



# *25D Linux Foundation Course*

## 06 – Managing Ownership and Permissions

# Overview



- Managing file ownership
- Mounting file and directory permissions

# How Ownership Works



- When new files or directories are created, they are automatically assigned to the user as the "owner"
- The owner of a file receives read and write permissions by default
- The users group owns the file as well
  - In the below example a file called aryalist was created

```
astark@openSUSE:~> ls -l
total 16
-rw-r--r-- 1 astark housestark  138 Nov  9 10:53 aryalist
drwxr-xr-x 2 astark housestark 4096 Nov  9 10:45 aryastuff
drwxr-xr-x 2 astark housestark 4096 Sep 27  2013 bin
drwxr-xr-x 2 astark housestark 4096 Nov  6  2013 public_html
```

# Managing Ownership from the Command Line



- Using chown
  - used to change the user or group that owns a file or directory
  - chown *user.group file* or *directory*
  - Enter just the user name with the chown command will change just the user; entering a period before a user name will change the group owner

- Using chgrp
  - Can be used to change the group that owns a file or directory
  - chgrp *group file* or *directory*

# Managing Ownership from the Command Line



- In the below example a file named queenslist with an owner of jlannister was changed to an owner of clannister…

```
drwxr-xr-x 2 clannister houselannister 4096 Sep 27  2013 .fonts
-rw-r--r-- 1 clannister houselannister  861 Oct 14  2013 .inputrc
drwx------ 3 clannister houselannister 4096 Nov  9 11:06 .local
-rw-r--r-- 1 clannister houselannister 1028 Sep 27  2013 .profile
-rw------- 1 clannister houselannister  833 Nov  9 10:59 .viminfo
-rw-r--r-- 1 clannister houselannister 1940 Oct 23  2013 .xim.template
-rwxr-xr-x 1 clannister houselannister 1112 Sep 27  2013 .xinitrc.template
drwxr-xr-x 2 clannister houselannister 4096 Sep 27  2013 bin
drwxr-xr-x 2 clannister houselannister 4096 Nov  6  2013 public_html
-rw-r--r-- 1 jlannister houselannister   86 Nov  9 10:57 queenslist
drwxr-xr-x 2 clannister housebaratheon 4096 Nov  9 10:59 queenstuff
openSUSE:/home/clannister # chown clannister /home/clannister/queenslist
openSUSE:/home/clannister # ls -l
total 60
-rw------- 1 clannister houselannister  246 Nov  9 11:15 .bash_history
-rw-r--r-- 1 clannister houselannister 1177 Sep 27  2013 .bashrc
drwx------ 2 clannister houselannister 4096 Sep 27  2013 .config
-rw-r--r-- 1 clannister houselannister 1637 Oct 14  2013 .emacs
drwxr-xr-x 2 clannister houselannister 4096 Sep 27  2013 .fonts
-rw-r--r-- 1 clannister houselannister  861 Oct 14  2013 .inputrc
drwx------ 3 clannister houselannister 4096 Nov  9 11:06 .local
-rw-r--r-- 1 clannister houselannister 1028 Sep 27  2013 .profile
-rw------- 1 clannister houselannister  833 Nov  9 10:59 .viminfo
-rw-r--r-- 1 clannister houselannister 1940 Oct 23  2013 .xim.template
-rwxr-xr-x 1 clannister houselannister 1112 Sep 27  2013 .xinitrc.template
drwxr-xr-x 2 clannister houselannister 4096 Sep 27  2013 bin
drwxr-xr-x 2 clannister houselannister 4096 Nov  6  2013 public_html
-rw-r--r-- 1 clannister houselannister   86 Nov  9 10:57 queenslist
drwxr-xr-x 2 clannister housebaratheon 4096 Nov  9 10:59 queenstuff
openSUSE:/home/clannister # _
```

# Managing Ownership from the Command Line



- In the below example a directory named queenstuff assigned to the group houselannister was changed to another group named housebaratheon…

```
drwxr-xr-x 2 clannister houselannister 4096 Sep 27  2013 .fonts
-rw-r--r-- 1 clannister houselannister  861 Oct 14  2013 .inputrc
drwx------ 3 clannister houselannister 4096 Nov  9 11:06 .local
-rw-r--r-- 1 clannister houselannister 1028 Sep 27  2013 .profile
-rw------- 1 clannister houselannister  833 Nov  9 10:59 .viminfo
-rw-r--r-- 1 clannister houselannister 1940 Oct 23  2013 .xim.template
-rwxr-xr-x 1 clannister houselannister 1112 Sep 27  2013 .xinitrc.template
drwxr-xr-x 2 clannister houselannister 4096 Sep 27  2013 bin
drwxr-xr-x 2 clannister houselannister 4096 Nov  6  2013 public_html
-rw-r--r-- 1 clannister houselannister   86 Nov  9 10:57 queenslist
drwxr-xr-x 2 clannister houselannister 4096 Nov  9 10:59 queenstuff
openSUSE:/home/clannister # chgrp housebaratheon /home/clannister/queenstuff
openSUSE:/home/clannister # ls -l
total 60
-rw------- 1 clannister houselannister  246 Nov  9 11:15 .bash_history
-rw-r--r-- 1 clannister houselannister 1177 Sep 27  2013 .bashrc
drwx------ 2 clannister houselannister 4096 Sep 27  2013 .config
-rw-r--r-- 1 clannister houselannister 1637 Oct 14  2013 .emacs
drwxr-xr-x 2 clannister houselannister 4096 Sep 27  2013 .fonts
-rw-r--r-- 1 clannister houselannister  861 Oct 14  2013 .inputrc
drwx------ 3 clannister houselannister 4096 Nov  9 11:06 .local
-rw-r--r-- 1 clannister houselannister 1028 Sep 27  2013 .profile
-rw------- 1 clannister houselannister  833 Nov  9 10:59 .viminfo
-rw-r--r-- 1 clannister houselannister 1940 Oct 23  2013 .xim.template
-rwxr-xr-x 1 clannister houselannister 1112 Sep 27  2013 .xinitrc.template
drwxr-xr-x 2 clannister houselannister 4096 Sep 27  2013 bin
drwxr-xr-x 2 clannister houselannister 4096 Nov  6  2013 public_html
-rw-r--r-- 1 clannister houselannister   86 Nov  9 10:57 queenslist
drwxr-xr-x 2 clannister housebaratheon 4096 Nov  9 10:59 queenstuff
openSUSE:/home/clannister # _
```

# Exercise 6-1: Managing Ownership



**Please open your Practical Exercise book to Exercise 6-1.**

**Time to Complete: 5 Minutes**

# How Permissions Work



| Permission | Symbol | Value | Effect on Files | Effect on Directories |
|---|---|---|---|---|
| Read | r | 4 | Allows a user to open and view a file. Does not allow a file to be modified or saved. | Allows a user to list the contents of a directory. |
| Write | w | 2 | Allows a user to open, modify, and save a file. | Allows a user to add or delete files from the directory. |
| Execute | x | 1 | Allows a user to run an executable file. | Allows a user to enter a directory. |

- **Owner**
  - This is the user account that has been assigned to be the file or directory's owner. Permissions assigned to the owner apply only to that user account.
- **Group**
  - This is the group that has been assigned ownership of the file or directory. Permissions assigned to the group apply to all user accounts that are members of that group.
- **Others**
  - This entity refers to all other users who have successfully authenticated to the system. Permissions assigned to this entity apply to these user accounts.

# How Permissions Work



```
openSUSE:/home/clannister # ls -l
total 60
-rw------- 1 clannister houselannister  246 Nov  9 11:15 .bash_history
-rw-r--r-- 1 clannister houselannister 1177 Sep 27  2013 .bashrc
drwx------ 2 clannister houselannister 4096 Sep 27  2013 .config
-rw-r--r-- 1 clannister houselannister 1637 Oct 14  2013 .emacs
drwxr-xr-x 2 clannister houselannister 4096 Sep 27  2013 .fonts
-rw-r--r-- 1 clannister houselannister  861 Oct 14  2013 .inputrc
drwx------ 3 clannister houselannister 4096 Nov  9 11:06 .local
-rw-r--r-- 1 clannister houselannister 1028 Sep 27  2013 .profile
-rw------- 1 clannister houselannister  833 Nov  9 10:59 .viminfo
-rw-r--r-- 1 clannister houselannister 1940 Oct 23  2013 .xim.template
-rwxr-xr-x 1 clannister houselannister 1112 Sep 27  2013 .xinitrc.template
drwxr-xr-x 2 clannister houselannister 4096 Sep 27  2013 bin
drwxr-xr-x 2 clannister houselannister 4096 Nov  6  2013 public_html
-rw-r--r-- 1 clannister houselannister   86 Nov  9 10:57 queenslist
drwxr-xr-x 2 clannister housebaratheon 4096 Nov  9 10:59 queenstuff
```

file permissions

# of hard links/directories

file owner

group owner

file size

timestamp

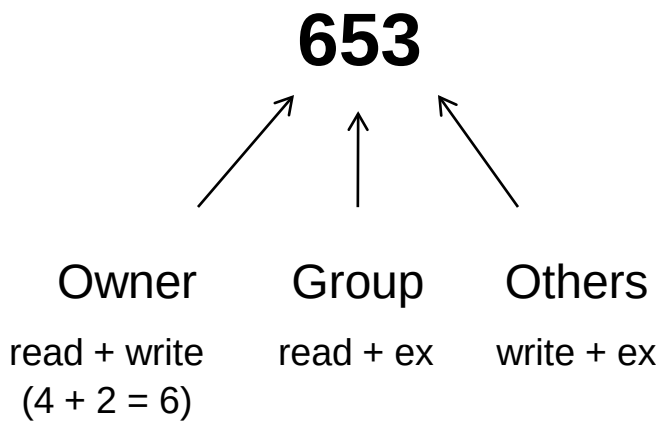
filename

# File Access Rights



## Use `ls -l` to check file access rights

- 3 types of access rights
  - Read access (r) (4)
  - Write access (w) (2)
  - Execute rights (x) (1)




- 3 types of access levels
  - User (u): for the owner of the file
  - Group (g): each file also has a “group” attribute, corresponding to a given list of users
  - Others (o): for all other users

# Access Right Constraints



- Execute is sufficient to execute binaries
  - Both x and r and required for shell scripts
- Both read and execute permissions needed for access to directories:
  - r to list the contents, x to access the contents
- Cannot rename, remove, copy files in a directory if you don't have write access to this directory
- If you have write access to a directory, you CAN:
  - remove a file even if you don't have write access to it
  - modify (remove + recreate) a file even without w access to it.

# Access Rights Examples



- -rw-r--r--
  - Readable and writable for file owner, only readable for group and others
- -rw-r-----
  - Readable and writable for file owner, only readable for users belonging to the file group
- drwx------
  - Directory only accessible by its owner
- -------r-x
  - File executable by others but not by the owner or group members

# chmod: changing permissions



- Syntax: chmod <permissions> <files>
- 2 formats for permissions:
  - Octal format (abc):
    - a,b,c = r*4+w*2+x (r, w, x: booleans)
    - Example: chmod 644 <file> (rw for u, r for g and o)
  - Or symbolic format. Easy to understand by examples:
    - chmod go+r: add read permissions to group and others
    - chmod u-w: remove write permissions from user
    - chmod a-x: (a: all) remove execute permission from all


# chmod: changing permissions




- **Example 1**

  **chmod –R a+rX linux/**

  - **Makes linux and everything in it available to everyone!**
    - **R: apply changes recursively**
    - **X: x, but only for directories and files already executable**

- **Very useful to open recursive access to directories, without adding execution rights to all files.**

# chmod: changing permissions



## Example 2

chmod a+t /tmp

- t: (sticky).
  - Special permission for directories, allowing only the directory and file owner to delete a file in a directory
- Useful for directories with write access to anyone, like /tmp.
- Displayed by `ls -l` with a t character.

# Exercise 6-2: Managing Permissions

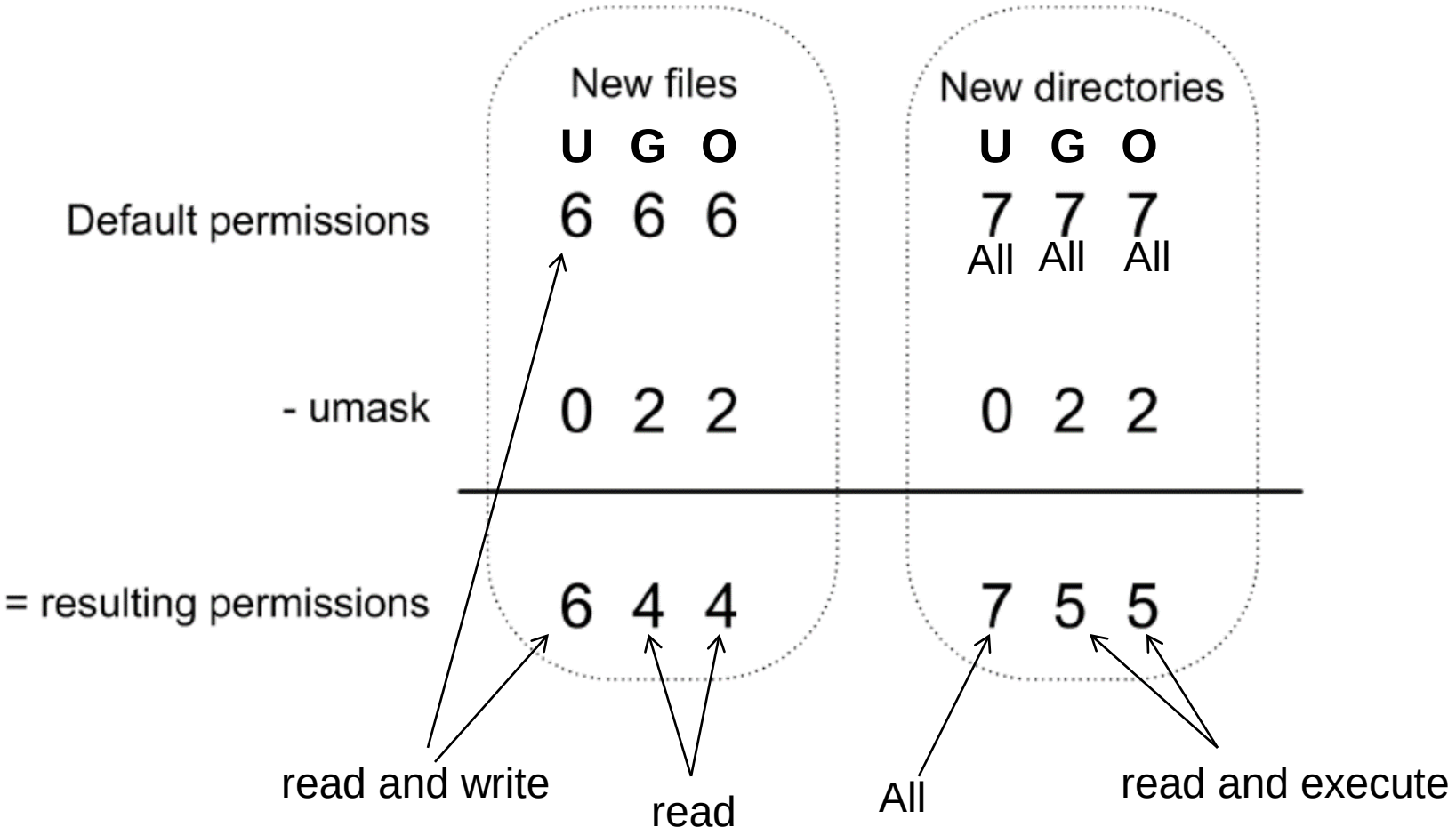


**Please open your Practical Exercise book to Exercise 6-2.**

**Time to Complete: 5 Minutes**

# Working with Default Permissions





# Working with Special Permissions



| Permission | Value | Description | Effect on Files | Effect on Directories |
|---|---|---|---|---|
| SUID | 4 | Set User ID. Can only be applied to binary executable files (not shell scripts). | When an executable file with the SUID set is run, the user who ran the file temporarily becomes the file's owner. | None. |
| SGID | 2 | Set Group ID. Can be applied to binary executable files (not shell scripts). | When a user runs an executable file with SGID set, the user temporarily becomes a member of the file's owning group. | When a user creates a file in a directory that has SGID set, the file's owner is set to the user's account (as per normal). However, the owning group assigned to the new file is set to the owning group of the parent directory. |
| Sticky Bit | 1 | None. | | When the Sticky Bit is assigned to a directory, users can only delete files within the directory for which they are the owner of the file or the directory itself. This negates the effect of having the write permission to a directory, which could allow a user to delete files in a directory that they don't own. |

# Exercise 6-3: Managing Default and Special Permissions



**Please open your Practical Exercise book to Exercise 6-3.**

**Time to Complete: 5 Minutes**

# Summary



- Managing file ownership
- Mounting file and directory permission


# Questions




# Questions?

# Check on Learning



## Question 1

**You need to change the owner of a file named /var/opt/runme from clittle, who is a member of the users group, to jsmith, who is a member of the editors group. Assuming you want to change both user and group owners, which command will do this?**

A. chown clittle jsmith /var/opt/runme

B. chown –u “jsmith” –g “editors” /var/opt/runme

C. chown jsmith /var/opt/runme

D. chown jsmith .editors /var/opt/runme

# Check on Learning



## Question 2

**Which permission, when applied to a directory in a file system, will allow a user to enter the directory?**

A. Read

B. Write

C. Execute

D. Access Control

# Check on Learning



## Question 3

**A user needs to open a file, edit it, and then save changes. What permissions does this user need to accomplish this? (Choose two.)**

A. Read

B. Write

C. Execute

D. Access Control

# Check on Learning



## Question 4

**A file named customers.odt has permissions established of rw-r--r--. If dedwards is not the file's owner but is a member of the group that owns this file, what can dedwards do with it?**

A. dedwards can open the file and view its contents, but can't save changes.

B. dedwards can open the file, make changes, and save the file.

C. dedwards can change ownership of the file.

D. dedwards can run the file if it is an executable.

# Check on Learning



## Question 5

**A file named sysapp has permissions established as 755. If jsmith does not own this file and isn't a member of the group that owns the file, what can jsmith do with the file?**

A. jsmith can change the group that owns the file.

B. jsmith can open the file, make changes, and save the file.

C. jsmith can change ownership of the file.

D. jsmith can run the file.

# Check on Learning



## Question 6

**You need to change the permissions of a file named calendar.odt so that the file owner can edit the file, users who are members of the group that owns the file can edit it, and users who are not owners and don't belong to the owning group can view it but not modify it. Which command will do this?**

A. chmod 664 calendar.odt

B. chmod 555 calendar.odt

C. chmod 777 calendar.odt

D. chmod 644 calendar.odt

# Check on Learning



## Question 7

**Your Linux system's umask variable is currently set to a value of 077. A user named dalexander (a member of the users group) creates a file named mythoughts.odt. What can users who are members of the users group do with this file?**

A. They can view the file, but they can't modify or save it.

B. They can open, modify, and save the file.

C. They can open, modify, and save the file. They can also execute the file if it is an executable.

D. They have no access to the file at all.

# Check on Learning



## Question 8

**An executable file has the SUID permission set. If this file is run on the system, who owns the file?**

A. The user who created the file remains the owner.

B. The user who ran the file becomes the file's permanent owner.

C. The user who ran the file becomes the file's temporary owner.

D. The root user becomes the file's owner.

# Check on Learning



## Question 9

**A directory is owned by the users group and has the following permissions assigned rwxrwxr--. It also has the Sticky Bit permission set. What effect does this have on files within the directory?**

A. Users who are members of the users group can only delete files within the directory for which they are the owner.

B. No user is allowed to delete files in this directory.

C. Users who are members of the users group can delete any file within the directory.

D. Others can enter the directory and delete files within the directory for which they are the owner.