# lsof List open files

## CONCEPTS

On Unix everything is a file: regular files, directories, pipes, sockets and devices all show up. Run as root to see other users' processes.

**Selectors** -i -p -u -c names
↓ OR (or AND with -a)
**Matching files** every open file
↓ -t / -F / columns
**Output** table, PIDs, fields

Selection options are ORed by default, so lsof -u me -i lists both. Add -a to require all of them.

Always add -nP when scripting or debugging: it skips slow DNS and service-name lookups.

## FIND BY PROCESS

`lsof -p <pid>`
Open files of a process

`lsof -p <pid1>,<pid2>`
Several processes

`lsof -p ^<pid>`
Everything except a process

`lsof -c <prefix>`
Processes by command-name prefix

`lsof -R`
Add parent PID (PPID) column

`lsof -d cwd -a -p <pid>`
Working directory of a process

## FIND BY USER AND FILE

`lsof -u <user>`
Files opened by a user

`lsof -u ^<user>`
Everyone except a user

`lsof <file>`
Processes holding a file open

`lsof <mountpoint>`
Processes using a filesystem

`lsof +d <dir>`
Open files directly in a dir

`lsof +D <dir>`
Open files in a dir tree

## FIND BY USER AND FILE (cont.)

`lsof -d <fds>`
Select by descriptor (0-2, cwd, mem)

`lsof -a -u <user> -d ^mem`
AND selectors together

## NETWORK

`lsof -i`
All network files

`lsof -i :<port>`
Who uses a port

`lsof -i TCP:<port>`
Restrict to TCP port

`lsof -i :<p1>-<p2>`
Port range

`lsof -i :<p1>,<p2>`
Port list

`lsof -i @<host>`
Connections to a host

`lsof -i <4|6>`
IPv4 or IPv6 only

`lsof -iTCP -sTCP:LISTEN`
Listening TCP sockets

`lsof -iTCP -sTCP:ESTABLISHED`
Established connections

`lsof -iUDP`
UDP sockets

`lsof -nP -i`
No DNS or port-name lookups

`lsof -U`
Unix domain sockets

`lsof -i -a -p <pid>`
Network files of one process

## OUTPUT CONTROL

`lsof -t <selectors>`
PIDs only, no header

`lsof -s`
Show file size

`lsof -F pcn`
Machine-readable fields

`lsof -w`
Suppress warnings

`lsof +c 0`
Full command names

`lsof -r <secs>`
Repeat forever

## OUTPUT CONTROL (cont.)

`lsof +r <secs>`
Repeat until no files match

`lsof +L1`
Unlinked (deleted) but open files

## READING THE TABLE

### Common FD values

| FD | Meaning |
|---|---|
| `cwd` | Current directory |
| `txt` | Program binary |
| `mem` | Memory-mapped file or library |
| `rtd` | Root directory |
| `3u` | Descriptor 3, open read and write |
| `4r / 5w` | Read-only / write-only |
| `DEL` | Mapped file deleted |

### TYPE column

| TYPE | Meaning |
|---|---|
| `REG` | Regular file |
| `DIR` | Directory |
| `CHR` | Character device |
| `FIFO / PIPE` | Pipe |
| `unix` | Unix domain socket |
| `IPv4 / IPv6` | Network socket |

A '(deleted)' suffix on NAME means the file was removed but a process still holds it, so its disk space is not freed.

Without sudo, lsof only shows your own processes and may exit 1 with no output; that does not always mean nothing is open.

## RECIPES

### Free a busy port

```
# find the PID, then stop it
lsof -t -i :8080
kill $(lsof -t -i :8080)
```

### Show what is listening

```
sudo lsof -iTCP -sTCP:LISTEN -nP
# only one command (nginx here)
sudo lsof -nP -i -a -c nginx
```

### Reclaim space of deleted files

```
sudo lsof +L1 | grep -i deleted
# Linux only: empties the file
under a live process, so be sure
sudo truncate -s 0 /proc/<pid>/
fd/<fd>
```

### Watch connections live

```
lsof -nP -i -a -p <pid> -r 2
# unmount blocked?
lsof /mnt/usb
```