LPI 102.1- Create partitions and filesystems

Curs 2021 - 2022

ASIX M01-ISO

Create partitions and filesystems

Description

Key concepts:

- ❏ Manage MBR and GPT partition tables
- ❏ Use various mkfs commands to create various filesystems such as:
- ❏ ext2/ext3/ext4
- ❏ XFS
- ❏ VFAT
- ❏ exFAT
- ❏ Basic feature knowledge of Btrfs, including multi-device filesystems, compression and subvolumes.

Commands and files:

- ❏ fdisk
- ❏ gdisk
- ❏ parted
- ❏ mkfs
- ❏ mkswap

Designing a scheme

There are three steps in the process of making and using partitions:

1. Divide the hard drive into partitions.
2. Create and format filesystems inside the partitions.
3. Mount the filesystem into the directory tree.

Partition Namimg:

/dev/sda1

Drives that start with sd are either SATA (Serial ATA), SCSI (Small Computer System Interface) or USB drives. /dev/sd*
Examples: /dev/sda /dev/sdb /dev/sda1 /dev/sdb2

/dev/nvme0n1p1

Drives SSD. /dev/nvme0n*
Examples: /dev/nvme0n1 /dev/nve0n1p1

/dev/hda1

Drives that start with hd are PATA (Parallel ATA), also known as IDE (Integrated Drive Electronics) drives. /dev/hd*
Examples: /dev/hda /dev/hdb

/dev/fd0

diskette

Identify partitions

Bye:

- name
- Label
- UUID

- lsblk
- blkid
- tree /dev/disk

```
# tree /dev/disk
/dev/disk
├── by-id
│   ├── dm-name-fedora-home -> ../../dm-2
│   ├── dm-name-fedora-root -> ../../dm-0
│   ├── dm-name-fedora-swap -> ../../dm-1
│   ├── dm-uuid-LVM-Ly5zp6Y155RnTEzxvilb9yYXtIhDcGrH6r4DWOyYTmspFPJzJCtjBAmfDPwcvg27 ->
../../dm-1
│   ├── dm-uuid-LVM-Ly5zp6Y155RnTEzxvilb9yYXtIhDcGrHPNXp1us9fnVHwun8MZSEfwVW3vE7GtLU ->
```

```
../../dm-0
│   ├── dm-uuid-LVM-Ly5zp6Y155RnTEzxvilb9yYXtIhDcGrHWEbZDpxRt6TPrH9em3dQNHQYJ3zouzhh ->
../../dm-2
│   ├── lvm-pv-uuid-VgZxLA-HigE-eyyU-DBHj-XbyP-e2gF-LEzdtR -> ../../nvme0n1p3
│   ├── nvme-eui.002538ab01d2bf96 -> ../../nvme0n1
│   ├── nvme-eui.002538ab01d2bf96-part1 -> ../../nvme0n1p1
│   ├── nvme-eui.002538ab01d2bf96-part2 -> ../../nvme0n1p2
│   ├── nvme-eui.002538ab01d2bf96-part3 -> ../../nvme0n1p3
│   ├── nvme-SAMSUNG_MZALQ256HAJD-000L2_S4ULNF2NB44121 -> ../../nvme0n1
│   ├── nvme-SAMSUNG_MZALQ256HAJD-000L2_S4ULNF2NB44121-part1 -> ../../nvme0n1p1
│   ├── nvme-SAMSUNG_MZALQ256HAJD-000L2_S4ULNF2NB44121-part2 -> ../../nvme0n1p2
│   └── nvme-SAMSUNG_MZALQ256HAJD-000L2_S4ULNF2NB44121-part3 -> ../../nvme0n1p3
├── by-partlabel
│   └── EFI\x20System\x20Partition -> ../../nvme0n1p1
├── by-partuuid
│   ├── 11ae4b81-2f10-4a68-a7cb-4c7fd2229678 -> ../../nvme0n1p2
│   ├── d4560004-ae55-4c3c-adee-1729a188417e -> ../../nvme0n1p3
│   └── e556663a-ed3f-4d5b-b69e-0abe9158c16b -> ../../nvme0n1p1
├── by-path
│   ├── pci-0000:03:00.0-nvme-1 -> ../../nvme0n1
│   ├── pci-0000:03:00.0-nvme-1-part1 -> ../../nvme0n1p1
│   ├── pci-0000:03:00.0-nvme-1-part2 -> ../../nvme0n1p2
│   └── pci-0000:03:00.0-nvme-1-part3 -> ../../nvme0n1p3
└── by-uuid
    ├── 027d32d8-de03-40d5-ad80-453618bd71a8 -> ../../nvme0n1p2
    ├── 06206a20-a62a-498e-a8d8-cd126855b435 -> ../../dm-2
    ├── 6cc2a9b5-50dd-4786-b0da-8239c67ed9ee -> ../../dm-1
    ├── 7F70-2AC1 -> ../../nvme0n1p1
    └── a00558ae-7865-4baa-b186-df0aaf2ab482 -> ../../dm-0
```

```
# lsblk
NAME            MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
nvme0n1         259:0    0 238.5G  0 disk
├─nvme0n1p1     259:1    0   200M  0 part /boot/efi
├─nvme0n1p2     259:2    0     1G  0 part /boot
└─nvme0n1p3     259:3    0 237.3G  0 part
  ├─fedora-root 253:0    0    50G  0 lvm  /
  ├─fedora-swap 253:1    0   7.5G  0 lvm  [SWAP]
  └─fedora-home 253:2    0 179.8G  0 lvm  /home
```

