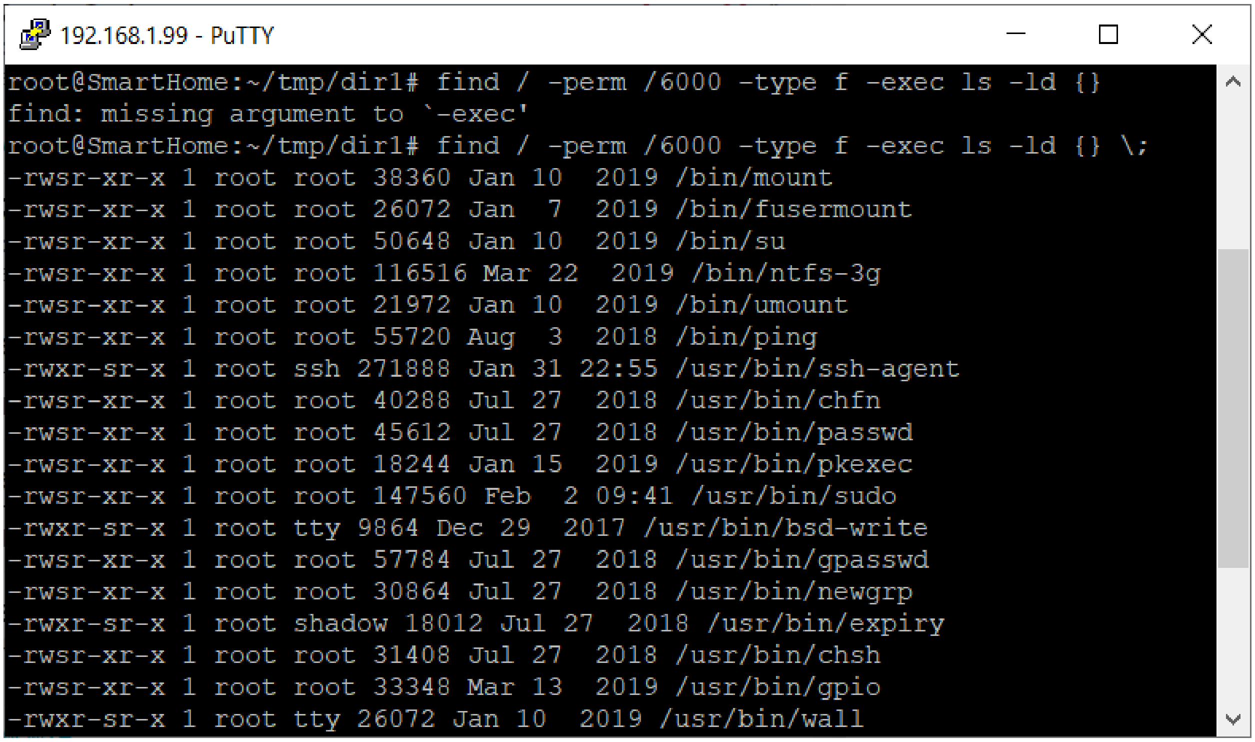
EPAM University Programs
DevOps external course
Module 4 Linux & Bash Essentials
TASK 4.5

1. To discover files with active sticky bits, use the following version of the **find** command:
sudo find / -perm /6000 -type f -exec ls -ld {} \;>setuid.txt



```
root@SmartHome:~/tmp/dir1# find / -perm /6000 -type f -exec ls -ld {}
find: missing argument to `-exec'
root@SmartHome:~/tmp/dir1# find / -perm /6000 -type f -exec ls -ld {} \;
-rwsr-xr-x 1 root root 38360 Jan 10  2019 /bin/mount
-rwsr-xr-x 1 root root 26072 Jan  7  2019 /bin/fusermount
-rwsr-xr-x 1 root root 50648 Jan 10  2019 /bin/su
-rwsr-xr-x 1 root root 116516 Mar 22  2019 /bin/ntfs-3g
-rwsr-xr-x 1 root root 21972 Jan 10  2019 /bin/umount
-rwsr-xr-x 1 root root 55720 Aug  3  2018 /bin/ping
-rwxr-sr-x 1 root ssh 271888 Jan 31 22:55 /usr/bin/ssh-agent
-rwsr-xr-x 1 root root 40288 Jul 27  2018 /usr/bin/chfn
-rwsr-xr-x 1 root root 45612 Jul 27  2018 /usr/bin/passwd
-rwsr-xr-x 1 root root 18244 Jan 15  2019 /usr/bin/pkexec
-rwsr-xr-x 1 root root 147560 Feb  2 09:41 /usr/bin/sudo
-rwxr-sr-x 1 root tty 9864 Dec 29  2017 /usr/bin/bsd-write
-rwsr-xr-x 1 root root 57784 Jul 27  2018 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 30864 Jul 27  2018 /usr/bin/newgrp
-rwxr-sr-x 1 root shadow 18012 Jul 27  2018 /usr/bin/expiry
-rwsr-xr-x 1 root root 31408 Jul 27  2018 /usr/bin/chsh
-rwsr-xr-x 1 root root 33348 Mar 13  2019 /usr/bin/gpio
-rwxr-sr-x 1 root tty 26072 Jan 10  2019 /usr/bin/wall
```

Put into your report a fragment of setuid.txt file. Explain meaning of parameters of the above **find** command (hint: use find’s man page).

