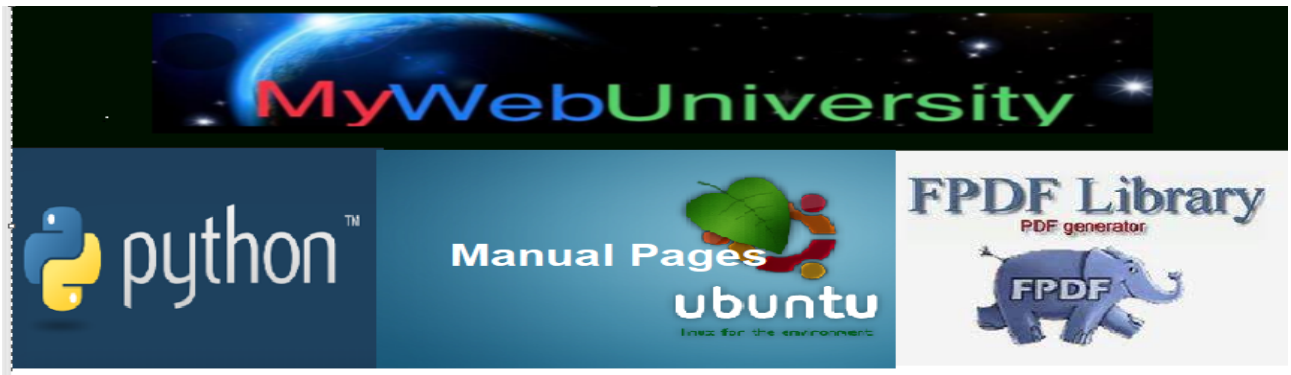

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'lsfd.1'

$ man lsfd.1

LSFD(1) User Commands LSFD(1)

NAME

lsfd - list file descriptors

SYNOPSIS

lsfd [option]

DESCRIPTION

lsfd is intended to be a modern replacement for lsof(8) on Linux systems. Unlike lsof, lsfd is specialized to Linux kernel; it supports Linux specific features like namespaces with simpler code. lsfd is not a drop-in replacement for lsof; they are different in the command line interface and output formats.

The default output is subject to change. So whenever possible, you should avoid using default outputs in your scripts. Always explicitly define expected columns by using --output columns-list in environments where a stable output is required.

lsfd uses Libsmartcols for output formatting and filtering. See the description of --output option for customizing the output format, and --filter option for filtering. Use lsfd --list-columns to get a list of all available columns.

OPTIONS

-l, --threads

List in threads level.

-J, --json

Use JSON output format.

-n, --noheadings

Don?t print headings.

-o, --output list

Specify which output columns to print. See the OUTPUT COLUMNS section for details of available columns.

The default list of columns may be extended if list is specified in the format +list (e.g., lsfd -o +DELETED).

-r, --raw

Use raw output format.

--notruncate

Don?t truncate text in columns.

-p, --pid pids

Collect information only for specified processes. pids is a list of pids. A comma or whitespaces can be used as separators. You can use this option with pidof(1). See FILTER EXAMPLES.

Both -Q option with an expression including PID, e.g. -Q (PID == 1), and -p option, e.g. -p 1, may print the same output but using -p option is much more efficient because -p option works at a much earlier stage of processing than the -Q option.

-i[4|6], --inet[=4|=6]

List only IPv4 sockets and/or IPv6 sockets.

-Q, --filter expr

Print only the files matching the condition represented by the expr. See also scols-filter(5) and FILTER EXAMPLES.

-C, --counter label:filter_expr

Define a custom counter used in --summary output. lsfd makes a counter named label. During collect information, lsfd counts files matching filter_expr, and stores the counted number to the counter named label. lsfd applies filters defined with --filter options before counting; files excluded by the filters are not counted.

See scols-filter(5) about filter_expr. label should not include { nor :. You can define multiple counters by specifying this option multiple times.

See also COUNTER EXAMPLES.

--summary[=when]

This option controls summary lines output. The optional argument when can be only, append or never. If the when argument is omitted, it defaults to only.

The summary reports counters. A counter consists of a label and an integer value.

