AutoZip: An Integrated & Convenient Way for File Compression

Zhejian YU

***Abstract*—Compressing files and folders with a handful of algorithms is a daily task for programmers in various aspects. Because there are plenty of compressing algorithms available, one needs to remember different syntaxes for corresponding software when using them, which can be a waste of time. So, in order to eliminate the time wasted on recalling syntaxes, we invented AutoZip, whose aim is to make its users more efficient and productive by free them from memorizing the different syntax of different compressing software. This article describes how AutoZip archives its goal.**

***Index Terms*—File Compression.**

I. Introduction

For daily system administration work, one need to compress/decompress files or folders for various purposes such as backup or file transporting within or among computers. The appearance of file-compressing algorithms successfully makes our life easier.

There are several widely-used compression algorithms in our society. For users under the Microsoft Windows platform, WinRAR, WinZip, 7Zip and Bandizip are most popular among users in China while GNU Tar [1], gzip [2], XZ Utils [3], bzip2 [4] and Zip are mostly used under GNU/Linux. However, the different syntaxes used by these software makes it hard for ordinary people to memorize. So, when they received an archive of unknown format, they have to look up the manual and learn the new syntax used by corresponding software, which slows down the speed of processing data and thus, influencing efficiency and productibility.

So, in order to solve this problem, we invented AutoZip, whose aim is to libre us from memorizing all those syntaxes by memorizing only one set of syntax: the syntax of AutoZip! AutoZip provided us with great flexibility and convenience, whose syntax is easy enough for everyone to memorize.

II. Usage of AutoZip

A. You Need to Know ...

You need to know how to operate package management systems on your machine such as `apt`, `yum` or `pacman`, or how to build a software from its source code. You need to know how to operate at least one text editor such ad `vim`, `emacs` or `nemo`. You need to know the basic knowledge about GNU/Linux filesystems and environment variable `PATH`.

B. Installation

AutoZip is designed to be portable; if you want to install AutoZip, you should firstly make sure that you have installed GNU Bash at `/bin/bash`.The dependencies for AutoZip are as follows:

1) GNU Tar (optional)
2) gzip (optional)
3) pigz (optional)
4) XZ Utils (optional)
5) bzip2 (optional)
6) Zip & UnZip (optional)
7) RAR & UNRAR (optional)
8) p7zip (optional)
9) bgzip (optional, for bioinformatical use only)
10) GNU Parallel (optional)
11) GNU Split (optional)

They can be installed by package management system provided by your GNU/Linux distributions. You need to install GNU Parallel and pigz if you wish to (de)compress the files with a greater speed.

The installation of the main AutoZip program is simple: you need to copy `autozip`, `autounzip`, `autozip.Usage` and `libautozip` to a folder and add the path to that folder to `PATH` environmenta varible of your system (if you have root previlidge) or your own account. This can be done by:

```
mkdir [dest]
cp autounzip [dest]
cp autozip [dest]
cp autozip.Usage [dest]
cp libautozip [dest]
echo "export PATH=\$PATH:[dest]">>~/.bashrc
```

if `[dest]` is your destination folder (e.g. `/usr/bin` if you have root previlidge or `~/bin` if not.).

C. Self-Check after Installation

After installation, you should open a new terminal emulator and excute `autozip` with NO arguments and use `sudo` if you installed it with root previlidge. This can form a report for your system. The output of which are as follows (We assume that you have ALL the depencencies installed.):

1) Copyright information:

```
$ sudo autozip
YuZJLab AutoZip.
Copyright (C) 2019-2020 YU Zhejian
```

2) Generating a configure file. This will only appear for the first time:

Zhejian YU <Zhejian.19@zju.edu.cn> is a student of Zhejiang University.

```
WARNING: Configure file NOT exist. \
Will generate one by default value.
```

3) Check for all compoments:

```
Start checking all compoments...
Checking for 'tar'...OK
Checking for 'gzip'...OK
Checking for 'pigz'...OK
Checking for 'bgzip'...OK
Checking for 'xz'...OK
Checking for 'bzip2'...OK
Checking for '7z'...OK
Checking for 'zip'...OK
Checking for 'rar'...OK
Checking for 'unzip'...OK
Checking for 'unrar'...OK
Checking for 'parallel' in\
/usr/bin...OK
```

4) Report available extensions:

```
Available extension name on your\
computer: tar, gz, GZ, tar.gz,\
tar.GZ, tgz, bgz, xz, lzma, lz,\
tar.xz, txz, tar.lzma, tlz, bz2,\
tar.bz2, tbz, 7z, zip, rar
```

5) Show you the current configure file:

```
Configure file /usr/bin/autozip.conf\
are as follows:
=====Begin /usr/bin/autozip.conf=====
NOPARALLEL
=====End   /usr/bin/autozip.conf=====
```

By default, GNU Parallel will NOT be used when (de)compressing files. If you want to use GNU Parallel, you should install it to `\usr\bin` and make the first line of `[dest]/autozip.conf PARALLEL`. After altering, you can see `Will use GNU Parallel if possible.` when running the command `autozip` again.

