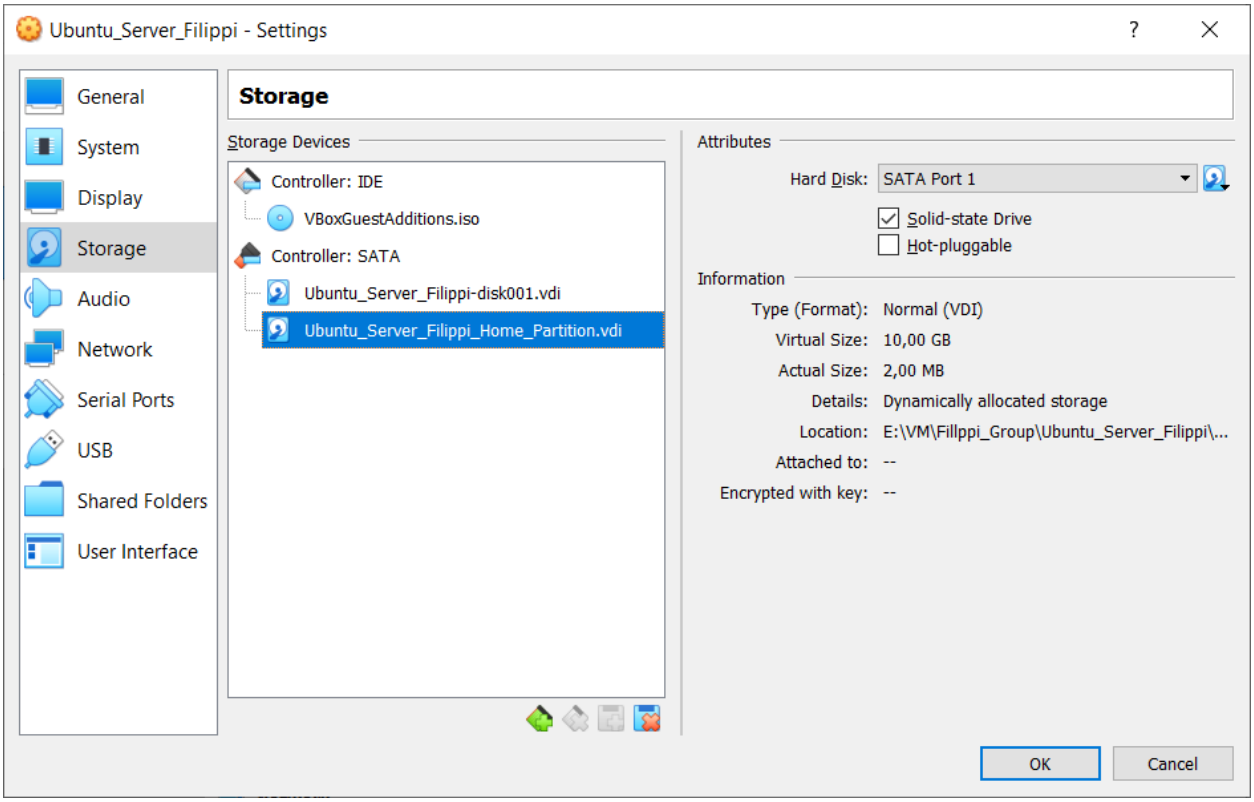
EPAM University Programs
DevOps external course
Module 4 Linux & Bash Essentials
TASK 4.7

Part1. **Quota allocation mechanism.**

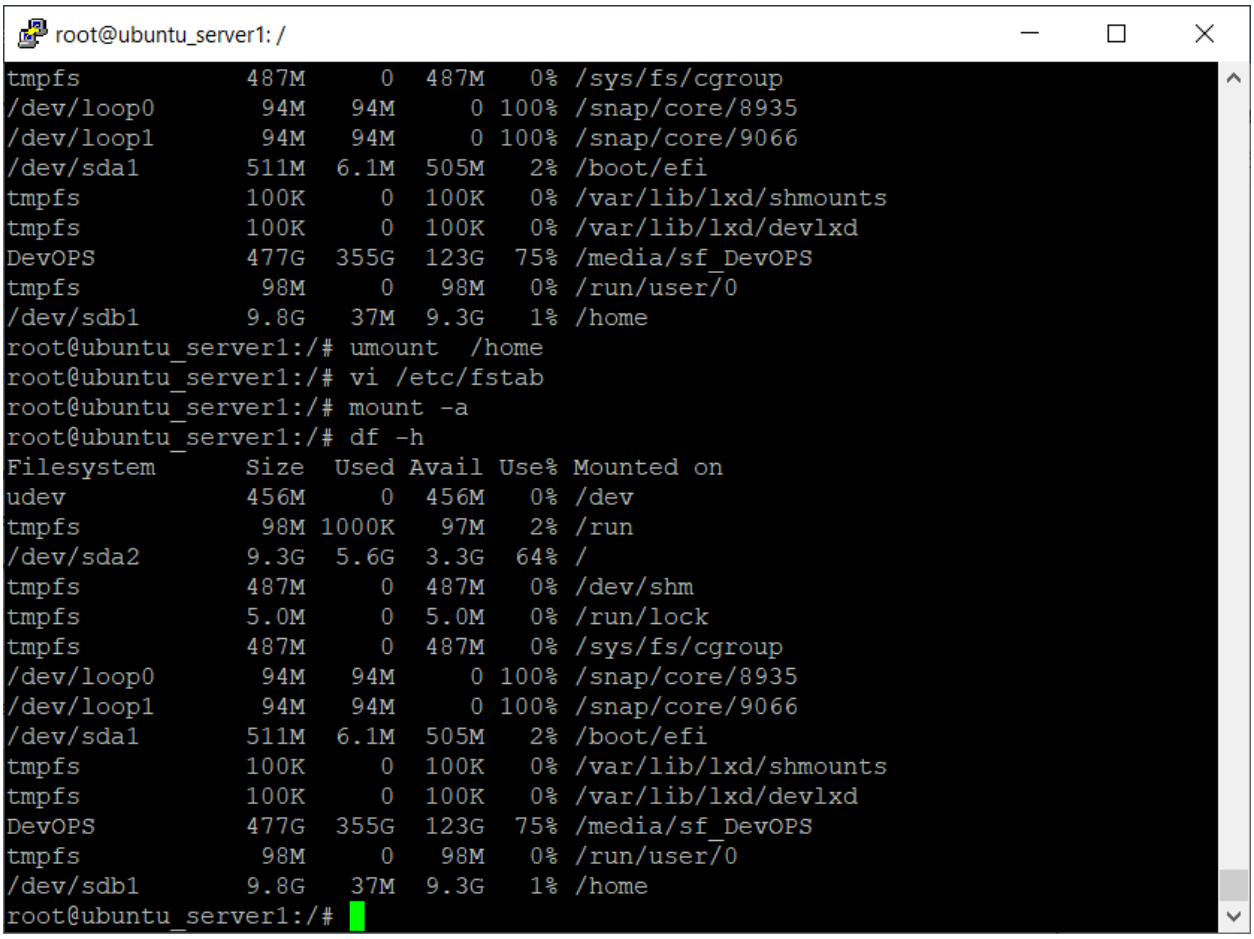
Employing commands from presentation #4.6, create a new user, say, *utest*. Based on the quota mechanism, limit the available disk space for this user to **soft**: 100M and **hard**: 150M.

