

FHS. Disks. Filesystems. Archives

Knowing FHS. Working with Archives, Disks, and Filesystems



SoftUni Team
Technical Trainers



SoftUni



Software University

<https://softuni.bg>

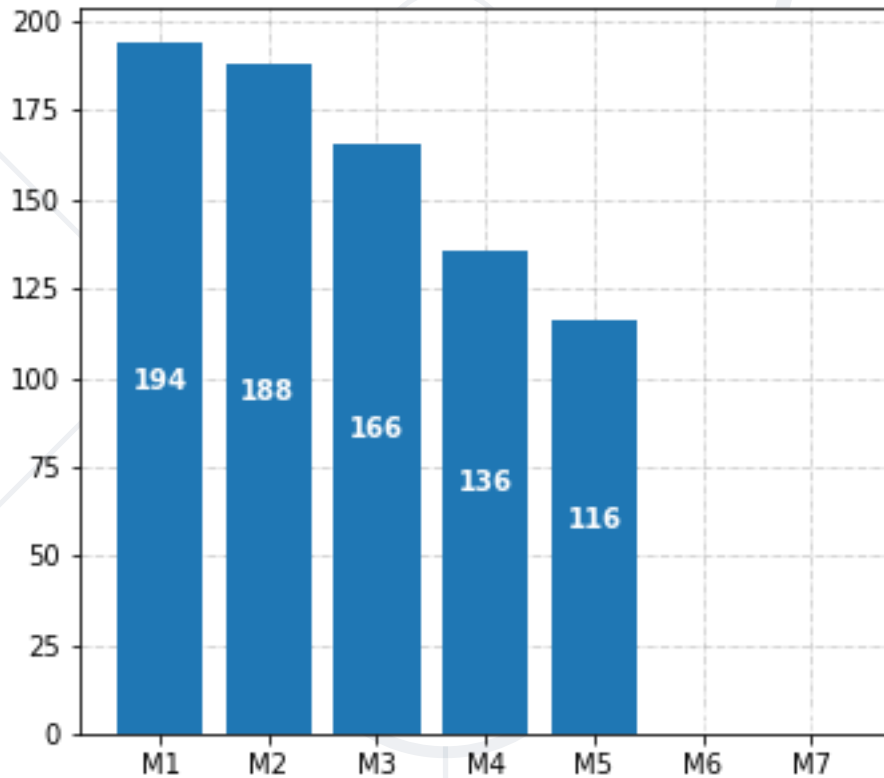
Have a Question?

sli.do

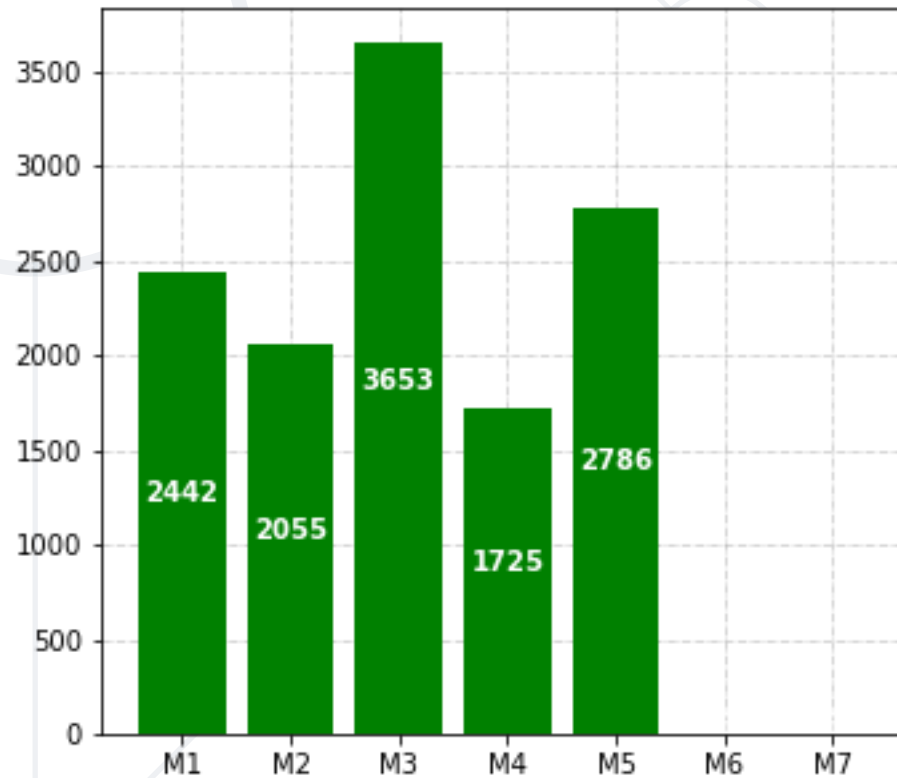
#LSA

Homework Progress

Submitted Homeworks



Homework Checks



Solutions for **M5**
can be submitted
until **23:59:59**
on **10.04.2025**

Solutions for **M6**
can be submitted
until **23:59:59**
on **17.04.2025**



Quick Overview

Previous Module (M5)

What We Covered

1. Boot Managers
2. System Startup Process
3. Systemd Components
4. System Management
5. Processes Monitoring & Management
6. Resource Monitoring



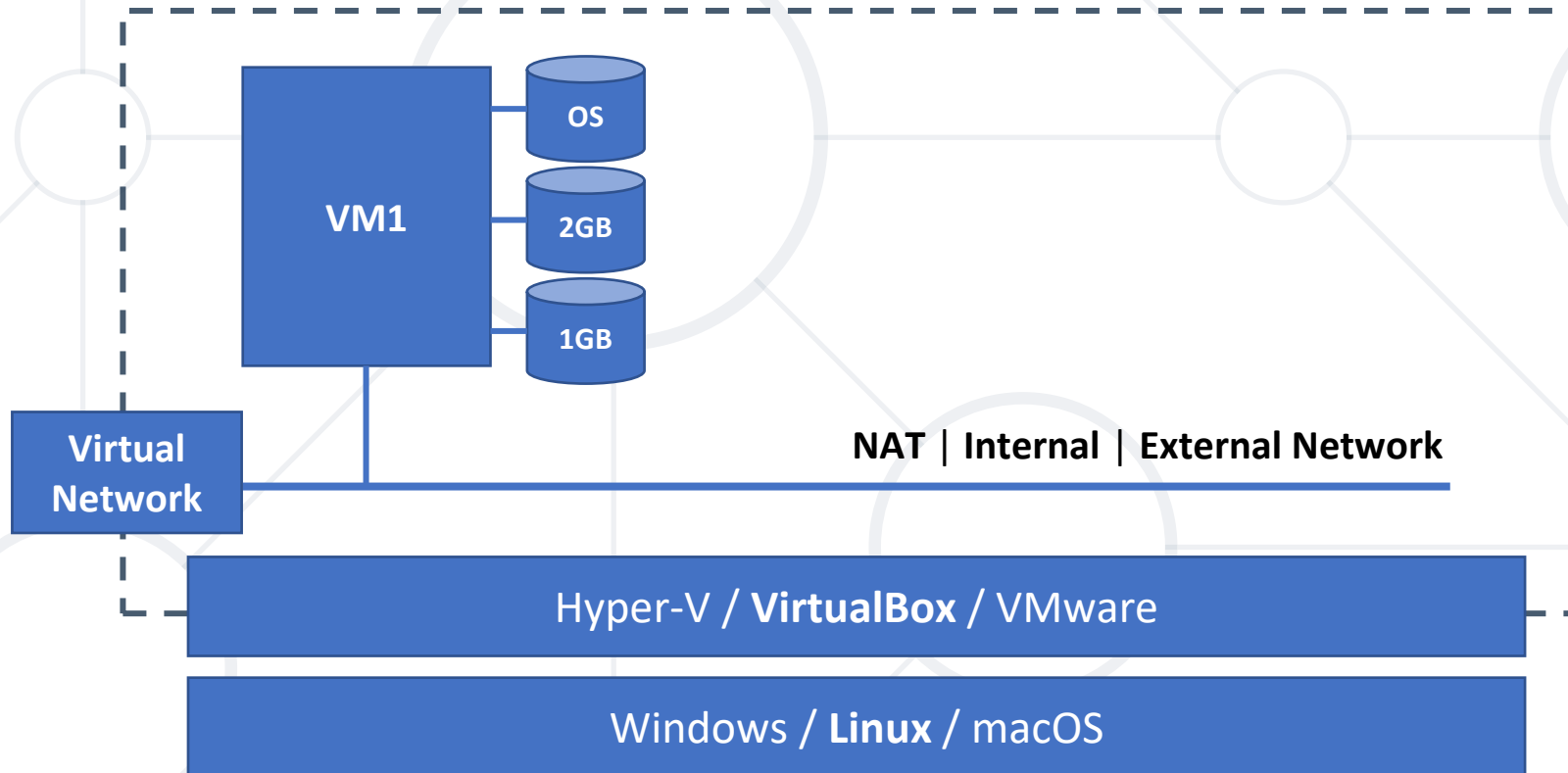
This Module (M6)

