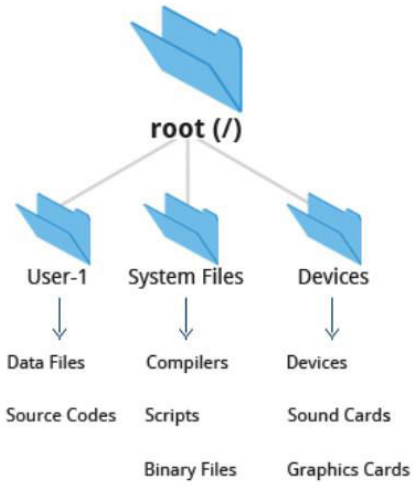
File Operations

Introduction to Filesystems:

1. On many systems (including Linux), the filesystem is structured like a **tree**.
2. The **tree** is usually portrayed as inverted, and starts at what is most often called the **root** directory, which marks the beginning of the hierarchical filesystem and is also sometimes referred to as the trunk, or simply denoted by /.
3. The **root** directory is not the same as the **root user**.
4. The hierarchical filesystem also contains other elements in the path (directory names), which are separated by forward slashes (/), as in /usr/bin/emacs, where the last element is the actual file name.



Filesystems

Filesystem Varieties:

1. Linux supports a number of native filesystem types, expressly created by Linux developers, such as:
    - ext3
    - ext4
    - squashfs
    - btrfs.
2. It also offers implementations of filesystems used on other alien operating systems, such as those from:
    - Windows (ntfs, vfat)
    - SGI (xfs)
    - IBM (jfs)
    - MacOS (hfs, hfs+).
3. It is often the case that more than one filesystem type is used on a machine, based on considerations such as the size of files, how often they are modified, what kind of hardware they sit on and what kind of access speed is needed, etc.

4. The most advanced filesystem types in common use are the journaling varieties: ext4, xfs, btrfs, and jfs.

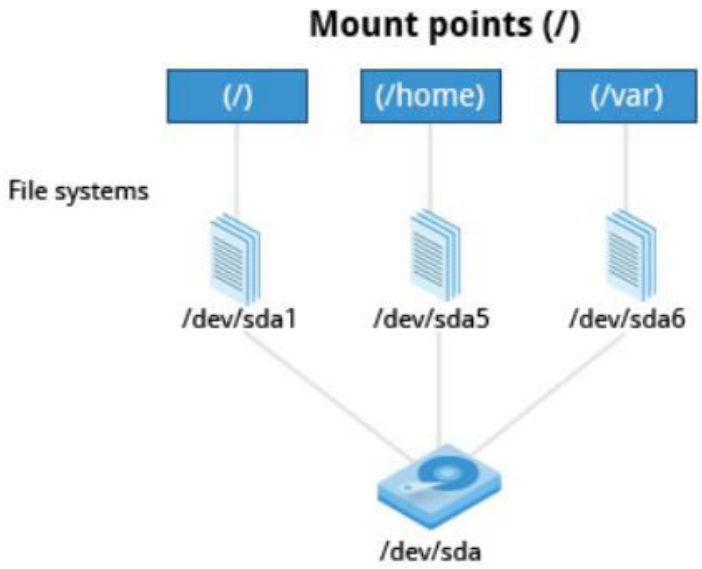
Linux Partitions:

1. Each filesystem on a Linux system occupies a disk partition.
2. Partitions help to organize the contents of disks according to the kind and use of the data contained.
3. For example, important programs required to run the system are often kept on a separate partition (known as root or /) than the one that contains files owned by regular users of that system (/home).
4. In addition, temporary files created and destroyed during the normal operation of Linux may be located on dedicated partitions.
5. One advantage of this kind of isolation by type and variability is that when all available space on a particular partition is exhausted, the system may still operate normally.

Mount Points:

1. Before you can start using a filesystem, you need to mount it on the filesystem tree at a mount point.
2. This is simply a directory (which may or may not be empty) where the filesystem is to be grafted on.
3. Sometimes, you may need to create the directory if it does not already exist.

