IST Lab 01: File Systems

Authors: Kevin Auberson and Paul Gillet

Group: L1GrG

Date: February 29, 2024

TASK 1: EXPLORE BLOCK DEVICES AND FILESYSTEMS

> 1.1 On which block device is the boot partition mounted?

The boot partition is mounted on the sda2

> 1.2 On which block device is the root (/) partition mounted? What is the name of its special file?

The following command gives us this result :

```
$ ls -l /dev/sda3

brw-rw---- 1 root disk 8, 3 Mär  1 10:52 /dev/sda3
```

> 1.3 With hdparm -t do a timing test on the boot partition. What throughput do you get?

This is the throughput of the command :

```
$ sudo hdparm -t /dev/sda2

/dev/sda2:
 Timing buffered disk reads: 512 MB in  2.57 seconds = 199.45 MB/sec
```

> 2. Now open the special file with the same command. You may pipe its output into less. What do you see?

Running the command will display the hexadecimal representation of the content of the block device.

> 3. As the special file represents all the blocks of a partition, the content of all files of the root partition should be there. Pick a text file at random (for example a file in

> /usr/share/doc/) and try to find its content in the special file

A creation of a text file was created in the folder /home/kevin/Documents with the content "bonjour test".

```
$ sudo strings /dev/sda3 | grep "bonjour test"

text/plainbonjour test
bonjour test
```

With this command the content of this file is found in the root partition.

TASK 2: PREPARE AND PARTITION A DISK

> 2. List again the block devices. Which new block devices and special files appeared? These represent the disk and its partitions you just attached.

```
$ findmnt --real
TARGET                                  SOURCE      FSTYPE OPTIONS
/                                       /dev/sda3   ext4   rw,relatime,errors
├─/run/user/1000/doc                    portal      fuse.p rw,nosuid,nodev,re
├─/snap/bare/5                          /dev/loop0  squash ro,nodev,relatime,
├─/snap/core20/2182                     /dev/loop2  squash ro,nodev,relatime,
├─/snap/core20/1974                     /dev/loop1  squash ro,nodev,relatime,
├─/snap/gnome-3-38-2004/143             /dev/loop3  squash ro,nodev,relatime,
├─/snap/snapd/20671                     /dev/loop4  squash ro,nodev,relatime,
├─/snap/gnome-42-2204/141               /dev/loop7  squash ro,nodev,relatime,
├─/snap/firefox/2987                    /dev/loop5  squash ro,nodev,relatime,
├─/snap/core22/858                      /dev/loop6  squash ro,nodev,relatime,
├─/snap/gnome-42-2204/120               /dev/loop11 squash ro,nodev,relatime,
├─/snap/snapd/19457                     /dev/loop10 squash ro,nodev,relatime,
├─/var/snap/firefox/common/host-hunspell /dev/sda3[/usr/share/hunspell]
│                                                   ext4   ro,noexec,noatime,
├─/snap/gtk-common-themes/1535          /dev/loop12 squash ro,nodev,relatime,
├─/snap/snap-store/959                  /dev/loop8  squash ro,nodev,relatime,
├─/snap/snapd-desktop-integration/83    /dev/loop9  squash ro,nodev,relatime,
├─/boot/efi                             /dev/sda2   vfat   rw,relatime,fmask=
├─/snap/firefox/3836                    /dev/loop13 squash ro,nodev,relatime,
└─/media/kevin/FE0A-E642                /dev/sdb1   vfat   rw,nosuid,nodev,re
```

The new block that appeared and the file is "sdb1".

> Display the existing partitions with the print command. If the disk is completely blank you will get an error message about a missing disk label.

This is the result of the command print on the USB key

```
(parted) print
Model: Generic Flash Disk (scsi)
Disk /dev/sdb: 31.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      65.5kB  31.5GB  31.5GB  primary  fat32        boot, lba
```

> Display the free space with the command print free (roughly the size of the disk minus some overhead). Write the value down.

Before displaying the free space I remove the default partition of the key with the command rm NUMBER.

