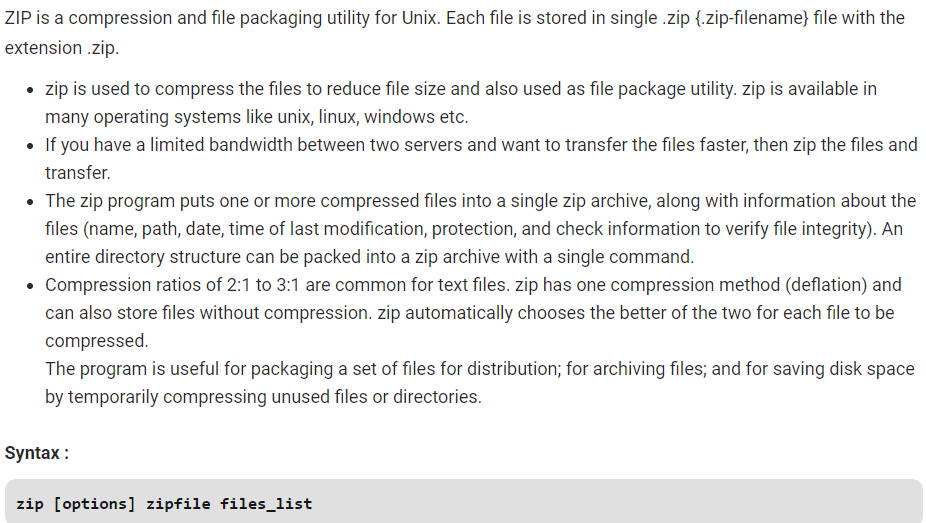
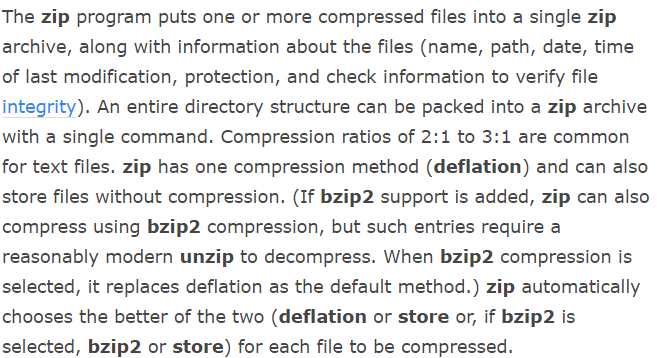
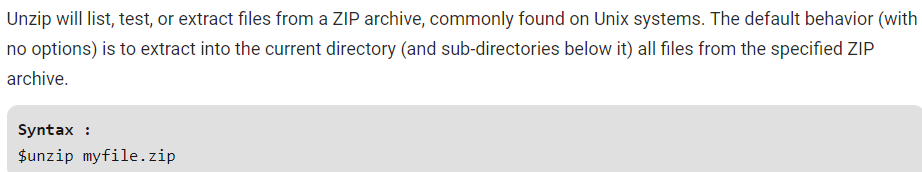
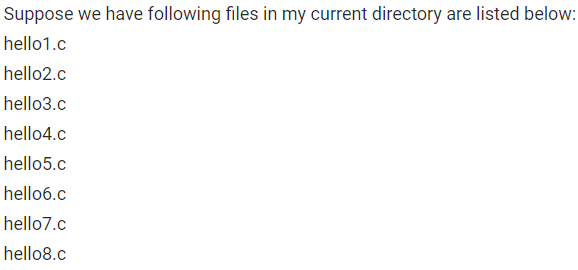
**zip command:**

