

# HowTo ASIX LVM

*Logical Volume Manager*

<b>Documentació</b>	<b>1</b>
<b>Logical Volume Manager</b>	<b>2</b>
<b>Exercici Pràctic 1: utilitzar LVM</b>	<b>3</b>
<b>Exercici Pràctic 2: modificacions en calent</b>	<b>8</b>
Primera part	9
Segona Part	11
Tercera Part	12
Quarta Part	14
<b>Entorn Gràfic</b>	<b>16</b>

## Aprenentatges Treballats

---

1. Components LVM.
  - a. PV Physical Volumen
  - b. VG Volume Group
  - c. LV Logical Volume
2. Creació de sistemes de fitxers usant LVM.
3. Modificació de LVM
  - a. Afegir noves unitats / particions.
  - b. Incrementar / decrementar l'espai dels volums lògics.
  - c. Incrementar / decrementar l'espai dels grups de volum.
  - d. Moure / alliberar particions. Eliminar-les.
4. RAID i LVM.
  - a. Creació de sistema amb RAID i LVM.

## Documentació

---

Aquest document ha estat elaborant utilitzant com a eina de treball un sistema GNU/Linux Fedora 20.

- Apunts @edt [ASIX-M11](#)
- LVM
  - [objectius-lvm](#)
  - 
  - [HowTo-ASIX-LVM](#)

- Documentació de les pàgines man de les ordres.
- Fedora Documentation: Fedora 14, Storage Administration Guide, [Chapter 3: LVM \(Logical Volume Manager\)](#)
- Fedora Documentation: Fedora 20, Installation Guide, [Appendix D: Understanding LVM](#)
- Fedora Documentation: Fedora 20, Installation Guide, [Chapter 9.14: Creating a custom partition Layout](#)
- Red Hat Documentation: [Chapter 4: LVM Administration with CLI commands](#)

#### # apropos lvm

```

clvm_d_selinux (8) - Security Enhanced Linux Policy for the clvm_d processes
lvm (8)            - LVM2 tools
lvm.conf (5)       - Configuration file for LVM2
lvm_selinux (8)    - Security Enhanced Linux Policy for the lvm processes
lvmchange (8)      - change attributes of the logical volume manager
lvmconf (8)        - LVM configuration modifier
lvmdiskscan (8)    - scan for all devices visible to LVM2
lvmdump (8)        - create lvm2 information dumps for diagnostic purposes
lvmsadc (8)        - LVM system activity data collector
lvmsar (8)         - LVM system activity reporter
perlvmesa (1)      - building and installing Perl for VM/ESA.
perlvms (1)        - VMS-specific documentation for Perl
pvcreate (8)       - initialize a disk or partition for use by LVM
pvresize (8)       - resize a disk or partition in use by LVM2

```

#### Ordres de gestió de Volumes Lògics:

```

# pv
pvchange pvcreate pvmove      pvresize pvscan
pvck     pvdisplay pvremove pvs

# vg
vgcfgbackup vgconvert    vgextend    vgmknodes vgs
vgcfgrestore vgcreate     vgimport    vgreduce  vgscan
vgchange     vgdisplay    vgimportclone vgremove  vgsplit
vgck         vgexport     vgmerge     vgrename

# lv
lvchange     lvdisplay    lvmchange   lvmdump   lvreduce   lvresize

```

lvconvert	lvextend	lvminconf	lvmsadc	lvremove	lvs
lvcreate	lv	lvmdiskscan	lvmsar	lvrename	lvscan

```
# system-config-lvm &
```

## Logical Volume Manager

Descripció de LVM extreta de Fedora Documentation/14 Storage Guide:

LVM (Logical Volume Management) partitions provide a number of advantages over standard partitions. LVM partitions are formatted as *physical volumes*. One or more physical volumes are combined to form a *volume group*. Each volume group's total storage is then divided into one or more *logical volumes*. The logical volumes function much like standard partitions. They have a file system type, such as ext4, and a mount point.

To understand LVM better, imagine the physical volume as a pile of *blocks*. A block is simply a storage unit used to store data. Several piles of blocks can be combined to make a much larger pile, just as physical volumes are combined to make a volume group. The resulting pile can be subdivided into several smaller piles of arbitrary size, just as a volume group is allocated to several logical volumes.

