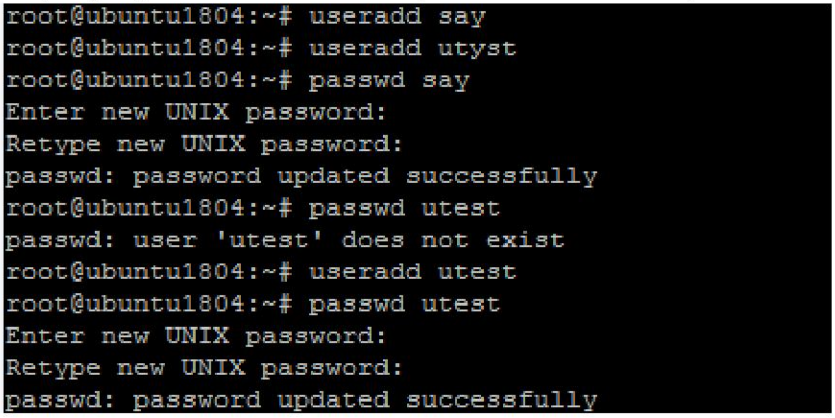
EPAM University Programs
DevOps external course
Module 4 Linux & Bash Essentials
TASK 4.7

Part1. **Quota allocation mechanism.**

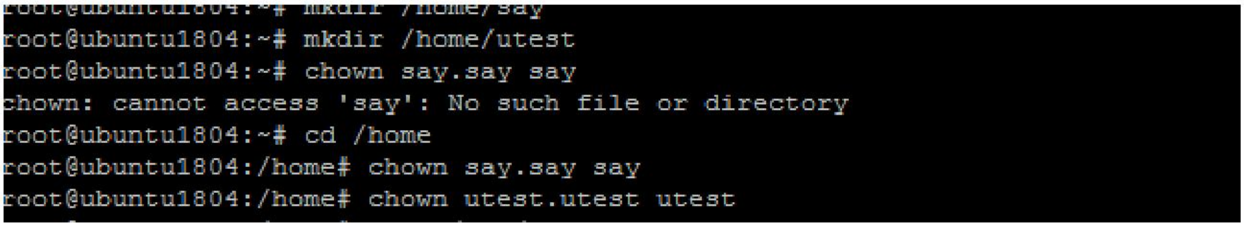
Employing commands from presentation #4.6, create a new user, say, *utest*.



```
root@ubuntu1804:~# useradd say
root@ubuntu1804:~# useradd utyst
root@ubuntu1804:~# passwd say
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@ubuntu1804:~# passwd utest
passwd: user 'utest' does not exist
root@ubuntu1804:~# useradd utest
root@ubuntu1804:~# passwd utest
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

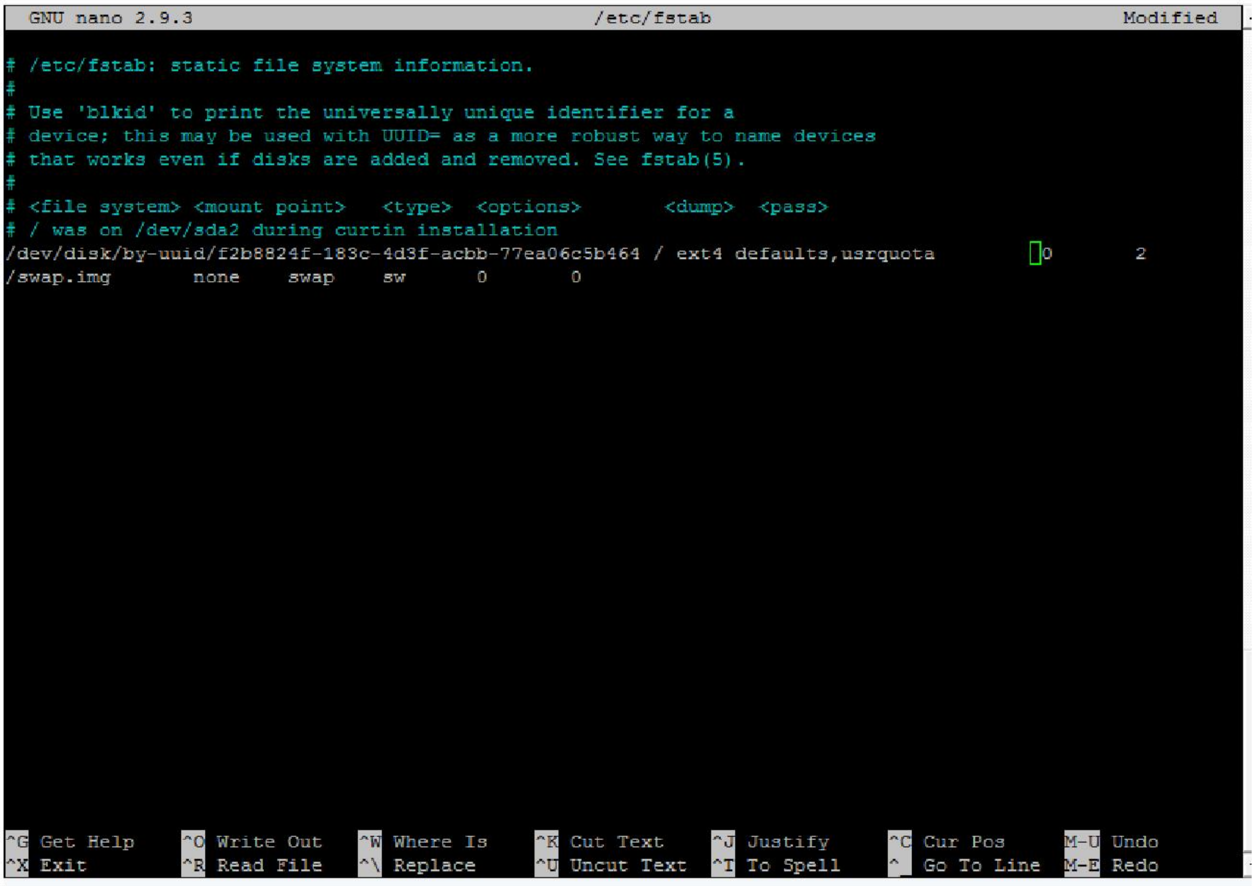
oops, we forgot create homedirs

Let’s create it



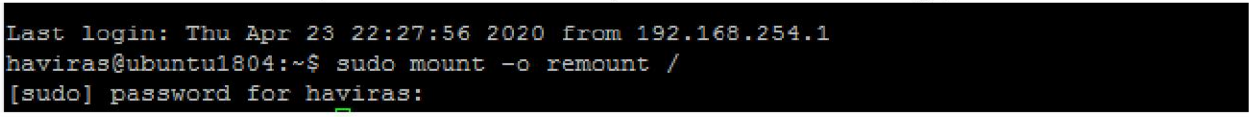
```
root@ubuntu1804:~# mkdir /home/say
root@ubuntu1804:~# mkdir /home/utest
root@ubuntu1804:~# chown say.say say
chown: cannot access 'say': No such file or directory
root@ubuntu1804:~# cd /home
root@ubuntu1804:/home# chown say.say say
root@ubuntu1804:/home# chown utest.utest utest
```

editing fstab



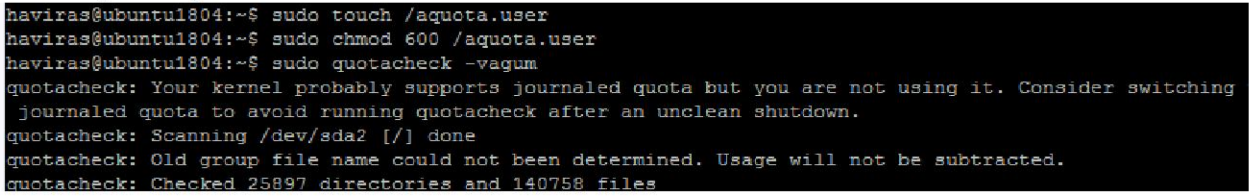
```
  GNU nano 2.9.3                          /etc/fstab                          Modified