This is the result of the command print free.

```
(parted) print free
Model: Unknown (unknown)
Disk /dev/sdb1: 31.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start  End     Size    Type  File system  Flags
        1024B  31.5GB  31.5GB        Free Space
```

> Use the mkpart command to create the partitions.
>
> - The first partition will

```
- be a primary partition
- have a file system type of fat32
- start at 0
- end at about half the free space.
```

> - The second partition will

```
- be a primary partition
- have a file system type of ext4
- start at half the free space
- end at the free space.
```

This is the command to create the partitions.

```
(parted) mkpart primary fat32 0% 50%
(parted) mkpart primary ext4 50% 100%
(parted) print
Model: Generic Flash Disk (scsi)
Disk /dev/sdb: 31.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  15.7GB  15.7GB  primary  fat32        lba
 2      15.7GB  31.5GB  15.7GB  primary  ext4         lba
```

> Quit parted and verify that there are now two special files in /dev that correspond to the two partitions.

The two special files in /dev are created.

```
kevin@kevin-VirtualBox:/dev$ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0    7:0    0     4K  1 loop /snap/bare/5
loop1    7:1    0  63.4M  1 loop /snap/core20/1974
loop2    7:2    0 266.6M  1 loop /snap/firefox/3836
loop3    7:3    0 237.2M  1 loop /snap/firefox/2987
```

```
loop4    7:4    0  63.9M  1 loop /snap/core20/2182
loop5    7:5    0 349.7M  1 loop /snap/gnome-3-38-2004/143
loop6    7:6    0  73.9M  1 loop /snap/core22/858
loop7    7:7    0   497M  1 loop /snap/gnome-42-2204/141
loop8    7:8    0 485.5M  1 loop /snap/gnome-42-2204/120
loop9    7:9    0  91.7M  1 loop /snap/gtk-common-themes/1535
loop10   7:10   0  53.3M  1 loop /snap/snapd/19457
loop11   7:11   0  12.3M  1 loop /snap/snap-store/959
loop12   7:12   0  40.4M  1 loop /snap/snapd/20671
loop13   7:13   0   452K  1 loop /snap/snapd-desktop-integration/83
sda      8:0    0    25G  0 disk
├─sda1   8:1    0     1M  0 part
├─sda2   8:2    0   513M  0 part /boot/efi
└─sda3   8:3    0  24.5G  0 part /var/snap/firefox/common/host-hunspell
                                 /
sdb      8:16   1  29.3G  0 disk
├─sdb1   8:17   1  14.6G  0 part
└─sdb2   8:18   1  14.6G  0 part
sr0     11:0    1  1024M  0 rom
```

> Format the two partitions using the mkfs command.
>
> - The first partition should have the file system type vfat.
> - The second partition should have the file system type ext4

```
kevin@kevin-VirtualBox:/dev$ sudo mkfs.vfat /dev/sdb1
mkfs.fat 4.2 (2021-01-31)
kevin@kevin-VirtualBox:/dev$ sudo mkfs.ext4 /dev/sdb2
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 3840000 4k blocks and 960992 inodes
Filesystem UUID: 81bd6894-fae1-4dc9-8bcc-36cfee546957
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

> 5.Create two empty directories in the /mnt directory as mount points, called part1 and part2. Mount the newly created file systems in these directories.

```
kevin@kevin-VirtualBox:/mnt$ sudo mkdir /mnt/part1
kevin@kevin-VirtualBox:/mnt$ sudo mkdir /mnt/part2
kevin@kevin-VirtualBox:/mnt$ sudo mount /dev/sdb1 /mnt/part1
kevin@kevin-VirtualBox:/mnt$ sudo mount /dev/sdb2 /mnt/part2
```

> 6.How much free space is available on these filesystems? Use the df command to find out. What does the -h option do?

```
kevin@kevin-VirtualBox:/mnt$ df -h /mnt/part1 /mnt/part2
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb1        15G  8.0K   15G   1% /mnt/part1
/dev/sdb2        15G   24K   14G   1% /mnt/part2
```

There is 15GB on the first partition and 14GB on the second partition.
The option -h stands for "human-readable" and formats the output in a more readable format, showing sizes in kilobytes, megabytes, gigabytes, etc., rather than in blocks.

TASK 3: EXPLORE THE FILE SYSTEM SUPPORT IN THE KERNEL

> List the content of /proc. What is the version of the kernel in /proc/version?

