



---

## Blktrace



A quality version of this page, approved on 4 November 2019, was based off this revision.

## Contents

1 Article purpose .....	3
2 Introduction .....	4
3 Installing the trace and debug tool on your target board .....	5
3.1 Using the STM32MPU Embedded Software distribution .....	5
3.1.1 Starter Package .....	5
3.1.2 Developer Package .....	5
3.1.3 Distribution Package .....	5
3.2 Using the STM32MPU Embedded software for Android™ .....	6
3.2.1 Starter Package .....	6
3.2.2 Developer Package .....	6
3.2.3 Distribution Package .....	6
4 Getting started .....	7
5 To go further .....	9
5.1 blkparse usage .....	9
5.2 ftrace usage .....	12
6 References .....	16



---

## 1 Article purpose

---

This article provides the basic information needed to start using the Linux<sup>®</sup> tool: **blktrace**<sup>[1]</sup> (*block tracer*).



## 2 Introduction

The following table provides a brief description of the tool, as well as its availability depending on the software packages:

✓: this tool is either present (ready to use or to be activated), or can be integrated and activated on the software package.

✗: this tool is not present and cannot be integrated, or it is present but cannot be activated on the software package.

Tool			STM32MPU Embedded Software distribution			STM32MPU Embedded Software distribution for Android™		
Name	Category	Purpose	Starter Package	Developer Package	Distribution Package	Starter Package	Developer Package	Distribution Package
blktrace	Tracing tools	blktrace <sup>[1]</sup> generates traces of the I/O traffic on block devices (SD card, USB, eMMC...)	✗	✓	✓	✗	✗	✓

The blktrace tool is composed of three main components:

- a kernel
- a utility to record the I/O trace information from the kernel on the user space
- utilities to analyze and view the trace information.

blkparse<sup>[2]</sup> is also associated to blktrace. It takes the raw output from the blktrace utility and converts it to a nicely formatted and easy-to-read form.



## 3 Installing the trace and debug tool on your target board

It consists of placing the **blktrace** binary in the rootfs, and modifying the Linux kernel configuration.

In order to use **blktrace**, the Linux kernel configuration must activate `CONFIG_BLK_DEV_IO_TRACE` using the Linux kernel menuconfig tool:

```
Symbol: BLK_DEV_IO_TRACE
Location:
  Kernel Hacking --->
    [*] Tracers --->
      [*] Support for tracing block I/O actions
```

### 3.1 Using the STM32MPU Embedded Software distribution

The **blktrace** binary is available by default in all STM32MPU Embedded Software Packages:

```
Board $> which blktrace
/usr/bin/blktrace
```

It is integrated in weston image distribution through openembedded-core package: `openembedded-core/meta/recipes-core/packagegroups/packagegroup-core-tools-profile.bb`.

```
RRECOMMENDS_${PN} = "\
  ${PERF} \
  trace-cmd \
  blktrace \
  ${PROFILE_TOOLS_X} \
  ${PROFILE_TOOLS_SYSTEMD} \
  "
```

#### 3.1.1 Starter Package

Not applicable as Linux kernel configuration cannot be updated.

#### 3.1.2 Developer Package

To enable `CONFIG_BLK_DEV_IO_TRACE` in Linux kernel configuration, please refer to the [Menuconfig](#) or [how to configure kernel](#). This article provides instructions for modifying the configuration and recompiling the Linux kernel image in the Developer Package context.

#### 3.1.3 Distribution Package

To enable `CONFIG_BLK_DEV_IO_TRACE` in Linux<sup>®</sup> kernel configuration, please refer to the [Menuconfig](#) or [how to configure kernel](#). This article provides instructions for modifying the configuration and recompiling the Linux kernel image in the Distribution Package context.



## 3.2 Using the STM32MPU Embedded software for Android™

The **blktrace** binary is not available by default in all STM32MPU Embedded Software Packages for Android, and so must be added.

### 3.2.1 Starter Package

Not applicable as Linux kernel configuration cannot be updated.

### 3.2.2 Developer Package

Not applicable as Linux kernel configuration cannot be updated.

### 3.2.3 Distribution Package

To enable **CONFIG\_BLK\_DEV\_IO\_TRACE** in Linux® kernel configuration, please refer to the [How to customize kernel for Android](#). This article provides instructions for modifying the configuration and recompiling the Linux kernel image in the Distribution Package for Android context.

