lsof

Johannes Franken
<jfranken@jfranken.de>

Auf dieser Seite zeige ich Anwendungsbeispiele für **lsof**, ein Kommandozeilentool zur Diagnose von Unixsystemen.

Inhalt

Übersicht

lsof zeigt geöffnete Dateien, Verzeichnisse, Unixsockets, IP-Sockets und Pipes an. Mit den passenden Optionen aufgerufen, zeigt es z.B.

- alle Dateien und Netzverbindungen, die ein bestimmter Prozess geöffnet hat,
- alle Prozesse, die eine bestimmte Datei oder Netzverbindung geöffnet haben oder
- die Namen aller Prozesse, die auf eine Netzverbindung warten.

Die folgende Dokumentation bezieht sich auf **lsof** Version 4.57 vom 19. Juli 2001.

Security

Zum Zugriff auf die Prozesse anderer User benötigt **lsof** Rootrechte. Wenn auch die übrigen User mit **lsof** arbeiten sollen, kann man entweder

- **lsof** "setuid root machen" oder
- beim Compilieren den Parameter **HASSECURITY** setzen, wodurch jeder User **lsof** aufrufen kann, aber nur seine eigenen Prozesse sehen wird.

Bei Debian 3.0 ist die HASSECURITY Option gesetzt.

Filter-Optionen

Wenn root **lsof** ohne Parameter aufruft, zeigt es alles, was von sämtlichen Prozessen geöffnet ist. Das ergibt selbst auf meinem Notebook schon 713 Einträge:

```
$ lsof | 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  lsof      5873     root  mem    REG        3,3 1153784    160196 /lib/libc-2.2.5.so
   713  lsof      5873     root    4r  FIFO        0,6              31306 pipe
   714  lsof      5873     root    7w  FIFO        0,6              31307 pipe
```

Mit den folgenden Parametern kann man die Ausgabe auf die interessanten Zeilen eingrenzen. Gibt man mehrere Filterkriterien an, so zeigt **lsof** die Zeilen an, die entweder mindestens eine oder (wenn man zusätzlich **-a** angibt) alle Bedingungen erfüllen.

Parameter	Bedeutung

Datei(en)	Zugriffe auf diese Datei(en) anzeigen.
	Beispiel: Wer benutzt gerade vim?
	`$ lsof /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`

Device(s) oder Mountpoint(s)	Zugriffe auf diese Device(s) oder Mountpoint(s) anzeigen.
	Beispiel: Wer greift auf das CD-Laufwerk zu?
	`$ lsof /dev/cdrom`
	`COMMAND  PID     USER   FD   TYPE DEVICE SIZE  NODE NAME`
	`bash    3050 jfranken  cwd    DIR   3,64 2048 53248 /cdrom`

+D Verzeichnis	Zugriffe auf die Dateien unterhalb des Verzeichnisses anzeigen.
	Beispiel: Wer arbeitet mit Dateien im /tmp-Verzeichnis?
	`$ lsof +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-XXWzoqlO/agent.452`

+p PIDs	Alles anzeigen, was die Prozesse mit diesen PIDs geöffnet haben. Um mehrere PIDs anzugeben, diese mit Komma trennen.
	Beispiel: Welche Dateien benötigt meine Shell?
	`$ lsof +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`

-c Name	Alles anzeigen, was die Prozesse geöffnet haben, deren Namen mit diesem Text beginnen.
	Beispiel: Auf welche Dateien greift der init-Prozess zu?
	`$ 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`

-u user[,user...]	Alles anzeigen, was die Prozesse geöffnet haben, die einem der angegebenen Usernamen oder User-ID gehören. Um mehrere User anzugeben, diese mit Komma trennen. Um die User auszuwählen, deren Name oder UID *nicht* betroffen ist, dem User ein **^**-Zeichen voranstellen.
	Beispiel: Welche Dateien werden zurzeit von echten Usern benutzt?
	`$ 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`
	`[...]`

-i [**TCP**|**UDP**][@host][**:**ports]	Die Netzverbindungen des übergebenen Hosts und Ports anzeigen. Dabei kann der Host wahlweise als Hostname oder IP-Adresse und die Ports als Portnummern oder Servicenamen angegeben werden. Um mehrere Ports zu betrachten, kann man Listen (z.B. **ssh,www**) oder Bereiche (z.B. **1-1024**) angeben.
	Beispiel: Welche Prozesse kommunizieren da auf Port 80 miteinander?
	`$ 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)`

+L 1	Gelöschte Dateien, die noch geöffnet sind.
	Beispiel: Welche Dateien wurden gelöscht, obwohl sie noch offen sind?
	`$ 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)`

Ausgabe-Optionen

Parameter	Bedeutung
-r Sekunden	Die Ausgabe regelmäßig wiederholen. Wenn man **+r** übergibt, beendet sich **lsof**, sobald die Liste leer ist.
-n	IP-Adressen an Stelle von Hostnamen ausgeben.
-l	UIDs an Stelle von Usernamen ausgeben.
-p	Portnummern an Stelle von Servicenamen ausgeben.

-t	Eine PID-Liste an Stelle der Tabelle ausgeben. Das Ergebnis eignet sich natürlich zur Weiterverarbeitung in Command Substitutions (Backticks etc.)
	Beispiel: Welche Prozesse greifen auf mein Homeverzeichnis zu?
	`$ lsof -t /home/jfranken/`
	`5711`
	`5754`
	`5780`
	`5781`

-F	gibt die Tabelle (alle Spalten) in einem Format aus, das besonders gut zum Weiterverarbeiten in anderen Programmen geeignet ist.

Anwendungsbeispiele

Wiederherstellen gelöschter Dateien

Wenn man eine geöffnete Datei löscht, hat man gute Chancen, sie aus dem Proc-Filesystem zu restaurieren:

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

Solange die Dateien geöffnet sind, bleiben Verweise auf ihre Inoden im /proc-Filesystem erhalten. **`lsof`** zeigt uns, welche Dateien betroffen sind und wo genau wir sie unter /proc finden:

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

Der Spalte **`FD`** entnehmen wir die Nummer des Filedescriptors und die Zugriffsart, also ob die Datei zum

- Lesen (n**`r`***),
- Schreiben (n**`w`*** oder n**`u`***) oder
- Ausführen (**`txt`**, **`mem`** oder **`ltx`**)

geöffnet war.

```
$ 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)
```

Von der Zugriffsart hängt die Vorgehensweise bei der Wiederherstellung ab:

```
$ # 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
```

Die Wiederherstellung von Dateien, die zum Schreiben geöffnet sind, gelingt nicht immer:

- bei Logdateien fehlen evtl. die Einträge der letzten Sekunde,
- bei sequentiellen Dateien (z.B. Datenbank-Tabellen) hilft möglicherweise folgende Konstruktion:

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

Softwareupdates prüfen

Beim Update eines Programmes oder einer Library wird gelegentlich vergessen, die aktualisierten Programme auch tatsächlich zu restarten. Sie laufen dann unter der alten Version weiter. Das erkennt man daran, dass die alten Executables zwar gelöscht, aber noch von dem laufenden Prozess geöffnet sind.
Beispiel: Welche Prozesse sollte ich nach dem Update der libc stoppen und starten?

```
$ 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
[...]
```

Nach dem Restarten aller betroffenen Prozesse ist die Liste leer:

```
$ /etc/init.d/portmap restart
Stopping portmap daemon: portmap.
Starting portmap daemon: portmap.
$ /etc/init.d/sysklogd 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 /
$
```

ssh-agent wiederfinden

Damit ssh den gestarteten ssh-agent anspricht, muß man zwei Umgebungstvariablen Werte zuweisen, die man z.B. mit **`lsof`** abfragen kann:

```
$ /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 0xc1da1aa0      1155 /tmp/ssh-XXWzoqlO/agent.452
$ export SSH_AUTH_SOCK=/tmp/ssh-XXWzoqlO/agent.452 SSH_AGENT_PID=477
```

Mountpoints freigeben

Manchmal legt man einfach keinen Wert auf die Prozesse, die auf genau das Medium zugreifen, das man aus wichtigen Gründen gerade aus dem System entfernen möchte. In diesem Fall kann **`lsof`** dem kill-Befehl eine Liste von PIDs liefern:

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

Weiterführende Verweise

- Die lsof(1) manpage
- Die lsof FAQ
- Die Einführung in lsof
- **big_brother.pl** [5 kB] , ein Überwachungs-Script, das neue Netzverbindungen entdeckt und anzeigt.
- **count_pf.pl** [1 kB] , ein Script, das permanent die Zahl der Prozesse, offenen Dateien, tcp- und udp-Verbindungen anzeigt.