

lsof

Johannes Franken
<jfranken@jfranken.de>

On this page I show example uses of `lsof` - a command line tool for diagnosis of unix systems.

Contents

1. [Overview](#)
2. [Security](#)
3. [Filtering options](#)
4. [Output options](#)
5. [Example uses](#)
 - a) [Rescuing deleted files](#)
 - b) [Verifying software updates](#)
 - c) [Finding the ssh-agent](#)
 - d) [Freeing mountpoints](#)
6. [Links to advanced topics](#)

Overview

`lsof` shows you any open files, directories, unix sockets, ip sockets and pipes. Called with the right options, it can show you

- any files and network connections, which have been opened by a certain process.
- any processes, which have opened a certain file or network connection, or
- the names of any processes, which are waiting for a network connection.

The following documentation correlates to `lsof` version 4.57 from July 19th, 2001.

Security

`lsof` needs root access to gather information about other user's processes. If you want your users to run `lsof`, you can either

- make `lsof` setuid root
- set the parameter `HASSEURITY` at compile-time. This way, any user can run `lsof`, but will see his own processes only.

For Debian 3.0, the option `HASSEURITY` is set.

Filtering options

When root calls `lsdf` without any parameters, it will show everything open by any processes. Even on my notebook, this is a long list of 713 entries:

```
$ lsdf | nl
 1 COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE  NODE NAME
 2 init     1    root  cwd  DIR   3,3    4096   2 /
 3 init     1    root  rtd  DIR   3,3    4096   2 /
 4 init     1    root  txt  REG   3,3   27844 175778 /sbin/init
[... ]
712 lsdf     5873 root  mem  REG   3,3 1153784 160196 /lib/libc-2.2.5.so
713 lsdf     5873 root  4r   FIFO  0,6                31306 pipe
714 lsdf     5873 root  7w   FIFO  0,6                31307 pipe
```

Using the following parameters, you can restrict `lsdf`'s output to the interesting lines. If you pass more than one filtering parameter, `lsdf` will show the lines matching either any or (if you additionally pass `-a`) all criteria.

Parameter	Meaning
File(s)	<p>Show accesses to these files. Example: Who is using vim?</p> <pre>\$ lsdf /usr/bin/vim COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME vim 495 jfranken txt REG 3,3 1102088 175460 /usr/bin/vim vim 1919 jfranken txt REG 3,3 1102088 175460 /usr/bin/vim</pre>
Device(s) or mount-point(s)	<p>Show accesses to these device(s) or mountpoint(s). Example: Who is accessing the CD drive?</p> <pre>\$ lsdf /dev/cdrom COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME bash 3050 jfranken cwd DIR 3,64 2048 53248 /cdrom</pre>
+D Verzeichnis	<p>Show accesses to any files underneath this directory. Example: Who is accessing files in the /tmp directory?</p> <pre>\$ lsdf +D /tmp/ COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME xfs 398 root 3u unix 0xc2019400 993 /tmp/.font-unix/fs7100 XFree86 433 root 1u unix 0xc1da0da0 1111 /tmp/.X11-unix/X0 ssh-agent 477 jfranken 3u unix 0xc1da1aa0 1155 /tmp/ssh-XXWzoq10/agent.452</pre>
+p PIDs	<p>Show anything, what is opened by processes with these PIDs. If you want to specify several PIDs, separate them with commas. Example: What files is my shell using?</p> <pre>\$ lsdf +p 3050 COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME bash 3050 jfranken cwd DIR 3,64 2048 53248 /cdrom bash 3050 jfranken rtd DIR 3,3 4096 2 / bash 3050 jfranken txt REG 3,3 511400 191483 /bin/bash bash 3050 jfranken mem REG 3,3 90210 159620 /lib/ld-2.2.5.so bash 3050 jfranken mem REG 3,3 248132 160128 /lib/libncurses.so.5.2 bash 3050 jfranken mem REG 3,3 8008 160201 /lib/libdl-2.2.5.so bash 3050 jfranken mem REG 3,3 1153784 160196 /lib/libc-2.2.5.so bash 3050 jfranken mem REG 3,3 40152 160223 /lib/libnss_compat-2.2.5.so bash 3050 jfranken mem REG 3,3 69472 160205 /lib/libnsl-2.2.5.so bash 3050 jfranken 0u CHR 136,3 5 /dev/pts/3 bash 3050 jfranken 1u CHR 136,3 5 /dev/pts/3 bash 3050 jfranken 2u CHR 136,3 5 /dev/pts/3 bash 3050 jfranken 255u CHR 136,3 5 /dev/pts/3</pre>