**blktrace** source code module is available Distribution Package in *external/blktrace*.

- To compile it (ensure the build environment is correctly set):

```
PC $> cd $ANDROID_BUILD_TOP
PC $> mma blktrace
```

- Check if **blktrace** binary is available in the system image:

```
PC $> ls out/target/product/<BoardId>/system/bin/blktrace
```

- Push the binary to the remote target file system:

```
# Remount first the target file system with write access
PC $> adb root; adb remount
PC $> adb sync
```



## 4 Getting started

To start with **blktrace**, trace I/O traffic on a mounted block device (illustrated with the example of the block device `/dev/mmcblk0p6`, which is mounted as `/usr/local`):

```
Board $> mount | grep mmcblk
/dev/mmcblk0p5 on / type ext4 (rw,relatime,data=ordered)
/dev/mmcblk0p4 on /boot type ext4 (rw,relatime,data=ordered)
/dev/mmcblk0p6 on /usr/local type ext4 (rw,relatime,data=ordered)
```

- Launch the **blktrace** tool in background:

```
Board $> blktrace -d /dev/mmcblk0p6 &
```

- Read `/usr/local` content (it generates read access to the associated block device):

```
Board $> ls -la /usr/local
```

- Put **blktrace** activity in foreground and then stop it by pressing **Ctrl+C**:

```
Board $> fg
blktrace -d /dev/mmcblk0p6
Board $> <Ctrl-C>
^C=== mmcblk0p6 ===
CPU 0:          98 events,          5 KiB data
CPU 1:          79 events,          4 KiB data
Total:         177 events (dropped 0), 9 KiB data
```

### Below information is related to the Android™ distribution

ADB must be used as the file system is readonly.



```
PC $> adb root; adb remount
PC $> adb shell
Board $> ...
```

For detailed guidelines on **blktrace** usage, refer to the man page <sup>[1]</sup> or to the help information:

```
Board $> blktrace --help
Usage: blktrace

-d <dev>                | --dev=<dev>
[ -r <debugfs path>    | --relay=<debugfs path> ]
[ -o <file>            | --output=<file>]
[ -D <dir>             | --output-dir=<dir>
[ -w                  | --stopwatch=]
[ -a <action field>   | --act-mask=<action field>]
```



```

[ -A <action mask> | --set-mask=<action mask>]
[ -b <size> | --buffer-size]
[ -n <number> | --num-sub-buffers=<number>]
[ -l | --listen]
[ -h <hostname> | --host=<hostname>]
[ -p <port number> | --port=<port number>]
[ -s | --no-sendfile]
[ -I <devs file> | --input-devs=<devs file>]
[ -v <version> | --version]
[ -V <version> | --version]
-d Use specified device. May also be given last after options
-r Path to mounted debugfs, defaults to /sys/kernel/debug
-o File(s) to send output to
-D Directory to prepend to output file names
-w Stop after defined time, in seconds
-a Only trace specified actions. See documentation
-A Give trace mask as a single value. See documentation
-b Sub buffer size in KiB (default 512)
-n Number of sub buffers (default 4)
-l Run in network listen mode (blktrace server)
-h Run in network client mode, connecting to the given host
-p Network port to use (default 8462)
-s Make the network client NOT use sendfile() to transfer data
-I Add devices found in <devs file>
-v Print program version info
-V Print program version info

```

To get started with **blktrace**, you can also refer to the Yocto Project wiki page for **blktrace**<sup>[3]</sup>, which provides additional details as well as an example.





