UNIX

Goal

The goal of this exercise is to learn to work directories and files on a Unix or Linux web server (which is most of the web servers), using a text editor, terminal emulator program, and FTP client. To help you with this assignment, there are UNIX command references **provided at the end of the document.** Take a look at them before you start; it will help you out.
[Note: The web server (what used in the URL) is called people.rit.edu. What you use, as a developer, to upload/edit files is called banjo.rit.edu .]

Part 1: Transferring Files and SSH

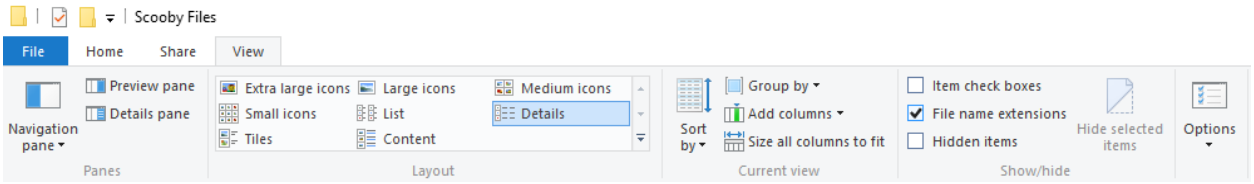
1. Create Files

On your *local* machine, create the following files using Komodo Edit (available on lab machines, and freely downloadable), or any other plain text editor of your choice. (Do not use a word processor, or Windows Notepad, as they can leave hidden characters that will create problems for you.)

In the left column of the table is the file name (remember that Unix is case sensitive!). On the right side is the text that should be in the file. Save all of the files in a folder of your choice—preferably to a cloud storage space like Google Drive or Dropbox, or to a USB drive, rather than to the local computer's hard drive.

Filename	Contents
Scooby.txt	**Hello, I'm Scooby**
Shaggy.txt	**Hello, I'm Shaggy**
Velma.txt	**Hello, I'm Velma**
Daphne.txt	**Hello, I'm Daphne**
Fred.txt	**Hello, I'm Fred**
Ghost.txt	**Hello, I'm Ghost**
OldManWithers.txt	**Hello, I'm Old Man Withers**

If you don't see the .txt extension when viewing the files on your computer, choose "View" from the top of the folder, and make sure the File name extensions box is checked:



3. **Open your FTP client [FileZilla is available on the lab machines, and is a free download]**
   Enter in the connection data:

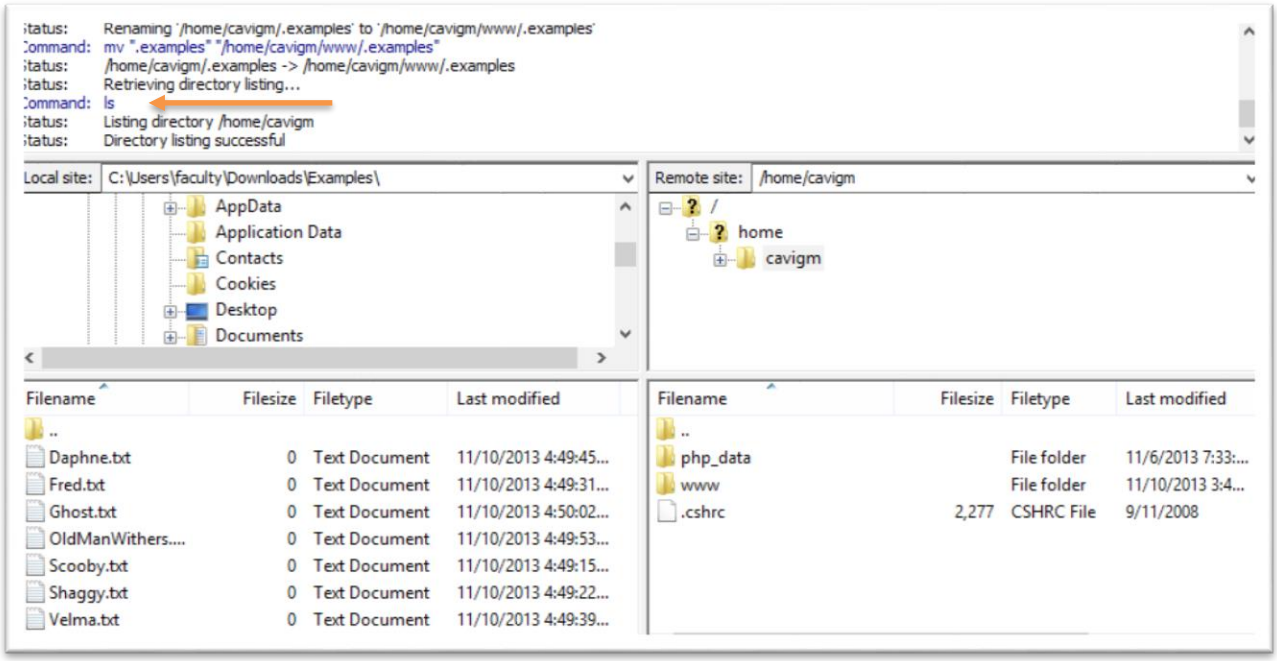
   Host: banjo.rit.edu
   Username: (your RIT username – Example: abc1234)
   Password: (your RIT password)
   Port: 22

