Operating Systems (CS 3430/5430) Lecture 25 Name:_____

1. Questions about Window's Figures:

a) How many entries are in the ACL for directory "max" in Figure 7.13?

- groups:

- individual users:

b) Clicking the "Advanced" button of 7.13 and "Edit/View" button of Figure 7.14 brings up 13 different permissions. What do some permissions have slashes ("/") in their names?

- Traverse Folder / Execute File
- List Folder / Read Data
- Create Files / Write Data
- Create Folders / Append Data

c) In Figure 7.16 some permissions are neither "Allow" or "Deny". This permits the other ACL entries to decide the access right since a user can be a member of several ACL entries. If all ACL entries "abstain" (neither "Allow" or "Deny"). Should the access be permitted?

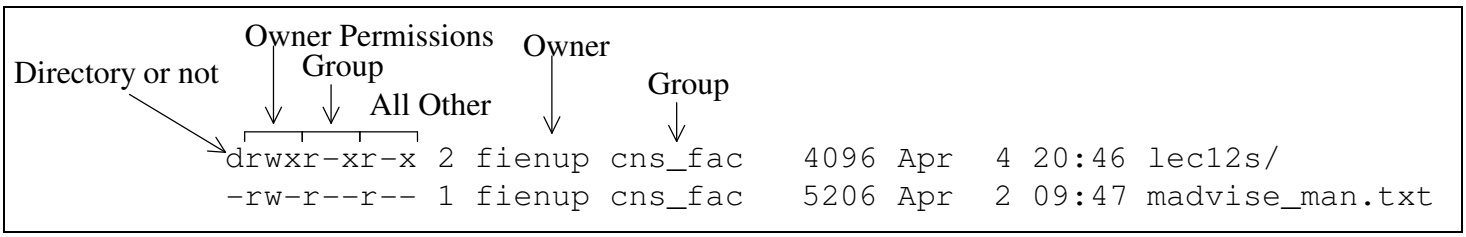
d) If a user belongs to two ACL entries with one is "Allow" and another is "Deny", should the access be allowed?

2. POSIX (Linux/Mac OS X/UNIX) uses a restricted form of ACL for file permissions. The ACL for each file always has three entries:

- owner permissions - a particular user (usually its creator) that owns the file
- owning group - all users who are members of the owning group, except the owner
- all other users who are neither the owner or in the owning group ("world")

a) Can a user be in more than one of the above groups?

b) Permissions for each of the three ACL entries are read (r), write (w), and execute (x). The Linux "`ls -l`" command will list the contents of the current working directory. A partial listing of my `cs3430` directory is:



What permissions does the ower, group, and other have for the directory `lec12s`?

What permissions does the ower, group, and other have for the file `madvise_man.txt`?

c) For a directory, what does it mean to have permission for:

- read (r)?
- write (w)?
- execute (x)?

d) Why are r and x used together on most directories?

e) Each directory has a *sticky bit* that can be set to limit the power of write (w). If set, then a file may be deleted or renamed only by the owner, the owner of the directory, or the system administrator. Why is the sticky bit useful?

3. Some POSIX systems have option of full ACLs. However, in all POSIX systems ACLs are used in conjunction with capabilities. For example, consider a C process opening a file for reading:

```
fileDesciptor = open("myDir/data.txt", "r");
```

a) Normally, the `open` uses the process's user and group credentials to check for necessary permissions. What permissions are needed to allow the file read on:

- current working directory
- subdirectory `myDir`
- file `data.txt`

b) What's different if the process being executed has the `setuid` bit set?

c) What's different if the process being executed has the `setgid` (group ID) bit set?

d) Once permissions are successfully checked, `open` creates a read-only capability for the file and returns an integer file descriptor. While the file is being used, what happens if the owner changes the permissions to deny the access previously granted?