

Create, Modify and Remove Files and Folders in Linux

1 hour 5 Credits

[Rate Lab](#)

Introduction

Today, Linux is everywhere. Lots of the server systems behind your favorite websites are Linux-based. As an IT Support Specialist, you'll most likely be interacting with Linux on a regular basis, mainly through the command line. Actions like modifying configuration files and moving or copying them may become part of your everyday tasks. File management in Linux is a super important core skill to have as an IT Support Specialist. So, let's dive in!

Head's up: You'll experience a delay as the labs initially load (particularly for Windows labs). So, please **wait a couple of minutes for the labs to load**. Please also make sure to access the labs **directly through Coursera** and not in the Qwiklabs catalog. If you access the labs through the Qwiklabs catalog, you will *not* receive a grade. (As you know, a passing grade is required to matriculate through the course.) The grade is calculated when the lab is complete, so be sure to hit "**End Lab**" when you're done!

You'll have 60 minutes to complete this lab.

What you'll do

You'll learn how to navigate a Linux file system from the command line and understand the basics of creating, modifying, copying, and deleting files and directories. Your main learning objective for this lab is to practice these commands in the Linux VM.

Learning tip

We encourage you to try and memorize all of these commands as best you can. With enough practice, using Linux commands will become second nature to you. If you have access to your own Linux machine, try out the commands as you follow along in the next section.

If you don't have Linux available on your local machine, no worries! You can type these commands in a text editor, so you can refer back to them when you're doing the active lab exercises.

Linux overview

Before we kick things off, below is a brief intro into the Linux file system.

The **file system** controls how data is stored and retrieved in a computer. All files and folders in a Linux system are part of a bigger tree-like structure rooted at `/`. Files and folders are added to the file system by appending them to this tree structure, and deleted by removing them. **All file names are case sensitive**. When working with files and directories on the command line, special characters, like space and brackets, have to be **escaped** using a backslash.

Note: This section consists of basic commands to help you explore the Linux file system. Do not expect any interesting response back within this section, but they would be helpful in various stages of the lab.

Here are a few helpful navigation commands to help you explore the Linux file system:

You can check out the contents of the current directory using the `ls` command.

```
ls
```

You can view more details about the files, like ownership and permissions, by adding the flag `-l` to the `ls` command.

```
ls -l
```

You can see hidden files in the current directory by passing flag `a` to the `ls` command.

```
ls -a
```

You can find out where you are in relation to the rest of the file system using the `pwd` command.

```
pwd
```

You can navigate to different directories using the `cd` command.

```
cd /path/to/other/directory
```

You can check out the contents of a file using the `cat` command.

```
cat /path/to/file/file_name
```

For large input files, the `less` command allows movement within the files. The syntax is similar to that of the `cat` command, but you can move.

Reading large file using `less`

```
less /path/to/file/file_name
```

The command will provide you with a scrollable view of the content within the file, up to the end of the file content. Scroll down using "Enter", and exit the view by pressing "q".

The rest of this lab will teach you how to create, copy, modify, and remove files and folders.

Start the lab

You'll need to start the lab before you can access the materials in the virtual machine OS. To do this, click the green "Start Lab" button at the top of the screen.

Note: For this lab you are going to access the **Linux VM** through your **local SSH Client**, and not use the **Google Console (Open GCP Console** button is not available for this lab).

Start Lab

After you click the "Start Lab" button, you will see all the SSH connection details on the left-hand side of your screen. You should have a screen that looks like this:



Accessing the virtual machine

Please find one of the three relevant options below based on your device's operating system.

Note: Working with Qwiklabs may be similar to the work you'd perform as an **IT Support Specialist**; you'll be interfacing with a cutting-edge technology that requires multiple steps to access, and perhaps healthy doses of patience and

persistence(!). You'll also be using **SSH** to enter the labs -- a critical skill in IT Support that you'll be able to practice through the labs.

