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.

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).

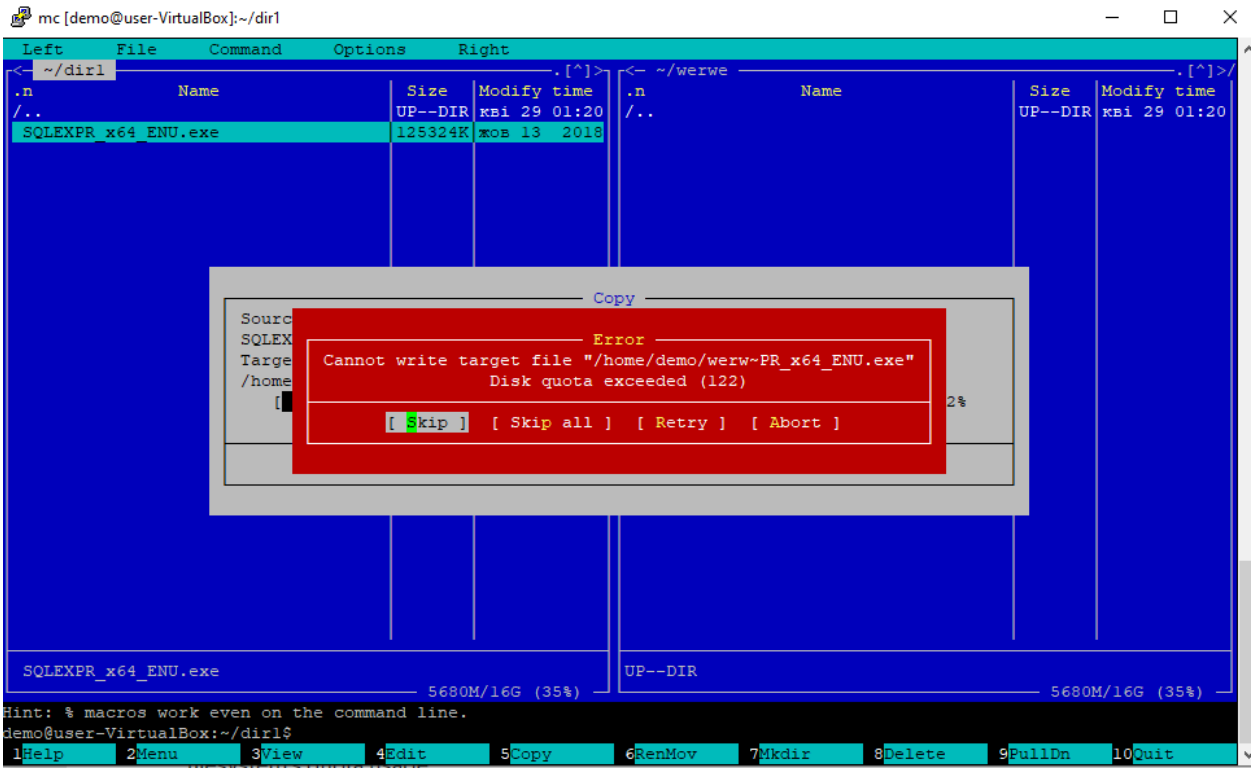
Note: if /home is not a mount point, then the **mount** and **quotaon** commands should be called with respect to the root partition /.

