

Linux Administration

Azza Khalel

khalelazza@gmail.com

<https://www.linkedin.com/in/azzakhalel/>



Azza Khalil ✓

Openstack | Terraform | RHCE | AWS | kubernetes | Docker | Git | Gitlab
CICD | Openshift

Day4 contents

- Linux file system
 - Inode table
 - Disk label
- Soft link
- Hard link
- df command
- du command
- String processing
- Archiving
- Compression
- Transferring files
- search

Linux file system

- It's responsible for how data stored in HD.
- Examples of Linux file systems
 - Ext2 (second extended file system)
 - Ext3 (Third extended file system)
 - Ext4 (Fourth extended file system)
 - Xfs (Extended file system) \Rightarrow the default file system from centos7
- Hard disk consists of
 - Disk label \rightarrow at the beginning of the whole HD
 - Inode table \rightarrow at the beginning of each partition
 - Each file is a record in the inode table

Inode record

The information stored in this table for each entry includes the following:

- 1.The type of file.
- 2.The file's permissions.
- 3.The number of links.
- 4.The file owner's user ID.
- 5.The group owner's GID.
- 6.When the file was last changed.
- 7.When the file was last accessed.
- 8.When the file last modified
9. Inode ID → `#ls -li` is used to know the inode ID of any file
10. Size → exactly size of the file
11. Blocks → no of blocks that file occupied

Notes:

- file name isn't in the inode table as file may has more than one name.
- moving file in the same partition, it created faster than if I move it in another partition.
- If i want to read any file from any dir on the system, the system go to the inode table and ask what are the permissions of this user on this file.
- Each partition has limit of inode numbers depending on file system
- You can change the default number of indodes per partition during setting the filesystem for partition
 - `#mkfs.ext4 -i <bytes> <partition_name>` (bytes mean size of file in bytes in the partition)
 - the larger the bytes-per-inode ratio, the fewer inodes will be created.

Hard link

Hard link

- `#ln`
- It must be created in the same partition.
- Take the same inode no.
- If the original file is deleted, the other file can be read.
- New file has 1 hard link by default(`file_name`).
- New dir has 2 hard links by default(`dir_name, .`).
- If the file is updated, it's hard link also updated(difference between it and copy)
- You can't create a hard link for dir.

Soft link

Soft link

- `#ln -s`
- `ln -sr`
- Have different inode no.
- If the original file deleted, the soft link also be deleted.
- Soft link contains the path to the original file.
- You can create a soft link to files and dirs.
- `# ln -s <full-path-of-main-file> <path-of-soft-file>`
- You should write the full path of main file in the soft link file

Linux files types

Symbol	Meaning
-	Regular file
d	Directory
l	Link
c	Special File
s	Socket
p	Named Pipe
b	Block Device

df command

- Print all system partitions and some information about each one.
 - `#df`
 - `#df -h` → human readable
 - `#df -i` → to see max, used and available inodes per partition

du command

- Print contents of a certain dir and size of each contents +total size
 - `#du /dir`
 - `#du -h /dir` (use 1024)
 - `#du --si` (use 1000)
 - `#du -s`
- Note: what is the difference between getting the size from `du -h` and `ls -l`?
 - `ls -l` ⇒ get actual size
 - `du` ⇒ get block size

String processing

- Use the **wc** and the **diff** commands to gather word file statistics and compare two files.
 - `#wc <filename> ➔` to get number of lines, words and characters.
 - `#diff <file1> <file2> ➔` to get the difference between 2 files
- Search strings for patterns using the **grep** command.
 - `#grep [options] regular-expression filename(s)`
 - `-i ➔` ignore letter case
 - `-v ➔` only shows lines that do not contain the regular expression.
 - `-n ➔` set the number the lines `cat /etc/passwd | grep -n azza`
- **cut** command cuts fields or columns of text from standard input or the named file and displays the result to standard output
 - `cut option[s] [filename]`
 - Options
 - `-f` specifies field or column.
 - `-d` specifies field delimiter (default is TAB).
 - `-c` specifies characters and cuts by characters.
`cut -f3 -d: /etc/passwd`
`cut -f1,6 -d: /etc/passwd`
- Organize data using the **sort** command.
 - `sort option[s] [filename]`
 - `sort /etc/passwd ➔` rearrange by first field ASC
 - `sort -k4 -t: /etc/passwd ➔` rearrange by 4th field ASC and terminator is :
 - `sort -k4 -t: -n ➔` rearrange by 4th field numeric ASC
 - `sort -r ➔` rearrange Descending

