

LPI 102.3- Control mounting and unmounting filesystems

Curs 2021 - 2022

ASIX M01-ISO

Control mounting and umounting filesystems	2
Description	2
Mounting filesystems	2
Automatically mounting filesystems on boot: fstab	7
Systemd mount units	9
Examples	10
Example exercises	10

Control mounting and unmounting filesystems

Description

Key concepts:

- ❑ Manually mount and unmount filesystems.
- ❑ onfigure filesystem mounting on bootup.
- ❑ Configure user mountable removable filesystems.
- ❑ Use of labels and UUIDs for identifying and mounting file systems.
- ❑ Awareness of systemd mount units.

Commands anf files:

- ❑ /etc/fstab
- ❑ /media/
- ❑ mount
- ❑ umount
- ❑ blkid
- ❑ lsblk
- ❑ lsof
- ❑ fuser

Mounting filesystems

Recall the three steps in the process of making and using filesystems:

1. Divide the hard drive into partitions.
2. Format each partition with a filesystem.
3. Mount the formatted partitions onto the directory tree of an existing filesystem.

Show options:

- mount. Shows all the mounted filesystems
- mount -t fstype show all the mounted fstype filesystems

device	mountpoint	fstype	options
--------	------------	--------	---------

<pre>\$ mount -t ext4 /dev/mapper/fedora-root on / type ext4 (rw,relatime,seclabel) /dev/nvme0nlp2 on /boot type ext4 (rw,relatime,seclabel) /dev/mapper/fedora-home on /home type ext4 (rw,relatime,seclabel) \$ mount -t proc</pre>			
--	--	--	--

```
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)

$ mount -t tmpfs
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,mode=755,inode64)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,seclabel,inode64)
tmpfs on /run/user/1001 type tmpfs
(rw,nosuid,nodev,relatime,seclabel,size=764812k,mode=700,uid=1001,gid=1001,inode64)

# mount
```

Manual mount / umount

- directory mountpoint must exist
- contents of the original directory will be masked (not visible until unmounted)
- /mnt usual mount point
- /media mount point for removable media used by graphical desktops.
- Options
 - -t fstype
 - -o options
 - options: rw, ro, remount, user, users, noauto, loop
 - fstype: ext4 etx3, xfs, vfat, msdos, auto
- umount
 - umount device
 - umount mountpoint
 - umount device mountpoint

```
# mount /dev/sdb1 /imatges
```

Example:

- create raw virtual partitions
- format the partitions
- mount/umount the partitions

```
#1
# dd if=/dev/zero of=linux.img bs=1k count=200K
204800+0 records in
204800+0 records out
209715200 bytes (210 MB, 200 MiB) copied, 0.362003 s, 579 MB/s

# dd if=/dev/zero of=windows.img bs=1k count=200K
204800+0 records in
204800+0 records out
209715200 bytes (210 MB, 200 MiB) copied, 0.258256 s, 812 MB/s

# dd if=/dev/zero of=swap.img bs=1k count=200K
204800+0 records in
204800+0 records out
209715200 bytes (210 MB, 200 MiB) copied, 0.248884 s, 843 MB/s

# dd if=/dev/zero of=xfs.img bs=1k count=200K
204800+0 records in
204800+0 records out
209715200 bytes (210 MB, 200 MiB) copied, 0.253501 s, 827 MB/s
```

```
#2
# mkfs -t ext4 linux.img
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: c2e64874-2258-492d-8954-0fa7643b8c10
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

# mkfs -t vfat windows.img
mkfs.fat 4.1 (2017-01-24)

# mkswap swap.img
mkswap: swap.img: insecure permissions 0644, 0600 suggested.
Setting up swapspace version 1, size = 200 MiB (209711104 bytes)
no label, UUID=6b9f8511-101d-4aa9-889e-953ada153530

# chmod 600 swap.img

# mkswap swap.img
mkswap: swap.img: warning: wiping old swap signature.
Setting up swapspace version 1, size = 200 MiB (209711104 bytes)
no label, UUID=e6a213d6-67fa-4ec0-928c-b0f9b493756e

# mkfs -t xfs xfs.img
meta-data=xfs.img          isize=512    agcount=4, agsize=12800 blks
                           =          sectsz=512    attr=2, projid32bit=1
                           =          crc=1         finobt=1, sparse=1, rmapbt=0
                           =          reflink=1
data                       =          bsize=4096    blocks=51200, imaxpct=25
                           =          sunit=0       swidth=0 blks
naming                     =version 2           bsize=4096    ascii-ci=0, ftype=1
log                       =internal log         bsize=4096    blocks=1368, version=2
                           =          sectsz=512    sunit=0 blks, lazy-count=1
realtime =none             extsz=4096    blocks=0, rtextents=0
```

```
#3
# file *.img
disk.img: DOS/MBR boot sector; partition 1 : ID=0xee, start-CHS (0x0,0,2), end-CHS
(0x15,21,16), startsector 1, 4194303 sectors
linux.img: Linux rev 1.0 ext4 filesystem data,
UUID=c2e64874-2258-492d-8954-0fa7643b8c10 (extents) (64bit) (large files) (huge files)
swap.img: Linux/i386 swap file (new style), version 1 (4K pages), size 51199 pages,
no label, UUID=e6a213d6-67fa-4ec0-928c-b0f9b493756e
windows.img: DOS/MBR boot sector, code offset 0x3c+2, OEM-ID "mkfs.fat", sectors/cluster
8, reserved sectors 8, root entries 512, Media descriptor 0xf8, sectors/FAT 200,
sectors/track 32, heads 64, sectors 409600 (volumes > 32 MB), serial number 0xb3fa37cf,
unlabeled, FAT (16 bit)
xfs.img: SGI XFS filesystem data (blksize 4096, inosize 512, v2 dirs)
```

```
#4
# mount linux.img /mnt/
# ls /mnt/
lost+found

# echo "linux fs" > /mnt/message.txt

# mount -t ext4
/dev/mapper/fedora-root on / type ext4 (rw,relatime,seclabel)
/dev/nvme0n1p2 on /boot type ext4 (rw,relatime,seclabel)
/dev/mapper/fedora-home on /home type ext4 (rw,relatime,seclabel)
/home/images/VM/linux.img on /mnt type ext4 (rw,relatime,seclabel)
```

```
# mount windows.img /mnt

# mount -t vfat
/dev/nvme0n1p1 on /boot/efi type vfat
(rw,relatime,fmask=0077,dmask=0077,codepage=437,iocharset=ascii,shortname=winnt,errors=r
emount-ro)
/home/images/VM/windows.img on /mnt type vfat
(rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=ascii,shortname=mixed,errors=r
emount-ro)

# umount windows.img
```

```
# mount -t xfs -o ro xfs.img /mnt/

# mount -t xfs
/home/images/VM/xfs.img on /mnt type xfs
(ro,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)

# echo "fstype xfs" > /mnt/message.txt
-bash: /mnt/message.txt: Read-only file system

# umount /mnt
```

```
# swapon swap.img

# swapon -s
Filename                                Type              Size              Used              Priority
/dev/dm-1                               partition         7811068 5640      -2
/home/images/VM/swap.img                 file              204796 0             -3

# swapoff swap.img

# swapon -s
Filename                                Type              Size              Used              Priority
/dev/dm-1                               partition         7811068 4868      -2
```

```
#5
# create an iso image
$ mkisofs -o lpi.iso .

$ mv lpi.iso /tmp/

# mount -t iso9660 /tmp/lpi.iso /mnt/
mount: /mnt: WARNING: device write-protected, mounted read-only.

# tree -L 1 /mnt/
/mnt/
├── 101_syst
├── 102_linu
├── 103_gnu_
├── 104_devi
├── data_sou
├── _git
├── howto_lp.pdf
├── question
├── readme.md
├── resource.md
└── workmap.md

# umount /mnt
```

The **loop** option to the mount command is used to mount special filesystems that are stored within a file. These files have the extension of .img or .iso

```
# 6
# mount -t ext4 -o loop linux.img /mnt
# umount /mnt
```

```
# mount -t iso9660 -o loop /tmp/lpi.iso /mnt/
mount: /mnt: WARNING: device write-protected, mounted read-only.
# umount /mnt
```