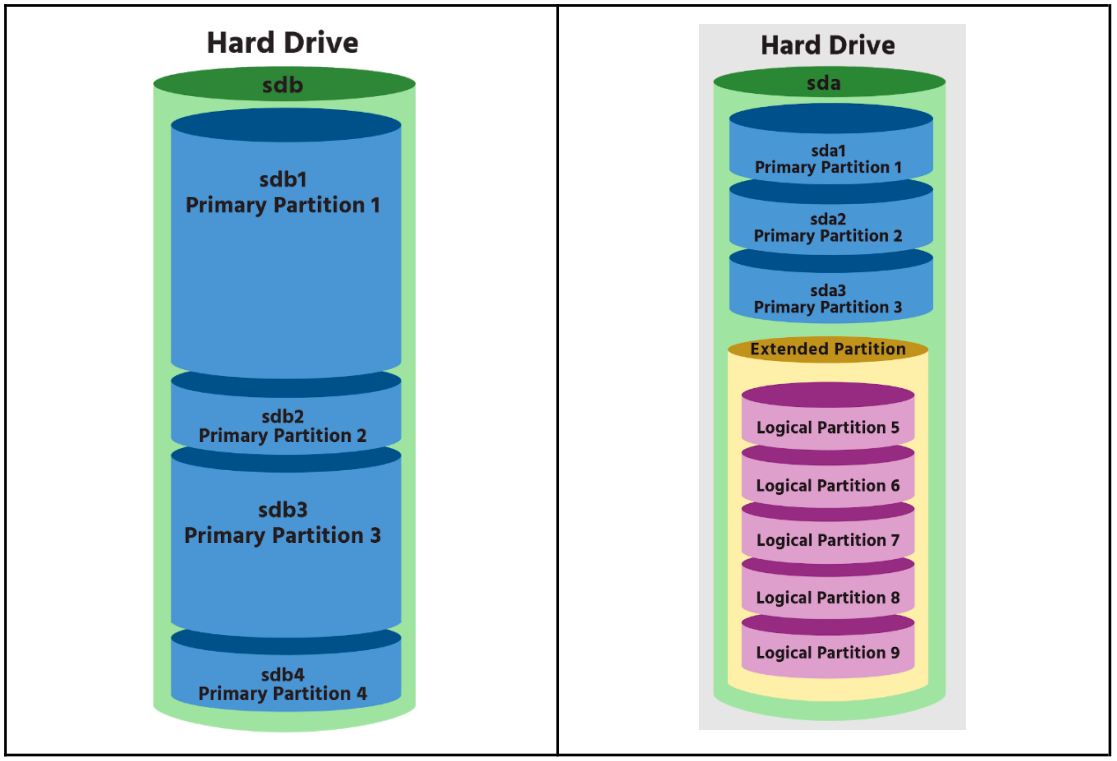
```
# blkid
/dev/nvme0n1p1: SEC_TYPE="msdos" UUID="7F70-2AC1" BLOCK_SIZE="512" TYPE="vfat"
PARTLABEL="EFI System Partition" PARTUUID="e556663a-ed3f-4d5b-b69e-0abe9158c16b"
/dev/nvme0n1p2: UUID="027d32d8-de03-40d5-ad80-453618bd71a8" BLOCK_SIZE="4096"
TYPE="ext4" PARTUUID="11ae4b81-2f10-4a68-a7cb-4c7fd2229678"
/dev/nvme0n1p3: UUID="VgZxLA-HigE-eyyU-DBHj-XbyP-e2gF-LEzdtR" TYPE="LVM2_member"
PARTUUID="d4560004-ae55-4c3c-adee-1729a188417e"
/dev/mapper/fedora-root: UUID="a00558ae-7865-4baa-b186-df0aaf2ab482" BLOCK_SIZE="4096"
TYPE="ext4"
/dev/mapper/fedora-swap: UUID="6cc2a9b5-50dd-4786-b0da-8239c67ed9ee" TYPE="swap"
/dev/mapper/fedora-home: UUID="06206a20-a62a-498e-a8d8-cd126855b435" BLOCK_SIZE="4096"
TYPE="ext4"
```