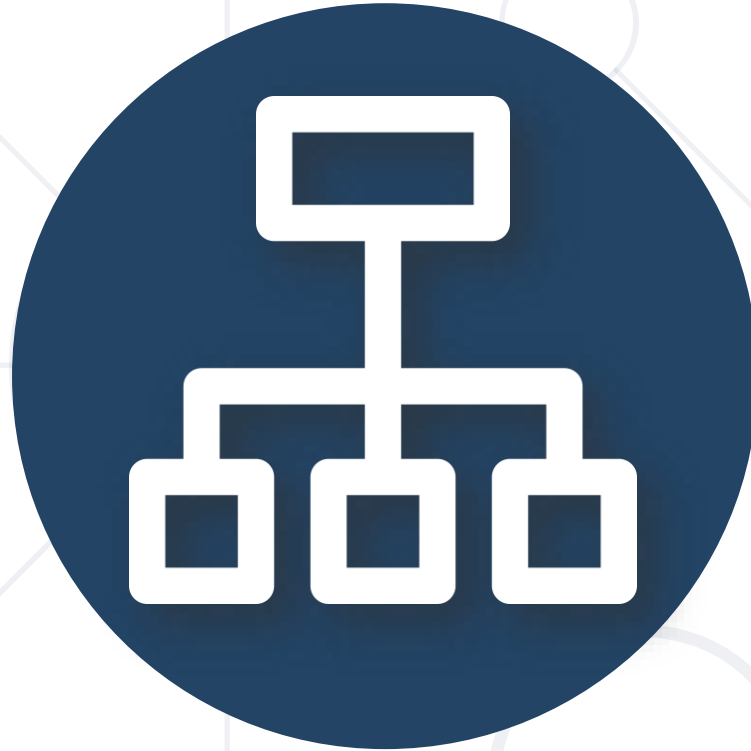
Topics and Lab Infrastructure

Table of Contents

1. Filesystem Hierarchy Standard (FHS)
2. Archiving Tools
3. Disks and Partitions Schemes
4. File Systems





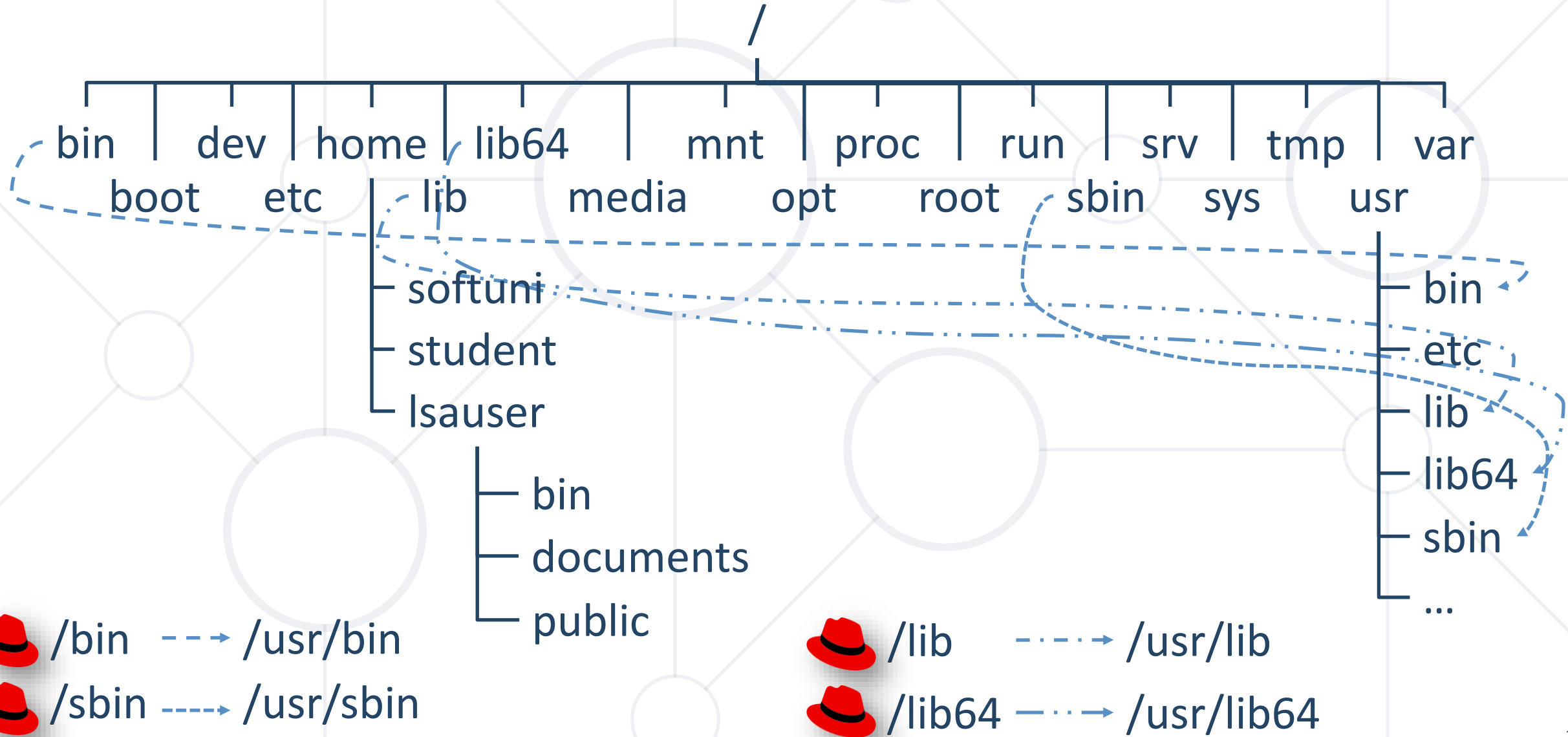


Filesystem Hierarchy Standard

Structure and Guidelines

- Defines the **directory structure** and **directory contents** in Linux distributions
- Maintained by the **Linux Foundation**
- Linux distributions can voluntarily conform to the standard
- All files and directories appear under the **root directory (/)**, even if they may be stored on **different physical** (separate **hard disk**) or **virtual devices** (for example, a **memory disk**)
- Some of these directories only exist on a particular system if certain subsystems are installed
- Information about the standard can be seen by executing **man hier**

Filesystem Hierarchy Standard Illustrated



- **Shareable**
 - User data files and program binary files
- **Un-shareable**
 - System-specific information
- **Static**
 - Binaries and scripts
- **Variable**
 - User files, mail, ...