```
kevin@kevin-VirtualBox:/proc$ ls -l
...
dr-xr-xr-x  3 root             root                           0 Mär  3 12:11 acpi
dr-xr-xr-x 12 root             root                           0 Mär  3 12:11 asound
-r--r--r--  1 root             root                           0 Mär  3 12:11
bootconfig
-r--r--r--  1 root             root                           0 Mär  3 12:11
buddyinfo
dr-xr-xr-x  4 root             root                           0 Mär  3 12:11 bus
-r--r--r--  1 root             root                           0 Mär  3 12:11
cgroups
-r--r--r--  1 root             root                         105 Mär  3 12:11
cmdline
-r--r--r--  1 root             root                           0 Mär  3 12:11
consoles
```

```
-r--r--r--  1 root             root                           0 Mär  3 12:11
cpuinfo
-r--r--r--  1 root             root                           0 Mär  3 12:11 crypto
-r--r--r--  1 root             root                           0 Mär  3 12:11
devices
-r--r--r--  1 root             root                           0 Mär  3 12:11
diskstats
-r--r--r--  1 root             root                           0 Mär  3 12:11 dma
dr-xr-xr-x  3 root             root                           0 Mär  3 12:11 driver
dr-xr-xr-x  3 root             root                           0 Mär  3 12:11
dynamic_debug
-r--r--r--  1 root             root                           0 Mär  3 12:11
execdomains
-r--r--r--  1 root             root                           0 Mär  3 12:11 fb
-r--r--r--  1 root             root                           0 Mär  3 12:11
filesystems
dr-xr-xr-x  5 root             root                           0 Mär  3 12:11 fs
-r--r--r--  1 root             root                           0 Mär  3 12:11
interrupts
-r--r--r--  1 root             root                           0 Mär  3 12:11 iomem
-r--r--r--  1 root             root                           0 Mär  3 12:11
ioports
dr-xr-xr-x 24 root             root                           0 Mär  3 12:11 irq
-r--r--r--  1 root             root                           0 Mär  3 12:11
kallsyms
-r--------  1 root             root              140737471590400 Mär  3 12:11 kcore
-r--r--r--  1 root             root                           0 Mär  3 12:11 keys
-r--r--r--  1 root             root                           0 Mär  3 12:11 key-
users
-r--------  1 root             root                           0 Mär  3 12:11 kmsg
-r--------  1 root             root                           0 Mär  3 12:11
kpagecgroup
-r--------  1 root             root                           0 Mär  3 12:11
kpagecount
-r--------  1 root             root                           0 Mär  3 12:11
kpageflags
-r--r--r--  1 root             root                           0 Mär  3 12:11
loadavg
-r--r--r--  1 root             root                           0 Mär  3 12:11 locks
-r--r--r--  1 root             root                           0 Mär  3 12:11 mdstat
```

```
-r--r--r--  1 root             root                           0 Mär  3 12:11
meminfo
-r--r--r--  1 root             root                           0 Mär  3 12:11 misc
-r--r--r--  1 root             root                           0 Mär  3 12:11
modules
lrwxrwxrwx  1 root             root                          11 Mär  3 12:11 mounts
-> self/mounts
-rw-r--r--  1 root             root                           0 Mär  3 12:11 mtrr
lrwxrwxrwx  1 root             root                           8 Mär  3 12:11 net ->
self/net
-r--------  1 root             root                           0 Mär  3 12:11
pagetypeinfo
-r--r--r--  1 root             root                           0 Mär  3 12:11
partitions
dr-xr-xr-x  5 root             root                           0 Mär  3 12:11
pressure
-r--r--r--  1 root             root                           0 Mär  3 12:11
schedstat
dr-xr-xr-x  6 root             root                           0 Mär  3 12:11 scsi
lrwxrwxrwx  1 root             root                           0 Mär  3 12:11 self -
> 1863
-r--------  1 root             root                           0 Mär  3 12:11
slabinfo
-r--r--r--  1 root             root                           0 Mär  3 12:11
softirqs
-r--r--r--  1 root             root                           0 Mär  3 12:11 stat
-r--r--r--  1 root             root                           0 Mär  3 12:11 swaps
dr-xr-xr-x  1 root             root                           0 Mär  3 12:11 sys
--w-------  1 root             root                           0 Mär  3 12:11 sysrq-
trigger
dr-xr-xr-x  5 root             root                           0 Mär  3 12:11
sysvipc
lrwxrwxrwx  1 root             root                           0 Mär  3 12:11
thread-self -> 1863/task/1863
-r--------  1 root             root                           0 Mär  3 12:11
timer_list
dr-xr-xr-x  6 root             root                           0 Mär  3 12:11 tty
-r--r--r--  1 root             root                           0 Mär  3 12:11 uptime
-r--r--r--  1 root             root                           0 Mär  3 12:11
version
```

