Introduction To Ubuntu

MRE/EME 5983 Robot Operating Systems

Overview

- Introduction to Ubuntu / command line interface
- Text file editors
- Compiling C++ code
- File permissions
- Python interface
- Screen capture
- Summary

Overview

- Introduction to Ubuntu / command line interface
- Text file editors
- Compiling C++ code
- File permissions
- Python interface
- Screen capture
- Summary

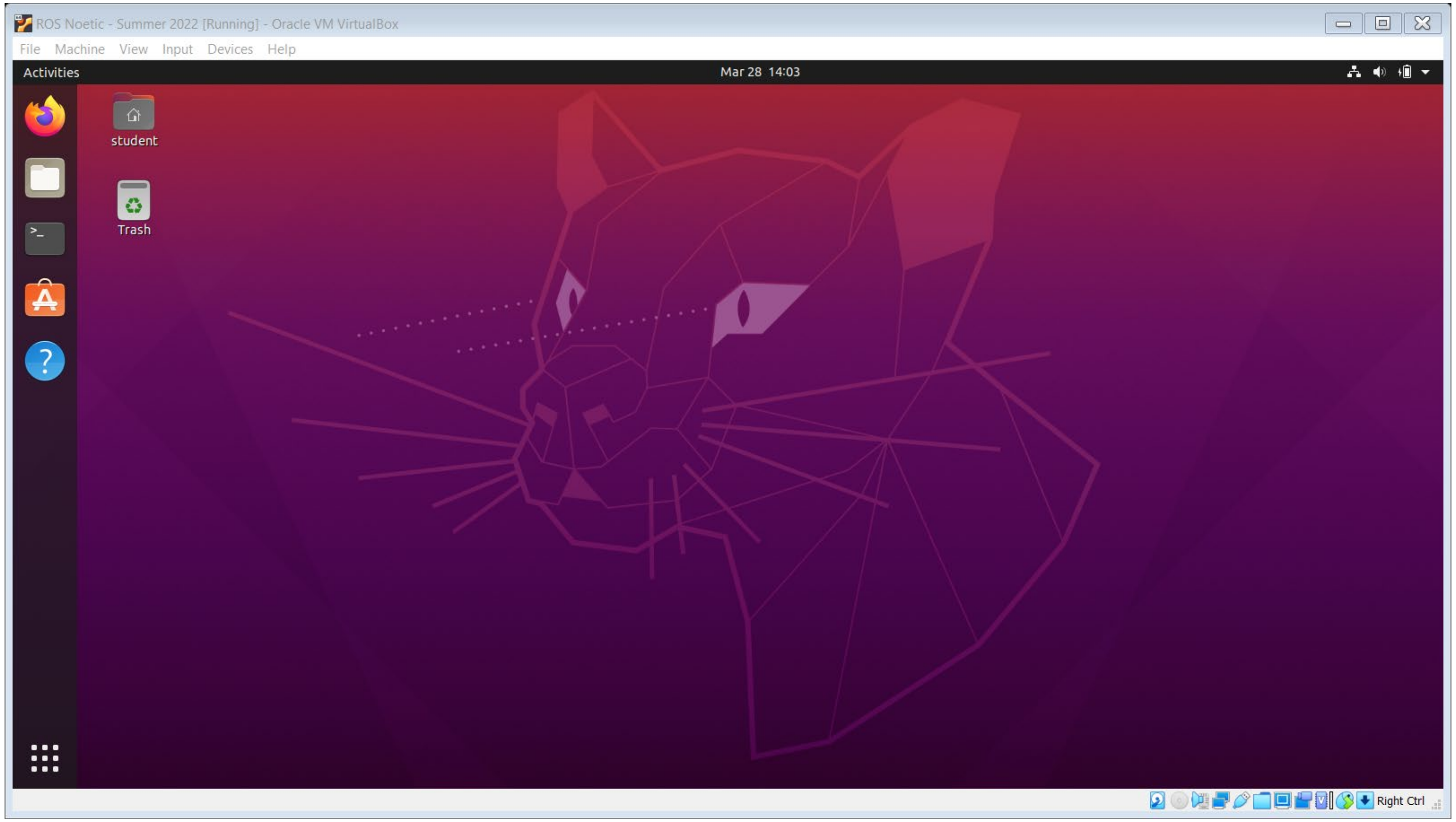
What is Ubuntu?

- Ubuntu is a Linux distribution based on Debian and composed mostly of free and open-source software. Ubuntu is officially released in three editions: Desktop, Server and Core for Internet of things devices and robots. All the editions can run on the computer alone, or in a virtual machine. Ubuntu is a popular operating system for cloud computing, with support for OpenStack. Ubuntu's default desktop has been GNOME since version 17.10.

- We will leverage
  - Ubuntu 20.04 Desktop
  - Run in virtual machine (Oracle Virtual Box)
  - Using GNOME desktop

