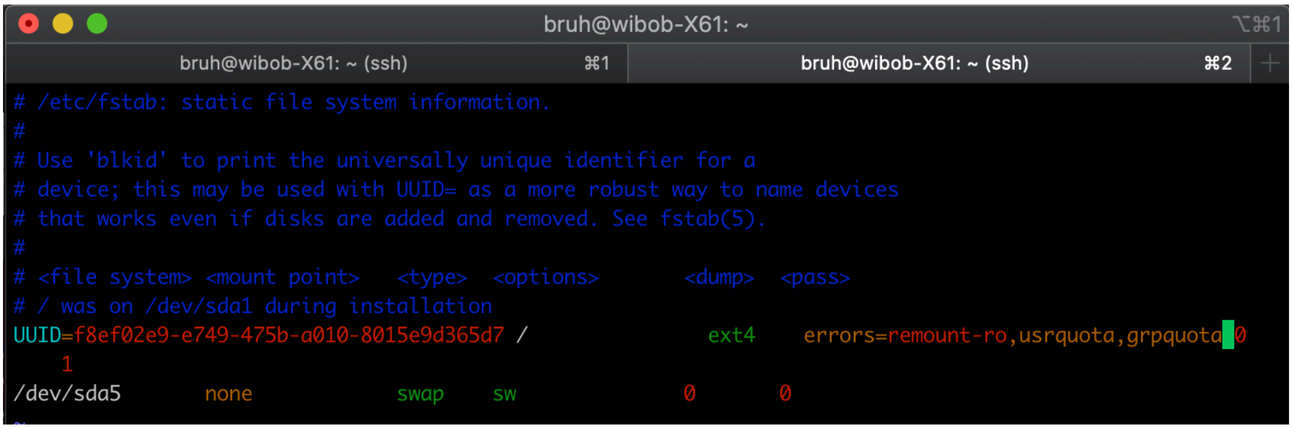
Module 4 Linux & Bash Essentials (Task 4.7)

Part1. **Quota allocation mechanism.**

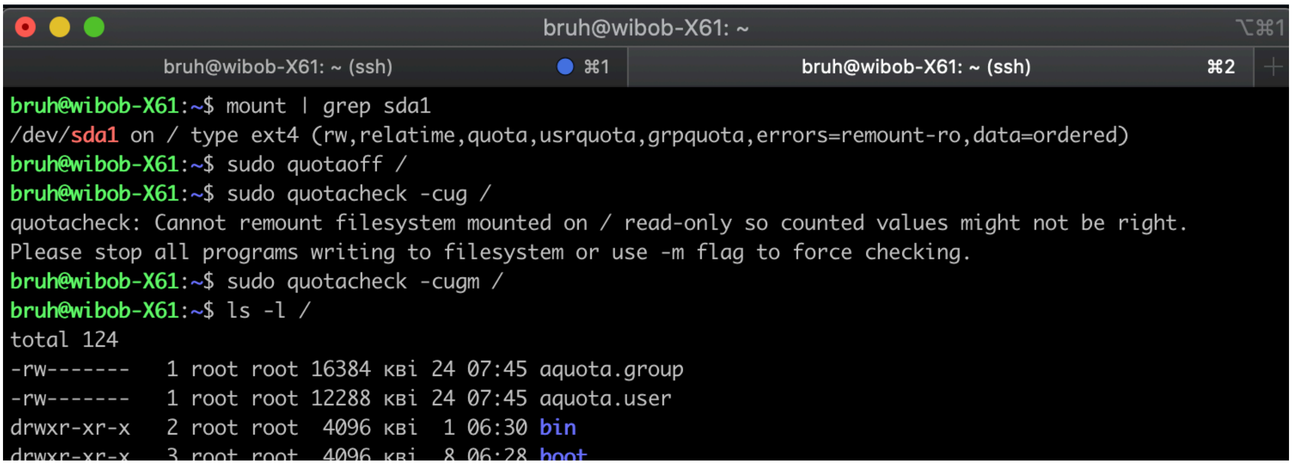
Employing commands from presentation #4.6, create a new user, say, *utest*. Based on the quota mechanism, limit the available disk space for this user to **soft**: 100M and **hard**: 150M.
Then, using Midnight Commander (since MC shows warnings about exceeding the limits of available to a user disk space), 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).

## modifying fstab to enable quotas



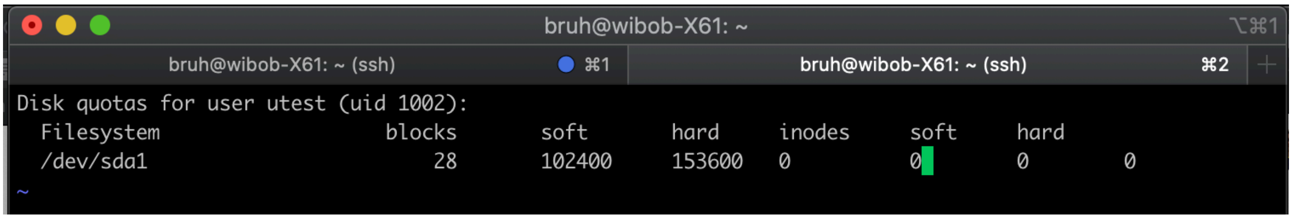
```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=f8ef02e9-e749-475b-a010-8015e9d365d7 /               ext4    errors=remount-ro,usrquota,grpquota 0
    1
/dev/sda5       none            swap    sw              0       0
```

## remount partition with $ sudo mount -n -o remount,defaults /dev/sda1
## check mount options for sda1
## disable quota
## check and creating quota-data files
## checking wether files were created (aquota.group, aquota.user)