Well, “**find** / -perm /6000 -type f -exec ls -ld {} \;” means:
try to find files **(-type -f**) with sticky bit permissions **(/ -perm /6000**) starting from root dir (**/**) and for each of finding item execute ls -ld command (-**exec ls -ld {} \;**)

2. Discovering soft and hard links.
Comment on results of these commands (place the output into your report):
cd (change dir to $home dir)
mkdir test(create test dir in current dir)

cd test (change dir to test)

touch test1.txt (create empty file test1.txt)

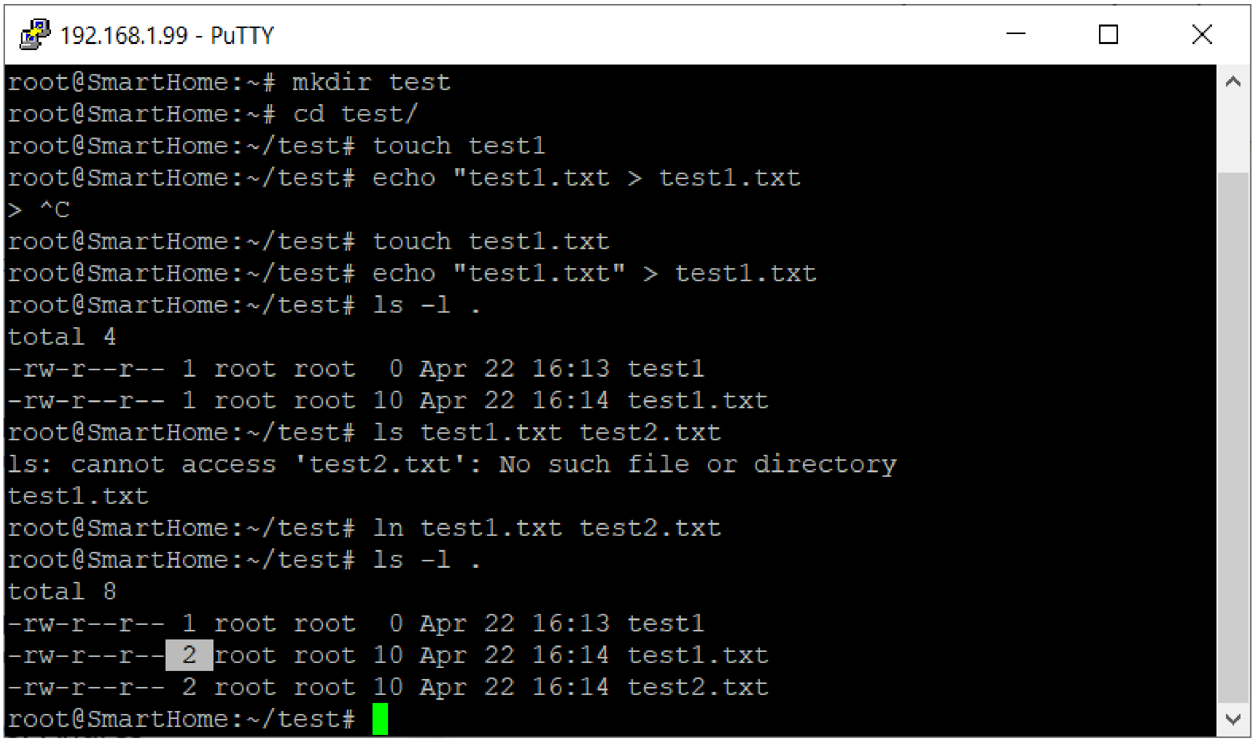
echo “test1.txt” > test1.txt (overwrite by string test1.txt   test1.txt file)

ls -l . (extended list of current directory)

(a hard link)

ln test1.txt test2.txt (create hard link)

ls -l . (pay attention to column $2 - changed from 1 to 2 for hard links)



```
root@SmartHome:~# mkdir test
root@SmartHome:~# cd test/
root@SmartHome:~/test# touch test1
root@SmartHome:~/test# echo "test1.txt > test1.txt
> ^C
root@SmartHome:~/test# touch test1.txt
root@SmartHome:~/test# echo "test1.txt" > test1.txt
root@SmartHome:~/test# ls -l .
total 4
-rw-r--r-- 1 root root  0 Apr 22 16:13 test1
-rw-r--r-- 1 root root 10 Apr 22 16:14 test1.txt
root@SmartHome:~/test# ls test1.txt test2.txt
ls: cannot access 'test2.txt': No such file or directory
test1.txt
root@SmartHome:~/test# ln test1.txt test2.txt
root@SmartHome:~/test# ls -l .
total 8
-rw-r--r-- 1 root root  0 Apr 22 16:13 test1
-rw-r--r-- 2 root root 10 Apr 22 16:14 test1.txt
-rw-r--r-- 2 root root 10 Apr 22 16:14 test2.txt
root@SmartHome:~/test#
```

