该命令无法正常执行。(未检测到配额的文件系统，/home 文件系统未挂载)

22.On another virtual terminal, log in as tux1, and create five files of 1 MB each. Watch what happens.

在另一个虚拟终端上仍以 tux1 身份登录，并使用 dd 命令同时创建 5 个大小为 1MB 的文件：



```
tux1@tux1-virtual-machine:~$ for i in {1..5};do dd if=/dev/zero of=file$i bs=1M
 count=1;done
1+0 records in
1+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.000964502 s, 1.1 GB/s
1+0 records in
1+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00168899 s, 621 MB/s
1+0 records in
1+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00134817 s, 778 MB/s
1+0 records in
1+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00101203 s, 1.0 GB/s
1+0 records in
1+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.000870499 s, 1.2 GB/s
```

当同时创建 5 个大小为 1MB 的文件时，由于没有超过硬限制，因此可以创建成功，但同时由于超过软限制，会发出警告，提醒超过软限制，须在宽限时间内删除多余文件。

23.Still logged in as tux1, check the quota.

由于本次实验环境选择 Ubuntu 操作系统，并非实验要求的 Fedora，RHEL 或 SLES 操作系统，因此该命令无法正常执行。(无输出结果，因为未检测到配额文件)

24.Switch back to your root session and check the quota of all users.

由于本次实验环境选择 Ubuntu 操作系统，并非实验要求的 Fedora，RHEL 或 SLES 操作系统，因此该命令无法正常执行。(/home 文件系统未挂载)