

## برخی از انواع ابزار فشرده سازی در Ubuntu

### Tar

Tar یک ابزار محبوب برای فشرده سازی چندین فایل در یک فایل آرشیو واحد است. این برنامه افزودنی های مختلفی را پشتیبانی می کند که برخی از محبوب ترین آنها عبارتند از tar، tar.gz، و tar.bz2. در آخرین توزیع اوبونتو، tar ابزار فشرده سازی پیش فرض است. دستور tar می تواند به صورت زیر استفاده کرد.

```
$ tar [flags] destinationFileName sourceFileName
```

### Gzip

Gzip یک ابزار فشرده سازی متن باز است که به دلیل ویژگی فشرده سازی فایل بالا شهرت دارد. این ابزار به GNU zip اشاره دارد که با هر نرم افزار گنو سازگار است و پسوند gz. دارد. Gzip دارای نرخ فشرده سازی بالایی است زیرا می تواند اندازه را تا 90 درصد کاهش دهد که بسیار شگفت انگیز است.

```
$ gzip [OPTION]... [FILE]...
```

## Bzip2

bzip2، یک ابزار فشرده‌سازی منبع باز که از الگوریتم فشرده‌سازی مرتب‌سازی بلوک Burrows-Wheeler و کدگذاری هافمن برای فشرده‌سازی فایل‌ها استفاده می‌کند. این یک بایگانی‌کننده فایل نیست، بنابراین فقط از فشرده‌سازی تک فایل پشتیبانی می‌کند، اما از روش‌های فشرده‌سازی مختلف مانند آرایه بیت پراکنده، جداول هافمن، طول اجرا و غیره پشتیبانی می‌کند. در هنگام فشرده‌سازی و رفع فشرده‌سازی فایل‌ها در مقایسه با gzip، حافظه بالایی مصرف می‌کند و کاهش فشرده‌سازی کندتری دارد. اما به طور کلی، bzip2 به دلیل قابلیت فشرده‌سازی قوی، ابزار فشرده‌سازی مناسبی است.

```
$ bzip2 option(s) filenames
```

## Zstandard

Zstandard که zstd نیز نامیده شد در فیس بوک توسط Yann Collet که یک الگوریتم فشرده‌سازی داده بدون تلفات است توسعه یافت. نسبت فشرده‌سازی بالایی دارد و همچنین ویژگی خاصی به نام فشرده‌سازی دیکشنری برای داده‌های کوچک ارائه می‌دهد.

```
$ zstd [OPTIONS] [-|INPUT-FILE] [-o OUTPUT-FILE]
```

## LZ4

LZ4 ابزار فشرده‌سازی عالی برای سرعت فشرده‌سازی بالا است زیرا می‌توان بیش از نیم گیگابایت داده را در هر ثانیه توسط آن فشرده کرد. با استفاده از الگوریتم‌های فشرده‌سازی بدون اتلاف LZ4\_HC و LZ77 با انگیزه ارائه سرعت فشرده‌سازی و رفع فشرده‌سازی بالا ساخته شده است. سرعت فشرده‌سازی را می‌توان به صورت پویا با افزایش سرعت فشرده‌سازی داده‌ها تغییر داد.

```
$ lz4 [OPTIONS] [-|INPUT-FILE] <OUTPUT-FILE>
```

## lzop

lzop یک کمپرسور فایل است که با استفاده از کتابخانه فشرده سازی داده های LZO ساخته شده و کاملاً شبیه gzip است. با مبادله مقداری نسبت نرخ تراکم، سرعت فشرده سازی و رفع فشرده سازی بالاتری را در مقایسه با zip فراهم می کند. فایل ها را می توان به دو پسوند فایل tar و tzo. فشرده کرد. معمولاً فقط از ده درصد از CPU برای تولید فایل های کمی بزرگتر از gzip استفاده می کند.

```
$ lzop [ command ] [ options ] [ filename ... ]
```

## P7zip

p7zip یک ابزار کاربردی خط فرمان یونیکس/لینوکس از 7-Zip است که فایل ها را به فرمت 7z همراه با نسبت فشرده سازی بالا بایگانی می کند. همچنین از بسیاری از فرمت های محبوب مانند tar، bzip2، zip، xz و gzip پشتیبانی می کند. این یک ویژگی رمزگذاری را ارائه می دهد که از فناوری استاندارد رمزگذاری پیشرفته (AES – 256) برای رمزگذاری فایل بایگانی استفاده می کند. فایل های فشرده خراب را می توان با استفاده از آن بازیابی کرد.

```
$ 7z <command> [<switch>...] <base_archive_name> [<arguments>...] [<@listfiles...>]
```

## Pigz

Pigz به پیاده سازی موازی gzip اشاره دارد که جایگزینی برای gzip با کاهش زمان فشرده سازی است. کتابخانه های Zlib و pthread را پیاده سازی می کند که از هسته ها و پردازنده های موجود برای فشرده سازی فایل ها استفاده می کنند که در مقایسه با gzip سریع تر است. به بیان دیگر، عملکرد آن مشابه gzip است که فایل های بزرگ را تنها با استفاده از یک دستور فشرده می کند.

```
$ pigz filename
```