## 5 To go further

### 5.1 blkparse usage

In addition to displaying information for the user on the console, **blktrace** also creates trace files (one per CPU), which can be viewed using **blkparse**. These files are stored under the current path directory.

```
Board $> blkparse mmcblk0p6
Input file mmcblk0p6.blktrace.0 added
Input file mmcblk0p6.blktrace.1 added
179,6 0 1 0.000000000 99 P N [jbd2/mmcblk0p5-]
179,6 0 0 0.000337792 0 m N cfq99SN insert_request
179,6 0 0 0.000344292 0 m N cfq99SN add_to_rr
179,6 0 0 0.000366250 0 m N cfq99SN insert_request
179,6 0 0 0.000374792 0 m N cfq99SN insert_request
179,6 0 0 0.000382458 0 m N cfq99SN insert_request
179,6 0 0 0.000390167 0 m N cfq99SN insert_request
179,6 0 2 0.000396625 99 U N [jbd2/mmcblk0p5-] 5
179,6 0 0 0.000452542 0 m N cfq workload slice:150000000
179,6 0 0 0.000460750 0 m N cfq99SN set_active_wl_class:0 wl_type:1
179,6 0 0 0.000469083 0 m N cfq99SN dispatch_insert
179,6 0 0 0.000478500 0 m N cfq99SN dispatched a request
179,6 0 0 0.000484583 0 m N cfq99SN activate rq, drv=1
179,6 0 0 0.000747292 0 m N cfq99SN dispatch_insert
179,6 0 0 0.000755042 0 m N cfq99SN dispatched a request
179,6 0 0 0.000759708 0 m N cfq99SN activate rq, drv=2
179,6 0 0 0.004197583 0 m N cfq99SN complete rqnoidle 1
179,6 0 0 0.004206958 0 m N cfq99SN set_slice=120000000
179,6 0 0 0.004231917 0 m N cfq99SN dispatch_insert
179,6 0 0 0.004239208 0 m N cfq99SN dispatched a request
179,6 0 0 0.004243958 0 m N cfq99SN activate rq, drv=2
179,6 0 0 0.006287958 0 m N cfq99SN complete rqnoidle 1
179,6 0 0 0.006305208 0 m N cfq99SN dispatch_insert
179,6 0 0 0.006310542 0 m N cfq99SN dispatched a request
179,6 0 0 0.006315417 0 m N cfq99SN activate rq, drv=2
179,6 0 0 0.008340833 0 m N cfq99SN complete rqnoidle 1
179,6 0 0 0.008359000 0 m N cfq99SN dispatch_insert
179,6 0 0 0.008365083 0 m N cfq99SN dispatched a request
179,6 0 0 0.008370000 0 m N cfq99SN activate rq, drv=2
179,6 0 0 0.011504375 0 m N cfq99SN complete rqnoidle 1
179,6 0 0 0.014671667 0 m N cfq99SN insert_request
179,6 0 0 0.014727458 0 m N cfq99SN complete rqnoidle 1
179,6 0 0 0.014731250 0 m N cfq schedule dispatch
179,6 0 0 0.014808000 0 m N cfq99SN dispatch_insert
179,6 0 0 0.014813875 0 m N cfq99SN dispatched a request
179,6 0 0 0.014819958 0 m N cfq99SN activate rq, drv=1
179,6 0 0 0.017068708 0 m N cfq99SN complete rqnoidle 1
179,6 0 0 0.017078375 0 m N cfq schedule dispatch
179,0 0 3 5.759777419 193 A WS 1706892 + 2 <- (179,6) 32778
179,6 0 4 5.759784878 193 Q WS 1706892 + 2 [jbd2/mmcblk0p6-]
179,6 0 5 5.759849336 193 G WS 1706892 + 2 [jbd2/mmcblk0p6-]
179,6 0 6 5.759854253 193 P N [jbd2/mmcblk0p6-]
179,6 1 1 5.759855044 99 P N [jbd2/mmcblk0p5-]
179,0 0 7 5.759887628 193 A WS 1706894 + 2 <- (179,6) 32780
179,6 0 8 5.759893919 193 Q WS 1706894 + 2 [jbd2/mmcblk0p6-]
179,6 0 9 5.759947753 193 G WS 1706894 + 2 [jbd2/mmcblk0p6-]
179,6 1 0 5.759975503 0 m N cfq99SN insert_request
179,6 1 0 5.759990336 0 m N cfq99SN insert_request
179,6 1 2 5.759999253 99 U N [jbd2/mmcblk0p5-] 2
179,6 0 10 5.760024669 193 I WS 1706892 + 2 [jbd2/mmcblk0p6-]
```



