CY 2550 Foundations of Cybersecurity

Intro to Linux

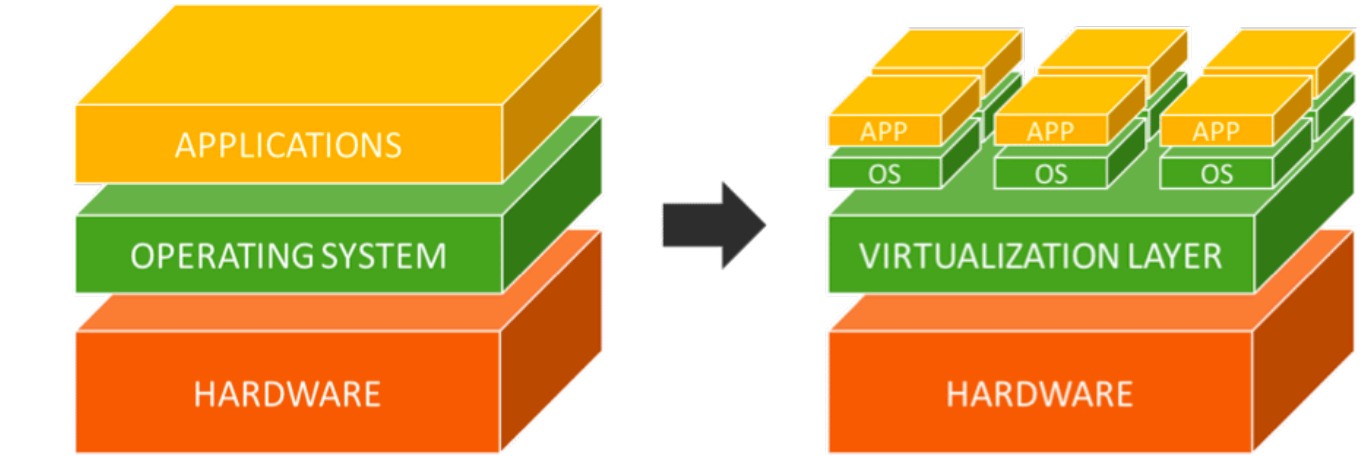
Alina Oprea
Associate Professor, Khoury College
Northeastern University

Required Software

- Linux ISO File
  - Ubuntu (https://www.ubuntu.com/download/desktop)
- Virtualization Software
  - Oracle VirtualBox (http://www.virtualbox.org/wiki/Downloads)
  - VMware Workstation (https://www.vmware.com/products/workstation-pro.html)

Virtualization

- The process of running a virtual instance of a computer system in a layer abstracted from the actual hardware.
- Provides isolation between Virtual Machines (VMs) mediated by the hypervisor

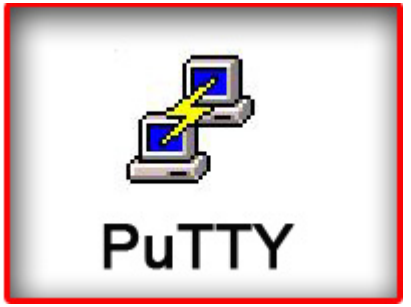


Different Ways to Learn Linux

- Install it as the host OS on your computer
- For Windows users, Windows Subsystem for Linux (WSL) is a good option to run the bash terminal on your Windows machine
- Virtual Machines – install Linux on a virtual machine, a computer within a computer

Different Ways to Learn Linux

- For the Mac, you can use your default terminal with Homebrew
- PuTTY, an SSH client for Windows users, so you can login to the CCIS Server from your laptop



Resources

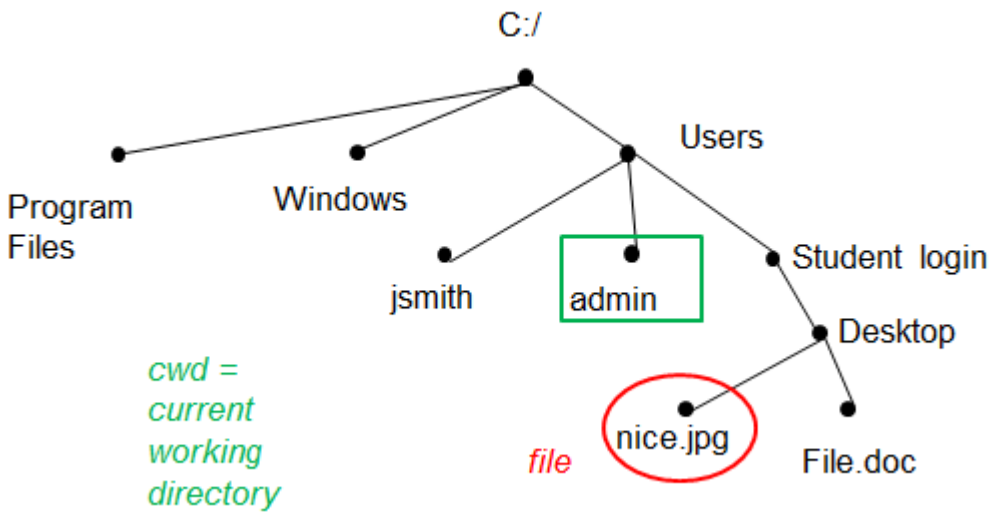
- https://cbw.sh/linuxbasics/
  - Tutorial written by Martin Petrauskas
- http://linuxcommand.org/
- Linux Command Line by W. Shotts:
  - http://linuxcommand.org/tlcl.php
- The man command in Linux

Installing Necessary Programs

- We need to install some miscellaneous programs that you will need throughout the semester
- We will install vim, emacs, pip, python, ruby, perl, and git
- **sudo apt-get install python-pip vim emacs ruby perl git**

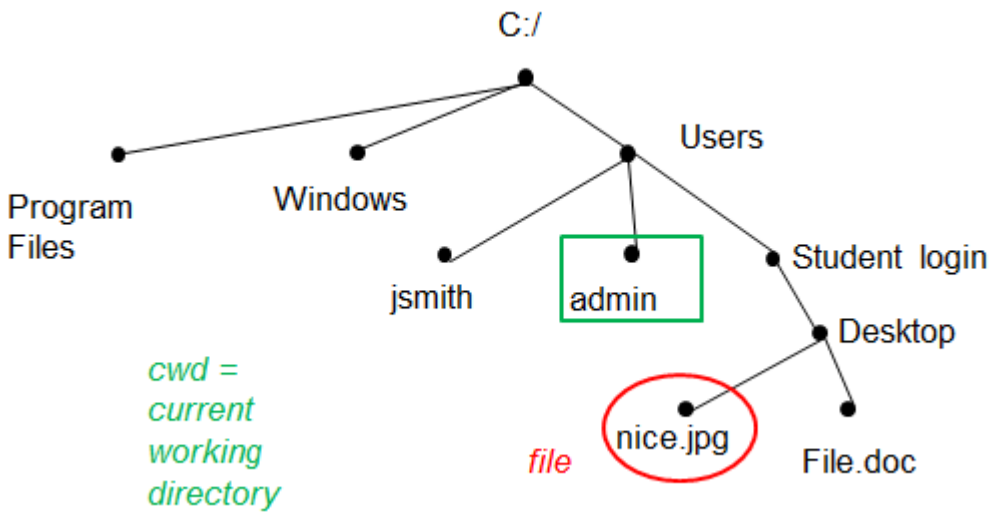
Directories and File Systems

- Root directory – top most directory in a file system ("C:/" for Windows, "/" for Unix/Mac)
- Home directory – directory for a specific user in a file system ("C:/Users/<USERNAME>" for Windows "/home/<USERNAME>" for Unix/Mac



