

## Re: Sync option destroys flash!

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-05/2654.html>

---

**From:** Michael H. Warfield (*mhw\_at\_wittsend.com*)

**Date:** 05/13/05

To: Lennart Sorensen <lsorensen@csclub.uwaterloo.ca>

Date: Fri, 13 May 2005 13:53:48 -0400

On Fri, 2005-05-13 at 13:17 -0400, Lennart Sorensen wrote:

- > *Certainly causes lots of unnecessary writes which flash doesn't like.*
- > *Given sync doesn't appear to be the default, whyever would you add sync*
- > *to vfat?*

RedHat does it when they automount USB drives. It's in their "hal" package. Law of unintended consequences, I guess.

- >> *Flash drives have a limited number of write cycles. Many many*
- >> *thousands of write cycles, but limited, none the less. They are also*
- >> *written in blocks which are much larger than the "sector" size report*
- >> *(several K in a physical nand flash block, IRC).*
- >>
- >> *What happens, with the sync option on a VFAT file system, is that the*
- >> *FAT tables are getting pounded and over-written over and over and over*
- >> *again as each and every block/cluster is allocated while a new file is*
- >> *written out. This constant overwriting eventually wears out the first*
- >> *block or two of the flash drive.*
  
- > *All the flash I have used do automatic wear leveling. Maybe I have only*
- > *used high quality flash media (given I am doing work with embedded*
- > *industrial grade gear, that is quite plausible). Of course wear*
- > *leveling doesn't mean you aren't doing way more writes than necessary,*
- > *but it helps spread the load away from the FAT, which would otherwise*
- > *quickly die on most flash cards no matter what system you use for*
- > *writing to it.*

Yeah, I've heard that claimed but a real problem is that nobody can tell which ones do or don't until you've got a crispy chip. I've looked and I haven't been able to find a single reference on any that you might pick up at Best Buy or Fry's. I've never seen one that has it written on it that it has any sort of wear leveling like that. But I guess we don't all buy our flash drives at an industrial supply house (forget Tiger Direct or CDW either - consumer grade). I would bet that most people don't even REALIZE that flash has a limited life and wears out.

## Linux-Kernel: Re: Sync option destroys flash!

- > *700M = 1.4M 512byte sectors. I guess if it actually writes one sector*
- > *at a time and syncs, and it's not a media with good wear leveling, then*
- > *yes that would destroy the sectors holding the FAT. Ouch. Crappy*
- > *media, and bad way to treat it. Unfortunately there is no bug, just*
- > *user error, and potentially badly designed flash media firmware.*

Unfortunately, it's the default under some RedHat stuff (yes, I've filed the bug reports) and the documentation on "mount" is incorrect and it's arguable if it's any value, and there's no warning of the risks. "User error" is a bit off the mark when the user has to take action he has no knowledge of to work around a destructive default he doesn't even know exists (here I'm referring specifically to the need to unmount the automounted flash and then remount it somewhere else without the unsafe options).

User error also boils down to removing drives before they are unmounted or sync'ed. Windows seems to deal with this problem without jacking off on the FAT tables. Why is it that Linux has a procedure that is potentially destructive and degrades performance so badly when MS gets away without it in Windows? It's an MS file system. One of the contributors on the Fedora list remarked that this explained why his USB keys were so slow on Linux compared to Windows.

- >> *On a floppy, this would result in an insane amount of jacking around*
- >> *back and forth between data sectors and the FAT sectors. In addition to*
- >> *taking forever, that would shorten the life of the diskettes and the*
- >> *drive itself, but who cares about floppies any more. On a real hard*
- >> *drive, this will cause "head resonances" as the heads go through*
- >> *constant high speed seeks between the cylinder with the FAT tables and*
- >> *the data cylinders. That can't be good, on a continuous basis, for*
- >> *drive life. But it's really a disaster for flash memory. It's going to*
- >> *cause premature failure in most flash memory, even if it doesn't kill*
- >> *them right off as it did in my case with a 700 Meg file.*
- >>
- >> *Can we go back to ignoring "sync" on FAT and VFAT? I can't see where*
- >> *it does much good. You might corrupt a file system if you unplugged it*
- >> *while dirty but it beats the hell out of physically burning it up and*
- >> *destroying the drive!*
- > *How about you just don't use the sync option with fat when you don't*
- > *mean to use sync? sync does exactly what it should, which just happens*
- > *to not be what you want, so don't use it.*

Already filed bug reports with RedHat. On both HAL and on the documentation for mount.

- >> *If it's decided that the FAT and VFAT file systems MUST obey the sync*
- >> *option then please do something about a special case for the FAT tables!*
- >> *Sync the data if thou must buti... Thou shalt not, must not, whack off*
- >> *on the FAT tables!!!*

## Linux-Kernel: Re: Sync option destroys flash!

> *Then the sync option wouldn't be much use anymore.*

Oh? It's of some use now? Beyond degrading performance to the point that it takes many times longer to write a flash system in Linux than in Windows? Yes, I know, don't use it. What's it there for, then? To protect idiots from removing drives before they're unmounted? Those are the same idiots who will not know what happened when it burns up there drive and they are the least likely to use that option overtly. Most people don't realize that it was being used without their knowledge (much less the risks associated with using it in the first place) and just thought it was a Linux problem that it took many times longer to write USB keys in Linux than in Windows.

> > *Another option would be to only sync the FAT and VFAT file systems upon close of the file being written or upon close of the last file open on the file system (fs not busy) but that might not help in the case of a whole lotta little files...*

> *Again not very useful then.*

Rule number one should be "do no harm". Maybe it should be under some "force" flag or something just warning people not to use it unless they really REALLY know what their doing. I, personally, would never have used it under any circumstances. I was shocked to discover that RedHat was doing this by default. A warning to THEM would have been better than a warning to me, since I already knew not to do stupid shit like that. Not sure who is responsible for that HAL package or what other distros may also be vulnerable to this. I mostly blame RedHat for this problem first plus the mount man pages (which I suspect they replied on) second but I question if there shouldn't be a better way to avoid destroying hardware here.

> > *I'm also going to file a couple of bug reports in bugzilla at RedHat but this seems to be a more fundamental problem than a RedHat specific problem. But, IMHO, they should never be setting that damn sync flag arbitrarily.*

> *No they certainly should not, but it may have something to do with making life easier for kde/gnome desktops and automatic mount/umount of media. Dumb idea still, but that happens sometimes.*

> *Len Sorensen*

Mike

--

Michael H. Warfield		(770) 985-6132		mhw@WittsEnd.com
/\ / =mhw= \ /		(678) 463-0932		<a href="http://www.wittsend.com/mhw/">http://www.wittsend.com/mhw/</a>
NIC whois: MHW9		An optimist believes we live in the best of all		
PGP Key: 0xDF1DD471		possible worlds. A pessimist is sure of it!		

## Linux-Kernel: Re: Sync option destroys flash!

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to [majordomo@vger.kernel.org](mailto:majordomo@vger.kernel.org)

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>

---

- application/pgp-signature attachment: This is a digitally signed message part