8.12.5 Archive Facts

Being able to backup, restore, and compress files will help in protecting and coping files and even disks or partitions.

This lesson covers the following topics:

- Using tar
- Using gzip
- Using xz
- Using bkzip2
- Using zip

Using tar

The tar (tape archive) utility takes the contents of several files and stores them as a single file. The tar command can be used to backup directories or entire file systems. To help others know that the tar utility must be used to extract files from it, by convention, an archive file created with tar is assigned an extension of .tar.

Command	Options and Descriptions	Examples
tar	**-A** Appends one tar file to another archive file. **-c** Creates a new archive. **-d** Identifies differences between the files in an archive file and the same files in the file system. **-v** Displays a list of all files being written into the archive. **-f** Specifies the file to create or unpack. Without this option, tar uses standard input and output as the destination. **-x** Extracts the files. If no destination directory is specified, then tar extracts the files to the current working directory. **-z** Compresses and decompresses a file using the **gzip** utility (normally named with a .gz extension). **-j** Compresses and decompresses a file using the **bzip2** utility (normally named with a .bz2 extension). **-J** Compresses and decompresses a file using the **xz** utility (normally named with a .xz or .lzma extension. **-C** Changes to a specific directory to extract the files. **-t** Lists the contents of an archive. **-P** Tells tar to not strip the leading / from filenames as they are added to the archive. **-r** Adds files to the end of an existing tar archive. **-u** Adds files to the end of an existing tar archive only if they are newer than the existing files in an archive. **-X** Causes tar to exclude the file names contained	**tar -cf /root/tarbackups/oct17backup.tar /home** Writes a backup of the /home directory to the **/root/tarbackups/oct17backup.tar** file. **tar -cvf /root/tarbackups/oct17backup.tar /home** Writes a backup of the /home directory to the **/root/tarbackups/oct17backup.tar** file with verbose output. **tar -xvf /root/tarbackups/oct17backup.tar -C /home** Extracts the files and decompresses them to the **/home** directory.

file_name	in the specified file when creating an archive file.

Using gzip

The **gzip** command reduces the size of the named files using Lempel–Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension '.gz', while keeping the same ownership modes, access and modification times. The default extension is -gz for VMS, z for MSDOS, OS/2 FAT, Windows NT FAT and Atari.) If no files are specified, or if a file name is "-", the standard input is compressed to the standard output. Gzip will only attempt to compress regular files. In particular, it will ignore symbolic links.

Command	Options and Description	Examples
gzip	**Option** / **Description** **-c** — Writes the file to standard output. **-d** — Decompresses the file. **-l** — Displays information about files in an archive. **-r** — Recursively compresses all files in directories and subdirectories. This is the same as the **tar -z** command.	**gzip file**.tar Compresses an archive file created with tar. The original uncompressed file is removed. **gzip -c file.tar > file.tar.gz** Compresses a tar archive, but leaves the original file unchanged. **gzip -d file.tar.gz** Decompresses the tar archive.

Using xz

The **xz** command is a general-purpose data compression tool with command line syntax similar to gzip and bzip2. The native file format is the .xz format, but also the legacy .lzma format and raw compressed streams with no container format headers are supported.

The **xz** command compresses or decompresses each file according to the selected operation mode. If no files are given or file is -, xz reads from the standard input and writes the processed data to standard output. **xz** will refuse (display an error and skip the file) to write compressed data to standard output if it is a terminal. Similarly, xz will refuse to read compressed data from standard input if it is a terminal.

Command	Options and Descriptions	Examples
xz	**Option** / **Description** **-z** — Compresses a file. **-d** — Decompresses a file **-k** — Keeps the original file unchanged. This is the same as the **tar -J** command.	**xz file** Compresses the archive file and removes the original file. **xz -k file** Compresses the archive file, but leaves the original file unchanged. **xz -d file.gz** Decompresses the archive file.

Using bzip2

Compared with **gzip**, the **bzip2** command will create smaller archives but has a slower decompression time and higher memory use.

bzip2 compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding.
The command-line options are deliberately very similar to those of GNU gzip, but they are not identical.

bzip2 expects a list of file names to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the name original_name.bz2. Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original, so that these properties can be correctly restored at decompression time.

Command	Options and Descriptions	Examples

<table>
<tr>
<td>bzip2</td>
<td>
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>-z</td><td>Compresses a file.</td></tr>
<tr><td>-d</td><td>Decompresses a file</td></tr>
<tr><td>-k</td><td>Keeps the original file unchanged

This is the same as the tar -j command.</td></tr>
</table>
</td>
<td>bzip2 file.tar
Compresses the tar archive file and removes the original file.

bzip2 -k file.tar
Compresses the tar archive file, but leaves the original file unchanged.

bzip2 -d file.tar.bz2
Decompresses the tar archive.</td>
</tr>
</table>

Using zip

The zip program puts one or more compressed files into a single zip archive, along with information about the files (name, path, date, time of last modification, protection, and check information to verify file integrity). An entire directory structure can be packed into a zip archive with a single command. This program is useful for packaging a set of files for distribution; for archiving files; and for saving disk space by temporarily compressing unused files or directories.

The **zip** command is a compression and file packaging utility for many operating systems including Linux, Unix, and Windows. It is analogous to a combination of the Linux tar command and is compatible with PKZIP (Phil Katz's ZIP for MSDOS systems). A companion program named **unzip**, unpacks zip archives.

The syntax for **zip** is: **zip [options] zipfile files_lis**t

<table>
<tr><th>Command</th><th>Options and Descriptions</th><th>Examples</th></tr>
<tr>
<td>zip</td>
<td>
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>-d</td><td>Removes a file from the zip archive. When a zip archive includes multiple files, use this option to remove a file from the archive.</td></tr>
<tr><td>-u</td><td>Updates the file in the zip archive. The opposite of -d, meaning you can use this option to add a new file to the zip file already created.</td></tr>
<tr><td>-m</td><td>Deletes the original files after zipping.</td></tr>
<tr><td>-r</td><td>Lets you zip a directory recursively.</td></tr>
<tr><td>-x</td><td>Lets you exclude the files files while creating the zip of multiple files, such as a directory.</td></tr>
<tr><td>-v</td><td>Verbose mode or print diagnostic version information.</td></tr>
</table>
</td>
<td>zip –r my.zip mydir
Will recursively zip the files in the mydir directory. The results are saved to the my.zip file.

zip –m my.zip myfile.txt
Will compress the myfile.txt file into the my.zip file. It will then delete the myfile.txt file.</td>
</tr>
</table>