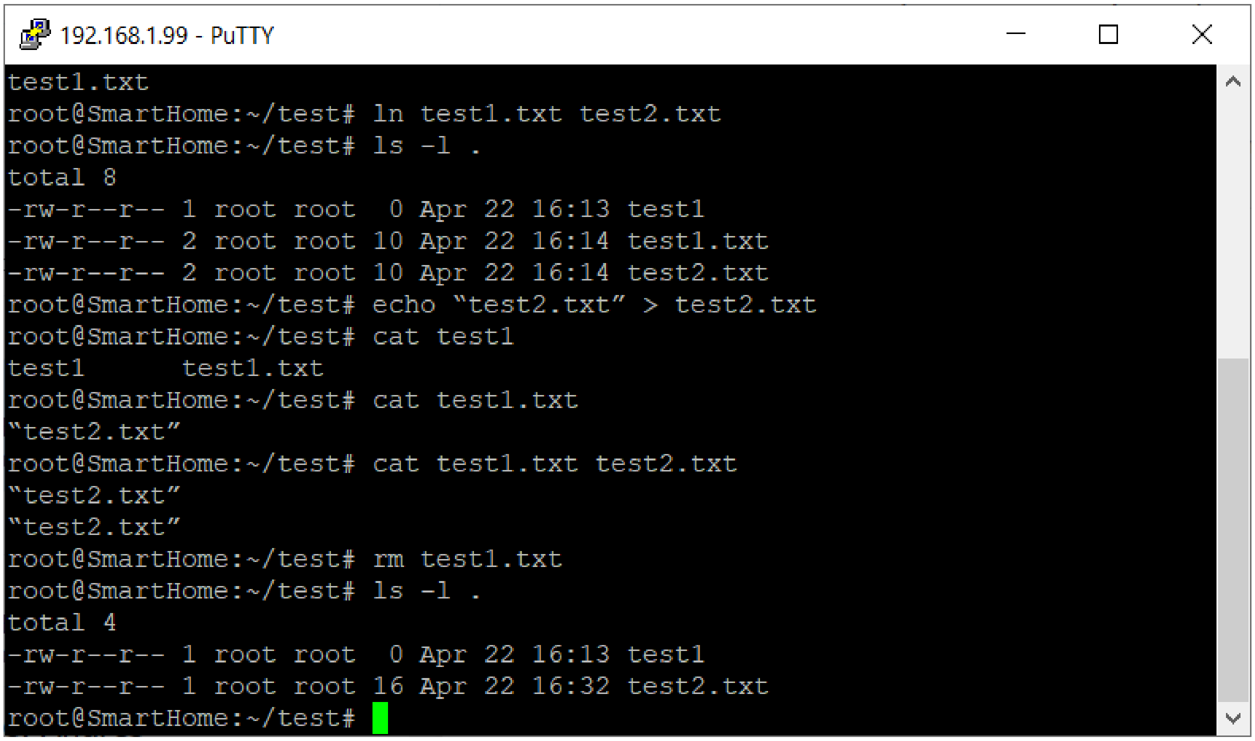
(pay attention to the number of links to test1.txt and test2.txt)

echo “test2.txt” > test2.txt (overwrite by string test2.txt   test2.txt file (hard link)

cat test1.txt test2.txt (test1.txt also overwritten because of hard link)

rm test1.txt (rm file 1.txt)

ls -l . (now test2.txt is one link only)