MBR Master Boot Record

Historically, the number of partitions a system can have is limited by the **Master Boot Record (MBR)**. Recall that the MBR is usually contained within the first sector or 512 bytes of the disk and contains a bootloader program, as well as the partition table. The partition table contains the definition of each partition on the hard drive, including the starting cylinder, ending cylinder, and size.

- Forest sector: 512 Bytes. Partition table (440) and Boot Loader.

- 4 partitions maximum.
- One of them can be extended.
- Primary partitions (1-4), logical partitions (5-).
- 2TB. Bios + fdisk.



GPT GUID Partition Table

GUID Partition Table (GPT). A globally unique identifier (GUID) is a universally unique 128-bit number used to identify information on a computer system. This partitioning scheme was designed to replace the traditional MBR partitioning is available on systems that support the Unified Extensible Firmware Interface (UEFI).

- 128 partitions max.
- No extended and logical partitions.
- 9ZB.
- gdisk, parted

Filesystems types

Exist multiple filesystem types, each has pros and cons. The file system types are optimized for different purposes.

ext
ext2

ext3

Third Extended Filesystem.
Can be upgraded from existing ext2 filesystem without data loss. This filesystem performs journaling, which allows for data recovery after a crash.
Writes more to disk than ext2 because of journaling, making it slower. Does not support very large filesystems. Preallocates the number of inodes.

ext4

Fourth Extended Filesystem
Support for very large disk volumes and file sizes. Can operate with or without a journal. Backwards compatible with ext3 and ext2.
Not a huge improvement over ext3. No dynamic inode creation. Preallocates the number of inodes.

xfs

Extents Filesystem.
Works very efficiently with large files. Compatible with the IRIX operating system from SGI. Announced to be the default filesystem for RHEL 7.
The filesystem cannot be shrunk. Not preallocates the number of inodes.
(xfsprogs on Debian)

vfat

File Allocation Table
Supported by almost all operating systems. Commonly used for removable media.
Unable to support very large disks or files. Microsoft's intellectual property claims.

iso9660

ISO 9660
The International Organization for Standardization standard for optical disc media that is supported by almost all operating systems.
The International Organization for Standardization standard for optical disc media that is supported by almost all operating systems.

Linux filesystem components

The International Organization for Standardization standard for optical disc media that is supported by almost all operating systems.

