Pi-mounting-filesystems

Intro.

Filesystems on disks are always of a particular type. This means that there are differences between filesystems in the sense of speed, scalability, ease of management and others. Some commonly used types are XFS, EXT2 and EXT4, BTRFS and ZFS.

Next to disk filesystems there are pseudo filesystems.
A pseudo file system maintains information about the currently running system.

This information doesn't persist across reboots. It exists while the system in running only in RAM; in Window this would be the HKLM.

In linux /dev this includes things like /dev/tty# /dev/ttyS# they indicate devices as they are connected and they be created dynamically.

/sys shows a representation of the physical devices in the machine.

/proc maintain a lot of info about the current control set.

Example: free command is just importing info from /proc/meminfo file.

Lab.

For this lab we have to login to some other virtual machines, unless there are disks that can be connected to our raspberries.

To login go here and login as centos-X, so that will be centos-1, centos-2 etc.
You will login on a REL8 virtual machine with a single small disk of 1GB. The diskname=/dev/vdb. Make sure you do not touch /dev/sda, because that is you bootdevice which contains the /boot filesystem and the /(root) filesystem.

To use filesystems in you directory tree you will first have to create a filesystem on a disk or on a disk-partion. The steps are as follows:

- Make sure you pick an available disk
- Partition the disk
- Create a filesystem
- Create a mountpoint (directory)
- Mount the filesystem

You are logged in.

1. Run the following command: **fdisk -l | grep vd**
   How many disks do you see and what are the names?

```
[user@c1~]$sudo  fdisk -l | grep vd
Disk /dev/vda: 20 GiB, 21474836480 bytes, 41943040 sectors
/dev/vda1  *       2048  2099199  2097152   1G 83 Linux
/dev/vda2       2099200 41943039 39843840  19G 8e Linux LVM
Disk /dev/vdb: 1 GiB, 1073741824 bytes, 2097152 sectors
Disk /dev/vdc: 1 GiB, 1073741824 bytes, 2097152 sectors
```

2. Use the fdisk command to create a partition of 500MB on the disk /dev/vdb

```
[user@c1~]$sudo  fdisk /dev/vdb
Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x776b6480.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-2097151, default 2048): <press enter>
Last sector, +sectors or +size{K,M,G,T,P} (2048-2097151, default 2097151): +500M

Created a new partition 1 of type 'Linux' and of size 500 MiB.

Command (m for help): w
The partition table has been altered.
```

3. Run 'fdisk -l' again to check your work.

```
[user@c1~]$ sudo  fdisk -l | grep vd
```

4. Create an xfs file system on the partition you created, and run **lsblk -f** to check.

```
[user@c1~]$ sudo  mkfs -t xfs /dev/vdb1
[user@c1~]$ sudo  lsblk -f
```

```
NAME        FSTYPE      LABEL                UUID
MOUNTPOINT
sda
|-sda1      vfat                             8FC5-D70F
/boot/efi
|-sda2      xfs                              da5d10d0-c685-461c-9d65-dcd0a1de3f50
/boot
`-sda3      LVM2_member                      zLwRO3-z1r2-8Rbb-0im2-D61S-oxfm-915UW0
  |-rl-root xfs                              a0a7659d-f6cc-483b-86d8-21de34b8ac43   /
  `-rl-swap swap                             11b6960c-8a3a-40ea-bfab-7b9118f16361
[SWAP]
vdb
`-vdb1      xfs                              092de58a-3e52-4013-935c-c3b54b760851
sr0         iso9660     Rocky-8-6-x86_64-dvd 2022-05-15-21-06-32-00
```

5. Create a mountpoint and mount the file system.

```
[user@c1~]$ sudo  mkdir /mnt/vdbdata
[user@c1~]$ sudo mount -t xfs /dev/vdb1 /mnt/vdbdata
```

6. Create a file in the file system.

```
[user@c1~]$ sudo touch /mnt/vdbdata/file
```

7. Unmount the file system.

```
[user@c1~]$ sudo umount /dev/vdb1
```

8. Create a file in the mountpoint.

```
[user@c1~]$ sudo touch /mnt/vdbdata/newfile
```

9. Mount the file system again.
Can you still see the file you created in 8?

```
[user@c1~]$ sudo mount -t xfs /dev/vdb1 /mnt/vdbdata
```

No, you will see the file you created at 6.

10. Make sure your file system is mounted at boot time. Either use vi or nano, or use the echo command. It is up to you how you want to edit the file: **/etc/fstab**.

```
[user@c1~]$ sudo  echo "/dev/vdb1 /mnt/vdbdata xfs defaults 0 0" >> /etc/fstab
```

11. Sometimes device names change. This means that using devicenames in /etc/fstab may Sometimes cause problems. To avoid this, you can either use the UUID of the filesystem Instead of the device name, or create a LABEL in the filesystem and use that LABEL.

    We will create a LABEL and replace /dev/vdb1 with the LABEL.

    - Unmount the filesystem
    - Create a label
    - Replace the device name in fstab
    - Mount the filesystem
      Again, it is up to you how to edit the file. I used **sed** in the example…but you can use **nano** or **vi**.

```
[user@c1~]$
sudo umount /dev/vdb1
[user@c1~]$ sudo  xfs_admin -L vdb1 /dev/vdb1
[user@c1~]$ sudo  echo "/dev/vdb1 /mnt/vdbdata xfs defaults 0 0" >> /etc/fstab
[user@c1~]$ sed -i 's/\/dev\/vdb1/LABEL=vdb1/' /etc/fstab
```

Mount the filesystem and check its size and available space.

```
[user@c1~]$ sudo mount /mnt/vdbdata
[user@c1~]$ sudo df -h /mnt/vdbdata
Filesystem      Size  Used Avail Use% Mounted on
/dev/vdb1       894M   39M  856M   5% /mnt/vdbdata
```