Add Additional storage to my VM



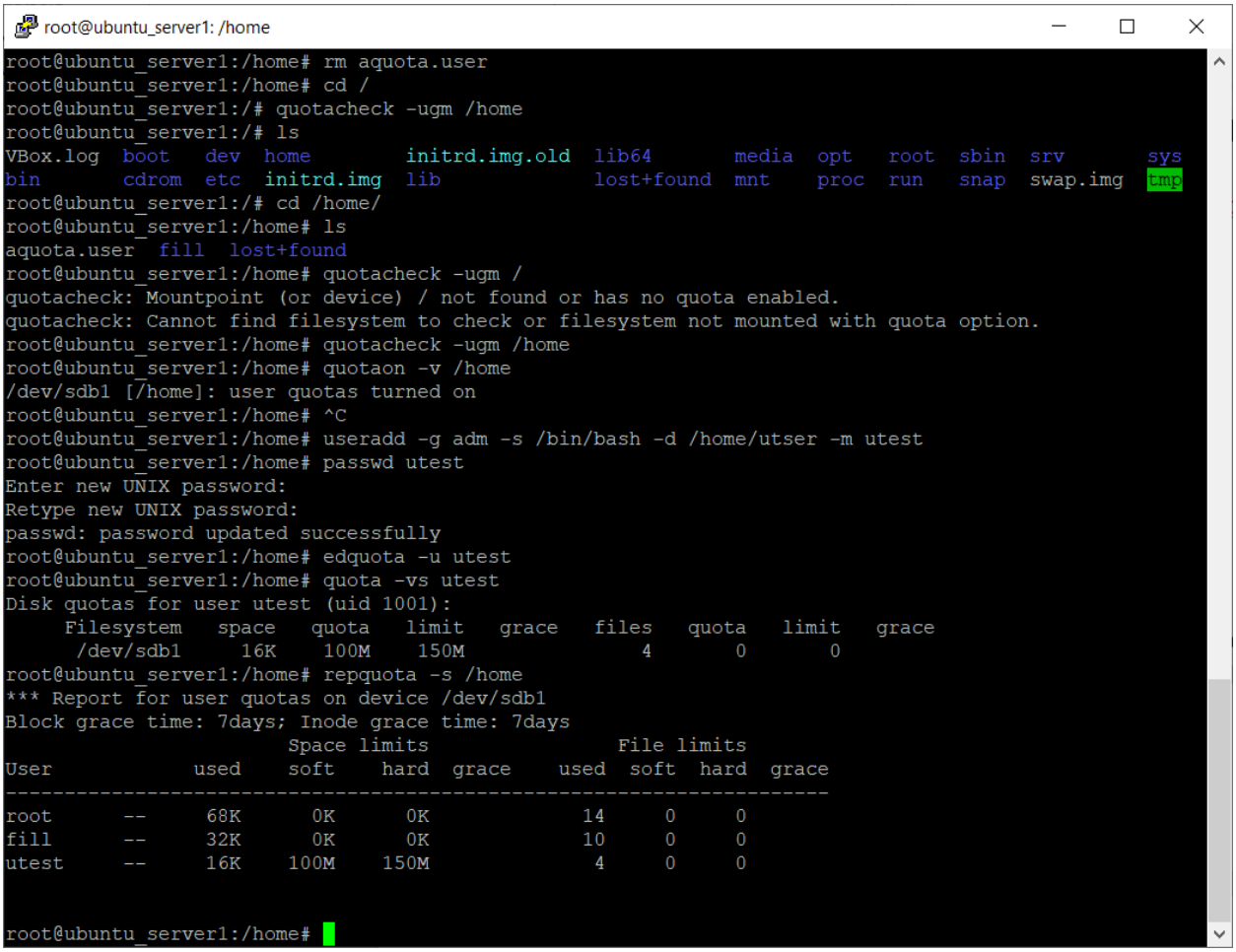
Next actions to add disk as /Home with usrquota

```
lshw -C disk                # Checking the location of the new drive. It is
/dev/sdb.
parted /dev/sdb mklabel gpt # Creating the GUID Partition Table (GPT)
parted /dev/sdb print       # Checking that the GPT has been created
parted /dev/sdb print unit MB print free # see 10737MB size
parted --align optimal /dev/sdb mkpart primary ext4 0% 10737MB # Creating
partition label
mkfs.ext4 /dev/sdb1         # Creating the partition
vi /etc/fstab             # Add line `/dev/sdb1 /home ext4 usrquota 0 1`
mount -a # Remount /etc/fstab without rebooting
```



I use for https://www.digitalocean.com/community/tutorials/how-to-set-filesystem-quotas-on-ubuntu-18-04 quota configure

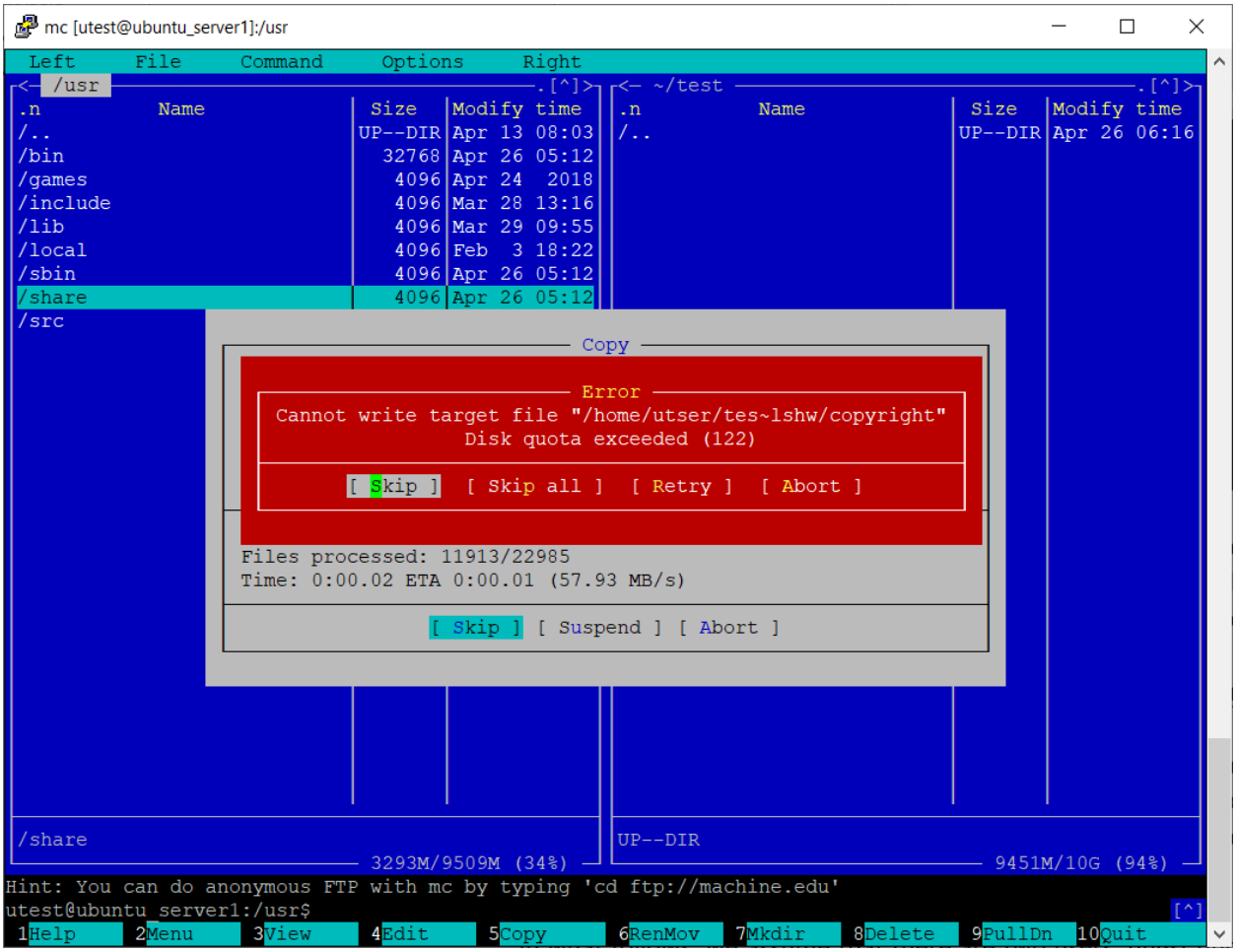
Main commands:
quotacheck -ugm /home
quotaon -v /home
useradd -g adm -s /bin/bash -d /home/utser -m utest
edquota -u utest
quota -vs utest
repquota -s /home