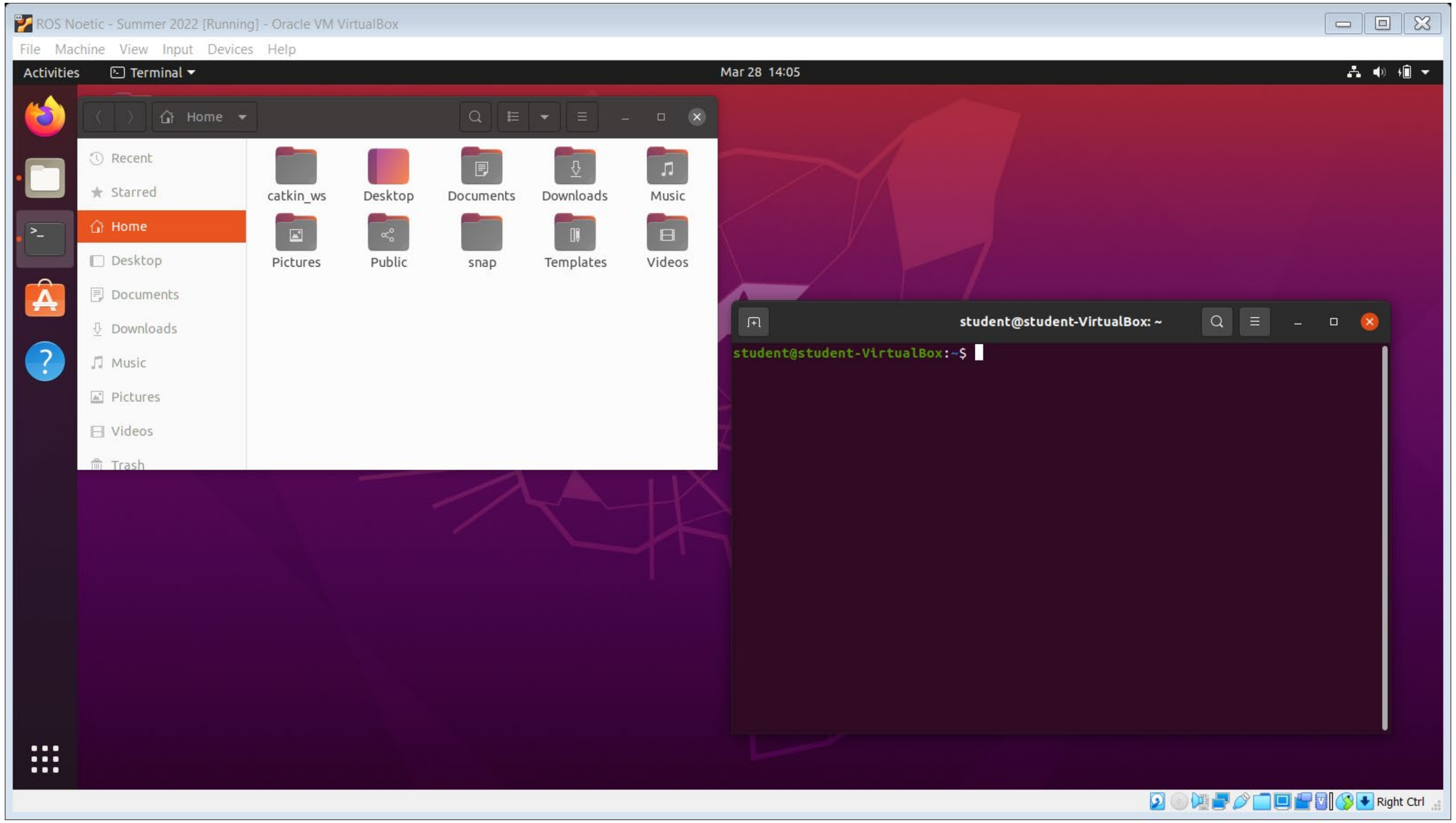
Operating Environment

- Assumptions
  - Ubuntu 20.04 running in Oracle VirtualBox or natively
  - GNOME desktop environment


ROS Noetic - Summer 2022 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Mar 28 14:03
student
Trash
Right Ctrl

Basic Ubuntu Gnome Navigation

- Multiple options
  - File Browser – Select file icon from favorites
  - Command line (terminal) – Select icon from favorites or Ctrl-Alt-t



```
student@student-VirtualBox:~$ 
```

Linux Basics: File Commands

```
$ ls          # list files in the current directory
$ ls –a       # list all files, including hidden files
$ ls –lt      # Sorting the formatted listing by time modification
$ cd          # change to HOME Directory
$ cd dir      # change directory to dir directory
$ pwd         # show current working directory
$ cp f1 f2    # copy file f1 to f2
$ cp -r d1 d2 # copy directory d1 to d2; create d2 if not present
$ mkdir dir   # make(create) a directory dir
$ cat         # display contents of a file
$ head file   # Output the first 10 lines of the file
$ tail file   # Output the last 10 lines of the file
$ touch file  # create or update date of a file
```

Linux Basics: File Commands

```
$ rm f1                # delete a file, f1
$ rmdir dir            # delete directory dir, if empty
$ rm -r dir            # delete non-empty directory dir
$ rm -rf dir           # Force to remove the directory dir
$ mv f1 f2             # Rename or move f1 to f2
$ ln -s file lnk       # Create symbolic link lnk to file
$ chmod +x file        # Change the permission of file to execute
$ tree                 # Display the tree hierarchy of a directory

$ source               # reads and executes commands from the file
                       # specified as its argument in the current shell
                       # environment. It is useful to load functions,
                       # variables and configuration files
```

Linux Basics: Archival / Compression

```
$ tar cvf archive.tar files         # Create a single file containing
                                    # multiple files
                                    # c – create, v – verify, f – files

$ tar xvf archive.tar               # Extract files from archive
                                    # x – execute

$ gzip archive.tar                  # Compress (creates archive.tar.gz)

$ gunzip archive.tar.gz             # Uncompress (creates archive.tar)

$ tar cvzf archive.tar.gz files     # Combines tar and gzip

$ tar xvzf archive.tar.gz           # Uncompress (creates files)
```

Linux Basics: System Commands

```
$ man command     # show the manual for the command

$ tldr command    # simplify “man” pages w/ practical examples
                  # (stands for “Too Long; Didn’t Read“. )

$ df              # show the disk usage
$ du              # Directory space usage

$ whereis app     # show possible location of app
$ which app       # Show which app will be run by default
```

