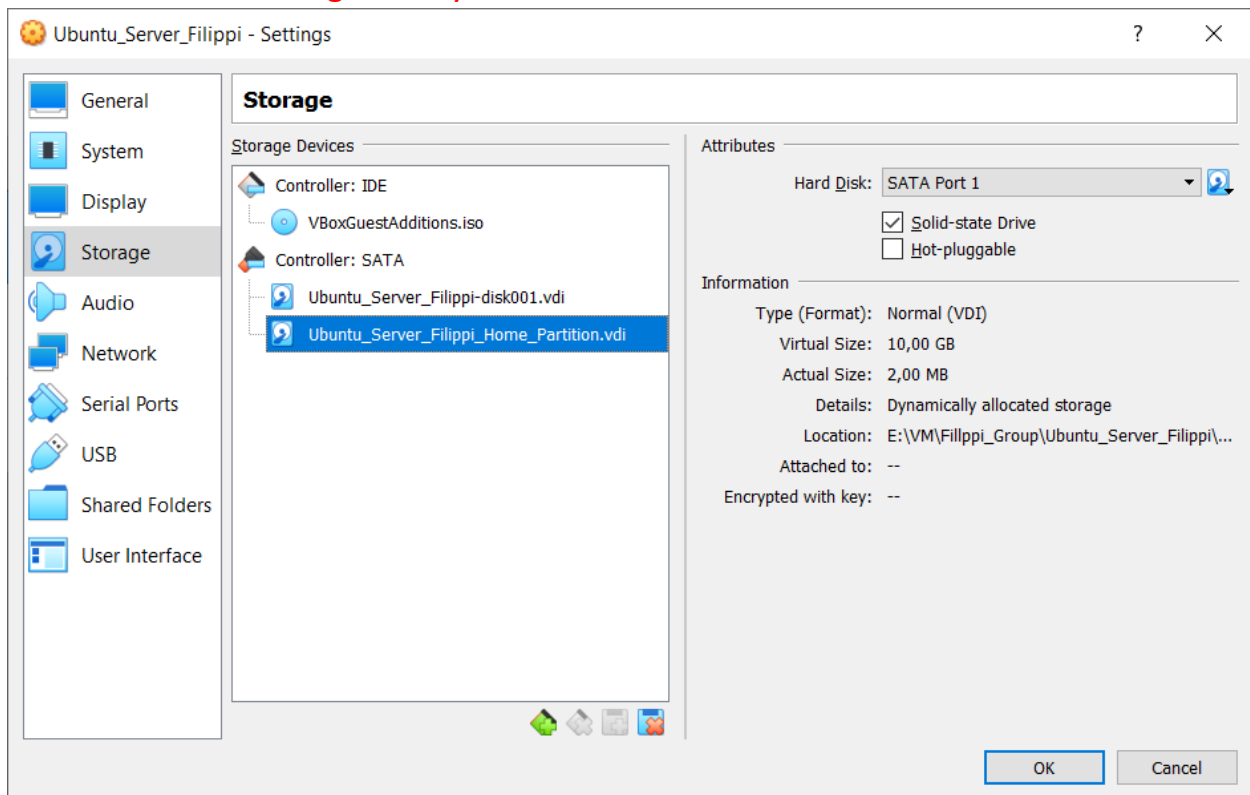


EPAM University Programs
DevOps external course
Module 4 Linux & Bash Essentials
TASK 4.7

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
```

```
root@ubuntu_server1: /
tmpfs          487M      0  487M   0% /sys/fs/cgroup
/dev/loop0     94M      94M    0 100% /snap/core/8935
/dev/loop1     94M      94M    0 100% /snap/core/9066
/dev/sda1      511M    6.1M   505M   2% /boot/efi
tmpfs          100K      0   100K   0% /var/lib/lxd/shmounts
tmpfs          100K      0   100K   0% /var/lib/lxd/devlxd
DevOPS         477G    355G   123G   75% /media/sf_DevOPS
tmpfs          98M      0    98M   0% /run/user/0
/dev/sdb1      9.8G     37M    9.3G   1% /home
root@ubuntu_server1:/# umount /home
root@ubuntu_server1:/# vi /etc/fstab
root@ubuntu_server1:/# mount -a
root@ubuntu_server1:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            456M      0  456M   0% /dev
tmpfs           98M 1000K   97M   2% /run
/dev/sda2       9.3G  5.6G  3.3G  64% /
tmpfs           487M      0  487M   0% /dev/shm
tmpfs           5.0M      0   5.0M   0% /run/lock
tmpfs           487M      0  487M   0% /sys/fs/cgroup
/dev/loop0      94M    94M    0 100% /snap/core/8935
/dev/loop1      94M    94M    0 100% /snap/core/9066
/dev/sda1       511M    6.1M   505M   2% /boot/efi
tmpfs           100K      0   100K   0% /var/lib/lxd/shmounts
tmpfs           100K      0   100K   0% /var/lib/lxd/devlxd
DevOPS          477G    355G   123G   75% /media/sf_DevOPS
tmpfs           98M      0    98M   0% /run/user/0
/dev/sdb1       9.8G     37M    9.3G   1% /home
root@ubuntu_server1:/#
```

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