179,6	0	0	5.760035461	0	m	N	cfq193SN	insert_request
179,6	0	0	5.760041711	0	m	N	cfq193SN	add_to_rr
179,6	0	0	5.760053169	0	m	N	cfq193SN	preempt
179,6	0	0	5.760059253	0	m	N	cfq99SN	slice expired t=1
179,6	0	0	5.760066503	0	m	N	cfq99SN	resid=-5635859836
179,6	0	0	5.760081503	0	m	N	cfq99SN	sl_used=120000000 disp=6
charge=120000000 iops=0 sect=12								
179,6	0	11	5.760087294	193	I	WS	1706894 + 2 [jbd2/mmcbk0p6-]	
179,6	0	0	5.760091586	0	m	N	cfq193SN	insert_request
179,6	0	12	5.760096294	193	U	N	[jbd2/mmcbk0p6-]	2
179,6	1	0	5.760111211	0	m	N	cfq	workload slice:300000000
179,6	1	0	5.760118086	0	m	N	cfq193SN	set_active wl_class:0 wl_type:1
179,6	1	0	5.760125669	0	m	N	cfq193SN	dispatch_insert
179,6	1	0	5.760133128	0	m	N	cfq193SN	dispatched a request
179,6	1	0	5.760137711	0	m	N	cfq193SN	activate rq, drv=1
179,6	1	3	5.760145253	80	D	WS	1706892 + 2 [mmcqd/0]	
179,6	1	0	5.760393003	0	m	N	cfq193SN	dispatch_insert
179,6	1	0	5.760399961	0	m	N	cfq193SN	dispatched a request
179,6	1	0	5.760404128	0	m	N	cfq193SN	activate rq, drv=2
179,6	1	4	5.760408211	80	D	WS	1706894 + 2 [mmcqd/0]	
179,6	1	5	5.764367836	80	C	WS	1706892 + 2 [0]	
179,6	1	0	5.764403128	0	m	N	cfq193SN	complete rqnoidle 1
179,6	1	0	5.764412461	0	m	N	cfq193SN	set_slice=120000000
179,6	1	0	5.764439169	0	m	N	cfq193SN	slice expired t=0
179,6	1	0	5.764450253	0	m	N	cfq193SN	sl_used=33750 disp=2
charge=33750 iops=0 sect=4								
179,6	1	0	5.764456836	0	m	N	cfq193SN	del_from_rr
179,6	1	0	5.764464711	0	m	N	cfq99SN	set_active wl_class:0 wl_type:1
179,6	1	0	5.764470211	0	m	N	cfq99SN	dispatch_insert
179,6	1	0	5.764475669	0	m	N	cfq99SN	dispatched a request
179,6	1	0	5.764479378	0	m	N	cfq99SN	activate rq, drv=2
179,6	1	6	5.767516794	80	C	WS	1706894 + 2 [0]	
179,6	1	0	5.767546961	0	m	N	cfq193SN	complete rqnoidle 1
179,6	1	0	5.767564836	0	m	N	cfq99SN	dispatch_insert
179,6	1	0	5.767571086	0	m	N	cfq99SN	dispatched a request
179,6	1	0	5.767575628	0	m	N	cfq99SN	activate rq, drv=2
179,0	0	13	5.767654378	193	A	FWFS	1706896 + 2 <- (179,6) 32782	
179,6	0	14	5.767662669	193	Q	WS	1706896 + 2 [jbd2/mmcbk0p6-]	
179,6	0	15	5.767726461	193	G	WS	1706896 + 2 [jbd2/mmcbk0p6-]	
179,6	0	16	5.767734169	193	I	WS	1706896 + 2 [jbd2/mmcbk0p6-]	
179,6	0	0	5.767744003	0	m	N	cfq193SN	insert_request
179,6	0	0	5.767750253	0	m	N	cfq193SN	add_to_rr
179,6	0	0	5.767760628	0	m	N	cfq193SN	preempt
179,6	0	0	5.767764336	0	m	N	cfq99SN	slice expired t=1
179,6	0	0	5.767770044	0	m	N	cfq99SN	resid=120000000
179,6	0	0	5.767781378	0	m	N	cfq99SN	sl_used=100000000 disp=2
charge=10000000 iops=0 sect=4								
179,6	0	0	5.767784378	0	m	N	cfq99SN	del_from_rr
179,6	1	0	5.772038836	0	m	N	cfq99SN	complete rqnoidle 1
179,6	1	0	5.772064253	0	m	N	cfq193SN	set_active wl_class:0 wl_type:1
179,6	1	0	5.772070878	0	m	N	cfq193SN	dispatch_insert
179,6	1	0	5.772077044	0	m	N	cfq193SN	dispatched a request
179,6	1	0	5.772081878	0	m	N	cfq193SN	activate rq, drv=2
179,6	1	7	5.772085086	80	D	WS	1706896 + 2 [mmcqd/0]	
179,6	1	0	5.774473336	0	m	N	cfq99SN	insert_request
179,6	1	0	5.774481419	0	m	N	cfq99SN	add_to_rr
179,6	1	0	5.774490461	0	m	N	cfq99SN	preempt
179,6	1	0	5.774496211	0	m	N	cfq193SN	slice expired t=1
179,6	1	0	5.774502586	0	m	N	cfq193SN	resid=120000000
179,6	1	0	5.774515044	0	m	N	cfq193SN	sl_used=100000000 disp=1
charge=10000000 iops=0 sect=2								
179,6	1	0	5.774519128	0	m	N	cfq193SN	del_from_rr
179,6	1	0	5.774558711	0	m	N	cfq99SN	complete rqnoidle 1
179,6	1	0	5.774577211	0	m	N	cfq99SN	set_active wl_class:0 wl_type:1
179,6	1	0	5.774583919	0	m	N	cfq99SN	dispatch_insert
179,6	1	0	5.774590253	0	m	N	cfq99SN	dispatched a request
179,6	1	0	5.774595086	0	m	N	cfq99SN	activate rq, drv=2



```

179,6 1 8 5.777926544 80 C WS 1706896 + 2 [0]
179,6 1 0 5.777956461 0 m N cfq193SN complete rqnoidle 1
179,6 1 0 5.782932794 0 m N cfq99SN complete rqnoidle 1
179,6 1 0 5.782943419 0 m N cfq99SN set_slice=120000000
179,6 1 0 5.782949919 0 m N cfq schedule dispatch
CPU0 (mmcblk0p6):
  Reads Queued: 0, 0KiB Writes Queued: 3, 3KiB
  Read Dispatches: 0, 0KiB Write Dispatches: 0, 0KiB
  Reads Requeued: 0 Writes Requeued: 0
  Reads Completed: 0, 0KiB Writes Completed: 0, 0KiB
  Read Merges: 0, 0KiB Write Merges: 0, 0KiB
  Read depth: 0 Write depth: 2
  IO unplugs: 2 Timer unplugs: 0
CPU1 (mmcblk0p6):
  Reads Queued: 0, 0KiB Writes Queued: 0, 0KiB
  Read Dispatches: 0, 0KiB Write Dispatches: 3, 3KiB
  Reads Requeued: 0 Writes Requeued: 0
  Reads Completed: 0, 0KiB Writes Completed: 3, 3KiB
  Read Merges: 0, 0KiB Write Merges: 0, 0KiB
  Read depth: 0 Write depth: 2
  IO unplugs: 1 Timer unplugs: 0
Total (mmcblk0p6):
  Reads Queued: 0, 0KiB Writes Queued: 3, 3KiB
  Read Dispatches: 0, 0KiB Write Dispatches: 3, 3KiB
  Reads Requeued: 0 Writes Requeued: 0
  Reads Completed: 0, 0KiB Writes Completed: 3, 3KiB
  Read Merges: 0, 0KiB Write Merges: 0, 0KiB
  IO unplugs: 3 Timer unplugs: 0
Throughput (R/W): 0KiB/s / 0KiB/s
Events (mmcblk0p6): 118 entries
Skips: 0 forward (0 - 0.0%)