Linux Basics: Network Commands

```
$ ping host           # ping host
$ dig domain          # Get DNS info for the domain
$ wget file           # download file
$ ifconfig            # initialize an interface, assign IP Address to
                      # interface and enable or disable interface on
                      # demand. With this command you can view IP
                      # Address and Hardware / MAC address assign to
                      # interface and also MTU (Maximum transmission
                      # unit) size.

$ netstat             # connection info, routing table info, etc
```

Linux Basics: Process Management Commands

```
$ ps                 # display the currently working processes

$ top                # Display all running process

$ kill pid           # Kill the process with given pid

$ killall proc       # Kill all the process named proc
```

Linux Basics: Searching Commands

```
$ grep pattern file      # Search for pattern in file
$ grep -r pattern dir    # Search recursively for pattern in dir
$ cmd | grep pattern     # Search pattern in the output of a cmd

$ locate file            # Find all instances of file
$ find . -name fn        # Searches in the current directory
                         # (represented by a period) and below it,
                         # for files and directories with names
                         # starting with filename, fn
```

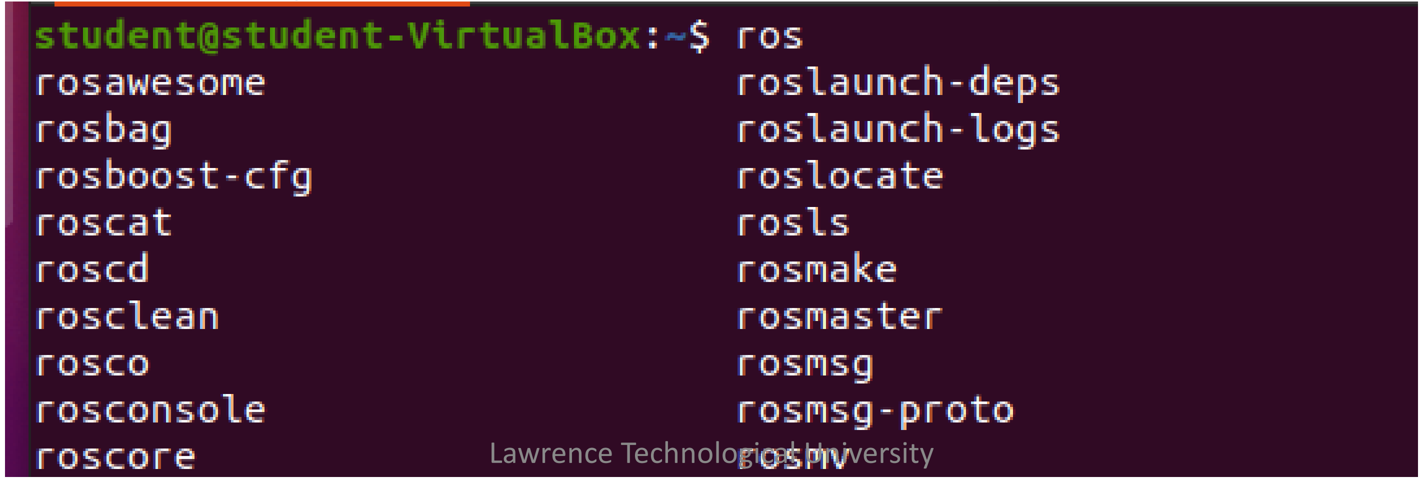
Linux Basics: Other Useful Commands

```
$ sudo                  # switch user and do this command
$ sudo apt install app  # installing new software app = new version
                        # of apt-get

$ wc                    # short for word count. Reads either
                        # standard input or a list of files and
                        # generates: newline count, word count, and
                        # byte count.
$ sort                  # sorting numerical values and strings.
                        # order the lines in a text file.
$ less                  # displays the contents of a file or a
                        # command output, one page at a time
$ clear                 # clears the terminal window
```

Linux Basics: Other Useful Commands

```
$ history           # history of all commands in a terminal


$ !!                # Repeat the last command
$ !<number>         # Repeat command with id number


$ Up arrow          $ command history traverse backward
$ Down arrow        $ command history traverse forward
```

- Use tab to auto complete commands, example $ ros <TAB> <TAB>



```
student@student-VirtualBox:~$ ros
rosawesome                    roslaunch-deps
rosbag                        roslaunch-logs
rosboost-cfg                  roslocate
roscat                        rosls
roscd                         rosmake
rosclean                      rosmaster
rosco                         rosmsg
rosconsole                    rosmsg-proto
roscore                       rosmv
```

Linux Basics – Bash

- Bash is a Unix shell and command language

- Code that interprets the commands you enter at the command line

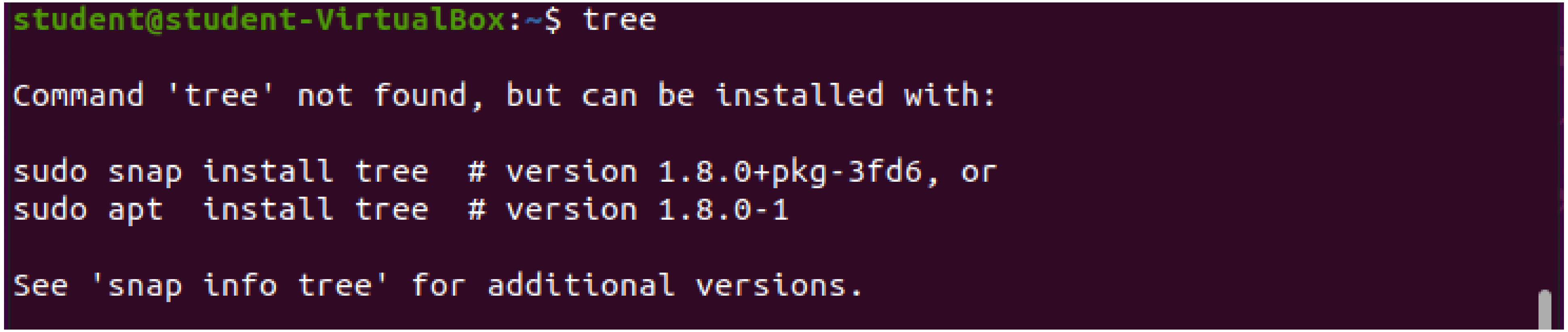
- Key component ~/.bashrc
  - Bash shell script that Bash runs whenever it is started interactively

- Users can create their own shell scripts
  - Bash shell scripts first line

    ```
    #!/bin/bash
    ```
  - Bash shell scripts must have executable permissions

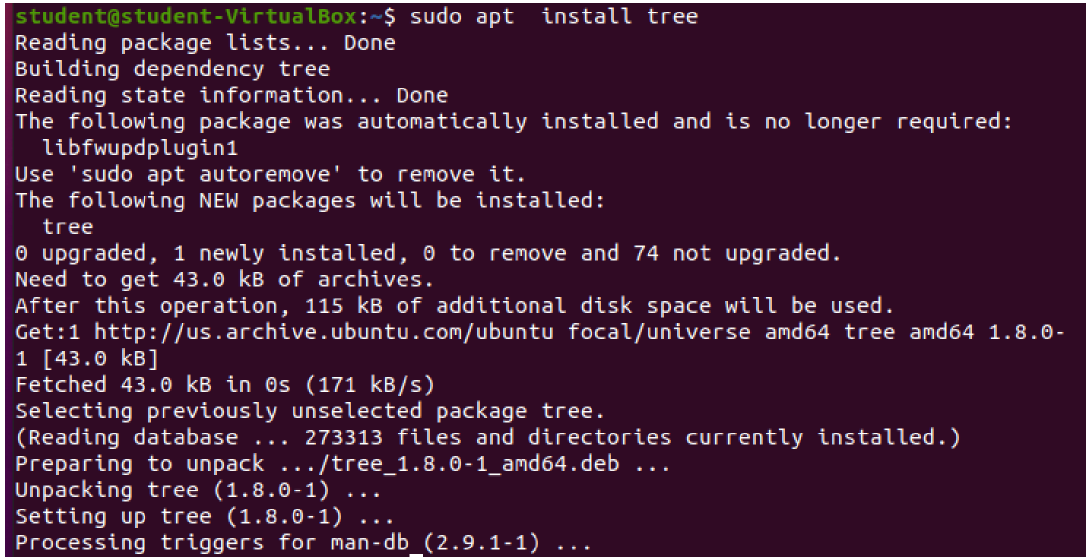
Examples

- tree command to display a directory hierarchy

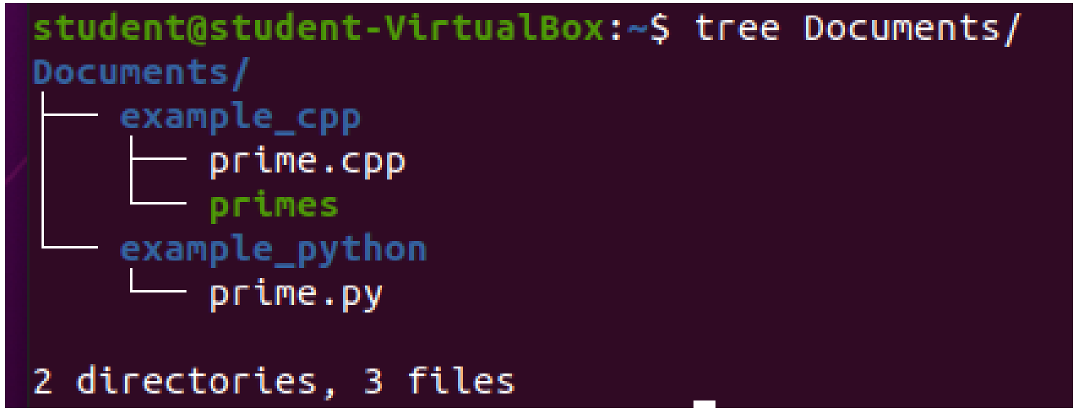


```
student@student-VirtualBox:~$ tree

Command 'tree' not found, but can be installed with:

sudo snap install tree  # version 1.8.0+pkg-3fd6, or
sudo apt  install tree  # version 1.8.0-1

See 'snap info tree' for additional versions.
```

- Install and run tree



```
student@student-VirtualBox:~$ sudo apt  install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 74 not upgraded.
Need to get 43.0 kB of archives.
After this operation, 115 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 tree amd64 1.8.0-
1 [43.0 kB]
Fetched 43.0 kB in 0s (171 kB/s)
Selecting previously unselected package tree.
(Reading database ... 273313 files and directories currently installed.)
Preparing to unpack .../tree_1.8.0-1_amd64.deb ...
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
```



```
student@student-VirtualBox:~$ tree Documents/
Documents/
├── example_cpp
│   ├── prime.cpp
│   └── primes
└── example_python
    └── prime.py

2 directories, 3 files
```

Some Linux Command Tutorials

- https://ubuntu.com/tutorials/command-line-for-beginners
- https://www.hostinger.com/tutorials/linux-commands
- https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners
- https://linuxize.com/post/basic-linux-commands/

Special Linux Symbols

#	comments
~	user HOME directory
.	current directory
..	parent directory
*	star wildcard, for any characters
?	question mark wildcard, for a character
$	command prompt for a regular user or Variable
|	Pipe output to another process/command
>	to capture the output of a command as a text file
>>	to append the output of a command to a text file
;	command separator