Superbloc

At the beginning of the filesystem is an area called the superblock. This area is used to store important information about the filesystem, including the size of the filesystem, the type of filesystem, and which data blocks (where file data is stored) are available. The superblock is a key component of the filesystem; a corrupted superblock would make the filesystem inaccessible.

Group block

The filesystem is divided into smaller sections called groups. The group block serves to hold data about the group. Each group block also contains a backup copy of the superblock.

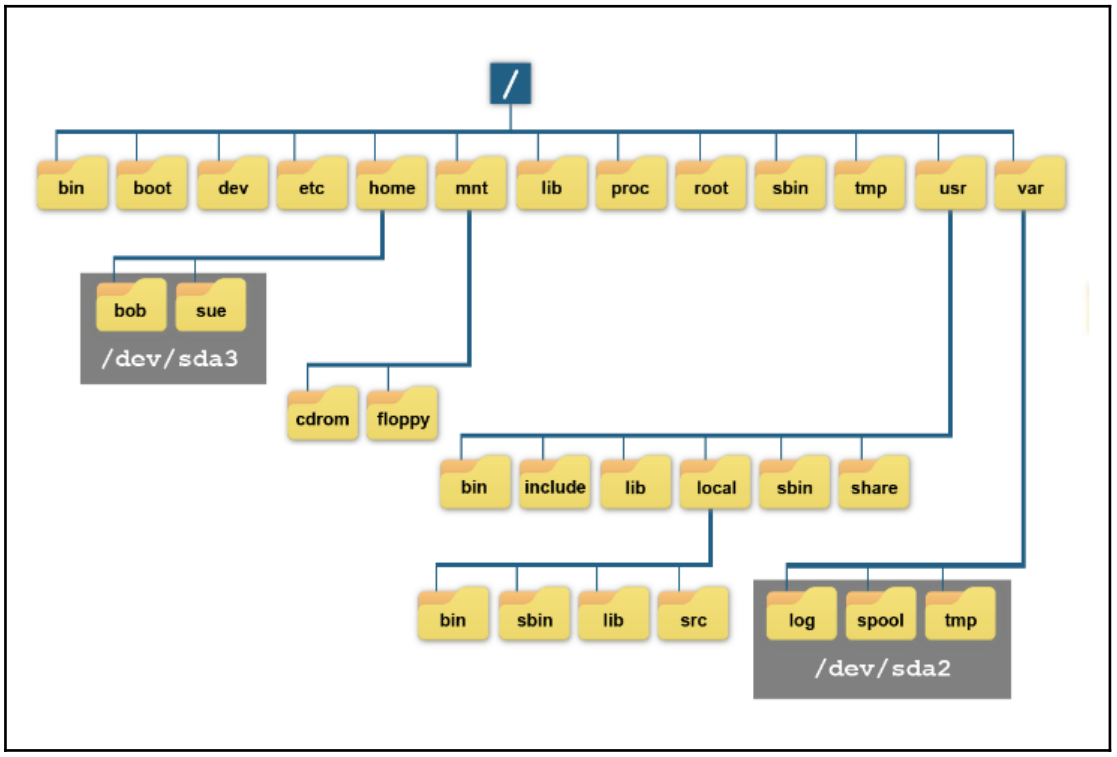
Inode table

Each file is assigned a unique inode number for the filesystem. This inode number is associated with a table that stores the file's metadata.

Linux uniq directory tree

In some operating systems each device has a letter C:, D: and a whole filesystem for each device. For example C: has a root directory and D: also has a root directory.

In linux only exists one file system, only one root. All the devices appear in the tree directory structure.



Partitioning considerations and layout

- Dual boot. Multiple partitions for multiple operating systems.
- Home directories. separe system information from home directories. System actualization independent. home backup independent.
- Logs (var, run) can grow.
- temporary directories /tmp, /var/tmp.

There are some requirements in the way partitions will map to the directories that they mount on. These requirements are documented in what is called the Filesystem Hierarchy Standard (FHS).