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/f2b8824f-183c-4d3f-acbb-77ea06c5b464 / ext4 defaults,usrquota        0       2
/swap.img       none    swap    sw      0       0
```

Because I don’t have /home as a mount point I’m remounting /



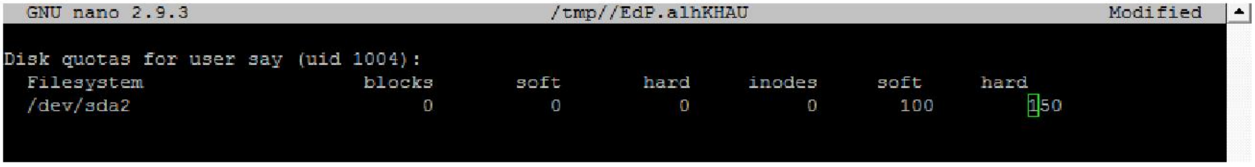
```
Last login: Thu Apr 23 22:27:56 2020 from 192.168.254.1
haviras@ubuntu1804:~$ sudo mount -o remount /
[sudo] password for haviras:
```

Making of aquta.user file



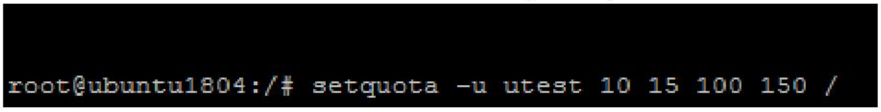
```
haviras@ubuntu1804:~$ sudo touch /aquota.user
haviras@ubuntu1804:~$ sudo chmod 600 /aquota.user
haviras@ubuntu1804:~$ sudo quotacheck -vagum
quotacheck: Your kernel probably supports journaled quota but you are not using it. Consider switching
 journaled quota to avoid running quotacheck after an unclean shutdown.
quotacheck: Scanning /dev/sda2 [/] done
quotacheck: Old group file name could not been determined. Usage will not be subtracted.
quotacheck: Checked 25897 directories and 140758 files
```

for user say



```
  GNU nano 2.9.3                     /tmp//EdP.alhKHAU                     Modified

Disk quotas for user say (uid 1004):
  Filesystem                   blocks       soft       hard     inodes     soft     hard
  /dev/sda2                         0          0          0          0      100      150
```

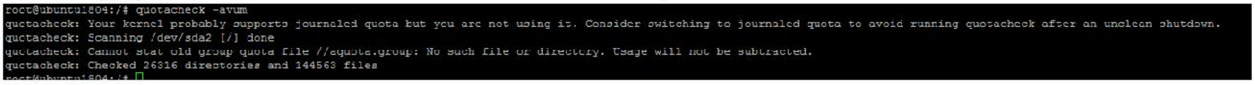
for user utest - also we can set it by setquota