--counter is the option for defining a counter. If a user defines no counter, lsfd uses the definitions of pre-defined built-in counters (default counters) to make the summary output.

CAUTION: Using --summary and --json may make the output broken. Only combining --summary=only and --json is valid.

--debug-filter

Dump the internal data structure for the filter and exit. This is useful only for lsfd developers.

--dump-counters

Dump the definition of counters used in --summary output.

--hyperlink[=mode]

Print paths as terminal hyperlinks. The mode can be set to "always", "never", or "auto". The optional argument when can be set to "auto", "never", or "always". If the when argument is omitted, it will default to "auto". The "auto" setting means that hyperlinks will only be used if the output is on a terminal.

-H, --list-columns

List available columns that you can specify at --output option.

-h, --help

Display help text and exit.

-V, --version

Display version and exit.

OUTPUT COLUMNS

Each column has a type. Types are surround by < and >.

CAUTION: The names and types of columns are not stable yet. They may be changed in the future releases.

AINODECLASS <string>

Class of anonymous inode.

ASSOC <string>

Association between file and process.

BLKDRV <string>

Block device driver name resolved by /proc/devices.

BPF-MAP.ID <number>

Bpf map ID.

BPF-MAP.TYPE <string>

Decoded name of bpf map type.

BPF-MAP.TYPE.RAW <number>

Bpf map type (raw).

BPF.NAME <string>

Bpf object name.

BPF-PROG.ID <number>

Bpf program ID.

BPF-PROG.TAG <string>

Bpf program TAG.

BPF-PROG.TYPE <string>

Decoded name of bpf program type.

BPF-PROG.TYPE.RAW <number>

Bpf program type (raw).

CHRDRV <string>

Character device driver name resolved by /proc/devices.

COMMAND <string>

Command of the process opening the file.

DELETED <boolean>

Reachability from the file system.

DEV <string>

ID of the device containing the file.

DEVTYPE <string>

Device type (blk, char, or nodev).

ENDPOINTS <string>

IPC endpoints information communicated with the fd.

lsfd collects endpoints within the processes that lsfd scans; lsfd may miss some

endpoints if you limits the processes with -p option.

The format of the column depends on the object associated with the fd:

FIFO type, mqueue type, ptmx and pts sources

PID,COMMAND,ASSOC[-r][-w]

The last characters ([-r][-w]) represent the read and/or write mode of the endpoint.

eventfd type

PID,COMMAND,ASSOC

UNIX-STREAM

PID,COMMAND,ASSOC[-r?][-w?]

About the last characters ([-r?][-w?]), see the description of SOCK.SHUTDOWN.

EVENTFD.ID <number>

Eventfd ID.

EVENTPOLL.TFDS <string>

File descriptors targeted by the eventpoll file.

FD <number>

File descriptor for the file.

FLAGS <string>

Flags specified when opening the file.

FUID <number>

User ID number of the file?s owner.

INET.LADDR <string>

Local IP address.

INET.RADDR <string>

Remote IP address.

INET6.LADDR <string>

Local IP6 address.

INET6.RADDR <string>

Remote IP6 address.

INODE <number>

Inode number.

INOTIFY.INODES <string>

Cooked version of INOTIFY.INODES.RAW. The format of the element is

inode-number,source-of-inode.

INOTIFY.INODES.RAW <string>

List of monitoring inodes. The format of the element is

inode-number,device-major:device-minor.

KNAME <string>

Raw file name extracted from from /proc/pid/fd/fd or /proc/pid/map_files/region.

KTHREAD <boolean>

Whether the process is a kernel thread or not.

MAJ:MIN <string>

Device ID for special, or ID of device containing file.

MAPLEN <number>

Length of file mapping (in page).

MISCDEV <string>

Misc character device name resolved by /proc/misc.

MNTID <number>

Mount ID.

MODE <string>

Access mode (rwx).

NAME <string>

Cooked version of KNAME. It is mostly same as KNAME.