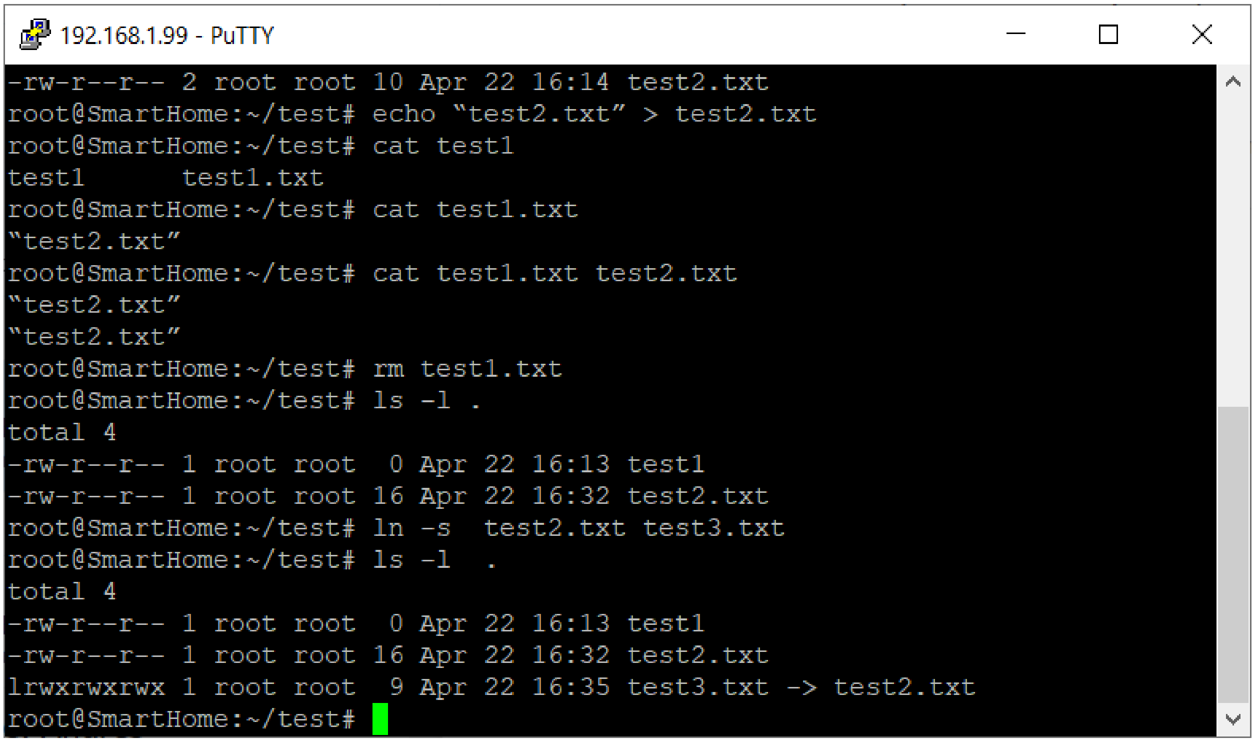
```
test1.txt
root@SmartHome:~/test# ln test1.txt test2.txt
root@SmartHome:~/test# ls -l .
total 8
-rw-r--r-- 1 root root  0 Apr 22 16:13 test1
-rw-r--r-- 2 root root 10 Apr 22 16:14 test1.txt
-rw-r--r-- 2 root root 10 Apr 22 16:14 test2.txt
root@SmartHome:~/test# echo “test2.txt” > test2.txt
root@SmartHome:~/test# cat test1
test1      test1.txt
root@SmartHome:~/test# cat test1.txt
“test2.txt”
root@SmartHome:~/test# cat test1.txt test2.txt
“test2.txt”
“test2.txt”
root@SmartHome:~/test# rm test1.txt
root@SmartHome:~/test# ls -l .
total 4
-rw-r--r-- 1 root root  0 Apr 22 16:13 test1
-rw-r--r-- 1 root root 16 Apr 22 16:32 test2.txt
root@SmartHome:~/test#
```

(now a soft link)

ln -s test2.txt test3.txt (create symbol link test3.txt to file text2.txt)

ls -l .

