

Polecenie dd

Jest to podstawowe polecenie służące do kopiowania plików w systemie Linux. Dlaczego jest takie fundamentalne, ponieważ w systemie Linux wszystko jest plikiem. Przy korzystaniu z tego polecenia trzeba dać pozor na poprawność semantyczną wprowadzanych parametrów. Pomyłka może doprowadzić do utraty danych. Można przyjąć, że dd jest akronimem od *delete data*.

Przykład. Kopia zapasowa pierwszego rekordu na dysku. Trzeba pamiętać, że polecenie *dd* wymaga uprawnień administracyjnych. Parametr *if* oznacza plik wejściowy, a *of* plik wyjściowy. Parametr *bs* oznacza rozmiar bloku. Parametr *count* określa liczbę bloków. Generalnie, należy pamiętać, że identyfikacja dysków w systemie Linux odbywa się przez pliki, na przykład /dev/sda, /dev/sdb, /dev/sdv i tak dalej. Poszczególne partycje identyfikuje się dodając do nazwy numer kolejnej partycji, na przykład dla dysku /dev/sda pierwsza partycja to /dev/sda1, druga /dev/sda2 i tak dalej.

```
dd if=/dev/sda bs=512 count=1 of=mbr.img
```

Przy pomocy polecenia *xxd* można wyświetlić zawartość pliku binarnego w zapisie heksadecymalnym lub binarnym. Dla binarnego trzeba użyć parametr *-b*.

```
xxd mbr.img  
xxd -b mbr.img
```

W ten sam sposób można zrobić kopię całej partycji lub klon w inne miejsce:

```
dd if=/dev/sdb1 of=/dev/sdc1
```

Możliwość zrobienia klonu całego dysku:

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

Przykład. Tworzenie pliku o wielkości 100 bajtów wypełnionymi losowymi danymi.

```
dd if=/dev/urandom of=file.bin bs=100 count=1  
cat /dev/urandom  
dd if=/dev/urandom of=file.bin bs=1 count=1; cat file.bin  
  
cat /proc/sys/kernel/random/uuid  
echo Polska > `cat /proc/sys/kernel/random/uuid`.txt
```

Przykład. W tym przykładzie wykorzystuje się polecenie *dd* do tworzenia pliku o wielkość 1kB, a następnie za pomocą polecenia *tr* zamienia się puste znaki na znaki A:

```
dd if=/dev/zero bs=1k count=1 | tr "\000" "\101" > file.txt  
echo A | xxd -b
```

Przykład. Poniższy przykład pokazuje, jak sprawdzić czas odczytu dużego pliku *big_file.bin*. Po każdym wykonaniu polecenia *dd* wyświetlany jest czas wykonania kopiowania:

```
dd if=big_file.bin of=/dev/null bs=1024
```

Przykład. Czyszczenie dysku /dev/sda:

```
dd if=/dev/zero of=/dev/sda bs=4096
```

Przykład. Tworzenie obrazu typu iso z nośnika CD:

```
dd if=/dev/sr0 of=file.iso bs=2048 conv=nosync
```

Polecenie *dd* posiada dodatkowe parametry, które umożliwiają tworzenie plików o danym przesunięciu (ang. offset) względem początku pliku.

Przykład. Kopiowanie pliku począwszy od drugiego bajta – parametr *skip*. Aby przesunąć o jeden bajt, kopiowanie musi być wykonane w blokach 1-bajtowych:

```
dd if=file.txt of=file2.txt bs=1 skip=1
```

Przykład. Kopiowanie pliku począwszy od pozycji 3, licząc w bajtach – parametr *seek*:

```
dd if=file.txt of=file2.txt bs=1 seek=3
```

Przykład. Konkatenacja dwóch plików. Mając dwa pliki *file1.bin* oraz *file2.bin* o pojemnościach po 100 bajtów:

```
dd if=file1.bin bs=1 count=100 of=file.bin
```

```
dd if=file2.bin bs=1 count=100 of=file.bin seek=100
```

Przykład. Tworzenie urządzenia blokowego z pliku, w którym korzysta się z polecenia *dd*:

```
dd if=/dev/zero of=file.img bs=1k count 10000
```

```
losetup /dev/loop0 file.img
```

```
mke2fs -c /dev/loop0 10000
```