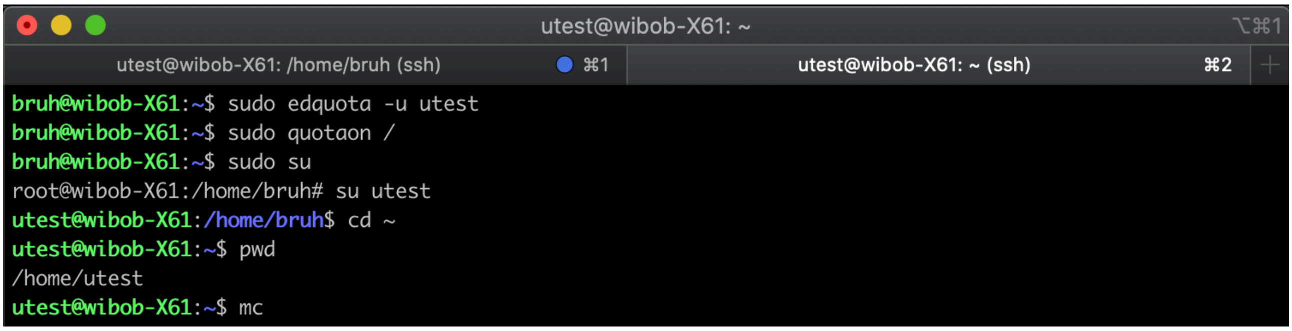
```
bruh@wibob-X61:~$ mount | grep sda1
/dev/sda1 on / type ext4 (rw,relatime,quota,usrquota,grpquota,errors=remount-ro,data=ordered)
bruh@wibob-X61:~$ sudo quotaoff /
bruh@wibob-X61:~$ sudo quotacheck -cug /
quotacheck: Cannot remount filesystem mounted on / read-only so counted values might not be right.
Please stop all programs writing to filesystem or use -m flag to force checking.
bruh@wibob-X61:~$ sudo quotacheck -cugm /
bruh@wibob-X61:~$ ls -l /
total 124
-rw-------  1 root root 16384 кві 24 07:45 aquota.group
-rw-------  1 root root 12288 кві 24 07:45 aquota.user
drwxr-xr-x  2 root root  4096 кві  1 06:30 bin
drwxr-xr-x  3 root root  4096 кві  8 06:28 boot
```

##running edquota command to edit quota for user utest



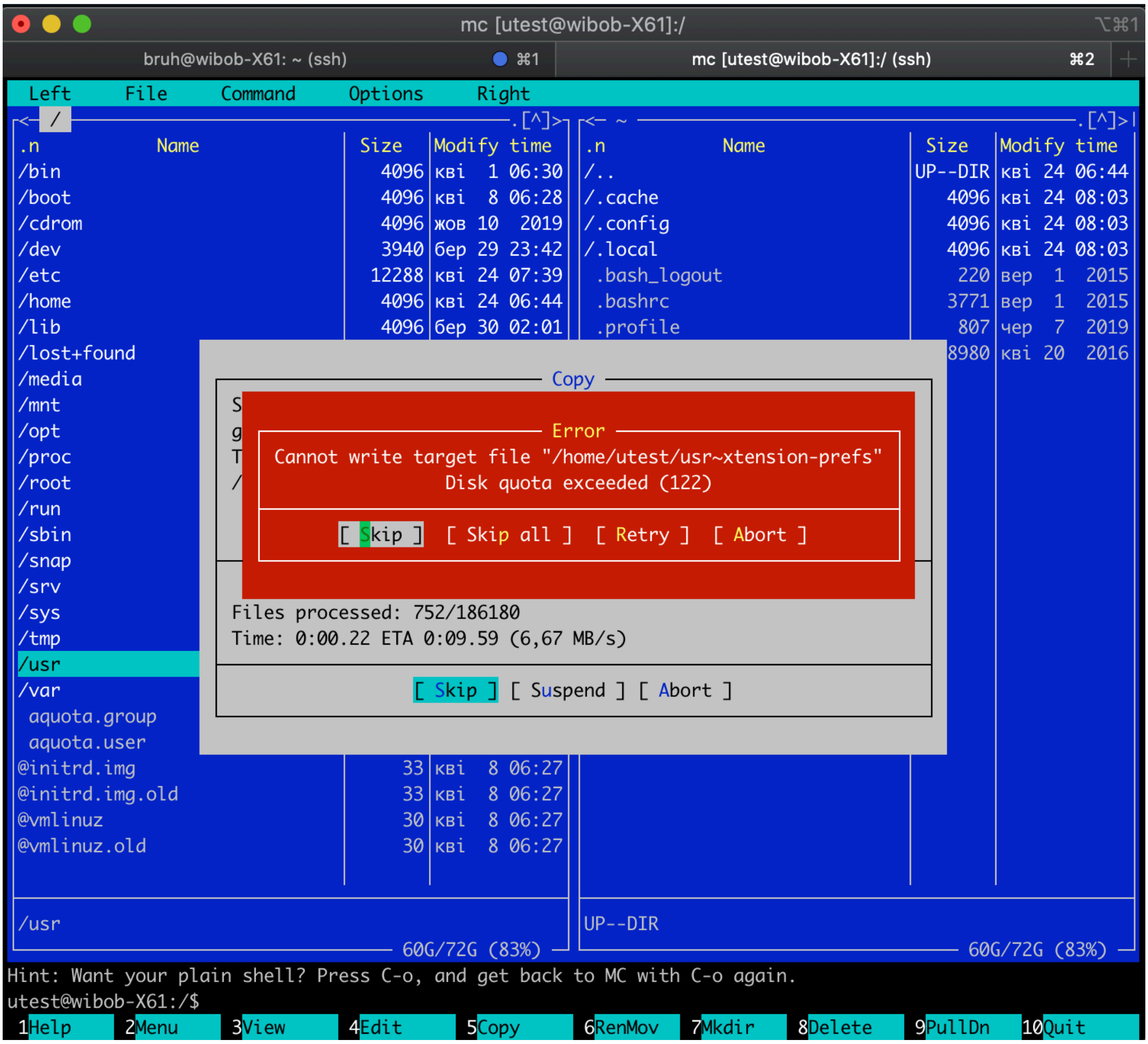
```
Disk quotas for user utest (uid 1002):
  Filesystem                   blocks       soft       hard     inodes     soft     hard
  /dev/sda1                        28     102400     153600     0          0        0        0
~
```