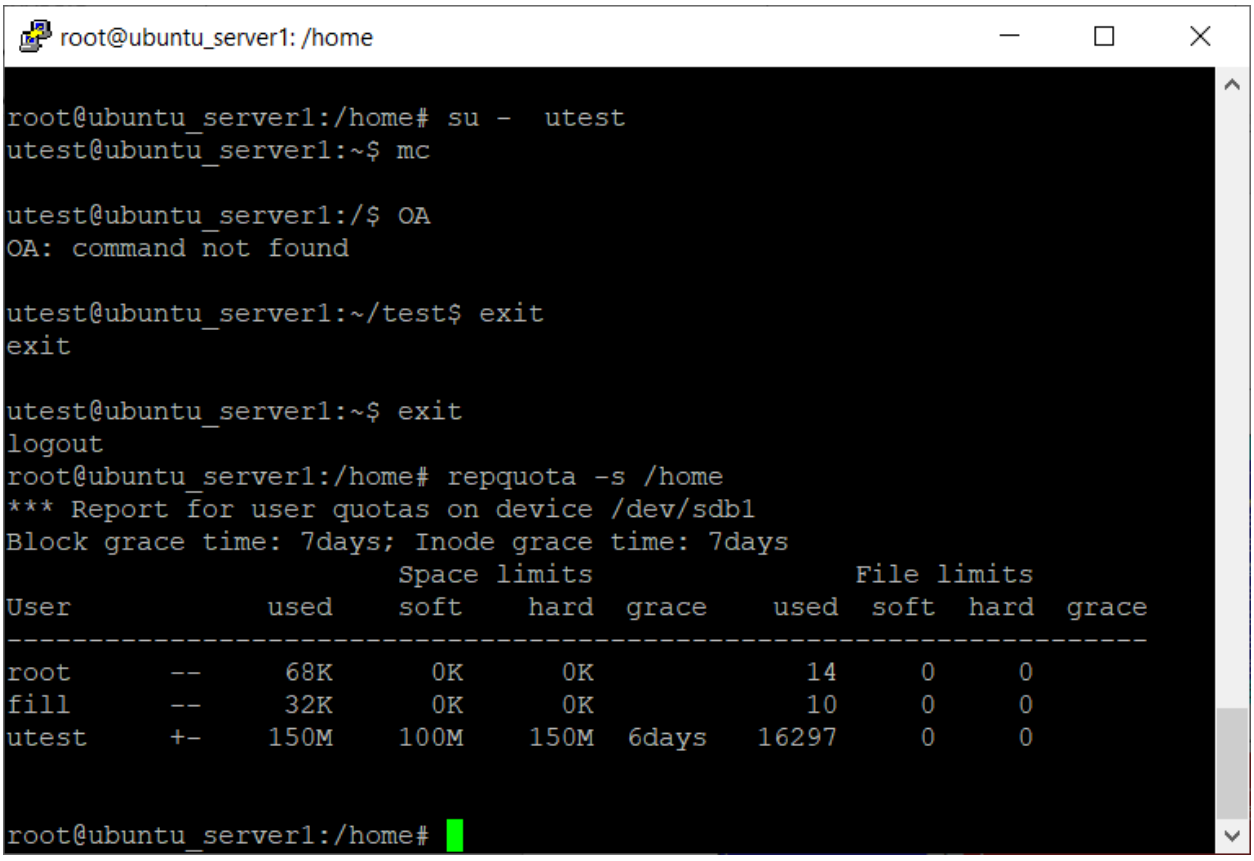


Then, using Midnight Commander (since MC shows warnings about exceeding the limits of available to a user disk space), copy content of /usr directory to utest's home directory (actually, /usr isn't mandatory, you are free to copy any other data, the only condition is sufficient total size of the files to copy).

Note: if /home is not a mount point, then the **mount** and **quotaon** commands should be called with respect to the root partition /.

Note 2: Please, put into your report screenshots of your terminal window with the executed commands, along with screenshots of MC panels over which quota warnings are shown (i.e. warnings about exceeding soft and hard limits).


mc [utest@ubuntu_server1]:/usr
Left File Command Options Right
/usr
.n Name Size Modify time
/.. UP--DIR Apr 13 08:03
/bin 32768 Apr 26 05:12
/games 4096 Apr 24 2018
/include 4096 Mar 28 13:16
/lib 4096 Mar 29 09:55
/local 4096 Feb 3 18:22
/sbin 4096 Apr 26 05:12
/share 4096 Apr 26 05:12
/src
~/test
.n Name Size Modify time
/.. UP--DIR Apr 26 06:16
Copy
Error
Cannot write target file "/home/utser/tes~lshw/copyright"
Disk quota exceeded (122)
[Skip] [Skip all] [Retry] [Abort]
Files processed: 11913/22985
Time: 0:00.02 ETA 0:00.01 (57.93 MB/s)
[Skip] [Suspend] [Abort]
/share
3293M/9509M (34%)
UP--DIR
9451M/10G (94%)
Hint: You can do anonymous FTP with mc by typing 'cd ftp://machine.edu'
utest@ubuntu_server1:/usr$
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit



Part2. **Access Control Lists, ACLs**

In what follows, we assume that there are two users: *guest* (included into the list of sudoers) and *utest*.

Command: usermod -aG sudo guest

None of the users is the superuser (i.e. UIDs of the users differ from 0).

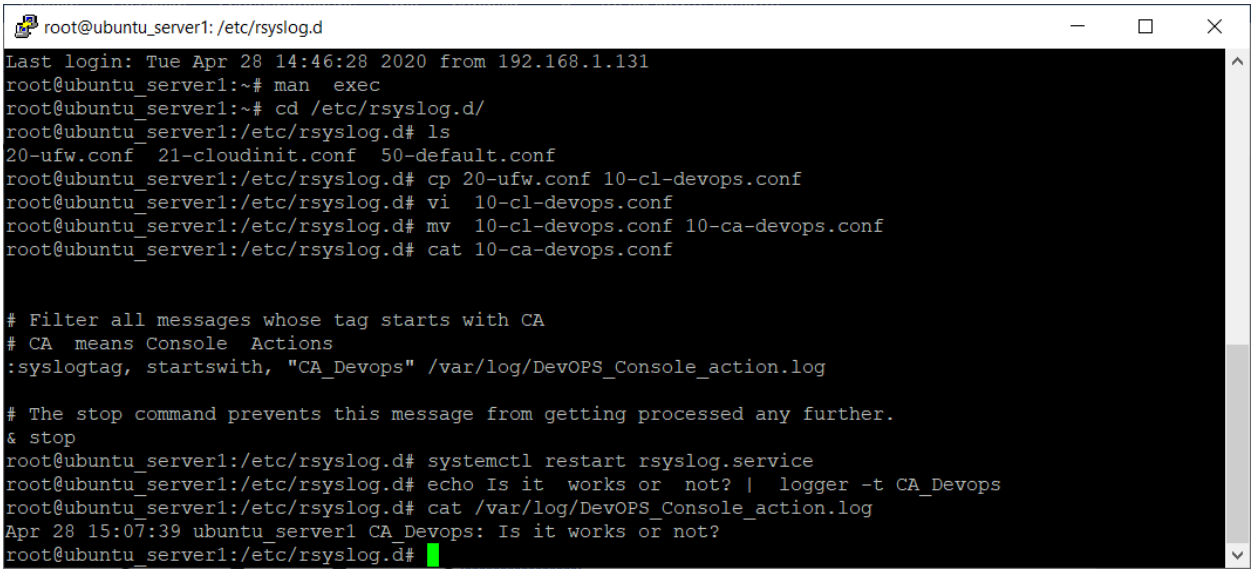
The most task: to allow user *utest* visit *guest*'s home directory.

The average task: to acquaint yourself with the basics of ACL and verify the fact that ACL privileges override the **chmod** ones.

Before proceeding to the task execution, please, visit the linux.org page describing ACL, https://linuxconfig.org/how-to-manage-acls-on-linux.

