LINUX FOUNDATION

TRAINING

Archiving and Compressing Files and Directories

A Linux Foundation Training Publication
www.training.linuxfoundation.org

Overview

Sometimes you need to send a number of files over a network, or as an email attachment. Or, having finished a project, you want to package up all the project artefacts together and save some disk space. Archiving files and directories is a way to combine many files into a single file but retain the original file hierarchy and directory structure. Compression causes those files and directories to occupy less space on disk than they otherwise would.

The most common way to archive files in Linux is to use the tar command, short for tape archive, which combines your content into a single .tar file. These tar files are usually compressed using either the gzip or bzip2 compression algorithms. The gzip algorithm is faster, but bz2 compressed archives are smaller.

Key Ideas

tar: The tar command creates an archive file consisting of some specified files and directories.

Usage: tar [OPTIONS] [TAR FILE NAME] [FILES AND DIRECTORIES TO ARCHIVE]

Common options: v - verbosely perform actions, displaying progress on the screen, z - use gzip compression, j - use bzip2 compression

Required options: c - create the tar file, f - specify the name of the resulting tar file

Example: tar archive.tar.gz file1 file2 dir1 dir2

Clarify that this command will archive AND compress file1, file2, dir1, and dir2 into a file named archive.tar.gz. Since the v option is used, verbose output will be shown in the screen.

.tar file: an archive file containing multiple files and directories. Tar files are the same size as the files they contain.

gzip and bzip2: Compression algorithms. Gzip produces larger files, fast. Bzip produces smaller files, slowly.

tarball: Jargon for a compressed tar file.

Example Scenario

Create a directory called hats, with a directory inside it called casual. In the hats directory, create files called fedora.txt and bowler.txt. In the casual directory, create a file called snapback.txt and file called bucket.txt.

Now Do It

Practice archiving and compressing files and directories with the following exercises.

1. Create a gziped tar file called casual.tar.gz that contains snapback.txt and bucket.txt, but no directories.
2. Create a bziped tar file called justincase.tar.bz that contains fedora.txt and the casual directory.
3. Create an uncompressed tar file that contains all the hats, but not the hats directory.

If you remember nothing else...

The most frequently used compression algorithm is gzip as large disks and fast networking has become increasingly common. The verbose options will show you what is happening as your files are being compressed. The c and f options are the ones you absolutely need to get a tar file: c for create, f to specify the name of the resulting archive.

Answer Key

Setup:
mkdir hats
mkdir hats/casual
touch hats/fedora.txt
touch hats/bowler.txt
touch hats/casual/snapback.txt
touch hats/casual/bucket.txt

1. # tar -cvzf casual.tar.gz hats/casual/bucket.txt hats/casual/snapback.txt
hats/casual/bucket.txt
hats/casual/snapback.txt

2. # tar -cvjf justincase.tar.bz hats/fedora.txt hats/casual/
hats/fedora.txt
hats/casual/
hats/casual/snapback.txt
hats/casual/bucket.txt

Note: if you receive an error like “tar (child): bzip2: Cannot exec: No such file or directory”, you may need install the bzip2 package using your operating system’s package manager.

3. tar -cvf hats.tar -C hats bowler.txt fedora.txt casual/snapback.txt casual/bucket.txt
bowler.txt
fedora.txt
casual/snapback.txt
casual/bucket.txt

LINUX FOUNDATION

Get Certified!

Get more information at http://training.linuxfoundation.org/certification/lfcs