## enabling quota
## switch user for username utest



```
bruh@wibob-X61:~$ sudo edquota -u utest
bruh@wibob-X61:~$ sudo quotaon /
bruh@wibob-X61:~$ sudo su
root@wibob-X61:/home/bruh# su utest
utest@wibob-X61:/home/bruh$ cd ~
utest@wibob-X61:~$ pwd
/home/utest
utest@wibob-X61:~$ mc
```

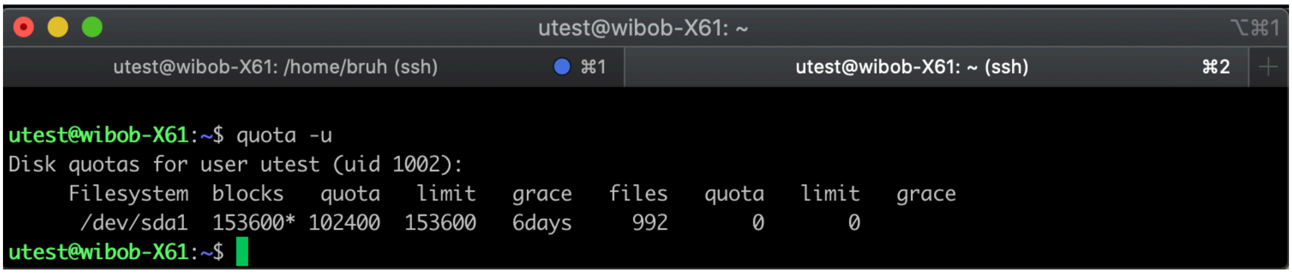
## exceeding disk quota



```
<- /                              .[^]>  <- ~                              .[^]>
.n          Name          Size |Modify time   .n          Name          Size |Modify time
/bin                      4096 |кві  1 06:30  /..                     UP--DIR|кві 24 06:44
/boot                     4096 |кві  8 06:28  /.cache                    4096|кві 24 08:03
/cdrom                    4096 |жов 10  2019  /.config                   4096|кві 24 08:03
/dev                      3940 |бер 29 23:42  /.local                    4096|кві 24 08:03
/etc                     12288 |кві 24 07:39   .bash_logout               220|вер  1  2015
/home                     4096 |кві 24 06:44   .bashrc                   3771|вер  1  2015
/lib                      4096 |бер 30 02:01   .profile                   807|чер  7  2019
/lost+found                                                              8980|кві 20  2016
/media
/mnt                          ─── Copy ───
/opt                S
/proc               g     ─── Error ───
/root               T     Cannot write target file "/home/utest/usr~xtension-prefs"
/run                /                  Disk quota exceeded (122)
/sbin
/snap                     [ Skip ]  [ Skip all ]  [ Retry ]  [ Abort ]
/srv
/sys
/tmp                Files processed: 752/186180
/usr                Time: 0:00.22 ETA 0:09.59 (6,67 MB/s)
/var
 aquota.group              [ Skip ] [ Suspend ] [ Abort ]
 aquota.user
@initrd.img                 33 |кві  8 06:27
@initrd.img.old             33 |кві  8 06:27
@vmlinuz                    30 |кві  8 06:27
@vmlinuz.old                30 |кві  8 06:27

/usr                                          UP--DIR
                          60G/72G (83%)                               60G/72G (83%)
Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.
utest@wibob-X61:/$
```

## checking status of user quota



```
utest@wibob-X61:~$ quota -u
Disk quotas for user utest (uid 1002):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/sda1  153600* 102400  153600   6days     992       0       0
utest@wibob-X61:~$
```