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).

```
user@user-VirtualBox:/tmp/testdir$ quota --version
Quota utilities version 4.04.
Compiled with: USE_LDAP_MAIL_LOOKUP EXT2_DIRECT HOSTS_ACCESS RPC RPC_SETQUOTA BSD_BEHAVIOUR
Bugs to jack@suse.cz
```

```
user@user-VirtualBox:~$ sudo mount -o remount /
user@user-VirtualBox:~$ cat /proc/mounts | grep ' / '
/dev/sda1 / ext4 rw,relatime,quota,usrquota,grpquota,errors=remount-ro 0 0
user@user-VirtualBox:~$ sudo quotacheck -ugm /
user@user-VirtualBox:~$ ls /
aquota.group  boot   etc         initrd.img.old  lost+found  opt   run   srv       tmp  VBox.log
aquota.user   cdrom  home        lib             media       proc  sbin  swapfile  usr  vmlinuz
bin           dev    initrd.img  lib64           mnt         root  snap  sys       var  vmlinuz.old
user@user-VirtualBox:~$ sudo quotaon -v /
/dev/sda1 [/]: group quotas turned on
/dev/sda1 [/]: user quotas turned on
user@user-VirtualBox:~$ sudo edquota -u demo
user@user-VirtualBox:~$ sudo quota -vs demo
Disk quotas for user demo (uid 1002):
     Filesystem   space   quota   limit   grace   files   quota   limit   grace
      /dev/sda1     28K    100M    150M              7       0       0
```

```
Done.
user@user-VirtualBox:/$ sudo repquota -s /
*** Report for user quotas on device /dev/sda1
Block grace time: 7days; Inode grace time: 7days
                        Space limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root       --    7638M      0K      0K            223k     0     0
man        --    1332K      0K      0K              83     0     0
lp         --    6528K      0K      0K               1     0     0
systemd-network --      36K      0K      0K               9     0     0
systemd-resolve --       0K      0K      0K               2     0     0
syslog     --    1288K      0K      0K              20     0     0
_apt       --      36K      0K      0K               4     0     0
avahi-autoipd --       4K      0K      0K               1     0     0
dnsmasq    --       4K      0K      0K               1     0     0
speech-dispatcher --       8K      0K      0K               2     0     0
colord     --      56K      0K      0K               5     0     0
hplip      --       4K      0K      0K               1     0     0
geoclue    --       8K      0K      0K               2     0     0
gdm        --     220K      0K      0K              43     0     0
user       --     670M      0K      0K            2710     0     0
vagrant    --      24K      0K      0K               6     0     0
lxd        --       8K      0K      0K               2     0     0
demo       +-     150M    100M    150M  6days       21     0     0
#232072    --      20K      0K      0K               3     0     0
#62583     --       4K      0K      0K               2     0     0
#231072    --     878M      0K      0K           36214     0     0
#231073    --      64K      0K      0K               4     0     0
#231174    --     156K      0K      0K               4     0     0
#231176    --      24K      0K      0K               4     0     0
#231078    --    1272K      0K      0K             141     0     0
```



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).

The below screenshot is an example of using logger with tag "demo47", all other commands are entered in similar way.

```
user@user-VirtualBox:/$ sudo tune2fs -l /dev/sda1|logger -t demo47
user@user-VirtualBox:/$ cat /var/log/syslog | grep demo47
Apr 29 01:45:05 user-VirtualBox demo47: user
Apr 29 01:45:37 user-VirtualBox demo47: Linux user-VirtualBox 5.3.0-45-generic #37~18.04.1-Ubuntu SMP Fri M
10 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
Apr 29 02:02:51 user-VirtualBox demo47: tune2fs 1.44.1 (24-Mar-2018)
Apr 29 02:02:51 user-VirtualBox demo47: Filesystem volume name:   <none>
Apr 29 02:02:51 user-VirtualBox demo47: Last mounted on:          /
Apr 29 02:02:51 user-VirtualBox demo47: Filesystem UUID:          4acf8985-7b5d-4d43-a671-9743d734105c
Apr 29 02:02:51 user-VirtualBox demo47: Filesystem magic number:  0xEF53
Apr 29 02:02:51 user-VirtualBox demo47: Filesystem revision #:    1 (dynamic)
Apr 29 02:02:51 user-VirtualBox demo47: Filesystem features:      has_journal ext_attr resize_inode dir_ind
needs_recovery extent 64bit flex_bg sparse_super large_file huge_file dir_nlink extra_isize metadata_csum
Apr 29 02:02:51 user-VirtualBox demo47: Filesystem flags:         signed_directory_hash
Apr 29 02:02:51 user-VirtualBox demo47: Default mount options:    user_xattr acl
Apr 29 02:02:51 user-VirtualBox demo47: Filesystem state:         clean
Apr 29 02:02:51 user-VirtualBox demo47: Errors behavior:          Continue
Apr 29 02:02:51 user-VirtualBox demo47: Filesystem OS type:       Linux
Apr 29 02:02:51 user-VirtualBox demo47: Inode count:              1042432
Apr 29 02:02:51 user-VirtualBox demo47: Block count:              4163840
Apr 29 02:02:51 user-VirtualBox demo47: Reserved block count:     208192
Apr 29 02:02:51 user-VirtualBox demo47: Free blocks:              2038258
Apr 29 02:02:51 user-VirtualBox demo47: Free inodes:              783355
Apr 29 02:02:51 user-VirtualBox demo47: First block:              0
Apr 29 02:02:51 user-VirtualBox demo47: Block size:               4096
Apr 29 02:02:51 user-VirtualBox demo47: Fragment size:            4096
Apr 29 02:02:51 user-VirtualBox demo47: Group descriptor size:    64
Apr 29 02:02:51 user-VirtualBox demo47: Reserved GDT blocks:      1024
Apr 29 02:02:51 user-VirtualBox demo47: Blocks per group:         32768
Apr 29 02:02:51 user-VirtualBox demo47: Fragments per group:      32768
Apr 29 02:02:51 user-VirtualBox demo47: Inodes per group:         8144
Apr 29 02:02:51 user-VirtualBox demo47: Inode blocks per group:   509
```