Archiving

- To safeguard your files and directories, you can create a copy, or archive, of the files and directories on a removable medium, such as a cartridge tape.
- You can use the archived copies to retrieve lost deleted, or damaged files.
- The Tape Archiver (tar) utility is used to archive files. It designed to stream files to a backup tape.
- To create an archive:
 - `#tar -cvf archivename.tar file1 file2 file3`
 - c: create a new tar file.
 - v: verbose mode.
 - f: specify the archive file.
- To add a file to an existing archive or to update an archive:
 - `#tar -cvf /root/homes.tar /home`
 - `#tar -rvf /root/homes.tar /etc/hosts`
 - r: Appends files to an archive even these files already exist
 - `#tar -uvf /root/homes.tar /home`
 - u: updates an archive, only newer files will be written to the archive.
- To see the contents of the tar archive:
 - `#tar -tvf /root/homes.tar`
 - t: List table of content.
- To extract the contents of an archive:
 - `#tar -xvf /root/homes.tar`
 - x: Extracts files from the tar command.
 - `#tar -xvf /root/homes.tar -C /tmp`
 - C: To specify the target directory where you want to extract the file to

Compression

- Many files contain a lot of redundancy. Compression programs allow you to make files take less disk space by taking out that redundancy.
- If there is no redundancy, you won't gain much by using compression.
- After creating the archive, it had to be compressed with a separate compression utility, such as gzip or bzip2.
- you can include the -z(gzip) or -j(bzip2) option while creating the archive with tar.
 - This will immediately compress the archive while it is created.
- To compress tar file
 - #gzip homes.tar
 - #bzip2 homes.tar
- To decompress tar file
 - #gunzip homes.tar
 - #bunzip2 homes.tar

Transferring files

- Secure copy (scp)
 - It is a part of OpenSSH suite (port 22)
 - Copies files from local server to remote one and vice verse
 - `#scp etc.tar root@192.168.220.129:/root`
 - `#scp root@192.168.220.129:/root/data.txt /Desktop/`
 - `#scp -r etc root@192.168.220.129:/root →` to transfer directory
- Secure File Transfer Program(sftp)
 - Interactively upload and download files from SSH server
 - It uses the secure authentication mechanism and encrypted data transfer to and from SSH server
 - `#sftp root@192.168.1.1 →` open sftp session between my local server and 192.168.1.1 server
 - `sftp>ls →` to list the remote server
 - `sftp>lls →` to run any command on my local server add l before any command
 - `sftp> put test_file →` to upload file to the server
 - `sftp> get test_file →` to download the file to the server

Search in linux system

- locate
 - The locate command searches for files based on the name or path to the file
 - The command is fast, because it looks up this information from the mlocate database. However, this database does not update in real time.
 - The locate database updates automatically every day
 - Use `#updatedb` command to force an immediate update.
 - `#locate passwd` `#locate -i passwd` → ignore case `#locate -n 5` → limit output 5 searches
- find
 - The find command locates files by searching in real time in the file-system hierarchy
 - This command is slower but more accurate than the locate command..

Expression	Definition
<code>-name filename</code>	Finds files matching the specified filename. Metacharacters are acceptable if placed inside " ".
<code>-size [+ -]n</code>	Finds files that are larger than +n, smaller than -n, or exactly n. The n represents 512-byte blocks.
<code>-atime [+ -]n</code>	Finds files that have been accessed more than +n days, less than -n days, or exactly n days.
<code>-mtime [+ -]n</code>	Finds files that have been modified more than +n days ago, less than -n days ago, or exactly n days ago.
<code>-user loginID</code>	Finds all files that are owned by the loginID name.
<code>-type</code>	Finds a file type, for example, f (file) or d (directory).
<code>-perm</code>	Finds files that have certain access permission bits