There are many tools that the linux shell has that are able to make your job as a developer much easier.

Let's start with something simple, first open the linux terminal.

**ls**

ls lists the files in the directory, by default your terminal will open to the home directory.

Now we will create a new folder “Lab2_YourLastName”

**mkdir Lab2_YourLastName**
Ex. Lab2_Zeliff

Once this folder has been created we will want to enter it, at the moment we are still in the home directory

**cd** Lab2_YourLastName

cd stands for change directory and takes a parameter of the path you want to move to.
(ex. Cd [file_path])

Now we should be in the directory you made, you can verify this by typing in:

**pwd**

This command will show you the current directory you’re in.

Say for example we want to go back to the home directory, cd has the option to go back

**cd ..**

This will move you “up” one level in the directory tree.
If you are within multiple folders and want to go back to your home directory you can type in

**cd ~**

Now go back to the folder you made earlier.
Once you are back, let's create three empty files

**touch file_1**
**touch file_2**
**touch file_3**

The touch command will create a file with the given name, imagine we want one more file but we want its contents to mirror that of file_1

**cp file_1 file_4**

cp is for copying a file, and file_4 will now be created as an exact copy of file_1 which at the moment will be an empty file.

Now as opposed to simply opening the file through the file browser, we can actually edit it through the command line. There are multiple text editors you can use for this, such as vim, vi, and nano.

**[text editor] file_4**
**(ex. nano file_4)**

Now that you're in the text editor type **"Hello there!"** and then close it.

We can also use the "cat" command. It reads data from the file and gives their content as output.

**cat file_4**

You should now see the contents of the file printed to the command line, all without having to ever open the file.

Another helpful command is chmod, this command is used to change permissions for a file or directory on a Unix machine. With this you can adjust the permissions of the user, a predefined group, and others. You can adjust if they have read (r), write (w), executable (x) or no permissions at all.

At the moment you should have full permissions over file_4 however we can change that.

**chmod 0 file_4**

The permissions have been changed for file_4, now lets try accessing it.

**cat file_4**

You should have gotten a permission denied error, you can also try accessing the file through a text editor.

**nano file_4**

Once again permissions are denied and you are not able to read from the file.

Lets restore our access:

**chmod u = rwx file_4**

You have restored read, write, and executable permissions and should once again be able to edit and read the contents in your file. (You can learn more about chmod here)

Suppose we don't actually want a fourth file, we can delete it through the command line

**rm file_4**

The rm command is very powerful, and once a file is deleted it becomes difficult to recover, the command line does not provide a "are you sure" prompt so be very careful when using it.

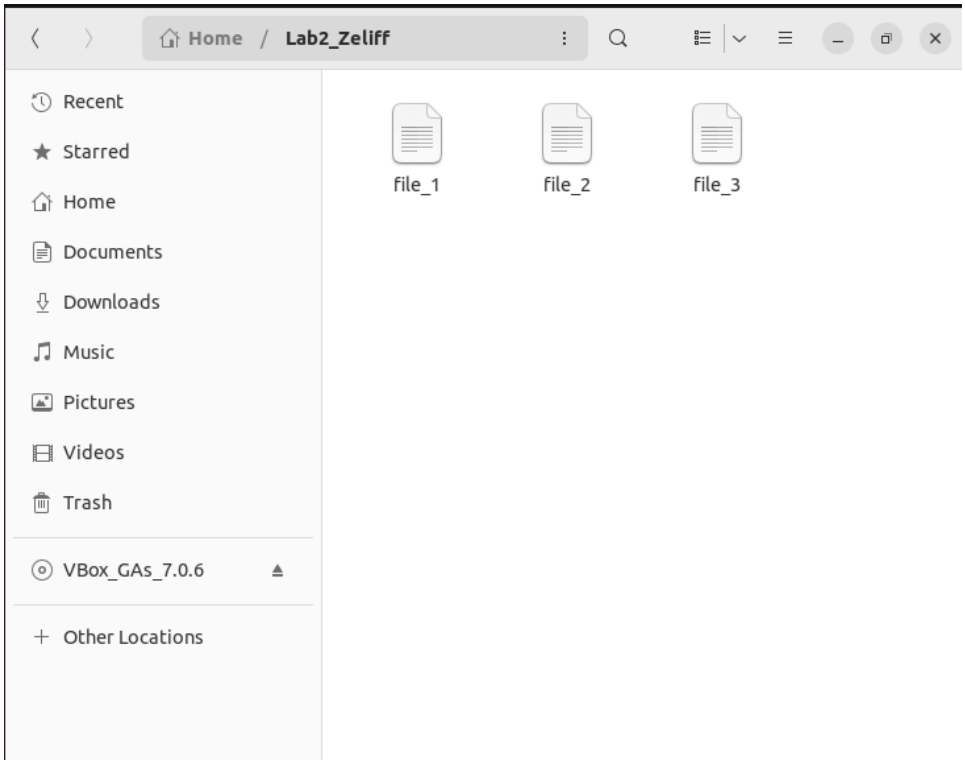
A few other helpful commands are:

**ctrl + c:** This will send a terminating signal to the current process running, for example say you have a program with an infinite loop you want to end.

**ctrl + l:** This clears the terminal, which can be helpful if you're typing multiple commands or if your terminal screen has filled up.

**man [command-name]:** This displays the manual for the given command

At this point we should have empty files in your previously created directory.



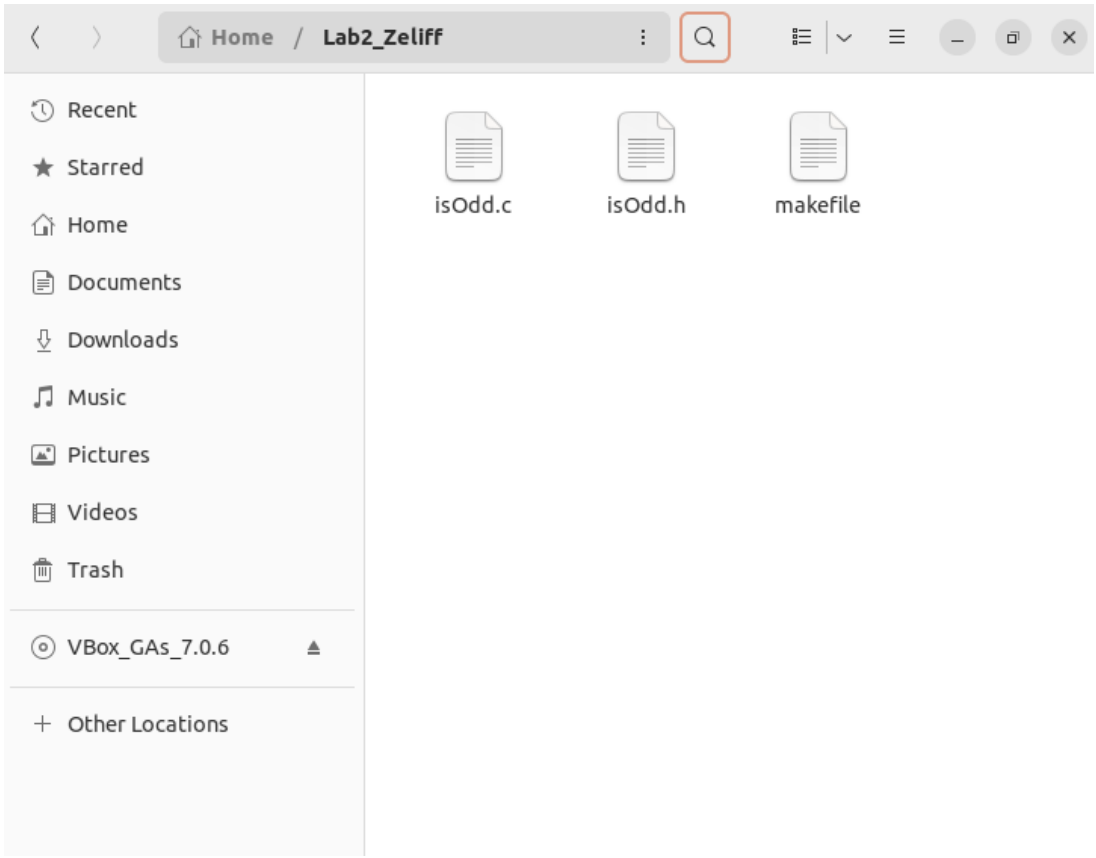

The last command we will use here is “mv”

mv is used to move one or more files/directories from one place to another in a file system. It has the ability to rename a file or folder and it has the ability to move a group of files to a different directory.

**mv file_1 isOdd.h**
**mv file_2 isOdd.c**
**mv file_3 makefile**

This should rename the files listed as the first parameter with the name given as a second parameter.

Your folder should at this point look something like this




*If your file system looks similar to the above, you can move on to the “make” section of this lab*