

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).

```
*** Report for user quotas on device /dev/sda1
Block grace time: 7days; Inode grace time: 7days
```

		Block limits				File limits			
User		used	soft	hard	grace	used	soft	hard	grace
root	--	7207456	0	0		217564	0	0	
man	--	1256	0	0		83	0	0	
systemd-network	--		12	0	0		3	0	0
syslog	--	2328	0	0		14	0	0	
_apt	--	28	0	0		4	0	0	
avahi-autoipd	--		4	0	0		1	0	0
dnsmasq	--	4	0	0		1	0	0	
speech-dispatcher	--		8	0	0		2	0	0
colord	--	56	0	0		5	0	0	
hplip	--	4	0	0		1	0	0	
geoclue	--	8	0	0		2	0	0	
gdm	--	232	0	0		41	0	0	
vmikern	--	63420	0	0		811	0	0	
utest	+-	150000	100000	150000	6days	17946	0	0	
#62583	--	4	0	0		2	0	0	

Left	File	Command	Options	Right
<- ~				<- /usr
.n	Name	Size	Modify time	.n Name Size Modify time
/..	UP--DIR	κBi 26	00:16	/.. UP--DIR κBi 26 01:15
/.cache	4096	κBi 26	02:02	/bin 49152 κBi 26 01:40
/.config	4096	κBi 26	02:02	/games 4096 лют 3 20:25
/.gnupg	4096	κBi 26	02:02	/include 4096 бер 29 22:51
/.local				
.bash_logout				κBi 26 01:40
.bashrc				лют 3 20:25
.profile				3 20:22
examples				26 00:27
				26 01:40
				21 23:04

Cannot write target file "/home/utest/sha~tart-panel.svg"
Disk quota exceeded (122)

[Skip] [Skip all] [Retry] [Abort]

Files processed: 10985/64447
Time: 0:00.20 ETA 0:01.37 (5,61 MB/s)

[kip] [Suspend] [Abort]

UP--DIR 11G/20G (58%) /share 0 B in 1 file 11G/20G (58%)

Hint: You may specify the editor for F4 with the shell variable EDITOR.
utest@vm1kern-VirtualBox:/usr\$

1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn10Quit

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](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).

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);

```
vmikern@vmikern-VirtualBox:~$ blkid
/dev/sr0: UUID="2020-02-18-17-20-05-35" LABEL="VBox_GAs_6.1.4" TYPE="iso
9660"
```

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

```
vmikern@vmikern-VirtualBox:~$ sudo blkid
/dev/loop0: TYPE="squashfs"
/dev/loop1: TYPE="squashfs"
/dev/loop2: TYPE="squashfs"
/dev/loop3: TYPE="squashfs"
/dev/loop4: TYPE="squashfs"
/dev/loop5: TYPE="squashfs"
/dev/loop6: TYPE="squashfs"
/dev/sr0: UUID="2020-02-18-17-20-05-35" LABEL="VBox_GAs_6.1.4" TYPE="iso
9660"
/dev/sda1: UUID="74f0345e-93ca-4bb9-9de8-40d28df9b873" TYPE="ext4" PARTU
UID="b5210a9d-01"
/dev/loop8: TYPE="squashfs"
```

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

```
vmikern@vmikern-VirtualBox:/tmp$ su utest
Password:
$ cd /tmp/acl_test
sh: 1: cd: can't cd to /tmp/acl_test
$ touch /tmp/acl_test/utest.txt
$ ls -ld /tmp/acl_test
drwxrwxrwx 2 vmikern vmikern 4096 Kbi 28 02:58 /tmp/acl_test
```

ls -l /tmp/acl_test

```
$ ls -l /tmp/acl_test
total 0
-rw-rw-r-- 1 utest utest 0 Kbi 28 02:58 utest.txt
```

To check ACL permissions do:

getfacl /tmp/acl_test

```
$ getfacl /tmp/acl_test
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test
# owner: vmikern
# group: vmikern
user::rwx
group::rwx
other::rwx
```

getfacl /tmp/acl_test/utest.txt

```
$ 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--
```

3. Employ ACL to block any activity except for reading, for user *utest* with respect to directory */tmp/acl_test*(hint: use **setfacl**).

```
$ getfacl /tmp/acl_test
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test
# owner: vmikern
# group: vmikern
user::rwx
user:utest:r--
group::rwx
mask::rwx
other::rwx
```

Test if the actions are effectively prohibited

touch /tmp/acl_test/prohibited.txt

Is it possible to invoke this command? -No, it's not.

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

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

```
$ echo "new content" > /tmp/acl_test/utest.txt
sh: 22: cannot create /tmp/acl_test/utest.txt: Permission denied
```

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*).

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

By the context of acl to the file *utest.txt* user *utest* is not prevented from modifying content of the file.

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).

```
d-----+ 2 vmikern vmikern 4096 kbi 28 03:07 acl_test
vmikern@vmikern-VirtualBox:/tmp$ setfacl -m u:utest:rwX /tmp/acl_test &&
logger setting acl rule for user utest to rwX the dir acl_test
```

touch */tmp/acl_test/prohibited.txt*

```
$ touch /tmp/acl_test/prohibited.txt
$ pwd
/home/utest
$ cd /tmp/acl_test/
$ ls -l
total 0
-rw-rw-r-- 1 utest utest 0 kbi 28 17:55 prohibited.txt
-rw-rw-r-- 1 utest utest 0 kbi 28 03:07 utest.txt
```

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

```
$ echo "new content" > /tmp/acl_test/utest.txt
$ cat ./utest.txt
new content
```

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**.

```
vm1kern@vm1kern-VirtualBox:/tmp$ getfacl acl_test
# file: acl_test
# owner: vm1kern
# group: vm1kern
user::---
user:utest:rwx
group::rwx
mask::rwx
other::---
default:user::---
default:user:utest:r--
default:group::rwx
default:mask::rwx
default:other::---

$ touch utest2.txt
$ getfacl utest2.txt
# file: utest2.txt
# owner: utest
# group: utest
user::---
user:utest:r--
group::rwx
mask::rw-
other::---
#effective:rw-
```

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**.

```
vm1kern@vm1kern-VirtualBox:/tmp$ sudo setfacl -m m:rwx /tmp/acl_test/utest.txt && logger setting rwx mask on utest.txt file as root user

$ getfacl utest.txt
# file: utest.txt
# owner: utest
# group: utest
user::rw-
group::rw-
other::r--

$ getfacl utest.txt
# file: utest.txt
# owner: utest
# group: utest
user::rw-
group::rw-
mask:rwx
other::r--
```

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

```
vm1kern@vm1kern-VirtualBox:/tmp$ getfacl ./acl_test
# file: acl_test
# owner: vm1kern
# group: vm1kern
user::---
group::rwx
other::---
```