HackerFrogs Afterschool
Beginner Cybersecurity Skills

Linux Operation Basics /w TryHackMe:
Linux Fundamentals Part 2

Class:	Linux OS Operations
Workshop Number:	AS-LIN-03
Document Version:	1.0
Special Requirements:	Registered account at tryhackme.com

Table of Contents

Introduction

Welcome to HackerFrogs Afterschool! HackerFrogs Afterschool is a a series of online workshops meant to teach cybersecurity skills and concepts to beginners who may not be familiar with the field of cybersecurity. This lesson is the final introductory lesson on Linux OS (operating system) operations, which is an essential skill set, due to the heavy preference of Linux OS in the cybersecurity field, and among ethical hackers in particular. This lesson will cover the following learning objectives:

Learning Objectives:

- Learn about Linux file management commands
- Learn about Linux file permissions
- Learn about common Linux directories

Brief Description of Capture The Flag (CTF) Games and TryHackMe

Capture the Flag (CTF) games are training exercises in the field of cybersecurity. The goal of a CTF exercise is to "capture the flag" through use of cybersecurity skills. In this context, "capture" means to gain access to a file or other resource, and "flag" refers to a secret phrase or password.

The CTF platform we will be playing to learn basic Linux skills is called TryHackMe, which is a popular cybersecurity education platform, with many education modules (called rooms) which span a wide range of cybersecurity topics.

Access to the TryHackMe room that is featured in this workshop requires a valid user account, so if we do not already have one, we can obtain one from the following link:

https://tryhackme.com/signup/

Workshop Guide Format Notes

When following instructions in this document, there will be instructions to type specific commands in the CLI window or in a search bar. These commands will be displayed in red, and will have a different `font` applied to them, as well as a light grey background color. For example:

```
this is a sample command text line
```

If there are code snippets in this document, they look similar to the format above, except that the text will be black. For example:

```
this is a sample code snippet line
```

Pre-Learning with JS Linux

Although TryHackMe is a great learning platform, there are some limitations for free users: a strict 1-hour time limit for the interactive Linux machine called the “AttackBox”. To maximize our time, we'll first use another service to learn and practice this workshop's Linux skills before starting the TryHackMe room.

Step 1

In a web browser, navigate to the following URL:

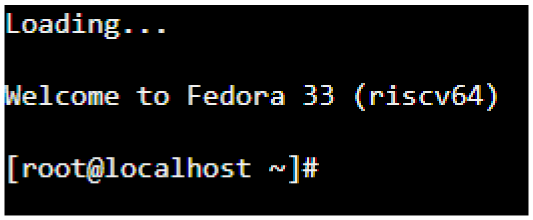
https://bellard.org/jslinux/vm.html?cpu=riscv64&url=fedora33-riscv.cfg&mem=256

NOTE: If for some reason the link doesn't work, navigate to this page

https://bellard.org/jslinux/index.html

From that page, click on any of the **Startup** links that correspond with a version of Linux with a **Console** User Interface.

When the JS Linux machine is loaded, it should look similar to this:



```
Loading...

Welcome to Fedora 33 (riscv64)

[root@localhost ~]#
```

Step 2

First we'll learn about the - **-help** flag. In the JS Linux console, enter the following command:

```
cat --help
```

In the output, we can see a short description of the command and the different ways to modify the output using command flags.

Step 3

Next, we'll learn about the a more-detailed way to learn about Linux commands: the **man** command

Enter the following command

```
man cat
```

This provides a much more in-depth description of each command and its various flags and switches. While running the **man** command we can use the up and down arrows to scroll through the text.

Exit out of the command by pressing the Q key.