## Zip

Zip ابزار فشرده‌سازی داده‌های بدون تلفات است که به شما کمک می‌کند آرشیوهای فشرده ایجاد کنید. Zip می‌تواند یک یا چند فایل فشرده را در یک فایل آرشیو ذخیره کند و به راحتی از یک سیستم به سیستم دیگر منتقل شود. از پلتفرم‌های مختلفی که فایل‌های فشرده را می‌توان استخراج کرد، پشتیبانی می‌کند. کل ساختار دایرکتوری را می‌توان در یک فایل فشرده در اجرای یک فرمان فشرده کرد.

```
$ zip [option] output_file_name input1 input2
```

## XZ Utils

XZ Utils یک ابزار کمپرسور داده بدون تلفات رایگان است که می‌تواند فایل‌های xz و lzma را فشرده و از حالت فشرده خارج کند. هنگام فشرده‌سازی داده‌ها، نرخ‌های فشرده‌سازی بالاتری نسبت به gzip و bzip2 دارد، اما سرعت فشرده‌سازی و رفع فشرده‌سازی برای نرخ‌های فشرده‌سازی بالاتر بسیار کندتر از gzip است. همچنین ویژگی‌هایی مانند ردیابی خطا در هنگام فشرده‌سازی فایل‌ها را ارائه می‌دهد. فرمت فایل بومی آن است که از الگوریتم LZMA برای فشرده‌سازی فایل استفاده می‌کند.

```
$ xz [option...] [file...]
```

## فشرده سازی فایل text1.txt با استفاده از ابزار فشرده سازی نام برده شده :

Tar :

```
shahrooz@shahrooz:~$ ls -sh text1.txt
42M text1.txt
shahrooz@shahrooz:~$ tar -cf text.tar text1.txt
shahrooz@shahrooz:~$ ls -sh text.tar
42M text.tar
shahrooz@shahrooz:~$ S
```

ابتدا اندازه فایل text1.txt را بررسی میکنیم: 42M  
سپس با استفاده از ابزار Tar آن را فشرده میکنیم  
سپس اندازه فایل فشرده شده را بررسی میکنیم  
مشاهده میشود که در اندازه فایل فشرده تغییری ایجاد  
نشده بدین ترتیب :

Compression ratio = 1

Space savings = 0 %

Gzip :

```
shahrooz@shahrooz:~$ ls -sh text1.txt
42M text1.txt
shahrooz@shahrooz:~$ gzip -c text1.txt > text.gz
shahrooz@shahrooz:~$ ls -sh text.gz
6.7M text.gz
shahrooz@shahrooz:~$ █
```

ابتدا اندازه فایل text1.txt را بررسی میکنیم: 42M  
سپس با استفاده از ابزار gzip آن را فشرده میکنیم  
سپس اندازه فایل فشرده شده را بررسی میکنیم  
مشاهده میشود که که اندازه فایل پس از فشرده سازی  
به 6.7M میرسد. بدین ترتیب :

Compression ratio  $\cong$  6

Space savings  $\cong$  84 %

## Bzip2 :

```
shahrooz@shahrooz:~$ ls -sh text1.txt
42M text1.txt
shahrooz@shahrooz:~$ bzip2 -c text1.txt > text.bz2
shahrooz@shahrooz:~$ ls -sh text.bz2
4.3M text.bz2
shahrooz@shahrooz:~$ █
```

ابتدا اندازه فایل text1.txt را بررسی میکنیم: 42M  
سپس با استفاده از ابزار Bzip2 آن را فشرده میکنیم  
سپس اندازه فایل فشرده شده را بررسی میکنیم  
مشاهده میشود که که اندازه فایل پس از فشرده سازی  
به 4.3M میرسد. بدین ترتیب :

Compression ratio  $\cong 10$

Space savings  $\cong 90 \%$

## Zstd :

```
shahrooz@shahrooz:~$ ls -sh text1.txt
42M text1.txt
shahrooz@shahrooz:~$ zstd -c text1.txt > text.zst
shahrooz@shahrooz:~$ ls -sh text.zst
5.5M text.zst
shahrooz@shahrooz:~$ █
```

ابتدا اندازه فایل text1.txt را بررسی میکنیم: 42M  
سپس با استفاده از ابزار Zstd آن را فشرده میکنیم  
سپس اندازه فایل فشرده شده را بررسی میکنیم  
مشاهده میشود که که اندازه فایل پس از فشرده سازی  
به 5.5M میرسد. بدین ترتیب :

Compression ratio  $\cong 8$

Space savings  $\cong 87 \%$

Lz4 :

```
shahrooz@shahrooz:~$ ls -sh text1.txt
42M text1.txt
shahrooz@shahrooz:~$ lz4 -c text1.txt > text.lz4
shahrooz@shahrooz:~$ ls -sh text.lz4
9.7M text.lz4
shahrooz@shahrooz:~$
```