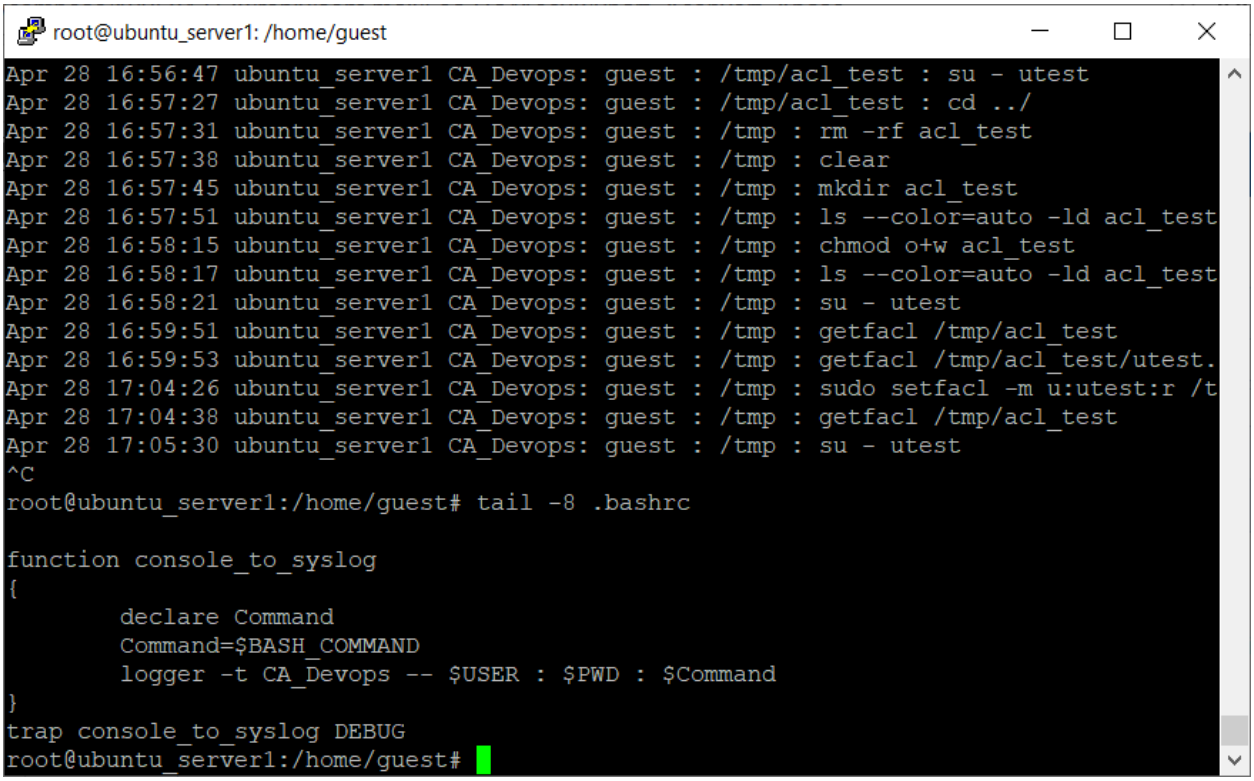
Every step of execution should be stored into some file **/var/log** directory (use logger, please).

Modify Syslogd configuration to put all log with tag "CA_Devops" to file /var/log/DevOPS_Console_action.log and check it



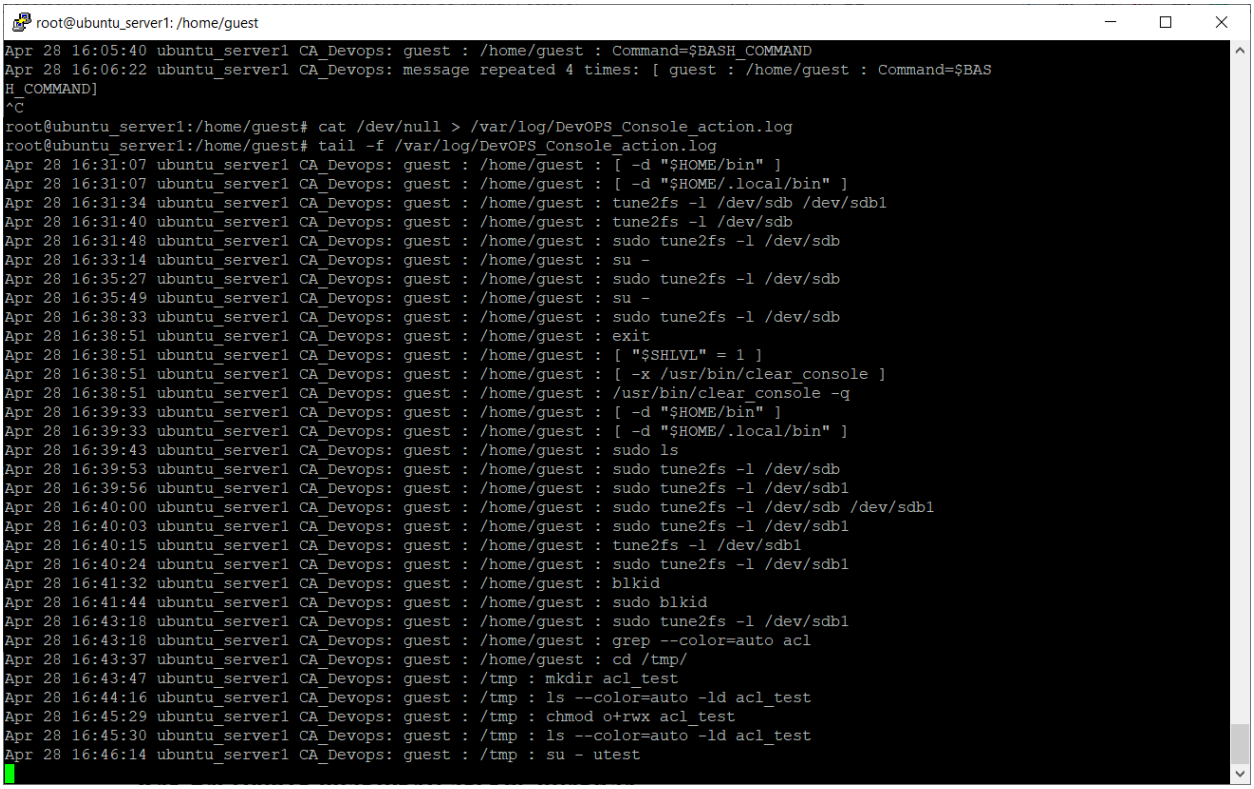
Use the Bash "trap" feature with signal DEBUG and my function.
Add to .bashrc file these lines for necessary user (I added it to guest user)

```
function console_to_syslog
{
    declare Command
    Command=$BASH_COMMAND
    logger -t CA_Devops -- $USER : $PWD : $Command
}
trap console_to_syslog DEBUG
```