```
root@ubuntu1804:/# setquota -u utest 10 15 100 150 /
```

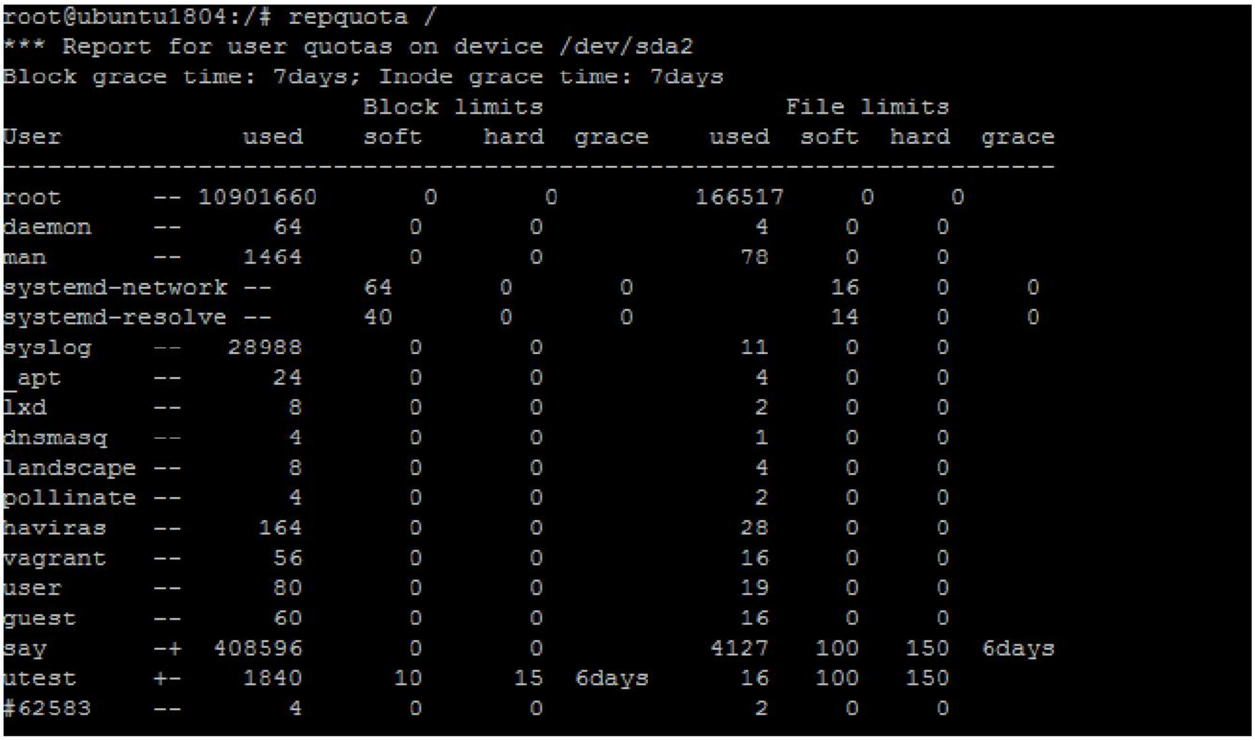
Based on the quota mechanism, limit the available disk space for this user to **soft**: 100M and **hard**: 150M.

making quotacheck –avum for recalculate qoutes



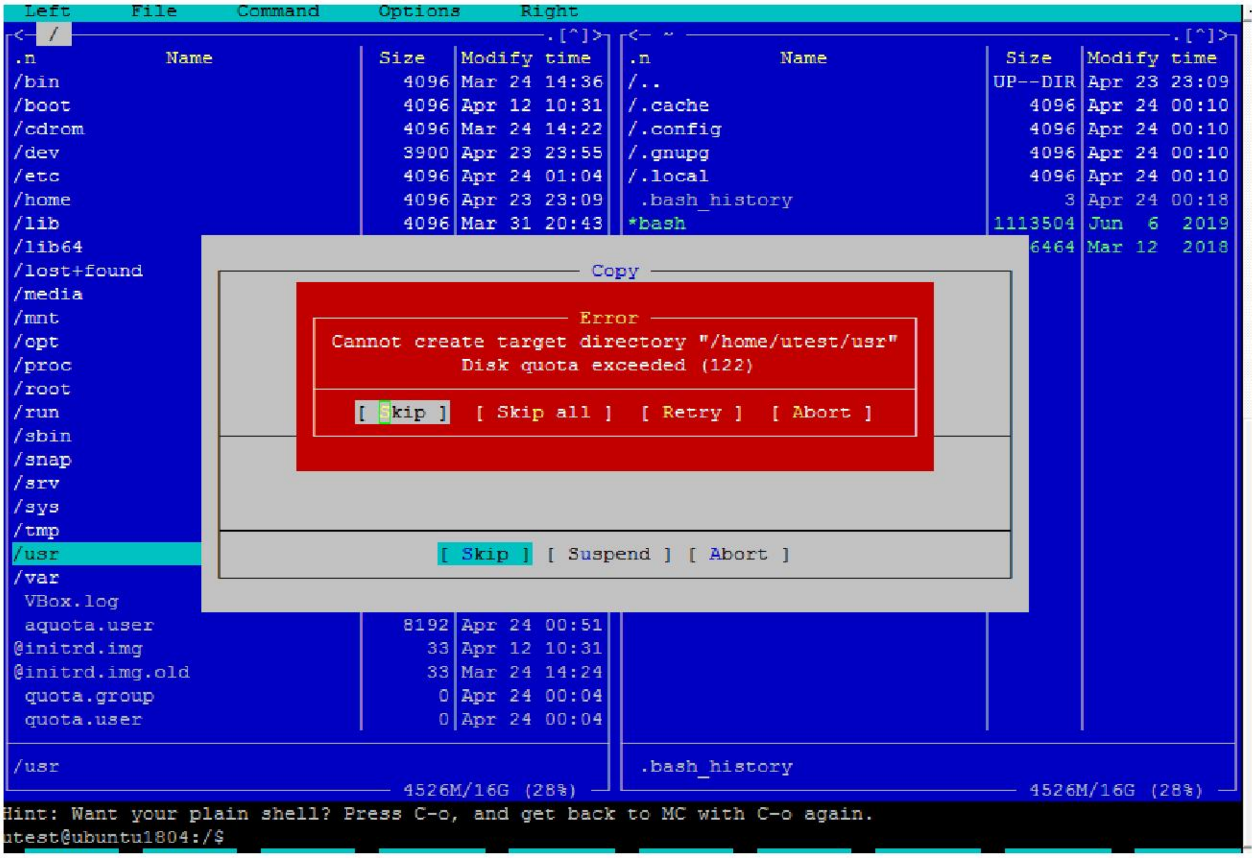
```
root@ubuntu1804:/# quotacheck -avum
quotacheck: Your kernel probably supports journaled quota but you are not using it. Consider switching to journaled quota to avoid running quotacheck after an unclean shutdown.
quotacheck: Scanning /dev/sda2 [/] done
quotacheck: Cannot stat old group quota file //aquota.group: No such file or directory. Usage will not be subtracted.
quotacheck: Checked 26316 directories and 144563 files
root@ubuntu1804:/# 
```

check it



```
root@ubuntu1804:/# repquota /
*** Report for user quotas on device /dev/sda2
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      -- 10901660       0       0         166517     0     0
daemon    --      64       0       0               4     0     0
man       --    1464       0       0              78     0     0
systemd-network --      64       0       0              16     0     0
systemd-resolve --      40       0       0              14     0     0
syslog    --   28988       0       0              11     0     0
_apt      --      24       0       0               4     0     0
lxd       --       8       0       0               2     0     0
dnsmasq   --       4       0       0               1     0     0
landscape --       8       0       0               4     0     0
pollinate --       4       0       0               2     0     0
haviras   --     164       0       0              28     0     0
vagrant   --      56       0       0              16     0     0
user      --      80       0       0              19     0     0
guest     --      60       0       0              16     0     0
say       -+  408596       0       0            4127   100   150  6days
utest     +-    1840      10      15  6days      16   100   150
#62583    --       4       0       0               2     0     0
```

and we have…. Error!



