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

Part1. **Quota allocation mechanism.**

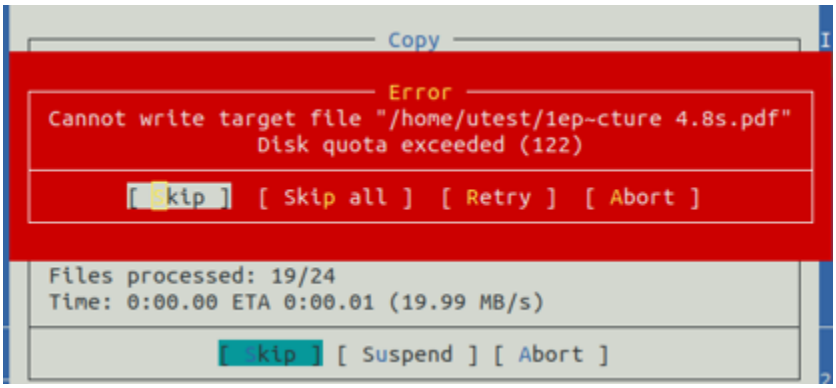
Employing commands from presentation #4.6, create a new user, say, *utest*. Based on the quota mechanism, limit the available disk space for this user to **soft**: 100M and **hard**: 150M.
Then, using Midnight Commander (since MC shows warnings about exceeding the limits of available to a user disk space),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

```
root@aku-ПК:~# groupadd testu
root@aku-ПК:~# useradd -g testu -s /bin/bash -d /home/testu -m testu
root@aku-ПК:~# passwd testu
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@aku-ПК:~#
```

apt install quota

```
sudo setquota -u utest 100M 150M 0 0 /
sudo quota -vs utest
```

```
Disk quotas for user utest (uid 1001):
  Filesystem                blocks       soft       hard     inodes     soft     hard
  /dev/sda1                     20     100000     150000          5        0        0
```



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).

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).

```
  GNU nano 2.9.3

# /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/8c2c6de1-c3db-4f0c-be52-2f7c5865af8a / ext4 defaults 0 0
/swap.img       none    swap    sw      0       0
```

```
admin1@checkout:~$ blkid
/dev/sda2: UUID="8c2c6de1-c3db-4f0c-be52-2f7c5865af8a" TYPE="ext4" PARTUUID="ca262c28-3db0-41e1-8bf3-c404fe353dc3"
admin1@checkout:~$
```

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

```
root@aku-ПК:/tmp# su guest
guest@aku-ПК:/tmp$ mkdir acl_test
guest@aku-ПК:/tmp$ chmod 777 acl_test/
guest@aku-ПК:/tmp$ ls -la
total 0
drwxrwxrwt 1 root   root   512 May  1 23:43 .
drwxr-xr-x 1 root   root   512 Jan  1  1970 ..
drwxrwxrwx 1 guest  testu  512 May  1 23:41 acl_test
drwx------ 1 felexa felexa 512 Mar  8 23:04 mc-felexa
```

```
guest@aku-ПК:/tmp$ su testu
Password:
testu@aku-ПК:/tmp$ cd /tmp/
testu@aku-ПК:/tmp$ cd acl_test/
testu@aku-ПК:/tmp/acl_test$ touch utest.txt
testu@aku-ПК:/tmp/acl_test$
```

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

```
testu@aku-ПК:/tmp/acl_test$ ls -ld /tmp/acl_test
drwxrwxrwx 1 guest testu 512 May  1 23:47 /tmp/acl_test
testu@aku-ПК:/tmp/acl_test$ ls -l /tmp/acl_test
total 0
-rw-rw-r-- 1 testu testu 0 May  1 23:47 utest.txt
testu@aku-ПК:/tmp/acl_test$ getfacl /tmp/acl_test
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test
# owner: guest
# group: testu
user::rwx
group::rwx
other::rwx

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

testu@aku-ПК:/tmp/acl_test$
```

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?

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*).

```
root@aku-ПК:/tmp# setfacl -m u:guest:r /tmp/acl_test/
root@aku-ПК:/tmp# getfacl /tmp/acl_test/utest.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/utest.txt
# owner: testu
# group: testu
user::rw-
group::rw-
other::r--
```

```
~$ touch /tmp/acl_test/prohibited.txt
p/acl_test/prohibited.txt': Permission denied
~$ echo "new content" > /tmp/acl_test/utest.txt
.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).

```
testu@aku-ПК:/home/felexa$ touch /tmp/acl_test/prohibited.txt
testu@aku-ПК:/home/felexa$ echo “new content” > /tmp/acl_test/utest.txt
testu@aku-ПК:/$ chmod 000 /tmp/acl_test/utest.txt
testu@aku-ПК:/$ ls -l /tmp/acl_test/
total 0
---------- 1 testu testu 18 May  2 01:56 utest.txt
testu@aku-ПК:/$ touch /tmp/acl_test/prohibited.txt
testu@aku-ПК:/$ echo “new content” > /tmp/acl_test/utest.txt
bash: /tmp/acl_test/utest.txt: Permission denied
testu@aku-ПК:/$
```

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**.

```
~$ setfacl -d -m u:utest:r- /tmp/acl_test
~$ 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:r--
default:group::rwx
default:mask::rwx
default:other::rwx
```

```
$ getfacl /tmp/acl_test | logger -t testacl2
getfacl: Removing leading '/' from absolute path names
```

```
$ su - utest
$ touch /tmp/acl_test/utest2.txt
$ getfacl /tmp/acl_test/utest2.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/acl_test/utest2.txt
# owner: utest
# group: utest
user::rw-
user:utest:r--
group::rwx                      #effective:rw-
mask::rw-
other::rw-
```

```
$ getfacl /tmp/acl_test/utest2.txt | logger -t acltest2
getfacl: Removing leading '/' from absolute path names
```

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**.

```
:/tmp$ cd acl_test/
:/tmp/acl_test$ sudo setfacl -m m::r utest.txt
:/tmp/acl_test$ getfacl utest.txt
```

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

```
/tmp$ sudo setfacl -b acl_test/
```