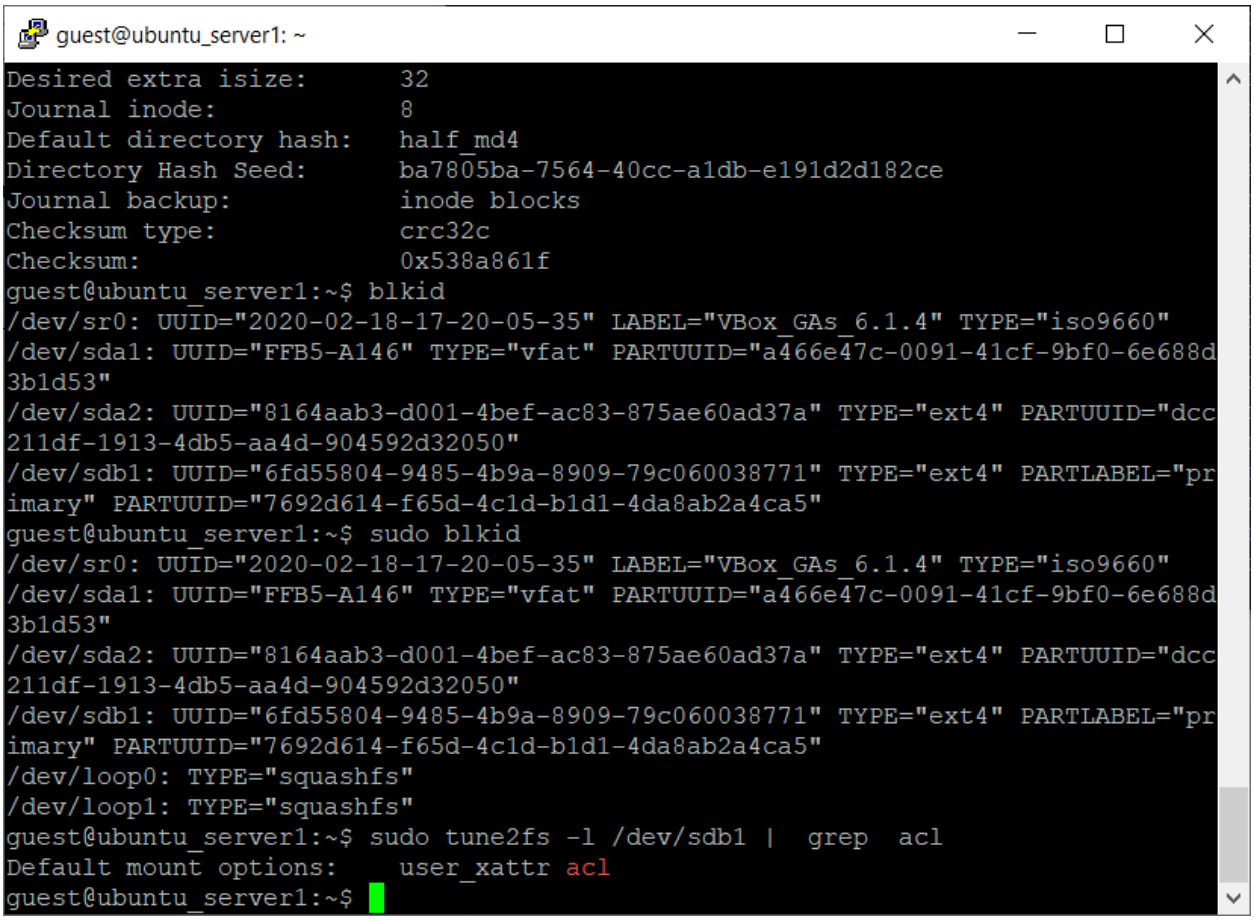


The goal is to use the trap feature and call a function each time the user generates activity.

And receive "Every step of execution")) in file /var/log/ DevOPS_Console_action.log



1. Based on given in presentation #4.7 instructions, turn on and set up the ACL. *Caution*! The fact that a file system has been mounted with the "acl" flag on by default, doesn't mean that the ACL package is installed.
Prior to any action, it is advised to check if the "acl" flag is on, using
tune2fs -l /dev/sda*
(a particular name of the device file sda*, is to be determined by calling to **blkid**, invoke it twice:
(i) on behalf of *guest* (i.e. without the superuser privileges);
(ii) with **sudo** (i.e. with the superuser privileges). Note the level of details provided by different **blkid** outputs).

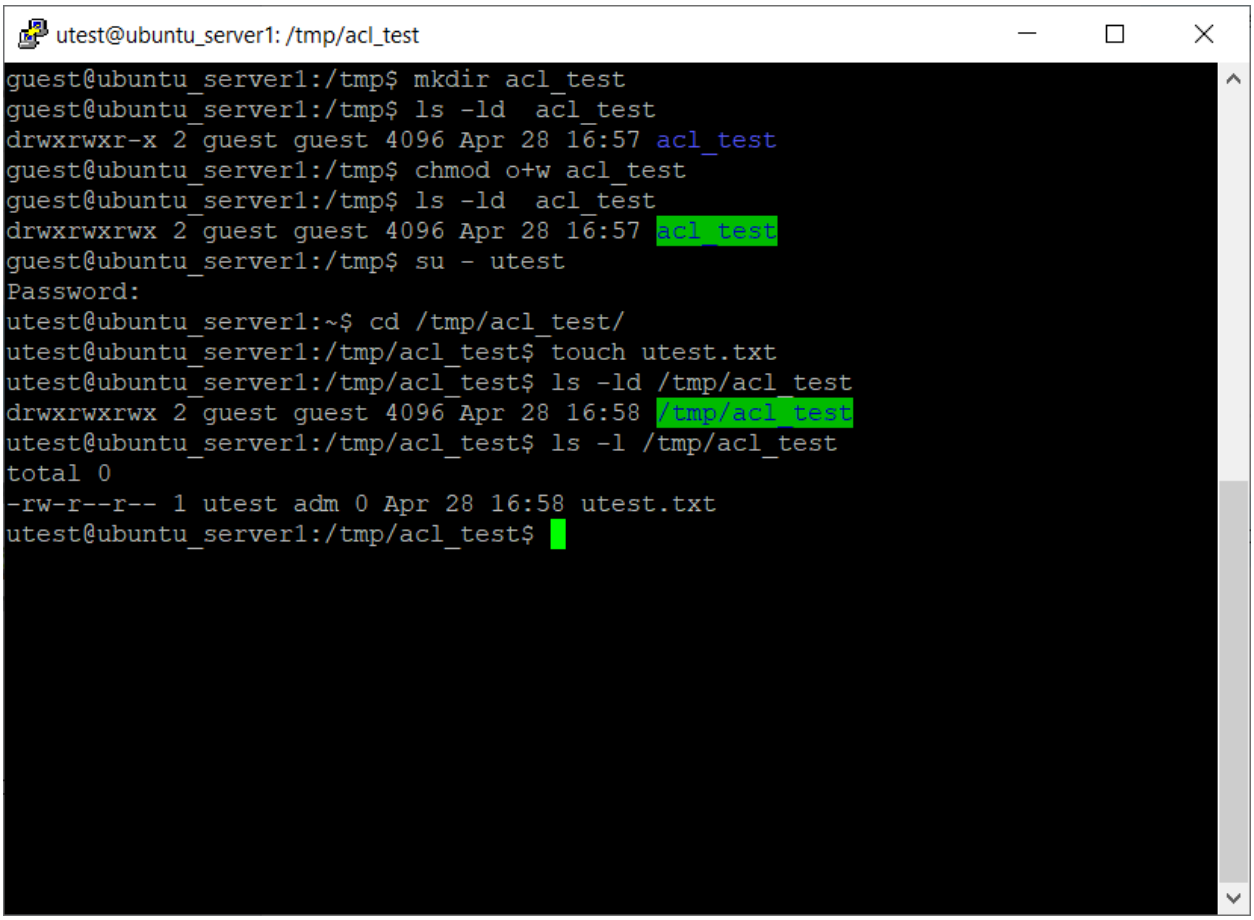


2. Log in as *guest*. Create in */tmp* a directory called *acl_test*. By means of **chmod,** allow user utest to perform all possible operations (rwx) with respect to *acl_test.* Verify that user *utest* is indeed capable of implementing granted him (her)

privileges. For example, acer logging in as *utest*, create a file in */tmp/acl_test*, say, *utest.txt* with the aid of **touch**. Query information about the directory and file by calling to