```
 Left       File       Command       Options       Right
<- /                           .[^]>  <- ~                                .[^]>
.n             Name        Size  Modify time  .n             Name        Size  Modify time
/bin                       4096 Mar 24 14:36  /..                      UP--DIR Apr 23 23:09
/boot                      4096 Apr 12 10:31  /.cache                     4096 Apr 24 00:10
/cdrom                     4096 Mar 24 14:22  /.config                    4096 Apr 24 00:10
/dev                       3900 Apr 23 23:55  /.gnupg                     4096 Apr 24 00:10
/etc                       4096 Apr 24 01:04  /.local                     4096 Apr 24 00:10
/home                      4096 Apr 23 23:09   .bash_history                 3 Apr 24 00:18
/lib                       4096 Mar 31 20:43  *bash                    1113504 Jun  6  2019
/lib64                                                                   6464 Mar 12  2018
/lost+found                    Copy
/media                            Error
/mnt            Cannot create target directory "/home/utest/usr"
/opt                      Disk quota exceeded (122)
/proc
/root              [ Skip ]  [ Skip all ]  [ Retry ]  [ Abort ]
/run
/sbin
/snap
/srv
/sys
/tmp
/usr                          [ Skip ] [ Suspend ] [ Abort ]
/var
 VBox.log
 aquota.user               8192 Apr 24 00:51
@initrd.img                  33 Apr 12 10:31
@initrd.img.old              33 Mar 24 14:24
 quota.group                  0 Apr 24 00:04
 quota.user                   0 Apr 24 00:04

/usr                                          .bash_history
                      4526M/16G (28%)                            4526M/16G (28%)
Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.
utest@ubuntu1804:/$
```

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).

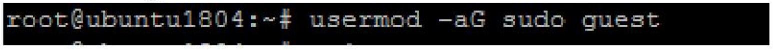
Note: if /home is not a mount point, then the **mount** and **quotaon** commands should be called with respect to the root partition /.

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).

Part2. **Access Control Lists, ACLs**

In what follows, we assume that there are two users: *guest* (included into the list of sudoers) and *utest*. None of the users is the superuser (i.e. UIDs of the users differ from 0).

Add user guest to sudoers



```
root@ubuntu1804:~# usermod -aG sudo guest
```

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).

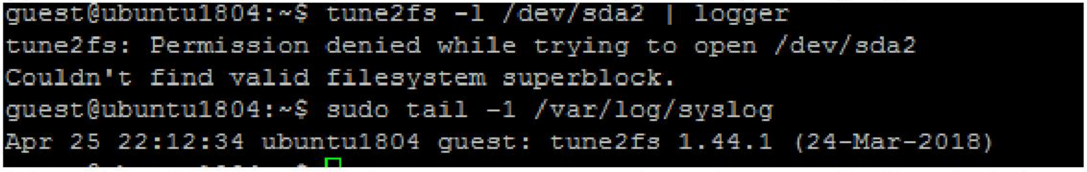
1. Based on given in presentation #4.7 instructions, turn on and set up the ACL. *Caution*! The fact that a file system has been mounted with the “acl” flag on by default, doesn’t mean that the ACL package is installed.

Prior to any action, it is advised to check if the “acl” flag is on, using

tune2fs -l /dev/sda*

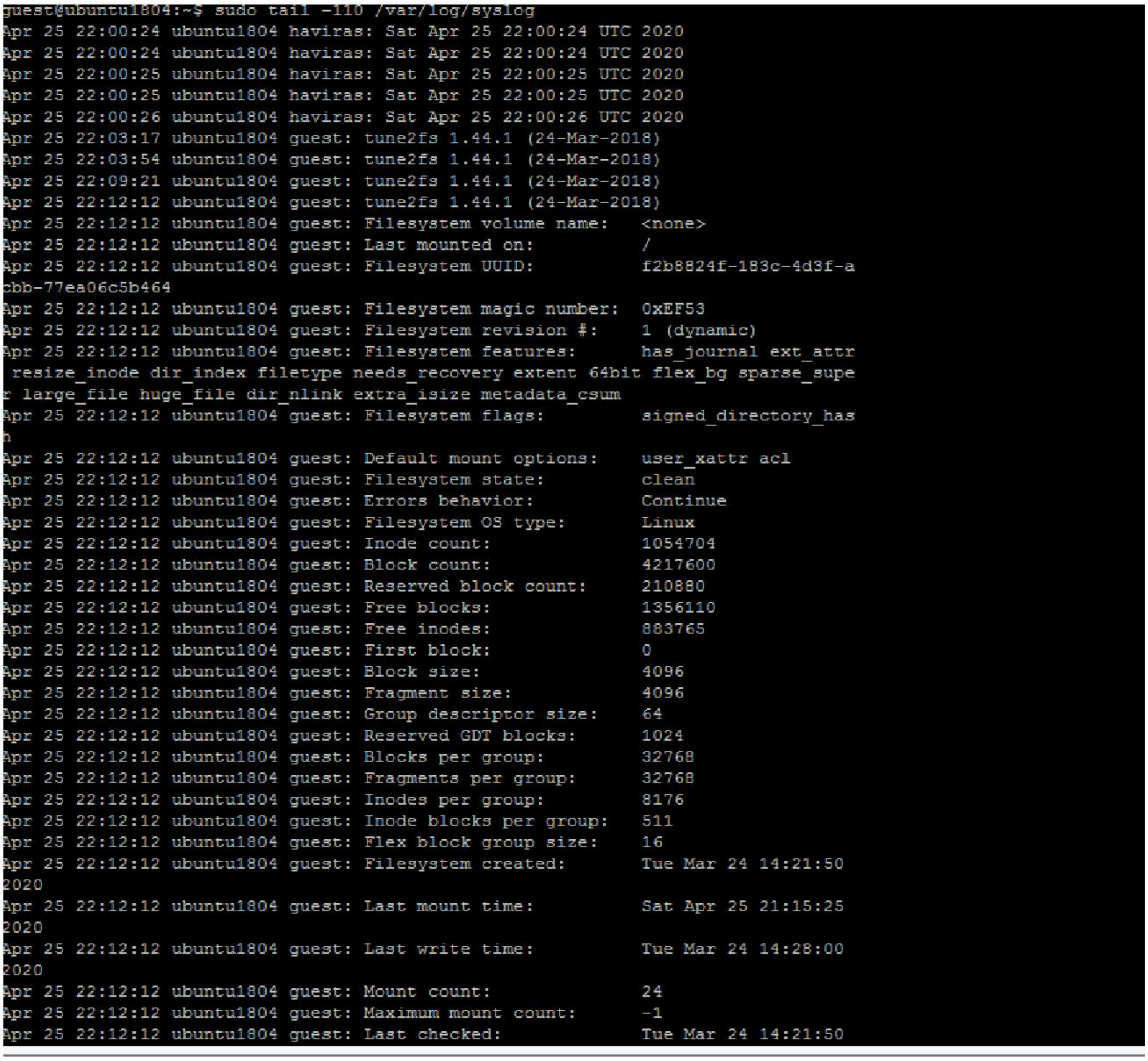
(a particular name of the device file sda*, is to be determined by calling to **blkid**, invoke it twice:

(i) on behalf of *guest* (i.e. without the superuser privileges);



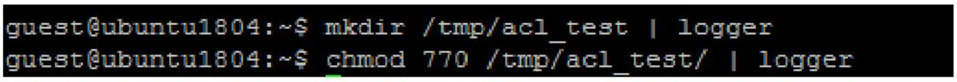
```
guest@ubuntu1804:~$ tune2fs -l /dev/sda2 | logger
tune2fs: Permission denied while trying to open /dev/sda2
Couldn't find valid filesystem superblock.
guest@ubuntu1804:~$ sudo tail -1 /var/log/syslog
Apr 25 22:12:34 ubuntu1804 guest: tune2fs 1.44.1 (24-Mar-2018)
```

(ii) with **sudo** (i.e. with the superuser privileges). Note the level of details provided by different **blkid** outputs).