Some files have special formats and information sources:

AF_VSOCK

state=SOCK.STATE type=SOCK.TYPE laddr=VSOCK.LADDR[raddr=VSOCK.RADDR]

raddr is not shown for listening sockets.

bpf-map

id=BPF-MAP.ID type=BPF-MAP.TYPE[name=BPF.NAME]

bpf-prog

id=BPF-PROG.ID type=BPF-PROG.TYPE tag= BPF-PROG.TAG [name=BPF.NAME]

eventpoll

tfds=EVENTPOLL.TFDS

eventfd

id=EVENTFD.ID

inotify

inodes=INOTIFY.INODES

misc:tun

iface=TUN.IFACE

NETLINK

protocol=NETLINK.PROTOCOL[lport=NETLINK.LPORT[group=NETLINK.GROUPS]]

PACKET

type=SOCK.TYPE[protocol=PACKET.PROTOCOL][iface=PACKET.IFACE]

pidfd

pid=TARGET-PID comm=TARGET-COMMAND nspid=TARGET-NSPIDS

lsfd extracts TARGET-PID and TARGET-NSPIDS from /proc/pid/fdinfo/fd.

PING

state=SOCK.STATE[id=PING.ID][laddr=INET.LADDR [raddr=INET.RADDR]]

PINGv6

state=SOCK.STATE[id=PING.ID][laddr=INET6.LADDR [raddr=INET6.RADDR]]

ptmx

tty-index=PTMX.TTY-INDEX

lsfd extracts PTMX.TTY-INDEX from /proc/pid/fdinfo/fd.

RAW

state=SOCK.STATE[protocol=RAW.PROTOCOL [laddr=INET.LADDR [raddr=INET.RADDR]]]

RAWv6

state=SOCK.STATE[protocol=RAW.PROTOCOL [laddr=INET6.LADDR [raddr=INET6.RADDR]]]

signalfd

mask=SIGNALFD.MASK

TCP, TCPv6

state=SOCK.STATE[laddr=TCP.LADDR [raddr=TCP.RADDR]]

timerfd

clockid=TIMERFD.CLOCKID[remaining=TIMERFD.REMAINING [interval=TIMERFD.INTERVAL]]

UDP, UDPv6

state=SOCK.STATE[laddr=UDP.LADDR [raddr=UDP.RADDR]]

lsfd hides raddr= if UDP.RADDR is 0.0.0.0 and UDP.RPORT is 0.

UDP-LITE, UDPLITEv6

state=SOCK.STATE[laddr=UDPLITE.LADDR [raddr=UDPLITE.RADDR]]

UNIX-STREAM

state=SOCK.STATE[path=UNIX.PATH]

UNIX

state=SOCK.STATE[path=UNIX.PATH] type=SOCK.TYPE

Note that (deleted) markers are removed from this column. Refer to KNAME, DELETED, or XMODE to know the readability of the file from the file system.

NETLINK.GROUPS <number>

Netlink multicast groups.

NETLINK.LPORT <number>

Netlink local port id.

NETLINK.PROTOCOL <string>

Netlink protocol.

NLINK <number>

Link count.

NS.NAME <string>

Name (NS.TYPE:[INODE]) of the namespace specified with the file.

NS.TYPE <string>

Type of the namespace specified with the file. The type is mnt, cgroup, uts, ipc, user, pid, net, time, or unknown.

OWNER <string>

Owner of the file.

PACKET.IFACE <string>

Interface name associated with the packet socket.

PACKET.PROTOCOL <string>

L2 protocol associated with the packet socket.

PARTITION <string>

Block device name resolved by /proc/partition.

PID <number>

PID of the process opening the file.

PIDFD.COMM <string>

Command of the process targeted by the pidfd.

PIDFD.NSPID <string>

Value of NSpid field in /proc/pid/fdinfo/fd of the pidfd.

Quoted from kernel/fork.c of Linux source tree:

If pid namespaces are supported then this function will also print the pid of a given pidfd refers to for all descendant pid namespaces starting from the current pid namespace of the instance, i.e. the Pid field and the first entry in the NSpid field will be identical.

Note that this differs from the Pid and NSpid fields in /proc/<pid>/status where Pid and NSpid are always shown relative to the pid namespace of the procfs instance.

PIDFD.PID <number>

PID of the process targeted by the pidfd.

PING.ID <`number`>

ICMP echo request id used on the PING socket.

POS <number>

File position.

RAW.PROTOCOL <number>

Protocol number of the raw socket.

