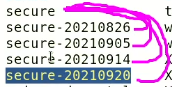
Lecture 10

**File System Hierarchy -Tape Backups**



Secure 🡪 file in /var/logs/secure

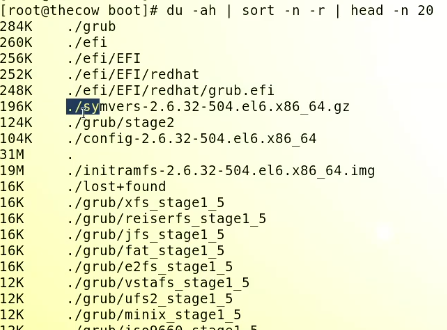
* Authentication related logs are in this file. -🡪 the creation of backup is called “log rotation”
* The “file rotation” process works in a way that it do not allow to increase the size of “secure” file above a certain size. Whenever this size is reached the main secure file is backed up and resets its size from zero bytes. This process continues to save the system resources and to avoid slowing it down.
* 
* Backup rotation is a data backup strategy that involves the creation and storage of multiple copies of data at different stages or points in time, and the systematic management and disposal of older backups as new ones are made. This helps to ensure that a recent, valid backup is available in the event of data loss and helps to reduce the risk of data corruption or obsolescence. The specific rotation scheme can vary depending on the organization's data backup needs and policies.
* Log rotation policy can be changed with log rotation service
* $ ls -d /etc/logrotate.d
* /lib 🡪 directory contains library files and SO files (shared object). The /lib directory **contains those shared library images needed to boot the system and run the commands in the root filesystem**, ie. by binaries in /bin and /sbin .
* /mnt 🡪 optional directory. Temporary used for mounting. The /mnt directory exists on all Linux systems, and it is **intended specifically for use as a mount point for temporary media like floppy disks or CDROMs**. It may be empty, or it may contain subdirectories for mounting individual devices. Linux does not require you to use /mnt as the mount point for other file systems.
* /tmp 🡪 temporary files, .log, .lck etc. temp files are handled by respective applications.
* “tmp” has sticky bit permission (interview question)

/proc directory keeps RAM related data

* “lost+found” directory 🡪 recovered data files are in this directory
* /boot 🡪 most important directory. **Kernel file is stored in /boot** .
* 
* It is developed in “C++” 🡪 complied in binary 🡪 compressed in bzip format.

To check its size

$ du -sh <file\_name> 🡪 size of disk

* 
* -s flag means “summarize”.
* “-h” flag means in human readable format.
* To search large files and directories.
* 

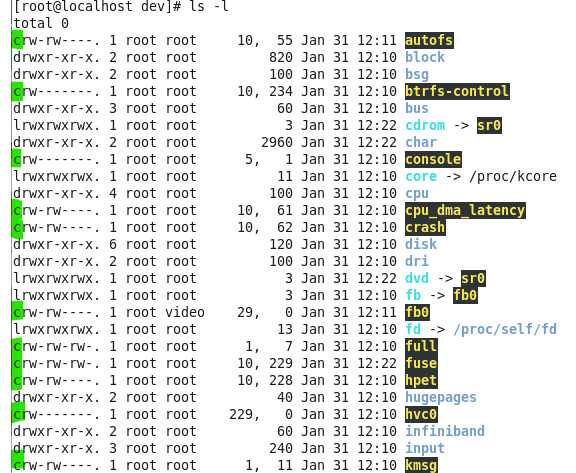
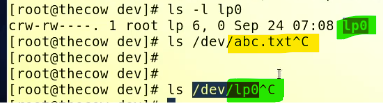
Here are 15 examples of the **du** command:

* **du** - Show disk usage of the current directory and its subdirectories.
* **du /var** - Show disk usage of the /var directory.
* **du -h** - Show disk usage in human-readable format (e.g. MB, GB, etc.).
* **du -sh** - Show the total disk usage of the current directory in human-readable format.
* **du -c** - Show a grand total of disk usage for all directories and files.
* **du -a** - Show disk usage of each file and directory in the current directory.
* **du -k** - Show disk usage in kilobytes.
* **du --max-depth=1** - Limit the depth of the output to only the current directory.
* **du -S** - Show disk usage of each directory, but not the usage of its subdirectories.
* **du --exclude=PATTERN** - Exclude directories and files that match the given pattern.
* **du -b** - Show disk usage in bytes.
* **du --time** - Show the last modification time of each directory and file.
* **du -m** - Show disk usage in megabytes.
* **du -x** - Limit disk usage analysis to a single file system.
* **du --block-size=SIZE** - Specify the block size to use when displaying disk usage.