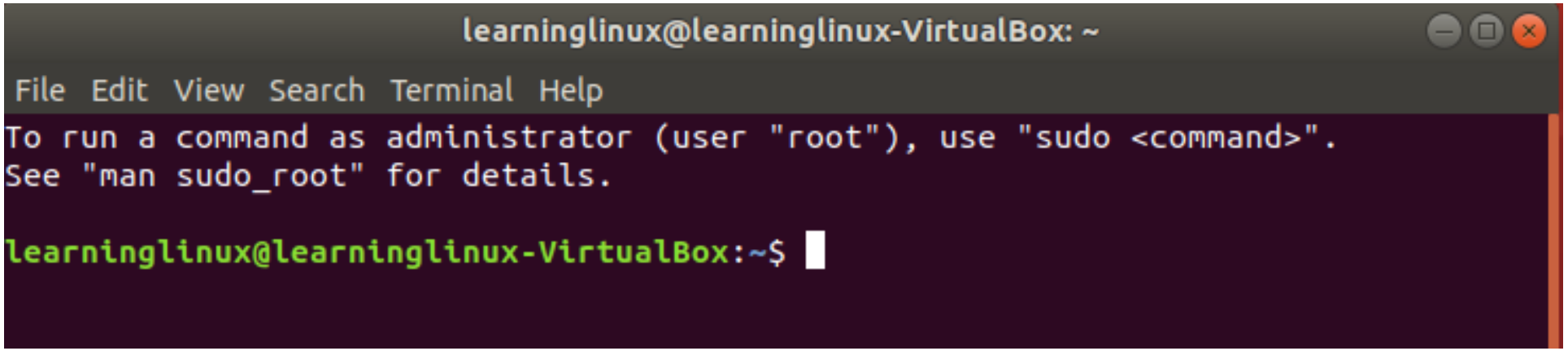
Filepaths

- Two types:
  - Absolute – always starts from the root directory
  - Relative – starts from the current working directory



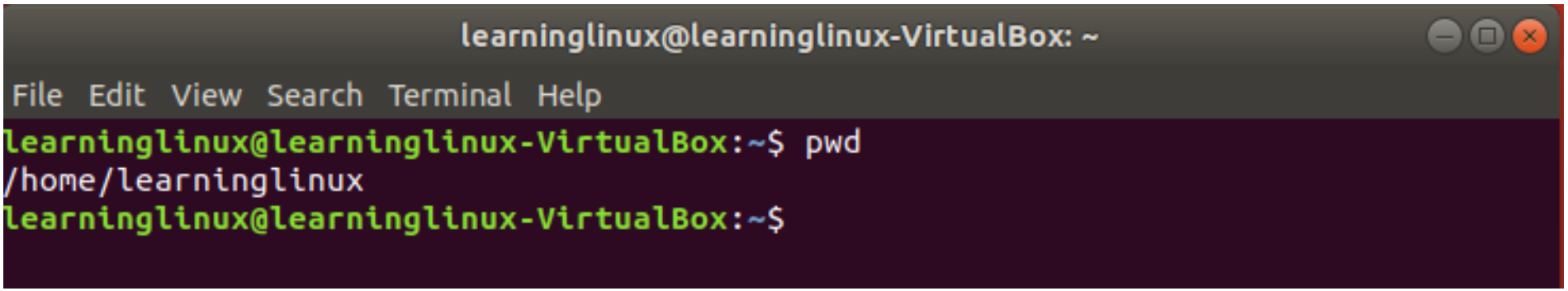
The Terminal

- Here is a breakdown of what we see in the terminal when we start it up
- learninglinux – username of the current person using the computer
- learninglinux-VirtualBox – the name of the computer
- ~ represents the filepath of the home directory
- $ prompt symbol



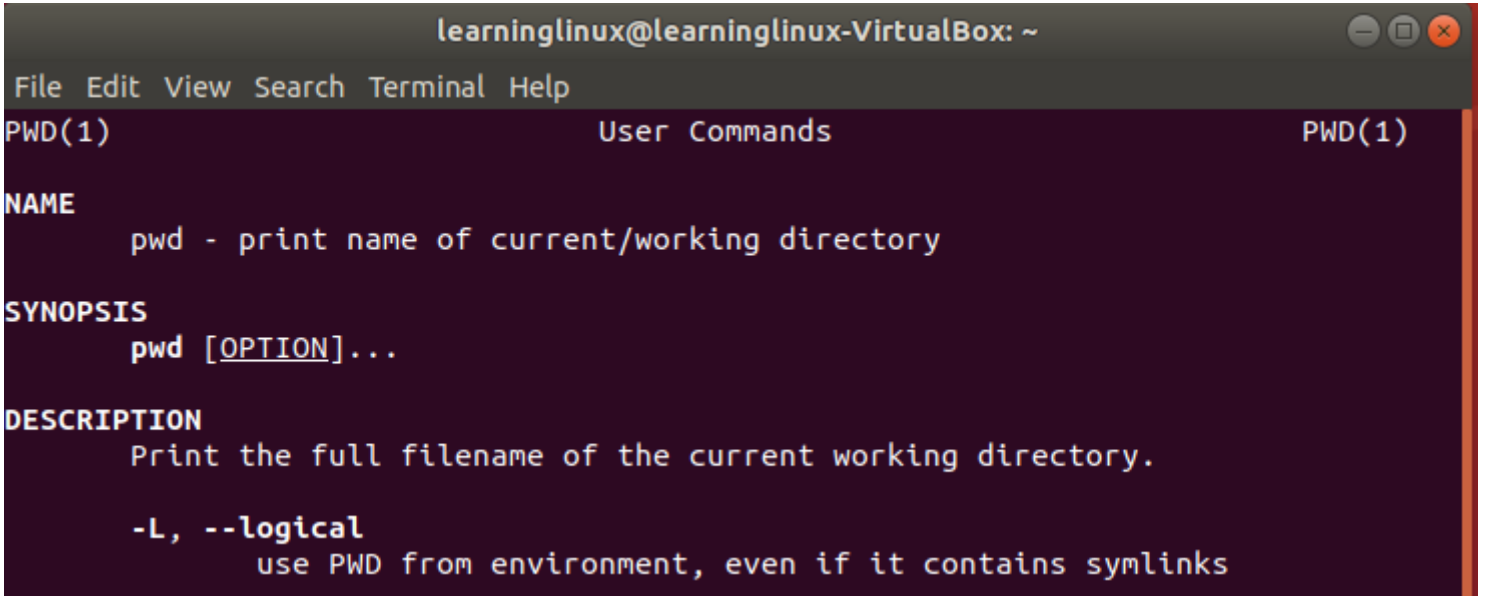
Print Working Directory (pwd)