Filesystem Hierarchy Standard (FHS)

/

The root filesystem holds the files essential to the operation of the system. It must contain the following directories or symbolic links: bin, boot, dev, etc, lib, media, mnt, opt, sbin, srv, tmp, usr, and var.
500MiB-50GiB+ Depends on what is mounted separately.

/boot

The /boot directory contains the Linux kernel and the boot loader files.
500MiB-2GB

/home

The /home directory is where user home directories are normally created.
500MiB+ per user.

/tmp

The /tmp directory is used to create temporary files for the system and users. If this directory is too small, it may prevent applications from functioning correctly.
Minimum of 5 GB +. 500MiB+ per user actively logged in

/opt

The /opt directory is where third-party software is often installed. Some examples include Google Chrome and Google Earth.
100MiB+ Depends on how many packages are installed

/usr

The `/usr` directory contains the bulk of the operating system's files, including most of the commands and system software.
2GiB-10GiB+

/usr/local

This directory is used for locally installed software that should not be upgraded or updated with the operating system.
100MiB+. Size depends on local needs

/var

There are many directories that may have heavy activity under `/var` for services like mail, ftp, http, and printing.
100MiB+. Depending on the volume of activity

/boot/efi

The `/boot/efi` directory contains the Linux kernel and the boot loader files. It is typically set up automatically by the installer.
100MB-250MB.

swap

Swap is virtual memory that is not mounted on a directory. This virtual memory is used when the actual memory of the system is low. On systems with large amounts of memory, swap is less important.

Amount of RAM in the system	Recommended swap space	Recommended swap space if allowing for hibernation
⩽ 2 GB	2 times the amount of RAM	3 times the amount of RAM
> 2 GB – 8 GB	Equal to the amount of RAM	2 times the amount of RAM
> 8 GB – 64 GB	At least 4 GB	1.5 times the amount of RAM
> 64 GB	At least 4 GB	Hibernation not recommended

Create Partitions

- Create partitions during the installation process, using the installation assistant.
- Create partitions manually after the installation.

Create partitions during the installation

Typical options:
- Use all space
- Replace existing Linux systems
- Shrink current system
- Use free space
- Create custom layout

Partition options:
- Mount point
- Filesystem type
- Size
- Label

Create partitions manually

fdisk

The most common command line tool for editing the partition tables on disks is called ***fdisk***. This command can be used to create, modify, and list the partitions on a hard drive.

```
# fdisk -l
Disk /dev/nvme0n1: 238.49 GiB, 256060514304 bytes, 500118192 sectors
Disk model: SAMSUNG MZALQ256HAJD-000L2
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: BFA8875E-2F5D-4504-812B-618CFD968EBD

Device           Start       End   Sectors   Size Type
/dev/nvme0n1p1    2048    411647    409600   200M EFI System
/dev/nvme0n1p2  411648   2508799   2097152     1G Linux filesystem
/dev/nvme0n1p3 2508800 500117503 497608704 237.3G Linux LVM
```

Displays:

- device
- start sector
- End sector
- Sectors (number of)
- Size
- Type / Id (filesystem type / Id)

```
# fdisk /dev/nvme0n1

Welcome to fdisk (util-linux 2.35.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/nvme0n1: 238.49 GiB, 256060514304 bytes, 500118192 sectors
Disk model: SAMSUNG MZALQ256HAJD-000L2
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: BFA8875E-2F5D-4504-812B-618CFD968EBD

Device           Start       End   Sectors   Size Type
/dev/nvme0n1p1    2048    411647    409600   200M EFI System
/dev/nvme0n1p2  411648   2508799   2097152     1G Linux filesystem
/dev/nvme0n1p3 2508800 500117503 497608704 237.3G Linux LVM
```

```
Command (m for help): m

Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
```

```
   w   write table to disk and exit
   x   extra functionality (experts only)
```

Actions:

- fdisk -l
- fdisk /dev/sda
- m man
- p print
- n new
- primary / extended / logical
- d delete
- t toggle (L list)
- w write
- q quit

After altering the partition table is necessary for the system to reload it using ***partprobe*** or kpartx.