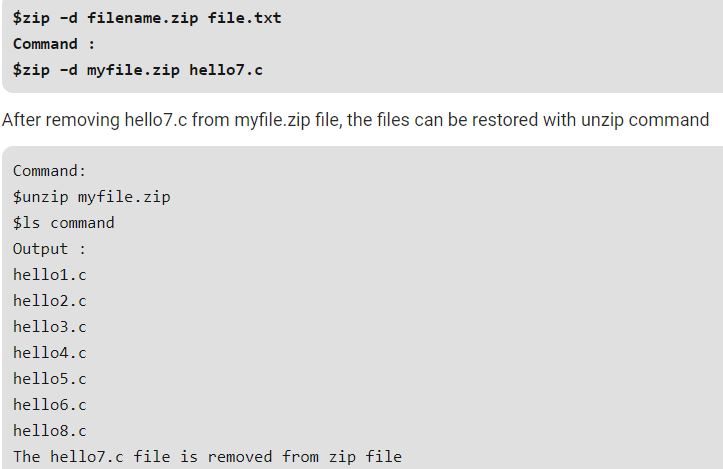




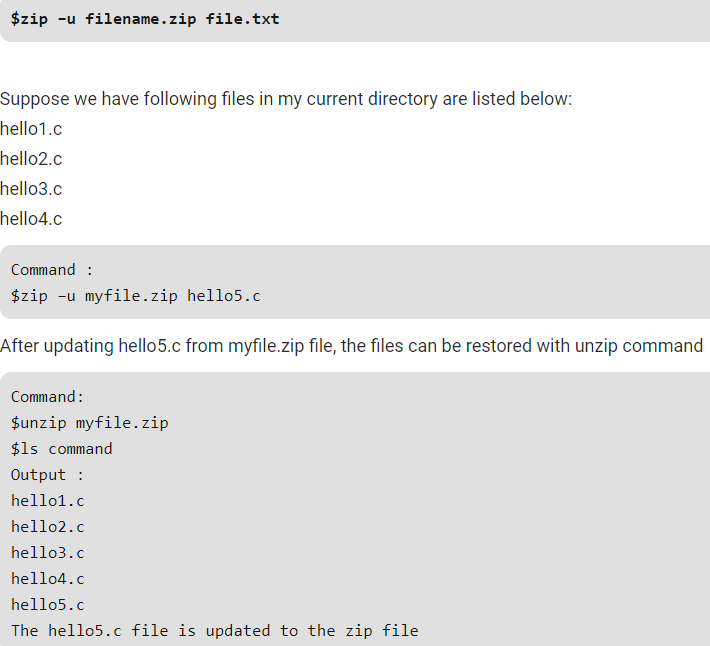


**-d Option:** Removes the file from the zip archive. After creating a zip file, you can remove a file from the archive using the **-d** option.

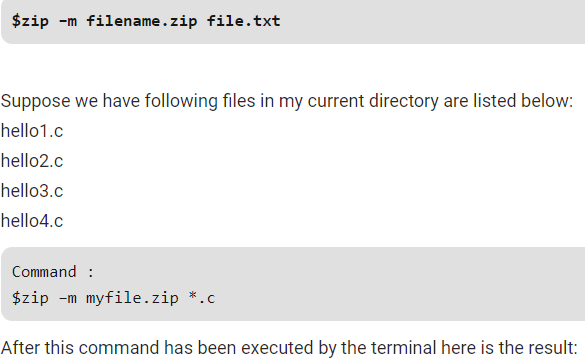


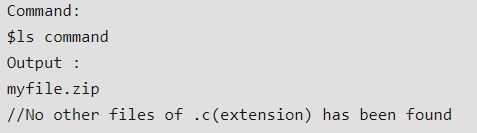


-**u Option:** **Updates the file in the zip archive. This option can be used to update the specified list of files or add new files to the existing zip file. Update an existing entry in the zip archive only if it has been modified more recently than the version already in the zip archive.**

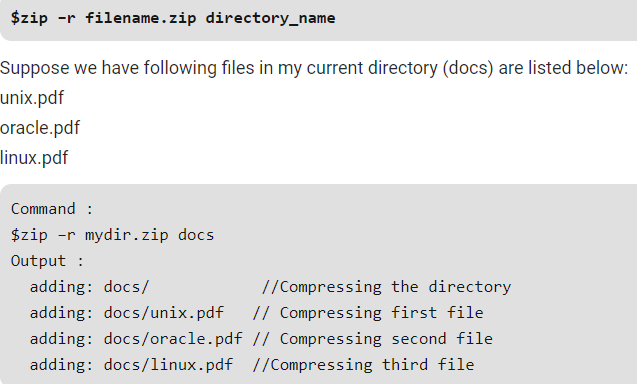


-**m Option**: Deletes the original files after zipping. Move the specified files into the zip archive actually, this deletes the target directories/files after making the specified zip archive. If a directory becomes empty after removal of the files, the directory is also removed. No deletions are done until zip has created the archive without error. This is useful for conserving disk space, but is potentially dangerous removing all input files.