```

### Below information is related to the Android™ distribution

**blkparse** source code module is available in *external/blktrace*.

- To compile it (ensure the build environment is correctly set):

```

PC $> cd $ANDROID_BUILD_TOP
PC $> mma blkparse

```

- Check **blkparse** binary is available in system image:



```

PC $> ls out/target/product/<BoardId>/system/bin
/blkparse

```

- Push the binary to the remote target file system:

```

# Remount first the target file system with write access
PC $> adb root; adb remount
PC $> adb sync

```



## 5.2 ftrace usage

As soon as the Linux kernel configuration `CONFIG_BLK_DEV_IO_TRACE` is active, the block layer (*blk*) action can be traced by using `ftrace`:

- In this case, only the *blk* tracer may be configured for `ftrace`, as shown by the results of the following command:

```
Board $> mount -t tracefs nodev /sys/kernel/tracing
Board $> cat /sys/kernel/tracing/available_tracers
blk nop
```

- To get more `ftrace` tracers (such as. "function"), additional Linux kernel configuration options must be activated through the Distribution Package, as explained in the `ftrace` article.

By taking previous example on `/usr/local` read content (mount point to `mcbk0p6`):

```
Board $> echo 1 > /sys/block/mmcblk0/mmcblk0p6/trace/enable
Board $> echo blk > /sys/kernel/tracing/current_tracer
Board $> ls -la /usr/local
Board $> cat /sys/kernel/tracing/trace_pipe
```

```
jbd2/mmcblk0p5--97 [001] ...1 100.412943: 179,0 A WS 415902 + 2 <- (179,5)
278652
jbd2/mmcblk0p6--196 [000] ...1 100.412943: 179,0 A WS 1706892 + 2 <- (179,6)
32778
jbd2/mmcblk0p5--97 [001] ...1 100.412974: 179,0 Q WS 415902 + 2 [jbd2/mmcblk0p5-
]
jbd2/mmcblk0p6--196 [000] ...1 100.412987: 179,0 Q WS 1706892 + 2 [jbd2
/mmcblk0p6-]
jbd2/mmcblk0p5--97 [001] ...1 100.413012: 179,0 G WS 415902 + 2 [jbd2/mmcblk0p5-
]
jbd2/mmcblk0p6--196 [000] ...1 100.413012: 179,0 G WS 1706892 + 2 [jbd2
/mmcblk0p6-]
jbd2/mmcblk0p6--196 [000] ...1 100.413018: 179,0 P N [jbd2/mmcblk0p6-]
jbd2/mmcblk0p5--97 [001] ...1 100.413019: 179,0 P N [jbd2/mmcblk0p5-]
jbd2/mmcblk0p5--97 [001] ...1 100.413034: 179,0 A WS 415904 + 2 <- (179,5)
278654
jbd2/mmcblk0p6--196 [000] ...1 100.413034: 179,0 A WS 1706894 + 2 <- (179,6)
32780
jbd2/mmcblk0p5--97 [001] ...1 100.413040: 179,0 Q WS 415904 + 2 [jbd2/mmcblk0p5-
]
jbd2/mmcblk0p6--196 [000] ...1 100.413040: 179,0 Q WS 1706894 + 2 [jbd2
/mmcblk0p6-]
jbd2/mmcblk0p5--97 [001] ...1 100.413054: 179,0 G WS 415904 + 2 [jbd2/mmcblk0p5-
]
jbd2/mmcblk0p6--196 [000] ...1 100.413056: 179,0 G WS 1706894 + 2 [jbd2
/mmcblk0p6-]
jbd2/mmcblk0p6--196 [000] ...2 100.413071: 179,0 I WS 1706892 + 2 [jbd2
/mmcblk0p6-]
jbd2/mmcblk0p6--196 [000] ...1 100.413092: 179,0 m N cfq196SN insert_request
jbd2/mmcblk0p6--196 [000] ...2 100.413104: 179,0 I WS 1706894 + 2 [jbd2
/mmcblk0p6-]
jbd2/mmcblk0p6--196 [000] ...1 100.413111: 179,0 m N cfq196SN insert_request
jbd2/mmcblk0p6--196 [000] ...2 100.413118: 179,0 U N [jbd2/mmcblk0p6-] 2
jbd2/mmcblk0p5--97 [001] ...2 100.413135: 179,0 I WS 415902 + 2 [jbd2/mmcblk0p5-
]
jbd2/mmcblk0p5--97 [001] ...1 100.413144: 179,0 m N cfq97SN insert_request
jbd2/mmcblk0p5--97 [001] ...1 100.413151: 179,0 m N cfq97SN add_to_rr
jbd2/mmcblk0p5--97 [001] ...1 100.413162: 179,0 m N cfq97SN preempt
jbd2/mmcblk0p5--97 [001] ...1 100.413168: 179,0 m N cfq196SN slice expired t=1
```



