Advanced Programming in the UNIX Environment

Week 03, Segment 3: st_mode

Department of Computer Science
Stevens Institute of Technology

Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/

struct stat: st_mode, st_uid, st_gid

The st_mode also encodes the file access permissions (S_IRUSR, S_IWUSR, S_IXUSR, S_IRGRP, S_IWGRP, S_IXGRP, S_IROTH, S_IWOTH, S_IXOTH). Uses of the permissions are summarized as follows:

- To *open* a file, need *execute* permission on each *directory* component of the path.
- To *open* a file with O_RDONLY or O_RDWR, need read permission.
- To *open* a file with O_WRONLY or O_RDWR, need write permission.
- To use O_TRUNC, must have *write* permission.
- To *create* a new file, must have *write+execute* permission for the *directory.*
- To *delete* a file, need *write+execute* on directory, file *doesn't matter.*
- To *execute* a file (via exec family), need execute permission.

struct stat: st_mode, st_uid, st_gid

Which permission set to use is determined (in order listed):

1. If effective-uid == 0, grant access
2. If effective-uid == st_uid
    2.1. if appropriate user permission bit is set, grant access;
    2.2. else, deny access
3. If effective-gid == st_gid
    3.1.if appropriate group permission bit is set, grant access
    3.2. else, deny access
4. If appropriate other permission bit is set, grant access, else deny access

```
apue$ ls -l /tmp/file
----r-----  1 jschauma  wheel  6 Sep 12 17:20 /tmp/file
apue$ groups
users wheel
apue$ cat /tmp/file
cat: /tmp/file: Permission denied
apue$ chown :users /tmp/file
apue$ ls -l /tmp/file
----r-----  1 jschauma  users  6 Sep 12 17:20 /tmp/file
apue$ cat /tmp/file
cat: /tmp/file: Permission denied
apue$ chmod g-r,o+r /tmp/file
apue$ ls -l /tmp/file
-------r--  1 jschauma  users  6 Sep 12 17:20 /tmp/file
apue$ chown :wheel /tmp/file
apue$ ls -l /tmp/file
-------r--  1 jschauma  wheel  6 Sep 12 17:20 /tmp/file
apue$ su
apue# cat /tmp/file
hello
apue# ^D
apue$ rm /tmp/file
override ------r--  jschauma:wheel for '/tmp/file'? y
apue$
```

`st_mode` and UIDs

We've learned all about permissions and file ownership and how access decisions are made.

We note that the order of checks is fixed and important; as a result, it's possible to create fine-grained access controls through group membership and carefully set file permissions.

Coming up next: using the syscalls that set the file permissions and ownerships. (Yay, more code!)