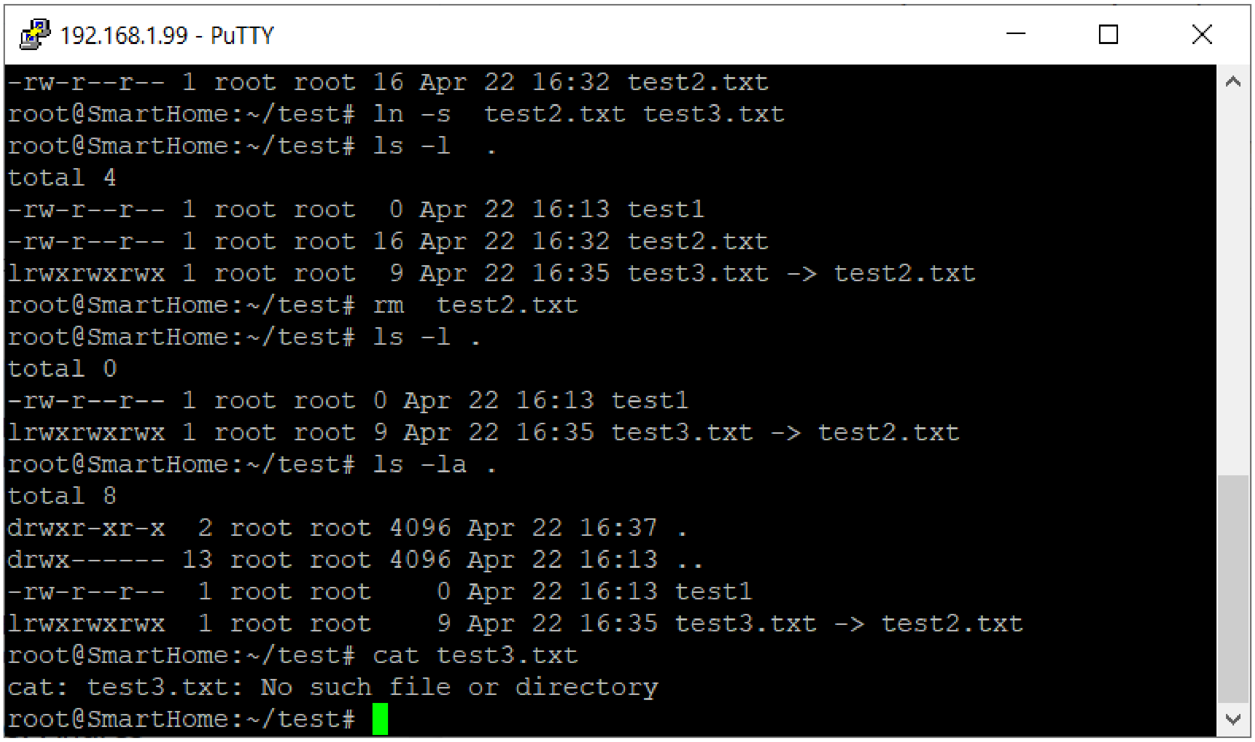
(pay attention to the number of links to the created files)



```
-rw-r--r-- 2 root root 10 Apr 22 16:14 test2.txt
root@SmartHome:~/test# echo “test2.txt” > test2.txt
root@SmartHome:~/test# cat test1
test1      test1.txt
root@SmartHome:~/test# cat test1.txt
“test2.txt”
root@SmartHome:~/test# cat test1.txt test2.txt
“test2.txt”
“test2.txt”
root@SmartHome:~/test# rm test1.txt
root@SmartHome:~/test# ls -l .
total 4
-rw-r--r-- 1 root root  0 Apr 22 16:13 test1
-rw-r--r-- 1 root root 16 Apr 22 16:32 test2.txt
root@SmartHome:~/test# ln -s  test2.txt test3.txt
root@SmartHome:~/test# ls -l  .
total 4
-rw-r--r-- 1 root root  0 Apr 22 16:13 test1
-rw-r--r-- 1 root root 16 Apr 22 16:32 test2.txt
lrwxrwxrwx 1 root root  9 Apr 22 16:35 test3.txt -> test2.txt
root@SmartHome:~/test#
```

rm test2.txt; **ls** -l .

bl



```
-rw-r--r-- 1 root root 16 Apr 22 16:32 test2.txt
root@SmartHome:~/test# ln -s  test2.txt test3.txt
root@SmartHome:~/test# ls -l  .
total 4
-rw-r--r-- 1 root root  0 Apr 22 16:13 test1
-rw-r--r-- 1 root root 16 Apr 22 16:32 test2.txt
lrwxrwxrwx 1 root root  9 Apr 22 16:35 test3.txt -> test2.txt
root@SmartHome:~/test# rm  test2.txt
root@SmartHome:~/test# ls -l .
total 0
-rw-r--r-- 1 root root 0 Apr 22 16:13 test1
lrwxrwxrwx 1 root root 9 Apr 22 16:35 test3.txt -> test2.txt
root@SmartHome:~/test# ls -la .
total 8
drwxr-xr-x  2 root root 4096 Apr 22 16:37 .
drwx------ 13 root root 4096 Apr 22 16:13 ..
-rw-r--r--  1 root root    0 Apr 22 16:13 test1
lrwxrwxrwx  1 root root    9 Apr 22 16:35 test3.txt -> test2.txt
root@SmartHome:~/test# cat test3.txt
cat: test3.txt: No such file or directory
root@SmartHome:~/test#
```

Link is present but original file test2.txt is missed

3. I/O redirect.

Execute these commands; comment on the output.