```
-r--r--r--  1 root             root                           0 Mär  3 12:11
version_signature
-r--------  1 root             root                           0 Mär  3 12:11
vmallocinfo
-r--r--r--  1 root             root                           0 Mär  3 12:11 vmstat
-r--r--r--  1 root             root                           0 Mär  3 12:11
zoneinfo
k

kevin@kevin-VirtualBox:/proc$ nano version

Linux version 6.5.0-21-generic (buildd@lcy02-amd64-091) (x86_64-linux-gnu-gcc-12
(Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0, GNU ld (GNU Binutils for Ubuntu) 2.38)
#21~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Feb  9 13:32:52 UTC 2
```

> The directories with numbers represent the running processes. The numbers are the process ids. Display the process id of your bash session with echo $$. List the information in the corresponding directory. What was the command line that started this process (look in cmdline)?

```
kevin@kevin-VirtualBox:/proc$ echo $$
1774
kevin@kevin-VirtualBox:/proc$ cd /proc/1774
kevin@kevin-VirtualBox:/proc/1774$ ls -l
total 0
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 arch_status
dr-xr-xr-x  2 kevin kevin 0 Mär  3 12:41 attr
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 autogroup
-r--------  1 kevin kevin 0 Mär  3 12:41 auxv
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 cgroup
--w-------  1 kevin kevin 0 Mär  3 12:41 clear_refs
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 cmdline
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 comm
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 coredump_filter
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 cpu_resctrl_groups
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 cpuset
lrwxrwxrwx  1 kevin kevin 0 Mär  3 12:41 cwd -> /proc/1774
-r--------  1 kevin kevin 0 Mär  3 12:41 environ
lrwxrwxrwx  1 kevin kevin 0 Mär  3 12:41 exe -> /usr/bin/bash
dr-x------  2 kevin kevin 4 Mär  3 12:41 fd
```

```
dr-xr-xr-x  2 kevin kevin 0 Mär  3 12:41 fdinfo
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 gid_map
-r--------  1 kevin kevin 0 Mär  3 12:41 io
-r--------  1 kevin kevin 0 Mär  3 12:41 ksm_merging_pages
-r--------  1 kevin kevin 0 Mär  3 12:41 ksm_stat
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 limits
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 loginuid
dr-x------  2 kevin kevin 0 Mär  3 12:41 map_files
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 maps
-rw-------  1 kevin kevin 0 Mär  3 12:41 mem
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 mountinfo
-r--r--r--  1 kevin kevin 0 Mär  3 12:23 mounts
-r--------  1 kevin kevin 0 Mär  3 12:41 mountstats
dr-xr-xr-x 54 kevin kevin 0 Mär  3 12:41 net
dr-x--x--x  2 kevin kevin 0 Mär  3 12:41 ns
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 numa_maps
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 oom_adj
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 oom_score
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 oom_score_adj
-r--------  1 kevin kevin 0 Mär  3 12:41 pagemap
-r--------  1 kevin kevin 0 Mär  3 12:41 patch_state
-r--------  1 kevin kevin 0 Mär  3 12:41 personality
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 projid_map
lrwxrwxrwx  1 kevin kevin 0 Mär  3 12:41 root -> /
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 sched
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 schedstat
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 sessionid
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 setgroups
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 smaps
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 smaps_rollup
-r--------  1 kevin kevin 0 Mär  3 12:41 stack
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 stat
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 statm
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 status
-r--------  1 kevin kevin 0 Mär  3 12:41 syscall
dr-xr-xr-x  3 kevin kevin 0 Mär  3 12:41 task
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 timens_offsets
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 timers
-rw-rw-rw-  1 kevin kevin 0 Mär  3 12:41 timerslack_ns
```

