Intro to Linux

Intro to Linux

Linux is an operating system just like Mac OSX or Windows. It has a GUI (graphical user interface) just like any other operating system, but most Linux users are familiar with the command line interface. In Linux this is known as the terminal.

In today's lesson you are going to learn some basic command line techniques so that when you start to play with your Nano, you will be able to understand what's going on.

Navigating the File System

You might be familiar with a computer that has files and folders. In Windows and Mac, you can click on folders, open them, and go down into nested folders. Linux is set up similarly, but instead of thinking of them as folders, Linux uses directories.

Directories work like folders, but they are usually accessed from the command line.

You can use commands in the terminal to navigate through this file system. Follow these steps to practice.

1. Open **this** **(https://bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192)** command line simulator.
2. Type `ls` to list all the files in your current directory.
3. Type enter. You should see a list of files.

```
Loading...
```

These files are what is located in your user/home directory. This directory can usually be referenced by the “~”. You can navigate back into this directory at any time.

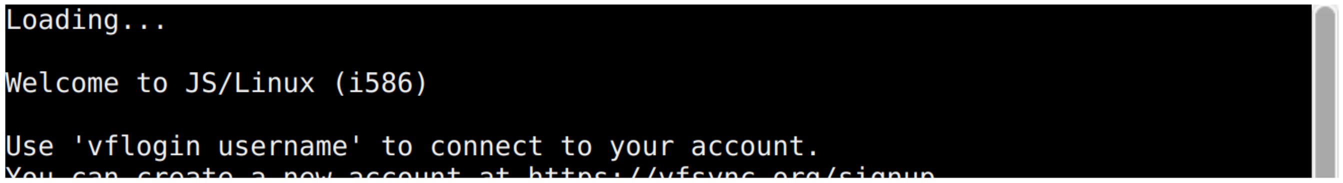
In order to navigate into different directories, you can use the change directory (**cd**) command. This command operates with two sections. The command itself, and then arguments for the command.

For example, look at this command:

The **cd** is the command itself, and the **..** tells the command where to change to. Two dots means to go up a directory.

4. In the terminal type **cd ..** and enter.
5. On the next line, type **ls** to see the files in the new directory.

You should see a list of directories. You were previously in your home directory, and now you are in the root directory. You cannot go any higher in the directory tree from here, but you can navigate down the tree to see any of the files and directories that make up the computer.



```
Loading...

Welcome to JS/Linux (i586)

Use 'vflogin username' to connect to your account.
```

Try this!

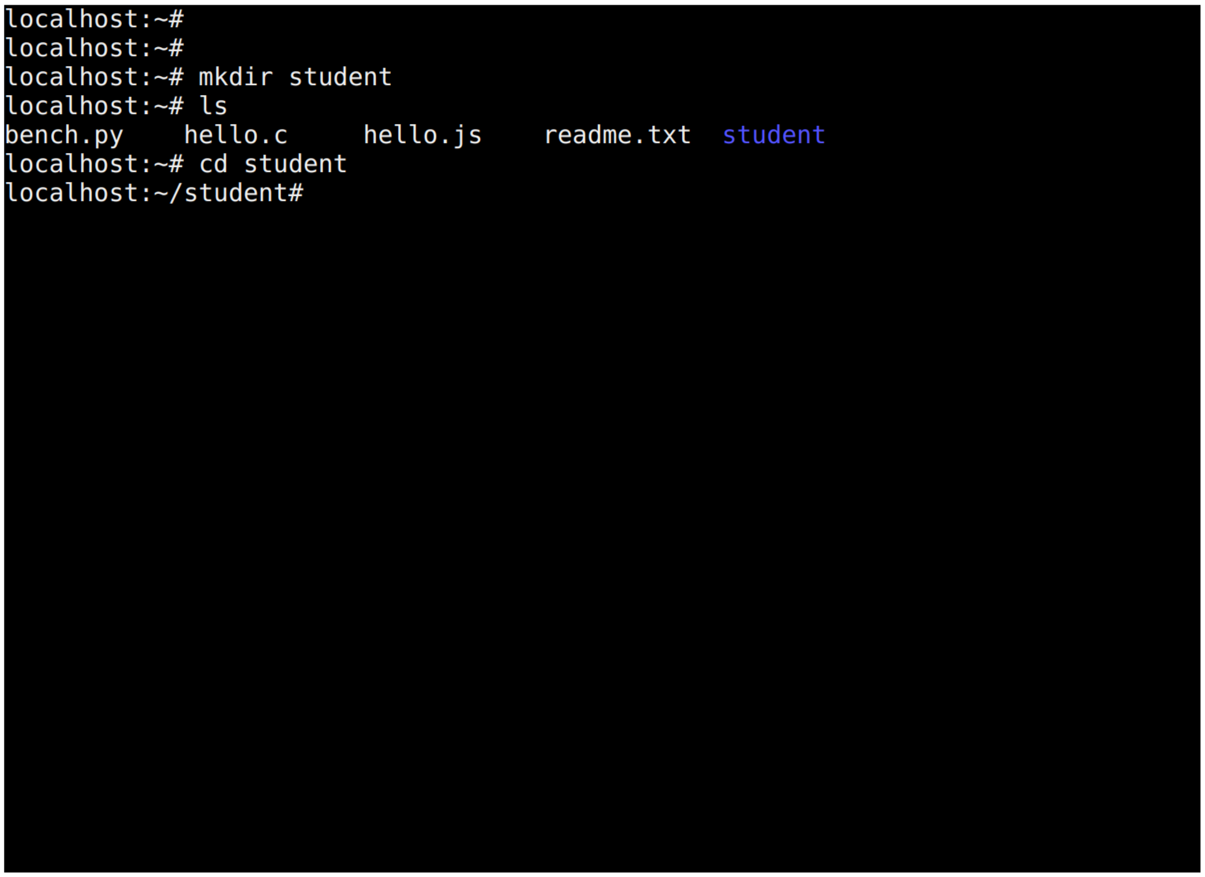
Can you navigate into the var folder?

