

**NAME**

**fstat** - identify active files

**SYNOPSIS**

**fstat** [-**f** *fmnsv*] [-**M** *core*] [-**N** *system*] [-**p** *pid*] [-**u** *user*] [*file ...*]

**DESCRIPTION**

The **fstat** utility identifies open files. A file is considered open by a process if it was explicitly opened, is the working directory, root directory, jail root directory, active executable text, or kernel trace file for that process. If no options are specified, **fstat** reports on all open files in the system for processes the user has access to.

The following options are available:

- f** Restrict examination to files open in the same file systems as the named file arguments, or to the file system containing the current directory if there are no additional filename arguments. For example, to find all files open in the file system where the directory */usr/src* resides, type `'fstat -f /usr/src'`.
- M core** Extract values associated with the name list from the specified core instead of the default */dev/kmem*.
- m** Include memory-mapped files in the listing; normally these are excluded due to the extra processing required.
- N system** Extract the name list from the specified system instead of the default, which is the kernel image the system has booted from.
- n** Numerical format. Print the device number (maj,min) of the file system the file resides in rather than the mount point name; for special files, print the device number that the special device refers to rather than the filename in */dev*; and print the mode of the file in octal instead of symbolic form.
- p pid** Report all files open by the specified process.
- s** Print socket endpoint information.
- u user** Report all files open by the specified user.
- v** Verbose mode. Print error messages upon failures to locate particular system data structures

rather than silently ignoring them. Most of these data structures are dynamically created or deleted and it is possible for them to disappear while **fstat** is running. This is normal and unavoidable since the rest of the system is running while **fstat** itself is running.

*file ...* Restrict reports to the specified files.

The following fields are printed:

**USER** The username of the owner of the process (effective uid).

**CMD** The command name of the process.

**PID** The process id.

**FD** The file number in the per-process open file table or one of the following special names:

**jail** jail root directory

**mmap**

memory-mapped file

**root**

root inode

**text**

executable text inode

**tr** kernel trace file

**wd** current working directory

If the file number is followed by an asterisk (\*), the file is not an inode, but rather a socket, FIFO, or there is an error. In this case the remainder of the line does not correspond to the remaining headers-- the format of the line is described later under *SOCKETS*.

**MOUNT** If the **-n** flag was not specified, this header is present and is the pathname that the file system the file resides in is mounted on.

**DEV** If the **-n** flag is specified, this header is present and is the number of the device that this file resides in.

**INUM** The inode number of the file.

**MODE** The mode of the file. If the **-n** flag is not specified, the mode is printed using a symbolic format (see *strmode(3)*); otherwise, the mode is printed as an octal number.

**SZ|DV** If the file is a semaphore, prints the current value of the semaphore. If the file is not a character or block special, prints the size of the file in bytes. Otherwise, if the **-n** flag is not specified, prints the name of the special file as located in */dev*. If that cannot be located, or the **-n** flag is specified, prints the major/minor device number that the special device refers to.

**R/W** This column describes the access mode that the file allows. The letter ‘r’ indicates open for reading; the letter ‘w’ indicates open for writing. This field is useful when trying to find the processes that are preventing a file system from being down graded to read-only.

**NAME** If filename arguments are specified and the **-f** flag is not, then this field is present and is the name associated with the given file. Normally the name cannot be determined since there is no mapping from an open file back to the directory entry that was used to open that file. Also, since different directory entries may reference the same file (via *ln(1)*), the name printed may not be the actual name that the process originally used to open that file.

## SOCKETS

The formatting of open sockets depends on the protocol domain. In all cases the first field is the domain name, the second field is the socket type (stream, dgram, etc.), and the third is the socket flags field (in hex). The remaining fields are protocol dependent. For TCP, it is the address of the *tcpcb*, and for UDP, the *inpcb* (socket *pcb*). For UNIX-domain sockets, it's the address of the socket *pcb* and the address of the connected *pcb* (if connected). Otherwise the protocol number and address of the socket itself are printed.

For example, the addresses mentioned above are the addresses which the ‘*netstat -A*’ command would print for TCP, UDP, and UNIX-domain. Note that since pipes are implemented using sockets, a pipe appears as a connected UNIX-domain stream socket. A unidirectional UNIX-domain socket indicates the direction of flow with an arrow (‘<-’ or ‘->’), and a full duplex socket shows a double arrow (‘<->’).

When the **-s** flag is used, socket endpoint information is shown after the address of the socket. For internet sockets the local and remote addresses are shown, separated with a double arrow (‘<->’). For UNIX/local sockets either the local or remote address is shown, depending on which one is available.

## EXIT STATUS

The **fstat** utility exits 0 on success, and >0 if an error occurs.

## EXAMPLES

Show all open files except those opened by **fstat** itself:

```
$ fstat | awk '$2 != "fstat"'
```

```
USER  CMD      PID  FD MOUNT  INUM MODE  SZ|DV R/W
```

```
alice bash    469 text /usr/local 143355 -rwxr-xr-x 1166448 r
alice bash    469 cty /dev      346 crw--w---- pts/81 rw
...
```

Report all files opened by the current shell in the same file system as */usr/local* including memory-mapped files:

```
$ fstat -m -p $$ -f /usr/local
USER  CMD      PID  FD MOUNT  INUM MODE  SZ|DV R/W
bob  bash    469 text /usr/local 143355 -rwxr-xr-x 1166448 r
bob  bash    469 mmap /usr/local 143355 -rwxr-xr-x 1166448 r
...
```

Requesting information about a file that is not opened results in just a header line instead of an error:

```
$ fstat /etc/rc.conf
USER  CMD      PID  FD MOUNT  INUM MODE  SZ|DV R/W NAME
```

All parameters after **-f** will be interpreted as files, so the following will not work as expected:

```
$ fstat -f /usr/local -m -p $$
fstat: -m: No such file or directory
fstat: -p: No such file or directory
fstat: 469: No such file or directory
...
```

Show number of pipes opened by firefox processes:

```
$ fstat | awk '$2=="firefox" && $5=="pipe" | wc -l
```

Show processes belonging to user "bob" whose standard error descriptor is opened in ttyv0:

```
$ fstat -u bob | awk '$4 == 2 && $8 == "ttyv0"
bob  firefox  77842  2 /dev    103 crw----- ttyv0 rw
bob  xinit    1194  2 /dev    103 crw----- ttyv0 rw
...
```

Show opened TCP sockets. This output resembles the one produced by `'netstat -A -p tcp'` :

```
$ fstat | awk '$7 == "tcp"
```

```
alice firefox 77991 32* internet stream tcp ffff800b7f147a0
alice firefox 77991 137* internet stream tcp ffff800b7f12b70
...
```

Show a list of processes with files opened in the current directory mimicking the output of `fuser(1)` :

```
$ fstat . | awk 'NR > 1 {printf "%d%s(%s) ", $3, $4, $1;}'
2133wd(alice) 2132wd(alice) 1991wd(alice)
```

Create a list of processes sorted by number of opened files in descending order:

```
$ fstat | awk 'NR > 1 {print $2;}' | sort | uniq -c | sort -r
728 firefox
23 bash
14 sort
8 fstat
7 awk
```

## SEE ALSO

`fuser(1)`, `netstat(1)`, `nfsstat(1)`, `procstat(1)`, `ps(1)`, `sockstat(1)`, `systat(1)`, `tcp(4)`, `unix(4)`, `iostat(8)`, `pstat(8)`, `vmstat(8)`

## HISTORY

The **fstat** command appeared in 4.3BSD-Tahoe.

## BUGS

Since **fstat** takes a snapshot of the system, it is only correct for a very short period of time.