```
-rw-r--r--  1 kevin kevin 0 Mär  3 12:41 uid_map
-r--r--r--  1 kevin kevin 0 Mär  3 12:41 wchan
```

BOOT_IMAGE=/boot/vmlinuz-6.5.0-21-generic root=UUID=c2d51f86-e667-4492-87f2-c2dc1ace3b16 ro quiet splash

> The kernel lists the file systems it supports right now file filesystems. List them.

```
$ cat /proc/filesystems
nodev   sysfs
nodev   tmpfs
nodev   bdev
nodev   proc
nodev   cgroup
nodev   cgroup2
nodev   cpuset
nodev   devtmpfs
nodev   configfs
nodev   debugfs
nodev   tracefs
nodev   securityfs
nodev   sockfs
nodev   bpf
nodev   pipefs
nodev   ramfs
nodev   hugetlbfs
nodev   devpts
        ext3
        ext2
        ext4
        squashfs
        vfat
nodev   ecryptfs
        fuseblk
nodev   fuse
nodev   fusectl
nodev   efivarfs
nodev   mqueue
nodev   pstore
```

```
nodev   autofs
nodev   binfmt_misc
```

> Can you find the proc filesystem itself in the list? How is it tagged? All file systems with that tag are pseudo file systems.

Yes the proc filesystem is itself in the file tagget "nodev proc"

> List the real (non-pseudo) file systems

The non-pseudo file systems are those who don't have the tag "nodev".

- ext3
- ext2
- ext4
- squashfs
- vfat
- fuseblk

> Find out which file systems the kernel is able to support by looking at the available kernel modules. The files containing kernel modules can be found at lib/modules//kernel/fs. List them.

The result of the listing :

```
kevin@kevin-VirtualBox:/lib/modules/6.5.0-21-generic/kernel/fs$ ls -l
total 264
drwxr-xr-x 2 root root  4096 Mär  2 13:32 9p
drwxr-xr-x 2 root root  4096 Mär  2 13:32 adfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 affs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 afs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 autofs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 befs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 bfs
-rw-r--r-- 1 root root 40153 Feb  9 11:46 binfmt_misc.ko
drwxr-xr-x 2 root root  4096 Mär  2 13:32 btrfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 cachefiles
drwxr-xr-x 2 root root  4096 Mär  2 13:32 ceph
drwxr-xr-x 2 root root  4096 Mär  2 13:32 coda
drwxr-xr-x 2 root root  4096 Mär  2 13:32 cramfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 dlm
```

```
drwxr-xr-x 2 root root  4096 Mär  2 13:32 efs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 erofs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 exfat
drwxr-xr-x 2 root root  4096 Mär  2 13:32 f2fs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 fat
drwxr-xr-x 2 root root  4096 Mär  2 13:32 freevxfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 fscache
drwxr-xr-x 2 root root  4096 Mär  2 13:32 fuse
drwxr-xr-x 2 root root  4096 Mär  2 13:32 gfs2
drwxr-xr-x 2 root root  4096 Mär  2 13:32 hfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 hfsplus
drwxr-xr-x 2 root root  4096 Mär  2 13:32 hpfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 isofs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 jffs2
drwxr-xr-x 2 root root  4096 Mär  2 13:32 jfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 lockd
drwxr-xr-x 2 root root  4096 Mär  2 13:32 minix
drwxr-xr-x 2 root root  4096 Mär  2 13:32 netfs
drwxr-xr-x 5 root root  4096 Mär  2 13:32 nfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 nfs_common
drwxr-xr-x 2 root root  4096 Mär  2 13:32 nfsd
drwxr-xr-x 2 root root  4096 Mär  2 13:32 nilfs2
drwxr-xr-x 2 root root  4096 Mär  2 13:32 nls
drwxr-xr-x 2 root root  4096 Mär  2 13:32 ntfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 ntfs3
drwxr-xr-x 5 root root  4096 Mär  2 13:32 ocfs2
drwxr-xr-x 2 root root  4096 Mär  2 13:32 omfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 orangefs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 overlayfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 pstore
drwxr-xr-x 2 root root  4096 Mär  2 13:32 qnx4
drwxr-xr-x 2 root root  4096 Mär  2 13:32 qnx6
drwxr-xr-x 2 root root  4096 Mär  2 13:32 quota
drwxr-xr-x 2 root root  4096 Mär  2 13:32 reiserfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 romfs
drwxr-xr-x 5 root root  4096 Mär  2 13:32 smb
drwxr-xr-x 2 root root  4096 Mär  2 13:32 sysv
drwxr-xr-x 2 root root  4096 Mär  2 13:32 ubifs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 udf
drwxr-xr-x 2 root root  4096 Mär  2 13:32 ufs
```