Making a Directory

In Linux, you can create new directories right in the command line. Using the mkdir command you can create as many directories as you need.

1. Navigate back to your home directory.
2. Type **mkdir student** and hit enter.
3. Type **ls** and hit enter to see your new directory.

4. Change directories into your new directory.



```
localhost:~#
localhost:~#
localhost:~# mkdir student
localhost:~# ls
bench.py    hello.c     hello.js    readme.txt  student
localhost:~# cd student
localhost:~/student#
```

Making New Files

You can also make files right in the command line. Using the **nano** package you can create and edit files from terminal. When you create files in Linux, it is important to make sure that they don't have spaces in them. You can use underscores, or capitalize the first letter of each word to differentiate them.

1. Type **nano new_file.txt** and enter.

```
localhost:~/student# ls
localhost:~/student#
```

2. Add some text into your new file, like the words “hello world”.
3. When you are finished press **Ctrl +X**.
4. Type “y” and hit enter where it asks if you want to save the modified buffer.

Viewing Your Files

In order to see the text you have written, use “cat” to view the file.

1. On the new line, type “**cat new_file.txt**” and enter.
2. You should see the contents of the file on the next line!

Hint: you can use the “tab” key after you start typing the file name to have the computer autofill the rest of the name.

Try this:

Make another text file that contains the names of all your pets. Make sure to give it a name that has no spaces and is descriptive.

Using the Python Shell

This terminal has python already installed, so you can run python commands from within the terminal. Follow these steps to test out the python shell from within the terminal.

1. Type “python3” and enter.
2. In the space with a cursor, use a python print statement to print hello world.
3. Press enter.

4. Using the python shell, create two variables and print out the result of them added together.

5. Press **Ctrl + D** to kill the python shell.

Try This:

Can you open the python shell and multiply two variables together?

Using Nano To Make Python Files

1. Using commands make another file called **helloworld.py**
2. In that file, create a python program. It can be anything you want.
3. When you are finished, press **Ctrl+X** to exit.
4. Type **Y** when asked if you want to save the program.

5. In your command line use **python3 helloworld.py** to run your python code.

Try This!

Can you create a choose your own adventure story with three choices and then run it using the command line?

Rename and Move Files

While using Linux you may also need to rename or move files around.

1. Create a new file called **my_cats.txt**
2. Inside it, create a grocery list.
3. Exit nano and save your file.

Now you are going to rename your file so that it reflects what is inside it. The mv (move) command can be used to rename your files.

4. In terminal type **mv my_cats.txt groceries.txt**
5. Then use **ls** to see your changes.

Try this!

Create a new directory called "**favorites**" in your student directory. Inside the new directory create another text file called favorite_foods.txt and add your favorite foods into it.

Move Files

You can also use the **mv** command to move files.

1. If you haven't already, create another directory inside **"student"** called "**favorites**".
2. Type **mv groceries.txt favorites**
3. Use "**ls**" to see if your file has moved.

Let's say you need a copy of groceries in both folders. You can use the copy command to copy files from one location to another.

4. Navigate into the favorites directory.
5. Inside your student directory, type **cp groceries.txt ../**
6. Use "**ls**" to see if your command worked.

Removing Files

You can also use commands to delete files.

1. Make a new directory called “**junk**”.
2. Inside that directory, make a file called “**trash.txt**”.

3. Put some random text in it.

4. Save it and use **ls** (or **cat**) to see the file.

5. Now it’s time to delete it. In the next line, type **rm trash.txt** .

Deleting Multiple Files

You can also delete multiple files with one command. You are going to use the wildcard to help delete files based on their extension.

1. Using **nano**, create a few files in your junk directory. They can be text files, python files, or end with any other file extension for this exercise.
2. Use “**ls**” to show your files.

3. Now, use rm to remove all the txt files like this “**rm *.txt**”
4. Use “**ls**” to see the results.

The ***** acts as a wildcard, and removes any file with the ending "**.txt**".

Deleting Directories

Now that you have so much practice deleting files, it's time to learn about deleting directories.

1. Create a directory in **junk** called stuff.
2. Use any command that you know to remove this directory.

You probably have noticed that the command line is giving you an error. This is because the rm command only works for files, nor directories. In order to delete directories, you will need to use rmdir.

3. To remove the **stuff** file, type `rmdir stuff` .
4. Use `ls` to see if the directory has been removed.

Now you know that works for empty directories. What about directories with stuff in them?

5. Change directories out of junk and back into your student directory.
6. Try to remove the junk directory.

You probably got an error about how junk is not empty. In order to delete a directory with stuff in it, you will need to use the **-r** argument with **rm**.

7. Type `rm -r junk` and enter.
8. Use `ls` to see if your command worked.

You need to be careful with using **-r,** with the **rm** command, because it will remove everything in that directory recursively, and you cannot undo it.

Recap

Now you have an idea of how to do some basics in Linux using the command line. There are tons of commands that you can use in the command line, but these are some of the most basic and useful ones to know as you begin your linux journey.