Rajesh Rao
Professor: Ivan Matic

HOMEWORK 1

1. **Search the internet to find useful information and introductory articles about terminal commands cp, mv, rm, and rmdir.**
    a. **List 3 web pages from the internet that you found to provide the best information.**
        - https://vitux.com/40-most-used-ubuntu-commands/
        - https://tutorials.ubuntu.com/tutorial/command-line-for-beginners#4
        - https://en.wikipedia.org/wiki/List_of_Unix_commands
    b. **Write a one page report on what you have learned.**

    According to vitux.com, `cp`, is a command in Ubuntu "used to copy data from a source file to the destination file." This is similar to Ctrl + C on a windows, where the source file is not deleted after being moved to another directory. According to wikipedia, the three general modes of operation are as follows, with accompanying syntax:

    - When copying a file to another file, we provide the source file followed by the target file we would like to paste contents in (if the target file is non-existent one will be created).

      ```
      cp source_file target_file
      ```

    - When copying more than one file to another directory, we provide the source files followed by the target directory we would like to paste contents in.

      ```
      cp source_file_1 source_file_2 ... target_directory
      ```

    - When copying the contents of one directory to another directory, we provide the full path of our source directory followed by the target directory (if the target directory is non-existent one will be created).

      ```
      cp source_directory ... target_directory
      ```

    According to vitux.com, `mv`, is used to accomplish two primary objectives, which I have illustrated with the accompanying syntax below:

    - When wanting to move files or directories from one path to another, while simultaneously deleting the source file in the process. This is similar to `cp`, without preservation of the source file.

      ```
      mv source_file target_file
      ```

    - When wanting to rename an existing file, simply provide the full path to the source file and modify the target name requested.

      ```
      mv source_file_name source_target_name
      ```

    According to vitux.com, `rm`, is used to delete specific files from a directory. This is done through the syntax provided below, specifying the full path of the file looking to be deleted from the directory.

```
rm source_file_name
```

According to vitux.com, `rmdir`, "allows users to remove directories from the system," contingent on the fact that the folder is empty. The following is done simply by specifying the full path of the directory looking to be deleted.

```
rmdir source_directory
```

2. Following the instructions from the article Creation of a bootable USB stick with Ubuntu 18.04, I have constructed such a bootable USB.

3. **Search the internet to find useful information and introductory articles about sudo. Warning: Some of the articles will suggest you experiment with the command sudo visudo. In the case you choose to do so, please make sure that you know how to reinstall the operating system Ubuntu 18.04 without my help.**
    a. **List 3 web pages from the internet that you found to provide the best information about sudo.**
        - https://www.linux.com/tutorials/linux-101-introduction-sudo/
        - https://www.tutorialspoint.com/unix_commands/sudo.htm
        - https://en.wikipedia.org/wiki/Sudo
    b. **Write a one page report on what you have learned.**
    First developed by Robert Coggeshall and Cliff Spencer in 1980 at the Department of Computer Science at SUNY/Buffalo, `sudo`, is often used to execute administrative tasks from the user. According to Linux.com, the usage of `sudo` can be justified through the need for "superuser privileges" that extend beyond the "standard user" permissions. These "superusers" as specified in the `sudoers` file, which controls which users have the ability to run commands on a particular machine, grant privileges that include but are not limited to installing applications and executing commands. However, prior to invoking `sudo` , the user is to provide a password to authenticate themselves. Yet, unlike the similar `su` command, `sudo` retains the invocation rights for five-minutes after entering the password. This in turn, enables successive commands to be executed without needing to repetitively authenticate one's self, as would be required while using `su`. It is also important to note that `sudo` allows for great optionality during execution, with keywords such as `-b` allowing for a command to be run in the background and `-p` to override the password prompt. In summary, `sudo` allows for an efficient and seamless way of executing commands on behalf of the user.