Manual Umount filesystems

- Filesystem can be detached, removed directly. First they had to be unmounted. Direct unplug removable devices may cause disk damage.
- Can't be unmounted if the active directory is in the device.
- Can't be unmounted if processes use the device.

```
#7
# mount -t ext4 linux.img /mnt

# cd /mnt/

# ls
lost+found  message.txt

# umount /mnt
umount: /mnt: target is busy.
```

```
#8
# lsof /mnt
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs
Output information may be incomplete.
lsof: WARNING: can't stat() fuse.portal file system /run/user/1001/doc
Output information may be incomplete.
COMMAND  PID USER  FD   TYPE DEVICE SIZE/OFF NODE NAME
bash     16214 root   cwd   DIR   7,2    1024    2 /mnt
lsof     17419 root   cwd   DIR   7,2    1024    2 /mnt
lsof     17420 root   cwd   DIR   7,2    1024    2 /mnt
```

```
#9
# fuser -v /mnt/

          USER      PID ACCESS COMMAND
/mnt:
          root      kernel mount /mnt
          root      16214 ..c.. bash
```

Fuser codes:

- c The process is using the mount point or a subdirectory as its current directory.
- e The process is an executable file that resides in the mount point structure.
- f The process has an open file from the mount point structure.
- F The proc has an open file from the mount point structure that it is writing to.
- r The process is using the mount point as the root directory.
- m The process is a mmap'ed file or shared library.

```
#10
# cd ..

# umount /mnt
```

Automatically mounting filesystems on boot: fstab

The `/etc/fstab` file is used to configure what filesystems will be mounted automatically at boot. It can also be used to activate swap devices automatically. Root privileges are required to make changes to the `/etc/fstab` file.

Device identifiers:

- device name
- device UUID
- device label

```
#11
# 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"
/dev/loop0p1: UUID="ff9408dc-9b60-417c-af70-9fc530986562" BLOCK_SIZE="4096" TYPE="ext4"
/dev/loop1: BLOCK_SIZE="2048" UUID="2021-10-26-17-26-48-00" LABEL="CDROM" TYPE="iso9660"

# tree /dev/disk
```

```
#12
# mount linux.img /mnt/

# losetup
NAME          SIZELIMIT OFFSET AUTOCLEAR RO BACK-FILE          DIO LOG-SEC
/dev/loop0    0          0          1  0 /home/images/VM/linux.img  0      512
#

# e2label /dev/loop0 Linux

# e2label /dev/loop0
Linux

# umount /mnt
```

```
#13
# mount xfs.img /mnt/

# xfs_admin -L mylabel /dev/loop0
label = "mylabel"

# blkid -t xfs

# umount /mnt
```

fstab

Filesystem tabulation file. Defines the filesystems to mount on startup.

device	mountpoint	fstype	options	dump	check
--------	------------	--------	---------	------	-------

Attributes:

- **device**. Identify the device. Can be device name, UUID or LABEL.
- **mountpoint**. Filesystem directory where to mount the device.
- **fstype**. identify the fstype. Can be auto to autodetect.
- **options**: options to apply to the filesystem. defaults, rw, ro, user, users, noauto...
- default options (and opposite):
 - **rw** Allow reading and writing. **ro**.
 - **suid** Allow suid executes **nosuid**
 - **dev** Allow device files **nodev**
 - **exec** Allow executable files **noexec**
 - **auto** Automatically mount **noauto**
 - **nouser** Prevent ordinary users from mounting or unmounting the partition. Using the user option allows non-root users to mount that particular device, which is helpful for removable media. **user users**
 - **async** All writes should be asynchronous **sync**
 - **relatime** Only update access time on file access if the file has been modified or its metadata changed since last access **N/A**
- **dump**. (obsolete) The purpose of this field is to tell an administrator which filesystems should be backed up when using the dump command.
- **check**. determine the order in which the filesystems will be checked by the fsck utility during system boot. 0 no check. 1 checks before 2, etc.

```
#14
# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Wed Apr  7 10:55:55 2021
#
# 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/fedora-root / ext4 defaults 1 1
UUID=027d32d8-de03-40d5-ad80-453618bd71a8 /boot ext4 defaults 1 2
UUID=7F70-2AC1 /boot/efi vfat umask=0077,shortname=winnt 0 2
/dev/mapper/fedora-home /home ext4 defaults 1 2
/dev/mapper/fedora-swap swap swap defaults 0 0
```