RDEV <string>

Device ID (if special file).

SIGNALFD.MASK <string>

Masked signals.

SIZE <number>

File size.

SOCK.LISTENING <boolean>

Listening socket.

SOCK.NETS <number>

Inode identifying network namespace where the socket belongs to.

SOCK.PROTONAME <string>

Protocol name.

SOCK.SHUTDOWN <string>

Shutdown state of socket.

[-r?]

If the first character is r, the receptions are allowed. If it is -, the receptions are disallowed. If it is ?, the state is unknown.

[-w?]

If the second character is w, the transmissions are allowed. If it is -, the transmissions are disallowed. If it is ?, the state is unknown.

SOCK.STATE <string>

State of socket.

SOCK.TYPE <string>

Type of socket. Here type means the second parameter of socket system call:

? stream

? dgram

? raw

? rdm

? seqpacket

? dccp

? packet

SOURCE <string>

File system, partition, or device containing the file. For the association having ERROR

as the value for TYPE column, lsfd fills this column with syscall:_errno_.

STTYPE <string>

Raw file types returned from stat(2): BLK, CHR, DIR, FIFO, LINK, REG, SOCK, or UNKN.

TCP.LADDR <string>

Local L3 (INET.LADDR or INET6.LADDR) address and local TCP port.

TCP.LPORT <number>

Local TCP port.

TCP.RADDR <string>

Remote L3 (INET.RADDR or INET6.RADDR) address and remote TCP port.

TCP.RPORT <number>

Remote TCP port.

TID <number>

Thread ID of the process opening the file.

TIMERFD.CLOCKID <string>

Clockid.

TIMERFD.INTERVAL <number>

Interval.

TIMERFD.REMAINING <number>

Remaining time.

PTMX.TTY-INDEX <number>

TTY index of the counterpart.

TUN.IFACE <string>

Network interface behind the tun device.

TYPE <string>

Cooked version of STTYPE. It is same as STTYPE with exceptions. For SOCK, print the

value for SOCK.PROTONAME. For UNKN, print the value for AINODECLASS if SOURCE is anon_inodefs.

If lsfd gets an error when calling a syscall to know about a target file descriptor, lsfd fills this column for it with ERROR.

UDP.LADDR <string>

Local IP address and local UDP port.

UDP.LPORT <number>

Local UDP port.

UDP.RADDR <string>

Remote IP address and remote UDP port.

UDP.RPORT <number>

Remote UDP port.

UDPLITE.LADDR <string>

Local IP address and local UDPLite port.

UDPLITE.LPORT <number>

Local UDP port.

UDPLITE.RADDR <string>

Remote IP address and remote UDPLite port.

UDPLITE.RPORT <number>

Remote UDP port.

UID <number>

User ID number.

UNIX.PATH <string>

Filesystem pathname for UNIX domain socket.

USER <string>

User of the process.

VSOCK.LADDR <string>, VSOCK.RADDR <string>

Local VSOCK address. The format of the element is VSOCK.LCID:VSOCK.LPORT. Well-known CIDs will be decoded: ?*?, ?hypervisor?, ?local?, or ?host?. Well-known ports will be decoded: ?*?.

VSOCK.LCID <number>, VSOCK.RCID <number>

Local and remote VSOCK context identifiers.

VSOCK.LPORT <number>, VSOCK.RPORT <number>

Local and remote VSOCK ports.

XMODE <string>

Extended version of MODE. This column may grow; new letters may be appended to XMODE when lsfd supports a new state of file descriptors and/or memory mappings.

[-r]

opened of mapped for reading. This is also in MODE.

[-w]

opened of mapped for writing. This is also in MODE.

[-x]

mapped for executing the code. This is also in MODE.

[-D]

deleted from the file system. See also DELETED.

[-Ll]

locked or leased. l represents a read, a shared lock or a read lease. L represents a write or an exclusive lock or a write lease. If both read/shared and write/exclusive locks or leases are taken by a file descriptor, L is used as the flag.

[-m]

Multiplexed. If the file descriptor is targeted by a eventpoll file or classical system calls for multiplexing (select, pselect, poll, and ppoll), this bit flag is set. Note that if an invocation of the classical system calls is interrupted, lsfd may fail to mark m on the file descriptors monitored by the invocation. See restart_syscall(2).

