

Prljavi bit



Kao što lijekovi imaju nuspojave, tako je istraživanje naredbe `blockdev` proizvelo zanimljiv "sporedni efekt". Shvatili smo da sve izmjene konfiguracije koje donosi `blockdev --setxxx` traju samo dok se ne ponište ili dok se računalo ne ugasi. Pokušali smo vanjski disk proglašiti za *read only*, kako bi zamrznuli podatke na njemu, ali nakon *reboota* disk se opet montirao u *read-write* načinu. No jedan USB stick ponašao se drugačije. Svaki put kad bi ga utaknuo u USB port, na njega se nije moglo pisati. Kao da je naredba `blockdev` zaista uspjela zaštititi stick od pisanja? Ili je posrijedi nešto drugo?

Naredba

```
$ sudo blockdev --report /dev/sdb1
RO    RA    SSZ    BSZ    StartSec          Size    Device
rw    256    512    512      18784    15928737792    /dev/sdb1
```

pokazuje je da je stick u *rw* modu, dok `mount` kaže da je `/dev/sdb1` montiran kao *ro*.

```
/dev/sdb1 on /media/hombre/26FE-D885 type vfat (ro,nosuid,nodev,relatime,uid=1000,gid=1000,
fmask=0022,dmask=0022,codepage=437,icharset=iso8859-1,shortname=mixed,showexec,utf8,flush,errors=remount-ro,uhelper=udisks2)
```

Tu vidimo i da se u slučaju da se pronađu greške disk montira u *read only* načinu (`errors=remount-ro`).

Bilo bi nam drago da se stick može softverski zaštititi od upisivanja, jer ga nosimo sa sobom kad pristupamo korisničkim računalima. Ipak, `blockdev` ga ne može trajno zaštititi od pisanja, jer se nakon *reboota* status *ro* gubi, ali zašto se disk ipak montira kao *ro*? O kakvim se greškama radi? Treba napraviti provjeru datotečnog sustava, možda tako nešto otkrijemo.

Ali prije toga želio sam probati još nešto: stick sam ubacio u USB port na Windows računalu. Radio je normalno, bio je otvoren i za čitanje i za pisanje. Pa u čemu se montiranje datotečnog sustava razlikuje na Linuxu i Windowsima?

```
$ sudo fsck /dev/sdb1
fsck from util-linux 2.27.1
fsck.fat 3.0.28 (2015-05-16)
0x41: Dirty bit is set. Fs was not properly unmounted and some data may be corrupt.
1) Remove dirty bit
2) No action
? 1
There are differences between boot sector and its backup.
This is mostly harmless. Differences: (offset:original/backup)
 67:85/5f, 68:d8/d4, 69:fe/0a, 70:26/3a
1) Copy original to backup
2) Copy backup to original
```

3) No action

? 1

Filesystem has 1942527 clusters but only space for 1942526 FAT entries.

Dakle dijagnosticirana su tri problema, a prvi krivac je "prljavi bit", na kojeg se, po svemu sudeći, Windowsi ne obaziru. Barem ne u ovom slučaju. Zanimljivo je da i Windowsi koriste "dirty bit", i tamo se javlja isti problem. Što je prljavi bit? Radi se o oznaci koja kaže da podaci za taj dio memorije nisu sinhronizirani, još su negdje keširani. Znamo da stick ne smijemo samo tako izvaditi iz porta, najprije ga treba "odjaviti", što zapravo znači da se pokreće "sync" pa onda "umount". Unixaši starog kova na to su navikli, njima je svetogrđe ukloniti neki datotečni sustav bez odmontiranja: kao što su ga spajali naredbom mount, tako ga treba odspojiti naredbom umount. S komandne linije, naravno. No mlađi korisnici navikli su na "automatiku", očekuju da će OS sve odraditi za njih. Dok montiranje ide automatski, za odmontiranje to ne vrijedi. Pa ako se stick samo iščupa, eto problema. U praksi sam vidio slučajeve da je nakon toga filesystem toliko korumpiran da se pokazuje kao prazan. (Sad bih se trebao posuti pepelom, ali zaista se ne sjećam da sam stick vadio a da ga nisam prije toga odmontirao! To je "automatika" ugrađena u moje prste).

Osim toga, brzo sam shvatio da sam uzalud birao akcije koje bi *fsck* trebao napraviti da riješi problem, jer su sve greške i dalje prisutne. Da bi se u to uvjerali, dovoljno je ponovo pokrenuti istu naredbu, pa će se sve ponoviti na isti način.

Kako natjerati *fsck* da stvarno napravi te promjene? Man stranica nije baš informativna, ali se na forumima može pronaći informacija kako *fsck* zapravo radi "na suho" (*dry run*), samo demonstrira, odnosno simulira rješavanje problema.