mount

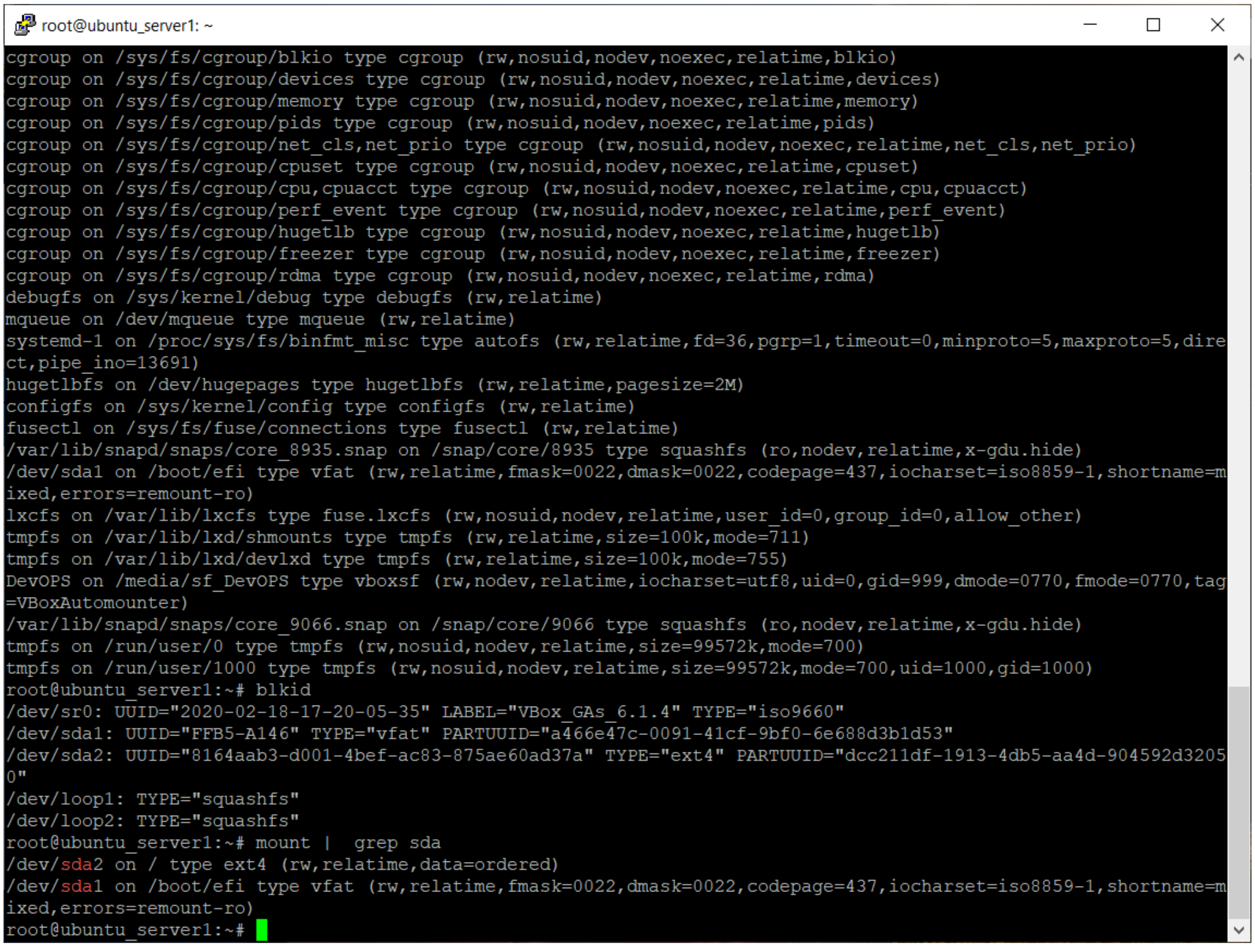
(list of mounting FS)

blkid

Display UUID for each of FS)

mount | **grep** sda

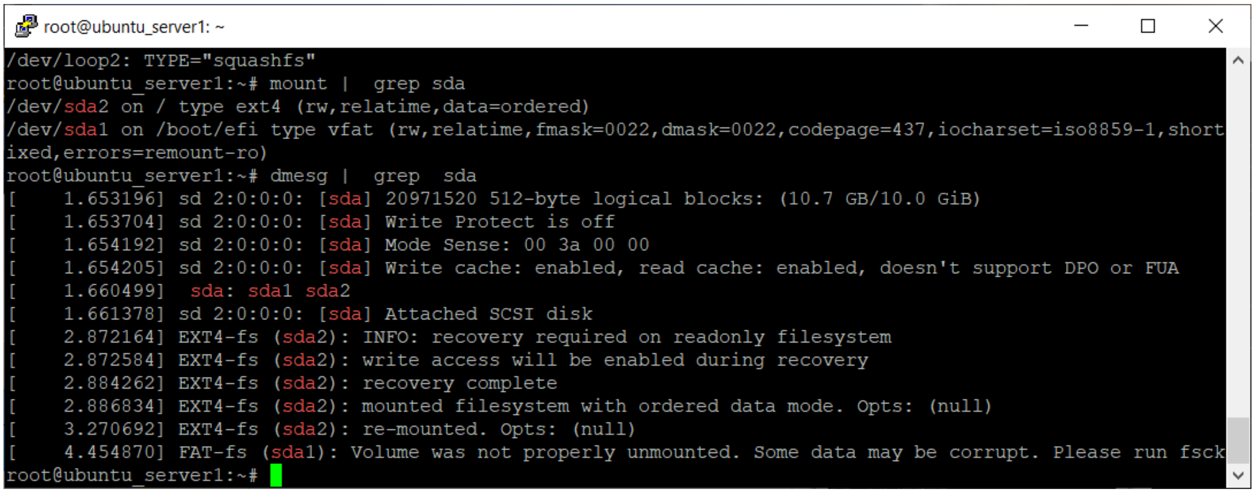
(display real disk partions)



```
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
mqueue on /dev/mqueue type mqueue (rw,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=36,pgrp=1,timeout=0,minproto=5,maxproto=5,dire
ct,pipe_ino=13691)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
configfs on /sys/kernel/config type configfs (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
/var/lib/snapd/snaps/core_8935.snap on /snap/core/8935 type squashfs (ro,nodev,relatime,x-gdu.hide)
/dev/sda1 on /boot/efi type vfat (rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=m
ixed,errors=remount-ro)
lxcfs on /var/lib/lxcfs type fuse.lxcfs (rw,nosuid,nodev,relatime,user_id=0,group_id=0,allow_other)
tmpfs on /var/lib/lxd/shmounts type tmpfs (rw,relatime,size=100k,mode=711)
tmpfs on /var/lib/lxd/devlxd type tmpfs (rw,relatime,size=100k,mode=755)
DevOPS on /media/sf_DevOPS type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=0,gid=999,dmode=0770,fmode=0770,tag
=VBoxAutomounter)
/var/lib/snapd/snaps/core_9066.snap on /snap/core/9066 type squashfs (ro,nodev,relatime,x-gdu.hide)
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,relatime,size=99572k,mode=700)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=99572k,mode=700,uid=1000,gid=1000)
root@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-904592d3205
0"
/dev/loop1: TYPE="squashfs"
/dev/loop2: TYPE="squashfs"
root@ubuntu_server1:~# mount |  grep sda
/dev/sda2 on / type ext4 (rw,relatime,data=ordered)
/dev/sda1 on /boot/efi type vfat (rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=m
ixed,errors=remount-ro)
root@ubuntu_server1:~#
```