4. **Hit the "Quickconnect" button.**
   You should be automatically connected to your personal directory (home directory) on Banjo.

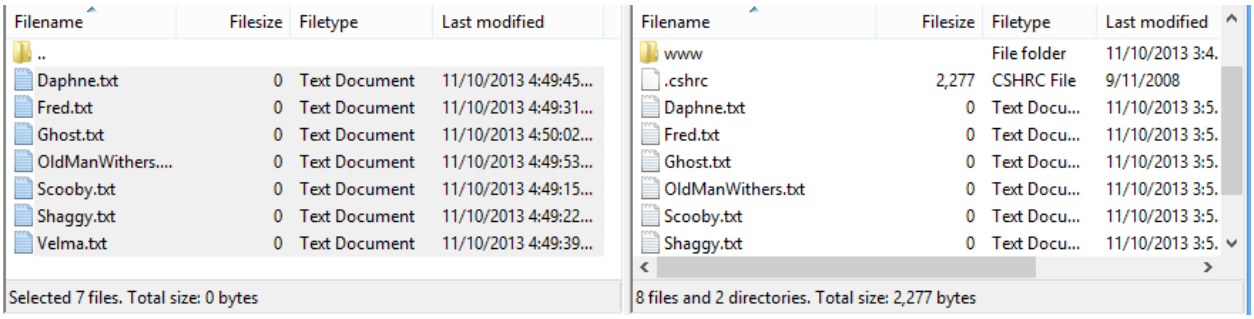
   *Note: Every student has their own personal directory on Banjo with 1GB of storage under their username. While this is your personal storage, it falls under RIT's Code of Computer Use, and violations of that code can result in you losing access to computer services. **As with all RIT services, you need to be professional about the things for which you use Banjo.**

5. **On the left pane, navigate to the folder where you saved your text files.**
   You may notice that in the top pane, you can actually see the Unix commands being run to show you folders contents and such.

   If you connected properly, you will also see your home directory on banjo in the right pane. It will most likely contain at least a **php_data** directory and a **www** directory. Ignore these for now.



6. **Drag all of the text files from the lower left pane to the lower right pane to upload them.**



7. **Now you can quit FileZilla, and switch to using Terminal or PuTTY to connect to banjo.rit.edu**

   Just as you did in the homework, use PuTTY to connect to banjo.rit.edu and login with your RIT account.

Part II: The Mystery Machine (Unix Exercise using your terminal program)

Assignment (read twice before you proceed to make sure you understand): You need to make a record all of the commands (like 'cd' or 'mkdir') as a sequential list that you used to do this exercise with the flags and filenames. Submit a word document with your written commands for each step to the dropbox before next class.

1) **Change directory to the www directory**
   Now that you are connected, use the **cd** command to move to the **www** directory. [In your document, you would record 'cd www' for step 1.]

2) **Create a new directory**
   Using the mkdir command to create a directory called **110**

3) **Change directories**
   Change directories to the new **110** directory using the **cd** command.

4) **Show your path**
   Show your current path using the **pwd** command, to ensure that you are in the 110 directory, inside of the www directory. It should look like this, with your user ID in place of abc1234:
   /home/abc1234/www/110

5) **Create directories**
   Inside of the current directory (110), create two more directories : one called **mysteryMachine** and another called **withersMansion** (remember, capitalization matters!)