```
root@ubuntu_server1: /home
root@ubuntu_server1:/home# rm aquota.user
root@ubuntu_server1:/home# cd /
root@ubuntu_server1:/# quotacheck -ugm /home
root@ubuntu_server1:/# ls
VBox.log  boot  dev  home  initrd.img.old  lib64  media  opt  root  sbin  srv  sys
bin       cdrom  etc  initrd.img  lib  lost+found  mnt  proc  run  snap  swap.img  tmp
root@ubuntu_server1:/# cd /home/
root@ubuntu_server1:/home# ls
aquota.user  fill  lost+found
root@ubuntu_server1:/home# quotacheck -ugm /
quotacheck: Mountpoint (or device) / not found or has no quota enabled.
quotacheck: Cannot find filesystem to check or filesystem not mounted with quota option.
root@ubuntu_server1:/home# quotacheck -ugm /home
root@ubuntu_server1:/home# quotaon -v /home
/dev/sdb1 [/home]: user quotas turned on
root@ubuntu_server1:/home# ^C
root@ubuntu_server1:/home# useradd -g adm -s /bin/bash -d /home/utuser -m utest
root@ubuntu_server1:/home# passwd utest
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@ubuntu_server1:/home# edquota -u utest
root@ubuntu_server1:/home# quota -vs utest
Disk quotas for user utest (uid 1001):
    Filesystem  space   quota   limit   grace   files   quota   limit   grace
    /dev/sdb1   16K    100M   150M           4         0         0
root@ubuntu_server1:/home# repquota -s /home
*** Report for user quotas on device /dev/sdb1
Block grace time: 7days; Inode grace time: 7days

      Space limits                File limits
User      used  soft  hard  grace  used  soft  hard  grace
-----
root      --   68K   0K   0K           14    0    0
fill      --   32K   0K   0K           10    0    0
utest     --   16K  100M  150M           4    0    0

root@ubuntu_server1:/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				<- ~/test			
.n	Name	Size	Modify time	.n	Name	Size	Modify time
/..		UP--DIR	Apr 13 08:03	/..		UP--DIR	Apr 26 06:16
/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							

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

```
root@ubuntu_server1: /home
root@ubuntu_server1:/home# su - utest
utest@ubuntu_server1:~$ mc
utest@ubuntu_server1:/~$ OA
OA: command not found
utest@ubuntu_server1:~/test$ exit
exit
utest@ubuntu_server1:~$ exit
logout
root@ubuntu_server1:/home# repquota -s /home
*** Report for user quotas on device /dev/sdb1
Block grace time: 7days; Inode grace time: 7days

```

		Space limits				File limits			
User	used	soft	hard	grace	used	soft	hard	grace	
root	-- 68K	0K	0K		14	0	0		
fill	-- 32K	0K	0K		10	0	0		
utest	+-- 150M	100M	150M	6days	16297	0	0		

```
root@ubuntu_server1:/home#
```

