



**RHSA1**  
**Red Hat System Administration I**  
**Day 5**

# Day 5 Contents

- Inode table.
- Archiving.
- Compression.
- Yum.
- Search.



# Listing Directory Contents

- **ls -l dir1**

```
-rwxr-xr-x 2 root root 20 512 May 21 16:06 file1
drwxr-xr-x 2 fatma fatma 20 512 May 21 16:06 dir2
```

Permission ↓ Owner Group

Number of links



# Inode

---

- **Linux stores administrative data about files in inodes.**
- **Linux see all files as numbers called “inodes”, or index nodes.**
- **Within each filesystem is an inode table, in which all of the used inodes are mapped to particular files.**

# Inode

- 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.Where the file is on the media.

# Inode

- But It does not contain the file name or file content.
- Names are stored in the directory.
- Each file name knows which inode it has to address to access further file information.
- An inode does not know which name it has; It just know how many names are associated with the inode, These names are referred to as hard links.

# Inode

---

- To view inode number of a file:

```
ls -li /etc/passwd
```

```
1971109 /etc/passwd
```

- To view inode number of a directory:

```
ls -ld /etc
```

```
1966081 /etc
```

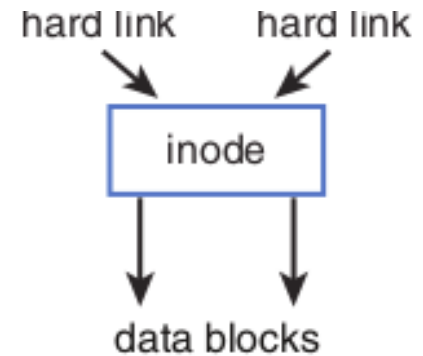
# Hard Link

- When you create a file, you give it a name. Basically, this name is a **hard link**.
- On a Linux file system, multiple hard links can be created to a file. This can be useful, because it enables you to access the file from multiple different locations.
- If the first hard link that ever existed for a file is removed, that does not impact the other hard links that still exist.
- **Some restrictions apply to hard links, though:**
  - Hard links must exist all on the same device (partition, logical volume, etc).
  - You cannot create hard links to directories.
  - When the last name (hard link) to a file is removed, access to the file's data is also removed.



# Hard Links

- To create hard link:
  - In source-file targetfile or directory
  - In /home/fatma/myfile hardlinkfile
  - `ls -i /home/fatma/myfile hardlinkfile`  
`11272876 myfile 11272876 hardlinkfile`
- To be able to create hard links, you must be the owner of the item that you want to link to .



# File Manipulation

---

- The **cp** command:
- Allocates a new inode number for the copy, placing a new entry in the inode table.
- Creates a directory entry, referencing the file name to the inode number within that directory.

# File Manipulation

---

- Example:
  - `ls -i f1`  
1196100 f1
  - `cp f1 f2`
  - `ls -i f1 f2`  
1196100 f1  
1196463 f2

# File Manipulation

- The **mv** command:
- If the destination is on the same file system as the source:
- mv creates a new directory entry with the new file name.
- Example:

■ **ls -i f1**

1196100 f1

■ **mv f1 f2**

■ **ls -i f2**

1196100 f2

# Symbolic Links

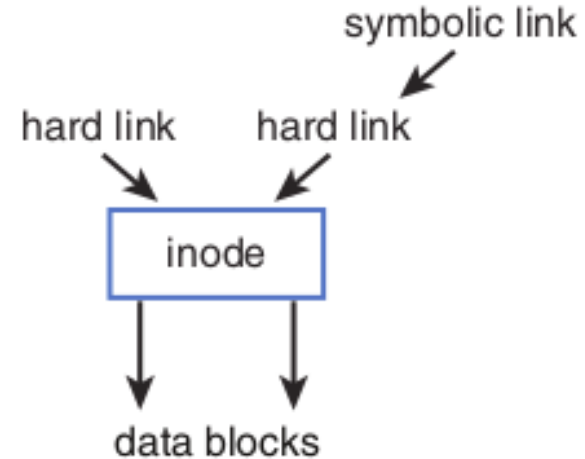
- New entry is made to the inode table for the link The content of this entry is the path to the original file.
- This allows you to use symbolic links across partition boundaries.
- The advantage of symbolic links is that they can link to files on other devices, as well as on directories.
- But when the original file is removed, the symbolic link becomes invalid and does not work any longer.

# Symbolic Links

- To create hard link:
  - `ln -s source-file targetfile or directory`
  - `ln -s testfile softlinkfile`
  - `ls -li testfile softlinkfile`

```
1127996 -rw-rw-r-- 1 user user 12 Mar 12 03:50 testfile
```

```
1127999 lrwxrwxrwx 1 user user 8 Mar 12 09:50 softlinkfile-> testfile
```



# 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.

# Managing Archives with tar

- The Tape Archiver (tar) utility is used to archive files. It designed to stream files to a backup tape.
- To put files on the directory, you need at least read permissions to the file and execute permissions on the directory the file resides in.
- 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.



# Managing Archives with tar

- 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.
  - `tar -uvf /root/homes.tar /home`  
**u:** updates an archive, only newer files will be written to the archive.

# Managing Archives with tar

- 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 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.

# 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.
  - **gzip homes.tar**
  - **bzip2 homes.tar**
- To decompress:
  - **gunzip homes.tar**
  - **bunzip2 homes.tar**

# Managing Software

- The default utility used to manage software packages on RHEL is yum ( **Yellowdog Updater, Modified**).
- Yum is designed to work with **repositories** which are online depots of available software packages.
- In RHEL 8, Yum has been replaced with the dnf utility. Because software in RHEL is based on Fedora software.
- It was expected that, the yum command would be replaced with the dnf command. But Red Hat decided that, with RHEL 8, a new version of yum has been introduced, which is based on the dnf command.
- You'll notice that in many cases, when requesting information about yum, you're redirected to dnf resources.
- So in fact you are using dnf, but RedHat has decide to rename it to yum.

# Yum

- **Basic command**
- **yum search somefile (look for the package)**
- **yum list somefile (get installed and available versions)**
- **yum list installed (same as rpm -qa)**
- **yum list available (what's available in repository)**
- **yum grouplist "some search string" (look for like packages to search string)**
- **yum install somefile (install the package and any dependencies)**
- **yum localinstall /path/to/somefile (yum install off local media)**

# Yum

- **Basic command**
- **yum remove somefile (uninstall the package)**
- **yum upgrade somefile (upgrade the package removing prior versions)**
- **yum update somefile (update the package keeping prior version)**
- **yum provides somefile (what packages are associated with a file)**
- **yum repolist all (list defined repositories)**
- **yum clean all (clean yum download directories)**

# Search

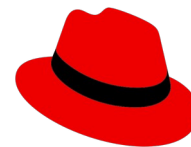
---

- The `find` command searches the live filesystem.
- You are limited by your own permissions.



# find

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



## Lab 5 Part 1

- 1. Install Packages called ncompress,ksh.
- 2. Compress a file by gzip, bzip2 commands and decompress it again.
- 3. List the directories that have 777 Permissions in the system.
- 4. Remove package ncompress and ksh.
- 5. Install Vlc , Vscode ,Atom ,Krita, Google Chrome ,Libre office, zoom and teams.