An administrator may grow or shrink logical volumes without destroying data, unlike standard disk partitions. If the physical volumes in a volume group are on separate drives or RAID arrays then administrators may also spread a logical volume across the storage devices.

You may lose data if you shrink a logical volume to a smaller capacity than the data on the volume requires. To ensure maximum flexibility, create logical volumes to meet your current needs, and leave excess storage capacity unallocated. You may safely grow logical volumes to use unallocated space, as your needs dictate.

## Exercici Pràctic 1: utilitzar LVM

Crear tres unitats físiques 'imaginaries' usant l'utilitat dd per generar espai de disc virtual. Assignar aquests fitxers a un dispositiu físic de loopback. És a dir, en lloc de crear tres particions de debò tipus /dev/sda2, /dev/sda3 i /dev/sda4 ens inventem les particions /dev/loop0, /dev/loop1 i /dev/loop2

```
# Crear les fitxers imatge
# dd if=/dev/zero of=disk01.img bs=1k count=100K
```

```

102400+0 registres llegits
102400+0 registres escrits
104857600 octets (105 MB) copiats, 0,676056 s, 155 MB/s
# dd if=/dev/zero of=disk02.img bs=1k count=100K
# dd if=/dev/zero of=disk03.img bs=1k count=100K

# Assignar-los al loopback
# losetup /dev/loop0 /opt/lvm/disk01.img
# losetup /dev/loop1 /opt/lvm/disk02.img
# losetup /dev/loop2 /opt/lvm/disk03.img

# losetup -a
/dev/loop0: [2053]:1217 (/opt/lvm/disk01.img)
/dev/loop1: [2053]:1218 (/opt/lvm/disk02.img)
/dev/loop2: [2053]:1220 (/opt/lvm/disk03.img)

```

Disposant de tres ‘troços’ d’espai d’emmagatzemament anem a crear un volum físic de cada un d’ells, és a dir, adaptar-los per ser usats com a emmagatzemament LVM. Aquest procés simplement adequa l’espai per a ser usat com a espai LVM i s’anomenent **Physical Volume**.

```

# pvcreate /dev/loop0
Writing physical volume data to disk "/dev/loop0"
Physical volume "/dev/loop0" successfully created

# pvcreate /dev/loop1 /dev/loop2
Writing physical volume data to disk "/dev/loop1"
Physical volume "/dev/loop1" successfully created
Writing physical volume data to disk "/dev/loop2"
Physical volume "/dev/loop2" successfully created

```

```

# pvdisplay /dev/loop0
"/dev/loop0" is a new physical volume of "100,00 MiB"
--- NEW Physical volume ---
PV Name           /dev/loop0
VG Name
PV Size           100,00 MiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           Z1vBu0-iZMk-A5v3-Sqh8-VUkX-c6vu-bIVlg9

```

Els espais d'emmagatzemament LVM, els **Physical Volume** s'agrupen per crear unitatgs d'emmagatzemament (equivalents a la visió que té el sistema de discs físics) anomenats **Volume Groups** o grups de volum.

En els següent proces es pot observar que dels tres volums físics s'utilitzarà els dos primers per crear un nou *Volume Group* anomenat **diskedt**.

```
# vgcreate diskedt /dev/loop0 /dev/loop1
Volume group "diskedt" successfully created

# vdisplay diskedt
--- Volume group ---
VG Name          diskedt
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 1
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          0
Open LV          0
Max PV           0
Cur PV          2
Act PV           2
VG Size          192,00 MiB
PE Size          4,00 MiB
Total PE         48
Alloc PE / Size  0 / 0
Free PE / Size   48 / 192,00 MiB
VG UUID          eXUDOP-9wqX-jXsE-ugt6-fVXv-rm2X-ee87Ng
```

És a dir, ara els espais d'emmagatzemament de 100M de loop0 i loop1 s'han ajuntat per crear un nou dispositiu que el sistema interpreta com un dispositiu físic de 200M anomenat **/dev/diskedt** (no apareix fins a no particionar-lo). Aquesta unitat és aproximadament de 192M, hi ha una pèrdua d'espai d'emmagatzemament degut a la necessitat de crear estructures de dades per a la gestió LVM. Sembla que el *Volume Group* consta de 48 PE de 4MB cada un, en total 192MB.