ls -ld /tmp/acl_test

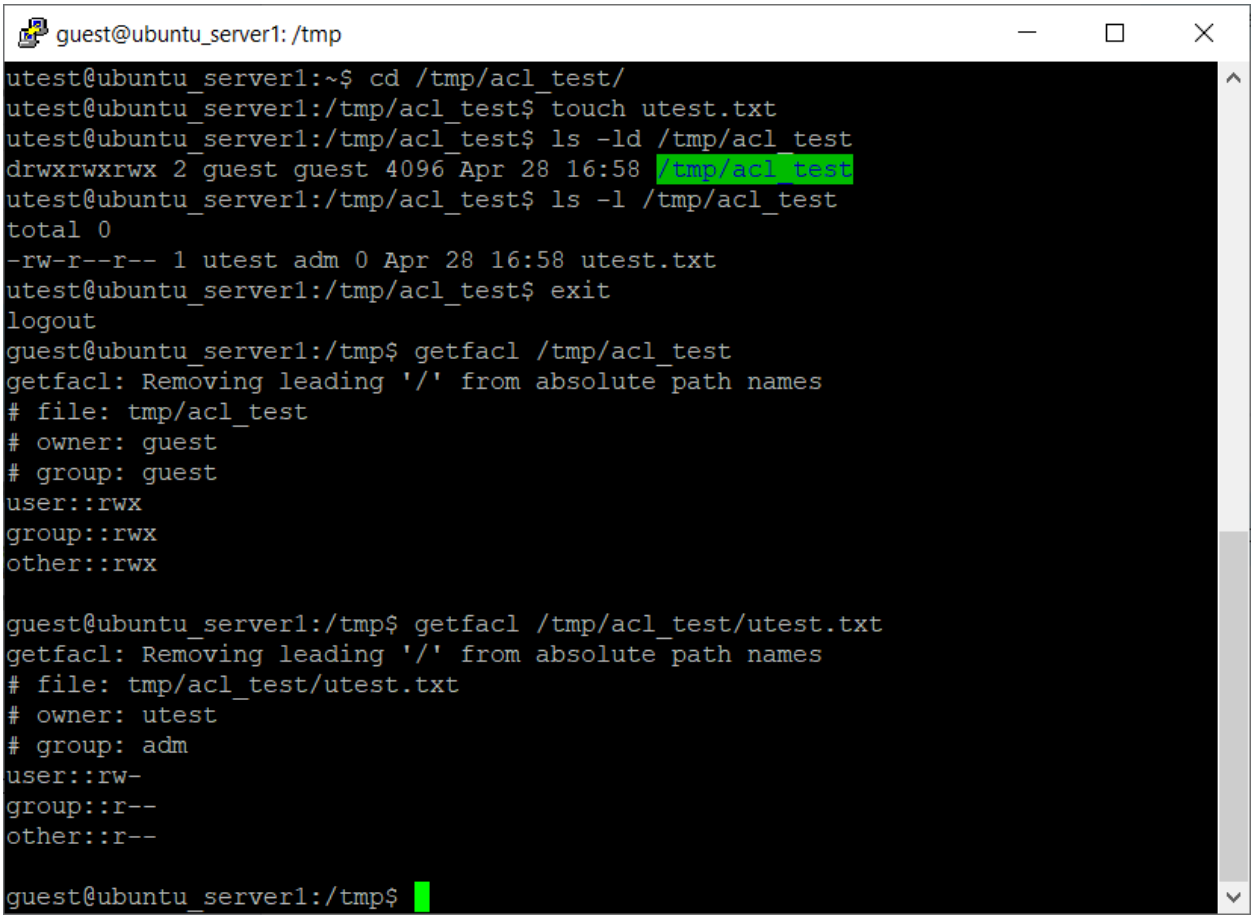
ls -l /tmp/acl_test



To check ACL permissions do:

getfacl /tmp/acl_test

getfacl /tmp/acl_test/utest.txt



3. Employ ACL to block any activity except for reading, for user *utest* with respect to directory */tmp/acl_test* (hint: use **setfacl**). Test if the actions are effectively prohibited

It is needed to use setfacl -m u:utest:rx /tmp/acl_test

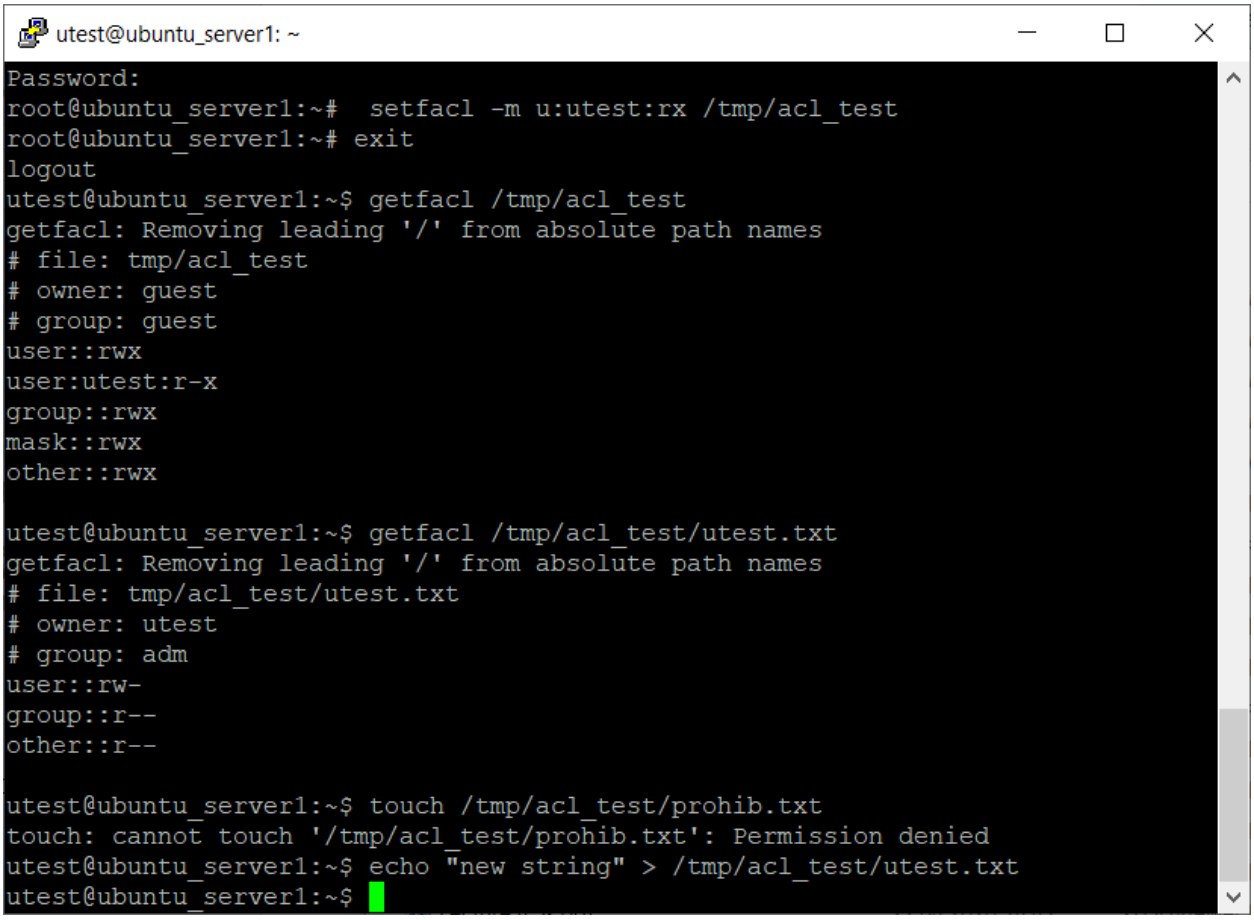
(rx – r - read directory list, x - enter to directory)

touch /tmp/acl_test/prohibited.txt

Is it possible to invoke this command?