ابتدا اندازه فایل text1.txt را بررسی میکنیم: 42M  
سپس با استفاده از ابزار Lz4 آن را فشرده میکنیم  
سپس اندازه فایل فشرده شده را بررسی میکنیم  
مشاهده میشود که که اندازه فایل پس از فشرده سازی  
به 9.7M میرسد. بدین ترتیب :

Compression ratio  $\cong 4.3$

Space savings  $\cong 77\%$

Lzop :

```
shahrooz@shahrooz:~$ ls -sh text1.txt
42M text1.txt
shahrooz@shahrooz:~$ lzop -c text1.txt > text.lzo
shahrooz@shahrooz:~$ ls -sh text.lzo
11M text.lzo
shahrooz@shahrooz:~$ █
```

ابتدا اندازه فایل text1.txt را بررسی میکنیم: 42M  
سپس با استفاده از ابزار Lzop آن را فشرده میکنیم  
سپس اندازه فایل فشرده شده را بررسی میکنیم  
مشاهده میشود که که اندازه فایل پس از فشرده سازی  
به 11M میرسد. بدین ترتیب :

Compression ratio  $\cong 3.8$

Space savings  $\cong 74\%$

P7zip :

```
shahrooz@shahrooz:~$ ls -sh text1.txt
42M text1.txt
shahrooz@shahrooz:~$ p7zip -c text1.txt > text.7z
shahrooz@shahrooz:~$ ls -sh text.7z
3.8M text.7z
shahrooz@shahrooz:~$
```

ابتدا اندازه فایل text1.txt را بررسی میکنیم: 42M  
سپس با استفاده از ابزار P7zip آن را فشرده میکنیم  
سپس اندازه فایل فشرده شده را بررسی میکنیم  
مشاهده میشود که که اندازه فایل پس از فشرده سازی  
به 3.8M میرسد. بدین ترتیب :

Compression ratio  $\cong 11$

Space savings  $\cong 91\%$

Pigz :

```
shahrooz@shahrooz:~$ ls -sh text1.txt
42M text1.txt
shahrooz@shahrooz:~$ pigz -c text1.txt > text-pigz.gz
shahrooz@shahrooz:~$ ls -sh text-pigz.gz
6.7M text-pigz.gz
shahrooz@shahrooz:~$
```

ابتدا اندازه فایل text1.txt را بررسی میکنیم: 42M  
سپس با استفاده از ابزار Pigz آن را فشرده میکنیم  
سپس اندازه فایل فشرده شده را بررسی میکنیم  
مشاهده میشود که که اندازه فایل پس از فشرده سازی  
به 6.7M میرسد. بدین ترتیب :

Compression ratio  $\cong 6$

Space savings  $\cong 84\%$



Zip :

```
shahrooz@shahrooz:~$ ls -sh text1.txt
42M text1.txt
shahrooz@shahrooz:~$ zip text.zip text1.txt
  adding: text1.txt (deflated 84%)
shahrooz@shahrooz:~$ ls -sh text.zip
6.7M text.zip
shahrooz@shahrooz:~$
```

ابتدا اندازه فایل text1.txt را بررسی میکنیم: 42M  
سپس با استفاده از ابزار Zip آن را فشرده میکنیم  
سپس اندازه فایل فشرده شده را بررسی میکنیم  
مشاهده میشود که که اندازه فایل پس از فشرده سازی  
به 6.7M میرسد. بدین ترتیب :

Compression ratio  $\cong 6$

Space savings  $\cong 84\%$

XZ Utils :

```
shahrooz@shahrooz:~$ ls -sh text1.txt
42M text1.txt
shahrooz@shahrooz:~$ xz -c text1.txt > text.xz
shahrooz@shahrooz:~$ ls -sh text.xz
3.7M text.xz
shahrooz@shahrooz:~$
```

ابتدا اندازه فایل text1.txt را بررسی میکنیم: 42M  
سپس با استفاده از ابزار XZ آن را فشرده میکنیم  
سپس اندازه فایل فشرده شده را بررسی میکنیم  
مشاهده میشود که که اندازه فایل پس از فشرده سازی  
به 3.7M میرسد. بدین ترتیب :

Compression ratio  $\cong 11.3$

Space savings  $\cong 91.1\%$

Result :

Compression tools	Uncompressing size	Compressing size	Compression ratio	Space savings
XZ Utils	42M	3.7M	11.3	91.1 %
P7zip	42M	3.8M	11	91 %
Bzip2	42M	4.3M	10	90 %
Zstd	42M	5.5M	8	87 %
Gzip	42M	6.7M	6	84%
Pigz	42M	6.7M	6	84 %
Zip	42M	6.7M	6	84 %
Lz4	42M	9.7M	4.3	77 %
Lzop	42M	11M	3.8	74 %
Tar	42M	42M	1	0 %

□ مطابق جدول فوق بهترین تا بدترین ابزار فشرده سازی را از بالا به پایین مشاهده میکنید.