- The **pwd** command will print the filepath of your current working directory



Manual Pages (man)

- The **man** command will display information about the given command
- The syntax is **man [command]**

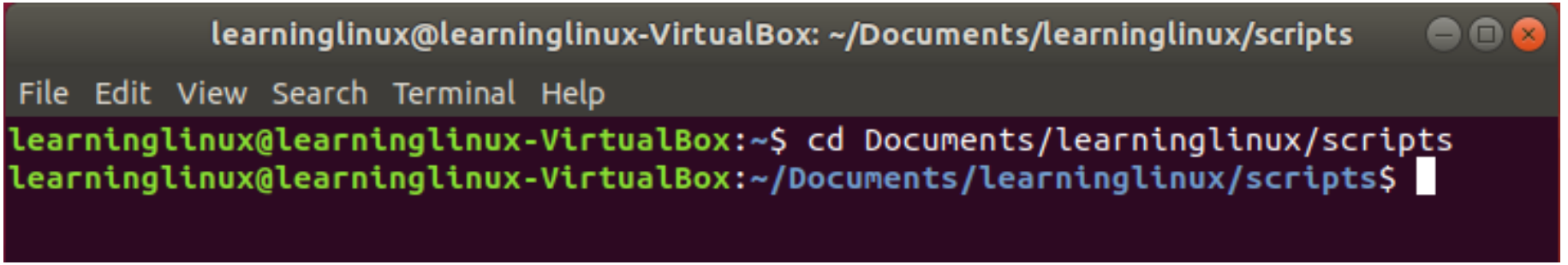


Change Directory (cd)

- The **cd** command allows you to move around the file system between all the different types of directories
- By default, typing in **cd** in the terminal will take you back to your home directory

Change Directory (cd)

- The syntax for this command is **cd [directory]**
- To change to a different directory, you must specify which directory you want to go to
- You must give an absolute filepath of relative filepath of the directory



List Segments (ls)

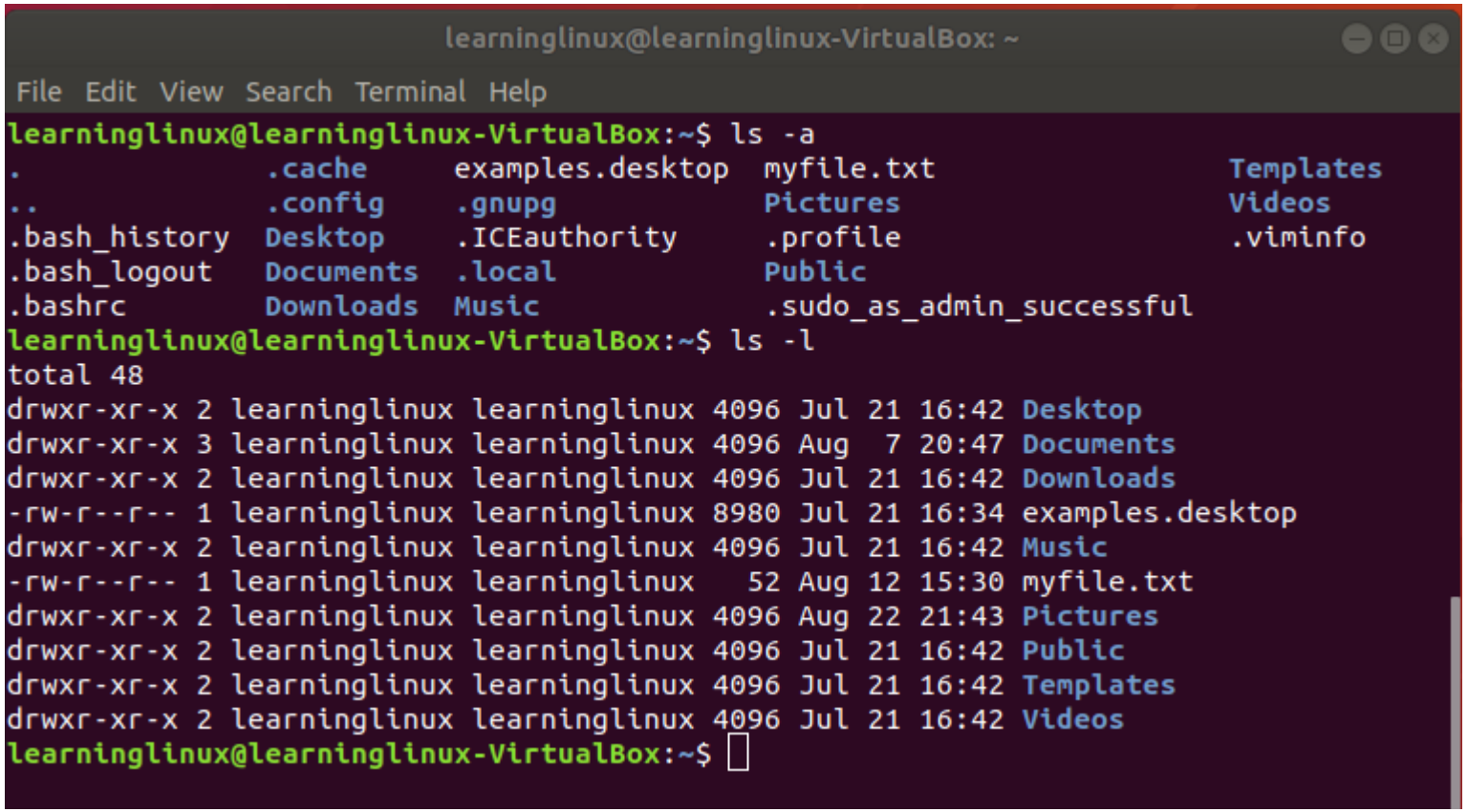
- The **ls** command lets you view the files/directories in the current working directory
- Two optional arguments you should know
  - **a** – will show ALL the files in the current working directory, including hidden files
  - **l** – will show the files with more specific information in long format

List Segments (ls)

Color	File Type
White	Regular Text File
Blue	Directory
Green	Executables or Scripts
Pink	Images
Cyan	Links (shortcuts)
Red	Archives