Observar que en el moment d'assignar els *Physical Volumes* loop0 i loop1 al *Volume Group* la seva descripció ha canviat. La següent taula mostra la diferència que mostra la descripció de /dev/loop0 abans i després de formar part del *Volume Group diskedt*.

```
# pvdisplay /dev/loop0 (abans)
"/dev/loop0" is a new physical volume of "100,00 MiB"
--- NEW Physical volume ---
PV Name          /dev/loop0
```

```

VG Name
PV Size      100,00 MiB
Allocatable  NO
PE Size      0
Total PE     0
Free PE      0
Allocated PE 0
PV UUID      Z1vBu0-iZMk-A5v3-Sqh8-VUkX-c6vu-bIVlg9

```

#### # pvdisplay /dev/loop0 (després)

```

--- Physical volume ---
PV Name      /dev/loop0
VG Name      diskedt
PV Size      100,00 MiB / not usable 4,00 MiB
Allocatable  yes
PE Size      4,00 MiB
Total PE     24
Free PE      24
Allocated PE 0
PV UUID      Z1vBu0-iZMk-A5v3-Sqh8-VUkX-c6vu-bIVlg9

```

Finalment igual que d'un dispositiu físic se'n pot fer particions lògiques, d'un dispositiu **Volume Group** se'n poden fer particions lògiques anomenades **Logical Volume**.

En l'exemple es faran dos particions lògiques una anomenada **sistema** de 50M i una altra anomenada **dades** de 150M. Fixeu-vos que de dos Physical Volumen de 100M s'ha creat un Volume Group diskedt de 200M que s'ha subdividit en dues particions Logical Volumen asimètriques.

#### # lvcreate -L 50M -n sistema /dev/diskedt

```

Rounding up size to full physical extent 52,00 MiB
Logical volume "sistema" created

```

#### # lvcreate -L 150M -n dades /dev/diskedt

```

Rounding up size to full physical extent 152,00 MiB
Volume group "diskedt" has insufficient free space (35 extents): 38 required.

```

#### # lvcreate -l100%FREE -n dades /dev/diskedt

```

Logical volume "dades" created

```

#### # lvdisplay /dev/diskedt/sistema

```

--- Logical volume ---
LV Path      /dev/diskedt/sistema
LV Name      sistema
VG Name      diskedt
LV UUID      OMPKNv-78L2-37y6-3kwO-U96L-3CXv-4OoS3l
LV Write Access  read/write
LV Creation host, time portatil.localdomain, 2015-02-06 13:02:17 +0100

```

```

LV Status      available
# open         0
LV Size        52,00 MiB
Current LE     13
Segments       1
Allocation     inherit
Read ahead sectors auto
- currently set to 256
Block device   253:0

```

#### # lvsdisplay /dev/diskedt/dades

```

--- Logical volume ---
LV Path        /dev/diskedt/dades
LV Name        daades
VG Name        diskedt
LV UUID        UTMphJ-dYlp-9g7W-bjWe-rs9H-BL7X-JNX0mG
LV Write Access read/write
LV Creation host, time portatil.localdomain, 2015-02-06 13:03:09 +0100
LV Status      available
# open         0
LV Size        140,00 MiB
Current LE     35
Segments       2
Allocation     inherit
Read ahead sectors auto
- currently set to 256
Block device   253:1

```

#### # tree /dev/disk

```

/dev/disk
|-- by-id
| |-- ata-FUJITSU_MHV2100AT_PL_NSA3T6329W69 -> ../../sda
| |-- ata-FUJITSU_MHV2100AT_PL_NSA3T6329W69-part1 -> ../../sda1
| .....
| |-- ata-FUJITSU_MHV2100AT_PL_NSA3T6329W69-part7 -> ../../sda7
| |-- ata-MATSHITADVD-RAM_UJ-841S -> ../../sr0
| |-- dm-name-diskedt-dades -> ../../dm-1
| |-- dm-name-diskedt-sistema -> ../../dm-0
| |--
dm-uuid-LVM-eXUDOP9wqXjXsEugt6fVXvrm2Xee87Ng0MPKNv78L237y63kwOU96L3C
Xv4OoS3I -> ../../dm-0
| `--
dm-uuid-LVM-eXUDOP9wqXjXsEugt6fVXvrm2Xee87NgUTMphJdYlp9g7WbjWers9HBL7
XJNX0mG -> ../../dm-1