Part2. **Access Control Lists, ACLs**

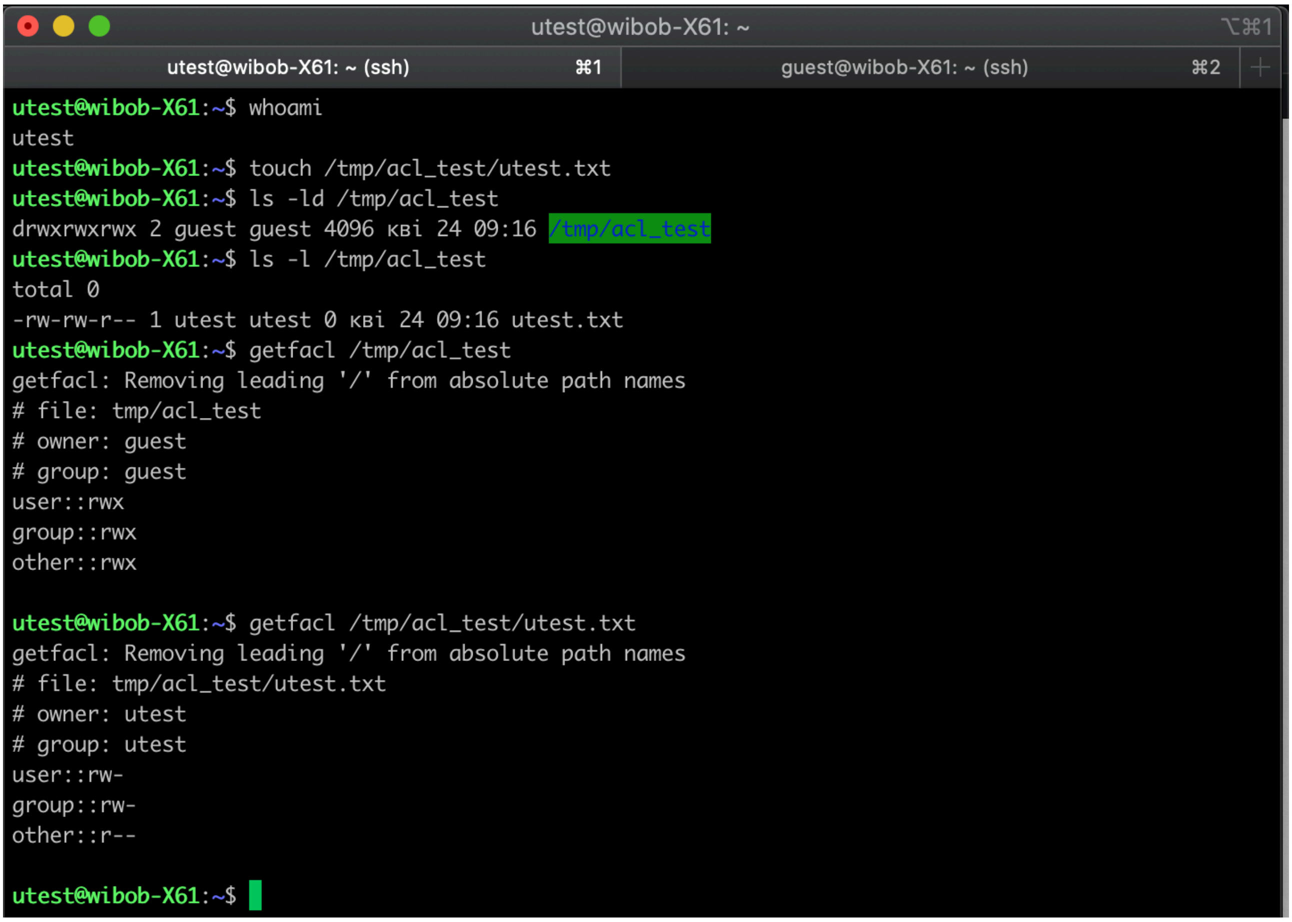
In what follows, we assume that there are two users: *guest* (included into the list of sudoers) and *utest*. None of the users is the superuser (i.e. UIDs of the users differ from 0).
The most task: to allow user *utest* visit *guest*'s home directory.
The average task: to acquaint yourself with the 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);
(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

```
guest@wibob-X61:~$ mkdir /tmp/acl_test
guest@wibob-X61:~$ ls -ld /tmp/acl_test/
drwxrwxr-x 2 guest guest 4096 кві 24 09:05 /tmp/acl_test/

guest@wibob-X61:~$ chmod o+rwx /tmp/acl_test/
guest@wibob-X61:~$ ls -ld /tmp/acl_test/
drwxrwxrwx 2 guest guest 4096 кві 24 09:05 /tmp/acl_test/
```

ls -ld /tmp/acl_test
ls -l /tmp/acl_test
To check ACL permissions do:
ge4acl /tmp/acl_test
ge4acl /tmp/acl_test/utest.txt

## ge4acl is it command alias?? I was using canonical getfacl command



```
utest@wibob-X61:~$ whoami
utest
utest@wibob-X61:~$ touch /tmp/acl_test/utest.txt
utest@wibob-X61:~$ ls -ld /tmp/acl_test
drwxrwxrwx 2 guest guest 4096 кві 24 09:16 /tmp/acl_test
utest@wibob-X61:~$ ls -l /tmp/acl_test
total 0
-rw-rw-r-- 1 utest utest 0 кві 24 09:16 utest.txt
utest@wibob-X61:~$ getfacl /tmp/acl_test
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test
# owner: guest
# group: guest
user::rwx
group::rwx
other::rwx

utest@wibob-X61:~$ getfacl /tmp/acl_test/utest.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/utest.txt
# owner: utest
# group: utest
user::rw-
group::rw-
other::r--

utest@wibob-X61:~$
```

