

## Naredba blockdev - izmjena postavki uređaja



U prošlom nastavku naučili smo kako naredbom blockdev iščitati zadane veličine sektora, bloka... No istom se naredbom te vrijednosti mogu i mijenjati, ako koristimo parametre --set.

Evo koje postavke možemo mijenjati:

```
$ blockdev --help | grep set
--setro          set read-only
--setrw          set read-write
--getalignoff    get alignment offset in bytes
--setbsz <bytes> set blocksize on file descriptor opening the block device
--setra <sectors> set readahead
--setfra <sectors> set filesystem readahead
```

Najprije upada u oči da ne možemo mijenjati veličinu sektora. U prošlom smo nastavku vidjeli da veličinu sektora zadaje proizvođač diska, pa se u to i ne treba petljati. Ali o veličini bloka, a to je već razina datotečnog sustava, možda bismo već mogli sami odlučivati. Blok je ponekad jednak sektoru, obično kod tvrdih diskova manjeg kapaciteta, ali danas je uobičajena vrijednost 4096 bajtova. Evo kako bismo je, bar u teoriji mogli zadati:

```
# sudo blockdev --setbsz 512 /dev/sda1
```

Ali oprezno s time! Ne bi bilo dobro mijenjati veličinu bloka na živom, montiranom datotečnom sustavu! Pogotovo ne na onom na kojem nam je root filesystem! U najmanju ruku pogubili bismo podatke, jer bi narušili strukturu inodova koji pokazuju na datoteke, odnosno na blokove u kojima su smještene. Da bi naredba blockdev sačuvala podatke, morala bi napraviti nešto slično komprimiranju diska na Windowsima: skloniti podatke, promijeniti veličinu bloka, napraviti novu strukturu inodova i onda vratiti podatke. Praktički bi trebalo nanovo napraviti filesystem i onda na njega vratiti podatke. Nema šanse da to može napraviti jednostavna naredba čiji izvršni kod ima svega 35 kilobajta.

Zato ćemo u dok za diskove, spojen na USB port, umetnuti stari disk, izvađen iz rashodovanog računala, na kojem nema ničeg vrijednog, pa na njemu isprobavati što se i kako može napraviti.

```
$ sudo blockdev --getbsz /dev/sdb1
4096
$ sudo blockdev --getpbsz /dev/sdb1
512
```

Fizički blok je velik 512 bajtova, a logički 4k. Pokušajmo logički smanjiti na 512, da bude isti kao fizički.

```
$ sudo blockdev --setbsz 512 /dev/sdb1
blockdev: ioctl error on BLKBSZSET: Ure?aj ili resurs je zauzet
```

Moramo odmontirati particiju.

```
$ umount /dev/sdb1
$ sudo blockdev --setbsz 512 /dev/sdb1
```

Ovog puta nema poruke o grešci. Montirajmo ponovo vanjski disk, pa provjerimo da li smo stvarno promijenili veličinu bloka:

```
$ sudo blockdev --getbsz /dev/sdb1
4096
```

Sve je po starom. Naredba blockdev nije javila grešku, ali na disku se ništa nije promijenilo!

Evo kako bi se promjena veličine bloka napravila školski. Najprije odmontiramo particiju.

```
$ umount /dev/vdb1
```

Kreiramo nanovo particiju s drugačijom veličinom bloka:

```
$ mkfs -t ext4 -b 512 /dev/sdb1
```

Ponovo montiramo particiju i provjerimo da li se promjena "primila":

```
$ mount /dev/sdb1 /test/
$ blockdev --getbsz /dev/sdb1
512
```

Ukratko, veličina bloka zadaje se prilikom formatiranja diska, a ne onako usput, kako se sjetimo. Pa se moramo zapitati: čemu uopće služe "set" parametri naredbe blockdev?

Prije nego donesemo zaključke isprobat ćemo još neke set parametre. Možda uspijemo promijeniti status diska *read-only* i *read-write*? To se tradicionalno zadaje prilikom montiranja, a može se promijeniti parametrom remount:

```
$ sudo mount -o remount ro /dev/sdb1
```

Kako bi to napravili koristeći naredbu blockdev? Najprije pogledajmo zadanu vrijednost.

```
# sudo blockdev --getro /dev/sdb1
0
```

Read only je 0, znači nije postavljeno.

```
# sudo blockdev --setro /dev/sdb1
```

Nema poruke o grešci. Provjerimo da li se promjena primila.

```
# sudo blockdev --getro /dev/sdb1
1
```

Čini se da jest. No kad kopiramo datoteku na tu particiju, ona se presnimi bez problema. Naredba

