

Archiwizacja i kompresja zbiorów w Linuksie

Różnice: archiwizacja to łączenie wielu plików w jeden zbiór, kompresja to zmniejszanie jego rozmiaru.

- tar
- gz
- bz2
- zip

tar

Rozpakowanie jednego archiwum:

`tar xfv archiwum.tar`

(x = extract - ekstrakcja plików z archiwum, f = file - plik archiwum, v = verbose - tryb gadatliwy)

Tworzenie archiwum plików / katalogów:

`tar cfv archiwum.tar zawartosc1 zawartosc2 zawartosc3`

(c = create - twórz archiwum)

Tworzenie skompresowanego archiwum:

`tar cfzv archiwum.tar zawartosc1 zawartosc2 zawartosc3`

(z = kompresja gzip)

Odczyt zawartości archiwum:

`tar tfv archiwum.tar`

gz

Kompresja pliku

`gzip plik`

Wynik: plik.gz

Dekompresja pliku:

`gunzip plik`

Zgromadzenie plików w jednym skompresowanym archiwum:

`tar cfvz archiwum.tar.gz plik1 plik2`

Wynik: archiwum.tar.gz

Dekompresja archiwum:

`tar xfvz archiwum.tar.gz`

bz2

Kompresja pliku

bzip2 plik

Wynik: plik.bz2

Dekompresja pliku:

bunzip2 plik.bz2

Zgromadzenie plików w jednym skompresowanym archiwum:

tar cfvj archiwum.tar.bz2 plik1 plik2

Wynik: archiwum.tar.bz2

Dekompresja archiwum:

tar xfvj archiwum.tar.bz2

zip

Zgromadzenie plików w jednym skompresowanym archiwum:

zip archiwum.zip plik1 plik2

Zgromadzenie kompletnych katalogów w jednym skompresowanym archiwum:

zip -r archiwum.zip katalog1 katalog2 katalog3

Dekompresja archiwum:

unzip archiwum.zip

Odczyt zawartości archiwum:

unzip -l archiwum.zip

Kolejne parametry i funkcje odpowiedniego programu opisane są na jego stronie MAN. Np. w przypadku programu tar to:

man tar

Niskopoziomowego kopiowania i konwersji surowych danych

dd - uniksowy program do niskopoziomowego kopiowania i konwersji surowych danych (ang. raw data). dd jest używane do kopiowania określonej liczby bajtów lub bloków danych wraz z opcjonalnym konwertowaniem kopiowanych danych

- Bezpośrednie klonowanie całego dysku na inny dysk

składnia: dd if=/dysk źródłowy of=/dysk docelowy

przykład: **dd if=/dev/sda of=/dev/sdb**

w przykładzie dysk sda jest klonowany na dysk sdb. Oba dyski nie mogą być zamontowane.

Uwaga: pełny sukces uzyskamy dla dwóch identycznych dysków >>> ta sama marka i model. Dla różnych dysków efekt może być różny.

- Wykonanie kopii zapasowej partycji

składnia: dd if=/partycja źródłowa of=/katalog/nazwa obrazu

przykład1: **dd if=/dev/sda1 of=/media/sda2/backup.img**

przykład2: **dd if=/dev/sda1 | gzip > /media/sda2/backup.gz**

w przykładzie wykonywana jest kopia partycji sda1 do pliku obrazu „backup.img” na pierwszej partycji dysku sdb. W przykładzie2 obraz kopii jest dodatkowo kompresowany.

- Przywrócenie partycji z kopii zapasowej

składnia: dd if=/katalog/nazwa obrazu of=/partycja docelowa

przykład1: **dd if=/media/sda2/backup.img of=/dev/sda1**

przykład2: **gzip -dcv /media/sda2/backup.gz | dd of=/dev/sda1**

w przykładzie przywracana jest partycja sda1 z utworzonej kopii backup.img z partycji sda2.

W przykładzie2 obraz kopii jest dodatkowo dekompresowany.