Useful Gnome Terminal Shortcuts

```
Ctrl-c             # Halt the current command
Ctrl-z             # Stops the current command,
                   # resume with fg (foreground) or bg (background)
Ctrl-d             # Logout the current session, similar to exit
Ctrl- Shift +      # Increase Terminal font
Ctrl- -            # Decrease Terminal font
Ctrl- Shift t      # To open a new tab inside a terminal
Ctrl- Shift c      # Copy
Ctrl- Shift v      # Paste to Terminal
Alt- i             # Switch to tab i
```

Overview

- Introduction to Ubuntu / command line interface
- Text file editors
- Compiling C++ code
- File permissions
- Python interface
- Screen capture
- Summary

Text File Editors / IDE’s

- The VirtualBox image has the following tools installed
  - emacs
  - gedit
  - nano
  - vi
  - VS Code (type **code** at the command line)

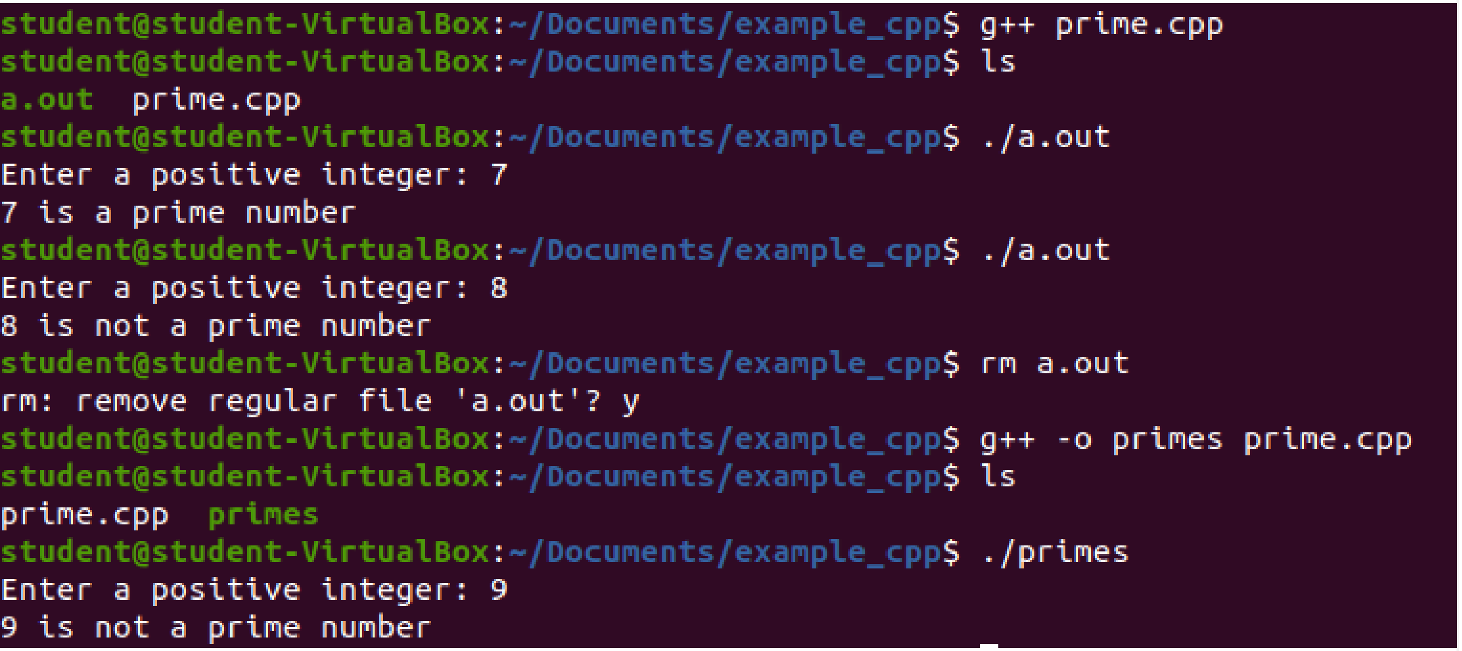
- Ubuntu/Linux offer several professional grade integrated development environments (IDE’s) available, but they are not covered in this material

Overview

- Introduction to Ubuntu / command line interface
- Text file editors
- Compiling C++ code
- File permissions
- Python interface
- Screen capture
- Summary