gdisk

The GPT disks use a newer type of partitioning, which allows the user to divide the disk into more partitions than what MBR supports. The tool for managing GPT disks is gdisk.

```
$ sudo gdisk /dev/sdb1
GPT fdisk (gdisk) version 1.0.3

Partition table scan:
MBR: not present
BSD: not present
APM: not present
GPT: not present

Creating new GPT entries.

Command (? for help): ?
b       back up GPT data to a file
c       change a partition's name
d       delete a partition
i       show detailed information on a partition
l       list known partition types
n       add a new partition
o       create a new empty GUID partition table (GPT)
p       print the partition table
q       quit without saving changes
r       recovery and transformation options (experts only)
s       sort partitions
t       change a partition's type code
v       verify disk
w       write table to disk and exit
x       extra functionality (experts only)
?       print this menu
```

Actions:

- n new

- p print
- v verify
- o delete all and create a new one
- s sort
- w write
- q quit

GNU parted

The GNU Parted program includes the parted command line tool and the gparted graphical interface tool. GNU Parted will non-destructively resize a partition as well as the filesystem on top of it.

```
# parted --help
Usage: parted [OPTION]... [DEVICE [COMMAND [PARAMETERS]...]...]
Apply COMMANDs with PARAMETERS to DEVICE.  If no COMMAND(s) are given, run in
interactive mode.

OPTIONs:
  -h, --help                     displays this help message
  -l, --list                     lists partition layout on all block devices
  -m, --machine                  displays machine parseable output
  -s, --script                   never prompts for user intervention
  -v, --version                  displays the version
  -a, --align=[none|cyl|min|opt]  alignment for new partitions

COMMANDs:
  align-check TYPE N                        check partition N for TYPE(min|opt)
        alignment
  help [COMMAND]                           print general help, or help on
        COMMAND
  mklabel,mktable LABEL-TYPE               create a new disklabel (partition
        table)
  mkpart PART-TYPE [FS-TYPE] START END     make a partition
  name NUMBER NAME                         name partition NUMBER as NAME
  print [devices|free|list,all|NUMBER]     display the partition table,
        available devices, free space, all found partitions, or a particular
partition
  quit                                     exit program
  rescue START END                         rescue a lost partition near START
        and END
  resizepart NUMBER END                    resize partition NUMBER
Output Omitted...
```

```
#  parted /dev/sdb print

# parted /dev/sdb mklabel msdos

# parted /dev/sdb mkpart primary 0% 50%
```

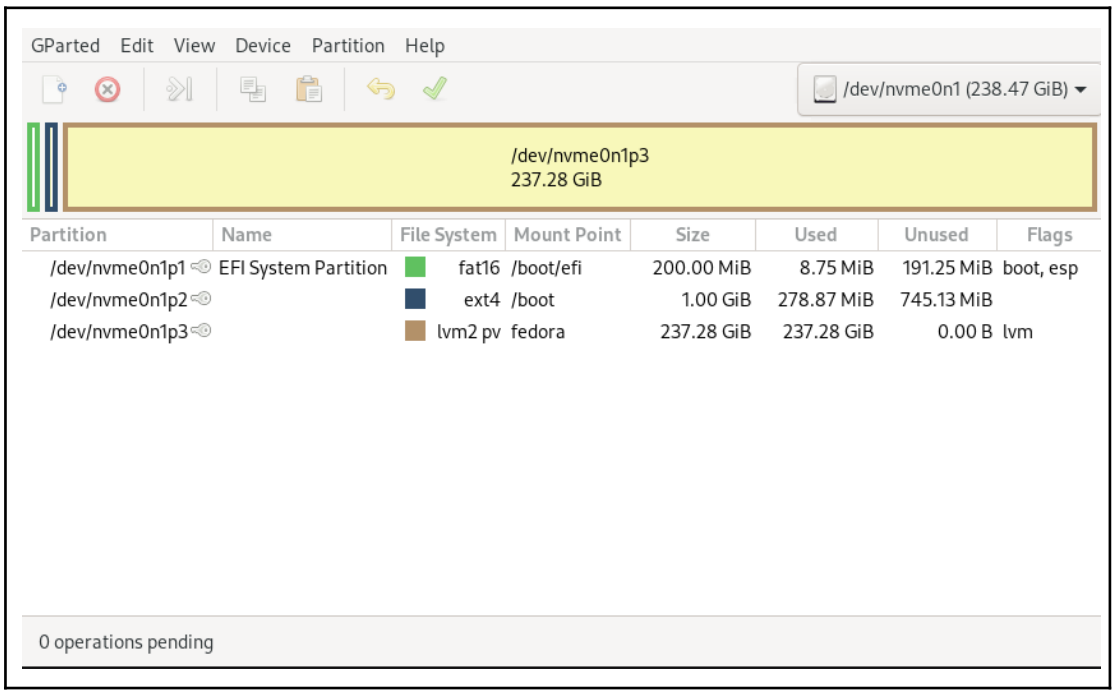
- Can be used in the command line.
- can be used interactive
- Actions:
  - mklabel msdos
  - mklabel gpt
  - mkpart
  - print
  - print free