	Shareable	Un-shareable
Static	/usr /opt	/etc /boot
Variable	/home /var/mail	/var/run /var/lock

- **/**
 - The root of the filesystem or the root directory
- **/boot**
 - Static and un-shareable files related to the booting process
- **/etc**
 - Static and un-shareable system configuration files
- **/bin**
 - Critical executable files

- **/sbin**
 - Similar to **/bin**, but contains programs usually executed by root
- **/lib** and **/lib64**
 - Contain program libraries
- **/usr**
 - Shareable and static. Hosts binaries, libraries, documentation, ...
- **/opt**
 - Used for external software, not part of the distribution

- **/home**
 - Variable and shareable. Contains used data
- **/root**
 - It is the home directory of the root user
- **/var**
 - Variable and partially shareable. Contains logs, mail spool, ...
- **/tmp**
 - Variable and temporary content

- **/mnt**
 - Used for mounting of removable media
- **/media**
 - Similar to **/mnt**, but should contain subfolders for different media

Filesystem Hierarchy Standard (5)

- **/dev**
 - Files used as interface to the underlying hardware devices. Uses **devtmpfs**
- **/proc**
 - Provides process and kernel information as files. Uses **proc (procfs)** filesystem
- **/sys**
 - Contains information about devices, drivers, and some kernel features. Uses **sysfs**
- **/run**
 - Contains run-time variable data. Exists in memory. Uses **tmpfs**

Virtual (pseudo) filesystems



Archiving Tools

Backup and Restore Techniques

- Purpose
 - Save and restore multiple files in/from single archive file

- Syntax

```
tar [options] [files]
```

- Example

```
# Create archive
```

```
[user@host ~]$ tar -cvf folder.tar /folder
```

```
# Extract files from archive
```

```
[user@host ~]$ tar -xvf folder.tar
```

- Purpose
 - Package and compress archive files

- Syntax

```
zip [options] archive infile
```

- Example

```
# Create archive of one file
```

```
[user@host ~]$ zip file.zip file
```

```
# Create archive of a folder
```

```
[user@host ~]$ zip -r folder.zip folder
```

- Purpose
 - List, test, and extract compressed files

- Syntax

```
unzip [options] file[.zip] [files]
```

- Example

```
# List files in archive
```

```
[user@host ~]$ unzip -l file.zip
```

```
# Extract files from archive, but skip *.conf
```

```
[user@host ~]$ unzip file.zip -x *.conf
```

- Purpose
 - Compress or expand files

- Syntax

```
gzip [options] [files]
```

- Example

```
# Create archive and replace the original file
```

```
[user@host ~]$ gzip file.txt
```

```
# Create archive and keep the original file
```

```
[user@host ~]$ gzip -c file.txt > file.txt.gz
```

- Purpose
 - Compress or expand files using the block-sorting technique

- Syntax

```
bzip2 [options] [files]
```

- Example

```
# Create archive and keep the original file
```

```
[user@host ~]$ bzip2 -zk file.txt
```

```
# Extract files from archive
```

```
[user@host ~]$ bunzip2 file.txt.bz2
```

- Purpose
 - General purpose data compression tool

- Syntax

```
xz [options] [files]
```

- Example

```
# Create archive and keep the original file
```

```
[user@host ~]$ xz -zk file.txt
```

```
# Archive several files
```

```
[user@host ~]$ xz -c *.txt > file.xz
```


- **(z | bz | xz)cat**
 - Usually the same as **[utility] -c** . Output contents to the STDOUT
- **(z | bz | xz)less**
 - Display text from compressed files to a terminal
- **(z | bz | xz)more**
 - Display text from compressed files to a terminal
- **(z | bz | xz)grep**
 - Search possibly compressed files for a regular expression

■ Construction

tar **c** (create) **z** (gzip)
t (list) + **v** (verbose) + **j** (bzip2) + **f** (filename)
x (expand) **J** (xz)

- Create archive

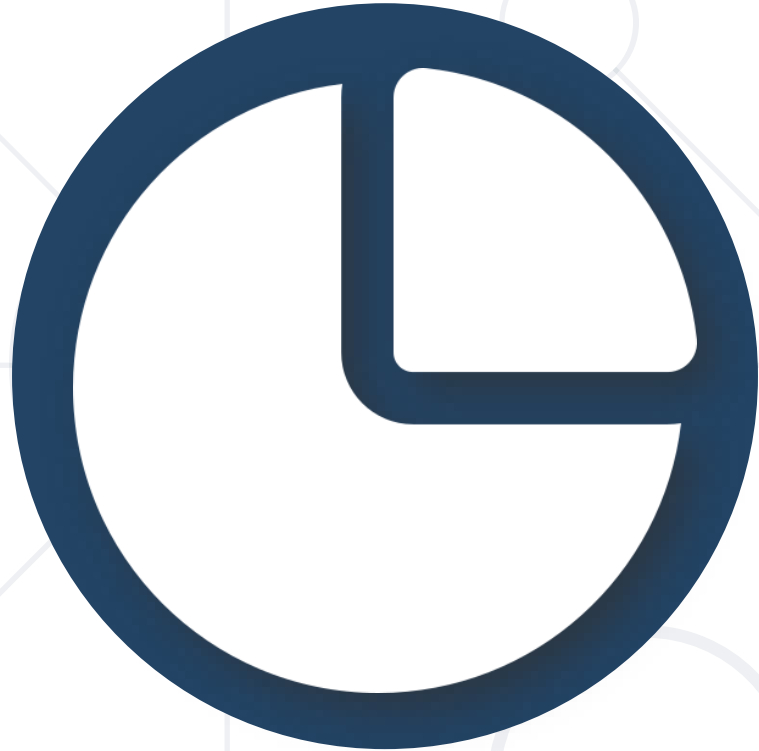
```
[user@host ~]$ tar -czvf archive.tar.gz /etc
```

- Extract archive

```
[user@host ~]$ tar -xzvf archive.tar.gz
```



Practice



Disks and Partition Schemes

Disk Types. Partition Schemes

Disk Types

- Parallel ATA (**PATA**)
 - 40/80 pins, 2 devices/channel, 128 - 1064 Mbps
 - **/dev/hdX**
- Serial ATA (**SATA**) *
 - 7 pins, 1 device/channel, 1.5 - 6.0 Gbps
 - **/dev/sdX** and **/dev/srN** for CD/DVD
- Small Computer System Interface (**SCSI**) *
 - varies, 8 or 16 devices/bus
 - **/dev/sdX** and **/dev/srN** for CD/DVD
- Non-Volatile Memory Express (**NVMe**)
 - Up to 16 GT/s (approx. 16 Gbps) per lane for PCIe Gen 4
 - **/dev/nvmeNnM**

In some virtualization solutions hard disk drives may appear as **/dev/vdX** or **/dev/xdX**

SD memory cards appear as **/dev/mmcblkN**

USB disks and thumb drives appear as **/dev/sdX**

X is a character like a, b, etc.
N is a number like 0, 1, etc.
M is a number like 0, 1, etc.