6) **List the contents of your current directory**
   Type in the the **ls** command to list the contents of the current directory. It should show you the two directories you just created.

7) **Return to your home directory**
   Now return to your home directory using the `cd` command (If you don't specify a destination, the cd command will take you to your home directory

8) **Use `pwd` and `ls` to verify that you are back in your starting directory**
   Make sure you are in your home directory (/home/abc1234/), and that you can see the scooby-doo text files you uploaded.

9) **Move Daphne to the 110 directory**
   Using the `mv` command move the Daphne.txt file to the 110 directory. Since the 110 directory is inside of the www directory, and the www directory is in the directory you're currently in, the path you're moving the file to would be www/110

10) **List the contents of the 110 directory with this command : `ls www/110`**
    If you moved the Daphne.txt file properly, it should show up in the listing. If not, you may have accidentally renamed the Daphne.txt file rather than moving it—in which case you may want to ask for help here ☺

11) **Once you're sure you've moved the Daphne.txt file properly, move the rest of the files to the 110 directory:**
    - `Scooby.txt`
    - `Shaggy.txt`
    - `Velma.txt`
    - `Fred.txt`
    - `OldManWithers.txt`
    - `Ghost.txt`

    (if those files are the only .txt files in the directory, a shortcut for moving them all at once would be `mv *.txt www/110`)

12) **Verify text files moved**
    `cd` to the 110 directory inside www and use the `ls` command to verify all of the text files are there. **The rest of the commands in this exercise can all be executed from within the 110 directory, by using relative path names!**

13) **Copy (don't move!) the Ghost to the withersMansion**
    Now using the `cp` command, copy the Ghost.txt file to the withersMansion directory.

14) **Verify that you have two Ghosts**
    Using `ls` verify that the Ghost.txt file is inside both the withersMansion directory and the 110 directory.

15) **Removing a Ghost**
    Once you have verified the Ghost.txt file is in withersMansion and in the 110 directory, use the `rm` command to remove the Ghost.txt file from the 110 directory.

16) **Move OldManWithers**
Move the OldManWithers.txt file to the withersMansion directory.

17) **Move Fred and Velma to the withersMansion**
Move Fred.txt and Velma.txt to the withersMansion

18) **Move Shaggy and Scooby to the mysteryMachine**
Shaggy.txt and Scooby.txt are too scared, so move them to the mysteryMachine.

19) **Move the Ghost from withersMansion to the mysteryMachine**
The Ghost.txt follows them to the mysteryMachine!

20) **Move Shaggy and Scooby to the withersMansion**
Shaggy.txt and Scooby.txt are scared out of the mysteryMachine by Ghost.txt. Move them to the withersMansion.

21) **Move Daphne to the mysteryMachine to drive out the Ghost**
Move Daphne.txt to the mysteryMachine to drive out the Ghost.

22) **Discover the Graveyard**
Create a directory in your 110 directory called `Graveyard`

23) **Move the Ghost to the Graveyard.**
Move the Ghost.txt from the mysteryMachine to the Graveyard.

24) **Hide the Ghost**
Make the Ghost a hidden file by using the mv command to change the name from Ghost.txt to **.**Ghost.txt (a period added to the front of the filename)

25) **Verify the Ghost is hidden**
Now use ls to make sure the Ghost is hidden. (You can use ls -a to view hidden files, too.)

26) **You solved the mystery**
Move OldManWithers.txt from withersMansion to Graveyard

27) **To the mystery machine!**
Now move all of the Scooby gang back to the mysteryMachine, using the `mv` command. (You can use mv ***.**txt rather than specifying each file individually.)

28) **Leaving withersMansion**
Verify that all of the scooby gang are in the mysteryMachine and that OldManWithers.txt and the hidden Ghost file are in the Graveyard. (Use `ls -a` to show all files, including hidden files.)

Once you have verified everyone's location, remove the withersMansion directory using the `rmdir` command.

29) Check your work

Open a web browser and go to **people.rit.edu/abc1234/110 (replacing abc1234 with your RIT user ID).** It should look similar to this. If you can't get to this page or get a **404 Error - Page not found error**, then your permissions are incorrect or the directory paths are wrong. Use pwd to make sure your 110 directory is inside of the www directory, and that the permissions are set properly.

If you do not see both the Graveyard and mysteryMachine directories, then your directory permissions are incorrect.