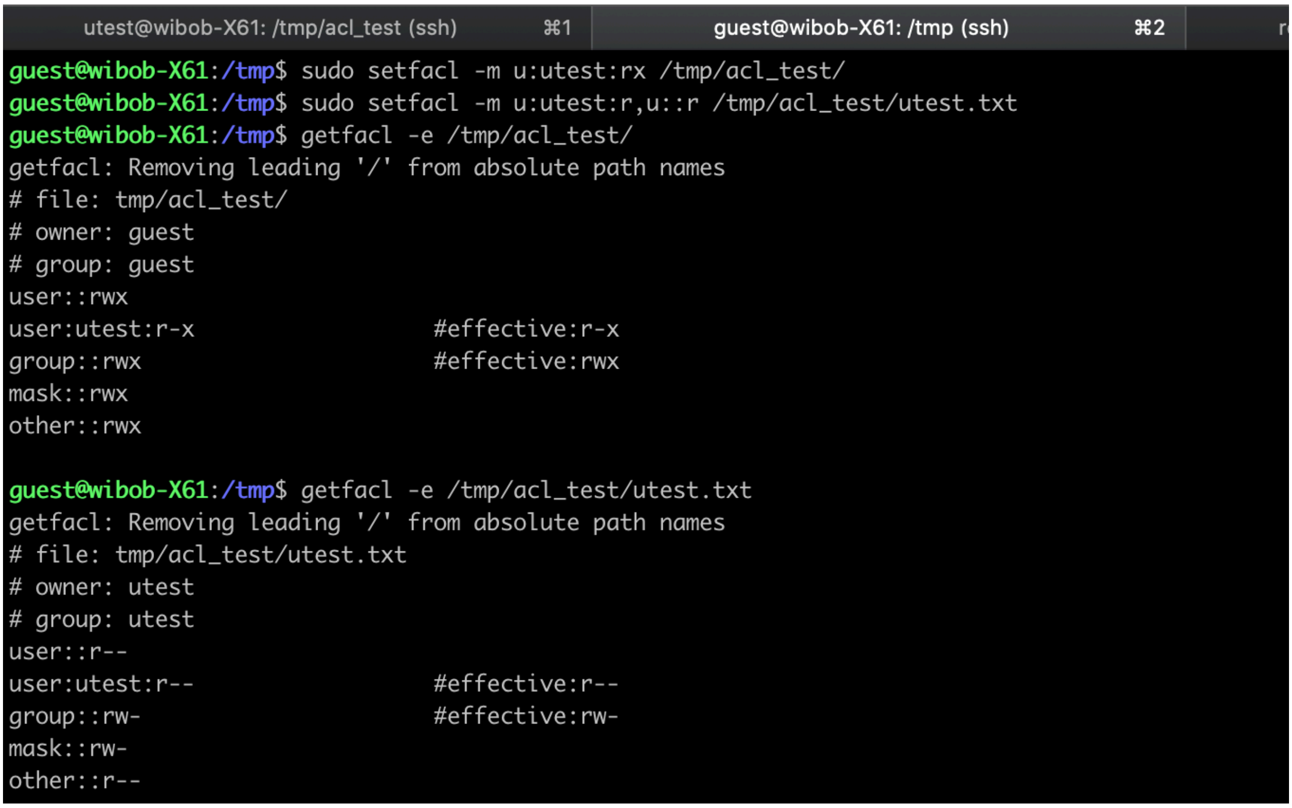
3. Employ ACL to block any activity except for reading, for user *utest* with respect to directory /*tmp/acl_test* (hint: use **se4acl**). Test if the actions are effectively prohibited

## first line set read and enter rights to folder

## second line set read-only rights on file for utest user for utest.txt

u:utest:r - its rule fore use itself, but in our case more important next

u::r - it's restrict acces to file to utest like a OWNER it has higher priority and without this string user has more priveleges (System UMASK)

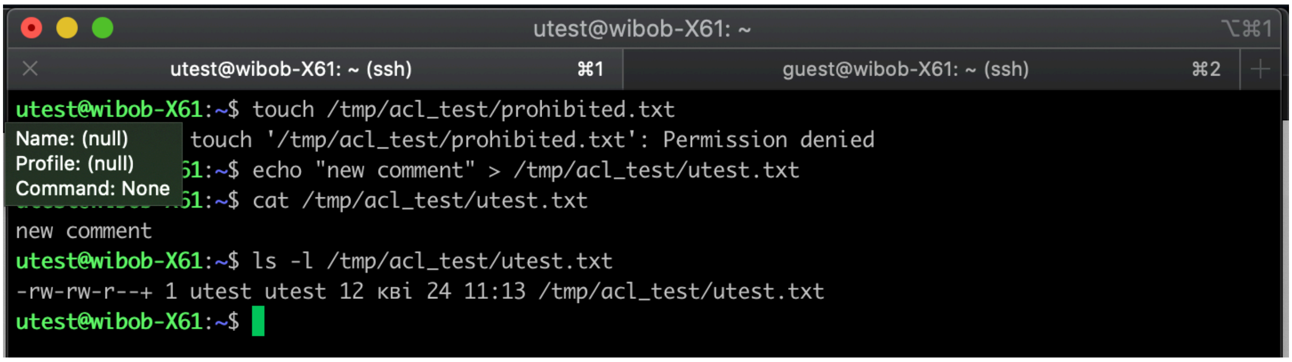

utest@wibob-X61: /tmp/acl_test (ssh)
⌘1
guest@wibob-X61: /tmp (ssh)
⌘2

```
guest@wibob-X61:/tmp$ sudo setfacl -m u:utest:rx /tmp/acl_test/
guest@wibob-X61:/tmp$ sudo setfacl -m u:utest:r,u::r /tmp/acl_test/utest.txt
guest@wibob-X61:/tmp$ getfacl -e /tmp/acl_test/
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/
# owner: guest
# group: guest
user::rwx
user:utest:r-x                  #effective:r-x
group::rwx                      #effective:rwx
mask::rwx
other::rwx

guest@wibob-X61:/tmp$ getfacl -e /tmp/acl_test/utest.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/utest.txt
# owner: utest
# group: utest
user::r--
user:utest:r--                  #effective:r--
group::rw-                      #effective:rw-
mask::rw-
other::r--
```

touch /tmp/acl_test/prohibited.txt
Is it possible to invoke this command?
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*).