- Wykonanie kopii zapasowej kodu MBR

dd if=/dev/sda of=/media/sda1/mbrsda.bak bs=512 count=1
(cały kod MBR)

dd if=/dev/sda of=/media/sda1/mbrsda.bak bs=446 count=1
(bez tablicy partycji)

- Przywrócenie MBR z kopii zapasowej

```
dd if=/media/sda1/mbrsda.bak of=/dev/sdx
```

(przywraca cały MBR; stosujemy tylko, gdy oba dyski mają partycje o identyczne wielkości; nie ma potrzeby dodawania parametrów)

```
dd if=/media/sda1/mbrsda.bak of=/dev/sdx bs=446 count=1
```

(przywraca tylko kod rozruchowy, pomija tablice partycji)

- Kopia i przywracanie tablicy partycji

```
sfdisk -d /dev/sda > /media/sdb1/backup-hda.sf
```

```
sfdisk /dev/sda < /media/sdb1/backup-hda.sf
```

- Przenieść zawartość całego dysku do pliku:

```
dd if=/dev/sda of=/mnt/sdb1/file1
```

- Kasowanie całego dysku (wraz z MBR i wszystkimi partycjami) przez nadpisanie losowymi danymi:

```
dd if=/dev/urandom of=/dev/sdb
```

dump – polecenie do tworzenia kopii zapasowej systemu plików

Aby utworzyć kopię zapasową partycji /usr w pliku o nazwie usr.dump w katalogu bieżącym (który znajduje się poza /usr), wpisz:

```
dump -0 -a -L -f usr.dump /usr
```

Aby utworzyć kopię zapasową partycji głównej / w pliku o nazwie root.dump w katalogu bieżącym, wpisz:

```
dump -0 -a -L -f root.dump
```

Tworzenie kopii zapasowej:

```
dump -f - /home | gzip -c > /kopia/kopiazapasowa.gz
```

Dekompresja archiwum

```
tar zxvf kopiazapasowa.gz
```

tar

Od wersji GNU tar 1.35 (2023) dodano lepsze wsparcie dla formatów POSIX i obsługę długich nazw plików.

gz

gzip pozostaje popularny, ale dla lepszej kompresji zaleca się zstd.

bz2

bzip2 jest rzadziej używany ze względu na wolne działanie. Zstd i xz są preferowane.

zip

zip obsługuje teraz lepsze szyfrowanie AES i kompresję Deflate64.

dd

dd obsługuje teraz parametr status=progress dla wyświetlania postępu kopiowania.

dump

dump jest rzadziej używany; alternatywy to rsync, borgbackup, restic.

Narzędzia zalecane w 2025:

Zstandard – nowoczesny algorytm kompresji, szybszy i bardziej efektywny niż gzip i bzip2.

Kompresja: **zstd plik**

Dekompresja: **zstd -d plik.zst**

- pigz: równoległa wersja gzip:

Parallel Implementation of Gzip – wykorzystuje wiele rdzeni CPU do szybszej kompresji.

Kompresja: **pigz plik**

Dekompresja: **unpigz plik.gz**

- borgbackup: deduplikacja i szyfrowanie kopii zapasowych:

Zaawansowane narzędzie do tworzenia kopii zapasowych z deduplikacją i szyfrowaniem.

Tworzenie kopii: **borg create /ścieżka/do/repo::nazwa-archiwum /folder**

- restic: szybkie, bezpieczne kopie zapasowe:

Szybkie i bezpieczne narzędzie do backupu z szyfrowaniem.

Tworzenie kopii: **restic backup /folder**

- duf: nowoczesny df:

Nowoczesne narzędzie do analizy użycia dysku – alternatywa dla df.

Wywołanie: **duf**

- btrfs/zfs: systemy plików z wbudowaną archiwizacją:

System plików z wbudowaną obsługą snapshotów i kompresji.

Tworzenie snapshotu: `btrfs subvolume snapshot /mnt/data /mnt/snapshot`

- zstd: kompresja o wysokiej wydajności:

Zaawansowany system plików z obsługą kompresji, snapshotów i replikacji.

Tworzenie snapshotu: `zfs snapshot pool/dataset@snapshot`