D. Compression

The syntax for compressing data are as follows:

```
autozip [SOURCE] [EXT] [LVL] [OPTS]
```

The argument `[SOURCE]` is the source of data. It can be a file or a folder under the working directory. `[EXT]` is the extension for compressed file. You can get a list of available extensions on your computer by running `autozip` without arguments. `[LVL]` is compression level. The RULES of arranging these arguments are as follows:

1) The arguments should be in a STRICT order of `[SOURCE] [EXT] [LVL]`.
2) You can ommit the arguments on the button. This means that you can omit `[LVL]`, both `[EXT]` and `[LVL]` or ommit all the arguments. If `[LVL]` is omitted, we'll use the default compression level provided by the algorithm. If both `[EXT]` and `[LVL]` are omitted, we'll compress FILES to `gz` and FOLDERS to `tar.gz`, which is widely used under GNU/Linux. If all the arguments are ommitted, AutoZip will run a self-check and print the resluts to your terminal.
3) For `[LVL]`, different software use different levels. There are 0-9 for XZ Utils, 0-5 for RAR, 0,1,3,5,7,9 for 7-Zip and 1-9 for gzip and bzip2. There will be no compression level for GNU Tar. For each level, 0 indicates "store only" (EXCEPT for XZ Utils!), which means just like GNU Tar, they make a folder a file instead of compressing a folder.

You can refer to Table I at page 4 for details. Available options:

- -h |–help
  Print standard Usage: this should be in the file `autozip.Usage`.
- -v |–version
  Show verion information.
- -s[:SPLIT] |–split[:SPLIT]
  Split the data by `[SPLIT]`. The standard `[SPLIT]` varies by format. For rar it is numbers+b/k/m; for zip, it is numbers+k/m/g/t; for 7z, it is numbers+b/k/m/g, for other archive, it is number only (bytes) & numbers+K/M/G/T/P/E/Z/Y (1024 based) or numbers+KB/MB/GB/TB/PB/EB/ZB/YB (1000 based). You can refer to Table II at page 4 for details.
- –force-parallel:[PATH_TO_PARALLEL]
  Force to use GNU Parallel in `[PATH_TO_PARALLEL]` insetad of standard `\usr\bin\parallel`.

E. Decompression

The syntax of autounzip are as follows:

```
autozip [SOURCE] [OPTS]
```

Extract `[SOURCE]`. If the archive is SPLITTED, suffix like "`001`" or "`z01`" should not be added. HOWEVER, for rar you should indicate the FIRST part. For example, "`r1.part1.rar`". Available options:

- -h |–help
  Same as those in autozip.
- -v |–version
  Same as those in autozip.
- –force-parallel:[PATH_TO_PARALLEL]
  Same as those in autozip.

III. Special Warnings

This section lists common mistakes you may make. **YOU SHOULD READ THIS SECTION WITH EXTRA CARE.**

1) DO NOT USE the 7-zip format for backup purpose on Linux/Unix because 7-zip does not store the owner/group of the file.
2) RAR is a PROPERTY software (while UNRAR is not) and use ABSOLUTE PATH when using RAR.
3) when you're about to compress & decompress files with lz, lzma, tar.lzma, tlz extensions, we'll use XZ Utils.
4) You should use GZIP to produce a GZIP COMPRESSED DATA instead of BGZIP to produce a BLOCKED GNU ZIP FORMAT which is only used by bioinformaticians! To produce a BLOCKED GNU ZIP

FORMAT with ”gz” extension, you should use `[EXT]` as `bgz`. Because it is capable of GNU GZip, we’ll not distinguish them when extracting data. ”tar.gz” is not supported in bgz format because it sounds weird.
5) DO PAY ATTENTION WHEN USING GNU PARALLEL! IT CONSUMES A LOT OF COMPUTER RESOURCES AND MAY CAUSE YOU COMPUTER TO BE “DEAD”. NEVER USE IT ON A PUBLIC COMPUTER OR ANY OTHER MACHINE THAT DE NOT BELONGS TO YOU! e.g, computing clusters.

IV. How AutoZip Compress Your Data

The route of AutoZip are as follows:

1) Load `libautozip`.
2) Check all compoments by function `preck` in `libautozip`.
3) Get all the commandline arguments and devide them into two parts: options and other arguments by function `isopt` in `libautozip`. Options are then checked by Regular Expressions.
4) Check if the filename, extension name and `[SPLIT]` value is valid.
5) Start making archive.

If you use GNU Parallel by altering default `autozip.conf` or by `--force-parallel`, the archive-making process is:

1) If you use gzip, we’ll turn to pigz (if available) for help. If there’s no pigz installation, we’ll use GBNU Parallel.
2) if you use bzip2, we’ll use GNU Parallel.
3) If you use XZ Utils (for xz or lzma format), we’ll use as much thread as your macjine can by the algorithm (add `-T0` option).
4) If you use RAR or 7-Zip, we will use 8 threads by the algorithm.
5) If you use Zip or GNU Tar, there will be no difference form ordinary compression.

