

# Tutorial Set 4

[4.1 Explore File Attributes](#)

[4.2 Modify File Permissions](#)

[4.3 Copy Files](#)

[4.4 Move and Remove Files](#)

[4.5 Find and Locate Files](#)

[4.6 Work with Files Outside Your Home Directory](#)

## 4.1 Explore File Attributes

1. Log into atlas.sheridanc.on.ca remotely (if you are not already logged in).
2. Display long listing of your current directory.
3. Note each line and identify the following attribute. Which column contains
  - a) name of the file or directory \_\_\_\_\_
  - b) date and time (last modification) \_\_\_\_\_
  - c) name (userID) of the owner (creator) of the file \_\_\_\_\_
  - d) default (also called primary) group of the owner \_\_\_\_\_
  - e) size of the file or directory \_\_\_\_\_
  - f) type of file (d, -, l, c, b, s...) \_\_\_\_\_
  - g) file permissions \_\_\_\_\_
4. What does it mean when the first character on the line is a dash?
5. What does it mean when the first character on the line is the letter d?
6. Note the nine characters (2 to 10) in the first column. In the next exercise you will be changing these characters.
  - a. Whose permissions are affected when the first triplet is modified? \_\_\_\_\_
  - b. Whose permissions are affected when the second triplet is modified? \_\_\_\_\_
  - c. Whose permissions are affected when the third triplet is modified? \_\_\_\_\_
  - d. What permission is represented by the letter r? \_\_\_\_\_
  - e. What permission is represented by the letter w? \_\_\_\_\_
  - f. What permission is represented by the letter x? \_\_\_\_\_
  - g. What permission is represented by the symbol dash? \_\_\_\_\_

## 4.2 Modify File Permissions

1. There are two modes for modifying permissions: a) \_\_\_\_\_ b) \_\_\_\_\_
2. Navigate to your sub-directory week04 under your home directory. (If you don't have one, create it first.) In the ~/week04 sub-directory, create a file called todayinfo using redirection that includes the system date and complete the following steps.  
\_\_\_\_\_
3. Note the permissions for the new file (HINT: use the long listing to view the permissions)  
\_\_\_\_\_
4. Give the following command and note the new permissions:  
`chmod 777 todayinfo`
5. Give the following command and note the new permissions:  
`chmod 000 todayinfo`
6. Give the following command and note the new permissions:  
`chmod u+rwx,go+r todayinfo`
7. Give the following command and note the new permissions:  
`chmod g+w,u-x todayinfo`
8. Change the permissions to include the ability for the owner to execute the file. Use symbolic representation.  
\_\_\_\_\_
9. Change the permissions absolutely, so that the owner can read and write, the group can read, write, and execute, and others have no permissions at all. Use symbolic representation  
\_\_\_\_\_
10. Change the permissions using octal representation so that the owner can write only, the group can read only, and others can execute only.  
\_\_\_\_\_
11. Restore the original permission, which you noted earlier. Save your history in file called Tutorial4.txt, exit, and close the application. Log out.  
\_\_\_\_\_
12. In what situation might the following file permissions be useful?
  - a) `r--rw-r--`
  - b) `r--r--rw-`
  - c) `rw--w--w-`
  - d) `-w--w--w-`
  - e) `rxwxr-xrwx`
  - f) `r-x--x--x`
13. Translate the above permissions strings into their binary and numeric equivalents.
  - a) `r--rw-r--`
  - b) `r--r--rw-`
  - c) `rw--w--w-`
  - d) `-w--w--w-`
  - e) `rxwxr-xrwx`
  - f) `r-x--x--x`

## 4.3 Copy Files

Make sure you are in your home directory. After each of the following commands, use the `ls` command to see what files you have created in which directory

1. Create a directory called `tutorial4`  
`mkdir tutorial4`
2. Change to the sub-directory `tutorial4`  
`cd tutorial4`
3. Create a file called `data` using redirection  
`man ls > data`
4. Copies existing file `data` to a new file called `extra`  
`cp data extra`
5. Copies the system file group in `/etc` sub-directory to your home directory, under the same name  
`cp /etc/group ~/tutorial4/group`
6. The option `-i` tells the `cp` command to ask you for permission to overwrite the existing file, if the file called `mycopy` already exists  
`cp -i data mycopy`
7. Still in the `tutorial4` sub-directory, create a directory called `backups`  
`mkdir backups`
8. Copy existing file `data` to the directory named `backups`  
`cp data backups`
9. Copy existing files, `data1`, `data2`, and `data3`, to the directory named `backups`  
`cp extra mycopy group backups`
10. Still in the `tutorial4` sub-directory, create a directory called `documents`  
`mkdir documents`
11. Copies the directory itself including all its files and subdirectories  
`cp -r backups essays`
12. Copies all the files in a directory named `documents` to a directory named `backups`  
`cp backups/* documents`