```
guest@ubuntu1804:~$ sudo tail -110 /var/log/syslog
Apr 25 22:00:24 ubuntu1804 haviras: Sat Apr 25 22:00:24 UTC 2020
Apr 25 22:00:24 ubuntu1804 haviras: Sat Apr 25 22:00:24 UTC 2020
Apr 25 22:00:25 ubuntu1804 haviras: Sat Apr 25 22:00:25 UTC 2020
Apr 25 22:00:25 ubuntu1804 haviras: Sat Apr 25 22:00:25 UTC 2020
Apr 25 22:00:26 ubuntu1804 haviras: Sat Apr 25 22:00:26 UTC 2020
Apr 25 22:03:17 ubuntu1804 guest: tune2fs 1.44.1 (24-Mar-2018)
Apr 25 22:03:54 ubuntu1804 guest: tune2fs 1.44.1 (24-Mar-2018)
Apr 25 22:09:21 ubuntu1804 guest: tune2fs 1.44.1 (24-Mar-2018)
Apr 25 22:12:12 ubuntu1804 guest: tune2fs 1.44.1 (24-Mar-2018)
Apr 25 22:12:12 ubuntu1804 guest: Filesystem volume name:   <none>
Apr 25 22:12:12 ubuntu1804 guest: Last mounted on:          /
Apr 25 22:12:12 ubuntu1804 guest: Filesystem UUID:          f2b8824f-183c-4d3f-a
cbb-77ea06c5b464
Apr 25 22:12:12 ubuntu1804 guest: Filesystem magic number:  0xEF53
Apr 25 22:12:12 ubuntu1804 guest: Filesystem revision #:    1 (dynamic)
Apr 25 22:12:12 ubuntu1804 guest: Filesystem features:      has_journal ext_attr
 resize_inode dir_index filetype needs_recovery extent 64bit flex_bg sparse_supe
r large_file huge_file dir_nlink extra_isize metadata_csum
Apr 25 22:12:12 ubuntu1804 guest: Filesystem flags:         signed_directory_has
h
Apr 25 22:12:12 ubuntu1804 guest: Default mount options:    user_xattr acl
Apr 25 22:12:12 ubuntu1804 guest: Filesystem state:         clean
Apr 25 22:12:12 ubuntu1804 guest: Errors behavior:          Continue
Apr 25 22:12:12 ubuntu1804 guest: Filesystem OS type:       Linux
Apr 25 22:12:12 ubuntu1804 guest: Inode count:              1054704
Apr 25 22:12:12 ubuntu1804 guest: Block count:              4217600
Apr 25 22:12:12 ubuntu1804 guest: Reserved block count:     210880
Apr 25 22:12:12 ubuntu1804 guest: Free blocks:              1356110
Apr 25 22:12:12 ubuntu1804 guest: Free inodes:              883765
Apr 25 22:12:12 ubuntu1804 guest: First block:              0
Apr 25 22:12:12 ubuntu1804 guest: Block size:               4096
Apr 25 22:12:12 ubuntu1804 guest: Fragment size:            4096
Apr 25 22:12:12 ubuntu1804 guest: Group descriptor size:    64
Apr 25 22:12:12 ubuntu1804 guest: Reserved GDT blocks:      1024
Apr 25 22:12:12 ubuntu1804 guest: Blocks per group:         32768
Apr 25 22:12:12 ubuntu1804 guest: Fragments per group:      32768
Apr 25 22:12:12 ubuntu1804 guest: Inodes per group:         8176
Apr 25 22:12:12 ubuntu1804 guest: Inode blocks per group:   511
Apr 25 22:12:12 ubuntu1804 guest: Flex block group size:    16
Apr 25 22:12:12 ubuntu1804 guest: Filesystem created:       Tue Mar 24 14:21:50
2020
Apr 25 22:12:12 ubuntu1804 guest: Last mount time:          Sat Apr 25 21:15:25
2020
Apr 25 22:12:12 ubuntu1804 guest: Last write time:          Tue Mar 24 14:28:00
2020
Apr 25 22:12:12 ubuntu1804 guest: Mount count:              24
Apr 25 22:12:12 ubuntu1804 guest: Maximum mount count:      -1
Apr 25 22:12:12 ubuntu1804 guest: Last checked:             Tue Mar 24 14:21:50
```

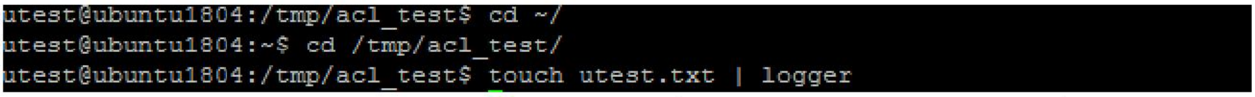
2. Log in as *guest*. Create in */tmp* a directory called *acl_test*. By means of **chmod,** allow user utest to perform all possible operations (rwx) with respect to *acl_test.*



