**Casey Bladow**

**CSIS 360 - Spring 2015**   
**Assignment #7 - 25 points**   
**Due Tuesday, March 10**

Follow the [**Creating Assignments**](http://dragon.mnstate.edu/~brekke/spring15/360/handouts/assignments.htm) handout while performing the below steps. The purpose of this assignment is to give you one more good tool for other CSIS classes and also to learn about Linux utilities.

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. I wasn't exactly sure how long it would take you to do assignment 6, so I pushed one thing to this assignment that I thought would be useful for you when using your machine in other CSIS classes. After reading multiple articles about this, I still don't have a very good handle on exactly what is necessary. Therefore, I'll make this a freebie (I'll simply give you the commands and not have you read the articles and extract what's pertinent). In exchange for this freebie, you'll have to read my long rambling description.

An earlier assignment had you use the Nautilus file manager to connect to your smaug account by clicking on *Connect to Server* then entering *sftp://yourStarID@smaug.mnstate.edu*. This made your smaug account available in your file manager. As far as I can tell, this is the same as using gvfs-mount from the command line with the command  
*gvfs-mount sftp://yourStarID@smaug.mnstate.edu*.  
After this command, you will see your smaug account under *Network* (on the left) in your file manager. If you look at the directory */run/user/1000/gvfs* after either typing the command or clicking on*Connect to Server* from within the file manager, you will see the same directory in there (*sftp:host=smaug.mnstate.edu,user=yourStarID*).

This step is an alternative to gvfs-mount (which used sftp in the case above but can use other protocols) by using ssh as the protocol. It offers certain distinct advantages as described later. Enter the following commands from a terminal window.

* + *sudo apt-get update*
  + *sudo apt-get install sshfs*
  + *sudo gpasswd -a $USER fuse*  
    (I have tried a fresh install and was still successful even when not giving this command, hence my earlier comment about not having a very good handle on exactly what is necessary)

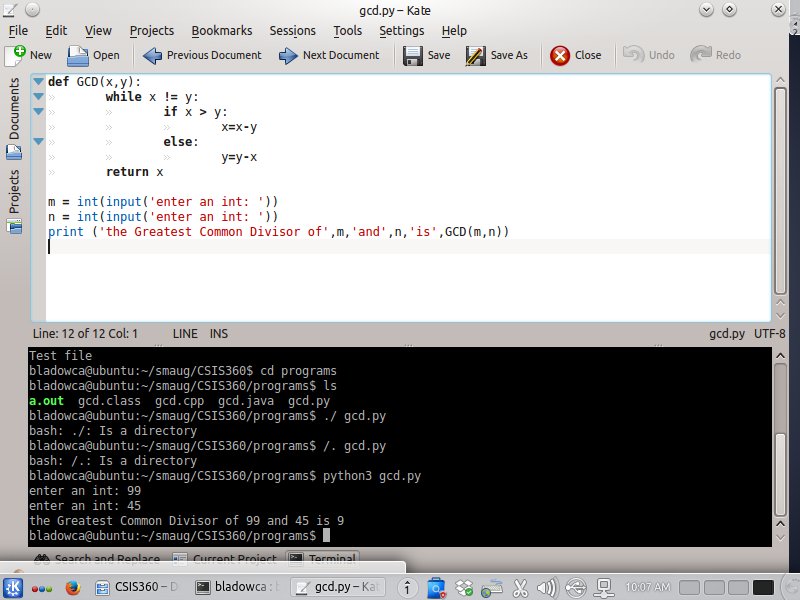
The above commands only need to be entered once and then you're good from now on. The following commands will mount a drive where ever you choose.

* + *mkdir smaug*  
    (This makes a directory for the mount point. You could choose any directory name you wish, I chose smaug. This command only needs to be done once if you simply want to keep the directory.)
  + *sshfs -o idmap=user yourStarID@smaug.mnstate.edu:  smaug*  
    (The ending *smaug* is your mount point, the directory you created. I have tried a fresh install and was still successful even not entering *-o idmap=user*).
  + enter your smaug password

You now have your smaug account file system as a directory on your local machine. To unmount it, enter the command  
*fusermount -u smaug*  
(or whatever directory name you happened to use for the mount point). You can also use paths to mount only certain directories if you wish. You cannot be in a directory that has been mounted if you're going to give the command to unmount it. The advantage to using this method over gvfs-mount is that you can specify your own mount point within your local directory and you can access files either within the file manager or from the command line (you can't access the files from the command line using gvfs-mount). This also provides a slick way of accessing your smaug account off campus in a windowed environment, though the recent installation of X2Go on smaug works pretty well. You could also use sshfs on other servers such as *puff* (which does not have X2Go as an option).