```

#### # ll /dev/diskedt/

```

total 0
lrwxrwxrwx 1 root root 7 6 feb 13:03 daades -> ../../dm-1
lrwxrwxrwx 1 root root 7 6 feb 13:02 sistema -> ../../dm-0

```

Observar la quantitat de PE que utilitza cada Logical Volume, *sistema* n'utilitza 13 de 4MB que proporcionen 52MB d'emmagatzemament, tots ells en un mateix segment (Physical Volumen). En canvi *dades* utilitza 35 unitats d'assignació PE de 4MB proporcionant 140MB d'emmagatzemament i ocupa dos segments, és a dir, utilitza tots dos Physical Volumes.

Ara ja és tot a punt per poder formatar aquests volums lògics i poder-los integrar al sistema de fitxers muntant-los on es cregui oportú. Un cop muntats s'utilitzen com a aqualsevol altre element d'emmagatzemament.

```
# mkfs -t ext4 /dev/diskedt/dades
mke2fs 1.42.3 (14-May-2012)
Discarding device blocks: fet
Etiqueta del sistema de fitxers=
OS type: Linux
Mida del bloc=1024 (log=0)
Mida del fragment=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
35856 inodes, 143360 blocks
7168 blocks (5.00%) reserved for the super user
Bloc de dades inicial=1
Màxim de blocs del sistema de fitxers=67371008
18 grups de blocs
8192 blocs per grup, 8192 fragments per grup
1992 nodes-i per grup
Còpies de seguretat del superbloc desades en els blocs:
  8193, 24577, 40961, 57345, 73729

Allocating group tables: fet
Escriptura de les taules de nodes-i:fet
Creació del registre de transaccions (4096 blocs): fet
Escriptura de la informació dels

# mkfs -t ext4 /dev/diskedt/sistema
```

```
# Crear els punts de muntatge i muntar-los
# mkdir /mnt/dades
# mkdir /mnt/sistema
# mount /dev/diskedt/dades /mnt/dades/
# mount /dev/diskedt/sistema /mnt/sistema/

# Copiar-hi dades a cada directori
# cp -R /boot/* /mnt/dades/
# cp /boot/* /mnt/sistema/

# df -h
S. fitxers          Mida En ús Lliure %Ús Muntat a
```



```
...
/dev/mapper/diskedt-dades 132M 93M 33M 75% /mnt/dades
/dev/mapper/diskedt-sistema 47M 47M 0 100% /mnt/sistema
```

#### # blkid

```
/dev/loop0: UUID="Z1vBu0-iZMk-A5v3-Sqh8-VUkX-c6vu-bIVlg9" TYPE="LVM2_member"
/dev/loop1: UUID="xgHNN9-Q1Mx-0Ojp-03Km-GVDC-2iQf-2FXp7k"
TYPE="LVM2_member"
.....
/dev/mapper/diskedt-sistema: UUID="0addb5b2-d3c2-4767-ab48-1e6d0e0297a3"
TYPE="ext4"
/dev/mapper/diskedt-dades: UUID="dbd2a7c8-67b0-4cbb-bd7b-ade958f9c3a5"
TYPE="ext4"
```

S'ha copiat a dades i a sistema el directori boot, que ocupa completament l'espai de sistema (no hi cap) i un 75% de l'espai de dades. Observem la descripció de l'ocupació:

## Exercici Pràctic 2: modificacions en calent

El principal avantatge de la utilització de LVM és que en **calent** es poden modificar les composicions dels grups de volum i es poden fer **resize** dels volums lògics. Així per exemple en el cas anterior es pot:

- ampliar el *Volume Group diskedt* **afegint-hi** 100M procedents del /dev/loop2.
- si el *Volume Group diskedt* disposa de més espai lliure es pot **assignar** aquest espai als volums lògics *dades*, *sistema* o fins i tot crear un nou volum lògic.
- si no s'afegeix espai nou a diskedt també es pot redistribuir el seu espai lliure. És a dir, es pot repartir l'espai del Volume Group entre les seves dues particions lògiques de manera diferent, sense haver d'esborrar-les i crear de nou.
- en tot moment els volums lògics es poden **redimensionar**, ampliant-los o reduint-los (shrink), sempre que les dades que contenen hi càpiguen.

