Removable Device Imaging

Clark Otte

February 4, 2022

CSC 5347

**Problem Description**

The goal of the project is to make a copy of a drive. In order to do so a drive must be found and should then be copied exactly bit by bit to create the image of the drive. The main drive on the Linux machine should not be allowed to be copied. The user should be able to specify where the copy file is placed.

**Problem Approach**

The approach taken to solve this problem was to first make a list of all the drives available on the Linux Machine that aren’t the main drive. Then the user can select which drive they want copied and specify what path they want to copy the file to. The user can also specify how big of a block size they want to use for the reading and writing of the drive. After gathering the information from the user, the program will then perform the copying of the file bit by bit. When finished the program then returns a confirmation message and ends.

**Reading the Drive**

When operating in Linux, reading the bytes from a drive is essentially the same as reading from a file because everything in Linux is represented by a file descriptor. Each can use the open() method to access the file or drive and specify how in which way they want to access said file or drive. The python methods to access a file and to access a drive are the exact same, aside from having to know where the location of the drive file is stored on the Linux machine.

**Problem Approach for Windows and Dual Functionality**

There would be a few key differences in the program if it were written for windows. The file system of Windows has a different naming system when it comes to drives and the commands to list them would also be different. Instead of running a lsblk command on the Linux terminal to get the information on the drives, the program would have to use os.path or use GetLogicalDriveStrings(). After getting the drives and their locations the program would run fairly similar to the Linux program. In order to have dual functionality there would have to be a check to see what type of os the machine is running and then have two different functions to find the list of drives, one for Linux and one for Windows. This could be done using os.name or platform.system().

**Problem Approach for Different Languages (C++)**

Approaching the problem in C++ would be similar in the process of what the program would do, however the ease of doing each step in the process would be much different. Python allows for more leniency on the declaration of variables and how they are used. Manipulating the outputs of the file path strings and getting them into the correct format is much easier for the programmer in python. C++ allows for more control of the variables and the memory behind the program. Each language has a different approach to certain methods and processes which causes them both to have upsides and downsides.

**Confirming the Bit for Bit Copy**

To confirm that an exact copy of the file was made, hashing each disk image to get a hash value and comparing them would be one way to ensure that an exact copy was made. When comparing the hash values, they should be the same value. This can be done either using a python script or by using the Linux terminal to hash the files and compare the sums. This ensures that an exact copy was made.