Course: Unix Name: Christina Kallas  
Section: 00001 License: CC by 4.0

Part 1 – GNU/Linux Distributions:

**Question 1)**

I have chosen to learn a bit more about Arch Linux, Fedora and Manjaro. A common reason I have chosen these three distributions is due to the fact that, according to this website, <https://analyticsindiamag.com/10-most-stable-linux-distros-in-2021/>, they are three of the most stable Linux distros.

According to its own website, <https://archlinux.org/about/>, Arch Linux is a x86-64 general purpose GNU/Linux distribution that is focused on simplicity, minimalism, and code elegance. It is so simplistic that the user can assemble their own environment by editing the shell with text files. Arch Linux takes pride in using its own Packman Package manager which allows users to easily manage and customize packages as well as owning a repository system that allows users to build and maintain their custom scripts, packages, and repositories. Its repository system also encourages community growth and contribution. The system is based on a “rolling- release”, which means the installation is done once and there are constant software upgrades. Patches are only used when ensuring the functionality of applications with other packages. To summarize, this Arch Linux is a simplistic and minimalistic distro for competent Linux users such as programmers and developers who do not mind having a GUI or creating their own trough packages and text files. It is a very user-oriented distros. The strengths that I see in this distro is that it is constantly kept up to date and gives full creative liberty to its users. The weakness, in my opinion, is that it does not come with an option of a GUI so it would not be suitable for beginners at all. Those who are skilled in Linux seem to like this distro. A reason I chose to learn a bit more about Arch Linux is because it is a distro, I hear about a lot but do not know anything about.