* Use the same driver and thus having the same device node files.

Why Partition?

1. Multiple OS Support
2. Filesystem Choice
3. Disk Space Management
4. Disk Error Protection
5. Security
6. Backup

Measure Units: MB vs MiB

10^6 B

(Decimal / Powers of 10)

1 MB

megabyte

$1 \text{ MB} = 1000 \text{ KB} = 1000 * 1000 \text{ B} = 1000000 \text{ B}$

<https://en.wikipedia.org/wiki/Megabyte>

2^{20} B

(Binary / Powers of 2)

1 MiB

mebibyte

$1 \text{ MiB} = 1024 \text{ KiB} = 1024 * 1024 \text{ B} = 1048576 \text{ B}$

<https://en.wikipedia.org/wiki/Mebibyte>



Software
University

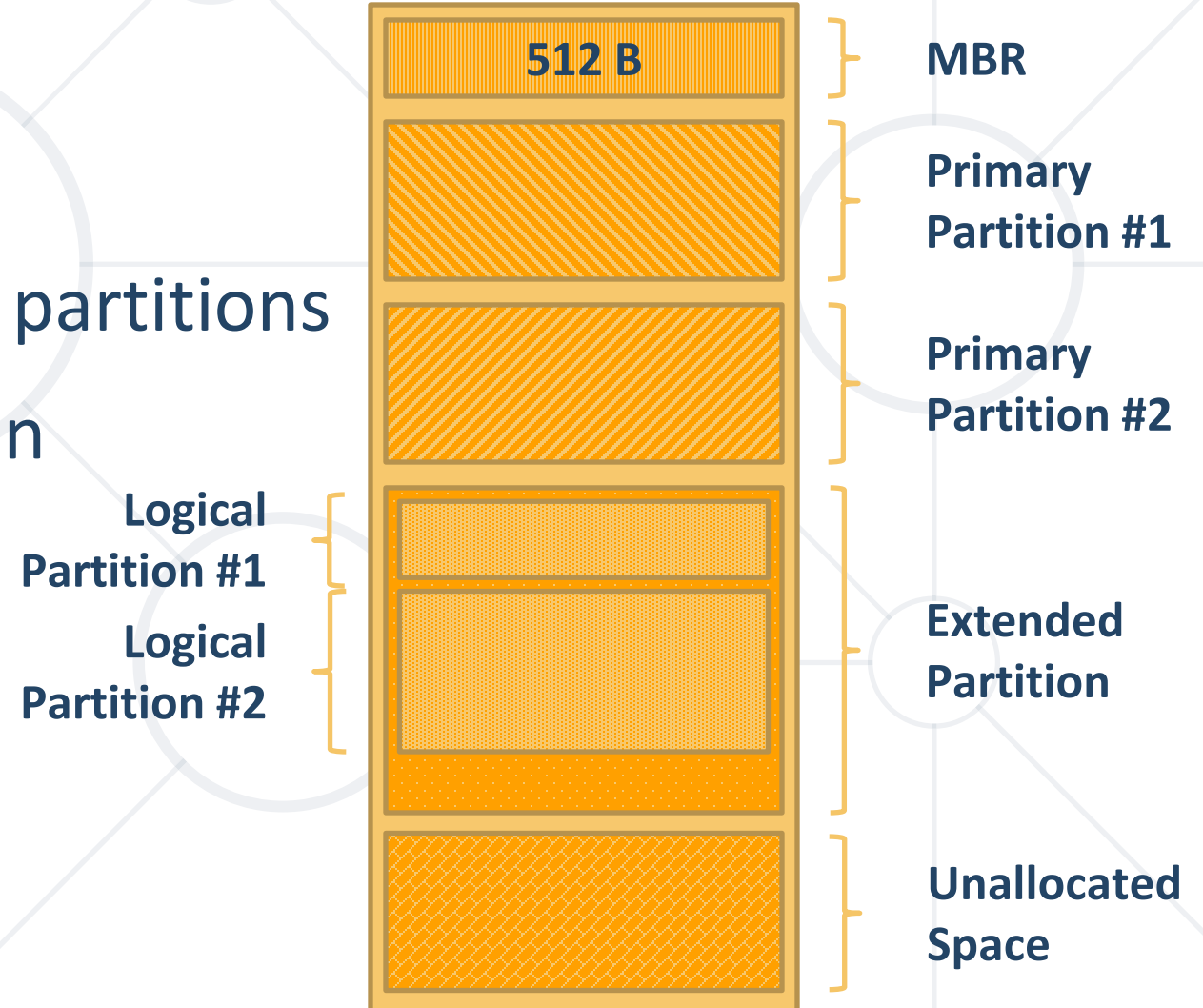
Master Boot Record (MBR)

■ Characteristics

- Occupies first 512 Bytes
- Up to 4 primary (or 3 + 1 ext) partitions
- Maximum size 2 TiB / partition

■ Partition types

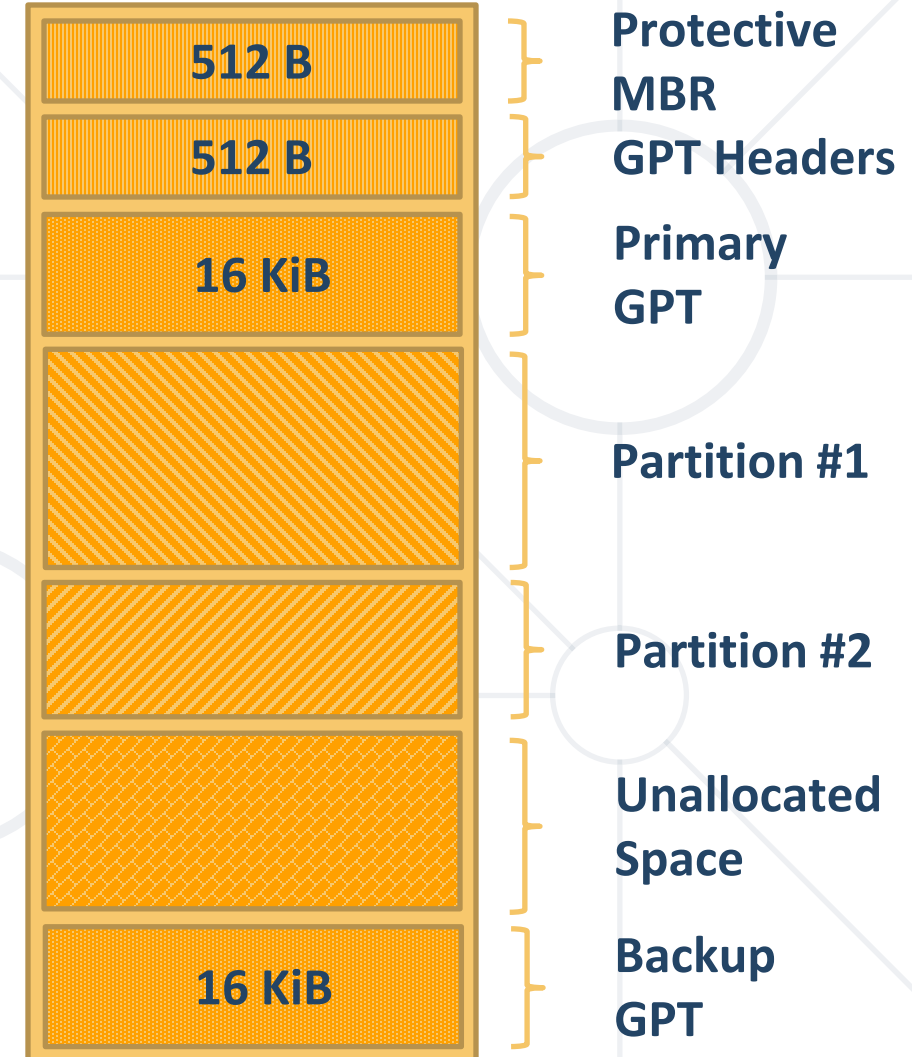
- Primary
- Extended
- Logical



GUID Partition Table (GPT)