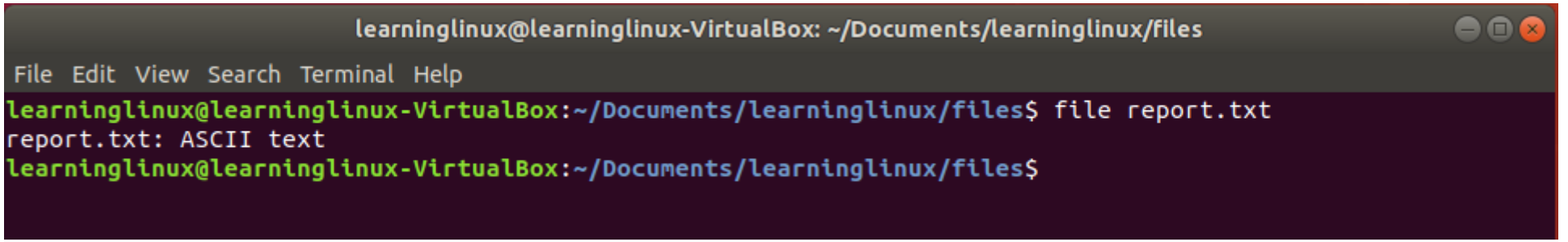
List Segments (ls)

- The syntax is **ls [options]**



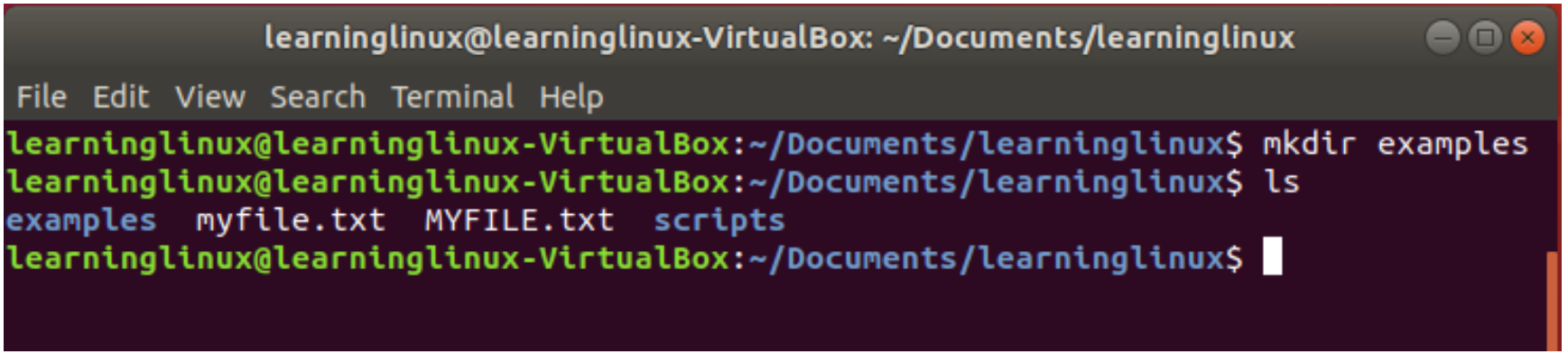
Making Files (touch, vim)

- There are multiple ways to make a new file
- **touch [filename]**
- This will create a new text file by default with the name filename
- **vim [filename]**
- Use the **file** command to see the the type of file



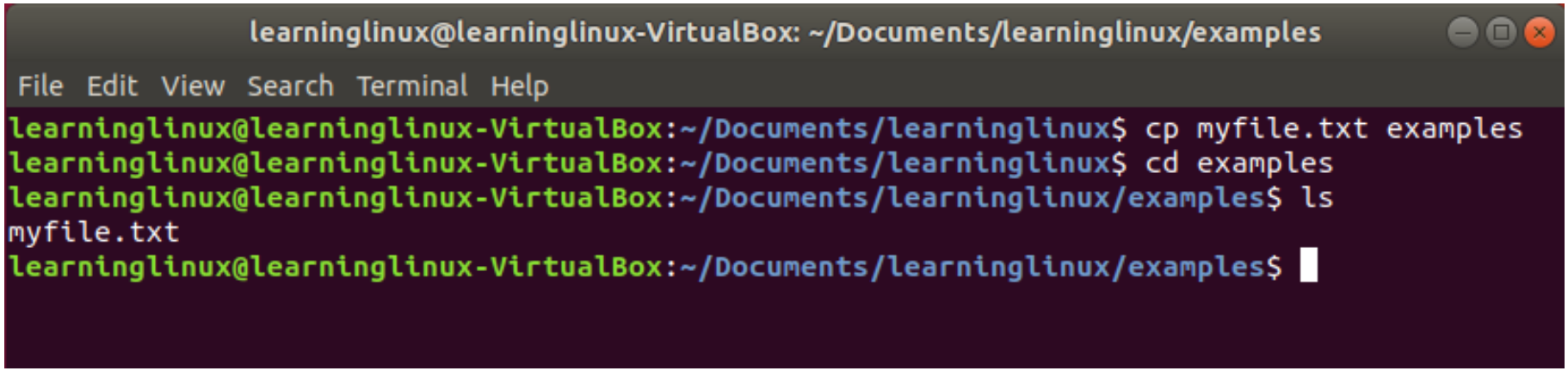
Making a Directory (mkdir)

- To make a new directory use the command **mkdir**
- The syntax is **mkdir [directory name]**



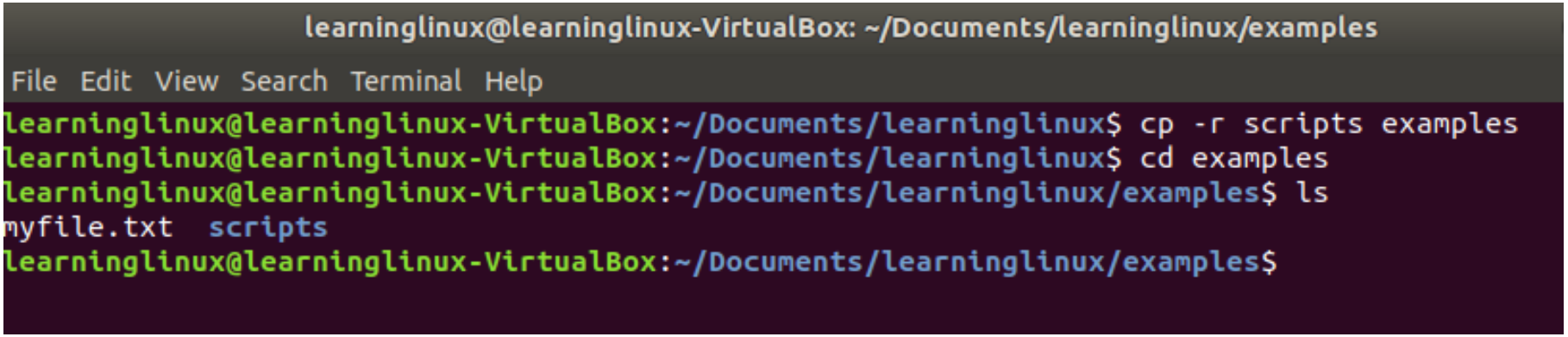
Copying Files (cp)

- To copy a file from one directory to another, use the **cp** command
- The syntax is **cp [file] [destination]**
- This will copy a file from the source directory to the destination directory