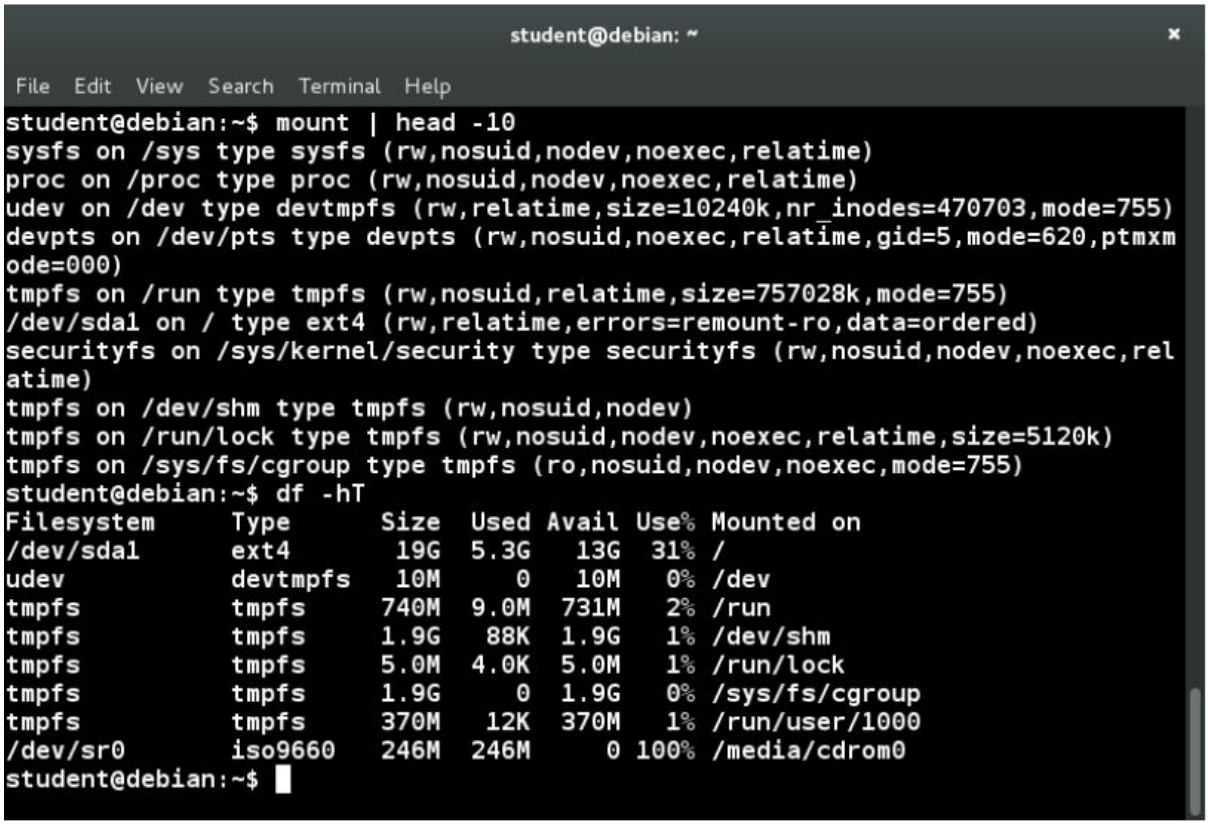
WARNING: If you mount a filesystem on a non-empty directory, the former contents of that directory are covered-up and not accessible until the filesystem is unmounted. Thus, mount points are usually empty directories.



Mount Points

Mounting and Unmounting:

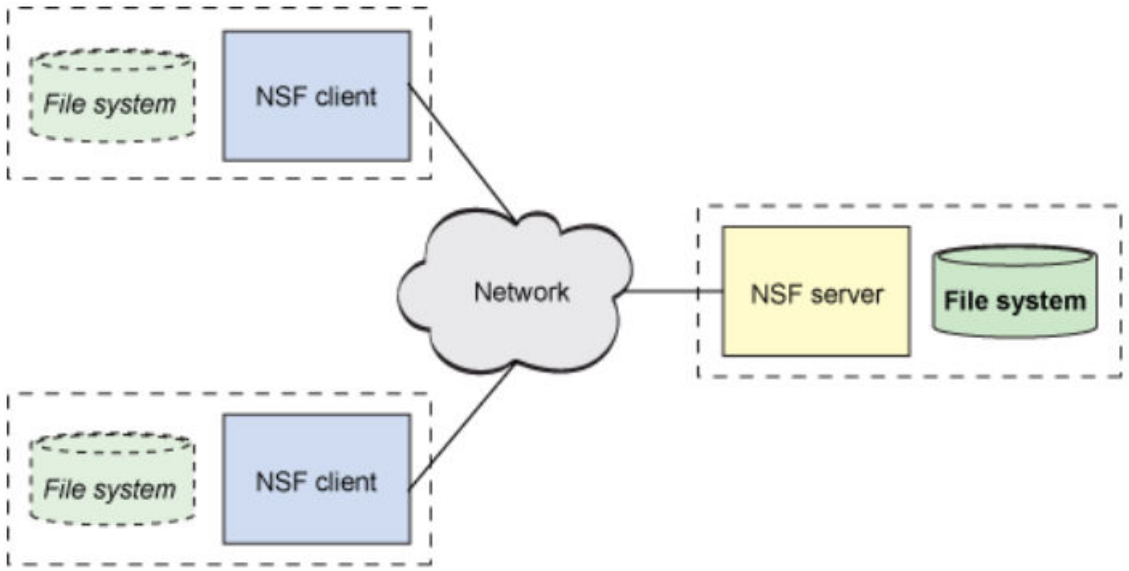
1. The mount command is used to attach a filesystem (which can be local to the computer or on a network) somewhere within the filesystem tree.
2. The basic arguments are the device node and mount point.
   Ex: sudo mount /dev/sda5 /home
3. The above example will attach the filesystem contained in the disk partition associated with the **/dev/sda5** device node, into the filesystem tree at the **/home** mount point.
4. There are other ways to specify the partition other than the device node, such as using the **disk label** or **UUID**.
5. The below is the command to unmount the partition.
   Ex: sudo umount /home
6. Note the command is **umount**, not unmount! Only a **root user** (logged in as **root**, or using **sudo**) has the privilege to run these commands, unless the system has been otherwise configured.
7. If you want it to be automatically available every time the system starts up, you need to edit /etc/fstab accordingly (the name is short for filesystem table).
8. Looking at this file will show you the configuration of all pre-configured filesystems.
9. **man fstab** will display how this file is used and how to configure it.
10. Executing **mount** without any arguments will show all presently mounted filesystems.
11. The command **df -Th** (disk-free) will display information about mounted filesystems, including the filesystem type, and usage statistics about currently used and available space.