■ Characteristics

- Part of the EFI specification
- Has protective MBR
- Two copies of the partition table
- Up to 128 partitions
- Maximum size 8 ZiB / partition



■ RAW Partitions

- No regular filesystem
- Used for special purposes, i.e. enterprise databases
- Better control and performance

■ Swap Partitions

- No further separation
- No regular filesystem
- Used to offload memory pages from RAM. 1-2x RAM size *

* Depends on the usage pattern and on the installed memory. For example, on a host with 32 GB of RAM, we would set it to anything between 2 and 8 GB.

Linux Filesystem
83 or 8300

Linux Swap
82 or 8200

Linux LVM
8e or 8e00

- **Could be separated**
 - **/boot** – 100MiB - 1GiB
 - **/home** – 200MiB - 3TiB (or more)
 - **/opt** – 100MiB - 5GiB
 - **/var** – 100MiB - 3TiB (or more)
 - **/tmp** – 100MiB - 20GiB
- **Should be together**
 - **/etc, /bin, /sbin, /lib, /lib64, and /usr ***

- Every application has different default settings
- KB is **not equal** to KiB
 - KB is kilobytes, **1KB = 1000 Bytes**
 - KiB is kibibytes, **1KiB = 1024 Bytes**
- Check twice before applying/writing any changes
- No easy way to revert in case of a mistake



- Purpose
 - List block devices
- Syntax

```
lsblk [options] [device]
```

- Examples

```
# List all non-empty block devices
```

```
[user@host ~]$ lsblk
```

```
# Show information about the filesystem type
```

```
[user@host ~]$ lsblk -f
```

- Purpose
 - Manipulate disk partition table (MBR)

- Syntax

```
fdisk [options] [device]
```

- Examples

```
# List all partitions of a drive  
[root@host ~]# fdisk -l /dev/sda  
# Enter maintenance mode for a device  
[root@host ~]# fdisk /dev/sda
```

- Purpose
 - Interactive GUID partition table (GPT) manipulator

- Syntax

```
gdisk [-l] device
```

- Examples

```
# List partition tables for a device
```

```
[root@host ~]# gdisk -l /dev/sda
```

```
# Enter maintenance mode for a device
```

```
[root@host ~]# gdisk /dev/sda
```


- Purpose
 - Partition manipulation program
- Syntax

```
parted [options] [device [command [options]]]
```

- Examples

```
# List partition layout for all block devices
```

```
[root@host ~]# parted -l
```

```
# Enter maintenance mode for a device
```

```
[root@host ~]# parted /dev/sda
```

- Purpose
 - Copy a file, converting and formatting according to the operands
- Syntax

```
dd [options] [device]
```

- Examples

```
# Backup the first sector of a disk
```

```
[root@host ~]# dd if=/dev/sda of=512.mbr bs=512 count=1
```

```
# Zero first 32 KB of a disk
```

```
[root@host ~]# dd if=/dev/zero of=/dev/sda bs=16K count=2
```

- Purpose
 - Setup a Linux swap area

- Syntax

```
mkswap [options] device [size]
```

- Examples

```
# Initialize new swap partition  
[root@host ~]# mkswap -L SWAP-2 /dev/sdb2
```

- Purpose
 - Enable devices and files for paging and swapping

- Syntax

```
swapon [options] [device|file]
```

- Examples

```
# Display information for all swap targets
```

```
[root@host ~]# swapon -s
```

```
# Enable a device for swapping with priority of 5
```

```
[root@host ~]# swapon -p 5 -v /dev/sdb11
```

- Purpose
 - Disable devices and files for paging and swapping

- Syntax

```
swapoff [options] [device]
```

- Examples

```
# Disable all swap devices and files
```

```
[root@host ~]# swapoff -a -v
```

```
# Turn off particular device
```

```
[root@host ~]# swapoff -v /dev/sdb11
```



Practice



Filesystems

Filesystem Types

What is a Filesystem?

- **Definition #1**
 - Hierarchy of directories (directory tree) used to organize files on a computer system
- **Definition #2**
 - The way the storage of data (directories, files, ...) is organized on a disk or on a partition of a disk

- **Superblock**

- **Contains the characteristics of the filesystem**, including its size, block size, empty and filled blocks and their counts, size and location of the **inode table**, etc. **One main and multiple copies**

- **Inode**

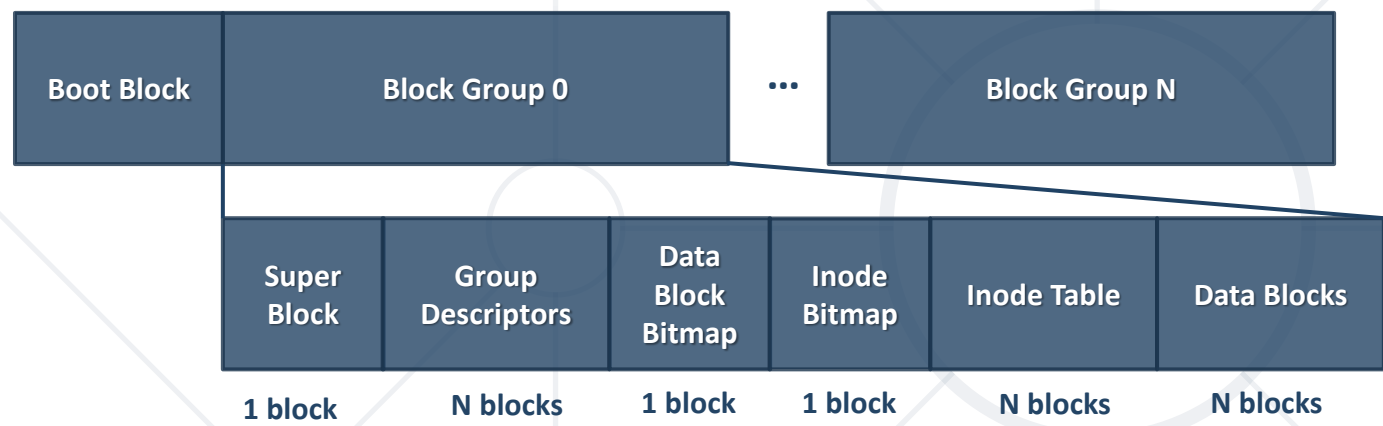
- **Data structure** that stores all the information (**metadata only**) about a file except its name and data