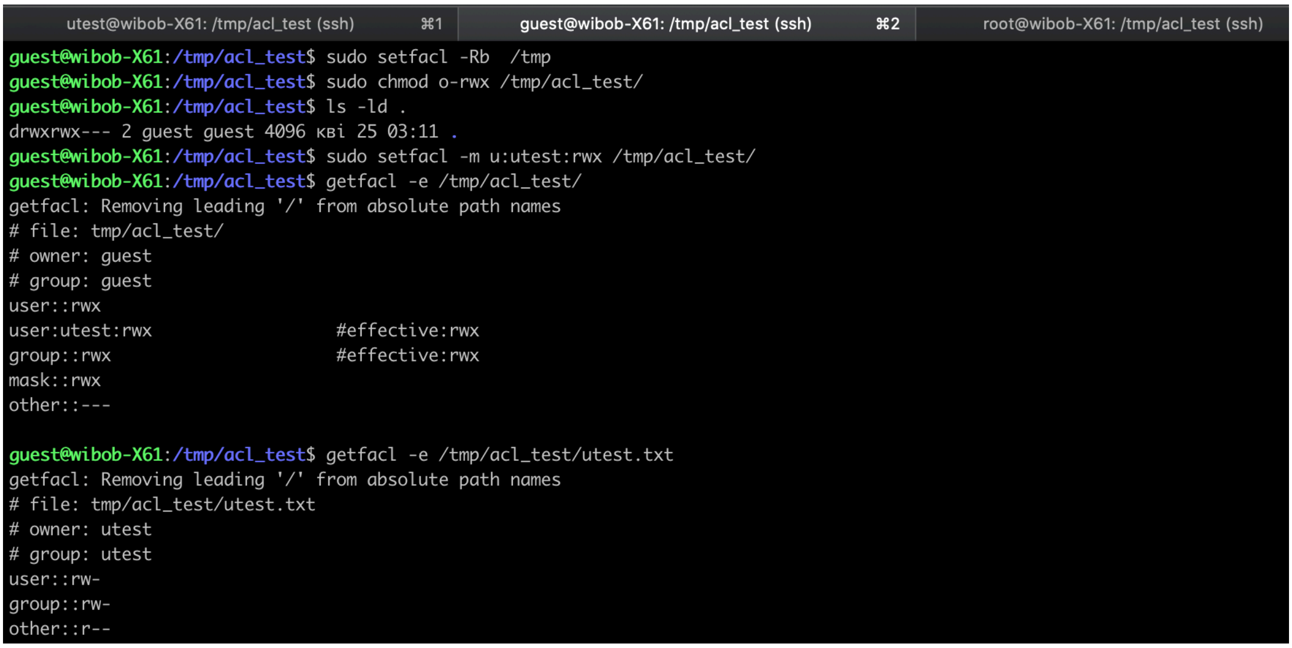
##Yes it's possible by forbidding not exectly utest user but OWNER with ACL rule .



```
utest@wibob-X61:~$ touch /tmp/acl_test/prohibited.txt
              touch '/tmp/acl_test/prohibited.txt': Permission denied
             51:~$ echo "new comment" > /tmp/acl_test/utest.txt
             61:~$ cat /tmp/acl_test/utest.txt
new comment
utest@wibob-X61:~$ ls -l /tmp/acl_test/utest.txt
-rw-rw-r--+ 1 utest utest 12 квi 24 11:13 /tmp/acl_test/utest.txt
utest@wibob-X61:~$
```

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).

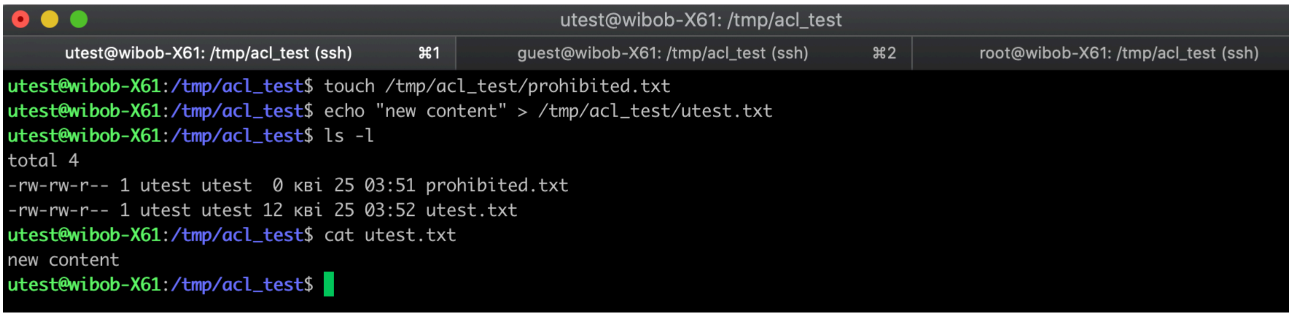
## to complete this task I have cleared all ACL rules to start from scratch.
## sudo chmod o-rwx /tmp/acl_test/ - remove POSIX permission for other user
## sudo setfacl -m u:utest:rwx /tmp/acl_test/ - creating rwx rights for user utest


utest@wibob-X61: /tmp/acl_test (ssh)
⌘1
guest@wibob-X61: /tmp/acl_test (ssh)
⌘2
root@wibob-X61: /tmp/acl_test (ssh)

```
guest@wibob-X61:/tmp/acl_test$ sudo setfacl -Rb  /tmp
guest@wibob-X61:/tmp/acl_test$ sudo chmod o-rwx /tmp/acl_test/
guest@wibob-X61:/tmp/acl_test$ ls -ld .
drwxrwx--- 2 guest guest 4096 квi 25 03:11 .
guest@wibob-X61:/tmp/acl_test$ sudo setfacl -m u:utest:rwx /tmp/acl_test/
guest@wibob-X61:/tmp/acl_test$ getfacl -e /tmp/acl_test/
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/
# owner: guest
# group: guest
user::rwx
user:utest:rwx                  #effective:rwx
group::rwx                      #effective:rwx
mask::rwx
other::---

guest@wibob-X61:/tmp/acl_test$ getfacl -e /tmp/acl_test/utest.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/utest.txt
# owner: utest
# group: utest
user::rw-
group::rw-
other::r--
```