Step 4

We'll now learn about the **&** operator, which is used to run commands as background processes. Enter the following command:

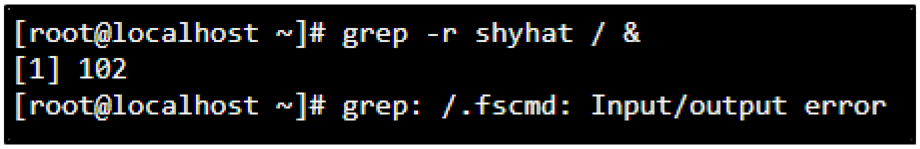
```
grep -r shyhat /
```

This command searches the entire filesystem for any files containing the string **shyhat**, and it's likely to take a very long time to complete. When running commands that are likely to take a long time to complete, it's very useful to use the & operator to run the process in the background so we can still use the terminal while the process is running.

Use the **Ctrl + C** keyboard shortcut to cancel out of the command. Then run the same command, only with the & operator appended to the end:

```
grep -r shyhat / &
```

We should see output similar to this:

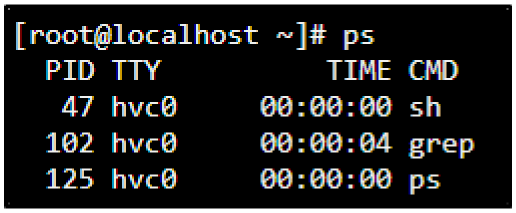


```
[root@localhost ~]# grep -r shyhat / &
[1] 102
[root@localhost ~]# grep: /.fscmd: Input/output error
```

In this example, the grep command has been assigned a process ID number of 102. We can confirm this with the following command:

```
ps
```

Which will result in output similar to this:



```
[root@localhost ~]# ps
  PID TTY          TIME CMD
   47 hvc0     00:00:00 sh
  102 hvc0     00:00:04 grep
  125 hvc0     00:00:00 ps
```

This command shows all of the currently running processes in the system, and we can see that the **PID 102** corresponds with the **grep** command we ran as a process.

Step 5

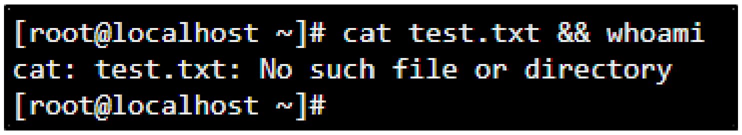
Now we'll learn about the && operator. This operator is instructs the system to run the command that follows if the previous command completes without errors. For example, run the following command:

```
pwd && whoami
```

The **whoami** command is used to output the name of the logged-in user. We see that both commands executed one after the other. However, when using the && operator, if the first command doesn't complete without errors, the second command doesn't run. For example:

```
cat test.txt && whoami
```

We should see the output similar to the following:



```
[root@localhost ~]# cat test.txt && whoami
cat: test.txt: No such file or directory
[root@localhost ~]#
```

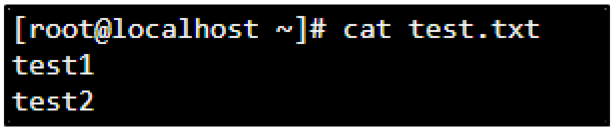
The **whoami** portion of the command doesn't run because the first command didn't complete without errors.

Step 6

Next, we'll cover the > and >> file redirection operators. When included in a command, these operators will redirect the output of the command into a file. The > operator will overwrite the file if the file already exists, while the >> operator will append to the end of the file if the file already exists. For example, the following command will create two lines of text:

```
echo test1 >> test.txt && echo test2 >> test.txt
```

If we read the file with the **cat** command, we will see the following:

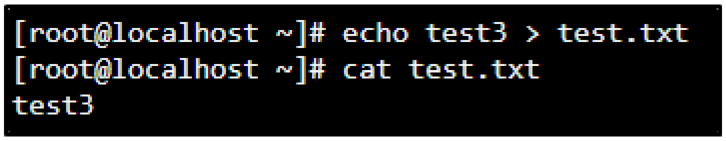


```
[root@localhost ~]# cat test.txt
test1
test2
```

We redirected the output of the echo commands into the test.txt file, and since we used the >> operator for both, the results were appended to each other. However, if we use the > operator for the next command, then read it:

```
echo test3 > test.txt
cat test.txt
```

We see the following result:



```
[root@localhost ~]# echo test3 > test.txt
[root@localhost ~]# cat test.txt
test3
```

The other lines in the **test.txt** file no longer appear because redirection with the > operator overwrites file names if they already exist.