FILTER EXAMPLES

lsfd has few options for filtering. In most of cases, what you should know is -Q (or --filter) option. Combined with -o (or --output) option, you can customize the output as you want.

List files associated with PID 1 and PID 2 processes:

```
# lsfd -Q '(PID == 1) or (PID == 2)'
```

Do the same in an alternative way:

```
# lsfd -Q '(PID == 1) || (PID == 2)'
```

Do the same in a more efficient way:

```
# lsfd --pid 1,2
```

Whitespaces can be used instead of a comma:

```
# lsfd --pid '1 2'
```

Utilize pidof(1) for list the files associated with "firefox":

```
# lsfd --pid "$(pidof firefox)"
```

List the 1st file descriptor opened by PID 1 process:

```
# lsfd -Q '(PID == 1) and (FD == 1)'
```

Do the same in an alternative way:

```
# lsfd -Q '(PID == 1) && (FD == 1)'
```

List all running executables:

```
# lsfd -Q 'ASSOC == "exe"'
```

Do the same in an alternative way:

```
# lsfd -Q 'ASSOC eq "exe"'
```

Do the same but print only file names:

```
# lsfd -o NAME -Q 'ASSOC eq "exe"' | sort -u
```

List deleted files associated to processes:

```
# lsfd -Q 'DELETED'
```

List non-regular files:

```
# lsfd -Q 'TYPE != "REG"'
```

List block devices:

```
# lsfd -Q 'DEVTYPE == "blk"'
```

Do the same with TYPE column:

```
# lsfd -Q 'TYPE == "BLK"'
```

List files including "dconf" directory in their names:

```
# lsfd -Q 'NAME =~ ".\*/dconf/.*"'
```

List files opened in a QEMU virtual machine:

```
# lsfd -Q '(COMMAND =~ ".\*qemu.*") and (FD >= 0)'
```

List timerfd files expired within 0.5 seconds:

```
# lsfd -Q '(TIMERFD.remaining < 0.5) and (TIMERFD.remaining > 0.0)'
```

List processes communicating via unix stream sockets:

```
# lsfd -Q 'TYPE == "UNIX-STREAM" && UNIX.PATH =~ ".+"'
-oUNIX.PATH,PID,COMMAND,FD,SOCK.STATE,ENDPOINTS
```

List processes communicating via a specified unix stream socket:

```
# lsfd -Q 'TYPE == "UNIX-STREAM" && UNIX.PATH == "@/tmp/.X11-unix/X0"'
-oUNIX.PATH,PID,COMMAND,FD,SOCK.STATE,ENDPOINTS
```

COUNTER EXAMPLES

Report the numbers of netlink socket descriptors and unix socket descriptors:

```
# lsfd --summary=only \
       -C 'netlink sockets':'(NAME =~ "NETLINK:.*")' \
       -C 'unix sockets':'(NAME =~ "UNIX:.*")'
VALUE COUNTER
   57 netlink sockets
 1552 unix sockets
```

Do the same but print in JSON format:

```
# lsfd --summary=only --json \
       -C 'netlink sockets':'(NAME =~ "NETLINK:.*")' \
       -C 'unix sockets':'(NAME =~ "UNIX:.*")'
{
   "lsfd-summary": [
      {
         "value": 15,
         "counter": "netlink sockets"
      },{
         "value": 798,
         "counter": "unix sockets"
      }
   ]
}
```

HISTORY

The lsfd command is part of the util-linux package since v2.38.

AUTHORS

Masatake YAMATO <yamato@redhat.com>, Karel Zak <kzak@redhat.com>

SEE ALSO

bpftool(8), bps(8), lslocks(8), lsof(8), pidof(1), proc(5), scols-filter(5), socket(2), ss(8), stat(2), vsock(7)

REPORTING BUGS

For bug reports, use the issue tracker <https://github.com/util-linux/util-linux/issues>.

AVAILABILITY

The lsfd command is part of the util-linux package which can be downloaded from Linux Kernel Archive <https://www.kernel.org/pub/linux/utils/util-linux/>.