

# How To Find Out a File Type and Permissions in Perl

A few months ago, I've given a really simple example of [using Perl for parsing directory trees in Unix](#). If you looked closer at it, you would have noticed that the script was working fine, but showing file modes as strange large numbers which didn't look like the usual file permissions you would expect. Today I'm going to explain why this happens, and show you how to find out a user type in Perl.

`lstat` and `stat` functions, return, among other things, the file mode value. While it looks confusing initially, it is in fact quite simply a combination field, which includes both the file type and all the permissions for it. If you print this field as a single decimal number, you will not recognize it, but if you simply convert it to octal, you will immediately start seeing the pattern:

Mysterious mode `33261` from the example below becomes `100755` when converted into octal, and you can easily see then the permission part of it: `0755`.

In order to find out a file type in Perl, you need to be able to separate the type part from the permissions part of the file mode value. You can do this manually, but applying the bitmasks, but the easier way is to use the `Fcntl` `:mode` Perl module.

`Fcntl`'`:mode`' module contains all the constants and functions you need to extract the file type and permissions without complicating the task too much.

`S_IMODE` is the function which helps you extract the permissions:

```
$permissions = S_IMODE($mode);
```

and `S_IFMT` is the function which extracts the file type:

```
$filetype = S_IFMT($mode);
```

Now, obviously these extracted values are just the raw numbers, that's why you'll likely want to convert the permissions into octal, and to somehow connect the file type with the common symbolic notation, that's exactly what I'm doing in today's code example.

Here's how it looks (I have the empty lines commented out just so that the code can be correctly shown in WordPress):

```
#!/usr/bin/perl
use Fcntl ':mode';
#
if ($ARGV[0] ne "") {
    $filename = $ARGV[0];
} else {
    print "Please specify a file!\n";
    exit;
}
#
if
(($dev,$ino,$mode,$nlink,$uid,$gid,$rdev,$size,$atime,$mtime,$
ctime,$blksize,$blocks) = lstat($filename)) {
    $user = getpwuid($uid);
    $group = getgrgid($gid);
#
    $ftypes[S_IFDIR] = "d";
    $ftypes[S_IFCHR] = "c";
    $ftypes[S_IFBLK] = "b";
    $ftypes[S_IFREG] = "-";
    $ftypes[S_IFIFO] = "p";
    $ftypes[S_IFLNK] = "l";
    $ftypes[S_IFSOCK] = "s";
#
    $permissions = sprintf "%04o", S_IMODE($mode);
    $filetype = S_IFMT($mode);
#
    $ftype = $ftypes[$filetype];
#
```

```
    print "File: $filename\n";
    printf "File mode: %o (%d)\n", $mode, $mode;
    printf "File type: %s (%o)\n", $ftype, $filetype;
    print "File permissions: $permissions\n";
    print "File size: $size\n";
    print "File owner user: $user\n";
    print "File group: $group\n";
} else {
    print "Please specify an EXISTING file!\n";
}
```

And this is how it works:

```
root@ubuntu:/usr/include# /tmp/stat.pl /tmp/stat.pl
File: /tmp/stat.pl
File mode: 100755 (33261)
File type: - (100000)
File permissions: 0755
File size: 918
File owner user: greys
File group: greys
```

## See also:

- [Directory tree in Perl](#)
- [stat function in Perl](#)