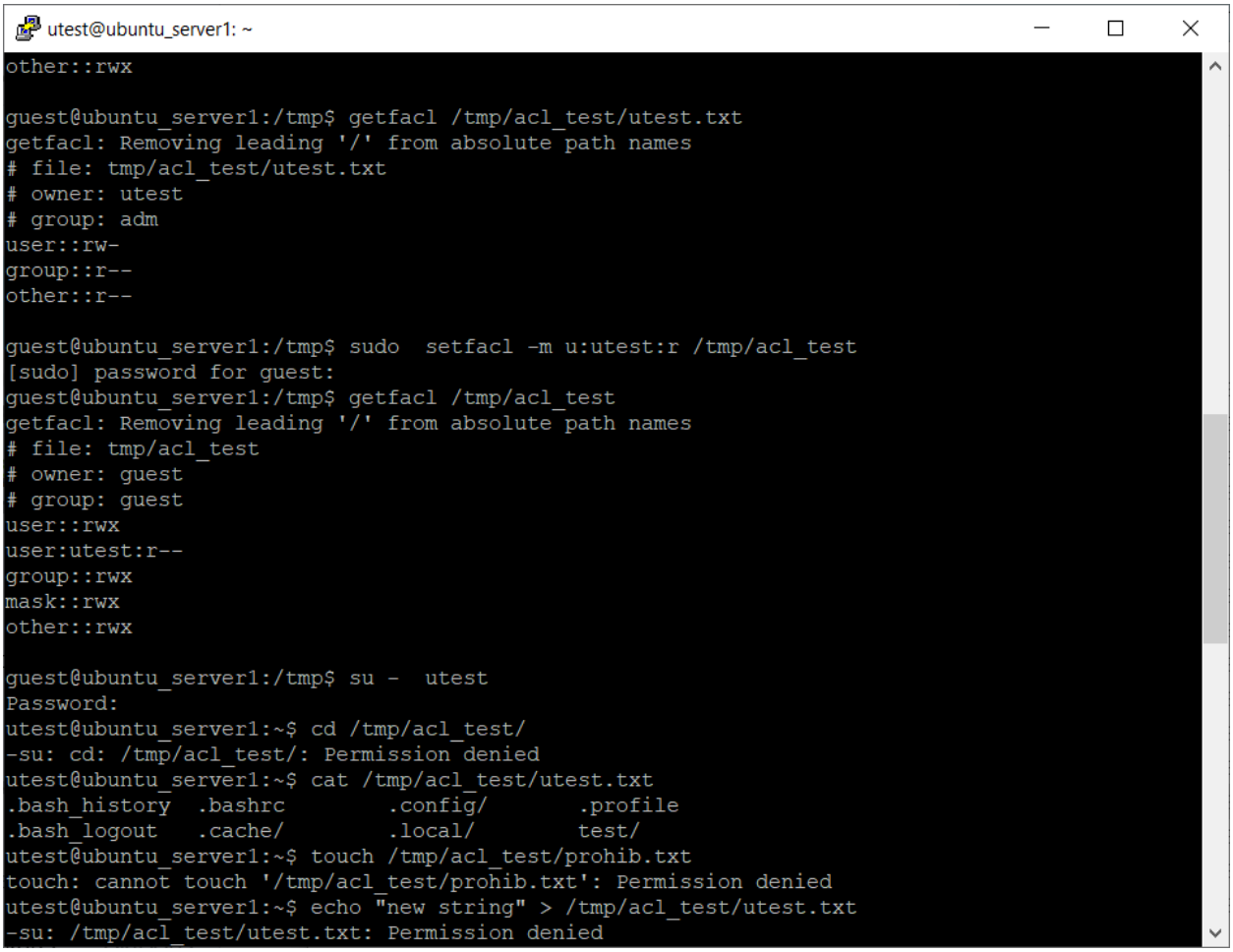
Access denied

echo “new content” > /tmp/acl_test/utest.txt



Test if user *utest* can be prevented from modifying content of the file *utest.txt* by means of ACL. (Note that user *utest* is the owner of the file *tmp/acl_test/utest.txt*).

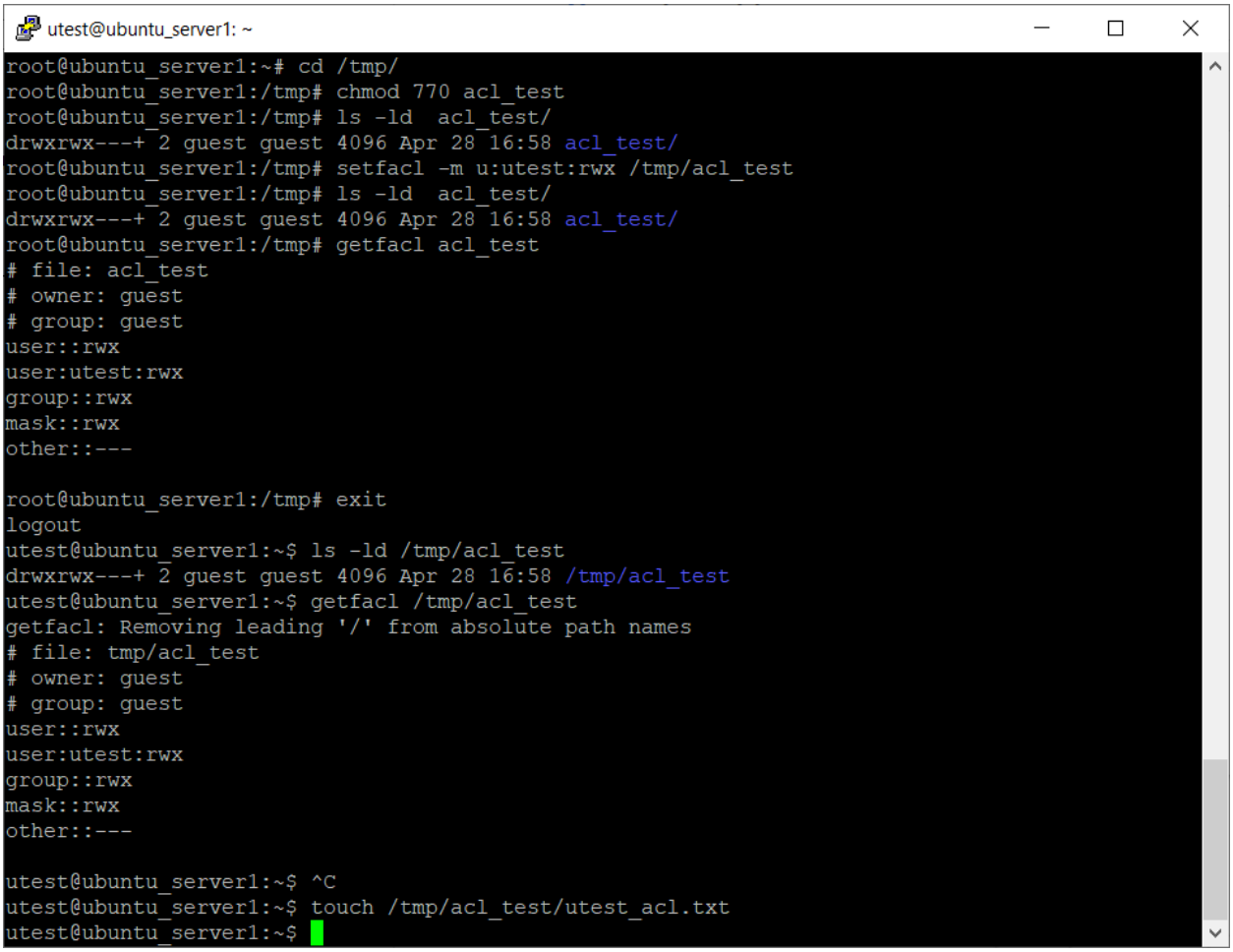
If I set permission for dir /tmp/acl_test r only : setfacl -m u:utest:rx /tmp/acl_test
I can not work with the files inside /tmp/acl_test
Just see files via ls /tmp/acl_test



