```
    use File::stat;
    $sb = stat($filename);
    printf "File is %s, size is %s, perm %04o, mtime %s\n",
        $filename, $sb->size, $sb->mode & 07777,
        scalar localtime $sb->mtime;
```

You can import symbolic mode constants (S_IF*) and functions (S_IS*) from the Fcntl module:

```
    use Fcntl ':mode';

    $mode = (stat($filename))[2];

    $user_rwx      = ($mode & S_IRWXU) >> 6;
    $group_read    = ($mode & S_IRGRP) >> 3;
    $other_execute =  $mode & S_IXOTH;

    printf "Permissions are %04o\n", S_IMODE($mode), "\n";

    $is_setuid     =  $mode & S_ISUID;
    $is_setgid     =  S_ISDIR($mode);
```

You could write the last two using the -u and -d operators. The commonly available S_IF* constants are

```
    # Permissions: read, write, execute, for user, group, others.

    S_IRWXU S_IRUSR S_IWUSR S_IXUSR
    S_IRWXG S_IRGRP S_IWGRP S_IXGRP
    S_IRWXO S_IROTH S_IWOTH S_IXOTH

    # Setuid/Setgid/Stickiness/SaveText.
    # Note that the exact meaning of these is system dependent.

    S_ISUID S_ISGID S_ISVTX S_ISTXT

    # File types.  Not necessarily all are available on your system.

    S_IFREG S_IFDIR S_IFLNK S_IFBLK S_ISCHR S_IFIFO S_IFSOCK S_IFWHT S_ENFMT

    # The following are compatibility aliases for S_IRUSR, S_IWUSR, S_IXUSR.

    S_IREAD S_IWRITE S_IEXEC
```

and the S_IF* functions are

```
    S_IMODE($mode)      the part of $mode containing the permission bits
                        and the setuid/setgid/sticky bits

    S_IFMT($mode)       the part of $mode containing the file type
                        which can be bit-anded with e.g. S_IFREG
                        or with the following functions

    # The operators -f, -d, -l, -b, -c, -p, and -S.
```