

ARCHIVE AND FILE COMPRESSION

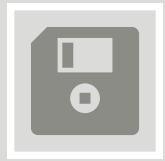
Unit-III

SCS281: Linux and Shell Programming

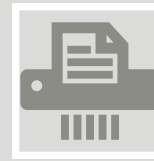
Mapped Course Outcomes (CO): CO4



Archive and File compression



In Linux, archiving and file compression help to save space and make file transfers more efficient.



Archiving combines multiple files into a single file, while compression reduces file size by removing redundant data.



File compression is a fundamental task in managing and transferring data efficiently on a Linux system.



The Tar command, short for Tape Archive, is a powerful tool that allows users to create compressed and archived files.

Archiving Tools: tar

The most common tool for archiving in Linux is tar, which stands for **Tape Archive**.

tar is used to combine multiple files and directories into a single archive file, often called a "tarball."

Syntax:

```
tar [options] archive_name file_or_directory
```

Common Options:

- **-c**: Create a new archive.
- **-x**: Extract files from an archive.
- **-v**: Verbose mode (shows progress).
- **-f**: Specify the filename of the archive.
- **-t**: List contents of an archive.
- **-z**: Compress or decompress an archive with gzip.
- **-j**: Compress or decompress an archive with bzip2.
- **-J**: Compress or decompress an archive with xz.

Examples:

1. Creating a tar Archive

Create a tar archive from a directory

```
tar -cvf archive.tar /path/to/directory
```

2. Creating a Compressed Archive

- **With gzip** (produces .tar.gz or .tgz files):

```
tar -cvzf archive.tar.gz /path/to/directory
```

- **With bzip2** (produces .tar.bz2 files):

```
tar -cvjf archive.tar.bz2 /path/to/directory
```

- **With xz** (produces .tar.xz files):

```
tar -cvJf archive.tar.xz /path/to/directory
```

Examples:

3. Extracting from an Archive

Standard tar:

```
tar -xvf archive.tar
```

Compressed tar (e.g., tar.gz):

```
tar -xvzf archive.tar.gz
```

4. Listing Contents of an Archive

```
tar -tvf archive.tar
```

cpio

cpio is another tool for archiving, often used with pipelines.

Example:

```
find /path -type f | cpio -o > archive.cpio
```

Compression Tools

Compression reduces the size of files or archives. In Linux, several tools are available for compressing files, including

- gzip,
- bzip2,
- xz, and
- zip.

1. gzip – Compress and Decompress Files

The gzip command compresses files and produces a .gz file. It's fast and commonly used, though it may not achieve the highest compression ratio.

Syntax:

```
gzip [options] file
```

```
gunzip [options] file.gz # For decompression
```

Common Options:

- **-k**: Keep the original file after compression.
- **-d**: Decompress a file (also done using gunzip).
- **-r**: Recursively compress all files in a directory.

Examples:

Compress a file

```
gzip filename.txt
```

Compress a file and keep the original

```
gzip -k filename.txt
```

Decompress a file

```
gunzip filename.txt.gz
```

2. bzip2 – High Compression Ratio

bzip2 offers better compression than gzip but is slower. It produces .bz2 files and is commonly used for archiving larger files where space is crucial.

Syntax:

bzip2 [options] file

bunzip2 file.bz2 # For decompression

Common Options:

- **-k**: Keep the original file after compression.

Examples:

Compress a file

```
bzip2 filename.txt
```

Compress and keep the original

```
bzip2 -k filename.txt
```

Decompress a file

```
bunzip2 filename.txt.bz2
```

3. xz – High Compression, High Efficiency

The xz command compresses files with a high compression ratio, often outperforming both gzip and bzip2 in terms of space savings but at the cost of speed. It produces .xz files.

Syntax:

xz [options] file

unxz file.xz # For decompression

Common Options:

- **-k**: Keep the original file after compression.

Examples:

Compress a file

xz filename.txt

Compress and keep the original

xz -k filename.txt

Decompress a file

unxz filename.txt.xz

4. zip – Compression and Archiving

The zip command combines archiving and compression into a single step. zip archives are widely compatible with other systems, making them ideal for cross-platform sharing.

Syntax:

```
zip [options] archive_name.zip file1 file2 ...
```

```
unzip archive_name.zip # For extraction
```

Common Options:

- **-r**: Recursively zip directories.
- **-d**: Delete files from the archive.
- **-l**: List contents of a zip archive.

Examples:

Create a zip archive from files

```
zip archive.zip file1.txt file2.txt
```

Recursively zip a directory

```
zip -r archive.zip /path/to/directory
```

Extract a zip archive

```
unzip archive.zip
```


Combined Archiving and Compression

By combining archiving and compression, you can organize files and save disk space.

Some examples include:

Creating a .tar.gz File

```
tar -czvf archive.tar.gz file1 file2
```

Extracting a .tar.gz File

```
tar -xzvf archive.tar.gz
```

Creating a .zip File

```
zip archive.zip file1 file2
```

Extracting a .zip File

```
unzip archive.zip
```

Choosing a Compression Tool

These tools allow efficient file storage and transfers, with options for different levels of speed and compression ratio. By choosing the right tool, users can optimize storage and improve file management on Linux systems.

Compression Tool	Extension	Speed	Compression Ratio	Common Use
gzip	.gz	Fast	Medium	General-purpose compression
bzip2	.bz2	Moderate	High	Compressing large files
xz	.xz	Slow	Very High	Archiving with best compression
zip	.zip	Fast	Medium	Cross-platform compatibility