Step 7

The **whoami** command outputs the name of the current user. Enter the following command:

```
whoami
```

And you should see that we're the root (superuser) user.

Step 8

The **touch** command creates empty files. For example:

```
touch touch_example
```

This creates an empty file named **touch_example**.

Step 9

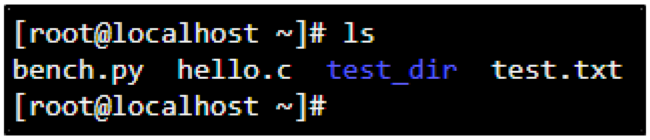
We can create directories with the **mkdir** command:

```
mkdir test_dir
```

We can check if the command executed successfully using the **ls** command:

```
ls
```

The output should look similar to this:



```
[root@localhost ~]# ls
bench.py  hello.c  test_dir  test.txt
[root@localhost ~]#
```

We see **test_dir** is included along with the other files, so our command executed successfully.

Step 10

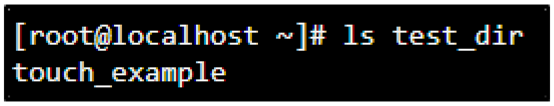
The next command we'll learn, **cp**, is used to copy files to other directories. We can use the following command to copy the **touch_example** file to the **test_dir** directory:

```
cp touch_example test_dir
```

We can then check the contents of the directory with the **ls** command:

```
ls test_dir
```

The results should look like the following:



```
[root@localhost ~]# ls test_dir
touch_example
```

So we see the file was copied successfully.

Step 11

The next command, **mv**, is very similar to the **cp** command, but the file that is copied does not remain the directory it is copied from. We'll move the **touch_example** file to the **test_dir** directory (overwriting it):

```
mv touch_example test_dir
```

We use the **ls** command to confirm that the file is no longer in the current directory:

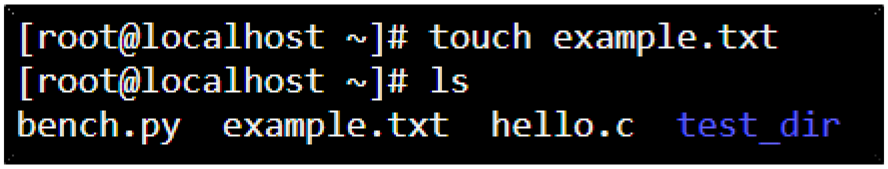
```
ls
```

Then **ls** again to check if the file is in the directory we specified:

```
ls test_dir
```

Another common use of the **mv** command is to rename files. First we'll use the **touch** command to create a file named **example.txt**, then confirm its creation with **ls**.

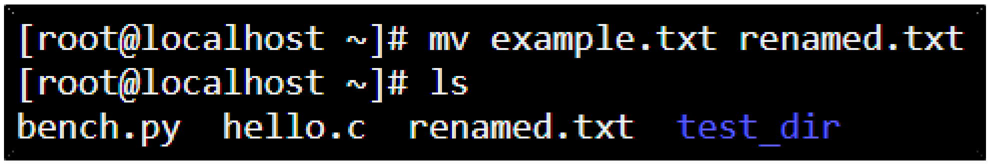
```
touch example.txt
ls
```



```
[root@localhost ~]# touch example.txt
[root@localhost ~]# ls
bench.py  example.txt  hello.c  test_dir
```

This confirms that the file was created. Next, we'll rename the **example.txt** file with the **mv** command:

```
mv example.txt renamed.txt
ls
```



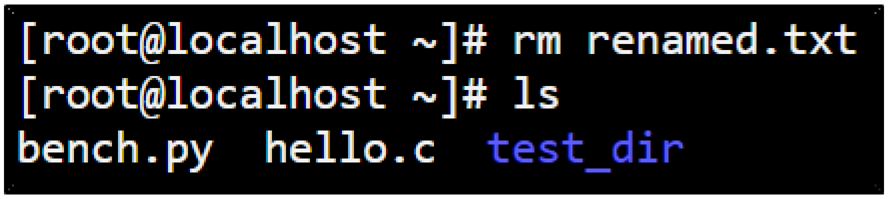
```
[root@localhost ~]# mv example.txt renamed.txt
[root@localhost ~]# ls
bench.py  hello.c  renamed.txt  test_dir
```

We see that the **example.txt** file has been renamed to **renamed.txt**.