Aquest exercici s'estructura en quatre parts on es realitzaran les tasques següents:

1. Assignar 100M del loop2 al grup de volum *diskedt*. Afegir 30M d'aquest nou espai al volum lògic *sistema*.
2. Crear una nova partició lògica anomenada *services* de 60M amb l'espai sobrant del grup de volum *diskedt*.
3. Redimensinar un volum lògic reduint-lo, empetitint (*shrink*) el volum lògic *sistema*.
4. Amb l'espai alliberat en empetitir *sistema* i l'espai lliure sobrant del volum *diskedt*, ampliar l'espai del volum lògic *dades*.

## Primera part

Primerament s'integrarà el Physical Volumen loop2 de 100M al Volume Group diskedt, de manera que ara es simula un espai d'emmagatzemament de 300M, 150M assignats a dades, 50M assignats a 50M i la resta (100M) sense assignar.

```
# man vg
vgcfgbackup  vgck          vgdisplay    vgimport     vgmknodes    vgrename
vgcfgrestore vgconvert      vgexport     vgimportclone vgreduce     vgs
vgchange     vgcreate       vgextend     vgmerge      vgremove     vgscan
vgsplit

# vgextend diskedt /dev/loop2
Volume group "diskedt" successfully extended

# vgdisplay diskedt
--- Volume group ---
VG Name          diskedt
System ID
Format           lvm2
Metadata Areas   3
Metadata Sequence No 4
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          2
Open LV          2
Max PV           0
Cur PV          3
Act PV           3
VG Size          288,00 MiB
PE Size          4,00 MiB
Total PE         72
Alloc PE / Size  48 / 192,00 MiB
Free PE / Size   24 / 96,00 MiB
VG UUID          eXUDOP-9wqX-jXsE-ugt6-fVXv-rm2X-ee87Ng
```

A continuació es repartirà aquest espai lliure per assignar ampliant en 30M la partició sistema.

```
# lv
lvchange      lvcreate      lvextend     lvmdiskscan  lvmsadc
lvconvert     lvdisplay     lvm          lvmdump      lvmsar
lvremove      lvresize      lvscan       lvreduce     lvrename     lvs

# lvextend -L +30 /dev/diskedt/sistema /dev/loop2
Rounding up size to full physical extent 32,00 MiB
```

Extending logical volume sistema to **84,00 MiB**

Logical volume sistema successfully resized

#### # lvsdisplay /dev/diskedt/sistema

--- Logical volume ---

```
LV Path          /dev/diskedt/sistema
LV Name          sistema
VG Name          diskedt
LV UUID          0MPKNv-78L2-37y6-3kwO-U96L-3CXv-4OoS3l
LV Write Access   read/write
LV Creation host, time portatil.localdomain, 2015-02-06 13:02:17 +0100
LV Status        available
# open           1
LV Size          84,00 MiB
Current LE       21
Segments      2
Allocation       inherit
Read ahead sectors auto
- currently set to 256
Block device     253:0
```

Es pot observar que el *Logical Volumen* sistema s'ha incrementat en 30M, ara ocupa 21 unitats d'assignació PE de 4MB (21\*4=84). També es mostra que ara la partició lògica **dades** està formada per dos segments. Esta desada en dos *loop*, sabem del cert que el *loop2* proporciona 24 PE, però no sabem els 13 PE inicials de quin dels *loop0* o *loop1* prové.

Però fixeu-vos que passa encara amb les particions de dades i sistemes (que per cert, les havíem demuntat oi?):

#### # df -h

```
S. fitxers      Mida En ús Lliure %Ús Muntat a
devtmpfs        493M  0  493M  0% /dev
...
/dev/mapper/diskedt-dades 132M  93M   33M 75% /mnt/dades
/dev/mapper/diskedt-sistema 47M  47M   0 100% /mnt/sistema
```

# umount /mnt/dades

# umount /mnt/sistema

#### # resize2fs /dev/diskedt/sistema

resize2fs 1.42.3 (14-May-2012)

El sistema de fitxers a /dev/diskedt/sistema està muntat a /mnt/sistema; cal un canvi de mida en línia

old\_desc\_blocks = 1, new\_desc\_blocks = 1

The filesystem on /dev/diskedt/sistema is now 86016 blocks long.