Option 1: Windows Users: Connecting to your VM

In this section, you will use the PuTTY Secure Shell (SSH) client and your VM's External IP address to connect.

Download your PPK key file

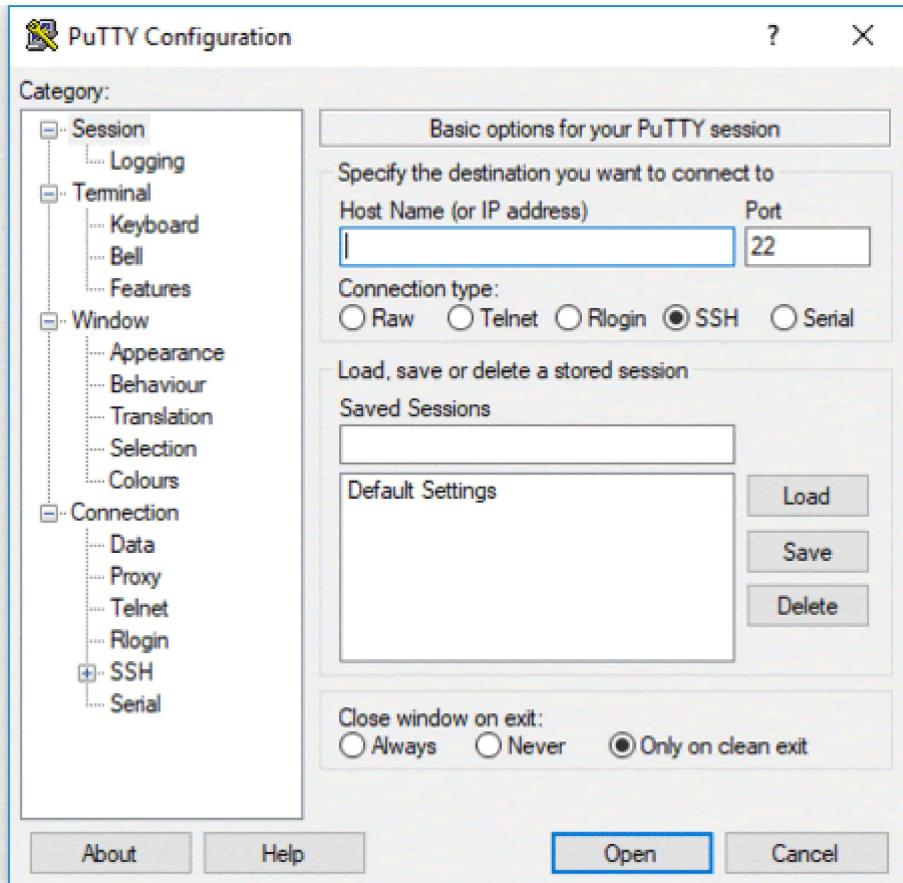
You can download the VM's private key file in the PuTTY-compatible **PPK** format from the Qwiklabs Start Lab page. Click on **Download PPK**.