<p>-c name</p>	<p>Show anything, what is opened by processes, whose name starts like this. Example: What files are in use by the init process?</p> <pre> \$ lsof -c init COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME init 1 root cwd DIR 3,3 4096 2 / init 1 root rtd DIR 3,3 4096 2 / init 1 root txt REG 3,3 27844 175778 /sbin/init init 1 root mem REG 3,3 90210 159620 /lib/ld-2.2.5.so init 1 root mem REG 3,3 1153784 160196 /lib/libc-2.2.5.so init 1 root 10u FIFO 3,3 98956 /dev/initctl </pre>
<p>-u user[,user...]</p>	<p>Show anything, what is opened by processes, whose user name matches one of the given names or user-IDs. If you want to specify several users, separate them with commas. If you want to specify any users <i>not</i> matching a certain name or ID, prepend a ^-character to it. Example: What files are being accessed by real users?</p> <pre> \$ lsof -u ^root COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME bash 5712 jfranken cwd DIR 3,3 4096 146941 /home/jfranken bash 5712 jfranken rtd DIR 3,3 4096 2 / bash 5712 jfranken txt REG 3,3 511400 191483 /bin/bash bash 5712 jfranken mem REG 3,3 90210 159620 /lib/ld-2.2.5.so bash 5712 jfranken mem REG 3,3 248132 160128 /lib/libncurses.so.5.2 bash 5712 jfranken mem REG 3,3 8008 160201 /lib/libdl-2.2.5.so bash 5712 jfranken mem REG 3,3 1153784 160196 /lib/libc-2.2.5.so bash 5712 jfranken mem REG 3,3 40152 160223 /lib/libnss_compat-2.2.5.so bash 5712 jfranken mem REG 3,3 69472 160205 /lib/libnsl-2.2.5.so [...]</pre>
<p>-i [TCP UDP][@host][:ports]</p>	<p>Show any network connections matching a given host or port. The host can be named by its hostname or IP address, and the port by its number or service name. To specify more than one port, you can write them as a list (e.g. ssh, www) or range (e.g. 1-1024). Example: What processes are talking on port 80?</p> <pre> \$ lsof -i :80 COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME thttpd 569 root 0u IPv4 2886 TCP *:www (LISTEN) opera 3834 jfranken 20u IPv4 86644 TCP localhost:1055->localhost:www (CLOSE_WAIT) </pre>
<p>+L 1</p>	<p>Deleted files, which are still opened. Example: What open files were deleted?</p> <pre> \$ lsof -a +L1 / COMMAND PID USER FD TYPE DEVICE SIZE NLINK NODE NAME cardmgr 280 root 1u CHR 254,0 0 112863 /var/lib/pcmcia/cm-280-0 (deleted) cardmgr 280 root 2u CHR 254,1 0 112870 /var/lib/pcmcia/cm-280-1 (deleted) </pre>

Output options

Parameter	Meaning
-r seconds	Repeat output. If you pass +r , lsof will quit when the list becomes empty.
-n	Print IP-addresses in place of hostnames.
-l	Print UIDs in place of usernames.
-p	Print port numbers in place of service names.
-t	Print a list of PIDs in place of the table. The result can naturally be processed by command substitutions (backticks etc.). Example: What processes are accessing my home directory? <pre>\$ lsof -t /home/jfranken/ 5711 5754 5780 5781</pre>
-F	Print the table (all rows) in a format, which is particularly eligible for processing in other programs.

Example uses

Rescuing deleted files

If you delete an open file, you have good chance to rescue it from the proc-filesystem:

```
$ rm /sbin/cardmgr
$ rm /etc/wwwoffle/wwwoffle.conf
$ rm /var/log/lpr.log
```

As long as they're still opened, links to their inodes stay in the proc-Filesystem. `lsdf` shows us, what files are involved and where to find them under `/proc`:

```
$ lsdf -a +Ll /
COMMAND  PID USER  FD   TYPE DEVICE  SIZE NLINK  NODE NAME
cardmgr  251 root   txt  REG   3,3 37288    0 175473 /sbin/cardmgr (deleted)
wwwoffled 359 root   3r   REG   3,3 39780    0 53223  /etc/wwwoffle/wwwoffle.conf (deleted)
syslogd  223 root   6w   REG   3,3    0       0 111877 /var/log/lpr.log (deleted)
```

The row `FD` tells us the number of the filedescriptor and the open mode, that is if the file is open for

- reading (`rx*`),
- writing (`nw*` or `nu*`) or
- executing (`txt`, `mem` or `ltx`)

```
$ ls -l /proc/251/exe /proc/359/fd/3 /proc/223/fd/6
lrwxrwxrwx 1 root root 0 Apr 17 14:28 /proc/251/exe -> /sbin/cardmgr (deleted)
lr-x----- 1 root root 64 Apr 17 14:52 /proc/359/fd/3 -> /etc/wwwoffle/wwwoffle.conf (deleted)
l-wx----- 1 root root 64 Apr 17 15:08 /proc/223/fd/6 -> /var/log/lpr.log (deleted)
```

The open mode implicates the method of rescuing the file:

```
$ # restoring executable:
$ cat /proc/251/exe >/sbin/cardmgr
$ chmod +x /sbin/cardmgr
$ ls -l /sbin/cardmgr
-rwxr-xr-x 1 root root 37288 Apr 17 14:24 /sbin/cardmgr
$
$ # restoring readonly:
$ cat /proc/359/fd/3 >/etc/wwwoffle/wwwoffle.conf
$ ls -l /etc/wwwoffle/wwwoffle.conf
-rw-r--r-- 1 root root 39780 Apr 17 14:59 /etc/wwwoffle/wwwoffle.conf
$
$ # restoring writable:
$ nohup tail +0f --pid=223 /proc/223/fd/6 > /var/log/lpr.log &
$ ls -l /var/log/lpr.log
-rw-r--r-- 1 root root 320 Apr 17 15:12 /var/log/lpr.log
```

Rescuing files opened for writing does not always work.

