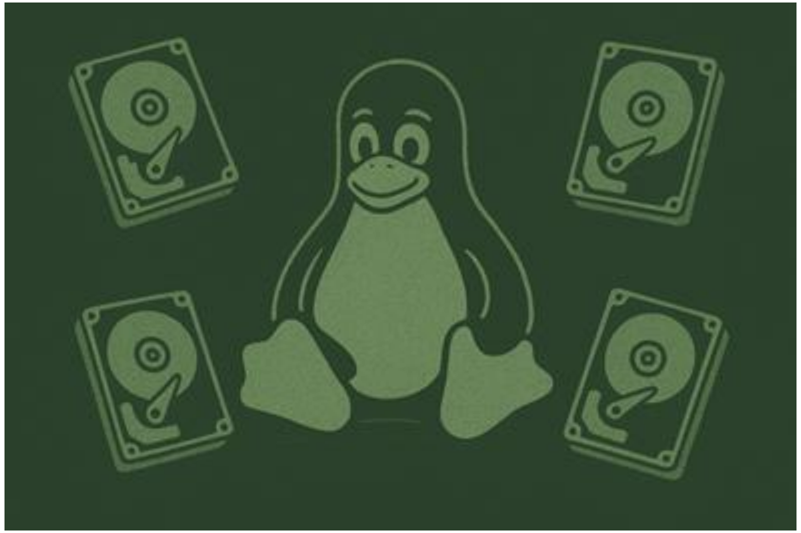
Operating System fundamentals

Disk- and partition management



Karel de Grote Hogeschool

Contents

Course text

- Chapter 16 Disk and partition management
  - (RedHat chapter 13)
  - Access Linux File Systems
  - Mount and Unmount File Systems

Block devices

Karel de Grote Hogeschool

Block devices: examples

Karel de Grote Hogeschool

Block devices

- consists of “blocks” of data (e.g. 512 bytes)
- appears in /dev (the udev daemon creates this)
- examples
  - SATA/SCSI disk -> /dev/sda, /dev/sdb, /dev/sdc, …
  - SSD disk -> /dev/nvme0n1, /dev/nvme0n2, …
  - virtual storage -> /dev/vda, /dev/vdb, …
  - SD cards -> /dev/mmcblk0, /dev/mmcblk1, …
- ls -l /dev/nvme0n1

```
brw-rw----   1 root disk      259,     0 nov 11 15:00 /dev/nvme0n1
```

Block devices

- You can see which block devices are connected to the computer with **lsblk**

```
[ user@server ~ ] $ lsblk
NAME          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
nvme0n1       259:0    0 953,9G  0 disk
├─nvme0n1p1 259:1    0   500M  0 part /boot/efi
├─nvme0n1p2 259:2    0   128M  0 part
├─nvme0n1p3 259:3    0   309G  0 part
└─nvme0n1p4 259:4    0   9,5G  0 part
```

Karel de Grote Hogeschool

Exercise

- what block devices are connected to your system?
- stop the virtual machine
- add a small hard disk to your headless server
  - 128 MiB
- restart the headless VM
- can you find the new hard disk?
- can you put data on it?

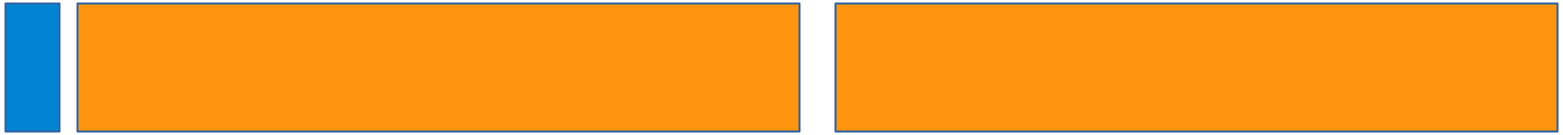
Partitions

2

Karel de Grote Hogeschool

Partitions

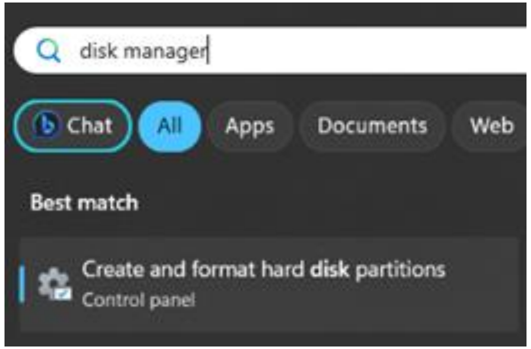
- A disk is often divided into partitions
- For example:
  - partition1 -> boot and base OS
  - partition2 -> home directories
  - parititon3 -> rest of the system and log files
- Information on partitions is stored in the MBR or GUID