Mounting and Unmounting

NFS and Network Filesystems:

1. It is often necessary to share data across physical systems which may be either in the same location or anywhere that can be reached by the Internet.
2. A network (also sometimes called distributed) filesystem may have all its data on one machine or have it spread out on more than one network node.
3. A variety of different filesystems can be used locally on the individual machines; a network filesystem can be thought of as a grouping of lower level filesystems of varying types.
4. Many system administrators mount remote users' home directories on a server in order to give them access to the same files and configuration files across multiple client systems.
5. This allows the users to log in to different computers, yet still have access to the same files and resources.
6. The most common such filesystem is named simply **NFS (the Network Filesystem).**



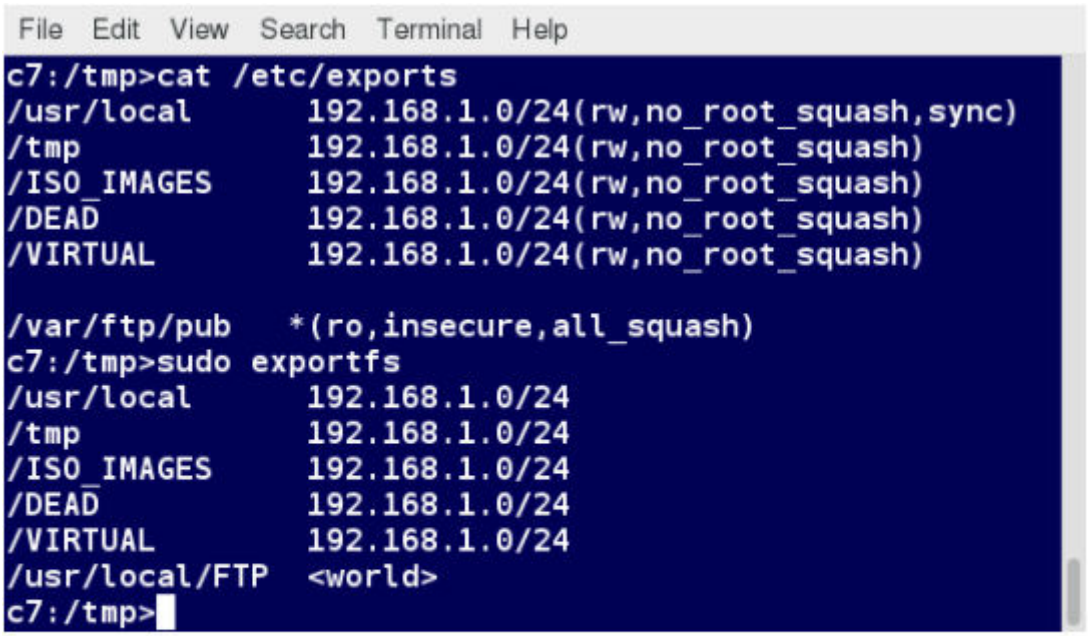
The Client-Server Architecture of NFS

(retrieved from www.ibm.com)

NFS on the Server:

1. On the server machine, NFS uses daemons (built-in networking and service processes in Linux) and other system servers are started at the command line by typing:
   Ex: sudo systemctl start nfs
2. **NOTE:** On RHEL/CentOS 8, the service is called nfs-server, not nfs.
3. The text file /etc/exports contains the directories and permissions that a host is willing to share with other systems over NFS.
4. A very simple entry in this file may look like the following:
   Ex: /projects *.example.com(rw)
5. The above entry allows the directory /projects to be mounted using NFS with read and write (rw) permissions and shared with other hosts in the example.com domain.

