

RHSA (Admin 1) Red Hat Enterprise Linux 8 Essentials

Prepared by Eng. Taghreed Elsayed

Lecture 9

جامعة الجميع الذكية

تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

Accessing Linux File



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

Accessing Linux File

File Compression and Archiving

- ❑ Sometimes it is useful to store a group of files in one file so that they can be backed up, easily transferred to another directory, or even transferred to a different computer. It is also sometimes useful to compress files into one file so that they use less disk space and download faster.

Important

An archive file is not compressed, but a compressed file can be an archive file.

It is important to understand the difference and relationship between an *archive file* and a *compressed file*. An archive file is a collection of files and directories that are stored in one file.



Accessing Linux File

- ❑ The archive file is not compressed — it uses the same amount of disk space as all the individual files and directories combined. A compressed file is a collection of files and directories that are stored in one file *and* stored in a way that uses less disk space than all the individual files and directories combined.
- ❑ If you do not have enough disk space on your computer, you can compress files that you do not use very often or files that you want to save but do not use anymore. You can even create an archive file and then compress it to save disk space.

Accessing Linux File

File Compressing

- ❑ Compressed files use less disk space and download faster than large, uncompressed files. In Red Hat Linux you can compress files with the compression tools **gzip**, **bzip2**, or **zip**.
- ❑ The **bzip2** compression tool is recommended because it provides the most compression and is found on most UNIX-like operating systems.
The **gzip** compression tool can also be found on most UNIX-like operating systems. If you need to transfer files between Linux and other operating system such as MS Windows, you should use **zip** because it is more commonly used on these other operating systems.



Accessing Linux File

Table 12-1. Compression Tools

Compression Tool	File Extension	Uncompression Tool
gzip	.gz	gunzip
bzip2	.bz2	bunzip2
zip	.zip	unzip

- ❑ By convention, files compressed with **gzip** are given the extension **.gz**, files compressed with **bzip2** are given the extension **.bz2**, and files compressed with **zip** are given the extension **.zip**.
- ❑ Files compressed with **gzip** are uncompressed with **gunzip**, files compressed with **bzip2** are uncompressed with **bunzip2**, and files compressed with **zip** are uncompressed with **unzip**.



Accessing Linux File

Bzip2 and Bunzip2

To use bzip2 to compress a file type the following command at a shell prompt:

```
bzip2 filename
```

❑ The file will be compressed and saved as filename.bz2.

To expand the compressed file, type the following command:

```
bunzip2 filename.bz2
```



Accessing Linux File

- ❑ The filename.bz2 is deleted and replaced with filename.

You can bzip2 multiple files and directories at the same time by listing them with a space between each one:

- ❑ The above command compresses file1, file2, file3, and the contents of the /usr/work/school directory (assuming this directory exists) and put them in filename.bz2.

```
bzip2 filename.bz2 file1 file2 file3 /usr/work/school
```



Accessing Linux File

Gzip and Gunzip

To use **gzip** to compress a file, type the following command at a shell prompt:

```
gzip filename
```

❑ The file will be compressed and saved as **filename.gz**.

To expand the compressed file, type the following command:

```
gunzip filename.gz
```

❑ The **filename.gz** is deleted and replaced with **filename**.

You can **gzip** multiple files and directories at the same time by listing them with a space between each one:

```
gzip -r filename.gz file1 file2 file3 /usr/work/school
```



Accessing Linux File

- ❖ The above command compresses file1, file2, file3, and the contents of the /usr/work/school directory (assuming this directory exists) and put them in filename.gz.

❑ Zip and Unzip

To compress a file with zip, type the following command:

```
zip -r filename.zip filesdir
```



Accessing Linux File

- ❑ In this example, filename.zip represents the file you are creating and filesdir represents the directory you want to put in the new zip file. The -r option specifies that you want to include all files contained in the filesdir directory *recursively*.

To extract the contents of a zip file, type the following command:

```
unzip filename.zip
```

- ❑ You can zip multiple files and directories at the same time by listing them with a space between each one:

The above command compresses file1, file2, file3, and the contents of the /usr/work/school directory (assuming this directory exists) and put them in filename.zip.

```
zip -r filename.zip file1 file2 file3 /usr/work/school
```



Accessing Linux File

File Archiving

A **tar** file is a collection of several files and/or directories in one file. This is a good way to create backups and archives.

Some of the options used with the **tar** are:

- **-C** — create a new archive.
- **-f** — when used with the **-C** option, use the filename specified for the creation of the **tar** file; when used with the **-x** option, unarchive the specified file.
 - **-t** — show the list of files in the **tar** file.
- **-v** — show the progress of the files being archived.
 - **-x** — extract files from an archive.
 - **-z** — compress the **tar** file with **gzip**.
 - **-j** — compress the **tar** file with **bzip2**.