```
guest@ubuntu1804:~$ mkdir /tmp/acl_test | logger
guest@ubuntu1804:~$ chmod 770 /tmp/acl_test/ | logger
```

But we don't have output in this commands (because everything is OK)!

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**.



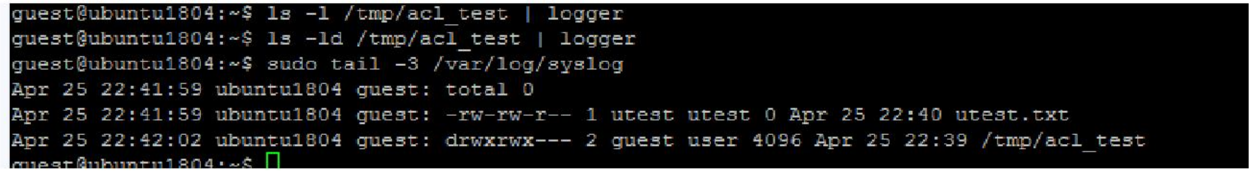
```
utest@ubuntu1804:/tmp/acl_test$ cd ~/
utest@ubuntu1804:~$ cd /tmp/acl_test/
utest@ubuntu1804:/tmp/acl_test$ touch utest.txt | logger
```

But we don't have output in this commands (because everything is OK)!

Query information about the directory and file by calling to

ls -ld /tmp/acl_test

ls -l /tmp/acl_test

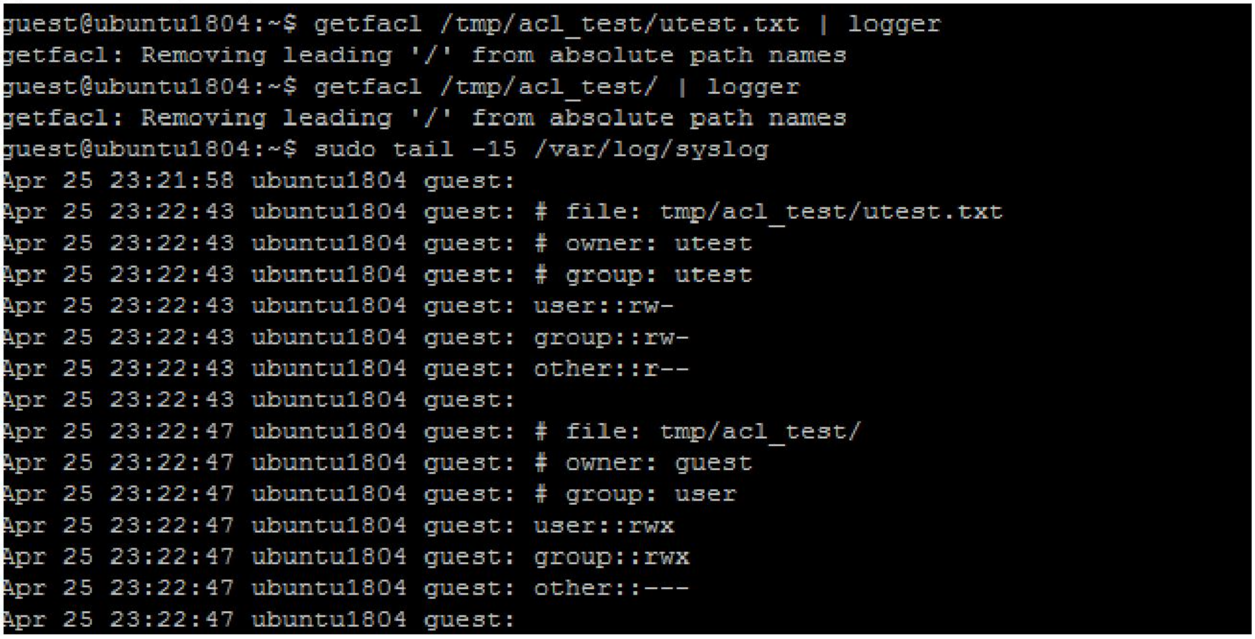


```
guest@ubuntu1804:~$ ls -l /tmp/acl_test | logger
guest@ubuntu1804:~$ ls -ld /tmp/acl_test | logger
guest@ubuntu1804:~$ sudo tail -3 /var/log/syslog
Apr 25 22:41:59 ubuntu1804 guest: total 0
Apr 25 22:41:59 ubuntu1804 guest: -rw-rw-r-- 1 utest utest 0 Apr 25 22:40 utest.txt
Apr 25 22:42:02 ubuntu1804 guest: drwxrwx--- 2 guest user 4096 Apr 25 22:39 /tmp/acl_test
guest@ubuntu1804:~$
```

To check ACL permissions do:

getfacl /tmp/acl_test

getfacl /tmp/acl_test/utest.txt

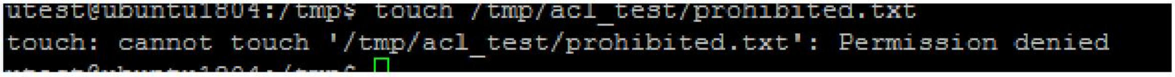