# mount /dev/diskedt/dades /mnt/dades/

```
# mount /dev/diskedt/sistema /mnt/sistema/

# df -h
S. fitxers          Mida En ús Lliure  %Ús Muntat a
...
/dev/mapper/diskedt-sistema  78M  47M   27M  64% /mnt/sistema
/dev/mapper/diskedt-dades   132M  93M   33M  75% /mnt/dades
```

## Segona Part

Crear una nova partició lògica, *Logical Volumen* anomenada **services** de 60MB dins del *Volume Group diskedt*.

```
# lvcreate -L 60 -n services /dev/diskedt
Logical volume "services" created

# lvs /dev/diskedt/services
--- Logical volume ---
LV Path                /dev/diskedt/services
LV Name                 services
VG Name                diskedt
LV UUID                S88Jxf-VS2Z-xTYf-w70M-l4RF-0ohM-WL6aor
LV Write Access         read/write
LV Creation host, time portatil.localdomain, 2015-02-06 14:15:44 +0100
LV Status               available
# open                  0
LV Size                 60,00 MiB
Current LE              15
Segments                1
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device            253:2
```

Així doncs, ara el volum **diskedt** disposa de 300M repartits en 80M per a sistema (50M+30M), 150M per a dades i 60M per al nou volum lògic **services**. El volum encara té uns 10M lliures per assignar, es poden observar quants blocs PE són amb l'ordre `vgdisplay`. De fet sembla que només un bloc PE de 4MB està no assignat (cal recordar que en crear les volums es desaprofita espai per crear les estructures de dades necessàries per a LVM).

```
# vgs /dev/diskedt
--- Volume group ---
VG Name                diskedt
System ID
Format                 lvm2
Metadata Areas         3
```

```

Metadata Sequence No 6
VG Access      read/write
VG Status      resizable
MAX LV        0
Cur LV        3
Open LV        2
Max PV         0
Cur PV        3
Act PV         3
VG Size        288,00 MiB
PE Size        4,00 MiB
Total PE       72
Alloc PE / Size 71 / 284,00 MiB
Free PE / Size  1 / 4,00 MiB
VG UUID        eXUDOP-9wqX-jXsE-ugt6-fVXv-rm2X-ee87Ng

```

## Tercera Part

En aquest exemple es mostrarà com redimensionar un volum lògic per fer-lo més petit. Així si observem les dades següents:

```

/dev/mapper/diskedt-dades 132M 93M 33M 75% /mnt/dades
/dev/mapper/diskedt-sistema 78M 47M 27M 64% /mnt/sistema

```

observem que sistema ocupa 47MB i té disponibles 27MB. S'intentarà reduir la partició a 50MB i traspasar tot l'espai lliure ala partició de dades.

Primerament cal realitzar el procés de *shrink* per reduir la partició.

```

# umount /mnt/sistema
# umount /mnt/dades

# resize2fs /dev/diskedt/sistema 50M
resize2fs 1.42.3 (14-May-2012)
resize2fs: New size smaller than minimum (56904)

# resize2fs /dev/diskedt/sistema 56M
resize2fs 1.42.3 (14-May-2012)
Executeu «e2fsck -f /dev/diskedt/sistema» primer.

# e2fsck -f /dev/diskedt/sistema
e2fsck 1.42.3 (14-May-2012)
Pas 1: comprovació de nodes-i, blocs i mides
Pas 2: comprovació de l'estructura del directori
Pas 3: comprovació de la connectivitat dels directoris
Pas 4: comprovació dels nombres de referències
Pas 5: comprovació del resum de la informació del grup
/dev/diskedt/sistema: 23/20944 files (4.3% non-contiguous), 54623/86016 blocks

```

**# resize2fs /dev/diskedt/sistema 56M**

resize2fs 1.42.3 (14-May-2012)

Resizing the filesystem on /dev/diskedt/sistema to 57344 (1k) blocks.

The filesystem on /dev/diskedt/sistema is now 57344 blocks long.

