

NAME

stat, **readlink** - display file status

SYNOPSIS

stat [-FHLnq] [-f *format* | -l | -r | -s | -x] [-t *timefmt*] [*file* ...]

readlink [-fn] [*file* ...]

DESCRIPTION

The **stat** utility displays information about the file pointed to by *file*. Read, write, or execute permissions of the named file are not required, but all directories listed in the pathname leading to the file must be searchable. If no argument is given, **stat** displays information about the file descriptor for standard input.

When invoked as **readlink**, only the target of the symbolic link is printed. If the given argument is not a symbolic link and the **-f** option is not specified, **readlink** will print nothing and exit with an error. If the **-f** option is specified, the output is canonicalized by following every symlink in every component of the given path recursively. **readlink** will resolve both absolute and relative paths, and return the absolute pathname corresponding to *file*. In this case, the argument does not need to be a symbolic link.

The information displayed is obtained by calling `lstat(2)` with the given argument and evaluating the returned structure. The default format displays the *st_dev*, *st_ino*, *st_mode*, *st_nlink*, *st_uid*, *st_gid*, *st_rdev*, *st_size*, *st_atime*, *st_mtime*, *st_ctime*, *st_birthtime*, *st_blksize*, *st_blocks*, and *st_flags* fields, in that order.

The options are as follows:

- F** As in `ls(1)`, display a slash ('/') immediately after each pathname that is a directory, an asterisk ('*') after each that is executable, an at sign ('@') after each symbolic link, a percent sign ('%') after each whiteout, an equal sign ('=') after each socket, and a vertical bar ('|') after each that is a FIFO. The use of **-F** implies **-l**.
- H** Treat each argument as the hexadecimal representation of an NFS file handle, and use `fhstat(2)` instead of `lstat(2)`. This requires root privileges.
- L** Use `stat(2)` instead of `lstat(2)`. The information reported by **stat** will refer to the target of *file*, if *file* is a symbolic link, and not to *file* itself. If the link is broken or the target does not exist, fall back on `lstat(2)` and report information about the link.
- n** Do not force a newline to appear at the end of each piece of output.

- q** Suppress failure messages if calls to `stat(2)` or `lstat(2)` fail. When run as **readlink**, error messages are automatically suppressed.
- f format**
Display information using the specified format. See the *Formats* section for a description of valid formats.
- l** Display output in **ls -lT** format.
- r** Display raw information. That is, for all the fields in the *stat* structure, display the raw, numerical value (for example, times in seconds since the epoch, etc.).
- s** Display information in "shell output" format, suitable for initializing variables.
- t timefmt**
Display timestamps using the specified format. This format is passed directly to `strftime(3)`.
- x** Display information in a more verbose way as known from some Linux distributions.

Formats

Format strings are similar to `printf(3)` formats in that they start with `%`, are then followed by a sequence of formatting characters, and end in a character that selects the field of the *struct stat* which is to be formatted. If the `%` is immediately followed by one of **n**, **t**, **%**, or **@**, then a newline character, a tab character, a percent character, or the current file number is printed, otherwise the string is examined for the following:

Any of the following optional flags:

- #** Selects an alternate output form for octal and hexadecimal output. Non-zero octal output will have a leading zero, and non-zero hexadecimal output will have "0x" prepended to it.
- +** Asserts that a sign indicating whether a number is positive or negative should always be printed. Non-negative numbers are not usually printed with a sign.
- Aligns string output to the left of the field, instead of to the right.
- 0** Sets the fill character for left padding to the '0' character, instead of a space.
- space** Reserves a space at the front of non-negative signed output fields. A '+' overrides a space if both are used.

Then the following fields:

- size* An optional decimal digit string specifying the minimum field width.
- prec* An optional precision composed of a decimal point '.' and a decimal digit string that indicates the maximum string length, the number of digits to appear after the decimal point in floating point output, or the minimum number of digits to appear in numeric output.
- fmt* An optional output format specifier which is one of **D**, **O**, **U**, **X**, **F**, or **S**. These represent signed decimal output, octal output, unsigned decimal output, hexadecimal output, floating point output, and string output, respectively. Some output formats do not apply to all fields. Floating point output only applies to *timespec* fields (the **a**, **m**, and **c** fields).

The special output specifier **S** may be used to indicate that the output, if applicable, should be in string format. May be used in combination with:

- amc** Display date in strftime(3) format.
- dr** Display actual device name.
- f** Display the flags of *file* as in **ls -ITdo**.
- gu** Display group or user name.
- p** Display the mode of *file* as in **ls -ITd**.
- N** Displays the name of *file*.
- T** Displays the type of *file*.
- Y** Insert a " -> " into the output. Note that the default output format for **Y** is a string, but if specified explicitly, these four characters are prepended.
- sub* An optional sub field specifier (high, middle, low). Only applies to the **p**, **d**, **r**, and **T** output formats. It can be one of the following:
- H** "High" -- specifies the major number for devices from **r** or **d**, the "user" bits for permissions from the string form of **p**, the file "type" bits from the numeric forms of **p**, and the long output form of **T**.

- L** "Low" -- specifies the minor number for devices from **r** or **d**, the "other" bits for permissions from the string form of **p**, the "user", "group", and "other" bits from the numeric forms of **p**, and the **ls -F** style output character for file type when used with **T** (the use of **L** for this is optional).
- M** "Middle" -- specifies the "group" bits for permissions from the string output form of **p**, or the "suid", "sgid", and "sticky" bits for the numeric forms of **p**.

datum A required field specifier, being one of the following:

- d** Device upon which *file* resides (*st_dev*).
- i** *file*'s inode number (*st_ino*).
- p** File type and permissions (*st_mode*).
- l** Number of hard links to *file* (*st_nlink*).
- u, g** User ID and group ID of *file*'s owner (*st_uid*, *st_gid*).
- r** Device number for character and block device special files (*st_rdev*).
- a, m, c, B**
The time *file* was last accessed or modified, or when the inode was last changed, or the birth time of the inode (*st_atime*, *st_mtime*, *st_ctime*, *st_birthtime*).
- z** The size of *file* in bytes (*st_size*).
- b** Number of blocks allocated for *file* (*st_blocks*).
- k** Optimal file system I/O operation block size (*st_blksize*).
- f** User defined flags for *file*.
- v** Inode generation number (*st_gen*).

The following five field specifiers are not drawn directly from the data in *struct stat*, but are:

- N** The name of the file.

- R** The absolute pathname corresponding to the file.
- T** The file type, either as in **ls -F** or in a more descriptive form if the *sub* field specifier **H** is given.
- Y** The target of a symbolic link.
- Z** Expands to "major,minor" from the *rdev* field for character or block special devices and gives size output for all others.

Only the % and the field specifier are required. Most field specifiers default to **U** as an output form, with the exception of **p** which defaults to **O**; **a**, **m**, and **c** which default to **D**; and **Y**, **T**, and **N** which default to **S**.

EXIT STATUS

The **stat** and **readlink** utilities exit 0 on success, and >0 if an error occurs.

EXAMPLES

If no options are specified, the default format is "%d %i %Sp %l %Su %Sg %r %z \"%Sa\" \"%Sm\" \"%Sc\" \"%SB\" %k %b %#Xf %N".

```
> stat /tmp/bar
0 78852 -rw-r--r-- 1 root wheel 0 0 "Jul 8 10:26:03 2004" "Jul 8 10:26:03 2004" "Jul 8 10:28:13 2004" "Jan 1 09:00 2004" 1148 1015432481
```

Given a symbolic link "foo" that points from */tmp/foo* to */*, you would use **stat** as follows:

```
> stat -F /tmp/foo
lrwxrwxrwx 1 jschauma cs 1 Apr 24 16:37:28 2002 /tmp/foo@ -> /
```

```
> stat -LF /tmp/foo
drwxr-xr-x 16 root wheel 512 Apr 19 10:57:54 2002 /tmp/foo/
```

To initialize some shell variables, you could use the **-s** flag as follows:

```
> csh
% eval set `stat -s .cshrc`
% echo $st_size $st_mtimespec
1148 1015432481

> sh
```

```
$ eval $(stat -s .profile)
$ echo $st_size $st_mtimespec
1148 1015432481
```

In order to get a list of file types including files pointed to if the file is a symbolic link, you could use the following format:

```
$ stat -f "%N: %HT%SY" /tmp/*
/tmp/bar: Symbolic Link -> /tmp/foo
/tmp/output25568: Regular File
/tmp/blah: Directory
/tmp/foo: Symbolic Link -> /
```

In order to get a list of the devices, their types and the major and minor device numbers, formatted with tabs and linebreaks, you could use the following format:

```
stat -f "Name: %N%n%tType: %HT%n%tMajor: %Hr%n%tMinor: %Lr%n%n" /dev/*
[...]
Name: /dev/wt8
    Type: Block Device
    Major: 3
    Minor: 8

Name: /dev/zero
    Type: Character Device
    Major: 2
    Minor: 12
```

In order to determine the permissions set on a file separately, you could use the following format:

```
> stat -f "%Sp -> owner=%SHp group=%SMp other=%SLp" .
drwxr-xr-x -> owner=rwx group=r-x other=r-x
```

In order to determine the three files that have been modified most recently, you could use the following format:

```
> stat -f "%m%t%Sm %N" /tmp/* | sort -rn | head -3 | cut -f2-
Apr 25 11:47:00 2002 /tmp/blah
Apr 25 10:36:34 2002 /tmp/bar
Apr 24 16:47:35 2002 /tmp/foo
```

To display a file's modification time:

```
> stat -f %m /tmp/foo
1177697733
```

To display the same modification time in a readable format:

```
> stat -f %Sm /tmp/foo
Apr 27 11:15:33 2007
```

To display the same modification time in a readable and sortable format:

```
> stat -f %Sm -t %Y%m%d%H%M%S /tmp/foo
20070427111533
```

To display the same in UTC:

```
> sh
$ TZ= stat -f %Sm -t %Y%m%d%H%M%S /tmp/foo
20070427181533
```

SEE ALSO

file(1), ls(1), lstat(2), readlink(2), stat(2), printf(3), strftime(3)

HISTORY

The **stat** utility appeared in NetBSD 1.6 and FreeBSD 4.10.

AUTHORS

The **stat** utility was written by Andrew Brown <atatat@NetBSD.org>. This man page was written by Jan Schaumann <jschauma@NetBSD.org>.