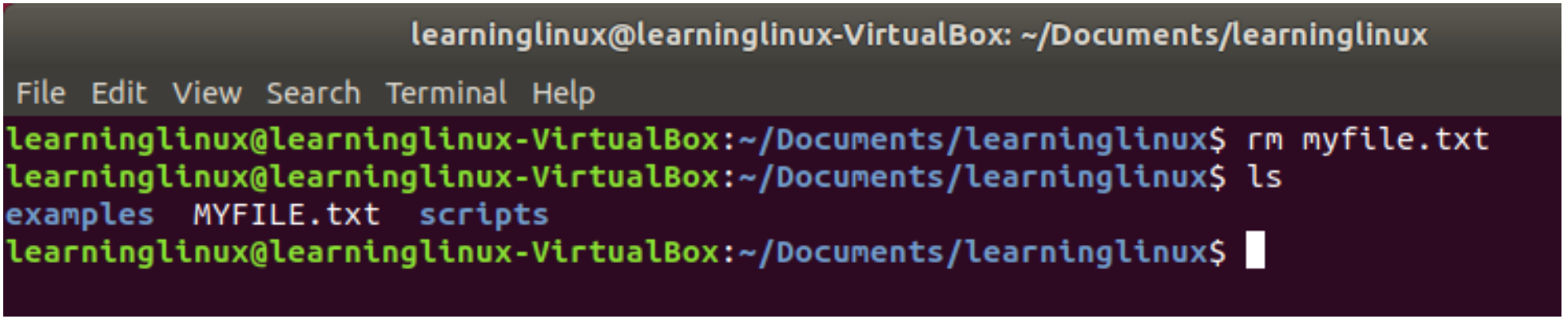
Copying Directories (cp)

- To copy an entire directory, you can still use the **cp** command
- The syntax is **cp -r [directory] [destination]**
- The **r** is an optional argument that will let you copy directory contents recursively



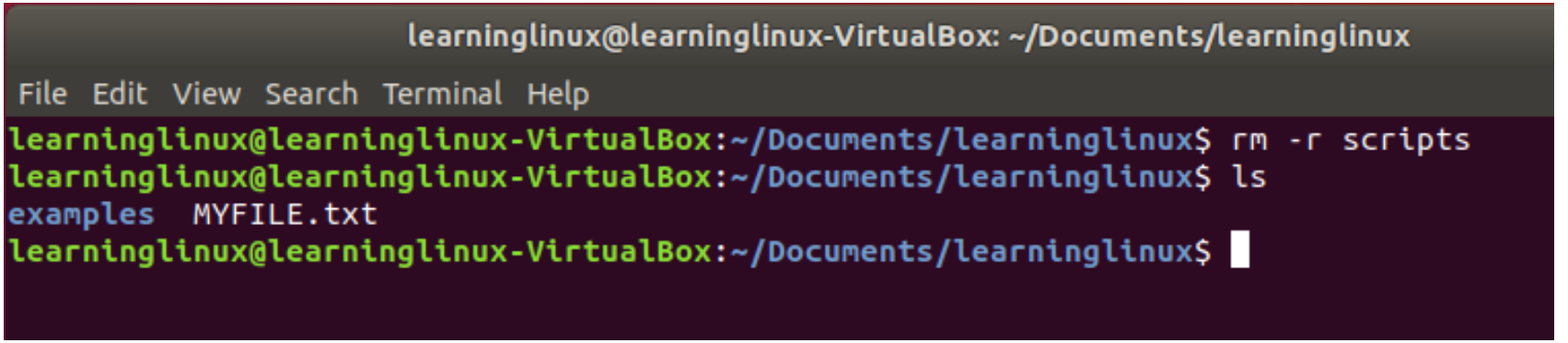
Removing Files (rm)

- To remove or delete a file, use the **rm** command
- The syntax is **rm [filename]**



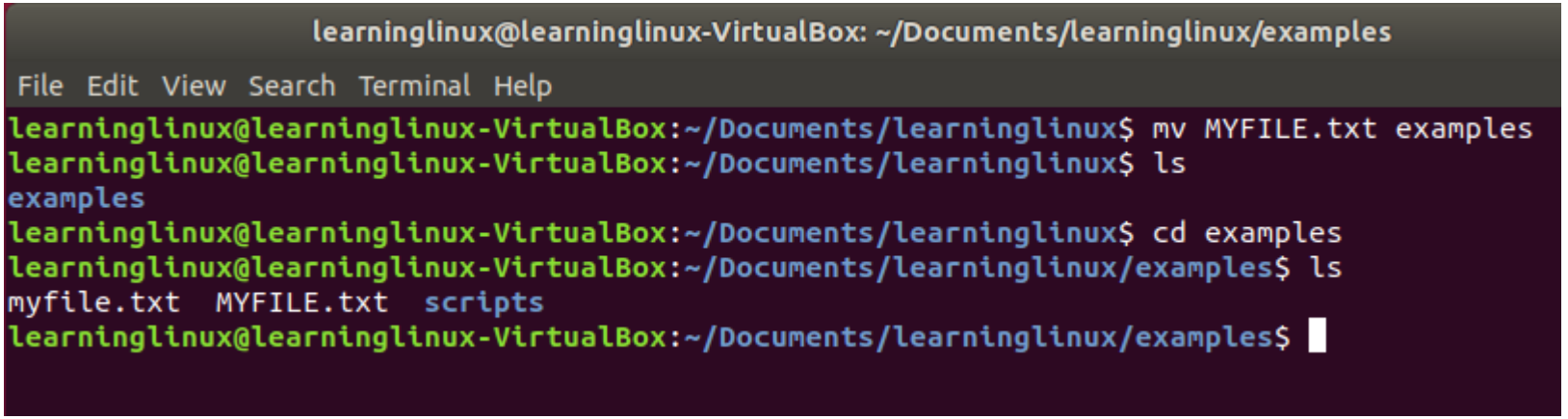
Removing Directories (rm)

- To remove or delete an entire directory, you can still use the **rm** command
- The syntax is **rm -r [filename]**
- Like the cp command, the r is an optional argument that you need to specify to work on directories