Step 12

The last regular command we'll cover in this section is the **rm** command, which is used to delete files or directories. First, we can use **rm** to delete the **renamed.txt** file, then check the directory contents with **ls**.

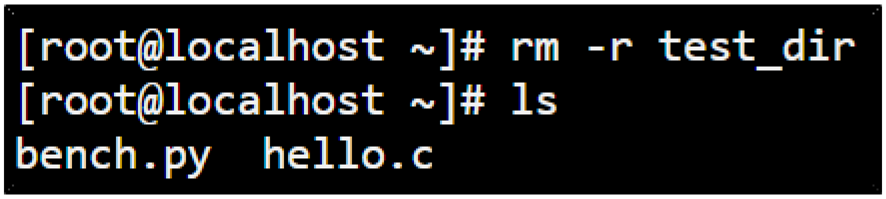
```
rm renamed.txt
ls
```



```
[root@localhost ~]# rm renamed.txt
[root@localhost ~]# ls
bench.py  hello.c  test_dir
```

Lastly, we'll delete the **test_dir** using rm. In order to delete directories with **rm**, we must use the **-r** switch with the command.

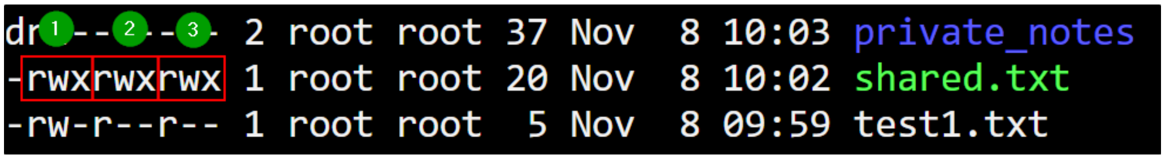
```
rm -r test_dir
ls
```



```
[root@localhost ~]# rm -r test_dir
[root@localhost ~]# ls
bench.py  hello.c
```

A Brief Introduction to Linux File Permissions

Before we move on to the TryHackMe room, we'll introduce the concept of Linux OS file permissions. Simply put, all files and directories in a Linux filesystem have different read, write, and executable permissions depending on the user account accessing them. With the exception of the root superuser, no user has access to all of the files on any Linux filesystem. Look at the following screenshot:



```
dr--r--r-- 2 root root 37 Nov  8 10:03 private_notes
-rwxrwxrwx 1 root root 20 Nov  8 10:02 shared.txt
-rw-r--r-- 1 root root  5 Nov  8 09:59 test1.txt
```

This is output of an **ls -l** command. It lists out the file permissions of these 3 files. There are ten spaces on the left-hand side of this output for file, the very first space either contains a dash (-) or a letter **d**. The letter **d** indicates a directory, otherwise it indicates a regular file. The other nine spaces indicate file / directory permissions. In the screenshot, the first three spaces, as indicated by the green 1, are the owner permissions, the second three spaces, as indicated by the green 2, are the group permissions, and the last three spaces, as indicated by the green 3, are the global permissions. A letter **r** in a space indicates that the file can be read. A letter **w** in a space indicates that the file can be written to. And a letter **x** in a space indicates that the file can be executed (opened or run). In addition, there are two names associated with each file. The first name is the file owner, and the second name is the group owner.

Now that we've coverd most of the material we'll encounter in the TryHackMe room, let's proceed there.

Part 0 – Accessing The TryHackMe Module

Step 1

Login to the TryHackMe website.