Accessing Linux File

To create a tar file, type:

```
tar -cvf filename.tar files/directories
```

- ❑ In this example, `filename.tar` represents the file you are creating and `files/directories` represents the files or directories you want to put in the archived file.

You can `tar` multiple files and directories at the same time by listing them with a space between each one:

```
tar -cvf filename.tar /home/mine/work /home/mine/school
```



Accessing Linux File

- ❑ The above command would place all the files in the **work** and the **school** subdirectories of **/home/mine** in a new file called **filename.tar** in the current directory.

To list the contents of a tar file, type:

```
tar -tvf filename.tar
```

- ❑ To extract the contents of a tar file, type:

```
tar -xvf filename.tar
```



Accessing Linux File

This command does not remove the **tar** file, but it places copies of its contents in the current working directory.

Remember, the **tar** command does not compress the files by default. To create a tarred and bziped compressed file, use the **-j** option:

```
tar -cjvf filename.tbz
```

tar files compressed with **bzip2** are conventionally given the extension **.tbz**.

This command creates an archive file and then compresses it as the file **filename.tbz**. If you uncompress the **filename.tbz** file with the **bunzip2** command, the **filename.tbz** file is removed and replaced with **filename.tar**.

You can also expand and unarchive a **bzip tar** file in one command:

```
tar -xjvf filename.tbz
```



Accessing Linux File

- ❑ To create a tarred and gunzipped compressed file, use the **-Z** option:
tar files compressed with **gzip** are conventionally given the extension **.tgz**.
This command creates the archive file **filename.tar** and then compresses it as the file **filename.tgz**. (The file **filename.tar** is not saved.) If you uncompress the **filename.tgz** file with the **gunzip** command, the **filename.tgz** file is removed and replaced with **filename.tar**.

```
tar -czvf filename.tgz
```

You can expand a **gzip** tar file in one command:

```
tar -xzvf filename.tgz
```



Accessing Linux File

MOUNTING FILE SYSTEMS MANUALLY

- ☐ A file system residing on a SATA/PATA or SCSI device needs to be mounted manually to access it.
- ☐ The mount command allows the root user to manually mount a file system. The first argument of the mount command specifies the file system to mount.
- ☐ The second argument specifies the target directory where the file system is made available after mounting it.
- ☐ The target directory is referred to as a mount point. The mount command expects the file system argument in one of two different ways:
 - The device file of the partition holding the file system, residing in /dev.
 - The UUID, a universal unique identifier of the file system.



Accessing Linux File

Mount by device file of the partition that holds the file system.

```
[root@serverX ~]#  
mount /dev/vdb1 /mnt/mydata
```

UNMOUNTING FILE SYSTEMS

- To unmount a file system, the umount command expects the mount point as an argument.
- Change to the /mnt/mydata directory. Try to umount the device mounted on the /mnt/mydata mount point. It will fail.



Accessing Linux File

```
[root@serverX ~]# cd /mnt/mydata[root@serverX mydata]# umount /mnt/mydata
umount: /mnt/mydata: target is busy.
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))
```

- ❑ **Unmounting** is not possible if the mount point is accessed by a process.
- ❑ For umount to be successful, the process needs to stop accessing the mount point.
- ❑ The lsof command lists all open files and the process accessing them in the provided directory.
- ❑ It is useful to identify which processes currently prevent the file system from successful unmounting.



Accessing Linux File

```
[root@serverX mydata]# lsof /mnt/mydata
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
bash	1593	root	cwd	DIR	253,2	6	128	/mnt/mydata
lsof	2532	root	cwd	DIR	253,2	19	128	/mnt/mydata
lsof	2533	root	cwd	DIR	253,2	19	128	/mnt/mydata

- ❑ Once the processes are identified, an action can be taken, such as waiting for the process to complete or sending a SIGTERM or SIGKILL signal to the process.
- ❑ In this case, it is sufficient to change the current working directory to a directory outside the mount point.

```
[root@serverX mydata]#  
cd  
[root@serverX ~]#  
umount /mnt/mydata
```



Accessing Linux File

ACCESSING REMOVABLE STORAGE DEVICES

- ❖ Removable media, such as USB flash devices and drives, get automatically mounted by the graphical desktop environment when plugged in.
- ❖ The mount point for the removable medium is `/run/media/`.
- ❖ The is the user logged into the graphical environment.
- ❖ The is the name given to the file system when it was created.

WARNING

- To safely remove USB media from the system, it is required to unmount it before physically removing it from the USB slot to synchronize the file system.
- Removing a USB storage device without unmounting the file system on it can result in data loss.



Accessing Linux File





جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