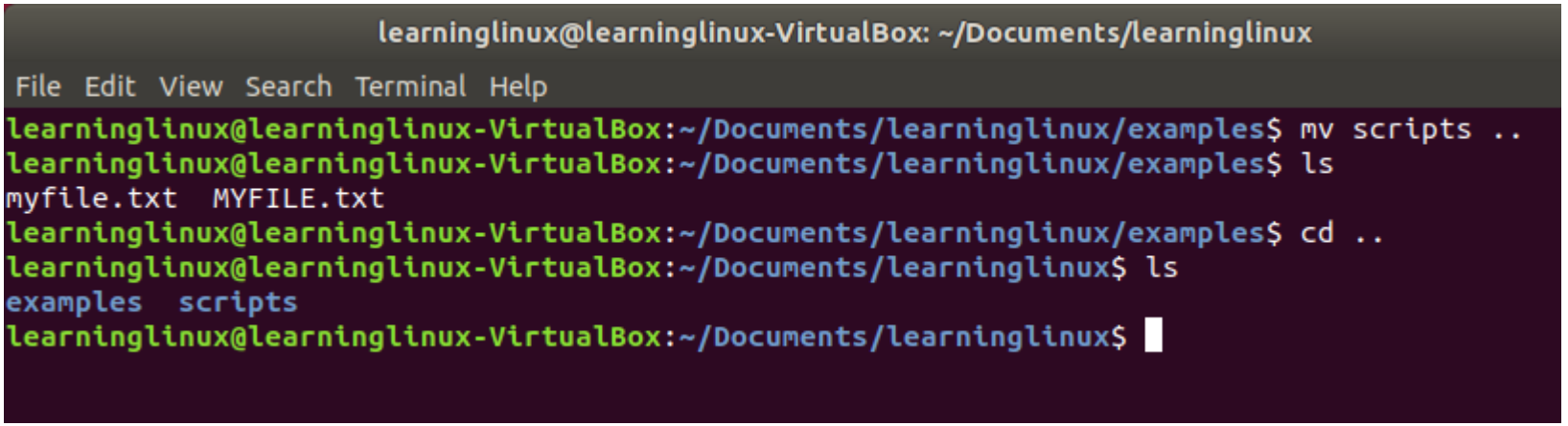
Moving Files (mv)

- To move a file from one directory to another, use the **mv** command
- The syntax is **mv [filename] [destination]**
- This moves the file to the destination folder



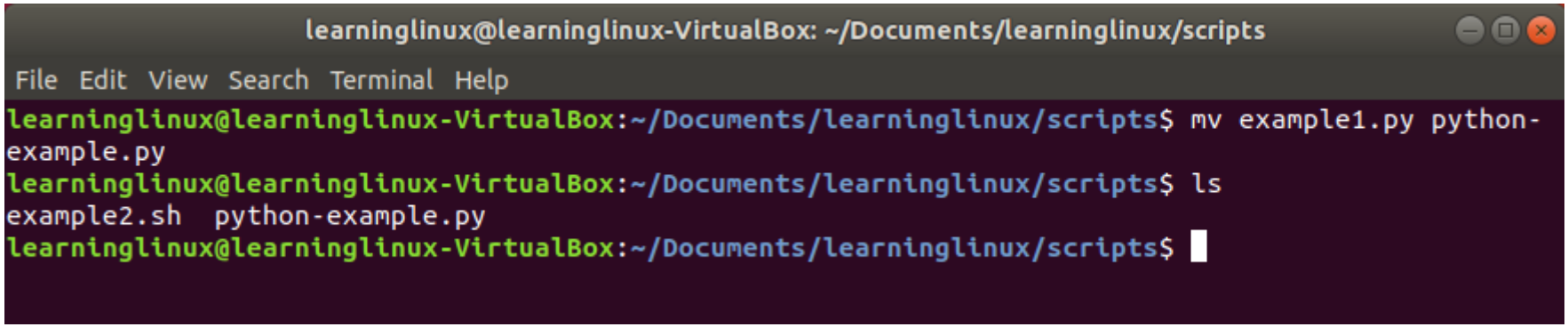
Moving Directories (mv)

- To move an entire directory, you can still use the **mv** command
- The syntax **mv -r [directory] [destination]**



Renaming Files (mv)

- To rename a file, you can also use the mv command
- The syntax is **mv [original filename] [new filename]**

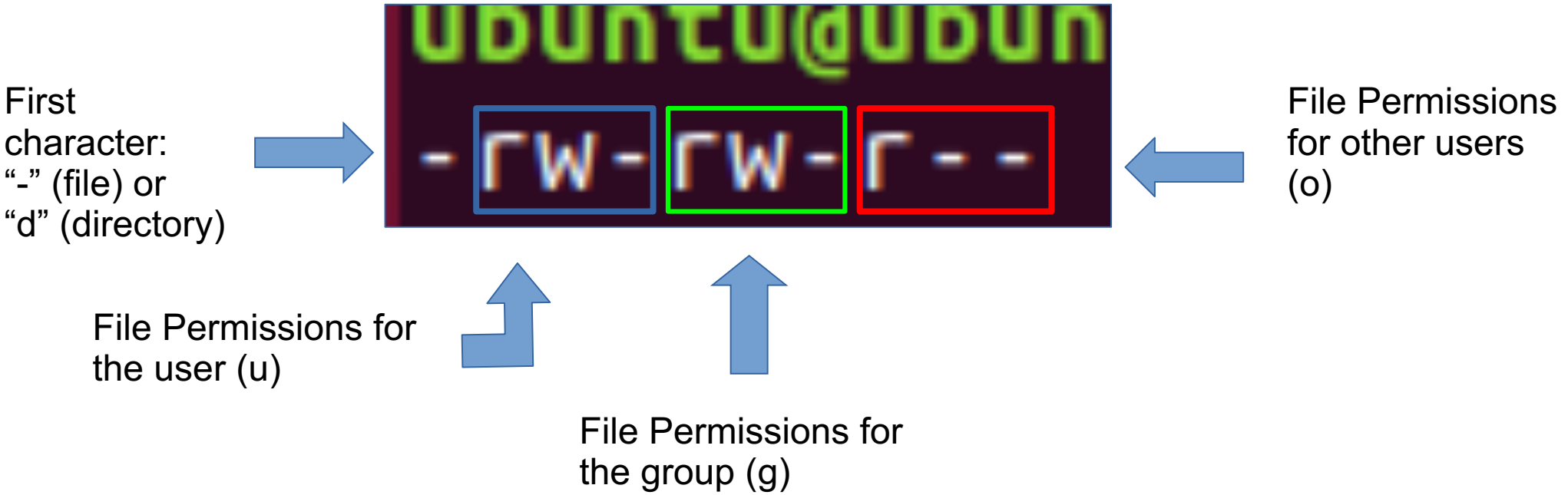


File Permissions – Background Info

- Unix systems have three levels of permissions:
  - **R**ead – user can view file contents (4)
  - **W**rite – user can edit file (2)
  - e**X**ecute – user can run file as a program or script (1)
- Users are split into three categories for permissions:
  - User/owner – user who created the file (u)
  - Group – group of users (g)
  - Other – all the other users (o)

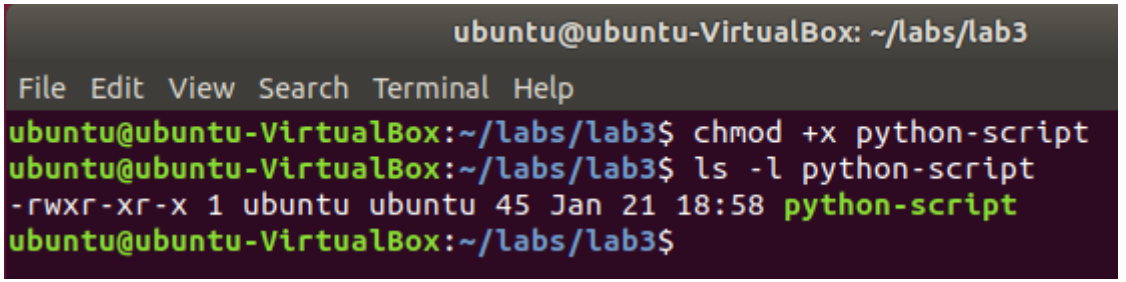
Reading File Permissions

- How to read file permissions:



Changing File Permissions – Basic

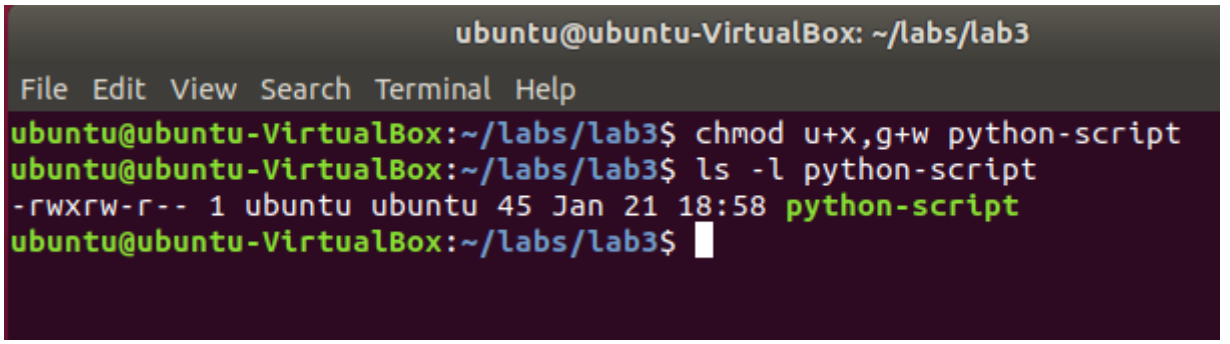
- Use the **chmod** command to change file permissions
- The syntax is **chmod [mode] [filename]**



Execute permissions are given to all users

Changing File Permissions – Advanced

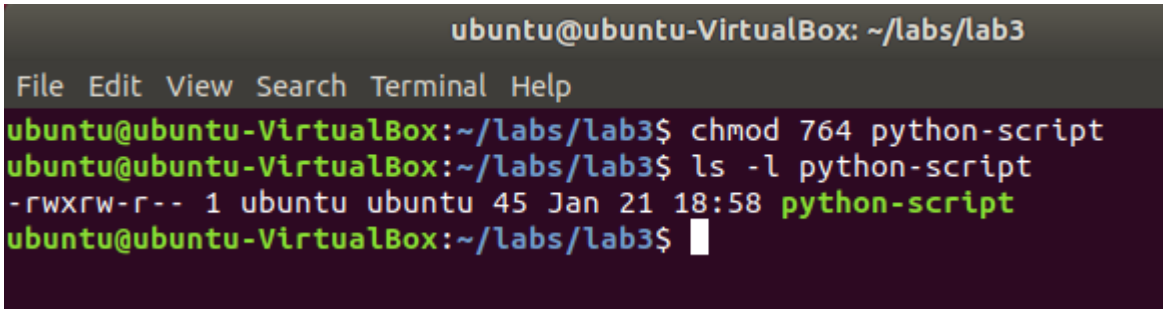
- **chmod u+x,g+w python-script**



Give execute permissions to the owner and write permissions to the group

Changing File Permissions – Octal

- We can use the octal number system to encode file permissions in numbers
- **chmod 764 python-script**
- Same thing as **chmod u+x,g+w python-script**
- Useful website: https://chmod-calculator.com/



7 is for rwx, 6 is rw-, and 4 is r--.

Source Code Available on Github

- All the code that is written in this lab is available on my github:
- https://github.com/petrauskasm/After-Hours-Command-Line-Basics

What is scripting?

- Scripting is a program that automates the execution of tasks
- Examples:
  - creating 100 directories
  - connecting to a server
- Scripting Languages:
  - Python
  - Ruby
  - Perl

Bash Scripting

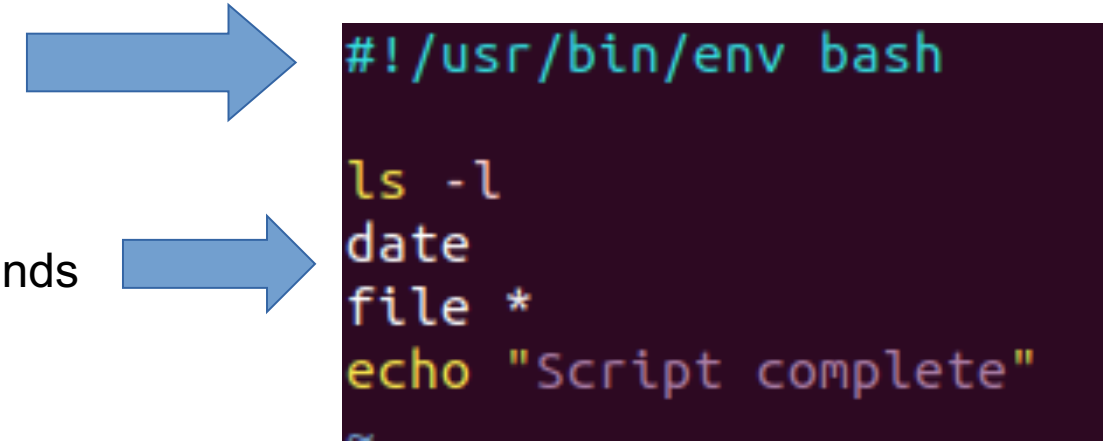
- All of the commands you have been entering on the command line are part of the Bash programming language
- Examples:
  - echo
  - ls
  - pwd
- You can write a script to execute these commands

Bash Scripting Example

- The following is a simple script written in bash

Shebang for bash

Some random bash commands



Environment Variables

- There are some special bash variables that you should take note of:
  - $PATH
  - $USER
  - $HOME
  - $SHELL
- Use the echo command to see what these variables are

Example scripts in bash

Print environment variables

```
#! /bin/bash
echo "Print script"
echo "User:" $USER
echo "HOME DIRECTORY: $HOME"
```

Create new files

```
#! /bin/bash
for i in {0..10}
do
  echo hello > "File$i.txt"
done
```

Wildcards

- There are three types of wildcards:
  - * (asterisk)
  - ? (question mark)
  - [] (square brackets)

Asterisk Wildcard

- The asterisk represents any number of characters
- Try the command **file *** in any directory that has some files

Question Mark Wildcard

- Rather than representing multiple characters like the asterisk, the question mark will only represent one character
- Run the command **ls -l example?.txt**

```
ubuntu@ubuntu-VirtualBox:~/labs/lab5/misc$ ls
christo.txt  example1.txt  example2.txt  example3.txt  example4.txt
ubuntu@ubuntu-VirtualBox:~/labs/lab5/misc$ ls -l example?.txt
-rw-r--r-- 1 ubuntu ubuntu 0 Feb 10 21:03 example1.txt
-rw-r--r-- 1 ubuntu ubuntu 0 Feb 10 21:04 example2.txt
-rw-r--r-- 1 ubuntu ubuntu 0 Feb 10 21:04 example3.txt
-rw-r--r-- 1 ubuntu ubuntu 0 Feb 10 21:04 example4.txt
-rw-r--r-- 1 ubuntu ubuntu 0 Feb 10 21:04 example5.txt
ubuntu@ubuntu-VirtualBox:~/labs/lab5/misc$
```