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](https://linuxconfig.org/how-to-manage-acls-on-linux) 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

```
root@ubuntu_server1: /etc/rsyslog.d
Last login: Tue Apr 28 14:46:28 2020 from 192.168.1.131
root@ubuntu_server1:~# man exec
root@ubuntu_server1:~# cd /etc/rsyslog.d/
root@ubuntu_server1:/etc/rsyslog.d# ls
20-ufw.conf  21-cloudinit.conf  50-default.conf
root@ubuntu_server1:/etc/rsyslog.d# cp 20-ufw.conf 10-cl-devops.conf
root@ubuntu_server1:/etc/rsyslog.d# vi 10-cl-devops.conf
root@ubuntu_server1:/etc/rsyslog.d# mv 10-cl-devops.conf 10-ca-devops.conf
root@ubuntu_server1:/etc/rsyslog.d# cat 10-ca-devops.conf

# Filter all messages whose tag starts with CA
# CA means Console Actions
:syslogtag, startswith, "CA_Devops" /var/log/DevOPS_Console_action.log

# The stop command prevents this message from getting processed any further.
& stop
root@ubuntu_server1:/etc/rsyslog.d# systemctl restart rsyslog.service
root@ubuntu_server1:/etc/rsyslog.d# echo Is it works or not? | logger -t CA_Devops
root@ubuntu_server1:/etc/rsyslog.d# cat /var/log/DevOPS_Console_action.log
Apr 28 15:07:39 ubuntu_server1 CA_Devops: Is it works or not?
root@ubuntu_server1:/etc/rsyslog.d#
```

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

```
root@ubuntu_server1: /home/guest
Apr 28 16:05:40 ubuntu_server1 CA_Devops: guest : /home/guest : Command=$BASH_COMMAND
Apr 28 16:06:22 ubuntu_server1 CA_Devops: message repeated 4 times: [ guest : /home/guest : Command=$BAS
H_COMMAND]
^C
root@ubuntu_server1: /home/guest# cat /dev/null > /var/log/DevOPS_Console_action.log
root@ubuntu_server1: /home/guest# tail -f /var/log/DevOPS_Console_action.log
Apr 28 16:31:07 ubuntu_server1 CA_Devops: guest : /home/guest : [ -d "$HOME/bin" ]
Apr 28 16:31:07 ubuntu_server1 CA_Devops: guest : /home/guest : [ -d "$HOME/.local/bin" ]
Apr 28 16:31:34 ubuntu_server1 CA_Devops: guest : /home/guest : tune2fs -l /dev/sdb /dev/sdb1
Apr 28 16:31:40 ubuntu_server1 CA_Devops: guest : /home/guest : tune2fs -l /dev/sdb
Apr 28 16:31:48 ubuntu_server1 CA_Devops: guest : /home/guest : sudo tune2fs -l /dev/sdb
Apr 28 16:33:14 ubuntu_server1 CA_Devops: guest : /home/guest : su -
Apr 28 16:35:27 ubuntu_server1 CA_Devops: guest : /home/guest : sudo tune2fs -l /dev/sdb
Apr 28 16:35:49 ubuntu_server1 CA_Devops: guest : /home/guest : su -
Apr 28 16:38:33 ubuntu_server1 CA_Devops: guest : /home/guest : sudo tune2fs -l /dev/sdb
Apr 28 16:38:51 ubuntu_server1 CA_Devops: guest : /home/guest : exit
Apr 28 16:38:51 ubuntu_server1 CA_Devops: guest : /home/guest : [ "$SHLVL" = 1 ]
Apr 28 16:38:51 ubuntu_server1 CA_Devops: guest : /home/guest : [ -x /usr/bin/clear_console ]
Apr 28 16:38:51 ubuntu_server1 CA_Devops: guest : /home/guest : /usr/bin/clear_console -q
Apr 28 16:39:33 ubuntu_server1 CA_Devops: guest : /home/guest : [ -d "$HOME/bin" ]
Apr 28 16:39:33 ubuntu_server1 CA_Devops: guest : /home/guest : [ -d "$HOME/.local/bin" ]
Apr 28 16:39:43 ubuntu_server1 CA_Devops: guest : /home/guest : sudo ls
Apr 28 16:39:53 ubuntu_server1 CA_Devops: guest : /home/guest : sudo tune2fs -l /dev/sdb
Apr 28 16:39:56 ubuntu_server1 CA_Devops: guest : /home/guest : sudo tune2fs -l /dev/sdb1
Apr 28 16:40:00 ubuntu_server1 CA_Devops: guest : /home/guest : sudo tune2fs -l /dev/sdb /dev/sdb1
Apr 28 16:40:03 ubuntu_server1 CA_Devops: guest : /home/guest : sudo tune2fs -l /dev/sdb1
Apr 28 16:40:15 ubuntu_server1 CA_Devops: guest : /home/guest : tune2fs -l /dev/sdb1
Apr 28 16:40:24 ubuntu_server1 CA_Devops: guest : /home/guest : sudo tune2fs -l /dev/sdb1
Apr 28 16:41:32 ubuntu_server1 CA_Devops: guest : /home/guest : blkid
Apr 28 16:41:44 ubuntu_server1 CA_Devops: guest : /home/guest : sudo blkid
Apr 28 16:43:18 ubuntu_server1 CA_Devops: guest : /home/guest : sudo tune2fs -l /dev/sdb1
Apr 28 16:43:18 ubuntu_server1 CA_Devops: guest : /home/guest : grep --color=auto acl
Apr 28 16:43:37 ubuntu_server1 CA_Devops: guest : /home/guest : cd /tmp/
Apr 28 16:43:47 ubuntu_server1 CA_Devops: guest : /tmp : mkdir acl_test
Apr 28 16:44:16 ubuntu_server1 CA_Devops: guest : /tmp : ls --color=auto -ld acl_test
Apr 28 16:45:29 ubuntu_server1 CA_Devops: guest : /tmp : chmod o+rwX acl_test
Apr 28 16:45:30 ubuntu_server1 CA_Devops: guest : /tmp : ls --color=auto -ld acl_test
Apr 28 16:46:14 ubuntu_server1 CA_Devops: guest : /tmp : su - utest
```