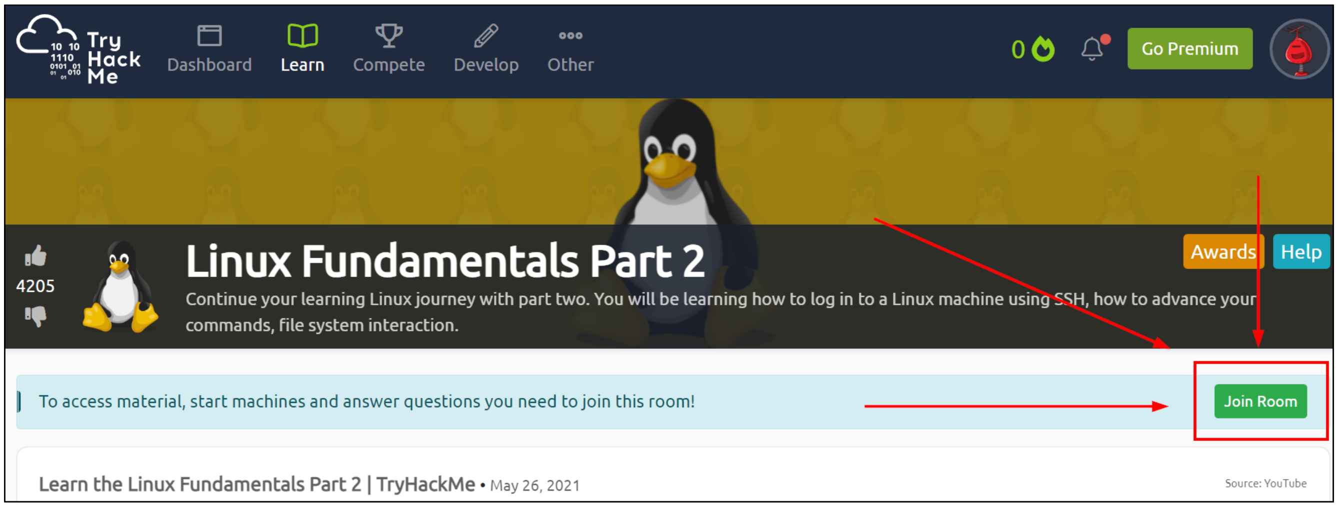
Step 2

Navigate to the following URL:

https://tryhackme.com/room/linuxfundamentalspart2

Step 3

Click the green **Join Room** button located inside the light-blue text box underneath the **Linux Fundamentals Part 2** banner at the top of the webpage. See screenshot below(relevant button outlined in red):



Linux Fundamentals Part 2

Continue your learning Linux journey with part two. You will be learning how to log in to a Linux machine using SSH, how to advance your commands, file system interaction.

To access material, start machines and answer questions you need to join this room!

Learn the Linux Fundamentals Part 2 | TryHackMe • May 26, 2021

Part 1 – Introduction

Step 1

Under the **Task 1** header, click on the white **Completed** button located under the **Let's proceed** text.

NOTE: There will be much less explanation in this document than usual HackerFrogs Afterschool workshop guides, because the task text provides good descriptions and explanations.