## Yes, it's possible to override POSIX permission to grant access through ACL rule (we can see folder still forbidden to other user, but not for utest)


utest@wibob-X61: /tmp/acl_test
utest@wibob-X61: /tmp/acl_test (ssh)
⌘1
guest@wibob-X61: /tmp/acl_test (ssh)
⌘2
root@wibob-X61: /tmp/acl_test (ssh)

```
utest@wibob-X61:/tmp/acl_test$ touch /tmp/acl_test/prohibited.txt
utest@wibob-X61:/tmp/acl_test$ echo "new content" > /tmp/acl_test/utest.txt
utest@wibob-X61:/tmp/acl_test$ ls -l
total 4
-rw-rw-r-- 1 utest utest  0 квi 25 03:51 prohibited.txt
-rw-rw-r-- 1 utest utest 12 квi 25 03:52 utest.txt
utest@wibob-X61:/tmp/acl_test$ cat utest.txt
new content
utest@wibob-X61:/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 **se4acl** 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 **ge4acl**.

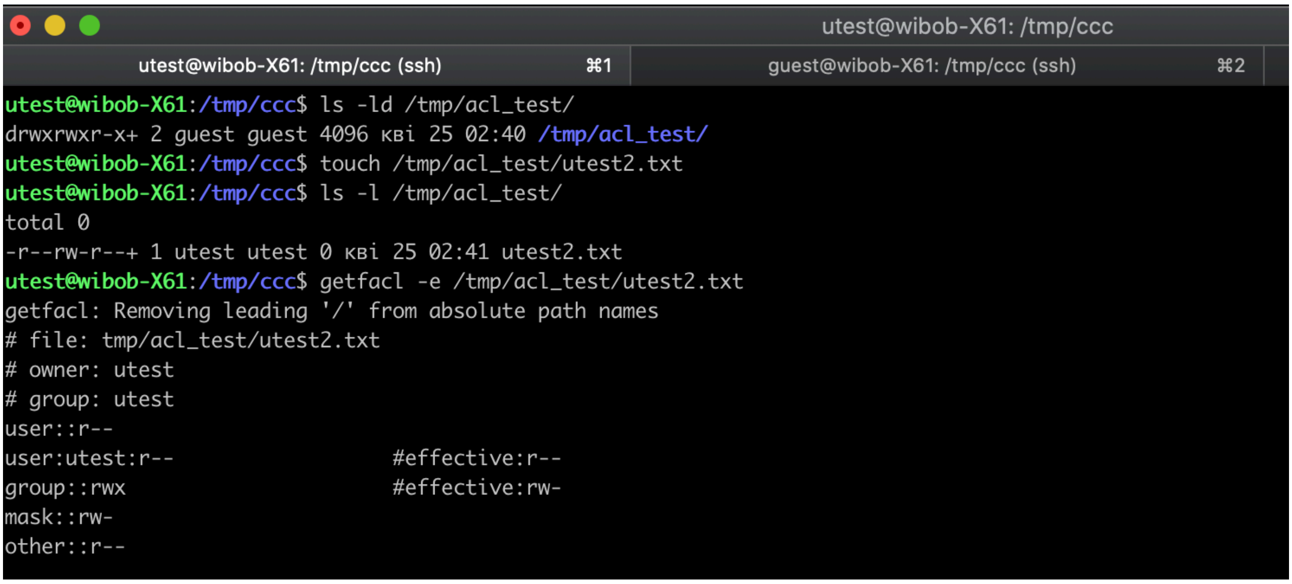
## sudo setfacl -m d:u:utest:r,d:u::r,u:utest:rwx /tmp/acl_test

d:u:utest:r - for all created files set read-only for utest

d:u::r, - for files created by utest, without this files take permissions from system umask. in this keys works high OWNER priority

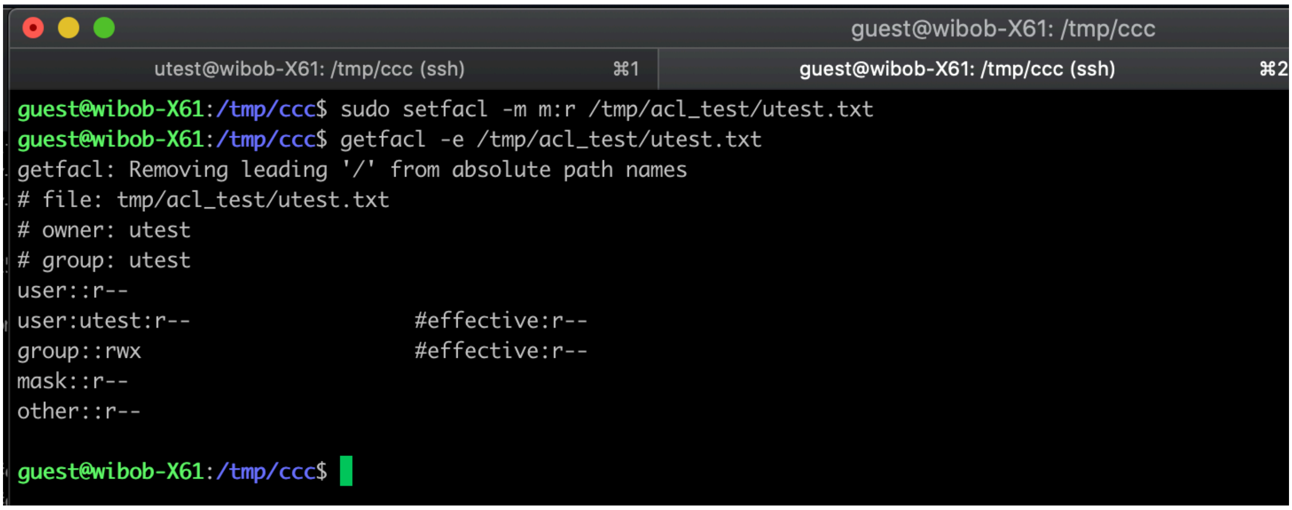
u:utest:rwx - set permissions exactly to folder

/tmp/acl_test - folder for operations


utest@wibob-X61: /tmp/ccc
utest@wibob-X61: /tmp/ccc (ssh) ⌘1
guest@wibob-X61: /tmp/ccc (ssh) ⌘2