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).

```
guest@ubuntu_server1: ~  
Desired extra isize:      32  
Journal inode:           8  
Default directory hash:  half_md4  
Directory Hash Seed:     ba7805ba-7564-40cc-a1db-e191d2d182ce  
Journal backup:          inode blocks  
Checksum type:           crc32c  
Checksum:                0x538a861f  
guest@ubuntu_server1:~$ blkid  
/dev/sr0: UUID="2020-02-18-17-20-05-35" LABEL="VBox_GAs_6.1.4" TYPE="iso9660"  
/dev/sda1: UUID="FFB5-A146" TYPE="vfat" PARTUUID="a466e47c-0091-41cf-9bf0-6e688d3b1d53"  
/dev/sda2: UUID="8164aab3-d001-4bef-ac83-875ae60ad37a" TYPE="ext4" PARTUUID="dcc211df-1913-4db5-aa4d-904592d32050"  
/dev/sdb1: UUID="6fd55804-9485-4b9a-8909-79c060038771" TYPE="ext4" PARTLABEL="primary" PARTUUID="7692d614-f65d-4c1d-b1d1-4da8ab2a4ca5"  
guest@ubuntu_server1:~$ sudo blkid  
/dev/sr0: UUID="2020-02-18-17-20-05-35" LABEL="VBox_GAs_6.1.4" TYPE="iso9660"  
/dev/sda1: UUID="FFB5-A146" TYPE="vfat" PARTUUID="a466e47c-0091-41cf-9bf0-6e688d3b1d53"  
/dev/sda2: UUID="8164aab3-d001-4bef-ac83-875ae60ad37a" TYPE="ext4" PARTUUID="dcc211df-1913-4db5-aa4d-904592d32050"  
/dev/sdb1: UUID="6fd55804-9485-4b9a-8909-79c060038771" TYPE="ext4" PARTLABEL="primary" PARTUUID="7692d614-f65d-4c1d-b1d1-4da8ab2a4ca5"  
/dev/loop0: TYPE="squashfs"  
/dev/loop1: TYPE="squashfs"  
guest@ubuntu_server1:~$ sudo tune2fs -l /dev/sdb1 | grep acl  
Default mount options:      user_xattr acl  
guest@ubuntu_server1:~$
```

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, after 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
```



```
utest@ubuntu_server1: /tmp/acl_test
guest@ubuntu_server1:/tmp$ mkdir acl_test
guest@ubuntu_server1:/tmp$ ls -ld  acl_test
drwxrwxr-x 2 guest guest 4096 Apr 28 16:57 acl_test
guest@ubuntu_server1:/tmp$ chmod o+w acl_test
guest@ubuntu_server1:/tmp$ ls -ld  acl_test
drwxrwxrwx 2 guest guest 4096 Apr 28 16:57 acl_test
guest@ubuntu_server1:/tmp$ su - utest
Password:
utest@ubuntu_server1:~$ cd /tmp/acl_test/
utest@ubuntu_server1:/tmp/acl_test$ touch utest.txt
utest@ubuntu_server1:/tmp/acl_test$ ls -ld /tmp/acl_test
drwxrwxrwx 2 guest guest 4096 Apr 28 16:58 /tmp/acl_test
utest@ubuntu_server1:/tmp/acl_test$ ls -l /tmp/acl_test
total 0
-rw-r--r-- 1 utest adm 0 Apr 28 16:58 utest.txt
utest@ubuntu_server1:/tmp/acl_test$
```

To check ACL permissions do:

getfacl /tmp/acl_test

getfacl /tmp/acl_test/utest.txt