Connect to your VM using SSH and PuTTY

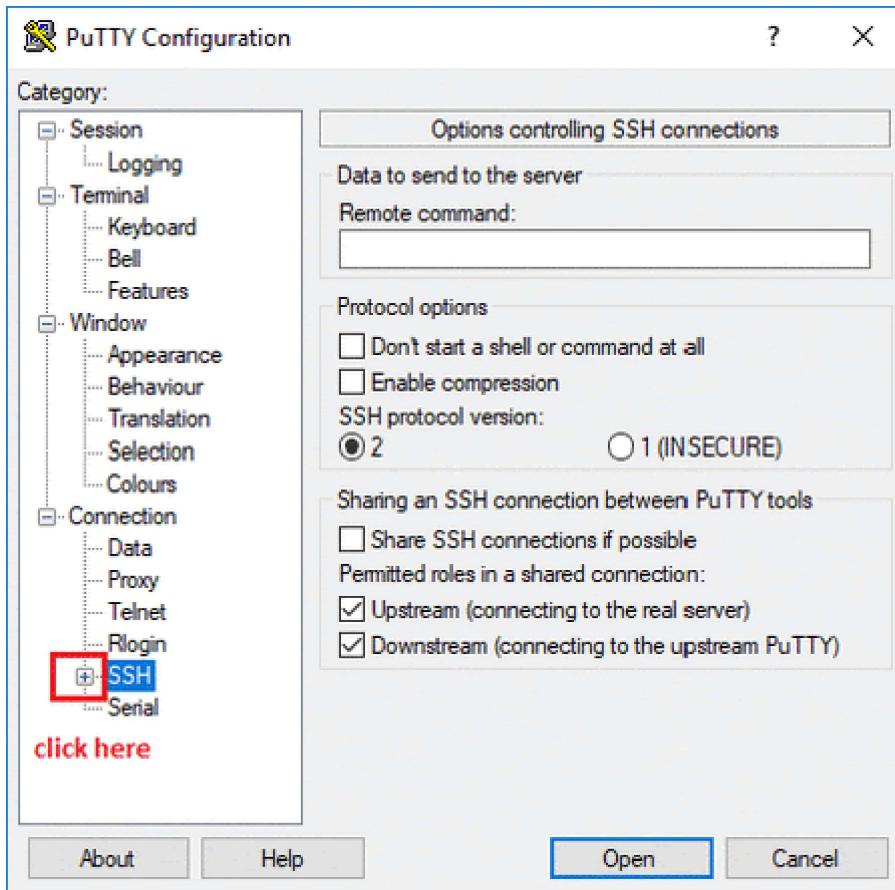
1. You can download Putty from [here](#)
2. In the **Host Name (or IP address)** box, enter
`username@external_ip_address`.

Note: Replace **username** and **external_ip_address** with values provided in the lab.



3. In the **Category** list, expand **SSH**.
4. Click **Auth** (don't expand it).
5. In the **Private key file for authentication** box, browse to the PPK file that you downloaded and double-click it.
6. Click on the **Open** button.

Note: PPK file is to be imported into PuTTY tool using the Browse option available in it. It should not be opened directly but only to be used in PuTTY.



7. Click **Yes** when prompted to allow a first connection to this remote SSH server. Because you are using a key pair for authentication, you will not be prompted for a password.

Common issues

If PuTTY fails to connect to your Linux VM, verify that:

- You entered <username>@<external ip address> in PuTTY.
- You downloaded the fresh new PPK file for this lab from Qwiklabs.
- You are using the downloaded PPK file in PuTTY.

Option 2: OSX and Linux users: Connecting to your VM via SSH

Download your VM's private key file.

You can download the private key file in PEM format from the Qwiklabs Start Lab page. Click on **Download PEM**.

 [Download PEM](#)



 [Download PPK](#)

Connect to the VM using the local Terminal application

A **terminal** is a program which provides a **text-based interface for typing commands**. Here you will use your terminal as an SSH client to connect with lab provided Linux VM.

1. Open the Terminal application.

- o To open the terminal in Linux use the shortcut key **Ctrl+Alt+t**.
- o To open terminal in **Mac** (OSX) enter **cmd + space** and search for **terminal**.

2. Enter the following commands.

Note: Substitute the **path/filename for the PEM** file you downloaded, **username** and **External IP Address**.

You will most likely find the PEM file in **Downloads**. If you have not changed the download settings of your system, then the path of the PEM key will be
~/Downloads/qwikLABS-XXXXXX.pem

```
chmod 600 ~/Downloads/qwikLABS-XXXXXX.pem
```

```
ssh -i ~/Downloads/qwikLABS-XXXXXX.pem username@External Ip Address
```

```
:~$ ssh -i ~/Downloads/qwikLABS-L923-42090.pem gcpstagingedit1370_student@35.239.106.192
The authenticity of host '35.239.106.192 (35.239.106.192)' can't be established.
ECDSA key fingerprint is SHA256:vrz8b4aYUtruh0A6wZn6Ozy1oqqPEfh931olvxtm8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '35.239.106.192' (ECDSA) to the list of known hosts.
Linux linux-instance 4.9.0-9-amd64 #1 SMP Debian 4.9.168-1+deb9u2 (2019-05-13) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
gcpstagingedit1370_student@linux-instance:~$
```

Option 3: Chrome OS users: Connecting to your VM via SSH

Note: Make sure you are not in **Incognito/Private mode** while launching the application.