Part 2 – Accessing The Linux Machine Using SSH

Step 1

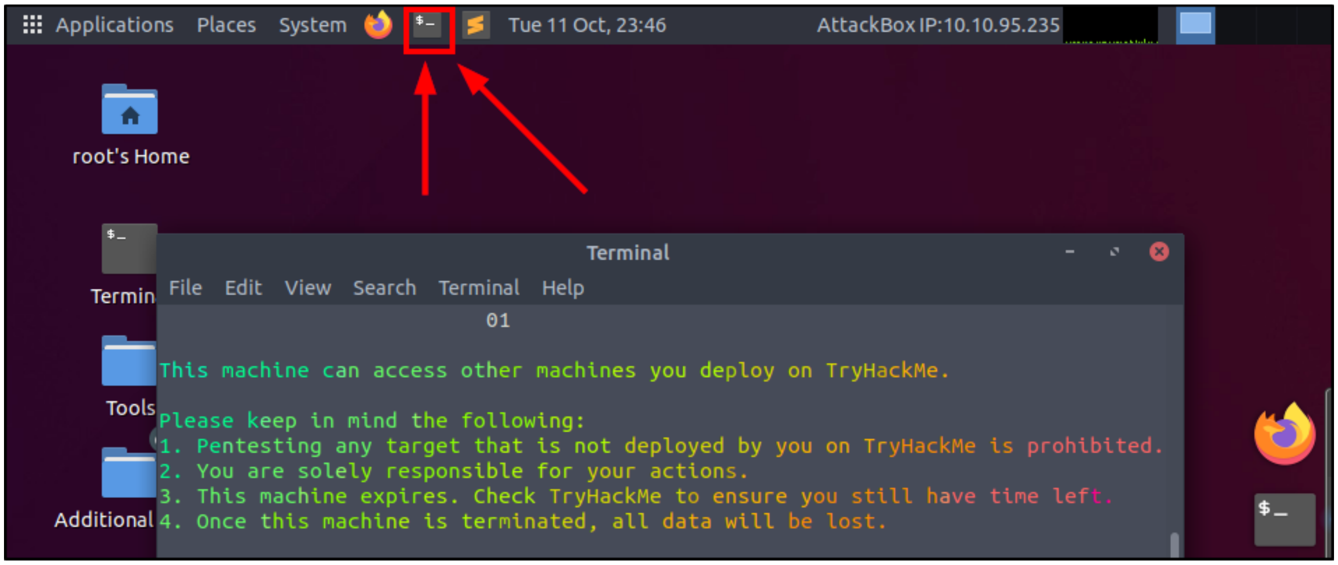
Under the **Task 2** header, click on the green **Start Machine** button located in the top-right portion under the **Task 2 Accessing Your Linux Machine Using SSH (Deploy)** heading.

Step 2

Click on the blue **Start AttackBox** button located at the top of the webpage, to the right of the **Linux Fundamentals Part 2** banner.

Step 3

Once the AttackBox has finished loading, click on the black terminal shortcut button located to the right of the System tab on the desktop (button outlined in red below):

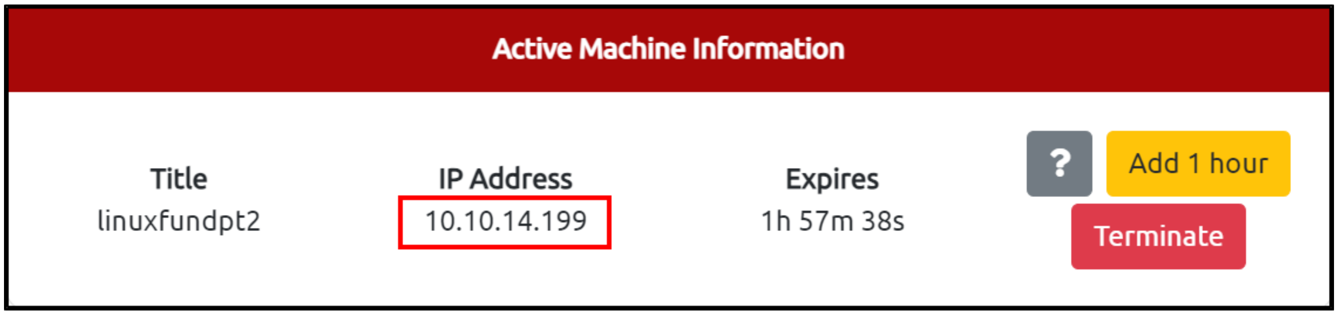


```
                    01

This machine can access other machines you deploy on TryHackMe.

Please keep in mind the following:
1. Pentesting any target that is not deployed by you on TryHackMe is prohibited.
2. You are solely responsible for your actions.
3. This machine expires. Check TryHackMe to ensure you still have time left.
4. Once this machine is terminated, all data will be lost.
```

Then close the **Terminal** window with the rainbow text.