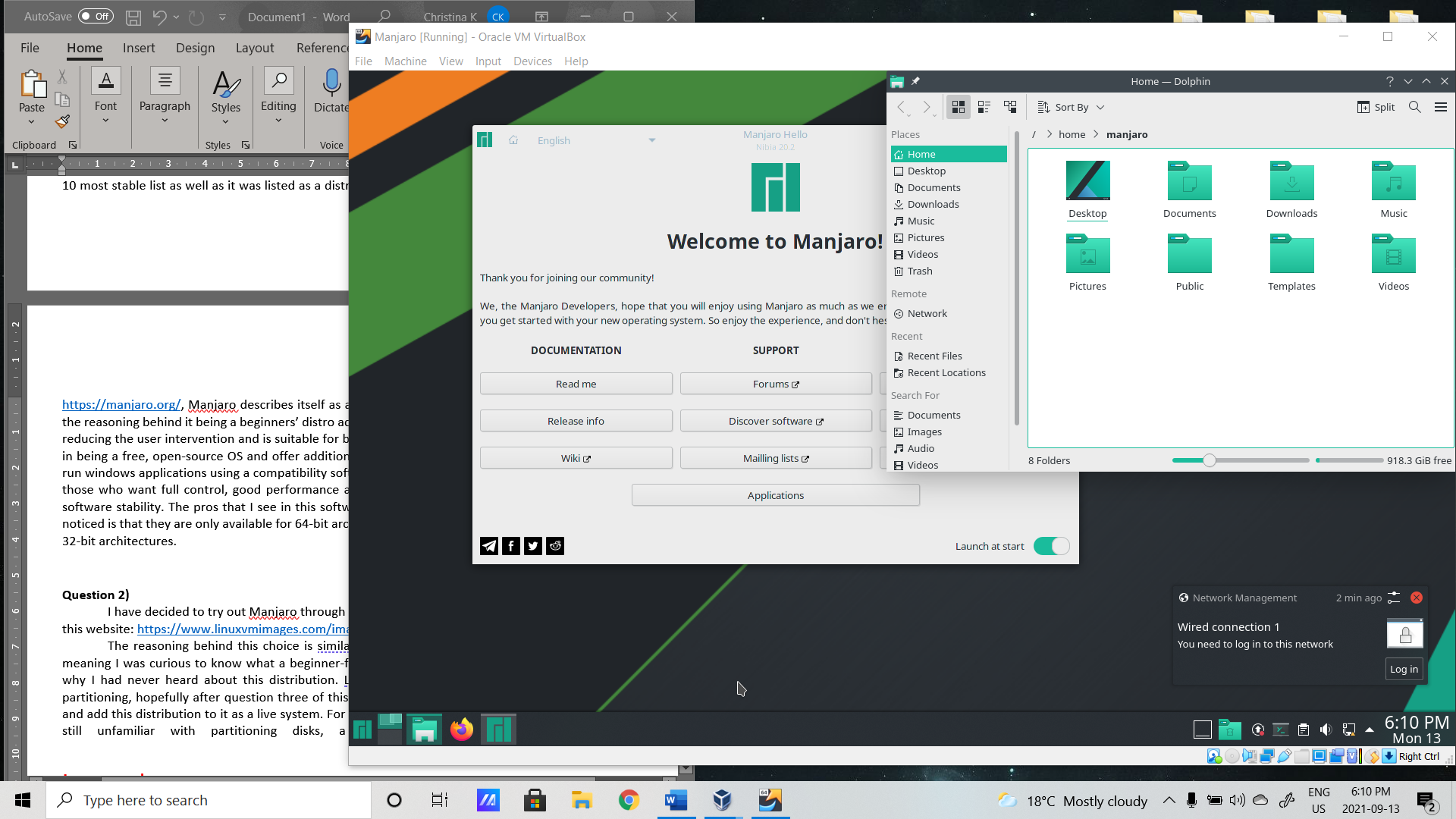
I chose to learn a bit more about Fedora because I keep hearing about it in class from other students and it is mentioned quite a lot by our teacher. Fedora has three operating systems that can be used. The first one is named *Fedora Workstation*. It is described as polished and easy to use, having a complete set of tools for users of all kinds. The second system is called *Fedora Server* and it is flexible and powerful, including the latest datacenter technologies. The Fedora website explains that it puts the user in control of infrastructure and services. *Fedora IoT* is the last operating system that this company offers. It provides an open-source platform as a strong foundation for IoT ecosystems, according to Fedora. Fedora also presents five more systems that will be released I the future. *Fedora CoreOS* which is minimal, and container focused as well as *Fedora Silverblue* which is an immutable OS aimed at good support. The next OS they mention is *Fedora Spins* which is an alternative desktop environment where the user can download a pre-configured version of Fedora, *Fedora Labs* which is a selection of content and software which is maintained by the Fedora community. It can be installed as an add-on or a full version of Fedora. Finally, *Fedora Alt Downloads* are more standard versions of Fedora, in alternative formats. What I feel is a strength in Fedora is that it offers different types of OS for different usages, meaning it can be applied to different types of users, may they be professionals, beginners, developers, or students. It feels like it is something many people can learn from and increase their knowledge on. A weakness that I have found in the different OS is that beginners, such as me, would not know which OS to download, although I would be interested in downloading their OS as they are upfront about Fedora’s future.

The third and final Linux distribution that I decided to learn more about is Manjaro. The reasons behind my picking of this distro are that I have never heard of this company, yet it did feature on the top 10 most stable list as well as it was listed as a distro for beginners, such as me. According to its website, <https://manjaro.org/>, Manjaro describes itself as an accessible and user-friendly OS. This would explain the reasoning behind it being a beginners’ distro according to the top 10 list. It contains automated tools, reducing the user intervention and is suitable for both beginners and experienced users. They take pride in being a free, open-source OS and offer additional free software. Their site explains that the user can run windows applications using a compatibility software and explain that they are the middle ground for those who want full control, good performance and a cutting-edge software with a certain degree of software stability. The pros that I see in this software is that it is beginner-friendly, the con that I have noticed is that they are only available for 64-bit architectures, XFCE, KDE and Gnome editions, but not for 32-bit architectures.

**Question 2)**

I have decided to try out Manjaro through Virtual box by downloading the image available on this website: <https://www.linuxvmimages.com/images/manjaro-20/#download-links>.