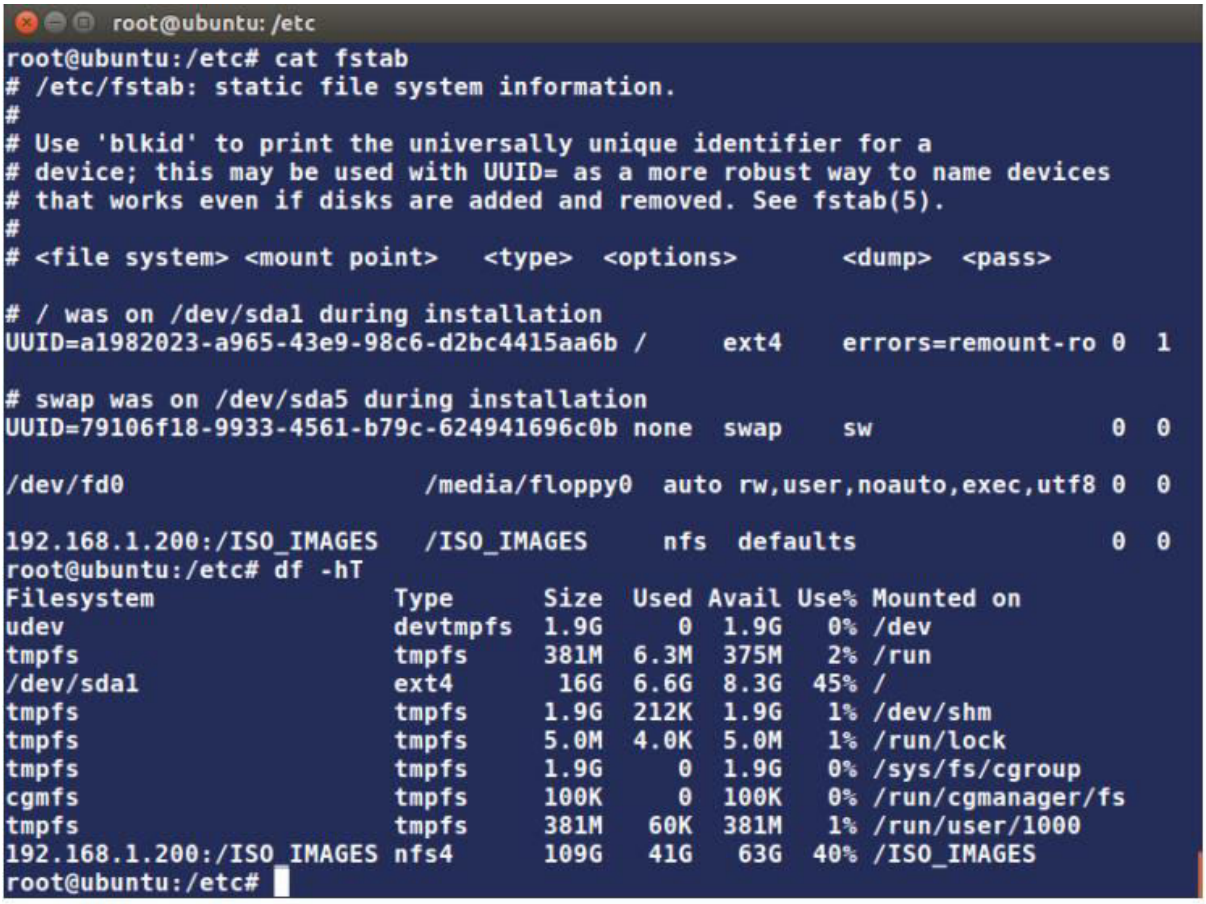
6. After modifying the /etc/exports file, you can type exportfs -av to notify Linux about the directories you are allowing to be remotely mounted using NFS.
7. You can also restart NFS with sudo systemctl restart nfs, but this is heavier, as it halts NFS for a short while before starting it up again.
8. To make sure the NFS service starts whenever the system is booted, issue sudo systemctl enable nfs.



NFS on the Server

NFS on the Client:

1. On the client machine, if it is desired to have the remote filesystem mounted automatically upon system boot, /etc/fstab is modified to accomplish this.
2. For example, an entry in the client's /etc/fstab might look like the below:
   Ex: servername:/projects /mnt/nfs/projects nfs defaults 0 0
3. You can also mount the remote filesystem without a reboot or as a one-time mount by directly using the mount command:
   Ex: sudo mount servername:/projects /mnt/nfs/projects
4. Remember, if /etc/fstab is not modified, this remote mount will not be present the next time the system is restarted.
5. Furthermore, you may want to use the **nofail** option in **fstab** in case the **NFS** server is not live at boot.



NFS on the Client