Step 4

Take note of the IP address under the red **Active Machine Information** header near the top of the webpage. We will use this IP address to login to the Linux Fundamentals VM using SSH. See the screenshot below (example IP address outlined in red):



Active Machine Information

Title	IP Address	Expires
linuxfundpt2	10.10.14.199	1h 57m 38s

Note that the above IP address is an example only, and the IP address of your own active machine will mostly likely be different.

Step 5

In the newly opened AttackBox terminal window, type the following SSH command, but remember to insert your own IP address that we retrieved from the previous step in the place of <IP_ADDRESS> :

```
ssh tryhackme@<IP_ADDRESS>
```

When asked if you want to continue connecting, type the following response:

```
yes
```

Finally, when asked for the password, enter the following:

```
tryhackme
```

NOTE: There will be no visual output as you type in the password, this is normal Linux OS behavior, and we will need to type out the password without visual feedback.

Step 6

At the bottom of the **Task 2** section, under the **I've logged into the Linux Fundamentals Part 2 machine using SSH** text, click on the white **Completed** button.

Part 3 – Introduction to Flags and Switches

Step 1

Under the Task 3 header, click on the white **Completed** button located below the **Explore the manual page of the ls command** text.

Step 2

Under the **What directional arrow key would we use to navigate down the manual page** question, enter the following answer:

```
down
```

Then click on the white **Submit** button to the right of the question field.

Step 3

Under the **What flag would we use to display the output in a “human-readable” way** question, enter the following answer:

```
-h
```

Then click on the white **Submit** button to the right of the question field.

Part 4 – Filesystem Interaction Continued

Step 1

Under the **Task 4** header, and under the **How would you create the file named “newnote”** question, enter the following answer:

```
touch newnote
```

Then click on the white **Submit** button to the right of the question field.

Step 2

Under the **On the deployable machine, what is the file type of “unkown1” in “tryhackme's” home directory** question, enter the following answer:

```
ASCII text
```

Then click on the white **Submit** button to the right of the question field.

Step 3

Under the **How would we move the file “myfile” to the directory “myfolder”** question, enter the following answer:

```
mv myfile myfolder
```

Then click on the white **Submit** button to the right of the question field.

Step 4

Under the **What are the contents of this file** question, enter the following answer:

```
THM{FILESYSTEM}
```

Then click on the white **Submit** button to the right of the question field.

Step 5

Click on the white **Completed** button located below the **Continue to apply your knowledge and practice the commands from this task** text.

Part 5 – Permissions 101

Step 1

Under the **Task 5** header, and under the **On the deployable machine, who is the owner of “important”** question, enter the following answer:

```
user2
```

Then click on the white **Submit** button to the right of the question field.

Step 2

Under the **What would the command be to switch to the user “user2”** question, enter the following answer:

```
su user2
```

Then click on the white **Submit** button to the right of the question field.

CONTEXT

The **su** command when used with a valid username on the system allows that user account to be accessed, provided that the user's password is known.

Step 3

Click on the white Completed button located below the **Now switch to this user “user2” using the password “user2”** text.

Step 4

In the AttackBox terminal, change users to the **user2** account by inputting the following command:

```
su user2
```

When prompted, enter **user2** as the password.

Then read the file named **important**:

```
cat important
```

Under the **Output the contents of "important", what is the flag** question, enter the following answer:

```
THM{SU_USER2}
```

Then click on the white **Submit** button to the right of the question field.

Part 6 – Common Directories

Review the materials under the **Task 6** header.

Step 1

Under the **Task 6** header, click on the white Completed button located below the **Read me** text.

Step 2

Under the **What would the command be to switch to the user "user2"** question, enter the following answer:

```
/var/log
```

Then click on the white **Submit** button to the right of the question field.

Step 3

Under the **What root directory is similar to how RAM on a computer works** question, enter the following answer:

```
/tmp
```

Then click on the white **Submit** button to the right of the question field.
Step 4

Under the **Name the home directory of the root user question**, enter the following answer:

```
/root
```

Then click on the white **Submit** button to the right of the question field.