Download your VM's private key file.

You can download the private key file in PEM format from the Qwiklabs Start Lab page. Click on **Download PEM**.

 [Download PEM](#)

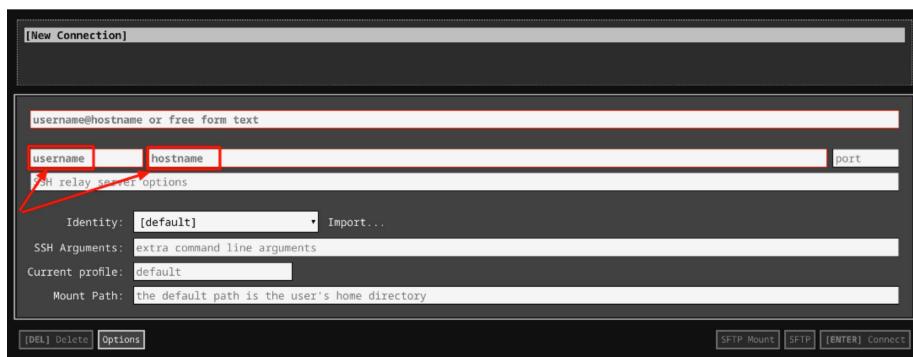
 [Download PPK](#)

Connect to your VM

1. Add Secure Shell from [here](#) to your Chrome browser.
2. Open the Secure Shell app and click on **[New Connection]**.



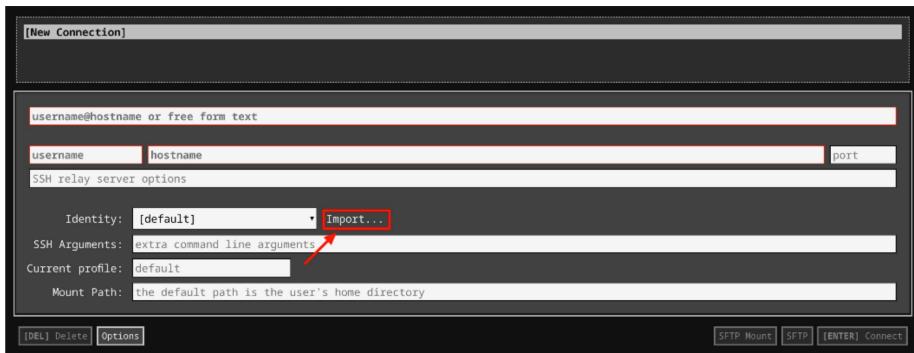
3. In the **username** section, enter the username given in the Connection Details Panel of the lab. And for the **hostname** section, enter the external IP of your VM instance that is mentioned in the Connection Details Panel of the lab.



4. In the **Identity** section, import the downloaded PEM key by clicking on the **Import...** button beside the field. Choose your PEM key and click on the **OPEN** button.

Note: If the key is still not available after importing it, refresh the application, and select it from the **Identity** drop-down menu.

5. Once your key is uploaded, click on the **[ENTER] Connect** button below.



6. For any prompts, type **yes** to continue.
7. You have now successfully connected to your Linux VM.

You're now ready to continue with the lab!

Creating directories (folders)

Directories (folders) in Linux are created using the *mkdir* command. The command takes the directory name as the argument.

Note: The example commands are for reference only.

Example 1

```
mkdir dir_name
```

Multiple directories can be supplied as arguments, and mkdir will create all of them.