- **Data block**

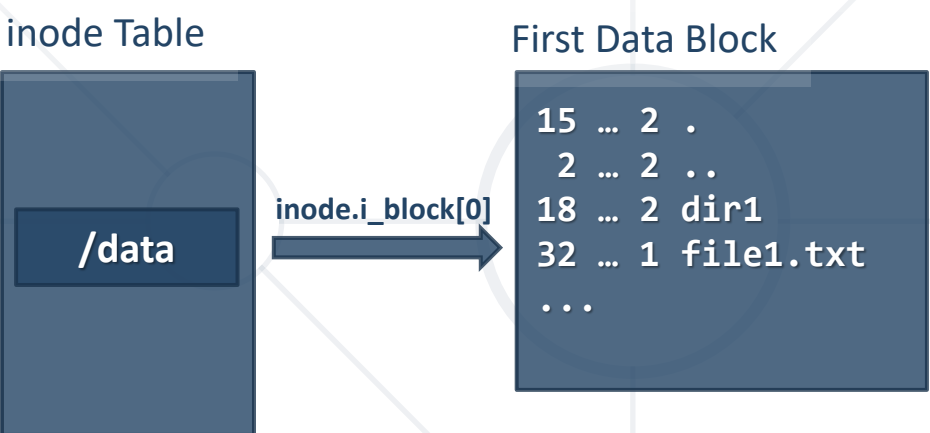
- **Contains the information** stored in the individual files

Filesystem Components Illustrated

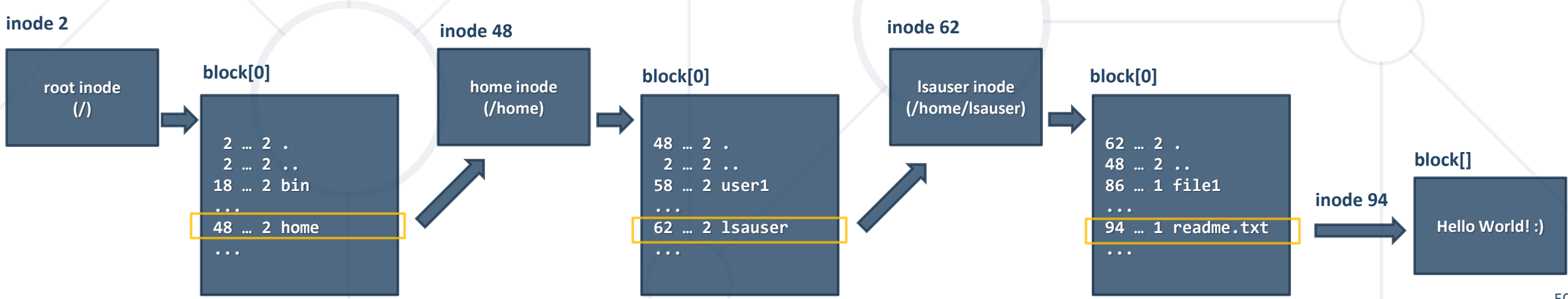
Simplified filesystem structure



Simplified inode table and directory entries



Reading a file (cat /home/lsauser/readme.txt)



Extended Filesystem – Ext(2|3|4)

- Version 2 – **No journaling**
- Version 3 – **Journaling added**
- Version 4 – **Still in active use**
 - Max volume size: 1 EiB
 - Max file size: 16 TiB
 - Max filename length: 255 bytes
 - Backward-compatible with ext2 and ext3

Journal keeps track of
filesystem changes

In case of sudden shutdown
consistency is guaranteed

■ Characteristics

- Max volume size: 8 EiB
- Max file size: 8 EiB
- Max filename length: 255 bytes

■ Features

- Journaling
- Online defragmentation
- Online resizing

■ Characteristics

- Max volume size: 16 EiB
- Max file size: 16 EiB
- Max filename length: 255 bytes

■ Features

- Online defragmentation
- Online growth and shrinking
- Transparent compression
- Snapshots



Filesystems

Filesystem Tools

- Purpose
 - Create an ext2/ext3/ext4 filesystem. Reads /etc/mke2fs.conf

- Syntax

```
mke2fs [options] device
```

- Examples

```
# Create ext4 filesystem
```

```
[root@host ~]# mke2fs -t ext4 /dev/sdb1
```

```
# Create ext3 filesystem with Label
```

```
[root@host ~]# mke2fs -t ext3 -L EXT3FS /dev/sdb2
```


- Purpose
 - Build a Linux filesystem

- Syntax

```
mkfs [options] device
```

- Examples

```
# Create ext4 filesystem
```

```
[root@host ~]# mkfs -t ext4 /dev/sdb1
```

```
# Create xfs filesystem
```

```
[root@host ~]# mkfs.xfs /dev/sdb2
```

- Purpose
 - Adjust tunable filesystem parameters on ext2/ext3/ext4

- Syntax

```
tune2fs [options] device
```

- Examples

```
# List the contents of the superblock
```

```
[root@host ~]# tune2fs -l /dev/sdb1
```

```
# Set volume label of the filesystem
```

```
[root@host ~]# tune2fs -L MY-DATA /dev/sdb1
```

- Purpose
 - Dump ext2/ext3/ext4 filesystem information

- Syntax

```
dumpe2fs [options] device
```

- Examples

```
# Display detailed information about a filesystem
```

```
[root@host ~]# dumpe2fs /dev/sdb1
```

```
# Show only superblock information
```

```
[root@host ~]# dumpe2fs -h /dev/sdb1
```

- Purpose
 - Resize ext2/ext3/ext4 filesystem

- Syntax

```
resize2fs [options] device [size]
```

- Examples

```
# Expand the filesystem to the available space  
[root@host ~]# resize2fs /dev/sdb2
```

- Purpose
 - Check ext2/ext3/ext4 filesystem

- Syntax

```
e2fsck [options] device
```

- Examples

```
# Check a filesystem without any changes
```

```
[root@host ~]# e2fsck -n /dev/sdb2
```

```
# Check and automatically fix any errors
```

```
[root@host ~]# e2fsck -p /dev/sdb2
```

- Purpose
 - Check and repair a Linux filesystem

- Syntax

```
fsck [options] {device | mount point}
```

- Examples

```
# Check a filesystem without any changes
```

```
[root@host ~]# fsck -N /dev/sdb2
```

```
# Check and fix any errors and show details
```

```
[root@host ~]# fsck -V /dev/sdb2
```

- Purpose
 - Show XFS filesystem information
- Syntax

```
xfs_info mount-point
```

- Examples

```
# Display detailed information about a filesystem  
[root@host ~]# xfs_info /media/disk1
```

- Purpose
 - Change parameters of an XFS filesystem
- Syntax

```
xfs_admin [options] device
```

- Examples

```
# Print the current Label
```

```
[root@host ~]# xfs_admin -l /dev/sdb1
```

```
# Set new Label
```

```
[root@host ~]# xfs_admin -L MY-DATA /dev/sdb1
```


- Purpose
 - Extend XFS filesystem

- Syntax

```
xfs_growfs [options] mount-point
```

- Examples

```
# Add all space available in the Logical Volume  
[root@host ~]# xfs_growfs /media/disk1
```

- Purpose
 - Repair an XFS filesystem
- Syntax

```
xfs_repair [options] device
```

- Examples

```
# Execute a dry-run repair. No modifications
```

```
[root@host ~]# xfs_repair -nv /dev/sdb1
```

```
# Execute the repair procedure with details
```

```
[root@host ~]# xfs_repair -vv /dev/sdb1
```