Here's a description of the slick access using the GUI *kate* editor as an example, but this works for any windowed application. As you know, you can use ssh to smaug with the -X option then run the*kate* editor on smaug. This works fine on campus but if you do it off campus, you don't have the necessary bandwidth to make it a good experience. That's where sshfs comes in. You can run the *kate*editor on your local machine and open a file on smaug (sshfs transfers the file to your local machine). Or you could navigate to the smaug file you want and open it with *kate* (which would be *kate* on your local machine). When you save the file, sshfs saves the file back to smaug). These are simple text file transfers so it doesn't take very long. Extending this example to a computer program, say C++, it will compile the program on your local machine and transfer the executable back to smaug. As long as you're using the same compiler (g++), the source should compile on either your machine or on smaug. The executable *should* also run fine on smaug and also on other Linux distros. You certainly should test to make sure if you're turning in an assignment for a class. Seem a little convoluted? Well, it works and the lag from file transfers is hardly noticeable.

***Open one of the smaug files you created in assignment 6 using the kate editor on your local machine. Be sure that the terminal window is displaying in kate so that I can see the prompt. Take a screen shot of the window and include it in your assignment 7 document.***



Here are a few of the articles I've read on *sshfs* that are the easiest to understand. You can read them if you are interested. All have different options when entering the commands, hence leading to my lack of full understanding. I chose the first article to give you the commands listed above. If you have additional information and knowledge on this subject, please discuss with me.

* + **[https://help.ubuntu.com/community/SSHFS](https://help.ubuntu.com/community/SSHFS" \t "blank)**
  + **[https://www.digitalocean.com/community/tutorials/how-to-use-sshfs-to-mount-remote-file-systems-over-ssh](https://www.digitalocean.com/community/tutorials/how-to-use-sshfs-to-mount-remote-file-systems-over-ssh" \t "blank)**
  + **[http://tecadmin.net/install-sshfs-on-linux-and-mount-remote-filesystem/](http://tecadmin.net/install-sshfs-on-linux-and-mount-remote-filesystem/" \t "blank)**

1. For the next part of the assignment, the first thing you need to do is read about the *script* utility on page 172. Everything you do below should be done with the *script* utilty recording everything that you are typing and saved to the file *typescript*. When you first open a terminal window, type  
     
   *script -a*  
     
   Since you just opened the terminal window, the *typescript* file will be saved in your home directory. The -a option is to append. That way if you go through Chapter 5 in multiple sessions, it will continue to append to the *typescript* file each time. Type *exit* or press Cntl-D to stop recording when you take a break from this assignment but remember to use the -a option to append when you resume. Will the *typescript* file be a mess? Yes, an unbelievable horrible mess. But I don't care if it contains mistakes you made while doing the assignment or whatever and the format of it will look like crap. *script* even records typing errors and the subsequent backspace to correct to it, but at least I'll be able to do a search to see if you entered the commands to run the utilities (and maybe I'll find a utility that will help clean it up). I'm going to give this a shot rather than have you doing multiple screen grabs. We'll see how it works out. Note: if you ssh to smaug during your session, it will NOT record your password.

***After you've completed all of chapter 5 in the step below, upload your typescript file to D2L in the assignment 7 folder. Oh, now that I think about it, you'll have to rename this file to typescript.txt so that D2L will accept it.***

1. Read all of this step before beginning to get an idea of what you need to do. Read chapter 5 while at the keyboard trying each of the commands as you read about them (again, *script* should be recording what you do in the file *typescript*). This chapter is entitled The Linux Utilities which by nature are entered at the command line interface at a terminal window. If you don't like working from the command line, you're likely not going to like this assignment (and you'll likely have many frustrations in your career after you graduate... especially if you're a CIT major). On the other hand, maybe becoming more comfortable with the command line will help give you a more positive view of using the command line. My initial thought was for you to include each utility and a brief description in your assignment 7 document, but the chapter summary has already done that. Therefore, you're pretty much on your honor to try each of the utilities (though I will have your *typescript* file for verification. Expect any of these utilities to be fair game in the first test.

If the textbook doesn't give a good enough description for you or you are not comfortable with a utility, **THAT'S WHAT CLASS TIME IS FOR**. Call me over and we'll discuss it.

You have a *CSIS360* directory that you created in the previous assignment that contains 7 files. This might be a good place to start. You may need to modify and / or create new files so that you can get a good understanding of the utility. Of course, I would recommend that any creation or modification of files be done using the *vim* editor. You don't need to print any files but read about the print utility.

There may be certain utilities that are not installed. Install them using apt-get so you can give them a try.

On page 174, you can quickly create the *letter\_e* file using the *vim* editor. Type the following.

* + *vim letter\_e* (edit the file letter\_e with the vim editor)
  + *72ie<ESC>* (72 times, insert e, then escape to command mode)
  + *yy7999p* (yank the line, then 7999 times paste it to give you 8000 lines)
  + *ZZ* (exit vim saving the file)

On page 180, the tip says "If you are not on a network, skip to the *vim* tutorial." Instead, log into your smaug account and try them.

Obviously, you can skip the *vim* tutorial starting on page 186 as you've already done it.

1. ***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?***

Probably only spent a little over 2 hours on this assignment. For the most part it was pretty easy going.

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

A few new commands for the command line. Some of those utilities could definitely come in handy, each having their own purpose.

Note: you will have 2 files uploaded to D2L, your *typescript.txt* and your assignment 7 document.