Example 2

```
mkdir dir1 dir2 dir3
```

Parameters

mkdir can take three options:

- **-p:** allow mkdir to create parent directories if they don't exist
- **-m:** (mode) used to set permissions of directories during creation
- **-v:** run command in verbose mode

Let's take a look at how to use mkdir by going through an example.

In this Linux virtual machine, directory */home/user/Desktop* contains a file called "colors". We'll open the file, and for every line listed in it, we'll create a new folder with that name in the directory */home/user/Documents*

Step 1: Change into working directory.

```
cd /home/user/Documents  
student-00-a4419d066aa2@linux-instance:~$ cd /home/user/Documents  
student-00-a4419d066aa2@linux-instance:/home/user/Documents$
```

Step 2: Show the contents of the file "colors" within the "Desktop" directory.

```
cat /home/user/Desktop/colors  
student-04-6ff0b48926b1@linux-instance:/home/user/Documents$ cat /home/user/Desktop/colors  
red  
blue  
green  
yellow  
magenta
```

Step 3: Create the directories.

```
mkdir red blue green yellow magenta  
student-04-6ff0b48926b1@linux-instance:/home/user/Documents$ mkdir red blue green yellow magenta  
student-04-6ff0b48926b1@linux-instance:/home/user/Documents$ ls
```

Click *Check my progress* to verify the objective.

Create directories

Removing empty directories

To remove empty directories, use the *rmdir* command. The name of the directory to be removed is passed as an argument.

Note: The example commands are for reference only.

Example 1

```
rmdir dir_name
```

Multiple directory names can be passed as arguments, and *rmdir* will remove all of them.

Example 2

```
rmdir dir1 dir2 dir3 dir4
```

Head's up: *rmdir* only removes empty directories. To remove a non-empty directory, the command *rm*, discussed later in this course, is used.

Options

rmdir takes only one option, which tells it to remove parent directories if they're also empty.

- -p: remove parent directories, if they're also empty

Creating files

By default, the *touch* command is used to change the modification and access times of a file. If the file doesn't exist, the *touch* command is used to create a file

with default permissions.

Let's take a look at an example of how to use the touch command. In the current directory, we can create an empty file called "empty_file":

```
touch empty_file
```

The touch command can take the c option to prevent a new file from being created.

Options

- -c: do not create file if it doesn't exist

Copying, moving and deleting files and directories (folders)

cp

The cp command is used to make a copy of one or more directories or files. The command takes **at least** one source name and one target name. If the target is a file, then the source must also be a file. A copy of the source will be made with the new name supplied in target. If the target name isn't specified, a copy of source will be made in the target directory under the same name. If a file with the target name already exists in the target directory, it'll be replaced. If the target is an existing directory, then all sources (one or more) will be copied into the target directory. If the target is a directory that doesn't exist, then the source must also be a directory. A copy of the directory and its contents will be made in target under the same name.

Note: The example commands are for reference only.

Example 1

Copy the file "source_file" in the directory */home/user/* to the directory "duplicates" as "target_file".

```
cp /home/user/source_file /home/user/duplicates/target_file
```

The duplicates directory now contains a copy of the original file.

mv

The move command is used to move one or more files or directories into a different location, or rename them to a new name. You're required to pass **at least** one source and target file names or directories. The *mv* command follows the rules for existing or non-existing directories or files, as does *cp*.

Example 2

Move the file "source_file" in /home/user/ to the directory "moved_files" and give it the name "target_name".

```
mv /home/user/source_file /home/user/moved_files/target_file
```

The original directory doesn't contain the file now. It's been moved to the new directory "moved_files".

rm

The *rm* command is used to remove one or more files. You need to supply **at least** one argument to remove.

Example 3

We can remove the duplicate file we created in the directory "duplicates" using *rm*

```
rm /home/user/duplicates/target_file
```

Let's see how to copy, move, and rename files by going through a few examples.