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*

```
user@user-VirtualBox:/$ sudo tune2fs -l /dev/sda1
tune2fs 1.44.1 (24-Mar-2018)
Filesystem volume name:   <none>
Last mounted on:          /
Filesystem UUID:          4acf8985-7b5d-4d43-a671-9743d734105c
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype ne
se_super large_file huge_file dir_nlink extra_isize metadata_csum
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              1042432
Block count:              4163840
Reserved block count:     208192
Free blocks:              2038258
Free inodes:              783355
First block:              0
Block size:               4096
```

(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

```
user@user-VirtualBox:/tmp$ ls -ld /tmp/acltest/
drwxrwxrwx 2 user user 4096 кві 29 02:08 /tmp/acltest/
user@user-VirtualBox:/tmp$ ls -l /tmp/acltest/
total 0
-rw-rw-r-- 1 demo demo 0 кві 29 02:08 test.txt
user@user-VirtualBox:/tmp$ getfacl /tmp/acltest/
getfacl: Removing leading '/' from absolute path names
# file: tmp/acltest/
# owner: user
# group: user
user::rwx
group::rwx
other::rwx

user@user-VirtualBox:/tmp$ getfacl /tmp/acltest/test.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acltest/test.txt
# owner: demo
# group: demo
user::rw-
group::rw-
other::r--
```

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

```
user@user-VirtualBox:/tmp$ setfacl -m u:demo:r acltest/
user@user-VirtualBox:/tmp$ getfacl /tmp/acltest/
getfacl: Removing leading '/' from absolute path names
# file: tmp/acltest/
# owner: user
# group: user
user::rwx
user:demo:r--
group::rwx
mask::rwx
other::rwx
```

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*).

```
demo@user-VirtualBox:/tmp$ touch /tmp/acltest/prohibited.txt
touch: cannot touch '/tmp/acltest/prohibited.txt': Permission denied
demo@user-VirtualBox:/tmp$ echo 'blahblah' > /tmp/acltest/test.txt
-bash: /tmp/acltest/test.txt: Permission denied
demo@user-VirtualBox:/tmp$
```