Man stranica spominje parametre koji bi mogli pomoći:

-r , za koji sam pomislio da znači *repair*, ali zapravo znači *report*

-a popravi sve greške bez pitanja

Uz -a je upozorenje da se koristi s oprezom. Također se spominje da je *fsck* samo "front end" koji poziva alate prilagođene datotečnom sustavu. Kako je stick formatiran za Windowse, *filesystem* je *vfat*, kako nam je i pokazala naredba mount. Zato pozivamo man stranicu za *fsck.vfat*. Zanimljivo, ovdje -r znači interaktivni *repair*, -w kaže da se ispravke odmah zapišu na disk, -v je *verbosely*, opisno. Za svaki slučaj najprije ćemo presnimiti sve datoteke, ako popravak ne uspije. Posve očekivano, nekoliko je datoteka korumpirano i nisu se dale kopirati. Možda ih spasi *fsck*:

```
$ sudo fsck.vfat -a -v -w /dev/sdb1
fsck.fat 3.0.28 (2015-05-16)
Checking we can access the last sector of the filesystem
0x41: Dirty bit is set. Fs was not properly unmounted and some data may be corrupt.
  Automatically removing dirty bit.
There are differences between boot sector and its backup.
This is mostly harmless. Differences: (offset:original/backup)
  67:85/5f, 68:d8/d4, 69:fe/0a, 70:26/3a
  Not automatically fixing this.
Filesystem has 1942527 clusters but only space for 1942526 FAT entries.
```

Samo je uklonjen prljavi bit, ostale dvije greške nisu. Zato pokušavamo ponovo, ali sada interaktivno:

```
$ sudo fsck.vfat -r -v -w /dev/sdb1
fsck.fat 3.0.28 (2015-05-16)
Checking we can access the last sector of the filesystem
There are differences between boot sector and its backup.
This is mostly harmless. Differences: (offset:original/backup)
```

```
67:85/5f, 68:d8/d4, 69:fe/0a, 70:26/3a
```

- 1) Copy original to backup
 - 2) Copy backup to original
 - 3) No action
- ? 1

```
Filesystem has 1942527 clusters but only space for 1942526 FAT entries.
```

I još jednom:

```
$ sudo fsck.vfat -r -v -w /dev/sdb1
fsck.fat 3.0.28 (2015-05-16)
Checking we can access the last sector of the filesystem
Filesystem has 1942527 clusters but only space for 1942526 FAT entries.
```

Ostala je zadnja pogreška. Datotečni sustav ima jedan klaster (blok) viška, za njega nema mjesta u FAT tablici, pa mu se ne može pristupiti. Što sad? Obaviti provjeru datotečnog sustava s Windowsa, pa ako to ne uspije, nanovo formatirati particiju? Ne vjerujem da je Linuxov *fsck* lošiji od Microsoftova *chkdsk*. Isti popravak moguće je napraviti s Microsoftovim alatima, ali ja koristim Linux 99,99% posto vremena.

Na forumima se spominje upravo takvo rješenje (kopiraj podatke, *dskchk*, pa format ako ne ide drugačije), a kao uzrok greške navodi se odlazak računala u "sleep mode" ili hibernaciju, prije nego je sinhronizacija dovršena. Dakle keširanje podataka u nekom međuspremniku, koje danas koriste svi operacijski sustavi da bi poboljšali performanse, nije 100% sigurno, ponekad se dogodi da upisivanje iz međuspremnika na disk ne uspije. Kako je notebook star već punih šest godina, baterija mu je oslabila, pa je vjerojatnost ovakve pogreške time veća. Stick sam koristio za ispitivanje rada s naredbom *blockdev*, često sam ga montirao, odmontirao, rebootao računalo, pokušavao sve ponovo. Vjerojatno nisam za to trebao koristiti stick koji koristim svakodnevno, već neki koji mi stoji zaboravljen u ladici. Ako ovaj slučaj ima pouku, onda je to da ni ljudi ni računala nisu 100% pouzdani, uvijek postoji mogućnost greške. "Errare humanum est", to znamo, ali trebalo bi dodati "Errare machinum est" (excuse my latin: ispravno je "Errare machinarum est"). I opet se treba podsjetiti: backup, backup, backup...

čet, 2018-05-31 22:05 - Aco Dmitrović **Kuharice:** [Linux](#) [1]

Kategorije: [Operacijski sustavi](#) [2]

Vote: 0

No votes yet

story_tag: [dirty bit](#) [3]

[korumpiran datotečni sistem](#) [4]

Source URL: <https://sysportal.carnet.hr/node/1811>

Links

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

[2] <https://sysportal.carnet.hr/taxonomy/term/26>

[3] <https://sysportal.carnet.hr/taxonomy/term/244>

[4] <https://sysportal.carnet.hr/taxonomy/term/245>