dmesg | **grep** sda

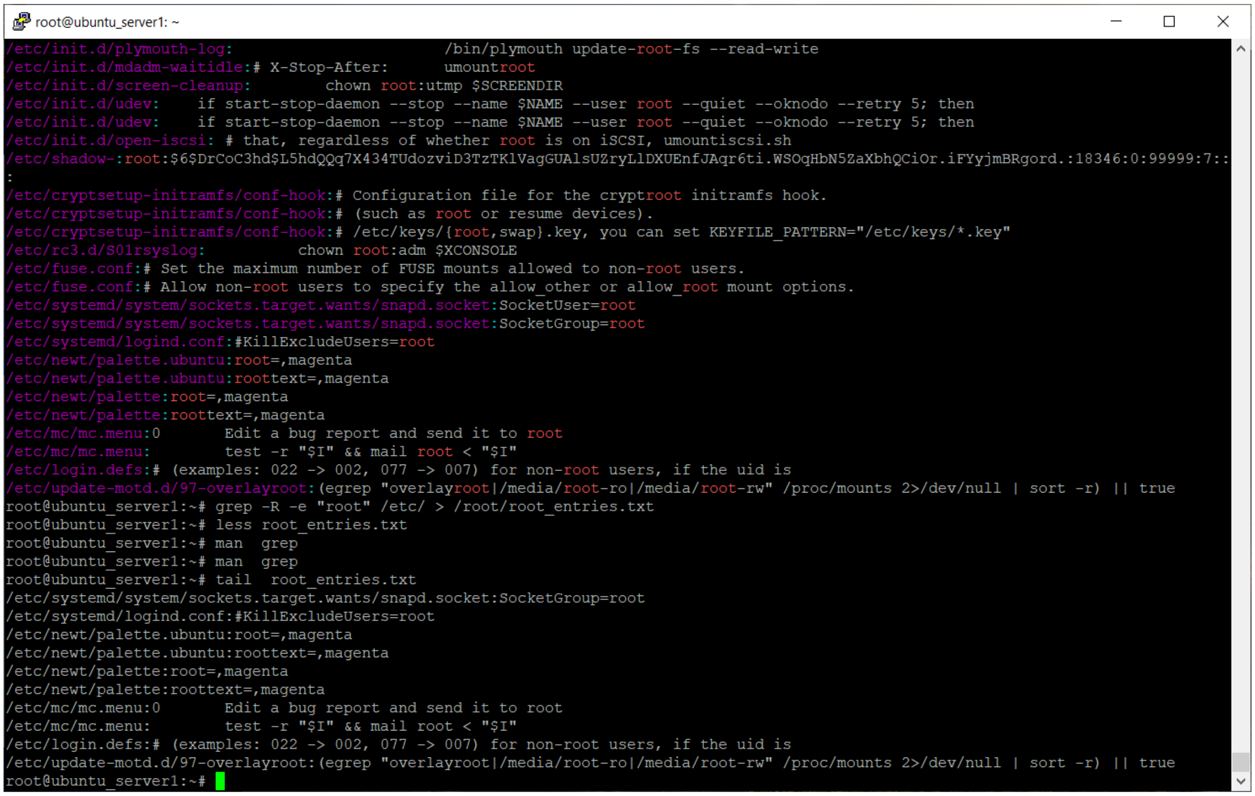
(Disk devices find and mount process during Linux boot)



```
/dev/loop2: TYPE="squashfs"
root@ubuntu_server1:~# mount |  grep sda
/dev/sda2 on / type ext4 (rw,relatime,data=ordered)
/dev/sda1 on /boot/efi type vfat (rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,short
ixed,errors=remount-ro)
root@ubuntu_server1:~# dmesg |  grep  sda
[    1.653196] sd 2:0:0:0: [sda] 20971520 512-byte logical blocks: (10.7 GB/10.0 GiB)
[    1.653704] sd 2:0:0:0: [sda] Write Protect is off
[    1.654192] sd 2:0:0:0: [sda] Mode Sense: 00 3a 00 00
[    1.654205] sd 2:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[    1.660499]  sda: sda1 sda2
[    1.661378] sd 2:0:0:0: [sda] Attached SCSI disk
[    2.872164] EXT4-fs (sda2): INFO: recovery required on readonly filesystem
[    2.872584] EXT4-fs (sda2): write access will be enabled during recovery
[    2.884262] EXT4-fs (sda2): recovery complete
[    2.886834] EXT4-fs (sda2): mounted filesystem with ordered data mode. Opts: (null)
[    3.270692] EXT4-fs (sda2): re-mounted. Opts: (null)
[    4.454870] FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck
root@ubuntu_server1:~#
```