jbd2/mmcblk0p5--97	[001]	...	1	100.413176:	179,0	m	N	cfq196SN	resid=-32494149849
jbd2/mmcblk0p5--97	[001]	...	1	100.413191:	179,0	m	N	cfq196SN	sl_used=120000000
disp=1 charge=120000000 iops=0 sect=2									
jbd2/mmcblk0p5--97	[001]	...	2	100.413199:	179,0	I	WS	415904	+ 2 [jbd2/mmcblk0p5-
]									
jbd2/mmcblk0p5--97	[001]	...	1	100.413205:	179,0	m	N	cfq97SN	insert_request
jbd2/mmcblk0p5--97	[001]	...	2	100.413211:	179,0	U	N	[jbd2/mmcblk0p5-]	2
200000000	mmcqd/0-80	[000]	...	1	100.413224:	179,0	m	N	cfq workload slice:
wl_class:0 wl_type:1									
request	mmcqd/0-80	[000]	...	1	100.413232:	179,0	m	N	cfq97SN set_active
	mmcqd/0-80	[000]	...	1	100.413241:	179,0	m	N	cfq97SN dispatch_insert
	mmcqd/0-80	[000]	...	1	100.413250:	179,0	m	N	cfq97SN dispatched a
request	mmcqd/0-80	[000]	...	1	100.413256:	179,0	m	N	cfq97SN activate rq, drv=1
	mmcqd/0-80	[000]	...	2	100.413261:	179,0	D	WS	415902 + 2 [mmcqd/0]
	mmcqd/0-80	[000]	...	1	100.413515:	179,0	m	N	cfq97SN dispatch_insert
request	mmcqd/0-80	[000]	...	1	100.413524:	179,0	m	N	cfq97SN dispatched a
	mmcqd/0-80	[000]	...	1	100.413530:	179,0	m	N	cfq97SN activate rq, drv=2
	mmcqd/0-80	[000]	...	2	100.413534:	179,0	D	WS	415904 + 2 [mmcqd/0]
	mmcqd/0-80	[000]	...	1	100.417441:	179,0	C	WS	415902 + 2 [0]
	mmcqd/0-80	[000]	...	1	100.417482:	179,0	m	N	cfq97SN complete rqnoidle 1
	mmcqd/0-80	[000]	...	1	100.417494:	179,0	m	N	cfq97SN set_slice=120000000
	mmcqd/0-80	[000]	...	1	100.417526:	179,0	m	N	cfq97SN slice expired t=0
	mmcqd/0-80	[000]	...	1	100.417539:	179,0	m	N	cfq97SN sl_used=40584
disp=2 charge=40584 iops=0 sect=4									
	mmcqd/0-80	[000]	...	1	100.417545:	179,0	m	N	cfq97SN del_from_rr
	mmcqd/0-80	[000]	...	1	100.417555:	179,0	m	N	cfq196SN set_active
wl_class:0 wl_type:1									
request	mmcqd/0-80	[000]	...	1	100.417561:	179,0	m	N	cfq196SN dispatch_insert
	mmcqd/0-80	[000]	...	1	100.417569:	179,0	m	N	cfq196SN dispatched a
request	mmcqd/0-80	[000]	...	1	100.417575:	179,0	m	N	cfq196SN activate rq, drv=2
	mmcqd/0-80	[000]	...	2	100.417579:	179,0	D	WS	1706892 + 2 [mmcqd/0]
	mmcqd/0-80	[000]	...	1	100.419353:	179,0	C	WS	415904 + 2 [0]
	mmcqd/0-80	[000]	...	1	100.419380:	179,0	m	N	cfq97SN complete rqnoidle 1
	mmcqd/0-80	[000]	...	1	100.419393:	179,0	m	N	cfq196SN dispatch_insert
request	mmcqd/0-80	[000]	...	1	100.419400:	179,0	m	N	cfq196SN dispatched a
	mmcqd/0-80	[000]	...	1	100.419405:	179,0	m	N	cfq196SN activate rq, drv=2
	mmcqd/0-80	[000]	...	2	100.419409:	179,0	D	WS	1706894 + 2 [mmcqd/0]
278656	jbd2/mmcblk0p5--97	[001]	...	1	100.419431:	179,0	A	FWFS	415906 + 2 <- (179,5)
]									
jbd2/mmcblk0p5--97	[001]	...	1	100.419440:	179,0	Q	WS	415906	+ 2 [jbd2/mmcblk0p5-
]									
jbd2/mmcblk0p5--97	[001]	...	1	100.419465:	179,0	G	WS	415906	+ 2 [jbd2/mmcblk0p5-
]									
jbd2/mmcblk0p5--97	[001]	...	2	100.419475:	179,0	I	WS	415906	+ 2 [jbd2/mmcblk0p5-
]									
jbd2/mmcblk0p5--97	[001]	...	1	100.419484:	179,0	m	N	cfq97SN	insert_request
jbd2/mmcblk0p5--97	[001]	...	1	100.419491:	179,0	m	N	cfq97SN	add_to_rr
jbd2/mmcblk0p5--97	[001]	...	1	100.419502:	179,0	m	N	cfq97SN	preempt
jbd2/mmcblk0p5--97	[001]	...	1	100.419507:	179,0	m	N	cfq196SN	slice expired t=1
jbd2/mmcblk0p5--97	[001]	...	1	100.419514:	179,0	m	N	cfq196SN	resid=120000000
jbd2/mmcblk0p5--97	[001]	...	1	100.419526:	179,0	m	N	cfq196SN	sl_used=10000000
disp=2 charge=10000000 iops=0 sect=4									
jbd2/mmcblk0p5--97	[001]	...	1	100.419530:	179,0	m	N	cfq196SN	del_from_rr
	mmcqd/0-80	[000]	...	1	100.424339:	179,0	C	WS	1706892 + 2 [0]
	mmcqd/0-80	[000]	...	1	100.424367:	179,0	m	N	cfq196SN complete rqnoidle
1									
	mmcqd/0-80	[000]	...	1	100.424388:	179,0	m	N	cfq97SN set_active
wl_class:0 wl_type:1									
request	mmcqd/0-80	[000]	...	1	100.424396:	179,0	m	N	cfq97SN dispatch_insert
	mmcqd/0-80	[000]	...	1	100.424404:	179,0	m	N	cfq97SN dispatched a
request	mmcqd/0-80	[000]	...	1	100.424409:	179,0	m	N	cfq97SN activate rq, drv=2