```
utest@wibob-X61:/tmp/ccc$ ls -ld /tmp/acl_test/
drwxrwxr-x+ 2 guest guest 4096 кві 25 02:40 /tmp/acl_test/
utest@wibob-X61:/tmp/ccc$ touch /tmp/acl_test/utest2.txt
utest@wibob-X61:/tmp/ccc$ ls -l /tmp/acl_test/
total 0
-r--rw-r--+ 1 utest utest 0 кві 25 02:41 utest2.txt
utest@wibob-X61:/tmp/ccc$ getfacl -e /tmp/acl_test/utest2.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/utest2.txt
# owner: utest
# group: utest
user::r--
user:utest:r--                  #effective:r--
group::rwx                      #effective:rw-
mask::rw-
other::r--
```

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 **ge4acl**.

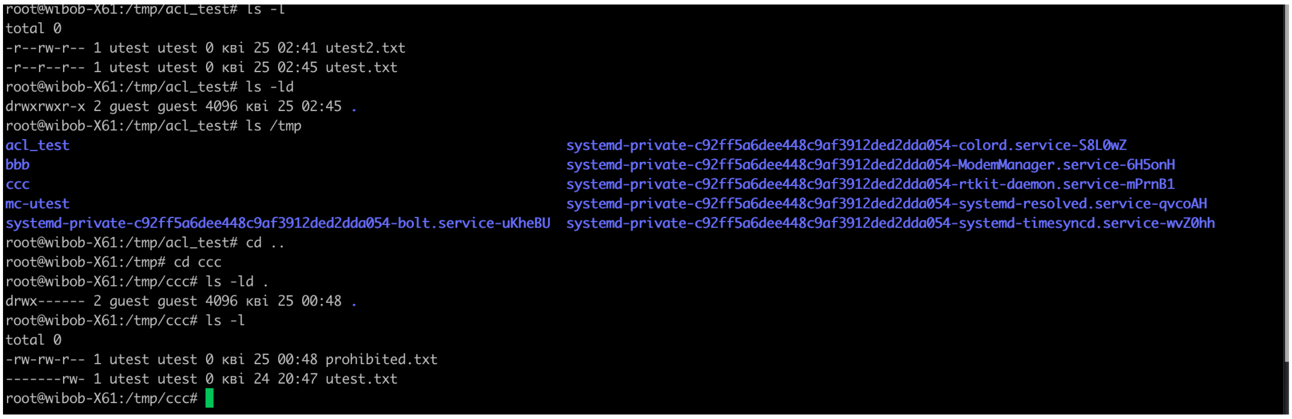

guest@wibob-X61: /tmp/ccc
utest@wibob-X61: /tmp/ccc (ssh) ⌘1
guest@wibob-X61: /tmp/ccc (ssh) ⌘2

```
guest@wibob-X61:/tmp/ccc$ sudo setfacl -m m:r /tmp/acl_test/utest.txt
guest@wibob-X61:/tmp/ccc$ getfacl -e /tmp/acl_test/utest.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/utest.txt
# owner: utest
# group: utest
user::r--
user:utest:r--                  #effective:r--
group::rwx                      #effective:r--
mask::r--
other::r--

guest@wibob-X61:/tmp/ccc$
```

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

## sudo setfacl -Rb /tmp



```
root@wibob-X61:/tmp/acl_test# ls -l
total 0
-r--rw-r-- 1 utest utest 0 кві 25 02:41 utest2.txt
-r--r--r-- 1 utest utest 0 кві 25 02:45 utest.txt
root@wibob-X61:/tmp/acl_test# ls -ld
drwxrwxr-x 2 guest guest 4096 кві 25 02:45 .
root@wibob-X61:/tmp/acl_test# ls /tmp
acl_test                                                   systemd-private-c92ff5a6dee448c9af3912ded2dda054-colord.service-S8L0wZ
bbb                                                        systemd-private-c92ff5a6dee448c9af3912ded2dda054-ModemManager.service-6H5onH
ccc                                                        systemd-private-c92ff5a6dee448c9af3912ded2dda054-rtkit-daemon.service-mPrnB1
mc-utest                                                   systemd-private-c92ff5a6dee448c9af3912ded2dda054-systemd-resolved.service-qvcoAH
systemd-private-c92ff5a6dee448c9af3912ded2dda054-bolt.service-uKheBU  systemd-private-c92ff5a6dee448c9af3912ded2dda054-systemd-timesyncd.service-wvZ0hh
root@wibob-X61:/tmp/acl_test# cd ..
root@wibob-X61:/tmp# cd ccc
root@wibob-X61:/tmp/ccc# ls -ld .
drwx------ 2 guest guest 4096 кві 25 00:48 .
root@wibob-X61:/tmp/ccc# ls -l
total 0
-rw-rw-r-- 1 utest utest 0 кві 25 00:48 prohibited.txt
-------rw- 1 utest utest 0 кві 24 20:47 utest.txt
root@wibob-X61:/tmp/ccc# 
```