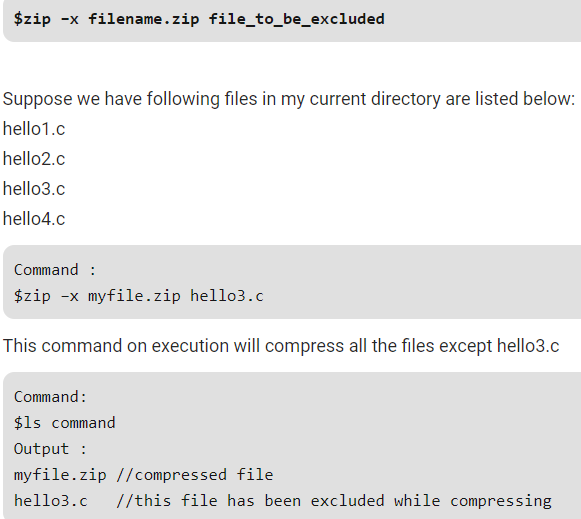




**-r Option:** **To zip a directory recursively, use the -r option with the zip command and it will recursively zips the files in a directory. This option helps you to zip all the files present in the specified directory.**

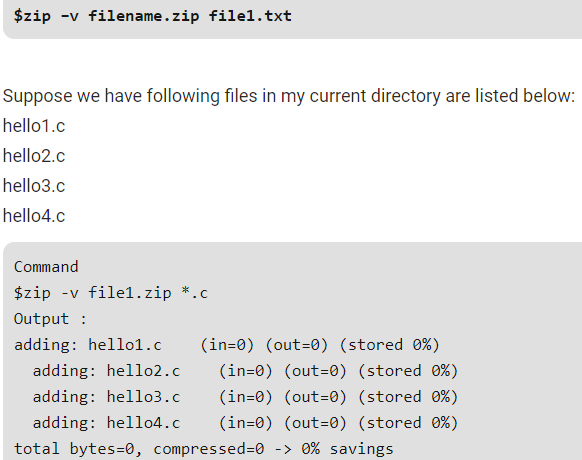


**-x Option:** **Exclude the files in creating the zip. Let say you are zipping all the files in the current directory and want to exclude some unwanted files. You can exclude these unwanted files using the -x option.**



-**v Option:** Verbose mode or print diagnostic version info. Normally, when applied to real operations, this option enables the display of a progress indicator during compression and requests verbose diagnostic info about zip file structure oddities.

When -v is the only command line argument, and either stdin or stdout is not redirected to a file, a diagnostic screen is printed. In addition to the help screen header with program name, version, and release date, some pointers to the Info-ZIP home and distribution sites are given. Then, it shows information about the target environment (compiler type and version, OS version, compilation date and the enabled optional features used to create the zip executable.



**unzip examples:**

**Unzip the hope.zip archive into the current directory, creating subdirectories as needed to match the archive's file hierarchy.**

* **unzip hope.zip**

**Extract the files from archive hope.zip into the current directory only, regardless of the archive's internal directory structure.**

* **unzip -j hope.zip**

Test hope.zip for errors, but do not extract anything. The q indicates "quiet" mode, which instructs unzip to return only a summary message, rather than a more detailed one.

* **unzip -tq hope.zip**

Display the contents of all files with the extension .txt contained in the archive hope.zip to the standard output.

* **unzip -ca hope.zip \*.txt**

Extract from the hope.zip archive any C source files with the extensions .c and .h, and any files named Makefile, into the directory /tmp

* **unzip hope.zip "\*.[ch]" Makefile -d /tmp**

Same as the above command, but case-insensitive; it will also extract any files with the extensions .C or .H, any files named MAKEFILE, any files named MakeFile, etc.

* **unzip -C hope.zip "\*.[ch]" Makefile -d /tmp**

**Extract files from the archive hope.zip, but only if they are newer versions than the files that already exist.**

* **unzip -fo hope.zip**

**Display a listiunng of the contents of hope.zip without extracting anything.**

* **unzip -l hope.zip**

**unzipping the file into particular directory**

* **unzip <zip file> -d <path>**

**unzipping the file into particular directory without creating any directory with zip file name there.**

* **unzip -j <zip file> -d <path>**