```
guest@ubuntu_server1: /tmp
utest@ubuntu_server1:~$ cd /tmp/acl_test/
utest@ubuntu_server1:/tmp/acl_test$ touch utest.txt
utest@ubuntu_server1:/tmp/acl_test$ ls -ld /tmp/acl_test
drwxrwxrwx 2 guest guest 4096 Apr 28 16:58 /tmp/acl_test
utest@ubuntu_server1:/tmp/acl_test$ ls -l /tmp/acl_test
total 0
-rw-r--r-- 1 utest adm 0 Apr 28 16:58 utest.txt
utest@ubuntu_server1:/tmp/acl_test$ exit
logout
guest@ubuntu_server1:/tmp$ 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

guest@ubuntu_server1:/tmp$ getfacl /tmp/acl_test/utest.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/utest.txt
# owner: utest
# group: adm
user::rw-
group::r--
other::r--

guest@ubuntu_server1:/tmp$
```

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

```
utest@ubuntu_server1: ~  
Password:  
root@ubuntu_server1:~# setfacl -m u:utest:rx /tmp/acl_test  
root@ubuntu_server1:~# exit  
logout  
utest@ubuntu_server1:~$ getfacl /tmp/acl_test  
getfacl: Removing leading '/' from absolute path names  
# file: tmp/acl_test  
# owner: guest  
# group: guest  
user::rwx  
user:utest:r-x  
group::rwx  
mask::rwx  
other::rwx  
  
utest@ubuntu_server1:~$ getfacl /tmp/acl_test/utest.txt  
getfacl: Removing leading '/' from absolute path names  
# file: tmp/acl_test/utest.txt  
# owner: utest  
# group: adm  
user::rw-  
group::r--  
other::r--  
  
utest@ubuntu_server1:~$ touch /tmp/acl_test/prohib.txt  
touch: cannot touch '/tmp/acl_test/prohib.txt': Permission denied  
utest@ubuntu_server1:~$ echo "new string" > /tmp/acl_test/utest.txt  
utest@ubuntu_server1:~$
```

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`

```
utest@ubuntu_server1: ~  
other::rwx  
  
guest@ubuntu_server1:/tmp$ getfacl /tmp/acl_test/utest.txt  
getfacl: Removing leading '/' from absolute path names  
# file: tmp/acl_test/utest.txt  
# owner: utest  
# group: adm  
user::rw-  
group::r--  
other::r--  
  
guest@ubuntu_server1:/tmp$ sudo setfacl -m u:utest:r /tmp/acl_test  
[sudo] password for guest:  
guest@ubuntu_server1:/tmp$ getfacl /tmp/acl_test  
getfacl: Removing leading '/' from absolute path names  
# file: tmp/acl_test  
# owner: guest  
# group: guest  
user::rwx  
user:utest:r--  
group::rwx  
mask::rwx  
other::rwx  
  
guest@ubuntu_server1:/tmp$ su - utest  
Password:  
utest@ubuntu_server1:~$ cd /tmp/acl_test/  
-su: cd: /tmp/acl_test/: Permission denied  
utest@ubuntu_server1:~$ cat /tmp/acl_test/utest.txt  
.bash_history .bashrc .config/ .profile  
.bash_logout .cache/ .local/ test/  
utest@ubuntu_server1:~$ touch /tmp/acl_test/prohib.txt  
touch: cannot touch '/tmp/acl_test/prohib.txt': Permission denied  
utest@ubuntu_server1:~$ echo "new string" > /tmp/acl_test/utest.txt  
-su: /tmp/acl_test/utest.txt: Permission 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 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

```
utest@ubuntu_server1: ~
root@ubuntu_server1:~# cd /tmp/
root@ubuntu_server1:/tmp# chmod 770 acl_test
root@ubuntu_server1:/tmp# ls -ld acl_test/
drwxrwx---+ 2 guest guest 4096 Apr 28 16:58 acl_test/
root@ubuntu_server1:/tmp# setfacl -m u:utest:rwx /tmp/acl_test
root@ubuntu_server1:/tmp# ls -ld acl_test/
drwxrwx---+ 2 guest guest 4096 Apr 28 16:58 acl_test/
root@ubuntu_server1:/tmp# getfacl acl_test
# file: acl_test
# owner: guest
# group: guest
user::rwx
user:utest:rwx
group::rwx
mask::rwx
other:---

root@ubuntu_server1:/tmp# exit
logout
utest@ubuntu_server1:~$ ls -ld /tmp/acl_test
drwxrwx---+ 2 guest guest 4096 Apr 28 16:58 /tmp/acl_test
utest@ubuntu_server1:~$ 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:---

utest@ubuntu_server1:~$ ^C
utest@ubuntu_server1:~$ touch /tmp/acl_test/utest_acl.txt
utest@ubuntu_server1:~$
```