```
drwxr-xr-x 2 root root  4096 Mär  2 13:32 vboxsf
drwxr-xr-x 2 root root  4096 Mär  2 13:32 xfs
drwxr-xr-x 2 root root  4096 Mär  2 13:32 zonefs
```

> When a new disk is inserted the kernel knows which file system to activate by looking at a label that indicates the type of file system. That label is part of the partition metadata (called signature). Use the blkid command to list the metadata of all known partitions (mounted or not). Note that you might need to run the command with admin permissions to display all partitions metadata.

Verify that the partitions you created are labeled correctly.

```
kevin@kevin-VirtualBox:/lib/modules/6.5.0-21-generic/kernel/fs$ sudo blkid
/dev/sda3: UUID="c2d51f86-e667-4492-87f2-c2dc1ace3b16" BLOCK_SIZE="4096"
TYPE="ext4" PARTUUID="3c988cc2-0a27-4eb0-8812-8ad10c061ee3"
/dev/loop1: TYPE="squashfs"
/dev/loop8: TYPE="squashfs"
/dev/loop6: TYPE="squashfs"
/dev/loop13: TYPE="squashfs"
/dev/loop4: TYPE="squashfs"
/dev/loop11: TYPE="squashfs"
/dev/loop2: TYPE="squashfs"
/dev/loop0: TYPE="squashfs"
/dev/loop9: TYPE="squashfs"
/dev/loop7: TYPE="squashfs"
/dev/sda2: UUID="C0D1-57F5" BLOCK_SIZE="512" TYPE="vfat" PARTLABEL="EFI System
Partition" PARTUUID="6992fdbe-abce-4ee3-a356-2fdd5dc9787d"
/dev/loop5: TYPE="squashfs"
/dev/loop12: TYPE="squashfs"
/dev/loop3: TYPE="squashfs"
/dev/loop10: TYPE="squashfs"
/dev/sdb2: UUID="7f1cb47b-d3ae-48f0-9df0-6f191eb28c29" BLOCK_SIZE="4096"
TYPE="ext4" PARTUUID="78ec729c-02"
/dev/sdb1: UUID="75D6-F01C" BLOCK_SIZE="512" TYPE="vfat" PARTUUID="78ec729c-01"
/dev/sda1: PARTUUID="b85a73f3-1cbf-4605-963f-45c562da0f6a"
```

> There is another piece of information in the partition metadata. What does it do?

```
As we can see the partitions have both UUIDs and and PARTUUIDs. The UUID
(Universally Unique Identifier) and PARTUUID (Partition UUID) are identifiers
```

```
assigned to partitions.
```

> List the content of /etc/fstab. What line is responsible for mounting the root (/) file system? This line has a particular way of referencing the partition, how?

The first line(without comment) of this file is responsible for the mounting the root file system.

```
# /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/sda3 during installation
UUID=c2d51f86-e667-4492-87f2-c2dc1ace3b16 /               ext4    errors=remount-ro
0       1
# /boot/efi was on /dev/sda2 during installation
UUID=C0D1-57F5  /boot/efi       vfat    umask=0077      0       1
/swapfile                                 none            swap    sw              0
0
```

As we can see here, it uses the UUID to identify the correct partitions. This allow the devices names to change while the UUID never changes.

TASK 4: MANAGE AN EXT4 PARTITION

> 1.Unmount the ext4 partition on the external disk.