sudo grep -R -e “root” /etc > root_entries.txt

command means – find in all files of /etc directory (-R – recursively) word “root” (-e means redexp PATTERN)



```
/etc/init.d/plymouth-log:                       /bin/plymouth update-root-fs --read-write
/etc/init.d/mdadm-waitidle:# X-Stop-After:      umountroot
/etc/init.d/screen-cleanup:         chown root:utmp $SCREENDIR
/etc/init.d/udev:    if start-stop-daemon --stop --name $NAME --user root --quiet --oknodo --retry 5; then
/etc/init.d/udev:    if start-stop-daemon --stop --name $NAME --user root --quiet --oknodo --retry 5; then
/etc/init.d/open-iscsi: # that, regardless of whether root is on iSCSI, umountiscsi.sh
/etc/shadow-:root:$6$DrCoC3hd$L5hdQQq7X434TUdozviD3TzTKlVagGUAlsUZryLlDXUEnfJAqr6ti.WSOqHbN5ZaXbhQCiOr.iFYyjmBRgord.:18346:0:99999:7::
:
/etc/cryptsetup-initramfs/conf-hook:# Configuration file for the cryptroot initramfs hook.
/etc/cryptsetup-initramfs/conf-hook:# (such as root or resume devices).
/etc/cryptsetup-initramfs/conf-hook:# /etc/keys/{root,swap}.key, you can set KEYFILE_PATTERN="/etc/keys/*.key"
/etc/rc3.d/S01rsyslog:          chown root:adm $XCONSOLE
/etc/fuse.conf:# Set the maximum number of FUSE mounts allowed to non-root users.
/etc/fuse.conf:# Allow non-root users to specify the allow_other or allow_root mount options.
/etc/systemd/system/sockets.target.wants/snapd.socket:SocketUser=root
/etc/systemd/system/sockets.target.wants/snapd.socket:SocketGroup=root
/etc/systemd/logind.conf:#KillExcludeUsers=root
/etc/newt/palette.ubuntu:root=,magenta
/etc/newt/palette.ubuntu:roottext=,magenta
/etc/newt/palette:root=,magenta
/etc/newt/palette:roottext=,magenta
/etc/mc/mc.menu:0       Edit a bug report and send it to root
/etc/mc/mc.menu:        test -r "$I" && mail root < "$I"
/etc/login.defs:# (examples: 022 -> 002, 077 -> 007) for non-root users, if the uid is
/etc/update-motd.d/97-overlayroot:(egrep "overlayroot|/media/root-ro|/media/root-rw" /proc/mounts 2>/dev/null | sort -r) || true
root@ubuntu_server1:~# grep -R -e "root" /etc/ > /root/root_entries.txt
root@ubuntu_server1:~# less root_entries.txt
root@ubuntu_server1:~# man  grep
root@ubuntu_server1:~# man  grep
root@ubuntu_server1:~# tail  root_entries.txt
/etc/systemd/system/sockets.target.wants/snapd.socket:SocketGroup=root
/etc/systemd/logind.conf:#KillExcludeUsers=root
/etc/newt/palette.ubuntu:root=,magenta
/etc/newt/palette.ubuntu:roottext=,magenta
/etc/newt/palette:root=,magenta
/etc/newt/palette:roottext=,magenta
/etc/mc/mc.menu:0       Edit a bug report and send it to root
/etc/mc/mc.menu:        test -r "$I" && mail root < "$I"
/etc/login.defs:# (examples: 022 -> 002, 077 -> 007) for non-root users, if the uid is
/etc/update-motd.d/97-overlayroot:(egrep "overlayroot|/media/root-ro|/media/root-rw" /proc/mounts 2>/dev/null | sort -r) || true
root@ubuntu_server1:~#
```