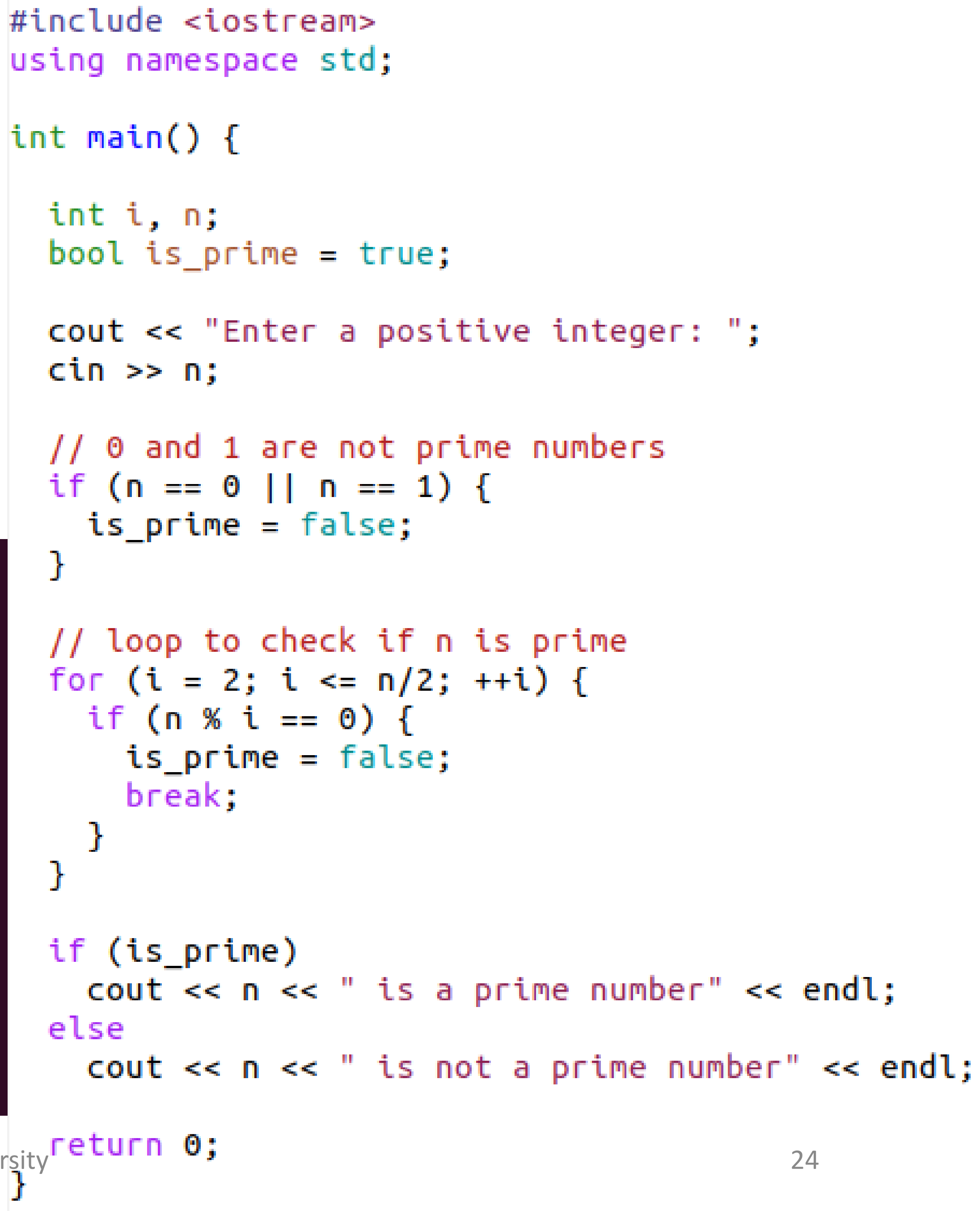
Compiling C++ Code

- Example C++ exercise
  - Create prime.cpp
  - Compile with g++ (default a.out exe)
  - Test
  - Re-compile with primes exe name
  - Re-test



```
student@student-VirtualBox:~/Documents/example_cpp$ g++ prime.cpp
student@student-VirtualBox:~/Documents/example_cpp$ ls
a.out  prime.cpp
student@student-VirtualBox:~/Documents/example_cpp$ ./a.out
Enter a positive integer: 7
7 is a prime number
student@student-VirtualBox:~/Documents/example_cpp$ ./a.out
Enter a positive integer: 8
8 is not a prime number
student@student-VirtualBox:~/Documents/example_cpp$ rm a.out
rm: remove regular file 'a.out'? y
student@student-VirtualBox:~/Documents/example_cpp$ g++ -o primes prime.cpp
student@student-VirtualBox:~/Documents/example_cpp$ ls
prime.cpp  primes
student@student-VirtualBox:~/Documents/example_cpp$ ./primes
Enter a positive integer: 9
9 is not a prime number
```

prime.cpp



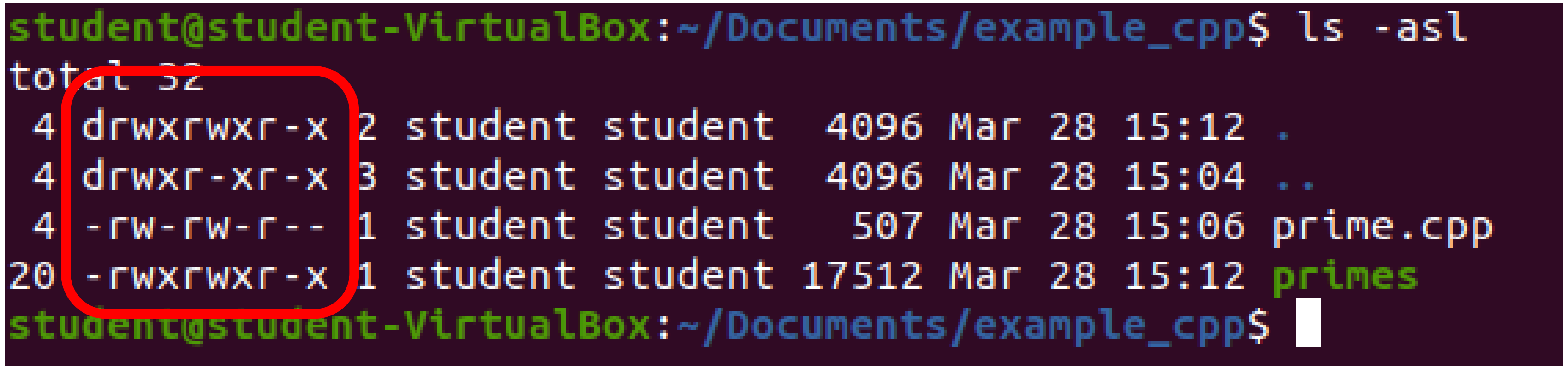
```cpp
#include <iostream>
using namespace std;

int main() {

  int i, n;
  bool is_prime = true;

  cout << "Enter a positive integer: ";
  cin >> n;

  // 0 and 1 are not prime numbers
  if (n == 0 || n == 1) {
    is_prime = false;
  }

  // loop to check if n is prime
  for (i = 2; i <= n/2; ++i) {
    if (n % i == 0) {
      is_prime = false;
      break;
    }
  }

  if (is_prime)
    cout << n << " is a prime number" << endl;
  else
    cout << n << " is not a prime number" << endl;

  return 0;
}
```