- Purpose
 - Toolbox for managing btrfs filesystems

- Syntax

```
btrfs <command> [<args>]
```

- Examples

```
# Show space usage information for /  
[root@host ~]# btrfs filesystem df /  
# Show detailed information about internal FS usage  
[root@host ~]# btrfs filesystem usage /
```

- Purpose
 - Mount a filesystem
- Syntax

```
mount [options] device directory
```

- Examples

```
# Mount partition to a directory
```

```
[root@host ~]# mount /dev/sdb1 /media/disk1
```

```
# Mount all filesystems mentioned in /etc/fstab
```

```
[root@host ~]# mount -a
```

- Purpose
 - Unmount filesystems
- Syntax

```
umount [options] {device|directory}
```

- Examples

```
# Unmount by partition
```

```
[root@host ~]# umount /dev/sdb1
```

```
# Unmount by directory
```

```
[root@host ~]# umount /media/drive
```

- Purpose
 - Locate or print block device attributes

- Syntax

```
blkid [options] device
```

- Examples

```
# List information about all block devices
```

```
[root@host ~]# blkid
```

```
# Search for devices with filesystem type ext4
```

```
[root@host ~]# blkid -l -t TYPE=ext4
```

- Purpose
 - List content of directories in a tree-like format

- Syntax

```
tree [options] [directory]
```

- Examples

```
# List content of the current directory
```

```
[user@host ~]$ tree
```

```
# List content of a directory including hidden info
```

```
[user@host ~]$ tree -a /directory
```

- Configuration file
 - **/etc/fstab**

```
[root@jupiter ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Sat Jun 10 19:27:51 2017
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/cl-root      /                    xfs     defaults        0 0
UUID=d72ce7e9-6647-402b-9ad8-73d42b447b0c /boot                xfs     defaults        0 0
/dev/mapper/cl-swap      swap                 swap     defaults        0 0
[root@jupiter ~]#
```

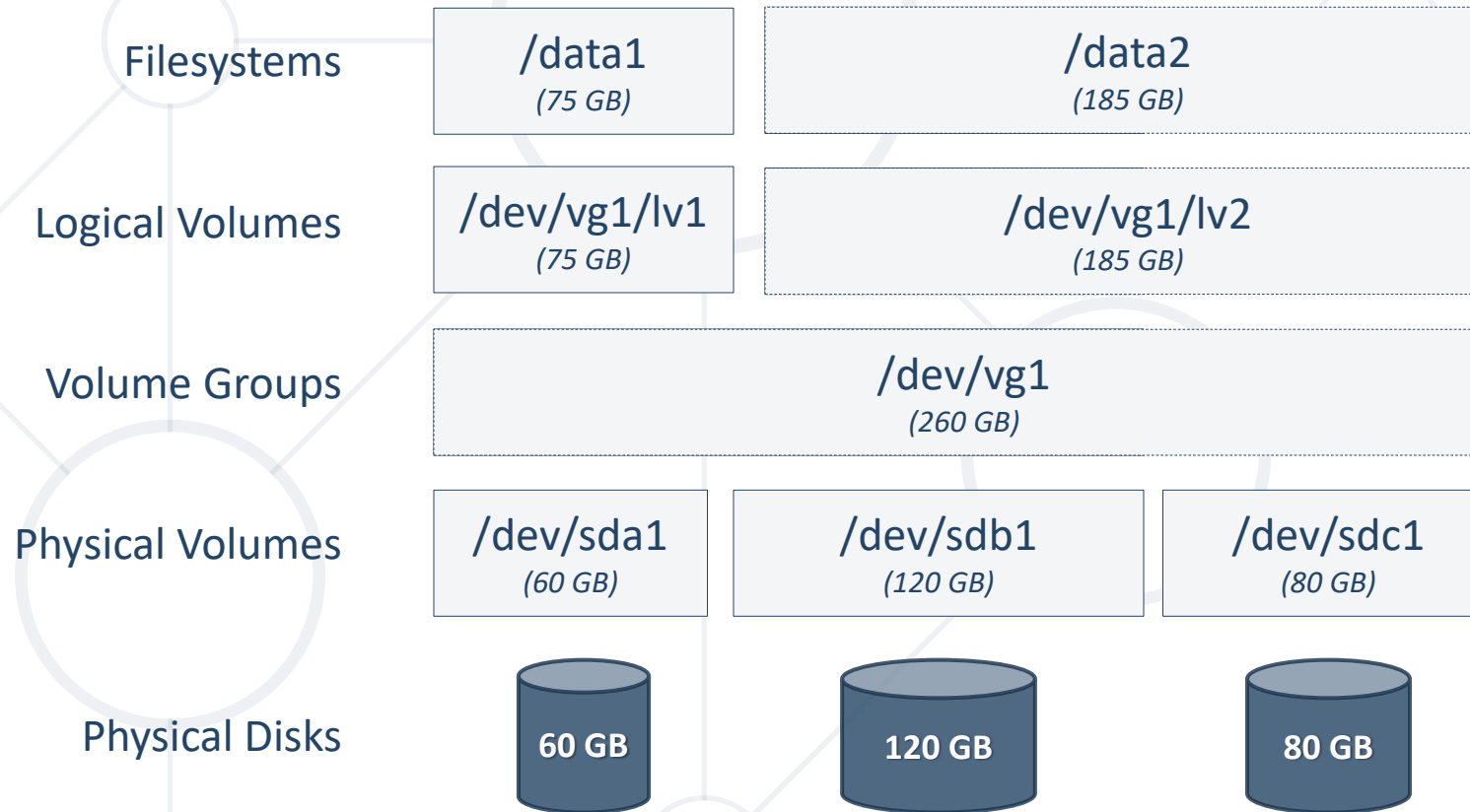
- Mount always in empty directory



Logical Volume Management

Components and Tools

Logical Volume Management (LVM)



- Purpose
 - Scan all disks for physical volumes

- Syntax

```
pvscan [options]
```

- Example

```
# Display information about available PVs  
[root@host ~]# pvscan
```

- Purpose
 - Report information about physical volumes

- Syntax

```
pvs [options]
```

- Example

```
# Display information about initialized PVs
```

```
[root@host ~]# pvs
```

```
# Display information about all PVs
```

```
[root@host ~]# pvs -a
```

- Purpose
 - Initialize a disk or partition for use by LVM

- Syntax

```
pvcreate [options] device
```

- Example

```
# Initialize a partition
```

```
[root@host ~]# pvcreate /dev/sdb1
```

```
# Initialize entire disk
```

```
[root@host ~]# pvcreate /dev/sdb
```

- Purpose
 - Scan all disks for volume groups and rebuild caches

- Syntax

```
vgscan [options]
```

- Example

```
# Scan and print extra information  
[root@host ~]# vgscan -v
```

- Purpose
 - Report information about volume groups

- Syntax

```
vgs [options]
```

- Example

```
# Report information about volume groups  
[root@host ~]# vgs
```

- Purpose
 - Create a volume group
- Syntax

```
vgcreate [options] group-name device
```

- Example

```
# Create volume group with one PV
```

```
[root@host ~]# vgcreate vg1 /dev/sda1
```

```
# Create volume group with two PVs and print info
```

```
[root@host ~]# vgcreate -v vg1 /dev/sda1 /dev/sda2
```