Square Brackets Wildcard

- The square brackets wildcard offers some flexibility in which characters you'd like to substitute
- With the square brackets, you can only substitute certain characters
- Try the command **file l[aeiou]st.txt**
- This will only return file names with the second character as a vowel and the other characters being fixed

```
ubuntu@ubuntu-VirtualBox:~/labs/lab5/misc$ file l[aeiou]st.txt
list.txt: ASCII text
lost.txt: ASCII text
ubuntu@ubuntu-VirtualBox:~/labs/lab5/misc$
```

Combining All the Wildcards

- The wildcards can be combined with each other to give more flexibility in your searches
- Examples:
  - **ls -l *.??** (this will search for any file which has a file extension which is two characters long)
  - **file [nmc]*** (this will search for anything which starts with “n”, “m”, or “c”
- There are endless ways to put together wildcards

What is a process?

- A process is an instance of a computer program being executed using code and instructions
- Each process uses system resources like CPU or RAM to complete the specific tasks

Different Types of Processes

- There are four types of processes:
  - Running: current process that is being executed in the operating system
  - Waiting: process which is waiting for system resources to run
  - Stopped: process that is not running
  - Zombie: process whose parent processes has ended, but the child process is still in the process table

Viewing Processes

- Two commands you can use to view the process from the command line: **ps** and **top**
- To view all the processes with **ps,** use **ps -ef**

```
ubuntu@ubuntu-VirtualBox:~/labs/lab6$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 10:27 ?        00:00:01 /sbin/init splash
root           2       0  0 10:27 ?        00:00:00 [kthreadd]
root           4       2  0 10:27 ?        00:00:00 [kworker/0:0H]
root           6       2  0 10:27 ?        00:00:00 [mm_percpu_wq]
root           7       2  0 10:27 ?        00:00:00 [ksoftirqd/0]
root           8       2  0 10:27 ?        00:00:00 [rcu_sched]
root           9       2  0 10:27 ?        00:00:00 [rcu_bh]
root          10       2  0 10:27 ?        00:00:00 [migration/0]
root          11       2  0 10:27 ?        00:00:00 [watchdog/0]
root          12       2  0 10:27 ?        00:00:00 [cpuhp/0]
root          13       2  0 10:27 ?        00:00:00 [kdevtmpfs]
root          14       2  0 10:27 ?        00:00:00 [netns]
root          15       2  0 10:27 ?        00:00:00 [rcu_tasks_kthre]
root          16       2  0 10:27 ?        00:00:00 [kauditd]
```

ps -ef

```
top - 10:48:42 up 21 min,  1 user,  load average: 0.03, 0.09, 0.20
Tasks: 212 total,   1 running, 180 sleeping,   0 stopped,   0 zombie
%Cpu(s): 17.0 us,  4.1 sy,  0.0 ni, 75.5 id,  3.4 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  8168488 total,  5414240 free,  1656284 used,  1097964 buff/cache
KiB Swap:  1459804 total,  1459804 free,        0 used.  6165520 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1294 ubuntu    20   0 2933144 202880  80452 S 13.5  2.5   0:20.53 gnome-shell
 1122 ubuntu    20   0  501344 122496  65504 S  3.0  1.5   0:08.06 Xorg
 1673 ubuntu    20   0  868420  37936  27812 S  2.0  0.5   0:01.47 gnome-terminal-
  915 gdm       20   0 2903920 129028  76872 S  0.7  1.6   0:03.48 gnome-shell
 1316 ubuntu     9 -11 1959040  12456   8944 S  0.7  0.2   0:00.08 pulseaudio
 1325 ubuntu    20   0  361564   7892   6416 S  0.7  0.1   0:00.68 ibus-daemon
 1453 ubuntu    20   0 1130700  24192  19160 S  0.7  0.3   0:00.08 gsd-media-keys
 1869 ubuntu    20   0 2124492 529224 172348 S  0.7  6.5   1:00.94 Web Content
  870 root      20   0  255476   2748   2376 S  0.3  0.0   0:00.30 VBoxService
  922 root      20   0  322300   8448   7328 S  0.3  0.1   0:00.09 upowerd
 1959 ubuntu    20   0 1518980 104680  80468 S  0.3  1.3   0:03.65 WebExtensions
    1 root      20   0  159948   9244   6764 S  0.0  0.1   0:01.54 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
```

top

Ending a Process In Linux

- Sometimes you need to end a program or process from the command line. Use the following steps:
  1. Locate the process id [PID] of the process/program you want to kill
  2. Use the **kill** command with the following syntax: **kill [PID]**
  3. If the process is still running, do the following: **kill -9 [PID]**
  4. The -9 is a SIGKILL signal telling the process to terminate immediately

What is filtering?

- Filtering is a process by which a large set of data is restricted by certain conditions to make the data set smaller

Head

- The **head** command will output the first part of a file
- The syntax is **head [OPTIONS] [FILE]**
- Example:
  - **head -5 random-words.txt**
  - **Head -1 random-passwords.txt**

Tail

- The **tail** command will output the last part of a file
- The syntax is **tail [options] [file]**
- Examples:
  - **tail -5 random-words.txt**
  - **tail -1 random-passwords.txt**

Sorting

- Sorting is a common filtering technique
- There is a built-in **sort** command
- The syntax is **sort [options] [file]**
- Example:
  - **sort random-passwords.txt**
  - This will sort the contents of the file alphabetically

More Sorting

- There are numerous options that you can use with the **sort** command
- Some common options:
  - -r: this will reverse the sorting
  - -c: this will check to see if the contents are already sorted
  - -o: let's you specify an output file for sorting

Word Count

- The **wc** command will print out information about word count, lines, and bytes in a file
- **wc random-passwords.txt**
- Optional Arguments:
  - -c: display the number of bytes in the file
  - -l: display the number of newline characters in the file
  - -w: display the number of words in the file

Unique Items

- To see unique items of duplicates in a file, you can use the **uniq** command
- The syntax is **uniq [options] [file]**
- **uniq dups.txt**
- Note: this command only works when duplicate items are adjacent to each other, run the **sort** command first before using **uniq**

More & Less

- The **more** and **less** commands can be used to help with reading large files
- They will display as much information as they can in the terminal and then you can scroll through the rest of it at your own leisure

Remote connections

- Use **ssh** to connect to a remote machine
  - ssh alina@login.ccs.neu.edu
  - First time connecting to a machine, need to verify its public key (aka digital certficate)
- Use **scp** to copy files on the remote machine
  - **scp** local_file remote_file
  - Use network path for the remote file
  - **scp** file.txt alina@login.ccs.edu.edu:~/files/text_files

Remote user and machine

File path on remote machine