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).

```
user@user-VirtualBox:/tmp$ setfacl -m u:demo:rwx acltest/
user@user-VirtualBox:/tmp$ getfacl /tmp/acltest/
getfacl: Removing leading '/' from absolute path names
# file: tmp/acltest/
# owner: user
# group: user
user::rwx
user:demo:rwx
group::rwx
mask::rwx
other::rwx

user@user-VirtualBox:/tmp$ sudo chmod o-rwx acltest/
user@user-VirtualBox:/tmp$ ls -ld /tmp/acltest/
drwxrwx---+ 2 user user 4096 кві 29 02:08 /tmp/acltest/
user@user-VirtualBox:/tmp$
```

```
demo@user-VirtualBox:/tmp$ touch /tmp/acltest/prohibited.txt
demo@user-VirtualBox:/tmp$ ls -lah acltest/
total 8,0K
drwxrwx---+  2 user user 4,0K кві 29 02:35 .
drwxrwxrwt  22 root root 4,0K кві 29 02:35 ..
-rw-rw-r--   1 demo demo    0 кві 29 02:35 prohibited.txt
-rw-rw-r--   1 demo demo    0 кві 29 02:08 test.txt
demo@user-VirtualBox:/tmp$ echo 'blahblah' > /tmp/acltest/test.txt
demo@user-VirtualBox:/tmp$ cat /tmp/acltest/test.txt
blahblah
demo@user-VirtualBox:/tmp$
```

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**.

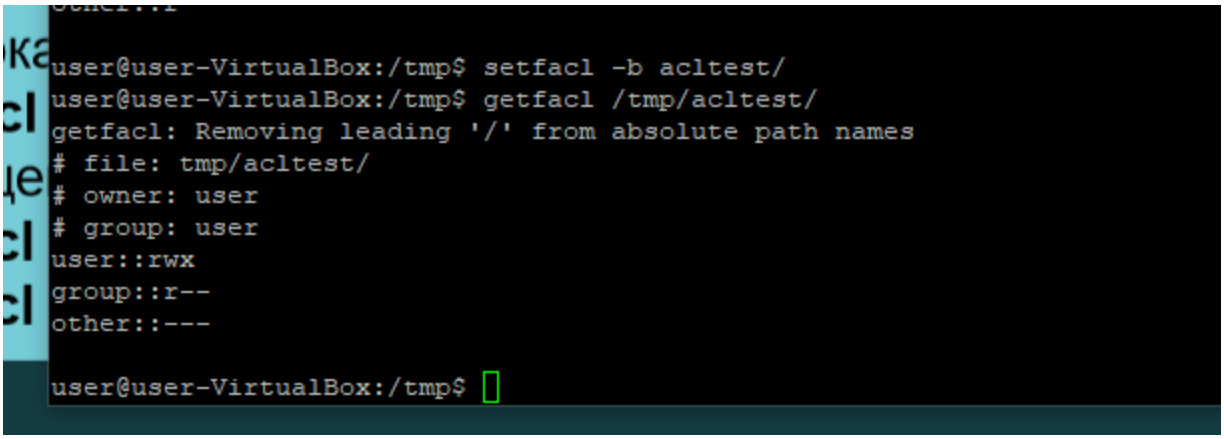
```
user@user-VirtualBox:/tmp$ setfacl -dm u:demo:r acltest/
user@user-VirtualBox:/tmp$ getfacl /tmp/acltest/
getfacl: Removing leading '/' from absolute path names
# file: tmp/acltest/
# owner: user
# group: user
user::rwx
user:demo:rwx
group::rwx
mask::rwx
other::---
default:user::rwx
default:user:demo:r--
default:group::rwx
default:mask::rwx
default:other::---
```

```
user@user-VirtualBox:/tmp$ getfacl /tmp/acltest/test2.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acltest/test2.txt
# owner: demo
# group: demo
user::rw-
user:demo:r--
group::rwx                      #effective:rw-
mask::rw-
other::---

user@user-VirtualBox:/tmp$ setfacl -m m::r acltest/
```

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**.

```
user@user-VirtualBox:/tmp$ sudo setfacl -m m::r acltest/test.txt
user@user-VirtualBox:/tmp$ getfacl /tmp/acltest/test.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acltest/test.txt
# owner: demo
# group: demo
user::rw-
group::rw-                      #effective:r--
mask::r--
other::r--

user@user-VirtualBox:/tmp$
```

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


user@user-VirtualBox:/tmp$ setfacl -b acltest/
user@user-VirtualBox:/tmp$ getfacl /tmp/acltest/
getfacl: Removing leading '/' from absolute path names
file: tmp/acltest/
owner: user
group: user
user::rwx
group::r--
other::---
user@user-VirtualBox:/tmp$