	mmcqd/0-80	[000]	...2	100.424413:	179,0	D	WS 415906 + 2 [mmcqd/0]
	mmcqd/0-80	[000]	...1	100.427349:	179,0	C	WS 1706894 + 2 [0]
	mmcqd/0-80	[000]	...1	100.427382:	179,0	m	N cfq196SN complete rqnoidle
1	jbd2/mmcblk0p6--196	[000]	...1	100.427441:	179,0	A	FWFS 1706896 + 2 <- (179,6)
32782	jbd2/mmcblk0p6--196	[000]	...1	100.427458:	179,0	Q	WS 1706896 + 2 [jbd2
	/mmcblk0p6-]						
	jbd2/mmcblk0p6--196	[000]	...1	100.427482:	179,0	G	WS 1706896 + 2 [jbd2
	/mmcblk0p6-]						
	jbd2/mmcblk0p6--196	[000]	...2	100.427492:	179,0	I	WS 1706896 + 2 [jbd2
	/mmcblk0p6-]						
	jbd2/mmcblk0p6--196	[000]	...1	100.427499:	179,0	m	N cfq196SN insert_request
	jbd2/mmcblk0p6--196	[000]	...1	100.427505:	179,0	m	N cfq196SN add_to_rr
	jbd2/mmcblk0p6--196	[000]	...1	100.427516:	179,0	m	N cfq196SN preempt
	jbd2/mmcblk0p6--196	[000]	...1	100.427521:	179,0	m	N cfq97SN slice expired t=1
	jbd2/mmcblk0p6--196	[000]	...1	100.427528:	179,0	m	N cfq97SN resid=120000000
	jbd2/mmcblk0p6--196	[000]	...1	100.427545:	179,0	m	N cfq97SN sl_used=10000000
disp=1 charge=10000000	iops=0 sect=2						
	jbd2/mmcblk0p6--196	[000]	...1	100.427550:	179,0	m	N cfq97SN del_from_rr
	mmcqd/0-80	[000]	...1	100.427598:	179,0	m	N cfq196SN set_active
wl_class:0 wl_type:1	mmcqd/0-80	[000]	...1	100.427605:	179,0	m	N cfq196SN dispatch_insert
request	mmcqd/0-80	[000]	...1	100.427612:	179,0	m	N cfq196SN dispatched a
	mmcqd/0-80	[000]	...1	100.427618:	179,0	m	N cfq196SN activate rq, drv=2
	mmcqd/0-80	[000]	...2	100.427622:	179,0	D	WS 1706896 + 2 [mmcqd/0]
	mmcqd/0-80	[000]	...1	100.431133:	179,0	C	WS 415906 + 2 [0]
	mmcqd/0-80	[000]	...1	100.431164:	179,0	m	N cfq97SN complete rqnoidle 1
	mmcqd/0-80	[000]	...1	100.434921:	179,0	C	WS 1706896 + 2 [0]
	mmcqd/0-80	[000]	...1	100.434951:	179,0	m	N cfq196SN complete rqnoidle
1	mmcqd/0-80	[000]	...1	100.434962:	179,0	m	N cfq196SN
set_slice=120000000	mmcqd/0-80	[000]	...1	100.434969:	179,0	m	N cfq schedule dispatch
	kworker/u4:1-65	[001]	...1	101.612916:	179,0	A	WM 137922 + 2 <- (179,5) 672
	kworker/u4:1-65	[001]	...1	101.612949:	179,0	Q	WM 137922 + 2 [kworker/u4:1]
	kworker/u4:1-65	[001]	...1	101.613001:	179,0	G	WM 137922 + 2 [kworker/u4:1]
	kworker/u4:1-65	[001]	...1	101.613012:	179,0	P	N [kworker/u4:1]
	kworker/u4:1-65	[001]	...1	101.613071:	179,0	A	WM 1674662 + 2 <- (179,6) 548
	kworker/u4:1-65	[001]	...1	101.613077:	179,0	Q	WM 1674662 + 2 [kworker/u4:1]
	kworker/u4:1-65	[001]	...1	101.613093:	179,0	G	WM 1674662 + 2 [kworker/u4:1]
	kworker/u4:1-65	[001]	...2	101.613119:	179,0	I	WM 137922 + 2 [kworker/u4:1]
	kworker/u4:1-65	[001]	...1	101.613131:	179,0	m	N cfq200A insert_request
	kworker/u4:1-65	[001]	...1	101.613139:	179,0	m	N cfq200A add_to_rr
	kworker/u4:1-65	[001]	...1	101.613152:	179,0	m	N cfq200A preempt
	kworker/u4:1-65	[001]	...1	101.613157:	179,0	m	N cfq196SN slice expired t=1
