

[Edit](#) [New issue](#)[Jump to bottom](#)

badblocks returns wrong diagnostics #185

[Open](#) ydemur opened this issue last month · 0 comments

ydemur commented last month • edited ▾

Description of problem:

While checking by badblocks in write mode a USB key corrupted with bad blocks, the numbers returned are not the offsets of the blocks. While using the same checked zone, changing the -c parameter changes the numbers output.

I checked the source file and found other bugs.

The good news is if badblocks returns (0,0,0) errors (and the media is honest), all the media has been correctly checked and is fully good.

The bad news is if there is any error (I/O or corruption), the program may

- not verify some read block versus the good one written
- not verify some blocks
- reports wrong blocks offsets

in short, the diagnostics are false

... just that ...

I use a "stable badblocks" TF card with a USB-A adapter.

It is supposed 16384256 blocks of 4096 bytes. I found a zone of badblocks starting at offset 7725542

The badblocks program allows to test a portion of a media by using the last_block first_block optional arguments.

Tests examples :

```
[root@a55 misc]# badblocks -w -b 4096 -c 512 -t 0x55 /dev/sdc 7725542 7725542 # only one block, bad = 7725542
7725543
```

-----> The returned number has +1 => this number is out of range

```
[root@a55 misc]# badblocks -w -b 4096 -c 512 -t 0x55 /dev/sdc 7725543 7725532 # 12 blocks, bads = 7725542 7725543
7725554
7725555
```

-----> The returned numbers have +12 (the number of read blocks) ; the numbers are out of range

```
[root@a55 misc]# badblocks -w -b 4096 -c 1 -t 0x55 /dev/sdc 7725543 7725532 # 12 blocks, bads = 7725542 7725543
7725543
7725544
```

-----> The returned numbers have +1

So, changing the -c parameter changes the result, while checking the same zone !

I have found the explanation in the source code. I have found too other bugs which need multiple write or read errors. As these errors are not easy to produce, I have simulated them in the code in order to debug.

I made partially corrected versions to highlight the bugs with the error simulation and the debug prints.

I preserved the possibility of allow to see the differences by only modifying what is necessary and providing intermediate versions.

The proposed version may be cleared of the debug lines.

Version-Release number of selected component (if applicable):

e2fsprogs-1.45.6-6.1.mga8 but I think these bugs have been in the code for a long time

I have called badblocks-1.45.6.c the source version.

I have too used the 1.47.0 updated version ; I reported the differences 1.47.0/1.45.6.c .

Proposed solution:

I have modified the badblocks.c source code and created multiple versions.

The joined badb.zip contains all my production.

[badb.zip](#)

Create a directory named badb, cd into it and unzip badb.zip .
\$ cd /tmp ; mkdir badb ; cd badb ; unzip ../badb.zip

The Readme.txt file continues these explanations.

I wish you a good reading ...

Yves DEMUR -- Thu Apr 25 23:30:35 CEST 2024
yves.demur@m4am.net

If you want to download the complete tarball containing the executable binaries and more, it's here :
<http://yves.demur.free.fr/badb.tgz> (~10MiB)



Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant

