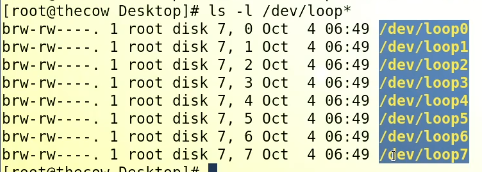
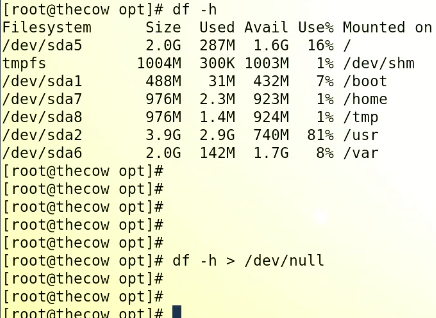
Lecture 11

**File System Hierarchy-YUM**

* /dev/loop 🡪 ISO driver
* To mount ISO with the help of “loop” driver
* Suppose CentOS ISO
* $ mount -o loop centos.iso /mnt
* By default 8 “loop” drivers are available which can be increased
* 
* Another driver
* /dev/null 🡪 character driver
* **/dev/null** is a virtual device file in Unix-like operating systems that acts as a "black hole." Data written to this file is discarded and has no effect on the system.
* It is commonly used to redirect the output of a command that produces unwanted or unnecessary results, so that they do not clutter the terminal or a log file. For example:
* $ command > /dev/null
* This will run the **command** and discard any output it generates.
* It can also be used to provide a "null" input to a command that expects input from a file or standard input:
* $ command < /dev/null
* This will run the “command” with no input.

The input vanishes in null driver

$ ls -l > /dev/null

* 

use case of /dev/null

**/dev/null** is a useful tool for several purposes, including:

1. Discarding unwanted output: By redirecting the output of a command to **/dev/null**, you can suppress any messages, errors, or other data that the command generates. For example:

**command > /dev/null 2>&1**

This will run the **command** and discard both standard output and standard error messages.

1. Testing commands: By redirecting the input of a command to **/dev/null**, you can test whether a command can run without input and whether it produces the desired output.
2. Debugging: By redirecting the output of a program to **/dev/null**, you can see if the problem is with the output or with the program's behavior.
3. Testing scripts: By using **/dev/null** as a placeholder for a real file, you can test shell scripts or other scripts to make sure they run correctly without having to provide real input or output files.
4. Creating empty files: By using **> /dev/null** you can create an empty file without having to specify a file name.

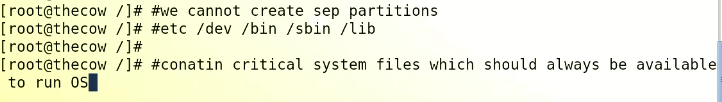
These are just a few examples of the many uses of **/dev/null** in Unix-like systems.

which directories will always remain in root

In Unix-like operating systems, there are several directories that are considered essential and are always present in the root directory **/**. These directories include:

1. **/bin**: contains executable files, also known as binary files, that are required for basic system functionality.
2. **/sbin**: similar to **/bin**, but contains executables for system administration tasks.
3. **/etc**: contains configuration files for the system and applications.
4. **/dev**: contains device files that represent devices attached to the system, such as hard drives, terminals, and printers.
5. **/proc**: a virtual file system that provides information about the system's processes and system status.
6. **/var**: contains variable data, such as log files, temporary files, and spool directories.
7. **/tmp**: contains temporary files that are deleted when the system reboots.
8. **/usr**: contains user-related files, such as libraries, executables, and documentation.

These directories are part of the basic file hierarchy in Unix-like operating systems and are always present in the root directory. However, the exact contents and structure of these directories may vary depending on the specific operating system and distribution.

* 
* Directories that are installed by default in the root directory typically include [*/bin*](http://www.linfo.org/bin.html), */boot*, */dev*, */etc*, [*/home*](http://www.linfo.org/home.html), */initrd*, */lib*, */lost+found*, */misc*, */mnt*, */opt*, */proc*, */root*, [*/sbin*](http://www.linfo.org/sbin.html), */tmp*, */usr* and */var*

**Yum (Yellow-dog update modifier)**

* Yum install a package with dependencies 🡪 its other name is patching (Interview question)
* Yum updates Kerenel
* 