

```
time ^ $$
```

for a seed can fall prey to the mathematical property that

```
a^b == (a+1)^(b+1)
```

one-third of the time. So don't do that.

## **stat FILEHANDLE**

## **stat EXPR**

## **stat**

Returns a 13-element list giving the status info for a file, either the file opened via FILEHANDLE, or named by EXPR. If EXPR is omitted, it stats \$\_. Returns a null list if the stat fails. Typically used as follows:

```
($dev,$ino,$mode,$nlink,$uid,$gid,$rdev,$size,
 $atime,$mtime,$ctime,$blksize,$blocks)
    = stat($filename);
```

Not all fields are supported on all filesystem types. Here are the meanings of the fields:

0 dev	device number of filesystem
1 ino	inode number
2 mode	file mode (type and permissions)
3 nlink	number of (hard) links to the file
4 uid	numeric user ID of file's owner
5 gid	numeric group ID of file's owner
6 rdev	the device identifier (special files only)
7 size	total size of file, in bytes
8 atime	last access time in seconds since the epoch
9 mtime	last modify time in seconds since the epoch
10 ctime	inode change time in seconds since the epoch (*)
11 blksize	preferred block size for file system I/O
12 blocks	actual number of blocks allocated

(The epoch was at 00:00 January 1, 1970 GMT.)

(\*) The ctime field is non-portable. In particular, you cannot expect it to be a "creation time", see *Files and Filesystems in perlport* for details.

If stat is passed the special filehandle consisting of an underline, no stat is done, but the current contents of the stat structure from the last stat, lstat, or filetest are returned. Example:

```
if (-x $file && (($d) = stat(_)) && $d < 0) {
    print "$file is executable NFS file\n";
}
```

(This works on machines only for which the device number is negative under NFS.)

Because the mode contains both the file type and its permissions, you should mask off the file type portion and (s)printf using a "%o" if you want to see the real permissions.

```
$mode = (stat($filename))[2];
printf "Permissions are %04o\n", $mode & 07777;
```

In scalar context, stat returns a boolean value indicating success or failure, and, if successful, sets the information associated with the special filehandle \_.

The File::stat module provides a convenient, by-name access mechanism: