

Casey Bladow

CSIS 360 - Spring 2015

Assignment 8 - 25 points

Due ~~Tuesday, March 31~~ Thursday, April 2

Follow the [Creating Assignments](#) handout while performing the below steps. The purpose of this assignment is to become familiar with the Linux file system as described in Chapter 6.

If any of you find that I am misusing or have incorrect terminology in any of my descriptions, please drop me a note with an explanation. I'm learning too!

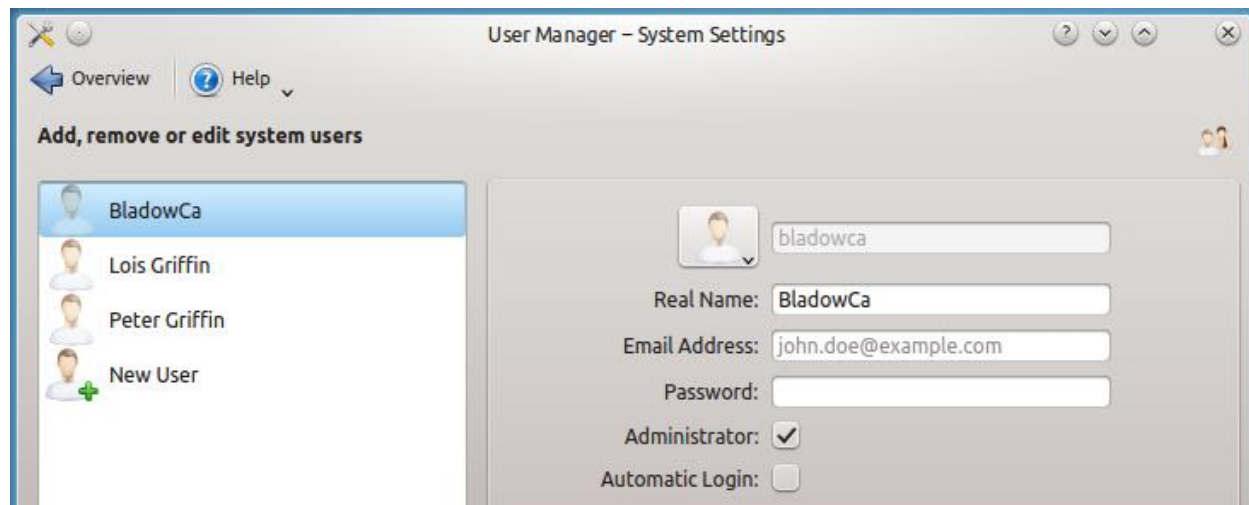
1. The goal of this step is to create user accounts and groups. By creating accounts and groups, it makes the material in Chapter 6 more meaningful and allows us to do some real examples as described later in this assignment. I would suggest that as you create accounts, you give them the same password (maybe same as your main account?) so that you do not forget them. The accounts and groups to be created are as follows.

griffinpe (Peter Griffin) in the group *parents*
griffinlo (Lois Griffin) in the group *parents*
griffinme (Meg Griffin) in the group *kids*
griffinch (Chris Griffin) in the group *kids*
griffinst (Stewie Griffin) in the group *kids*
griffinbr (Brian Griffin) in the group *animals*
griffinru (Rupert Griffin) in the group *animals*
monkeyev (Evil Monkey) in the group *animals*

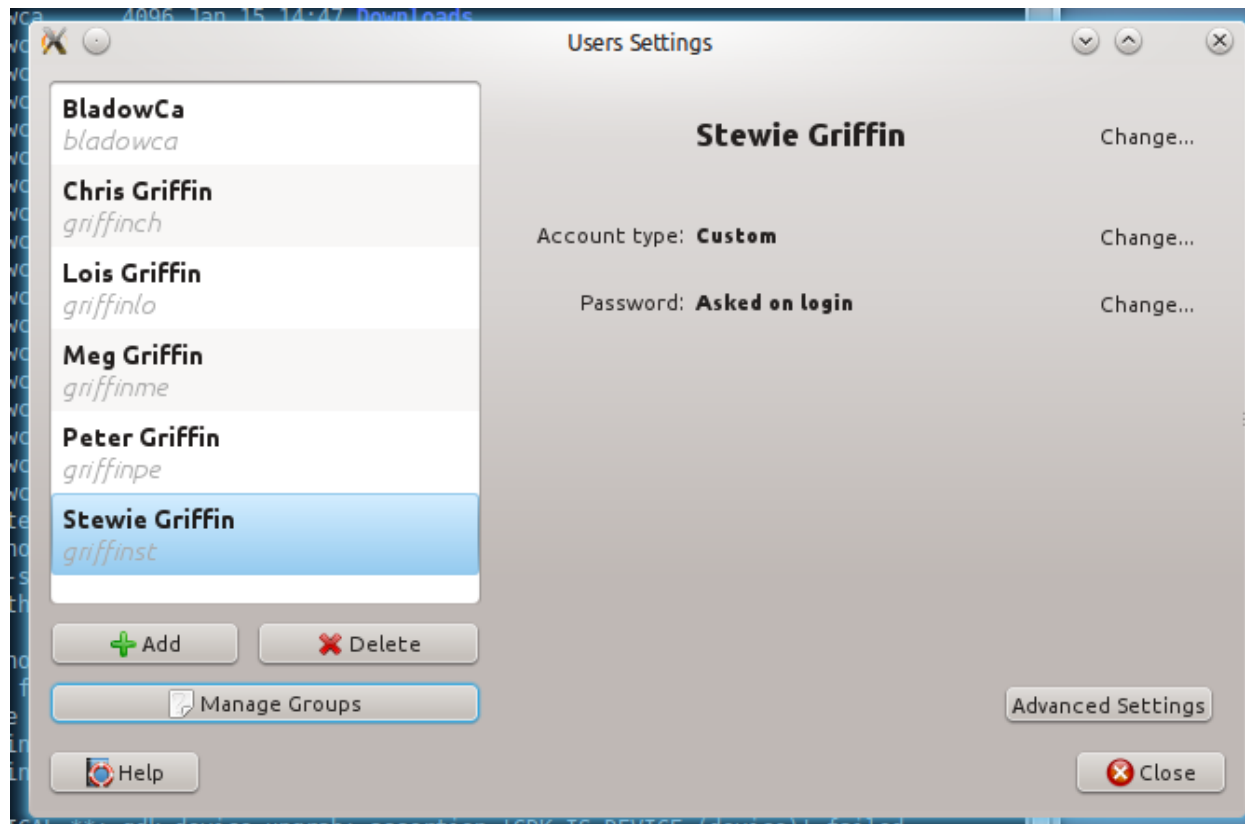
Chapter 16 pages 593-598 discusses creating accounts and groups but it is a bit outdated (though it still could be used) so let me guide you through the process. However, still read those pages in the textbook before you begin. To create these accounts and groups, follow the below steps. Note that each user's primary group has the same name as the user. That's why when you type `ls -l` you see two columns with your account name, the left one is the User and the right one is the Group.

- Create the first two accounts (for Peter and Lois) using the *User Accounts* found in *Systems Settings*. You must first click on Unlock in the upper right. Then you can click on the the + symbol in the lower left to

create the account. You'll note that *Account disabled* appears in the password box. Give the accounts a password to activate the account. After you are done, click on Lock in the upper right to lock it again and **do a window grab of the User Accounts window and include it in your assignment 8 document**. Note that there is no way to add the users to a group or even create groups using this tool.



- Install *gnome-system-tools* using your preferred method of installing software. This will give you *users-admin* described starting on page 594 and allow you to manage groups. *users-admin* is no longer part of the base install of Ubuntu 14.04. After it is installed, run it either by finding *Users and Groups* somewhere in the GUI menus or by typing *users-admin* from the command line (do not run it as sudo). Add the next 3 accounts (for Meg, Chris, and Stewie) using this tool. Create two groups, *parents* and *kids*, then add the accounts to the appropriate groups as given above. This should be fairly intuitive but feel free to get help in class if you need it. **Take a window grab of the Users Settings window showing the 3 accounts you just created and include it in the assignment 8 document.**



Note, this window grab won't show the groups but I'll get that verification later. Before exiting this tool, click on your main account then click on *Advanced Settings* and look at the 3 tabs *Contact Information*, *User Priviledges* and *Advanced* (especially noting *User Priviledges*). Now do the same thing for one of the five other accounts you've created (again, especially noting *User Priviledges*).

- Add the remaining users (for Brian, Rupert, and Evil Monkey) using the command line. The commands for doing this are listed starting on page 597. However, from my online readings, newer utilities simplify the process and even use the older utilities as the back end. For example, *useradd* (page 597) creates the user but it does not create a home directory for the user (even though textbook says it does... it doesn't... meaning it likely did in an earlier distribution of Ubuntu). Instead of using *useradd*, we can use *adduser* which takes care of everything. So for the first, type in the following command and enter what's appropriate at the prompts (Full Name[]: should be Brian Griffin and the rest can be left blank).

```
sudo adduser griffinbr
```

Do the same thing for the remaining users. Yes there are options that can be given

(http://linux.about.com/od/commands/l/blcmdl8_adduser.htm) to add the user to a group, specify the home directory, add a password, etc. so that you wouldn't be prompted for the information, but let's just stick with the basics. Here's another useful article on user management: <https://help.ubuntu.com/its/serverguide/user-management.html>

Now lets make a group for them.

```
sudo addgroup animals
```

Now let's add them to the group. I've seen many different ways (like using *usermod*) in my readings but this seems simplest.

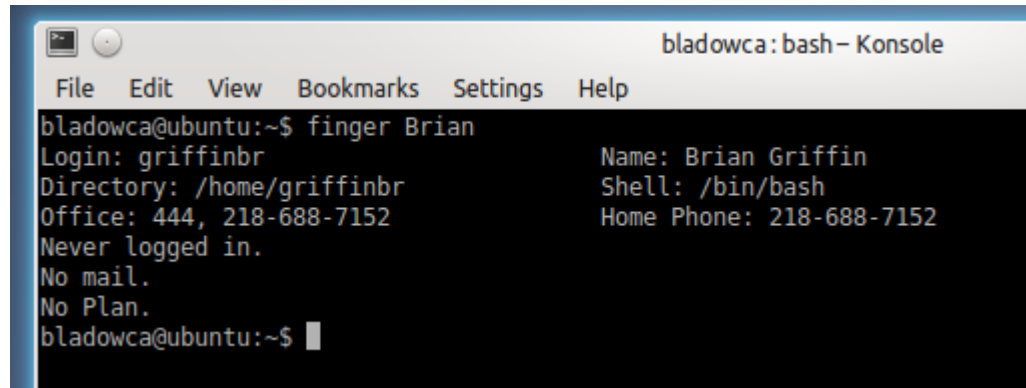
```
sudo adduser griffinbr animals
```

Repeat for the other two.

Now after having added new users into new groups with three different methods, think about this GUI junkies... what would be easier, creating 1000 accounts one at a time using the GUI or automating the process by writing a script file with a loop containing command lines whose user information can be extracted from another file?

- Type the command *ls /home* to see the accounts you created.
- To log into the other accounts, you can login from the GUI login screen or you could give the command *su username* from a terminal window (where *su* is substitute user and *username* is one of the accounts you created). If you use *su*, you will still be in the same direcotory in which you entered the command. Typing *cd* will get to the home directory of the substitute user. When you logout (cntl-d or *exit*), you'll be back to your own login.
- Log in as *griffinbr* then type the command *chfn* (not in your textbook) which will change the *finger* information (remember *finger* in chapter 5?). Make up information for everything you are prompted for.
- Log out from *griffinbr* and from the command line type *finger Brian* (lower or uppercase b doesn't matter) or *finger griffinbr*.

If *finger* isn't installed (it wasn't for me) install it. ***Take a screen grab of the terminal window showing the finger information for Brian Griffin and include it in your assignment 8 document.***

A screenshot of a terminal window titled 'bladowca: bash - Konsole'. The window has a menu bar with 'File', 'Edit', 'View', 'Bookmarks', 'Settings', and 'Help'. The terminal shows the command 'bladowca@ubuntu:~\$ finger Brian' and its output. The output is split into two columns. The left column contains: 'Login: griffinbr', 'Directory: /home/griffinbr', 'Office: 444, 218-688-7152', 'Never logged in.', 'No mail.', and 'No Plan.'. The right column contains: 'Name: Brian Griffin', 'Shell: /bin/bash', and 'Home Phone: 218-688-7152'. The prompt 'bladowca@ubuntu:~\$' is visible at the bottom.

```
bladowca@ubuntu:~$ finger Brian
Login: griffinbr                Name: Brian Griffin
Directory: /home/griffinbr      Shell: /bin/bash
Office: 444, 218-688-7152       Home Phone: 218-688-7152
Never logged in.
No mail.
No Plan.
bladowca@ubuntu:~$
```

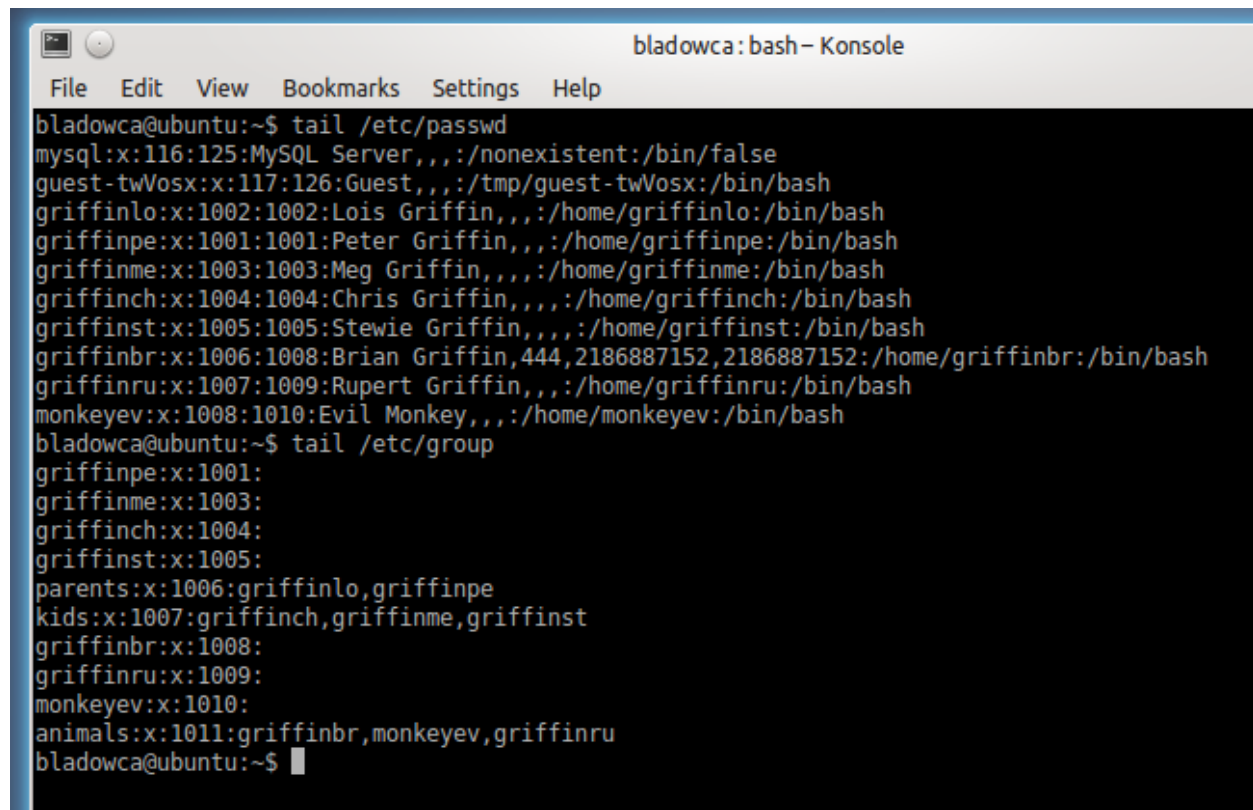
If you would type *finger griffin*, you would see all the Griffins!

- From a terminal window, type the commands (note that I'm trying to get you to incorporate previously learned material)

`tail /etc/passwd`

`tail /etc/group`

Take a window grab of the terminal window and include it in your assignment 8 document.

A terminal window titled 'bladowca: bash - Konsole' with a menu bar (File, Edit, View, Bookmarks, Settings, Help). The terminal shows the user 'bladowca@ubuntu:~\$' running 'tail /etc/passwd', which lists system and regular users like mysql, guest-twVosx, griffinlo, griffinpe, griffinme, griffinch, griffininst, griffinbr, griffinru, and monkeyev. Then, the user runs 'tail /etc/group', showing groups like griffinpe, griffinme, griffinch, griffininst, parents, kids, griffinbr, griffinru, monkeyev, and animals. The prompt returns to 'bladowca@ubuntu:~\$' at the end.

```
bladowca@ubuntu:~$ tail /etc/passwd
mysql:x:116:125:MySQL Server,,,:/nonexistent:/bin/false
guest-twVosx:x:117:126:Guest,,,:/tmp/guest-twVosx:/bin/bash
griffinlo:x:1002:1002:Lois Griffin,,,:/home/griffinlo:/bin/bash
griffinpe:x:1001:1001:Peter Griffin,,,:/home/griffinpe:/bin/bash
griffinme:x:1003:1003:Meg Griffin,,,:/home/griffinme:/bin/bash
griffinch:x:1004:1004:Chris Griffin,,,:/home/griffinch:/bin/bash
griffininst:x:1005:1005:Stewie Griffin,,,:/home/griffininst:/bin/bash
griffinbr:x:1006:1008:Brian Griffin,444,2186887152,2186887152:/home/griffinbr:/bin/bash
griffinru:x:1007:1009:Rupert Griffin,,,:/home/griffinru:/bin/bash
monkeyev:x:1008:1010:Evil Monkey,,,:/home/monkeyev:/bin/bash
bladowca@ubuntu:~$ tail /etc/group
griffinpe:x:1001:
griffinme:x:1003:
griffinch:x:1004:
griffininst:x:1005:
parents:x:1006:griffinlo,griffinpe
kids:x:1007:griffinch,griffinme,griffininst
griffinbr:x:1008:
griffinru:x:1009:
monkeyev:x:1010:
animals:x:1011:griffinbr,monkeyev,griffinru
bladowca@ubuntu:~$
```

Note the */etc/passwd* file shows you the users you created (plus note the *finger* information for Brian Griffin) and the */etc/group* shows you the group information. After you've done this, type the command *vim /etc/group* and scroll through the file to see the large number of groups. Note the *sudogroup* that only contains your original account. Also, remember the *User Priviledges* in a previous step when you were looking at that in the *Users and Groups* utility?

2. Since we now have other accounts, it will help with understanding Chapter 6, especially when it comes to access permissions, ACLs, and links. Read Chapter 6 in your textbook while at the keyboard, experimenting with the commands as you read. The next step will have you apply a number of concepts in the chapter.
3. This step makes use of access permissions, group access to files, and ACLs. All of this was described in Chapter 6, which you've now read. If the ACL utility is not installed (it already was for me) as described on page 222, you must install it using your favored method of installing software. The use of *su username* (substitute user) as described above will come in handy for this step

so that you don't have to continually log out then log in as a different user from the GUI log in screen. Remember, if you use *su*, you should immediately type *cd* to get to the home directory of the substitute user.

- Login as *griffinst* (Stewie Griffin) and create a file using the *vim* editor named *grievances.txt* containing whatever you want. Set access permission to this file to *rw- rw- ---* (read-write for the user and group, no access for other).

Next give group ownership to the *kids* group with read-write privileges with the command

```
chgrp kids grievances.txt
```

The *chgrp* utility is not covered in your text (which I find disappointing). It will change group access to the file. Your textbook also doesn't cover *chown* (change owner) which I also find disappointing. There was the perfect opportunity to cover these in Chapter 6 right along with *chmod* when it discussed UGO (User-Group-Owner). Make a special note about the bottom tip on page 217 about UGO.

- Type the command *ls -l* and you will notice that *griffinst* is the *User* that owns the file, and the the Group *kids* is now the group with the access privileges (rather than the original *griffinst* group... remember that each user's primary group has the same name as the user).
- Create an ACL for *grievances.txt* for the *parents* group that allows read only privileges. I'll give you this first command, but you'll need to be using *setfacl* on your own the rest of the way.

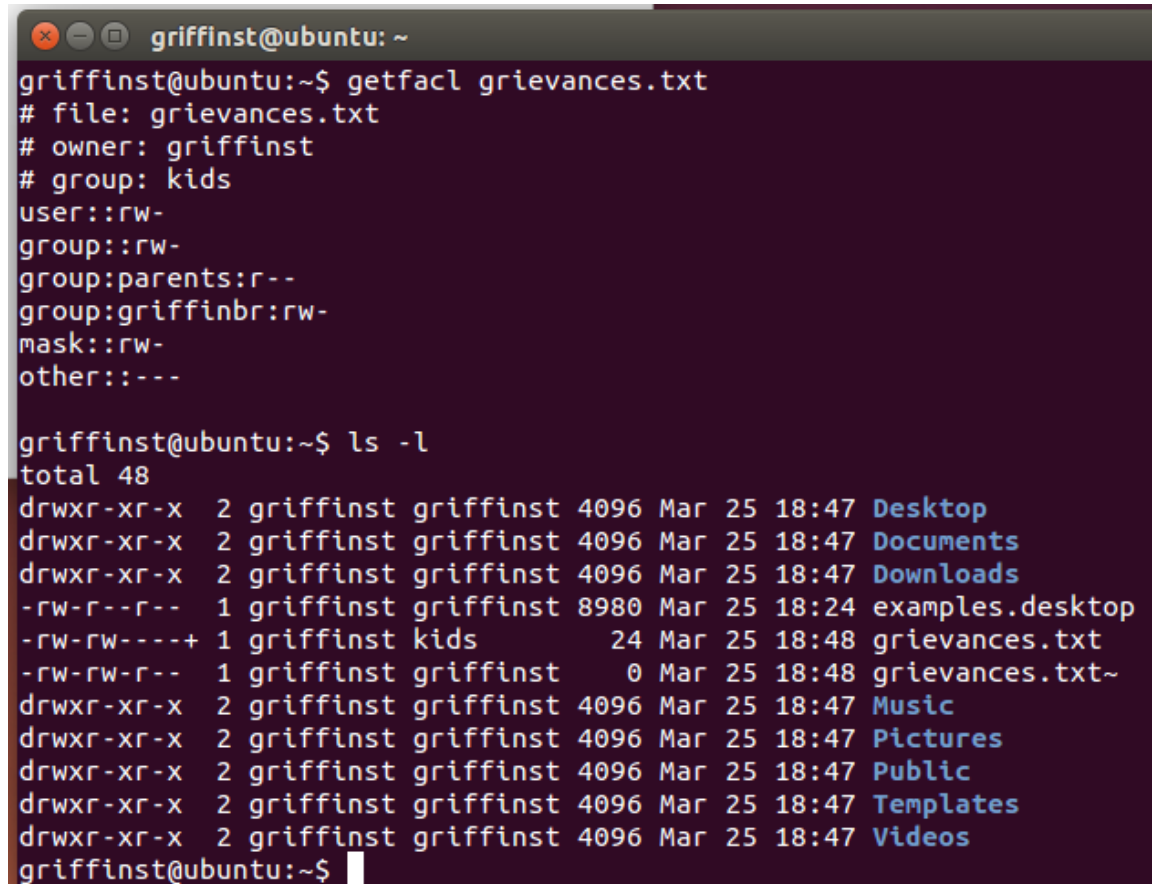
```
setfacl -m g:parents:r grievances.txt
```

Next create an ACL for the user *griffinbr* that allows read-write privileges. After this has been accomplished, type the following commands in a terminal window

```
ls -l  
getfacl grievances.txt
```

ls -l shows the standard Linux permissions with a + sign following the permissions of any file where an ACL has been assigned. The *getfacl*

grievances.txt shows the ACLs created for the file. **Do a window grab of the terminal window and include it in your assignment 8 document.**



```
griffinst@ubuntu: ~  
griffinst@ubuntu:~$ getfacl grievances.txt  
# file: grievances.txt  
# owner: griffinst  
# group: kids  
user::rw-  
group::rw-  
group:parents:r--  
group:griffinbr:rw-  
mask::rw-  
other::---  
  
griffinst@ubuntu:~$ ls -l  
total 48  
drwxr-xr-x  2 griffinst griffinst 4096 Mar 25 18:47 Desktop  
drwxr-xr-x  2 griffinst griffinst 4096 Mar 25 18:47 Documents  
drwxr-xr-x  2 griffinst griffinst 4096 Mar 25 18:47 Downloads  
-rw-r--r--  1 griffinst griffinst 8980 Mar 25 18:24 examples.desktop  
-rw-rw----+ 1 griffinst kids      24 Mar 25 18:48 grievances.txt  
-rw-rw-r--  1 griffinst griffinst  0 Mar 25 18:48 grievances.txt~  
drwxr-xr-x  2 griffinst griffinst 4096 Mar 25 18:47 Music  
drwxr-xr-x  2 griffinst griffinst 4096 Mar 25 18:47 Pictures  
drwxr-xr-x  2 griffinst griffinst 4096 Mar 25 18:47 Public  
drwxr-xr-x  2 griffinst griffinst 4096 Mar 25 18:47 Templates  
drwxr-xr-x  2 griffinst griffinst 4096 Mar 25 18:47 Videos  
griffinst@ubuntu:~$
```

To test these permissions (remember *su* described above for convenience), login as *griffinpe* (Peter Griffin) and navigate to this file (*/home/griffinst/grievances.txt*) and attempt to edit it using the *vim* editor (you can read it but not change it). Then attempt to edit the same file logged in as any of the *kids* OR as *griffinbr* and you can modify the contents of the file. Finally, login as *griffinru* (Rupert Griffin) and you will notice that you have no access to the file.

If all of the above was done right, it should give you a pretty good understanding of the standard Linux permissions and ACLs. If you're not clear or need help, ask for it in class.

- This step deals with links (and more access permissions and ACLs). You will be switching back and forth between the accounts *griffinpe* (Peter Griffin) and *griffinlo* (Lois Griffin) so I would suggest that you have two terminal windows open and use *su* as described above then simply flip back and forth. Remember that after you log in using *su*, you should type *cd* right away so that you get to the home directory of the substitute user.
 - In *griffinpe* (Peter Griffin), create a file named *TV.txt* using the *vim* editor that contains anything you want. Change the file access permissions to *rw- rw- ---*. Create an ACL for the *parents* group that allows read-write privileges.
 - In *griffinlo* (Lois Griffin), create a hard link to */home/griffinpe/TV.txt* and name it *PeterTV.txt*
 - In *griffinlo* (Lois Griffin), create a directory named *budget*. Change access permissions for the directory to *rw- rw- ---*. Create an ACL for the *parents* group that allows read-write-execute privileges. Within this *budget* directory, create a minimum of 3 files containing anything you want using the *vim* editor.
 - In *griffinpe* (Peter Griffin), create a soft link to */home/griffinlo/budget* and name it *junk*. Change to the *junk* directory and type *ls -l* and note the file permissions. Can you edit one of the files? Try changing a file. You can't because Peter Griffin is an *Other* (read only) in the access permissions unless an ACL is created for either the *parents* group or the *griffinpe* user. Try creating a file named anything you like and save it. Why did it work? Because the directory itself has an ACL for the group *parents* with read-write-execute privileges that was created earlier .
 - In *griffinlo* (Lois Griffin), navigate to the *budget* directory and attempt to edit the file created by Peter Griffin. It can be read but not written as Lois Griffin is an *Other* for this file. Verify this by typing the command *ls -l*. An ACL could be created to allow Lois Griffin to edit this file.
 - In *griffinlo* (Lois Griffin), navigate to the *budget* directory and create a directory called *March*. Navigate back to the home directory and create a soft link named *current* to the *March* directory (which is in the *budget* directory).
 - Type the following commands.

```
ls -l home/griffinpe
ls -l home/griffinlo
```