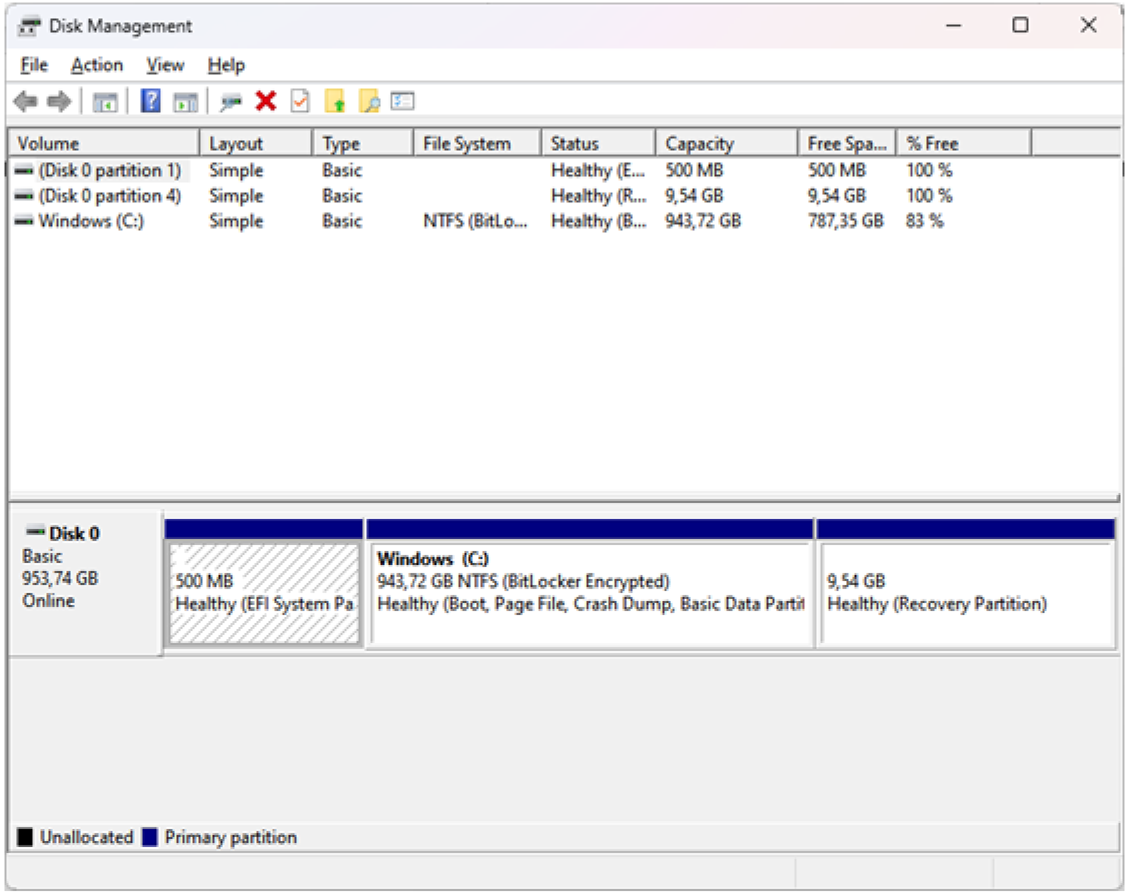


- Partitions are also block devices and get an extra number
  - e.g.: /dev/nvme0n1p1, /dev/nvme0n2p5, /dev/sda2, …

Windows Partitions

Disk Manager





Volume	Layout	Type	File System	Status	Capacity	Free Spa...	% Free
(Disk 0 partition 1)	Simple	Basic		Healthy (E...	500 MB	500 MB	100 %
(Disk 0 partition 4)	Simple	Basic		Healthy (R...	9,54 GB	9,54 GB	100 %
Windows (C:)	Simple	Basic	NTFS (BitLo...	Healthy (B...	943,72 GB	787,35 GB	83 %

Disk 0
Basic
953,74 GB
Online

500 MB
Healthy (EFI System Pa

Windows (C:)
943,72 GB NTFS (BitLocker Encrypted)
Healthy (Boot, Page File, Crash Dump, Basic Data Partit

9,54 GB
Healthy (Recovery Partition)