Step 5

Click on the white Completed button located below the **Now apply your learning and navigate through these directories on the deployed Linux machine** text.

Part 7 – Conclusions and Summaries

Step 1

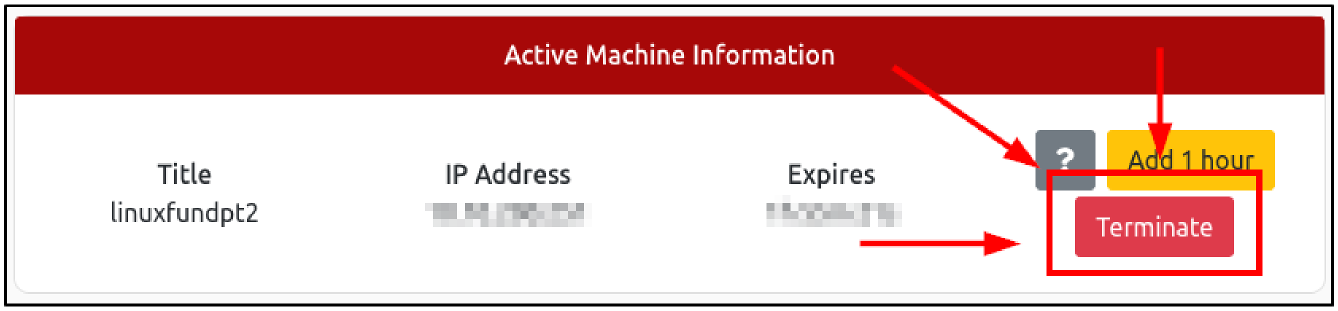
Under the **Task 7** header, read the summary for the module, then click on the white Completed button located below the **Proceed to the next task to continue your learning** text.

Part 8 – Linux Fundamentals Part 3

The last thing we need to do to complete the room is to complete last two questions.

Step 1

At the top of the webpage, under the **Active Machine Information** banner, click on the red **Terminate** button located on the right-hand side of the box (see screenshot below):



Active Machine Information

Title	IP Address	Expires
linuxfundpt2	[redacted]	[redacted]

Under **Task 8** header, click on the white **Completed** button under the **Terminate the machine from task 2** text.

Step 2

Under **Task 8** header, click on the white **Completed** button under the **Join Linux Fundamentals Part 3** text.

Summary

While working through the TryHackMe Linux Fundamentals Part 2 room, we learned to use the following commands to work inside of the Linux OS file system:

`man`	<-- this command is used to retrieve the manual document for a specified command
`whoami`	<-- this command is used to output the current user's account name
`touch`	<-- this command is used to create an empty file
`mkdir`	<-- this command is used to create directories
`cp`	<-- this command is used to copy files from one directory to another
`mv`	<-- this command is used to move files from one directory to another
`rm`	<-- this command is used to delete files from the system

We also learned about file permissions on the Linux OS, namely that each user account on the system only has read and / or write access to a portion of the files on the system, with the exception of the superuser, root, who has access to all files.

Lastly, we learned about the following common Linux directories

`/etc`	<-- this directory contains system files (such as user passwords)
`/var`	<-- this directory contains log files and database files
`/root`	<-- this directory is the root user's home directory (typically the most secure directory)
`/tmp`	<-- this directory is for temporary files and its contents are deleted regularly

Extra Credit

This is the last workshop for Linux OS Operations, but if we want to practice and learn more about working inside the Linux OS, we can work through part 3 of the Linux Fundamentals rooms, which can be found at the following URL:

https://tryhackme.com/room/linuxfundamentalspart3

There are also a number of other basic Linux courses that are available for free online:

Linux Journey

https://linuxjourney.com/

NOTE: All of the lessons under the Grasshopper header should be appropriate for beginner-level users.

Udemy – Learn the Linux Command Line: Basic Commands

https://www.udemy.com/course/command-line/

Until next time, HackerFrogs!