



Red Hat Enterprise Linux Release 9.2 Manual Pages on 'sg_read.8' command

\$ man sg_read.8

SG_READ(8) SG3_UTILS SG_READ(8)

NAME

sg_read - read multiple blocks of data, optionally with SCSI READ com?
mands

SYNOPSIS

```
sg_read [blk_sgio=0|1] [bpt=BPT] [bs=BS] [cdbsz=6|10|12|16] count=COUNT  
[dio=0|1] [dpo=0|1] [fua=0|1] if=IFILE [mmap=0|1] [no_dxfer=0|1]  
[odir=0|1] [skip=SKIP] [time=TI] [verbose=VERB] [--help] [--version]
```

DESCRIPTION

Read data from a Linux SCSI generic (sg) device, a block device or a normal file with each read command issued to the same offset or logical block address (lba). This can be used to test (or time) disk caching, SCSI (or some other) transport throughput, and/or SCSI command over? head.

When the COUNT value is positive, then up to BPT blocks are read at a time, until the COUNT is exhausted. Each read operation starts at the same lba which, if SKIP is not given, is the beginning of the file or device.

The COUNT value may be negative when IFILE is a sg device or is a block device with 'blk_sgio=1' set. Alternatively 'bpt=0' may be given. In these cases |COUNT| "zero block" SCSI READ commands are issued. "Zero block" means "do nothing" for SCSI READ 10, 12 and 16 byte commands (but not for the 6 byte variant). In practice "zero block" SCSI READ

commands have low latency and so are one way to measure SCSI command overhead.

Please note: this is a very old utility that uses 32 bit integers for disk LBAs and the count. Hence it will not be able to address beyond 2 Terabytes on a disk with logical blocks that are 512 bytes long. Alternatives are the `sg_dd` and `ddpt` utilities.

OPTIONS

`blk_sgio=0 | 1`

The default action of this utility is to use the Unix `read()` command when the `IFILE` is a block device. In `lk 2.6` many block devices can handle SCSI commands issued via the `SG_IO` ioctl. So when this option is set the `SG_IO` ioctl sends SCSI READ commands to `IFILE` if it is a block device.

`bpt=BPT`

where `BPT` is the maximum number of blocks each read operation fetches. Fewer blocks will be fetched when the remaining `COUNT` is less than `BPT`. The default value for `BPT` is 128. Note that each read operation starts at the same lba (as given by `skip=SKIP` or 0). If `'bpt=0'` then the `COUNT` is interpreted as the number of zero block SCSI READ commands to issue.

`bs=BS` where `BS` is the size (in bytes) of each block read. This must be the block size of the physical device (defaults to 512) if SCSI commands are being issued to `IFILE`.

`cdbsz=6 | 10 | 12 | 16`

size of SCSI READ commands issued on `sg` device names, or block devices if `'blk_sgio=1'` is given. Default is 10 byte SCSI READ `cdbs`.

`count=COUNT`

when `COUNT` is a positive number, read that number of blocks, typically with multiple read operations. When `COUNT` is negative then `|COUNT|` SCSI READ commands are performed requesting zero blocks to be transferred. This option is mandatory.

`dio=0 | 1`

default is 0 which selects indirect IO. Value of 1 attempts direct IO which, if not available, falls back to indirect IO and notes this at completion. This option is only active if IFILE is an sg device. If direct IO is selected and /proc/scsi/sg/allow_dio has the value of 0 then a warning is issued (and indirect IO is performed)

dpo=0 | 1

when set the disable page out (DPO) bit in SCSI READ commands is set. Otherwise the DPO bit is cleared (default).

fua=0 | 1

when set the force unit access (FUA) bit in SCSI READ commands is set. Otherwise the FUA bit is cleared (default).

if=IFILE

read from this IFILE. This argument must be given. If the IFILE is a normal file then it must be seekable (if (COUNT > BPT) or skip=SKIP is given). Hence stdin is not acceptable (and giving "-" as the IFILE argument is reported as an error).

mmap=0 | 1

default is 0 which selects indirect IO. Value of 1 causes memory mapped IO to be performed. Selecting both dio and mmap is an error. This option is only active if IFILE is an sg device.

no_dxfer=0 | 1

when set then DMA transfers from the device are made into kernel buffers but no further (i.e. there is no second copy into the user space). The default value is 0 in which case transfers are made into the user space. When neither mmap nor dio is set then data transfer are copied via kernel buffers (i.e. a double copy). Mainly for testing.

odir=0 | 1

when set opens an IFILE which is a block device with an additional O_DIRECT flag. The default value is 0 (i.e. don't open block devices O_DIRECT).

skip=SKIP

all read operations will start offset by SKIP bs-sized blocks from the start of the input file (or device).

time=TI

When TI is 0 (default) doesn't perform timing. When 1, times transfer and does throughput calculation, starting at the first issued command until completion. When 2, times transfer and does throughput calculation, starting at the second issued command until completion. When 3 times from third command, etc. An average number of commands (SCSI READs or Unix read(s) executed per second is also output.

verbose=VERB

as VERB increases so does the amount of debug output sent to stderr. Default value is zero which yields the minimum amount of debug output. A value of 1 reports extra information that is not repetitive.

--help Output the usage message then exit.

--version

Output the version string then exit.

NOTES

Various numeric arguments (e.g. SKIP) may include multiplicative suffixes or be given in hexadecimal. See the "NUMERIC ARGUMENTS" section in the sg3_utils(8) man page.

Data usually gets to the user space in a 2 stage process: first the SCSI adapter DMA's into kernel buffers and then the sg driver copies this data into user memory. This is called "indirect IO" and there is a "dio" option to select "direct IO" which will DMA directly into user memory. Due to some issues "direct IO" is disabled in the sg driver and needs a configuration change to activate it. This is typically done with "echo 1 > /proc/scsi/sg/allow_dio". An alternate way to avoid the 2 stage copy is to select memory mapped IO with 'mmap=1'.

SIGNALS

The signal handling has been borrowed from dd: SIGINT, SIGQUIT and SIG?

PIPE output the number of remaining blocks to be transferred; then they

have their default action. SIGUSR1 causes the same information to be output yet the copy continues. All output caused by signals is sent to stderr.

EXAMPLES

Let us assume that /dev/sg0 is a disk and we wish to time the disk's cache performance.

```
sg_read if=/dev/sg0 bs=512 count=1MB mmap=1 time=2
```

This command will continually read 128 512 byte blocks from block 0.

The "128" is the default value for 'bpt' while "block 0" is chosen because the 'skip' argument was not given. This will continue until

1,000,000 blocks are read. The idea behind using 'time=2' is that the first 64 KiB read operation will involve reading the magnetic media while the remaining read operations will "hit" the disk's cache. The output of third command will look like this:

```
time from second command to end was 4.50 secs, 113.70 MB/sec
```

```
Average number of READ commands per second was 1735.27
```

```
1000000+0 records in, SCSI commands issued: 7813
```

EXIT STATUS

The exit status of sg_read is 0 when it is successful. Otherwise see the sg3_utils(8) man page.

AUTHORS

Written by Douglas Gilbert.

REPORTING BUGS

Report bugs to <dgilbert at interlog dot com>.

COPYRIGHT

Copyright ? 2000-2019 Douglas Gilbert

This software is distributed under the GPL version 2. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

SEE ALSO

To time streaming media read or write time see sg_dd is in the sg3_utils package and ddpt in a package of the same name. The lmbench package contains lmddd which is also interesting. raw(8), dd(1)