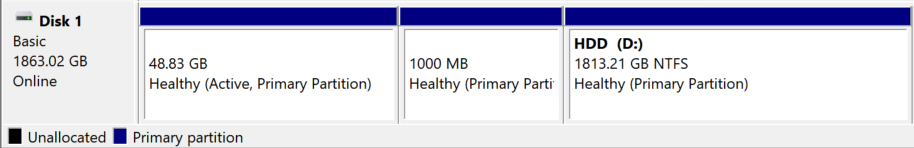
The reasoning behind this choice is similar to the one behind studying it a bit more in depth, meaning I was curious to know what a beginner-friendly distribution looked like and was curious as to why I had never heard about this distribution. Later on, if I enjoy it and get comfortable with disk partitioning, hopefully after question three of this lab, I might want to partition my external hard drive and add this distribution to it as a live system. For now, I will stick to the VirtualBox version of it, as I am still unfamiliar with partitioning disks, a friend helped me with my current one.



I did enjoy exploring Manjaro, and it was quite self-explanatory and easy to use, as it is described, but because I am unfamiliar with it, I had some difficulties with working it. For example, I changed the looks of it to darkmode, and learned that the right control button was the one I had to click to be able to leave the VM. Dolphin is their file manager and many themes were available to me for the OS.

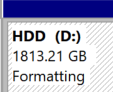
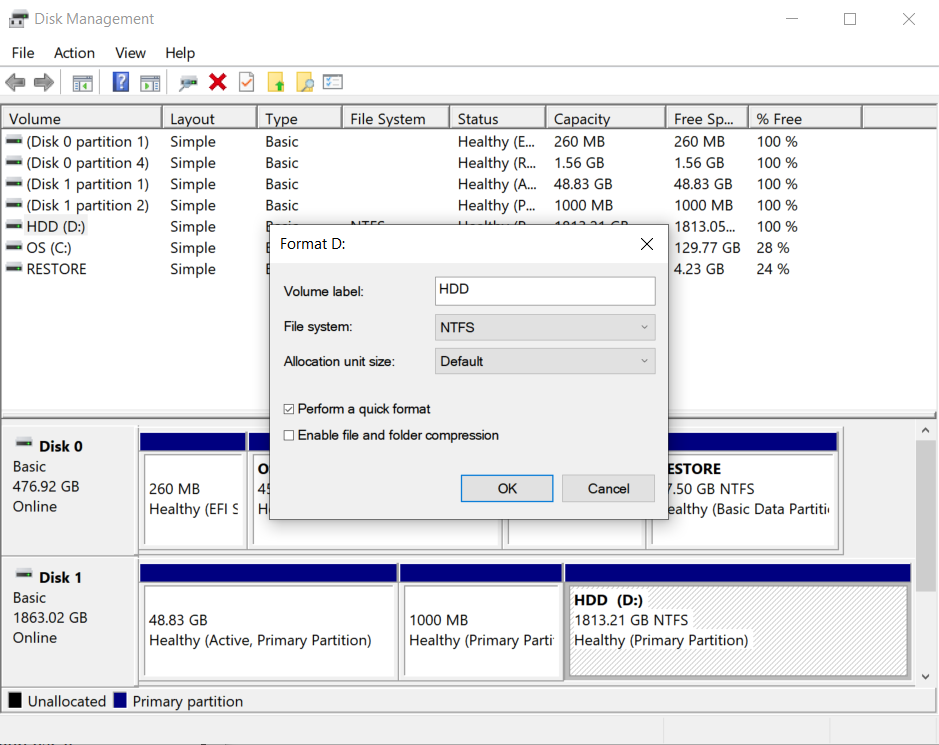
**Question 3)**

Original formatting of my HDD



After playing around with it

Right-click and press format:



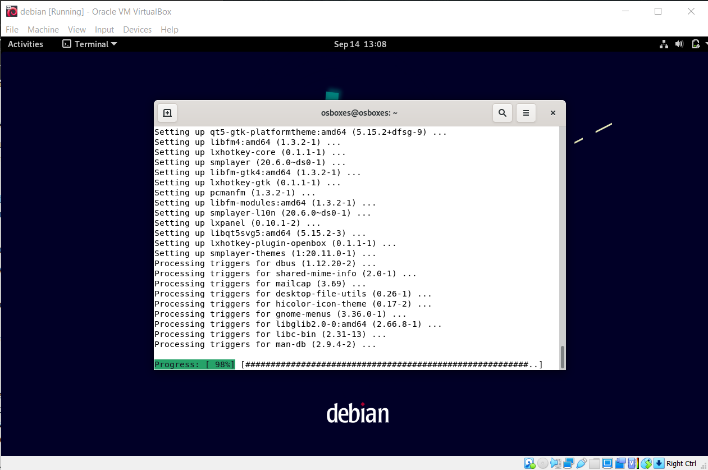
Once I was done – everything was like shown in the first picture

**Question 4)**

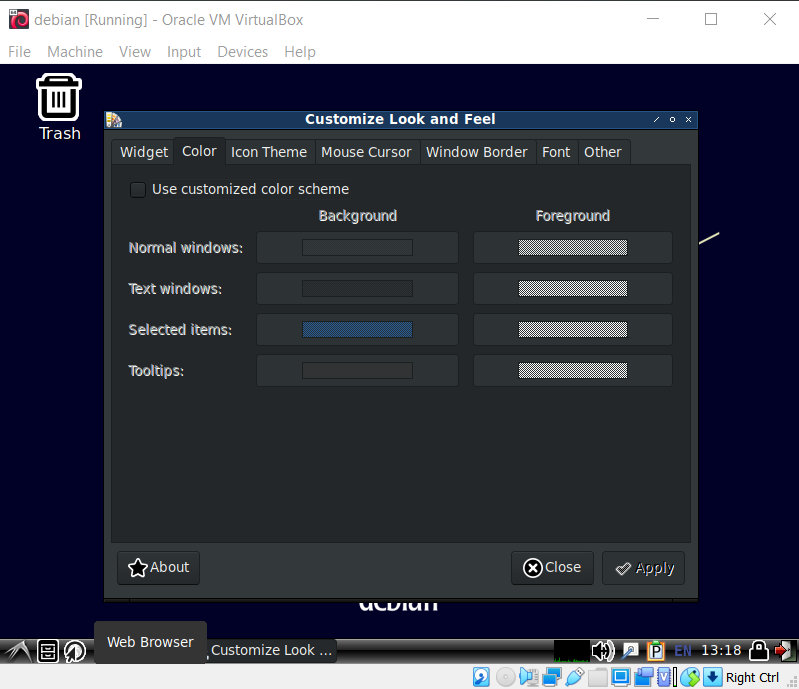
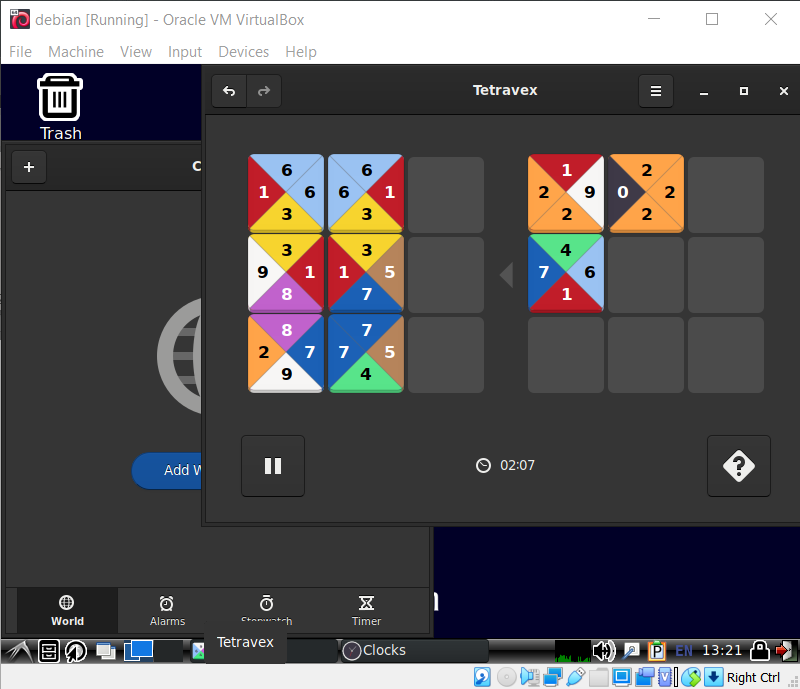
Until now, my impressions of the Debian live installer are listed in Lab01, but to recap, I enjoyed it, despite the slowness and discomfort of a new OS, although it did have some similarities to windows. The Debian Live install was a very slow process that took me many tries and I am not sure if I fully understood it. The Live image of Manjaro was fun to explore, but slightly difficult as I did not have a guide to follow like with Debian. I prefer Debian, but Manjaro is a solid choice if I were to decide to continue researching and learning a new distro.