```
guest@ubuntu1804:~$ getfacl /tmp/acl_test/utest.txt | logger
getfacl: Removing leading '/' from absolute path names
guest@ubuntu1804:~$ getfacl /tmp/acl_test/ | logger
getfacl: Removing leading '/' from absolute path names
guest@ubuntu1804:~$ sudo tail -15 /var/log/syslog
Apr 25 23:21:58 ubuntu1804 guest:
Apr 25 23:22:43 ubuntu1804 guest: # file: tmp/acl_test/utest.txt
Apr 25 23:22:43 ubuntu1804 guest: # owner: utest
Apr 25 23:22:43 ubuntu1804 guest: # group: utest
Apr 25 23:22:43 ubuntu1804 guest: user::rw-
Apr 25 23:22:43 ubuntu1804 guest: group::rw-
Apr 25 23:22:43 ubuntu1804 guest: other::r--
Apr 25 23:22:43 ubuntu1804 guest:
Apr 25 23:22:47 ubuntu1804 guest: # file: tmp/acl_test/
Apr 25 23:22:47 ubuntu1804 guest: # owner: guest
Apr 25 23:22:47 ubuntu1804 guest: # group: user
Apr 25 23:22:47 ubuntu1804 guest: user::rwx
Apr 25 23:22:47 ubuntu1804 guest: group::rwx
Apr 25 23:22:47 ubuntu1804 guest: other::---
Apr 25 23:22:47 ubuntu1804 guest:
```

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

touch /tmp/acl_test/prohibited.txt

Is it possible to invoke this command?

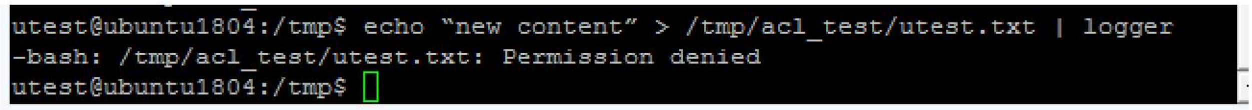


```
utest@ubuntu1804:/tmp$ touch /tmp/acl_test/prohibited.txt
touch: cannot touch '/tmp/acl_test/prohibited.txt': Permission denied
utest@ubuntu1804:/tmp$
```

oops, we can't invoke this command, because we don't have access to folder

echo "new content" > /tmp/acl_test/utest.txt

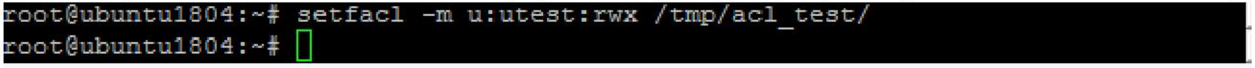
Test if user *utest* can be prevented from modifying content of the file *utest.txt* by means of ACL. (Note that user *utest* is the owner of the file *tmp/acl_test/utest.txt*).



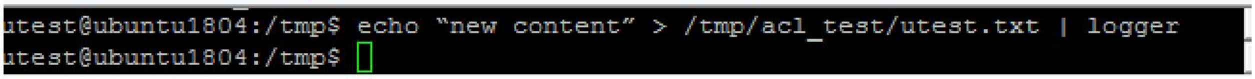
```
utest@ubuntu1804:/tmp$ echo "new content" > /tmp/acl_test/utest.txt | logger
-bash: /tmp/acl_test/utest.txt: Permission denied
utest@ubuntu1804:/tmp$
```

Yes, access denied

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 ac=on is allowed with **chmod** (conventional mechanism). (Hint: repeat step 3, but given the new context).



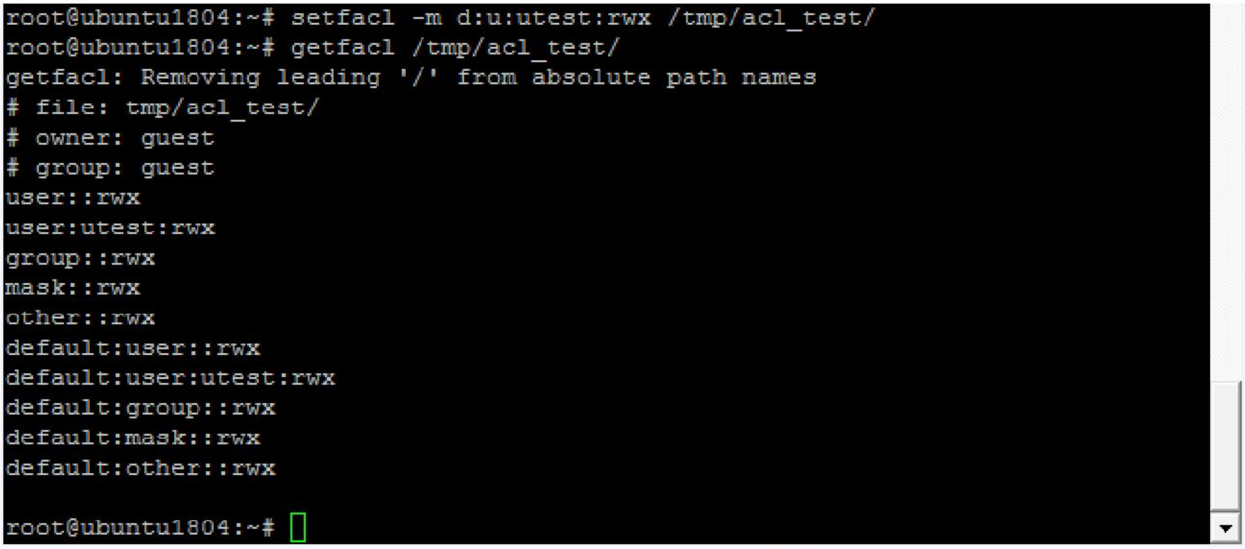
```
root@ubuntu1804:~# setfacl -m u:utest:rwx /tmp/acl_test/
root@ubuntu1804:~#
```



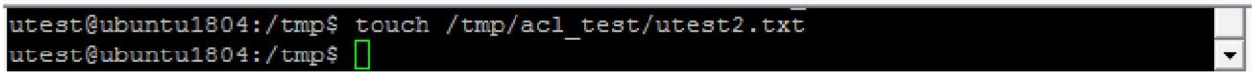
```
utest@ubuntu1804:/tmp$ echo "new content" > /tmp/acl_test/utest.txt | logger
utest@ubuntu1804:/tmp$
```

Wow! It works!

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**.

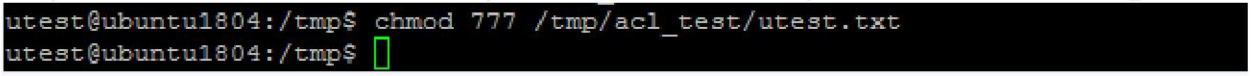