➔ **Take a screen shot of this page, making sure that the URL and the file list both appear. You'll need this at the end of the exercise.**

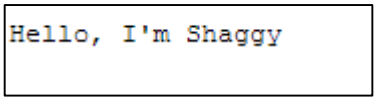
30) Check the mysteryMachine directory

Click the mysteryMachine directory. You should see all of the gang there.

➔ **Take a screen shot of this page, making sure that the URL and the file list both appear. You'll need this at the end of the exercise.**

31) Check Shaggy.txt

Hit the browser's back arrow and choose Shaggy.txt instead. You should see the text on the right. If you get a forbidden error, then the permissions on the Shaggy.txt file are incorrect. If it is blank, then you forgot to save the contents into the file.

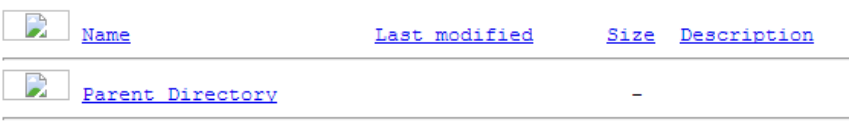


➔ **Take a screen shot of this page, making sure that the URL and the text both appear. You'll need this at the end of the exercise.**

32) Graveyard

Click the back arrow in the browser and choose the link for **parent directory**. This will take you back to the Graveyard and mysteryMachine directories.

Choose the Graveyard directory. It should look like this even if your permissions are correct. This is because the file is hidden.



➔ **Take a screen shot of this page, making sure that the URL and the file list both appear. You'll need this at the end of the exercise. Detecting a pattern?**

33) **Ghost**

The Ghost file is hidden, but it does have read permissions. Instead of trying to get there through the directory, go there directly. In the brower's address bar put in

http://people.rit.edu/abc1234/110/Graveyard/**.**Ghost.txt (replacing abc1234 with your RIT user ID).

**Make sure you put the period before* *Ghost.txt, so it is Graveyard/.Ghost.txt*

In the top left of the page you should see.

```
Hello, I'm Ghost
```

If the page is blank, then you forgot to save the contents of the file. If you get a 404, then the file pathes or browser address is wrong. Use pwd to check the path on unix and make sure the path is spelled correctly in the browser.

If you get a forbidden error, then your permissions for this file are incorrect.

➔ **Take a screen shot of this page, making sure that the URL and the contents both appear. You'll need this at the end of the exercise.**

34) **Submission**

a) At the beginning of the document where you recorded your UNIX commands, add your name and the URL to your 110 directory (e.g. http://people.rit.edu/abc1234/110/ , with your user ID replacing abc1234) at the top of the document.

b) Include all the screenshots you took in the earlier steps. (If you need help with this, just ask!)

c) Submit the document to the dropbox for this week's exercise, no later than your class time on Thursday. There is another activity for Thursday.

Reference:

Command Order

command -flags *argument argument argument …. Example ls –a h* meaning list all that starts with h*

command	Mnemonic for what you want to do
-flags	Letters that alter the command
argument	Zero or several additional pieces of information

Command Reference

Command	Function
pwd	**P**rint **W**orking **D**irectory
cd *path*	Change Directory (moves you to the path/directory specified)
man *command*	Read **man**ual for command
ls	Lists the files in a directory
mkdir *filename*	**Creates a directory**
mv *filename1 filename2*	**M**ov**e**s file1 to file2 (how to rename a file)
cp *filename1 filename2*	**Co**pies file1 to file2
rm –i *filename*	**Rem**ove **filename** **(i means interactive – asks for confirmation before doing anything)**
chmod *options filename*	**Ch**ange **Mod**e (sets permissions for a file)

Flag Reference (using the ls command for example)

Command -flag	Function
ls	**Lis**ts basic contents of current directory
ls -a	**A**ll files (even hidden ones)
ls -l	**L**ong details for
ls –al	**A**ll **L**ong details

Permission Number Table

Decimal Code	Read bit	Write bit	Execute bit
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Permissions order:
owner group others

owner	The owner of the file
group	The group to which the owner belongs
others	Everyone else (including the web server)

Example of changing to owner having full permissions and the group/others having no permissions
chmod 700 filename

Example of changing to owner and group having full permissions and others having no permissions
chmod 770 filename