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**.

```
utest@ubuntu_server1: ~  
user::rw-  
group::r--  
other::r--  
  
root@ubuntu_server1:~# setfacl -d u:utest:r /tmp/acl_test  
Usage: setfacl [-bkndRLP] { -m|-M|-x|-X ... } file ...  
Try `setfacl --help' for more information.  
root@ubuntu_server1:~# setfacl -m default:u:utest:r /tmp/acl_test  
root@ubuntu_server1:~# 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::---  
default:user::rwx  
default:user:utest:r--  
default:group::rwx  
default:mask::rwx  
default:other::---  
  
root@ubuntu_server1:~# exit  
logout  
utest@ubuntu_server1:~$ touch /tmp/acl_test/utest2.txt  
utest@ubuntu_server1:~$ 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::---  
default:user::rwx  
default:user:utest:r--  
default:group::rwx  
default:mask::rwx  
default:other::---  
  
utest@ubuntu_server1:~$ getfacl /tmp/acl_test/utest2.txt  
mask::rwx  
other::---  
default:user::rwx  
default:user:utest:r--  
default:group::rwx  
default:mask::rwx  
default:other::---  
  
utest@ubuntu_server1:~$ getfacl /tmp/acl_test/utest2.txt  
getfacl: Removing leading '/' from absolute path names  
# file: tmp/acl_test/utest2.txt  
# owner: utest  
# group: adm  
user::rw-  
user:utest:r--  
group::rwx  
mask::rw-  
other::---  
#effective:rw-  
  
utest@ubuntu_server1:~$
```

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**.

```
root@ubuntu_server1: ~  
mask::rw-  
other::---  
  
utest@ubuntu_server1:~$ su -  
Password:  
root@ubuntu_server1:~# getfacl /tmp/acl_test/utest.txt  
getfacl: Removing leading '/' from absolute path names  
# file: tmp/acl_test/utest.txt  
# owner: utest  
# group: adm  
user::rw-  
group::r--  
other::r--  
  
root@ubuntu_server1:~# setfacl -m m:r /tmp/acl_test/utest.txt  
root@ubuntu_server1:~# getfacl /tmp/acl_test/utest.txt  
getfacl: Removing leading '/' from absolute path names  
# file: tmp/acl_test/utest.txt  
# owner: utest  
# group: adm  
user::rw-  
group::r--  
mask::r--  
other::r--  
root@ubuntu_server1:~#
```

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

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

```
root@ubuntu_server1: /tmp  
default:mask::rwx  
default:other::---  
  
root@ubuntu_server1:/tmp# ls -ld acl_test  
drwxrwx---+ 2 guest guest 4096 Apr 28 17:29 acl_test  
root@ubuntu_server1:/tmp# 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::---  
default:user::rwx  
default:user:utest:r--  
default:group::rwx  
default:mask::rwx  
default:other::---  
  
root@ubuntu_server1:/tmp# setfacl -b /tmp/acl_test  
root@ubuntu_server1:/tmp# getfacl /tmp/acl_test/  
getfacl: Removing leading '/' from absolute path names  
# file: tmp/acl_test/  
# owner: guest  
# group: guest  
user::rwx  
group::rwx  
other::---  
  
root@ubuntu_server1:/tmp# getfacl /tmp/acl_test/utest  
getfacl: /tmp/acl_test/utest: No such file or directory  
root@ubuntu_server1:/tmp# getfacl /tmp/acl_test/utest  
utest.txt      utest2.txt      utest_acl.txt  
root@ubuntu_server1:/tmp# getfacl /tmp/acl_test/utest*  
getfacl: Removing leading '/' from absolute path names  
# file: tmp/acl_test/utest.txt
```

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