- Purpose
 - Add physical volumes to a volume group

- Syntax

```
vgextend [options] group-name device
```

- Example

```
# Add PV to a VG
```

```
[root@host ~]# vgextend vg1 /dev/sda5
```

- Purpose
 - Scan all disks for logical volumes

- Syntax

```
lvscan [options]
```

- Example

```
# Perform scan  
[root@host ~]# lvscan
```

- Purpose
 - Report information about logical volumes

- Syntax

```
lvs [options]
```

- Example

```
# Display the report  
[root@host ~]# lvs
```

- Purpose
 - Create a logical volume in an existing volume group

- Syntax

```
lvcreate [options] lv-name [options] vg-name
```

- Example

```
# Create new logical volume
```

```
[root@host ~]# lvcreate -n lv1 -L 500m vg1
```

- Purpose
 - Extend the size of a logical volume

- Syntax

```
lvextend [options] volume-name
```

- Example

```
# Increase the size with 100MB
```

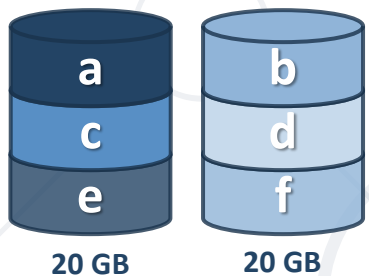
```
[root@host ~]# lvextend -L +100m /dev/vg1/lv1
```



RAID

Types and Tools

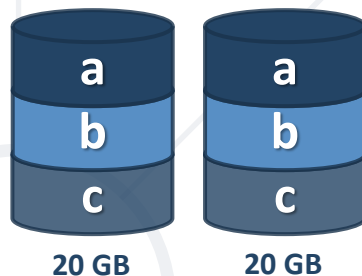
- **Redundant array of independent** (inexpensive) **disks**
- It can be either **hardware** or **software**
- Various implementations (levels)



RAID 0 (Stripe)

MI: 2 / FT: 0

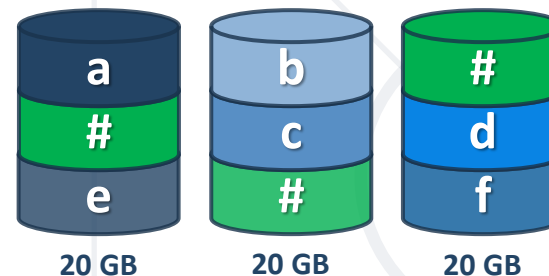
Total: 40 GB



RAID 1 (Mirror)

MI: 2 / FT: $n - 1$

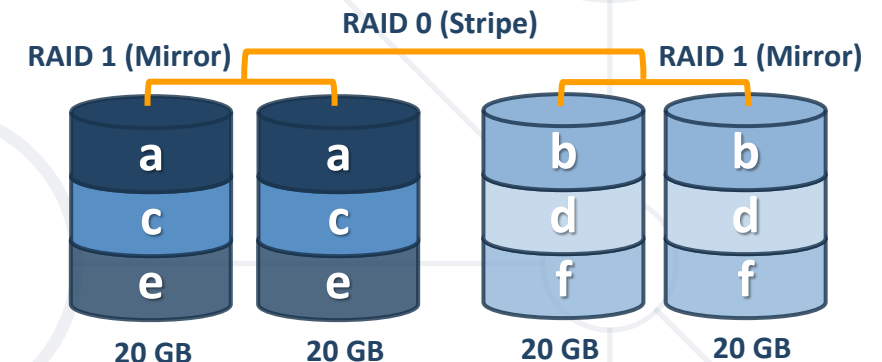
Total: 20 GB



RAID 5 (Stripe /w parity)

MI: 3 / FT: 1

Total: 40 GB



RAID 10 (RAID 1 + 0)*

MI: 4 (6, 8, ...) / FT: 1 (in each group)

Total: 40 GB

* RAID 1+0 is also known as **Stripe of Mirrors**. Other nested implementations are available as well (0+1, 5+0, ...)

- Purpose
 - Manage MD devices also known as Linux Software RAID

- Syntax

```
mdadm [mode] <raiddevice> [options] <component-devices>
```

- Example

```
# Create a RAID1 array
```

```
[root@host ~]# mdadm --create /dev/md0 --level=1 --  
raid-devices=2 /dev/sdb1 /dev/sdc1
```




Practice

- The layout of Linux filesystem is shaped by the FHS
- FHS implementation varies among distributions
- zip, xz, and bzip are just a few of all available archiving tools
- Usually archiving tools are combined with the tar utility
- MBR and GPT are two partitioning schemes

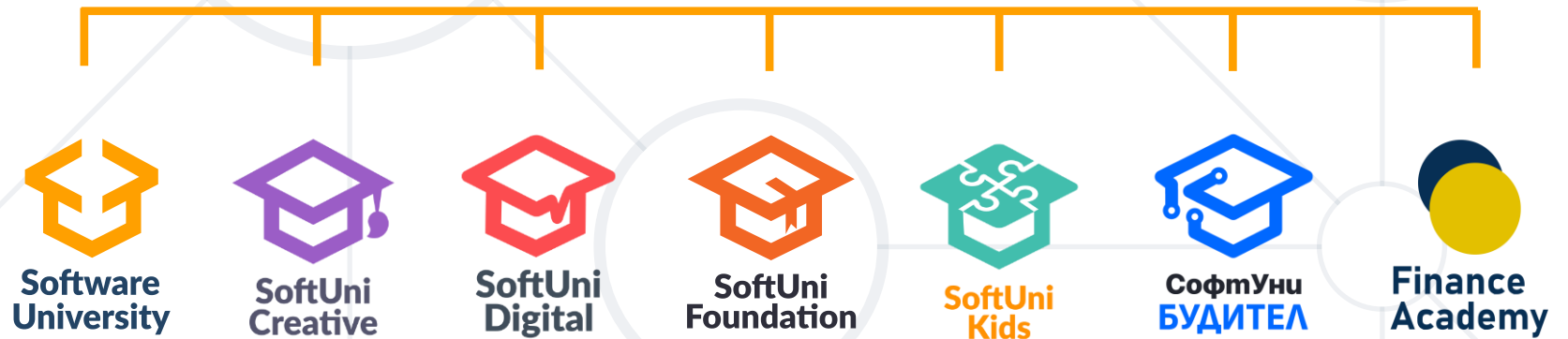


- Disks are further divided into partitions. They can be used directly or through LVM
- ext4, xfs, and btrfs are a few of the filesystems that are supported in Linux
- Each filesystem offers different capabilities thus suitable for different purposes
- Different filesystems can be used simultaneously



- File Hierarchy Standard
 - <http://refspecs.linuxfoundation.org/fhs.shtml>
- Linux filesystems concepts
 - <https://opensource.com/life/16/10/introduction-linux-filesystems>
- Introduction to EXT filesystem
 - <https://opensource.com/article/17/5/introduction-ext4-filesystem>
- Hard disk layout
 - <https://www.ibm.com/developerworks/library/l-lpic1-102-1/index.html>
- How to partition and format storage devices in Linux
 - <https://www.digitalocean.com/community/tutorials/how-to-partition-and-format-storage-devices-in-linux>

Questions?



SoftUni Diamond Partners



THE CROWN IS YOURS



- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg, about.softuni.bg
- Software University Foundation
 - softuni.foundation
- Software University @ Facebook
 - facebook.com/SoftwareUniversity



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is copyrighted content
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>