## 4.4 Move and Remove Files

Make sure you are in your home directory. After each of the following commands, use the `ls` command to see what files you have created in which directory

1. Still in the tutorial4 sub-directory, create a directory called archive  
`mkdir archive`
2. Move all files from the tutorial4 directory (where you currently are) to the archive sub-directory  
`mv data extra mycopy archive`
3. Now try the following command. Note what happens.  
`mv -i backups/data archive`
4. Create the following file:  
`man date > unimportant`
5. Run the following command and note the results:  
`mv unimportant important`
6. Create the following file:  
`man mv > incomplete`
7. Run the following command and note the results:  
`mv incomplete archive/complete`
8. Note the structure and content of the tutorial4 directory. Navigate to your home directory and give the command:  
`rm -r tutorial4`
9. Describe what is being removed with the following commands:  
`rm data`  
`rm ~/essay`  
`rm bin/something`  
`rm data[123]`

## 4.5 Find and Locate Files

### Finding forgotten files with find command

Sometimes finding a file requires more than cursing at your computer or listing directory contents with ls. Instead, you can use the find command which lets you search in dozens of ways, including through the entire directory tree or through directories you specify.

### Locating lost files with locate command

If you're looking for a system file, a program or file that is part of the Unix system itself, rather than one of your own personal files in your home directory, try locate command to find it> You'll get more results than you can handle, but it's a quick and easy way to locate system files. It is not available on all Unix systems, but it is worth a try at any rate.

1. Examine the following manual pages and note the description and browse the options  
`man find`
2. The find command allows you to search for files. To search for all regular files starting in the current directory and checking all its subdirectories:  
`find . -type f`
3. How would you change the command to look for directories starting in the current directory?  
`find . -type d`
4. What command would you give to find all files that have exactly 5 links, starting in the current directory?  
`find . -links 5`
5. Find all files that have at least 15 links in your home directory  
`find ~ -links +15`
6. Find all empty files, either regular or directory, in the /etc directory and its subdirectories.  
`find /etc -empty`
7. Find all files that have been changed within the last 48 hours.  
`find / -ctime -2`
8. Find all files in your home directory that you accessed more than 24 hours ago.  
`find ~ -atime +1`
9. Find the file called music.data starting in your home directory.  
`find ~ -name music.data`

NOTE: The locate command is not available on atlas.

10. The locate command allows you to search for system files.  
`locate useradd`

## 4.6 Work with Files Outside Your Home Directory

1. The /etc subdirectory contains configuration files. Most of these files are plain text files that can be viewed by all but modified only by the administrator (root).
2. Navigate to the /etc directory
3. File called passwd contains account information about each user on the system. Display the long listing of this file and note all the attributes of the file  
`ls -l passwd`
4. You can display the contents of the file:  
`cat passwd`
5. Since this is a very large file, you can display just the line count: (the option is 'l' not one)  
`cat passwd | wc -l`
6. You can also filter the lines to examine your entry only: (substitute userID with your user ID)  
`cat passwd | grep userID`
7. Once you have displayed your record, note the fields it contains. Note that the fields are separated (delimited) by the symbol colon. Note the field information:  
Field 1: \_\_\_\_\_  
Field 2: \_\_\_\_\_  
Field 3: \_\_\_\_\_  
Field 4: \_\_\_\_\_  
Field 5: \_\_\_\_\_  
Field 6: \_\_\_\_\_  
Field 7: \_\_\_\_\_
8. You can extract specific field or fields. For example, to extract (cut) your home directory:  
`cat passwd | grep userID | cut -d: -f7`
9. You can now assign this string value to a variable:  
**myVar**="cat passwd | grep userID | cut -d: -f7"
10. You can also create an alias: (note that for the time being these assignments are temporary, for the duration of the current session).  
`alias cgc='cat passwd | grep userID | cut -d: -f7'`
11. And then you can give the command (or whatever name you choose)to execute that long command:  
`Cgc`
12. If you assign the value to a variable, like myVar above, you can use the variable instead of the value, which can reduce the amount of typing for long pathnames:  
`cd $myVar`
13. **Create a subdirectory called downloads in your home directory.**
14. **Extract the home directory of your instructor (bajcar) and assign her home directory to a variable.**
15. **Use the variable to navigate to her home directory. Go to the 'public' subdirectory, and copy all files to your downloads subdirectory. You will use these in later exercises. Note the permissions on directories, you will not be able to copy everything. You will need to use the recursive option to copy the files in the subdirectories.**