*mount* pokazuje da je datotečni sustav i dalje montiran u rw načinu. Što ako ga odmontiramo pa nanovo montiramo?

```
$ sudo umount /dev/sdb1
$ sudo mount /dev/sdb1 /mnt
mount: /dev/sdb1 is write-protected, mounting read-only
```

Reklo bi se da smo naredbom *blockdev* stavili oznaku koja će vrijediti za naredno montiranje. Pokušajmo *remount*.

```
$ sudo mount -o remount rw /dev/sdb1
mount: /dev/sdb1 not mounted or bad option
```

Nije uspjelo. Particija je i dalje read-only. Hoće li pomoći ako je odmontiramo pa nanovo montiramo?

```
$ sudo umount /dev/sdb1
$ sudo mount /dev/sdb1 /mnt
mount: /dev/sdb1 is write-protected, mounting read-only
```

Rezultat je zapravo zanimljiv. Na prvi pogled čini se da ovako sadržaj tvrdog diska možemo "zamrznuti", sačuvati ga od brisanja, izmjena. Zamislite da smo na disk spremili *backup* koji želimo sačuvati za budućnost. Primjer koji mi pada na pamet: za potrebe forenzike napravimo *bitcopy* "osumnjičenog" diska na vanjski USB HD. Nakon toga naredbom *blockdev* taj HD proglasimo za *read only*. Sad ćemo moći na miru raditi forenziku, bez straha da ćemo svojim postupcima narušiti originalnu kopiju, što se može pokazati presudnim ako istraga završi na sudu, a sudac odredi vještaka koji će provjeriti postupke forenzičara! Prisjećam se nekadašnjih modela USB stickova s prekidačem kojim ste mogli zabraniti pisanje na njih. To je korisno ako s tim stickom idete okolo i instalirate softver: nećete pokupiti virus ako je stick zaštićen od upisivanja! Eto, sad nam je naredba *blockdev* omogućila da to napravimo softverski, ne reskirajući da će nam se, namjerno ili nenamjerno, prekidač prebaciti u pogrešan položaj.

No jesmo li se prerano obradovali? Kad bi to bilo tako, *blockdev* bi morao upisati *ro* u sam uređaj. Brzi test će nam to začas pokazati: nakon *reboota* disk se montira u read-write načinu. Šteta! Naredba zapravo mijenja vrijednosti postavki u tablicama kernela, izmjene ne prežive reboot sustava.

Pokušajmo promijeniti readahead postavku:

```
$ sudo blockdev --getra /dev/sdb1
256
$ sudo blockdev --setra 512 /dev/sdb1
$ sudo blockdev --getra /dev/sdb1
512
```

Uspjelo je!

Naredba *blockdev* nudi još jednu zanimljivu mogućnost, čišćenje *buffera*, forsiranje snimanja podataka na disk. Za to smo tradicionalno koristili naredbu *sync*, ali možda ćemo poželjeti pospremiti podatke iz privremene memorije samo na jedan disk/particiju. Na primjer prije nego je odmontirate?

```
$ sudo blockdev --flushbufs /dev/sdb1
```

Tako možete "isplahnuti" podatke i na više uređaja odjednom, samo ih nabrojite.

Baš smo htjeli zaključiti kako *blockdev* donosi mogućnosti koje ne možemo koristiti ili suvišnu redundanciju (*sync*). No autor je ove naredbe Andries E. Brouwer, matematičar i programer, doktor znanosti i predavač na sveučilištu. Skoro smo zaboravili: uključen je razvoj kernela, u području geometrije diska i particioniranja. Čovjek zna što radi, samo se mi moramo potruditi da ga razumijemo. Napravio je za nas zanimljivu naredbu, koja nas je navela da naučimo stvari koje smo dosad prešutno podrazumijevali. S ovom serijom članka učenje nije završeno, lako je moguće da se naredbi *blockdev* još vratimo, pogotovo što bi za njeno puno razumijevanje trebalo znati više o tome kako funkcionira kernel!

pon, 2018-04-30 18:18 - Aco Dmitrović **Kuharice:** [Linux](#) [1]

**Kategorije:** [Software](#) [2]

**Vote:** 0

No votes yet

**story\_tag:** [Linux](#) [3]

[blok uređaji](#) [4]

[blockdev](#) [5]

**Source URL:** <https://sysportal.carnet.hr/node/1803>

#### Links

[1] <https://sysportal.carnet.hr/taxonomy/term/17>

[2] <https://sysportal.carnet.hr/taxonomy/term/25>

[3] <https://sysportal.carnet.hr/taxonomy/term/119>

[4] <https://sysportal.carnet.hr/taxonomy/term/232>

[5] <https://sysportal.carnet.hr/taxonomy/term/220>