Use case example.

* Check the /var directory suppose it’s full and you need to list the lard size files for analysis.
* 
* Another important directory in /boot

grub

* it is boot loader.
* /dev 🡪 it has device drivers
* 
* “c” means these r “character” files 🡪 character files
* “s” files 🡪 socket files
* 
* 

Difference b/w a simple text file and lp0 file

lp0 is a special file, its not a normal file 🡪 it is loaded in RAM (character file)

driver file is for communication between two components.

* A txt file is a normal file 🡪 it is only to write data
* “/dev/sda” is also in /dev directory and it is also a directory
* What is the difference b/w /sda driver file and “lp0” driver file?

**/sda** and **lp0** are examples of different types of device files in a Unix-like operating system.

**/sda** is a block device file that represents a hard disk or solid-state drive (SDD) connected to the system. It is used to access the storage space on the device for reading and writing data.

**lp0** is a character device file that represents a parallel port for a printer. It is used for transmitting data between the computer and the printer.

The main difference between these two types of device files is that block device files are used for accessing large amounts of data stored on a storage device, while character device files are used for transmitting smaller amounts of data to and from specialized devices, such as printers, terminals, and audio devices.

* /sda is for reading and writing data from HDD
* lp0 is a printer driver

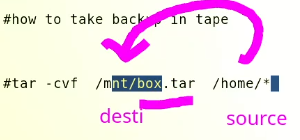
all the drivers in Linux OS are controlled by “Kernel”.

Continue …

**/dev/nst0 🡪 block device**

* tape device

|  |
| --- |
| What is a tape drive? |
| A tape device is a data storage device that uses magnetic tape to store data. Tape devices were commonly used in the past as a backup medium or for archiving large amounts of data but have largely been replaced by other storage technologies such as hard disk drives and solid state drives. Some examples of tape devices include floppy tapes, DAT (digital audio tape) drives, and DLT (digital linear tape) drives. |

* This device has driver in /dev/nst0
* So “nst0, nst1, nst2 …” are drivers for this device
  + This device stores data in “\*.dat “ format
* 
* For tape device
* $ tar -cvf /dev/nst0 /home/\*
* **-c** creates a new archive
* **-v** generates verbose output, printing the names of each file being archived
* **-f** specifies the file name of the archive to be created.
* This device has its own format 🡪 .dat
* For full OS back up
* **$ tar -xvf /dev/nst0 /\***
* Model nos of tape devices (interview questions)
* HP LTO 6
* IBM ts3500
* **To manage this device,**
* The headers in tape device can move forward and backword
* The command is,

$ mt 🡪 needed to be installed.

$ mt forward

The **mt** command is used to control magnetic tape drive operations. Here are ten common uses of the **mt** command:

1. Rewinding the tape: **mt -f /dev/nst0 rewind**
2. Forwarding the tape to a specific block: **mt -f /dev/nst0 fsb N**
3. Backspacing the tape by N blocks: **mt -f /dev/nst0 bsb N**
4. Skipping N files forwards: **mt -f /dev/nst0 fsfile N**
5. Skipping N files backwards: **mt -f /dev/nst0 bsfile N**
6. Retensioning the tape: **mt -f /dev/nst0 retension**
7. Ejecting the tape: **mt -f /dev/nst0 offline**
8. Displaying the status of the tape drive: **mt -f /dev/nst0 status**
9. Writing to the tape: **tar cf /dev/nst0 /path/to/files**
10. Reading from the tape: **tar xf /dev/nst0**

Note: **/dev/nst0** is an example tape device, and the actual name of the tape device may be different on your system.

Tape devices are often used for backup and archiving purposes, and there are several tools and methods to manage them. Here are some common ways:

1. **mt** command: **mt** is a command line utility used to control tape drive operations, such as rewinding, forwarding, and skipping.
2. Backup software: Backup software, such as **tar**, **dump**, or **cpio**, can be used to automate the process of backing up data to tape devices.
3. Device drivers: Tape devices require device drivers to work with the operating system. Make sure that the device drivers for the tape device are properly installed and configured.
4. Media management: Tape devices use removable media, such as tapes or cartridges, that need to be managed. This includes tasks such as labeling, cleaning, and inventory management.

It is important to regularly check the status and performance of tape devices and tapes to ensure that backups are reliable and can be restored in the event of data loss.

* 