gparted

GNU parted graphical tool. Installable on most systems. Can not modify the mounted partitions.

```
# apt-get install gparted

# gparted &
```



Creating a filesystem

mkfs

The mkfs command can be used to create a filesystem. Is a front end to each filesystem executable: mkfs.ext4, mkfs.vat…

Options:

- -t fstype
- -b block size
- -N number of inodes
- -m % system reserved
- Usually root reserved space: 5%

Create / active swap

Swap space can be a swap partition or a swap file. Swap partition is the method recommended.

- partition
- file

Commands:

- mkswap
- swapon
- swapoff
- swapon -a
- swapon -s

Practical exercises

Create a raw image

To practice partitions and filesystems let's create a virtual disk to practice.

- Create a raw image using dd
- Locate a free loop device
- Associate the raw image with the loop
- Now /dev/loop0 is a new device
- At the end of the practice, detach the raw image from the loop.

```
# dd if=/dev/zero of=disk.img bs=1k count=2M status=progress

# losetup -f
/dev/loop1

# losetup  /dev/loop0 disk.img

# blkid

# lsblk
```

```
# fdisk /dev/loop0
```

```
# gdisk /dev/loop0
```

```
# prated /dev/loop0
```

```
# losetup -d /dev/loop0
```

Format partitions

To practice formatting partitions let's use the raw.img disk in a Virtual Machine. For example an Debian 11 Bullseye nocloud image:

- Download an Debian 11 nocloud VM.
- Start the virtual machine using libvirtd, virt-manager.
- Define the amount of ram and attach the raw disk.
- Use the new disk as /dev/sdb to practice.

Debian Cloud images 11-Bullseye list

debian-11-nocloud-amd64.qcow2

```
# qemu-system-x86_64 -hda debian-11-nocloud-amd64.qcow2 -m 2048  -hdb disk.img
```

Example exercices

[Partitions]

1. Which is the name of the third partition of the second disk?
2. Which is the name of the second logical partition of the first disc?
3. Which fstype is created by default using mkfs?
4. An ext2 fstype with journal which fstype is?
5. Which is the usual root reserved space in a filesystem?
6. List all the devices and names using the tree command.
7. List all the devices using lsblk and blkid.
8. List all the partitions using fdisk.
9. Which is the linux swap ID?
10. Which is the ext4 partition UUID?
11. Observe the system partitions using gparted

[Real practice]

12. Create the partitions layout: sda1, sda5, sda6, sda7
13. Create the partition layout: sda1, sda2, sda3, sda4

14. Assign an ext4 fstype to a partition.
15. Assign an vfat type to a partition.
16. Assign an ntfs type to a partition
17. Assign an xfs type to a partition.

18. Create a swap partition.

19. Activate the swap partition.
20. Show all swap partitions.

[Installation]

21. Install a Fedora 32 system.
22. Install a Debian 11 Bullseye system.
23. Install a CenOS 8 system.

[Others]

24. LPI Exercices 104.1 Create partitions and filesystems