■ Unallocated ■ Primary partition

Modify partitions

You can add and remove partitions with **fdisk** (MBR) or **gdisk** (GUID)

```
[ student@server ~ ] $ sudo fdisk /dev/sda
Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

This disk is currently in use - repartitioning is probably a bad idea.
It's recommended to umount all file systems, and swapoff all swap
partitions on this disk.

Command (m for help): m

...
```

Karel de Grote Hogeschool

Exercise

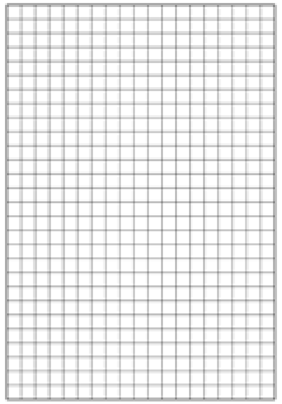
- Create 2 MBR primary partitions on the new hard disk in your VM
  - 20 MiB
  - the rest of the hard disk
- Can you find these partitions in /dev?
- Can you put data on these partitions?

File systems

Karel de Grote Hogeschool

File systems analogy

- A disk is like a binder
- Partitions are made using organizer sheets
- The blocks are empty pages with room for e.g. 512 characters



File systems

- A file system creates a kind of table of contents on the block device
- The table of contents contains for each file
  - name
  - length in bytes
  - permissions
  - timestamps (modified, created, …)
  - blocks where the file is stored
  - …
- The command **filefrag -v** shows where the file is stored
- A file is always stored in an integer amount of blocks
- The command **du** shows how much space a file takes on the disk

Karel de Grote Hogeschool

Example

```
[student@localhost ~]$ sudo filefrag -v /var/log/boot.log
Filesystem type is: 58465342
File size of /var/log/boot.log is 35286 (9 blocks of 4096 bytes)
 ext:     logical_offset:        physical_offset: length:   expected: flags:
   0:        0..       0:    2434813..   2434813:      1:
   1:        1..       4:    2433506..   2433509:      4:    2434814:
   2:        5..       8:    2452465..   2452468:      4:    2433510: last,eof
/var/log/boot.log: 3 extents found

[student@localhost ~]$ du -h *
4.0K     backup.sh
4.0K     copy2dir
196M     movie.mkv
```

File systems

- There are a lot of different file systems
  - simple multi-platform: FAT12, FAT16, FAT32, exFAT
  - Windows: NTFS
  - Apple: APFS, HFS+
  - Linux: ext4, XFS
  - …
- Every file system has its own structure and stores meta-information in its own way
  - FAT doesn’t support users, permissions, …
  - NTFS has a completely different permission system
  - …

Create a file system

- Creating a file system is done by formatting
- You use the **mkfs** command to do so

`mkfs -t` *`type block_device`*	Generic creation of a new filesystem on a block device
`mkfs.xfs` *`block_device`*	Create a new XFS filesystem on a block device
`mkfs.ext4` *`block_device`*	Create a new EXT4 filesystem on a block device
`mkfs.fat` *`block_device`*	Create a new FAT filesystem on a block device

Karel de Grote Hogeschool

Exercise

- Create a FAT file system on the first partition of the new hard disk
- Create an ext4 file system on the second partition
- Can you add files to the partitions now?

Karel de Grote Hogeschool

4

Mounting a file system

Karel de Grote Hogeschool

Mounting a file system

- In Linux a file system is associated with a directory in the existing hiërarchie
- There are no drive letters like in Windows (C:, D:, …)
- The directory is called the “mount point”
- You can mount a file system with the “mount” command
- e.g.: sudo **mount** /dev/sda1 /mnt
- You can see the mounted file systems with the **df** command
- in the file /etc/fstab you can find all file systems that should automatically be mounted after a boot

UUID

- When block devices are removed and re-inserted they can get another name in /dev
- Block devices can therefore also get a “UUID”
- You can find the block devices with their UUID in /dev/disk/by-uuid

Exercise

- Create 2 directories in /mnt named “part1” and “part2”
- Mount the two partitions on those directories
  - use the device name for the first partition
  - use the UUID for the second partition
- Verify that this succeeded, using df
- Can you now copy files to these partitions?
- Can you create directories?
- What do the permissions look like on both partitions? Can you change them?
- How do you unmount a file system?
- Are the files now lost?
- What happens after a restart?

Karel de Grote Hogeschool

Exercises

Exercises

- KdG
  - …
- RedHat
  - ch13s02
  - ch13s04

Karel de Grote Hogeschool

Karel de Grote
Hogeschool