Overview

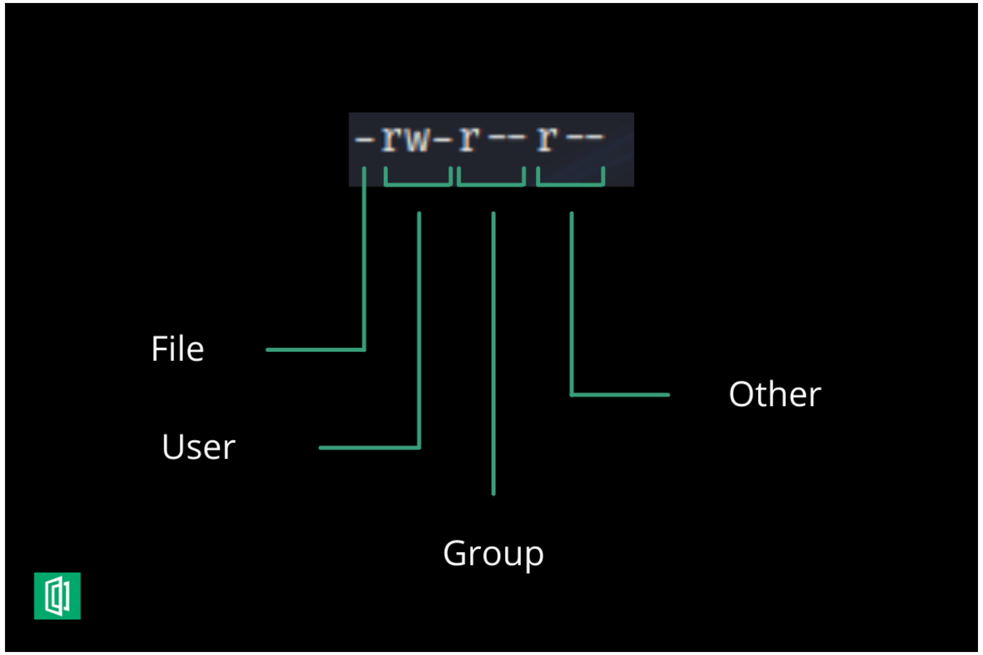
- Introduction to Ubuntu / command line interface
- Text file editors
- Compiling C++ code
- File permissions
- Python interface
- Screen capture
- Summary

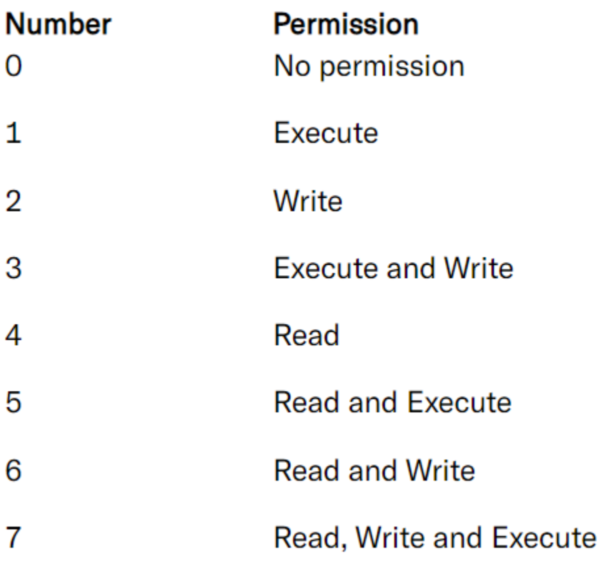
Understanding File Access

- Directory contents of last exercise
  - d = directory
  - r = read
  - w = write
  - x = execute



```
student@student-VirtualBox:~/Documents/example_cpp$ ls -asl
total 32
 4 drwxrwxr-x 2 student student  4096 Mar 28 15:12 .
 4 drwxr-xr-x 3 student student  4096 Mar 28 15:04 ..
 4 -rw-rw-r-- 1 student student   507 Mar 28 15:06 prime.cpp
20 -rwxrwxr-x 1 student student 17512 Mar 28 15:12 primes
student@student-VirtualBox:~/Documents/example_cpp$ 
```





Number	Permission
0	No permission
1	Execute
2	Write
3	Execute and Write
4	Read
5	Read and Execute
6	Read and Write
7	Read, Write and Execute

In our case,

prime.cpp has 664 permissions
primes has 775 permissions

Use chmod to change permissions

Images from: https://www.section.io/engineering-education/user-groups-and-permissions-linux/

Overview

- Introduction to Ubuntu / command line interface
- Text file editors
- Compiling C++ code
- File permissions
- Python interface
- Screen capture
- Summary