- Sometimes the last second's entries are missing.
- for sequential files (e.g. database tables), you might prefer the following script:

```
while test -e /proc/223/fd/6; do cat /proc/223/fd/6>/tmp/restored ; done
```

Verifying software updates

After updating a program or library, sometimes people miss restarting the processes updated. They stay on their old version. You can identify those processes affected by the fact, that their binaries are deleted, but still open.

Example: What processes should I stop and start after having updated the libc?

```
$ lsof -a +L1 /
COMMAND  PID    USER  FD  TYPE DEVICE  SIZE NLINK  NODE NAME
portmap  143   root  mem  DEL   3,3      0 159620 /lib/ld-2.2.5.so.dpkg-new
portmap  143   root  mem  DEL   3,3      0 160205 /lib/libnsl-2.2.5.so.dpkg-new
portmap  143   root  mem  DEL   3,3      0 160196 /lib/libc-2.2.5.so.dpkg-new
syslogd  223   root  mem  DEL   3,3      0 159620 /lib/ld-2.2.5.so.dpkg-new
syslogd  223   root  mem  DEL   3,3      0 160196 /lib/libc-2.2.5.so.dpkg-new
syslogd  223   root  mem  DEL   3,3      0 160225 /lib/libnss_files-2.2.5.so.dpkg-new
rpc.statd 291   root  mem  DEL   3,3      0 159620 /lib/ld-2.2.5.so.dpkg-new
rpc.statd 291   root  mem  DEL   3,3      0 160205 /lib/libnsl-2.2.5.so.dpkg-new
rpc.statd 291   root  mem  DEL   3,3      0 160196 /lib/libc-2.2.5.so.dpkg-new
rpc.statd 291   root  mem  DEL   3,3      0 160225 /lib/libnss_files-2.2.5.so.dpkg-new
apmd     295   root  mem  DEL   3,3      0 159620 /lib/ld-2.2.5.so.dpkg-new
apmd     295   root  mem  DEL   3,3      0 160196 /lib/libc-2.2.5.so.dpkg-new
inetd   313   root  mem  DEL   3,3      0 159620 /lib/ld-2.2.5.so.dpkg-new
inetd   313   root  mem  DEL   3,3      0 160196 /lib/libc-2.2.5.so.dpkg-new
inetd   313   root  mem  DEL   3,3      0 160225 /lib/libnss_files-2.2.5.so.dpkg-new
[...]
```

After having restarted any processes affected, the list should be empty:

```
$ /etc/init.d/portmap restart
Stopping portmap daemon: portmap.
Starting portmap daemon: portmap.
$ /etc/init.d/syslogd restart
Stopping system log daemon: syslogd.
Starting system log daemon: syslogd.
$ /etc/init.d/nfs-common restart
Stopping NFS common utilities: statd.
Starting NFS common utilities: statd.
$ /etc/init.d/apmd restart
Stopping advanced power management daemon: apmd.
Starting advanced power management daemon: apmd.
$ /etc/init.d/inetd restart
Restarting internet superserver: inetd.
[...]
```

```
$ lsof -a +L1 /
$
```

Finding the ssh-agent

To access a ssh-agent, which is already running, ssh needs two environment variables. With the help of `lsof`, you can easily find out their values:

```
$ /usr/sbin/lsof -a -u jfranken -U -c ssh-agent
COMMAND  PID    USER  FD  TYPE  DEVICE  SIZE NODE NAME
ssh-agent 477   jfranken  3u  unix 0xc1dalaa0 1155 /tmp/ssh-XXWzoql0/agent.452
$ export SSH_AUTH_SOCK=/tmp/ssh-XXWzoql0/agent.452 SSH_AGENT_PID=477
```

Freeing mountpoints

Sometimes you set no great store by the processes accessing just that media, you want to remove for important reasons. In this case, you can have `lsof` feeding the kill command with a list of PIDs:

```
$ umount /dev/cdrom
umount: /cdrom: device is busy
$ kill -9 `lsof -t /dev/cdrom`
$ umount /dev/cdrom
$ eject
```

Links to advanced topics

- The [lsof\(1\) manpage](#)
- The [lsof FAQ](#)
- The [Introduction to lsof](#)
- [big brother.pl \[5 kB\]](#) , a monitoring-script to detect and list new network connections.
- [count pf.pl \[1 kB\]](#) , a script, permanently showing the number of processes, open files, tcp- and udp-connections.