```
root@ubuntu1804:~# setfacl -m d:u:utest:rwx /tmp/acl_test/
root@ubuntu1804:~# getfacl /tmp/acl_test/
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/
# owner: guest
# group: guest
user::rwx
user:utest:rwx
group::rwx
mask::rwx
other::rwx
default:user::rwx
default:user:utest:rwx
default:group::rwx
default:mask::rwx
default:other::rwx

root@ubuntu1804:~#
```

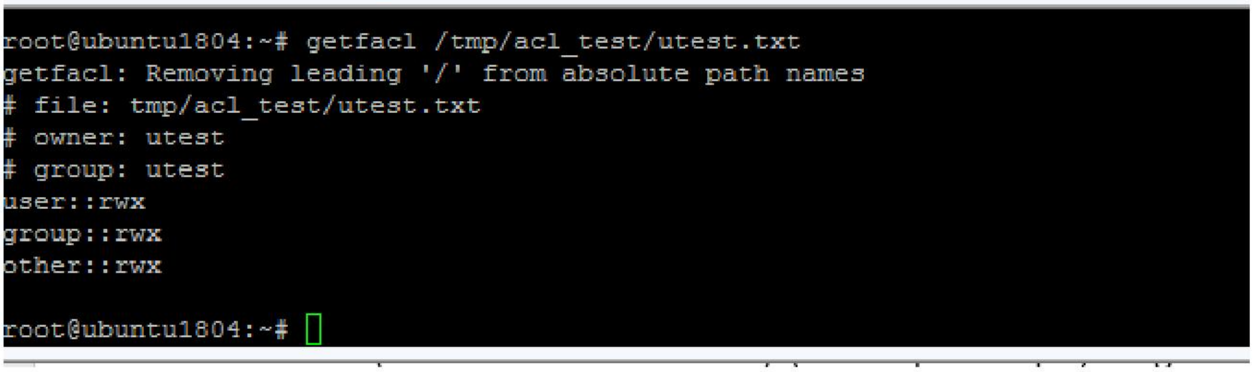


```
utest@ubuntu1804:/tmp$ touch /tmp/acl_test/utest2.txt
utest@ubuntu1804:/tmp$
```

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**.

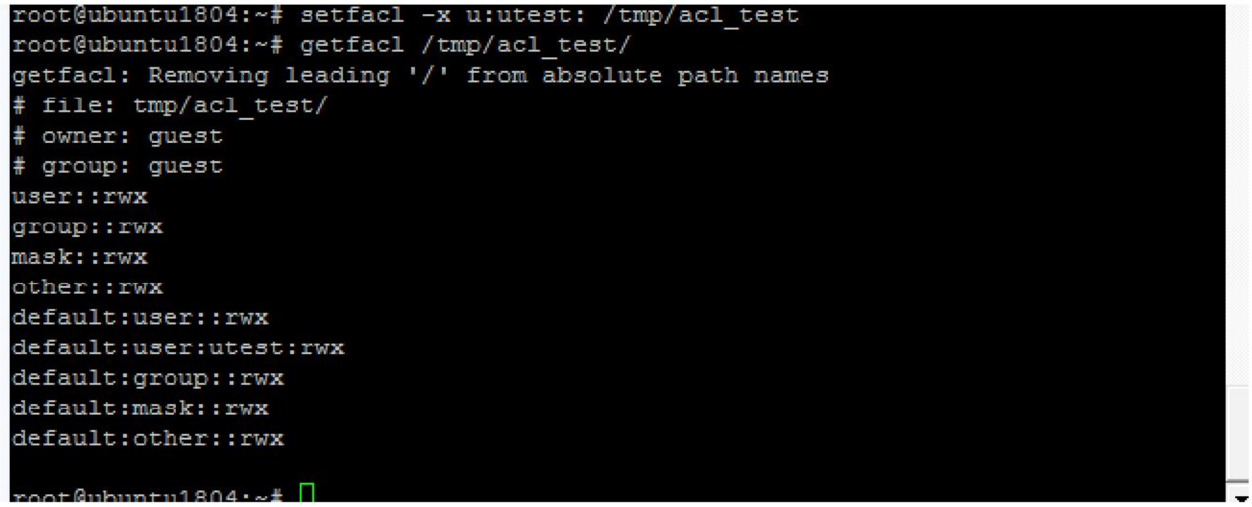


```
utest@ubuntu1804:/tmp$ chmod 777 /tmp/acl_test/utest.txt
utest@ubuntu1804:/tmp$ 
```



```
root@ubuntu1804:~# getfacl /tmp/acl_test/utest.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/utest.txt
# owner: utest
# group: utest
user::rwx
group::rwx
other::rwx

root@ubuntu1804:~# 
```

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



```
root@ubuntu1804:~# setfacl -x u:utest: /tmp/acl_test
root@ubuntu1804:~# getfacl /tmp/acl_test/
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/
# owner: guest
# group: guest
user::rwx
group::rwx
mask::rwx
other::rwx
default:user::rwx
default:user:utest:rwx
default:group::rwx
default:mask::rwx
default:other::rwx

root@ubuntu1804:~# 
```