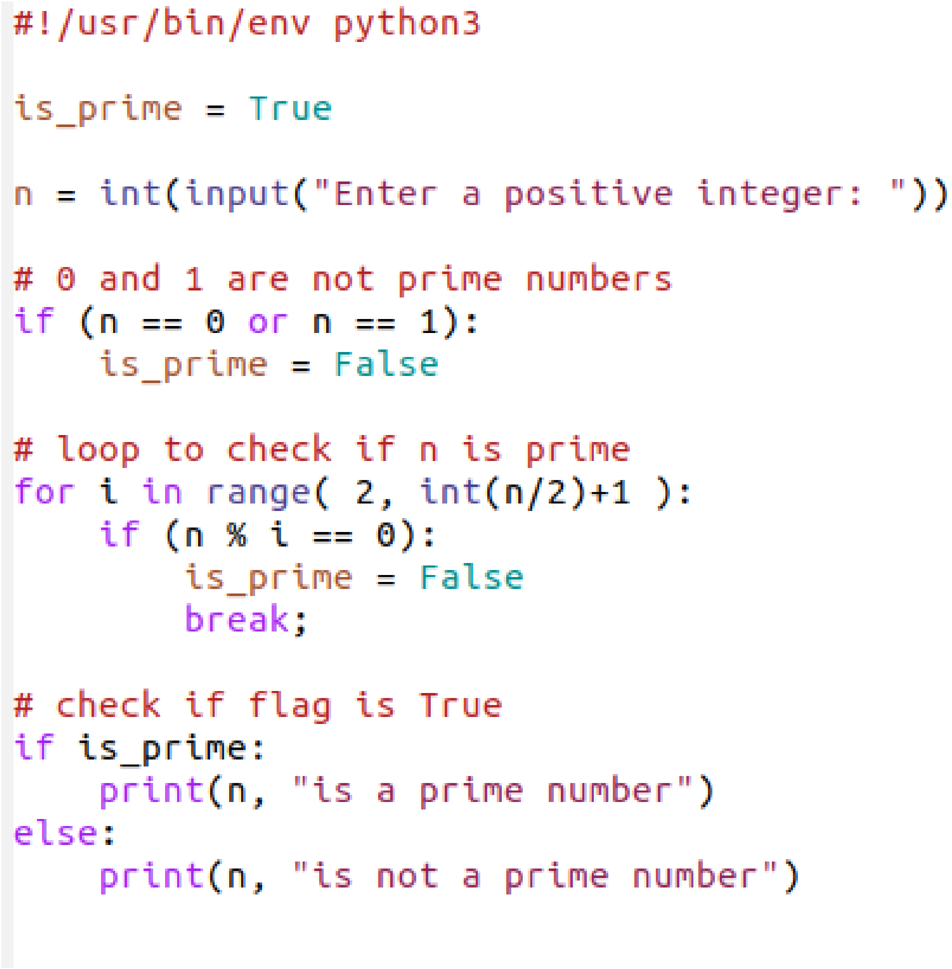
Python Interface

- In this course, we will be using Python3

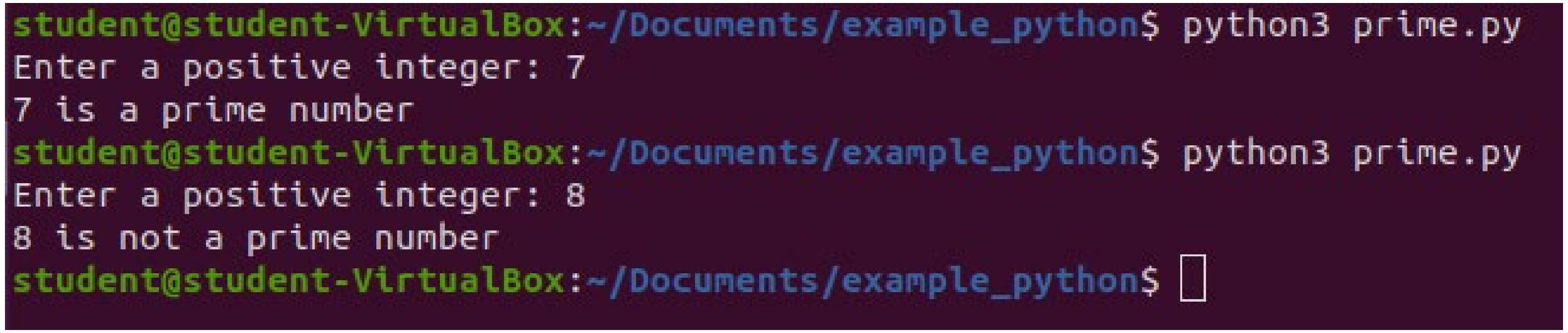
- Python may be executed in Interactive Mode or Command Line Mode

- Example python exercise
    - Create prime.py
    - Test in command line mode

prime.py



```
#!/usr/bin/env python3

is_prime = True

n = int(input("Enter a positive integer: "))

# 0 and 1 are not prime numbers
if (n == 0 or n == 1):
    is_prime = False

# loop to check if n is prime
for i in range( 2, int(n/2)+1 ):
    if (n % i == 0):
        is_prime = False
        break;

# check if flag is True
if is_prime:
    print(n, "is a prime number")
else:
    print(n, "is not a prime number")
```



```
student@student-VirtualBox:~/Documents/example_python$ python3 prime.py
Enter a positive integer: 7
7 is a prime number
student@student-VirtualBox:~/Documents/example_python$ python3 prime.py
Enter a positive integer: 8
8 is not a prime number
student@student-VirtualBox:~/Documents/example_python$ 
```

Python Learning Resources

- https://www.learnpython.org/
- https://docs.python-guide.org/intro/learning/
- https://www.python.org/about/gettingstarted/

Overview

- Introduction to Ubuntu / command line interface
- Text file editors
- Compiling C++ code
- File permissions
- Python interface
- Screen capture
- Summary

Screen Capture

- Ubuntu offers screen capturing tools using the following shortcuts:

Alt + Print	Capture the window that currently has focus
Shift + Print	Capture portion of screen
Shift + Ctrl + Alt + R	Start and stop recording a screen cast

- Window/screen captures are placed in ~/Pictures
- Screencast video records are paced in ~/Videos

- The default screencast capture is 30 seconds. To increase the time use the following command, where the last argument (60) is the time limit in seconds

```
$ gsettings set org.gnome.settings-daemon.plugins.media-keys max-screencast-length 60
```

Overview

- Introduction to Ubuntu / command line interface
- Text file editors
- Compiling C++ code
- File permissions
- Python interface
- Screen capture
- Summary

Summary

- We reviewed a Linux operating system, Ubuntu
- We reviewed how to interact with the file system and provide simple examples of how to write and compile C++ code
- We also reviewed how to execute Python code
- We will leverage many of these methods in upcoming lectures and homework assignments