If you want to split your archive, the archive-making process is:

1) If the target is a FOLDER, we firstly TAR it, then we split the tar to a temporary directory and finally compress the pieces respectively.
2) If the target is a FILE, we will split it to a temporary directory and finally compress the pieces respectively.
3) If you use RAR, Zip or 7-Zip, we’ll use the splitting function provided by the corresponding algorithms.

The reason why we split before compress rather than compress then split is that when extracting the splitted archives, the splitted pieces can be parallelly decompressed.

V. How AutoUnZip Deompress Your Data

The route of AutoUnZip are as follows:

1) Load `libautozip`.
2) Check all compoments by function `preck` in `libautozip`.
3) Get all the commandline arguments and devide them into two parts: options and other arguments by function `isopt` in `libautozip`. Options are then checked by Regular Expressions.
4) Check if the filename is valid.
5) Start decompressing archive.

If you use GNU Parallel by altering default `autozip.conf` or by `--force-parallel`, the archive decomprocession process is:

1) If you use gzip, we’ll turn to pigz (if available) for help. If there’s no pigz installation, there will be no difference form ordinary decompression.
2) If you use XZ Utils (for xz or lzma format), we’ll use as much thread as your macjine can by the algorithm (add `-T0` option).
3) If you use RAR or 7-Zip, we will use 8 threads by the algorithm.
4) If you use Zip, bz2 or GNU Tar, there will be no difference form ordinary compression.

If you want to decompress a splitted archive, the archive decomprocession process is:

1) Archives are parallely (with GNU Parallel. We `echo`ed the decompression commands into different scripts and let GNU Parallel to execute them parallelly) or orderly decompressed to a temporary folder.
2) The decompressed files are then assembled into a TAR file (for FOLDERS) or the original file (for FILE). Then, it is moved & extracted to your working directory.
3) If you use RAR or 7-Zip, we’ll use the splitting function provided by the corresponding algorithms.
4) If you use Zip, we’ll use `zip -FF` to assemble the archive and then, decompress it.

VI. Code Availibility

The code is available on https://github.com/YuZJLab/LinuxMiniPrograms/tree/master/AutoZip.

VII. Known Problems

1) Can NOT extract zip archives created by KuaiZip [5].
2) Unstable when extracting splitted zip archives under GNU/Linux UnZip [6].

References

[1] Free Software Foundation, “tar 1.32,” http://www.gnu.org/software/tar/manual. I
[2] ——, “gzip,” http://www.gnu.org/software/tar/manual. I
[3] Tukaani, “Xz utils 5.2.4,” https://tukaani.org/xz/. I
[4] J. Seward, “bzip2 1.0.8,” https://sourceware.org/bzip2/, 1996–2019. I
[5] KuaiZip.com, “Kuaizip - ultra fast compressor,” http://www.kuaizip.com/en/, 2010–2013. 1
[6] Info-ZIP, “Unzip 6.00,” http://www.kuaizip.com/en/. 2

[EXT]	Real Extension	Software	Command	[LVL]
tar	tar	GNU Tar	tar	NA
gz, GZ	gz, GZ	gzip & pigz	gzip & pigz	1-9
tar.gz, tgz, tGZ	tar, gz, tgz, tGZ	GNU Tar, -	tar, -	-
bgz	gz	bgzip	bgzip	-
bz2	bz2	bzip2	bzip2	-
tar.bz2, tbz	tar.bz2, tbz	GNU Tar, -	tar, -	-
xz	xz	XZ Utils	xz	0-9
tar.xz, txz	tar.xz, txz	GNU Tar, -	tar, -	-
lzma, lz	lzma, lz	XZ Utils	xz	-
tar.lzma, tlz	tar.lzma, tlz	GNU Tar, -	tar, -	-
zip	zip	Zip, UnZip	zip, unzip	-
7z	7z	p7zip	7za	0, 1, 3, 5, 7, 9
rar	rar	RAR, UNRAR	rar, unrar	1-9

TABLE I

A Table for All Extensions and Levels

[EXT]	Splitted by	[SPLIT]	Paralleled by
bz2, tar.bz2, tbz	GNU Split	number only (bytes) & numbers+K/M/G/T/P/E/Z/Y (1024 based) & numbers+KB/MB/GB/TB/PB/EB/ZB/YB (1000 based)	GNU Parallel
tar, gz, GZ, tar.gz, tgz, tGZ,	-	-	pigz, -
xz, tar.xz, txz, lzma, lz, tar.lzma, tlz	-	-	XZ Utils
bgz	NA	NA	GNU Parallel
zip	ZIP	numbers+k/m/g/t	NA
7z	p7zip	numbers+b/k/m/g	p7zip
rar	RAR	numbers+b/k/m	RAR

TABLE II

A Table for All Split & Parallel Mthods