	kworker/u4:1-65	[001]	...1	101.613164:	179,0	m	N cfq196SN resid=-1058203000
	kworker/u4:1-65	[001]	...1	101.613179:	179,0	m	N cfq196SN sl_used=120000000
disp=1 charge=120000000	iops=0 sect=2						
	kworker/u4:1-65	[001]	...1	101.613185:	179,0	m	N cfq196SN del_from_rr
	kworker/u4:1-65	[001]	...2	101.613198:	179,0	I	WM 1674662 + 2 [kworker/u4:1]
	kworker/u4:1-65	[001]	...1	101.613206:	179,0	m	N cfq200A insert_request
	kworker/u4:1-65	[001]	...2	101.613214:	179,0	U	N [kworker/u4:1] 2
	mmcqd/0-80	[000]	...1	101.613238:	179,0	m	N cfq workload slice:40000000
	mmcqd/0-80	[000]	...1	101.613248:	179,0	m	N cfq200A set_active
wl_class:0 wl_type:0	mmcqd/0-80	[000]	...1	101.613258:	179,0	m	N cfq200A dispatch_insert
request	mmcqd/0-80	[000]	...1	101.613267:	179,0	m	N cfq200A dispatched a
	mmcqd/0-80	[000]	...1	101.613274:	179,0	m	N cfq200A activate rq, drv=1
	mmcqd/0-80	[000]	...2	101.613280:	179,0	D	WM 1674662 + 2 [mmcqd/0]
	mmcqd/0-80	[000]	...1	101.613511:	179,0	m	N cfq200A dispatch_insert
request	mmcqd/0-80	[000]	...1	101.613520:	179,0	m	N cfq200A dispatched a
	mmcqd/0-80	[000]	...1	101.613525:	179,0	m	N cfq200A activate rq, drv=2
	mmcqd/0-80	[000]	...2	101.613529:	179,0	D	WM 137922 + 2 [mmcqd/0]



```

0      mmcqd/0-80 [000] ...1 101.618520: 179,0 C WM 1674662 + 2 [0]
      mmcqd/0-80 [000] ...1 101.618576: 179,0 m N cfq200A complete rqnoidle

0      mmcqd/0-80 [000] ...1 101.618586: 179,0 m N cfq200A set_slice=40000000
      mmcqd/0-80 [000] ...1 101.623185: 179,0 C WM 137922 + 2 [0]
      mmcqd/0-80 [000] ...1 101.623215: 179,0 m N cfq200A complete rqnoidle

      mmcqd/0-80 [000] ...1 101.623221: 179,0 m N cfq schedule dispatch

```

**Below information is related to the Android™ distribution**

ADB must be used as the file system is readonly.



```

PC $> adb root; adb remount
PC $> adb shell
Board $> ...

```



## 6 References

- 1.01.11.2 <https://linux.die.net/man/8/blktrace>
- <https://linux.die.net/man/1/blkparse>
- [https://wiki.yoctoproject.org/wiki/Tracing\\_and\\_Profiling#blktrace](https://wiki.yoctoproject.org/wiki/Tracing_and_Profiling#blktrace)

- Useful external links

Document link	Document Type	Description
<a href="#">blktrace tool source code including userspace tool</a>	Sources	blktrace sources git
<a href="#">blktrace Presentation</a>	Presentation	Presentation by Alan D. Brunelle

Linux® is a registered trademark of Linus Torvalds.

SD memory card (<https://www.sdcard.org>)

former spelling for e•MMC ('e' in italic)

input/output

eval,disco (Generic term used, to complete configuration modules paths depending on used board)

Central processing unit

Android debug bridge (Android specific)

Debug File System (See <https://en.wikipedia.org/wiki/Debugfs> for more details)

The Linux Foundation® and Yocto Project® are registered trademarks of the Linux Foundation. Linux® is a registered trademark of Linus Torvalds