```
$ sudo umount /mtn/part2
```

> 2.Run a file system check using the fsck command.

```
$ sudo fsck /dev/sdb2
fsck from util-linux 2.37.2
e2fsck 1.46.5 (30-Dec-2021)
/dev/sdb2: clean, 11/960992 files, 86958/3840000 blocks
```

> 3.Add the -f option to the command to force a complete verification.

```
$ sudo fsck -f /dev/sdb2
fsck from util-linux 2.37.2
e2fsck 1.46.5 (30-Dec-2021)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdb2: 11/960992 files (0.0% non-contiguous), 86958/3840000 blocks
```

> 4.Display the file system structure with the dumpe2fs command. How many inodes are unused?

Free inodes: 960981

```
$ sudo dumpe2fs /dev/sdb2
dumpe2fs 1.46.5 (30-Dec-2021)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          7f1cb47b-d3ae-48f0-9df0-6f191eb28c29
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype
extent 64bit flex_bg sparse_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:              960992
Block count:              3840000
Reserved block count:     192000
Overhead clusters:        86952
Free blocks:              3753042
Free inodes:              960981
First block:              0
```

```
Block size:               4096
Fragment size:            4096
Group descriptor size:    64
Reserved GDT blocks:      1024
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8144
Inode blocks per group:   509
Flex block group size:    16
Filesystem created:       Sun Mar  3 14:15:01 2024
Last mount time:          Sun Mar  3 15:03:37 2024
Last write time:          Sun Mar  3 16:40:53 2024
Mount count:              0
Maximum mount count:      -1
Last checked:             Sun Mar  3 16:40:53 2024
Check interval:           0 (<none>)
Lifetime writes:          8 MB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               256
Required extra isize:     32
Desired extra isize:      32
Journal inode:            8
Default directory hash:   half_md4
Directory Hash Seed:      f8592b40-d8d8-420e-b3fb-9f6433b31fc6
Journal backup:           inode blocks
Checksum type:            crc32c
Checksum:                 0x6ff314d0
Journal features:         journal_64bit journal_checksum_v3
Total journal size:       64M
Total journal blocks:     16384
Max transaction length:   16384
Fast commit length:       0
Journal sequence:         0x00000005
Journal start:            0
Journal checksum type:    crc32c
Journal checksum:         0x6db0b898
...
```

5.Intentionally corrupt the file system by overwriting 4 MB of data, starting 10 kB in:

```
$ sudo dd if=/dev/sdb2 of=/dev/sdb2 bs=1k seek=10 count=4k
4096+0 records in
4096+0 records out
4194304 bytes (4.2 MB, 4.0 MiB) copied, 1.89617 s, 2.2 MB/s
```

6.Try to mount the partition. You should get an error message. Repair the file system with the fsck command.

```
sudo mount /dev/sdb2 /mnt/part2
mount: /mnt/part2: mount(2) system call failed: Structure needs cleaning.

sudo fsck /dev/sdb2
...
/dev/sdb2: ***** FILE SYSTEM WAS MODIFIED *****
/dev/sdb2: 20/960992 files (10.0% non-contiguous), 86963/3840000 blocks
```

7.Mount the repaired partition.

```
sudo mount /dev/sdb2 /mnt/part2

$ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0    7:0    0     4K  1 loop /snap/bare/5
loop1    7:1    0  63.4M  1 loop /snap/core20/1974
loop2    7:2    0  63.9M  1 loop /snap/core20/2182
loop3    7:3    0  73.9M  1 loop /snap/core22/858
loop4    7:4    0 266.6M  1 loop /snap/firefox/3836
loop5    7:5    0 349.7M  1 loop /snap/gnome-3-38-2004/143
loop6    7:6    0 485.5M  1 loop /snap/gnome-42-2204/120
loop7    7:7    0   497M  1 loop /snap/gnome-42-2204/141
loop8    7:8    0 237.2M  1 loop /snap/firefox/2987
loop9    7:9    0  12.3M  1 loop /snap/snap-store/959
loop10   7:10   0  53.3M  1 loop /snap/snapd/19457
loop11   7:11   0  40.4M  1 loop /snap/snapd/20671
loop12   7:12   0   452K  1 loop /snap/snapd-desktop-integration/83
loop13   7:13   0  91.7M  1 loop /snap/gtk-common-themes/1535
sda      8:0    0    25G  0 disk
```