Part 2 – Desktop Environments

**Screenshot no1) installing lxde**

****

**Screenshot no2) Different applications**

** **

**Question no13) Comparing different desktop environments**

If I am honest, I preferred Gnome3 to LXDE as it was easier to understand the original icons: LXDE was fun to customize and play around with, but I had difficulties figuring out where everything was at first which made my experience less fun than I would have liked. With Gnome3, I did not have any difficulties with beginning my exploration.

Part 3 – Shell commands

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Description** | **Explored Options** | **Log** |
| Su - | Brings you to the root – are running as an administrator | (none) | $ su  (password)  apt install lxde |
| Apt install | Is a command line interface that allows you to install a package | lxde | Apt install lxde |
| df | Is used to see the amount of free disk space there is left in a file system | -h  Outputs in human-readable format  -T  Adds a string type to each format | $ df -h  $ df -T |
| Stat | Displays the statistics of timestamps of files | (none) | $ stat Templates |
| touch | The touch command is used to create, edit or change timestamps for files | -d  Changes the last access and modification times to a date string instead of a time  -t  Specifies a timestamp in the form of [[CC]YY]MMDDhhmm[.ss]  -m  Changes the file’s modification time  -a  Changes the file’s access time | $ touch -d 2021-09-14 Templates  $ stat Templates  $ touch -t 09131200 Templates  $ stat Templates  $ touch -m Templates  $ stat Templates  $ touch -a Templates  $ stat Templates |
| Pwd | Prints the working directory | (none) | $ pwd |
| Mount  \*not understood | Mounts and displays information about file systems | -t | >After adding two files to Templates  $ mount -t Templates  >no output |
| Umount  \*not understood | Removes a remote file system that is currently mounted | (none) | $ umount Templates  >output: not mounted |
| Rmdir | Removes empty directories from the filesystem | -v  Displays verbose info for each processed directory | $ rmdir -v Videos |
| Rm | Removes each specified file | -i  Prompts before removal  -R  Removes the directory and content recursively  -f  Ignores nonexistent files and does not prompt | $ cd Testing1  $ rm -i file1.txt  Y  $ rm -R file2.docx  $ rm -f file3.txt |
| Cp | This command is used to copy files, group of files or directories and creates an exact image of said copies on a disk under a different file name | -n  -u  Copies the file only if its newer than destination  -i  Prompts for confirmation  -f  Removes and tries again when destination file cannot be removed  -r  Copies recursively  -p  Preserves file mode, ownership and timestamps | $ cd TestingCP  $ cp -p file4.txt file4copy2.txt  $ls  $ touch file4destination.txt  $ touch file4.txt  $ cp -u file4.txt file4.destination.txt  $ cp -i file1.txt file1copy.txt  $ls  $ cp -f file1.txt file1copy3.txt  $ls  $ cp -r file2.txt file2copy.txt  $ ls  $ cp -p file1.txt file1copy2.txt  $ls |
| Mv | Moves files or directories from one place to another | -n  -u  updates  -i  Prompts before overwriting  -f  Force move by overwriting without prompt | \*not done |