Ara que el sistema de fitxers 'ha reduït a 56M (no es podia fer més petit perquè no hi cabien les dades) es pot procedir a reduir la mida del volum lògic.

**# lvreduce -L 56M -r /dev/diskedt/sistema**

fsck de util-linux 2.21.2

/dev/mapper/diskedt-sistema: clean, 23/13328 files, 53245/57344 blocks

resize2fs 1.42.3 (14-May-2012)

The filesystem is already 57344 blocks long. Nothing to do!

Reducing logical volume sistema to 56,00 MiB

Logical volume sistema successfully resized

**# lvdisplay /dev/diskedt/sistema**

--- Logical volume ---

LV Path /dev/diskedt/sistema

LV Name sistema

VG Name diskedt

LV UUID OMPKNv-78L2-37y6-3kwO-U96L-3CXv-4OoS3l

LV Write Access read/write

LV Creation host, time portatil.localdomain, 2015-02-06 13:02:17 +0100

LV Status available

# open 0

LV Size **56,00 MiB**Current LE **14**

Segments 2

Allocation inherit

Read ahead sectors auto

- currently set to 256

Block device 253:0

# mount /dev/diskedt/sistema /mnt/sistema/

# mount /dev/diskedt/dades /mnt/dades/

**# df -h**

S. fitxers Mida En ús Lliure %Ús Muntat a

...

/dev/mapper/diskedt-sistema **51M 47M 1,6M 97%** /mnt/sistema

/dev/mapper/diskedt-dades 132M 93M 33M 75% /mnt/dades

## Quarta Part

Finalment assignar al volum lògic **dades** tot l'espai lliure sobrant del grup de volum *diskedt*, tant el provenint d'empetitir sistema com del que encara estava pendent d'assignar.

```
# vdisplay diskedt
--- Volume group ---
VG Name          diskedt
System ID
Format           lvm2
Metadata Areas   3
Metadata Sequence No 7
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          3
Open LV          2
Max PV           0
Cur PV          3
Act PV           3
VG Size          288,00 MiB
PE Size          4,00 MiB
Total PE         72
Alloc PE / Size  64 / 256,00 MiB
Free PE / Size   8 / 32,00 MiB
VG UUID          eXUDOP-9wqX-jXsE-ugt6-fVXv-rm2X-ee87Ng

# lvdisplay /dev/diskedt/dades
--- Logical volume ---
LV Path          /dev/diskedt/dades
LV Name          dades
VG Name          diskedt
LV UUID          UTMphJ-dYIp-9g7W-bjWe-rs9H-BL7X-JNX0mG
LV Write Access  read/write
LV Creation host, time portatil.localdomain, 2015-02-06 13:03:09 +0100
LV Status        available
# open           1
LV Size          140,00 MiB
Current LE       35
Segments         2
Allocation       inherit
Read ahead sectors auto
- currently set to 256
Block device     253:1
```

Examinant els llistats dels exercicis anteriors es pot observar que el volum lògic *sistema* ha apassat de 21 PE a 14 PE, que sumats al únic PE que quedava pendent d'assignar en el grup de volum *diskedt* fan el total de 8 PE disponibles que té el grup de volum *diskedt*.

Els següents passos mostren com assignar aquest espai al volum lògic dades:

```
# umount /mnt/sistema
# umount /mnt/dades

# lvextend -l +100%FREE /dev/diskedt/dades
Extending logical volume dades to 172,00 MiB
Logical volume dades successfully resized

# lvdisplay /dev/diskedt/dades
--- Logical volume ---
LV Path                /dev/diskedt/dades
LV Name                dades
VG Name                diskedt
LV UUID                UTMphJ-dYlp-9g7W-bjWe-rs9H-BL7X-JNX0mG
LV Write Access        read/write
LV Creation host, time portatil.localdomain, 2015-02-06 13:03:09 +0100
LV Status              available
# open                 0
LV Size                172,00 MiB
Current LE           43
Segments              4
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device           253:1
```

Del llistat anterior es s'observa que el volum lògic *dades* s'ha incrementat en 8 PE passant de 35 a 43 blocs de dades, just els 8 PE que hi havia disponibles.

## Entorn Gràfic

---

Es poden gestionar els Volums Lògics amb la utilitat gràfica proporcionada per Fedora:

```
$ system-config-lvm
```