Take a window grab of the terminal window and include it in your assignment 8 document.

```
griffinpe@ubuntu: ~
griffinpe@ubuntu:~$ ls -l
total 48
drwxr-xr-x  2 griffinpe griffinpe 4096 Mar 25 19:01 Desktop
drwxr-xr-x  2 griffinpe griffinpe 4096 Mar 25 19:01 Documents
drwxr-xr-x  2 griffinpe griffinpe 4096 Mar 25 19:01 Downloads
-rw-r--r--  1 griffinpe griffinpe 8980 Mar 25 18:13 examples.desktop
lrwxrwxrwx  1 griffinpe griffinpe   22 Mar 25 19:09 junk -> /home/griffinlo/bud
drwxr-xr-x  2 griffinpe griffinpe 4096 Mar 25 19:01 Music
drwxr-xr-x  2 griffinpe griffinpe 4096 Mar 25 19:01 Pictures
drwxr-xr-x  2 griffinpe griffinpe 4096 Mar 25 19:01 Public
drwxr-xr-x  2 griffinpe griffinpe 4096 Mar 25 19:01 Templates
-rw-rw----+ 2 griffinpe parents   13 Mar 25 19:17 TV.txt
-rw-rw-r--  1 griffinpe griffinpe    0 Mar 25 19:01 TV.txt~
drwxr-xr-x  2 griffinpe griffinpe 4096 Mar 25 19:01 Videos
griffinpe@ubuntu:~$
```

```
griffinlo@ubuntu: ~
griffinlo@ubuntu:~$ ls -l
total 56
drwxrwx---+ 3 griffinlo parents  4096 Mar 25 19:30 budget
lrwxrwxrwx  1 griffinlo griffinlo   18 Mar 25 19:31 current -> /home/budget/March
drwxr-xr-x  2 griffinlo griffinlo 4096 Mar 25 19:02 Desktop
drwxr-xr-x  2 griffinlo griffinlo 4096 Mar 25 19:02 Documents
drwxr-xr-x  2 griffinlo griffinlo 4096 Mar 25 19:02 Downloads
-rw-r--r--  1 griffinlo griffinlo 8980 Mar 25 18:13 examples.desktop
drwxr-xr-x  2 griffinlo griffinlo 4096 Mar 25 19:02 Music
-rw-rw----+ 2 griffinpe parents   13 Mar 25 19:17 PeterTV.txt
-rw-rw----  1 griffinlo parents   10 Mar 25 19:17 PeterTV.txt~
drwxr-xr-x  2 griffinlo griffinlo 4096 Mar 25 19:02 Pictures
drwxr-xr-x  2 griffinlo griffinlo 4096 Mar 25 19:02 Public
drwxr-xr-x  2 griffinlo griffinlo 4096 Mar 25 19:02 Templates
drwxr-xr-x  2 griffinlo griffinlo 4096 Mar 25 19:02 Videos
griffinlo@ubuntu:~$
```

4. **Answer the following questions in your assignment 8 document.** As always, these give you potential final test questions. You are also responsible for all the commands that you used in the above steps. Table 6-3 on page 234 is also a good final test prep.
1. Give the single command will make a directory called *first* with the directory *second* inside *first*?

```
mkdir -p /home/first/second
```

2. What are the . and .. directories?

. refers to the working directory and the .. refers to the working directory's parent directory

3. What's the difference between the home and working directory?

A home directory is where you start when you open a shell. The working directory is where you are *right now*.

4. What utility shows the working directory?

```
pwd
```

5. What utility will create an empty file without getting into an editor and saving the file?

```
touch
```

6. What's the difference between an absolute and relative pathname?

An absolute path is a complete path from start of actual filesystem from / directory.

Relative path is defined as path related to the present working directory.

7. What is a ~ (Tilde) in a pathname?

The home directory.

8. What command will delete a directory named *myfile* and any files or directories contained in it?

```
rm -rf myfile
```

9. What utility will rename a directory?

```
mv oldName newName
```

10. Give a command that will protect your home directory by not allowing any access by any other users but still give you full access.

`chmod 700 directory`

11. What option (flag) with the `cp` utility allows you to copy a directory and all of its contents to another location?

`-R`

12. Give a command that will remove all ACLs from the file `xxx`?

`setfacl -b u:lisa xxx`

13. What do the following important standard directories contain? (this is pretty much copying down what's on pages 213-15 but you'll at least know what you're responsible for on the final test) Let me know if you think I should have included any other important directory.

- `/bin` – Essential command binaries
- `/boot` – Static files of the boot loader
- `/dev` – Device files
- `/etc` – Machine-local system configuration files
- `/etc/X11` – Machine-local configuration files for the X Window System
- `/home` – User home directories
- `/lib` – Shared libraries
- `/mnt` – Mount point for temporarily mounting filesystems
- `/opt` – Add-on (optional) software packages
- `/root` – Home directory for the root account
- `/sbin` – Essential system binaries
- `/tmp` – Temporary files
- `/usr/bin` – Most user commands
- `/var` – Variable data

5. ***What did you think of this assignment? How long did it take you? Do you have any suggestions for the next time I teach this class?***

Assignment took roughly 3 hours to complete. The assignment wasn't too difficult. Definitely had to read the instructions a few times and as careful as possible... but there really isn't any way to make that easier either.

6. ***What did you learn from this assignment?***

Creating users and groups in command line. Setting/changing permission settings from the command line. Got a rough understanding of ACLs and how they work.