In the directory */home/user/Pictures*, we'll take all the hidden files and move them into the directory */home/user/Documents/Hidden*

Step 1: Change into the Pictures directory.

```
cd /home/user/Pictures
```

```
student-04-6ff0b48926b1@linux-instance:~$ cd /home/user/Pictures
```

Step 2: Show the directory contents, including hidden files.

```
ls -a
```

```
student-04-6ff0b48926b1@linux-instance:~$ cd /home/user/Pictures
student-04-6ff0b48926b1@linux-instance:/home/user/Pictures$ ls -a
. .apple .banana .broccoli chocolate egg .milk
```

Step 3: Move the hidden files into the target directory.

```
mv .apple .banana .broccoli .milk /home/user/Documents/Hidden
```

```
student-04-6ff0b48926b1@linux-instance:/home/user/Pictures$ mv .apple .banana .broccoli .milk /home/user/Documents/Hidden
student-04-6ff0b48926b1@linux-instance:/home/user/Pictures$ cd
```

Click *Check my progress* to verify the objective.

Move hidden files

In the directory */home/user/Movies*, there's a folder called "Europe Pictures".

We'll move this folder into the correct directory for pictures: */home/user/Pictures*.

Note the use of the backslash "\\" to escape the space between "Europe" and "Pictures" in the directory name, "Europe Pictures".

```
mv /home/user/Movies/Europe\ Pictures /home/user/Pictures
student-04-6ff0b48926b1@linux-instance:~$ mv /home/user/Movies/Europe\ Pictures /home/user/Pictures
```

You can also use a "dot" to copy or move files to the current directory. In the directory `/home/user/Images`, we can move the file "Vacation.JPG" into the Pictures directory. To do that, we change into the Pictures directory, then add a "dot" to the `mv` command as the target.

```
cd /home/user/Pictures
mv /home/user/Images/Vacation.JPG .
student-04-6ff0b48926b1@linux-instance:~$ cd /home/user/Pictures
student-04-6ff0b48926b1@linux-instance:/home/user/Pictures$ mv /home/user/Images/Vacation.JPG .
student-04-6ff0b48926b1@linux-instance:/home/user/Pictures$ ls
chocolate egg Europe Pictures Vacation.JPG
student-04-6ff0b48926b1@linux-instance:/home/user/Pictures$ █
```

Click *Check my progress* to verify the objective.

Move files and folders

Some files in the directory `/home/user/Music` need to be cleaned up. We'll see an example of removing files and directories by removing:-

- Best_of_the_90s
- 80s_jams
- Classical
- Rock (folder)

Step 1: Navigate to the Music folder.

```
cd /home/user/Music
```

Step 2: Remove files.

```
rm Best_of_the_90s 80s_jams Classical
student-04-6ff0b48926b1@linux-instance:~$ cd /home/user/Music
student-04-6ff0b48926b1@linux-instance:/home/user/Music$ ls
80s_jams Best_of_the_90s Classical Electronic Rock Smooth_jazz
student-04-6ff0b48926b1@linux-instance:/home/user/Music$ rm Best_of_the_90s 80s_jams Classical
student-04-6ff0b48926b1@linux-instance:/home/user/Music$ ls
Electronic Rock Smooth_jazz
```

Step 3: Remove the directory.

```
rmdir Rock
```

```
student-04-6ff0b48926b1@linux-instance:/home/user/Music$ rmdir Rock
student-04-6ff0b48926b1@linux-instance:/home/user/Music$
```

To remove a directory with content, the `rm` command is used instead of `rmdir`. The option `-r` tells the command to remove the directory, along with its content recursively.

Click *Check my progress* to verify the objective.

Remove files and folders **Note:** The example commands are for reference only.

Example 4

```
rm -r non_empty_dir
```

Searching in files

grep

grep is a super powerful Linux command used to search through files for the occurrence of a string of characters that matches a specified pattern. We can use the command in combination with a bunch of different options and flags for efficient searching.