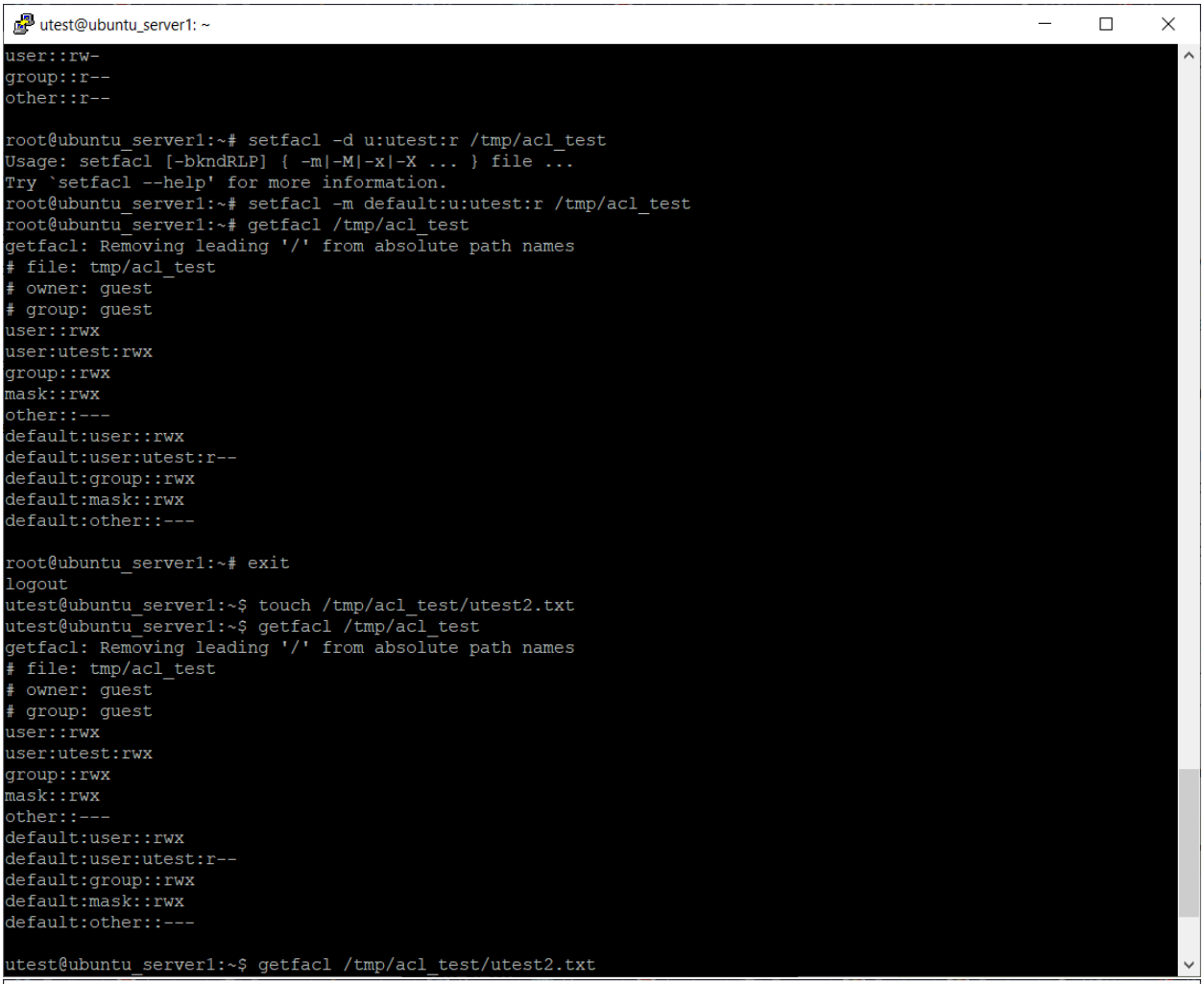
4. Consider a situation when at the ACL level user *utest* is allowed to have all possible privileges with respect to */tmp/acl_test*, while no action is allowed with **chmod** (conventional mechanism). (Hint: repeat step 3, but given the new context).

chmod 770 acl_test

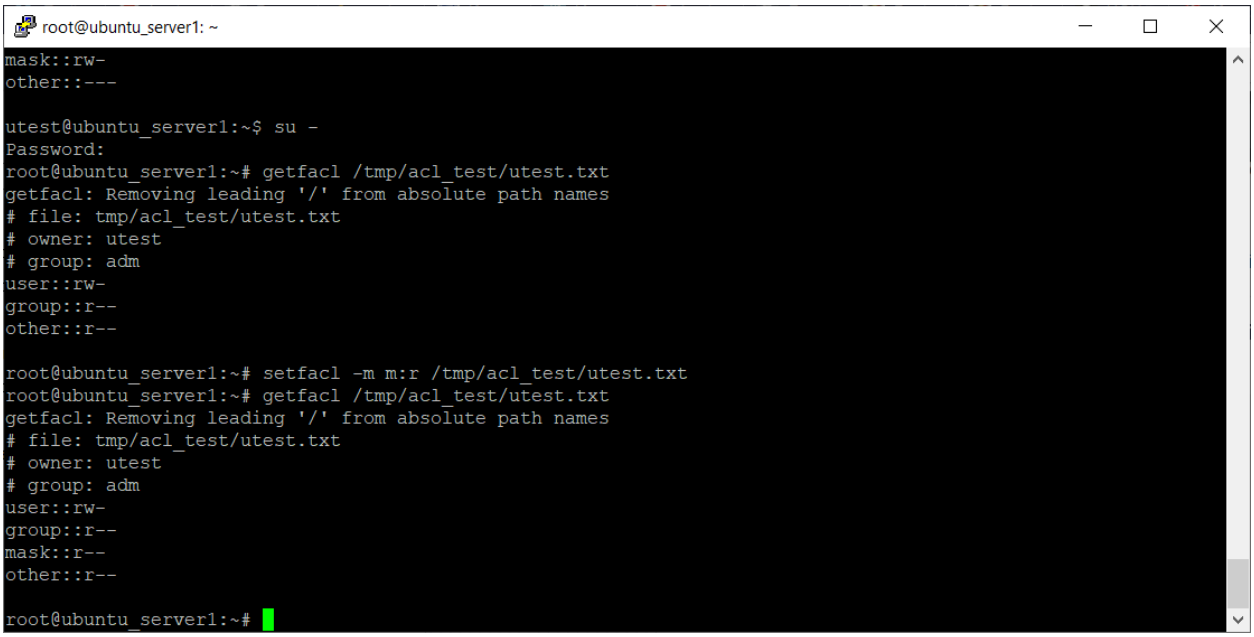
setfacl -m u:utest:rwx /tmp/acl_test



5. For user *utest*, set default ACLs to the directory */tmp/acl_test* which allow read-only access (hint: use the -d option of the **setfacl** command). Being logged in as *utest*, invoke **touch** to create the file *utest2.txt* in the */tmp/acl_test* directory. Query permissions on this file using **getfacl**.

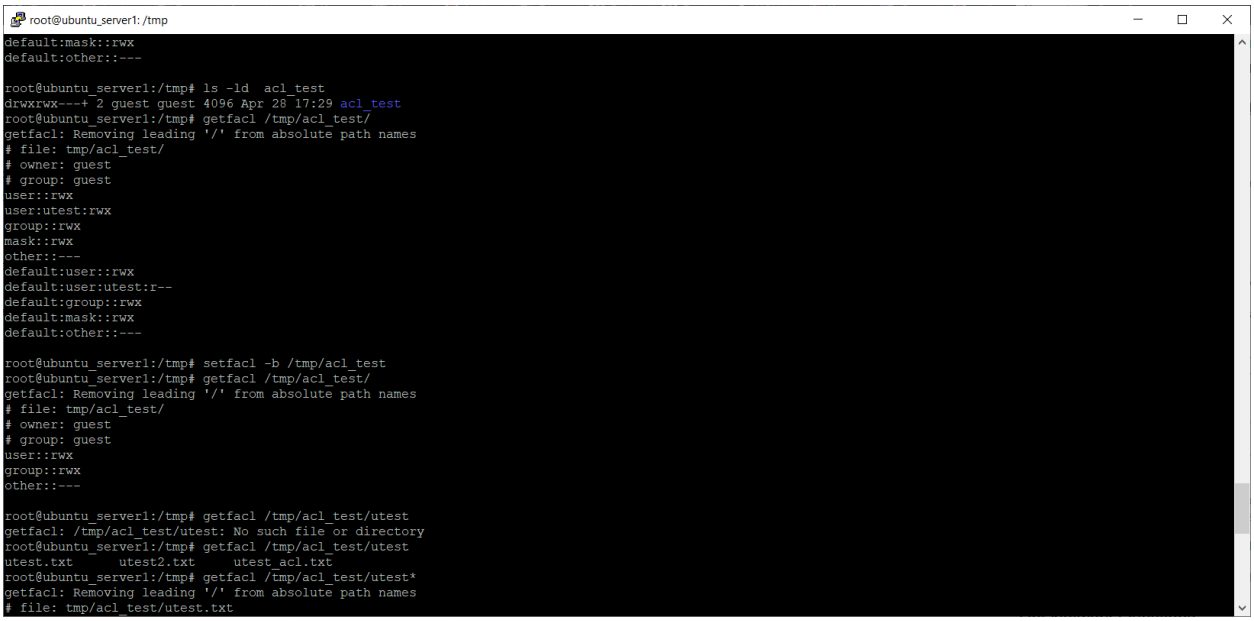


6. Set the maximum permissions mask on the */tmp/acl_test/utest.txt* file in such a way as to allow read-only access. Check permissions with **getfacl**.



7. Delete all ACL entries relative to the */tmp/acl_test* directory.

For all entries - setfacl -b /tmp/acl_test*



And check that no + in perm column

root@ubuntu_server1:/tmp# ls -ld acl_test

drwxrwx--- 2 guest guest 4096 Apr 28 17:29 acl_test