```
├─sda1   8:1    0     1M  0 part
├─sda2   8:2    0   513M  0 part /boot/efi
└─sda3   8:3    0  24.5G  0 part /var/snap/firefox/common/host-hunspell
                                 /
sdb      8:16   1  29.3G  0 disk
├─sdb1   8:17   1  14.6G  0 part /mnt/part1
└─sdb2   8:18   1  14.6G  0 part /mnt/part2
sr0     11:0    1  1024M  0 rom
```

TASK 5: CREATE A FILE SYSTEM IN A FILE

1. Create a 100 MB file using dd:

```
$ dd if=/dev/zero of=/tmp/bigfile  bs=1M count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 0.155997 s, 672 MB/s
```

2. Find the next available loopback device:

```
$ losetup -f
/dev/loop8
```

4. Verify that the association is OK:

```
$ losetup -a
/dev/loop1: []: (/var/lib/snapd/snaps/core22_1122.snap)
/dev/loop8: []: (/tmp/bigfile)
/dev/loop6: []: (/var/lib/snapd/snaps/snapd_20671.snap)
/dev/loop4: []: (/var/lib/snapd/snaps/gtk-common-themes_1535.snap)
/dev/loop2: []: (/var/lib/snapd/snaps/firefox_3836.snap)
/dev/loop0: []: (/var/lib/snapd/snaps/bare_5.snap)
/dev/loop7: []: (/var/lib/snapd/snaps/snapd-desktop-integration_83.snap)
/dev/loop5: []: (/var/lib/snapd/snaps/snap-store_959.snap)
/dev/loop3: []: (/var/lib/snapd/snaps/gnome-42-2204_141.snap)
```

5. Create an ext4 file system on block device /dev/loop6. Create a mountpoint in /mnt/bigfile. Mount the file system on the mountpoint. How does findmnt show the new file system?

```
$ sudo mkfs.ext4 /dev/loop8
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 25600 4k blocks and 25600 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

$ sudo mkdir /mnt/bigfile
$ sudo mount /dev/loop8 /mnt/bigfile/

$ findmnt /mnt/bigfile
TARGET       SOURCE     FSTYPE OPTIONS
/mnt/bigfile /dev/loop8 ext4   rw,relatime
```

6. Create a few files in the file system with unique strings. By searching the content of bigfile, can you find the strings? Use the sync command to force the kernel to write buffered data to disk.

```
$ echo "Hello test for file1" | sudo tee /mnt/bigfile/file1.txt

$ echo "Hello unique test for file2" | sudo tee /mnt/bigfile/file2.txt

$ sudo grep -r "unique" /mnt/bigfile/
/mnt/bigfile/file2.txt:Hello unique test for file2

$ sync

$ sudo grep -r "unique" /mnt/bigfile/
/mnt/bigfile/file2.txt:Hello unique test for file2
```

We could find the file containing our unique string even without using the sync command, once the sync command was done the result didn't change

7. Undo everything:
   7.1 Unmount the file system

```
$ sudo umount /mnt/bigfile
```

> 7.2 Free the loopback device with losetup -d /dev/loop6 and verify with losetup -a.

```
$ sudo losetup -d /dev/loop8

$ losetup -a
/dev/loop1: []: (/var/lib/snapd/snaps/core22_1122.snap)
/dev/loop6: []: (/var/lib/snapd/snaps/snapd_20671.snap)
/dev/loop4: []: (/var/lib/snapd/snaps/gtk-common-themes_1535.snap)
/dev/loop2: []: (/var/lib/snapd/snaps/firefox_3836.snap)
/dev/loop0: []: (/var/lib/snapd/snaps/bare_5.snap)
/dev/loop7: []: (/var/lib/snapd/snaps/snapd-desktop-integration_83.snap)
/dev/loop5: []: (/var/lib/snapd/snaps/snap-store_959.snap)
/dev/loop3: []: (/var/lib/snapd/snaps/gnome-42-2204_141.snap)
```