Options and flags

- **-r:** search recursively
- **-w:** match the whole word
- **-n:** only in line number
- **-e:** match pattern
- **--include** and **--exclude:** include and exclude files in the search
- **--include-dir** and **--exclude-dir:** include or exclude directories in the search

Let's take a look at the grep command in action. In the directory */home/user/Downloads* of your virtual machine, a number of files exist. We'll find the files that have the word "vacation" in them, and move them to */home/user/Documents*

Step 1: Find files.

```
grep -rw /home/user/Downloads -e "vacation"
student-04-6ff0b48926b1@linux-instance:~$ grep -rw /home/user/Downloads -e "vacation"
/home/user/Downloads/Japan:We enjoyed our vacation here.
/home/user/Downloads/Iceland:We had a great vacation here.
```

Step 2: Move the directories that match into the target directory.

```
mv /home/user/Downloads/Iceland /home/user/Downloads/Japan /home/user/Documents
student-04-6ff0b48926b1@linux-instance:~$ mv /home/user/Downloads/Iceland /home/user/Downloads/Japan
/home/user/Documents
student-04-6ff0b48926b1@linux-instance:~$
```

Click *Check my progress* to verify the objective.

Search the files and move them

Editing files

Lots of Linux distributions come with pre-installed text editors. The most popular ones are vi and nano, which will be included in nearly every distribution. Other text editors, like Emacs and Gedit, might also be present. In this lab, we'll modify files using the Nano editor.

You can use the *nano* command to open the Nano editor and modify an existing file, or create a new one. To edit an existing file, we'll first start with opening it.

Note: The example commands are for reference only.

Example:

```
nano /path/to/existing/file
```

The command will open the file in the terminal and display the current file contents. To modify, you can edit the content in the terminal, just like a normal editor. The editor is managed using various shortcuts.

To save modifications to the file, use Ctrl+O-:

CTRL-O

Once editing is done, we can close and exit the program using Ctrl+X

CTRL-X

NB: At any point in using the editor, you can get help using Ctrl+G

CTRL-G

To exit help mode, use Ctrl+X

CTRL-X

Alright, now let's **practice** how to edit files using *nano*.

In the current directory, create an empty file called *editor_test.txt*

```
touch editor_test.txt
```

```
student-04-6ff0b48926b1@linux-instance:~$ touch editor_test.txt
```

Open the file with the Nano editor.

```
nano editor_test.txt
```

```
GNU nano 2.5.3                               File: editor_test.txt
[ Read 0 lines ]
^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos
^X Exit          ^R Read File     ^\ Replace       ^U Uncut Text    ^T To Spell     ^_ Go To Line
```

Add content to the file. (In this case, we add five lines, each separated by an empty line.)

```
GNU nano 2.5.3                               File: editor_test.txt
Knock knock
Who's there?
Very long pause...
Java.
:-)
```

```
[ Read 9 lines ]
^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos
^X Exit          ^R Read File     ^\ Replace       ^U Uncut Text    ^T To Spell     ^_ Go To Line
```

Save the file by hitting Ctrl+O

CTRL-O

```
GNU nano 2.5.3                               File: editor_test.txt
Knock knock
Who's there?
Very long pause...
Java.
:-)
```

```
File Name to Write: editor_test.txt
^G Get Help      M-D DOS Format     M-A Append      M-B Backup File
^C Cancel        M-M Mac Format    M-P Prepend    ^T To Files
```

You'll need to confirm the file that you want to write the content to by hitting Enter. After this, exit the program by hitting Ctrl+X

CTRL-X

```
student-04-6ff0b48926b1@linux-instance:~$ nano editor_test.txt
student-04-6ff0b48926b1@linux-instance:~$
```

That's it! You've successfully created and modified a file.

Conclusion

In this lab, we've gone through the basics of creating, modifying, copying, and removing files and folders in Linux. As always, you can learn more about each of the commands we've covered by using the *man* command. Make sure to practice these commands so that you get comfortable using them.

End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you've used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.