When defined in /etc/fstab filesystems can be mounted and unmounted easily:

- only specify device or mountpoint (not necessary both).
- options defined in /etc/fstab are applied
- user can not mount unless user or users options are specified.

```
#15
# cat /etc/fstab
/dev/mapper/fedora-root / ext4 defaults 1 1
UUID=027d32d8-de03-40d5-ad80-453618bd71a8 /boot ext4 defaults 1 2
```



```

UUID=7F70-2AC1          /boot/efi          vfat      umask=0077,shortname=winnt 0 2
/dev/mapper/fedora-home /home          ext4      defaults          1 2
/dev/mapper/fedora-swap swap          swap      defaults          0 0
/home/images/linux.img  /mnt          ext4      defaults,ro       0 0

# mount -a

# mount -t ext4
/dev/mapper/fedora-root on / type ext4 (rw,relatime,seclabel)
/dev/nvme0n1p2 on /boot type ext4 (rw,relatime,seclabel)
/dev/mapper/fedora-home on /home type ext4 (rw,relatime,seclabel)
/home/images/VM/linux.img on /mnt type ext4 (ro,relatime,seclabel)

# umount /mnt

```

Define swap in /etc/fstab

- Usually the swap partitions are defined in the /etc/fstab file.

```

/device swap          swap      defaults          0 0

```

Systemd mount units

The systemd-mount utility is the mechanism that systemd uses to create and start a transient .mount file, destroy a transient .mount file, or to start an .automount unit of the filesystem.

Systemd converts all entries in fstab in systemd-mount units

Root can create systemd-mount units.

```

# 16
# locate .mount

# cat /usr/lib/systemd/system/proc-fs-nfsd.mount
[Unit]
Description=NFSD configuration filesystem

[Mount]
What=nfsd
Where=/proc/fs/nfsd
Type=nfsd

# cat /usr/lib/systemd/system/tmp.mount
# SPDX-License-Identifier: LGPL-2.1+
#
# This file is part of systemd.
#
# systemd is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation; either version 2.1 of the License, or
# (at your option) any later version.

[Unit]
Description=Temporary Directory (/tmp)
Documentation=https://systemd.io/TEMPORARY_DIRECTORIES

```

```
Documentation=man:file-hierarchy(7)
Documentation=https://www.freedesktop.org/wiki/Software/systemd/APIFileSystems
ConditionPathIsSymbolicLink=!/tmp
DefaultDependencies=no
Conflicts=umount.target
Before=local-fs.target umount.target
After=swap.target

[Mount]
What=tmpfs
Where=/tmp
Type=tmpfs
Options=mode=1777,strictatime,nosuid,nodev
```

Examples

```
# mount
# mount -a
# mount -t ext4
# mount -t tmpfs
```

```
# mount /dev/sdb1 /home/images
# mount -o loop data.img /mnt
# mount -o ro LABEL=DATA /opt/data
# mount -o sync UUID=027d32d8-de03-40d5-ad80-453618bd71a8 /var/log
```

```
# umount /dev/sdb1
# umount /mnt
# umount /dev/sda5 /mnt
```

```
# du -h
# du -h -text4

# lsof /mnt
# fuser -v /mnt
```

Example exercises

1. Create 4 raw images and format them as: ext4, vfat, xfs and swap
2. Assign each partition a label.
3. Create 3 directories in /mnt with the names: linux, windows, xfs.
4. mount the linux raw image to the linux directory read only.
5. mount the vfat partition to the windows directory using its label.
6. mount the xfs partition to the xfs directory using its UUID.
7. list all the mounted systems (mount and blkid).

8. umount all the raw images.
9. Modify the fstab file mounting the swap partition automatically.
10. Modify the fstab file mounting the raw image linux to /mnt automatically.
11. Mount all the defined mounts in fstab. Verify it.
12. Comment the last two exercise entries in the fstab
13. LPI Exercises [104.3 Control mounting and unmounting of filesystems](#)