**Objectives**

1. Research information about software for a specific operating system (OS) environment. You will be assigned one of the operating systems form the list of: Windows, Mac OS, Linux. You will also be provided with a list of topics to investigate.
2. Organize your rough research information into a list of topics, sub-topics and facts. This process will involve identifying sub-topics, rearranging your rough research notes, and selecting (or highlighting) interesting facts.
3. Report a summary of your research in the form of a “concept map”. Use the PowerPoint template provided as a starting point. The concept map should only include the best and most interesting information from your organized research notes.

Your assigned operating system is:

* Windows
* Mac OS
* Linux
* iOS
* Android

A concept map can be created using the “Smart Ideas” application or PowerPoint or other applications.

**Level 1 – Rough Research**

Research information about the software for your assigned operating system (OS) environment.

* Guide your research according to the suggested topic list below
* Feel free to copy-and-paste as long as you keep track of your bibliographic references.
* Do not be too picky or concerned about formatting as you will organize this information later in step 2
* Select things that look interesting and don’t forget to include graphics images as well
* Upload your rough research notes to your repository when you are done.

**Topic A – Productivity, Entertainment & Other Software Applications**

This task manager shows you all the running processes and their memory consumption. You can choose to end a process from this task manager application. When you have just begun with Linux, you look for a **task manager equivalent in Linux** as well. An expert Linux user prefers the command line way to find processes and memory consumption etc but you don’t have to go that way, at least not when you are just starting with Linux. All major Linux distributions depends on your Linux have a task manager equivalent. Mostly, **it is called System Monitor** but it actually distribution and the [desktop environment](https://wiki.archlinux.org/index.php/desktop_environment) it uses.

<https://itsfoss.com/task-manager-linux/>

version *1.1.0*:

* § You can now filter notes by tag in the search box, including multiple tags. For example, if you wanted to search for notes tagged ‘travel’ and ‘poetry’, you would enter ‘tag:travel tag:poetry’ in the search bar.
* § We’ve made some UI improvements to the app, including better support for running the app at smaller screen sizes and a new placeholder view that shows when there are no notes to display in the app.
* § Many performance and reliability updates.
* § A security fix related to cross-site scripting in the markdown preview.
* version *1.1.0*:
* § You can now filter notes by tag in the search box, including multiple tags. For example, if you wanted to search for notes tagged ‘travel’ and ‘poetry’, you would enter ‘tag:travel tag:poetry’ in the search bar.
* § We’ve made some UI improvements to the app, including better support for running the app at smaller screen sizes and a new placeholder view that shows when there are no notes to display in the app.
* § Many performance and reliability updates.
* § A security fix related to cross-site scripting in the markdown preview.

a bug fix release that includes:

* A fix for notes sometimes not syncing when the connection went offline.
* Links now open in an external browser instead of an internal window.
* Small design improvements.
* History: drag the History slider to view previous versions.
* Collaboration: share your note with others and allow them to edit.
* Publishing: make your note public with its own URL.
* Tags: organize your notes effectively with tags.
* Pinning: pin notes right from the note list so they’re easy to find.

<https://simplenote.com/category/linux/>

[FocusWriter](https://gottcode.org/focuswriter/) is a text processor that creates a distraction-free environment for writers. It supports popular text formats and uses a hide-away interface to block out all distractions. You can select any visual and sound theme that works best for your productivity, and focus on your work. FocusWriter also allows you to set daily goals, use timers, alarms, and look into statistics.

[Osmo](https://sourceforge.net/projects/osmo-pim/) is a personal organizer. It includes various modules: calendar, notes, tasks list and reminder, and contacts. It is a lightweight and easy to use tool for managing all important personal information. The app can run both in an open window or in the background mode, and it doesn’t need an Internet connection.Osmo offers various configuration and formatting options for different types of information you record in it: addresses, birthdays, ideas, events, etc. Its handy search allows to find and access necessary information quickly and easily.

<https://www.fossmint.com/linux-productivity-tools/>

**Topic B – User Interface (Window Management & Input Devices)**

There’s no one true desktop environment for Linux. Unlike competing operating systems like Windows, Linux users have a choice of many different desktop environments, all with their own styles and strengths.

You can install one of these desktop environments after installing your Linux distribution and switch between desktop environments from the login screen. You can also choose to install a Linux distribution that comes with the desktop environment. For example, [you can get Ubuntu in many different flavors](https://www.howtogeek.com/127259/choose-your-ubuntu-8-ubuntu-derivatives-with-different-desktop-environments/).

1. [Unity](https://www.howtogeek.com/113330/how-to-master-ubuntus-unity-desktop-8-things-you-need-to-know/) is Ubuntu’s own default desktop environment. If you’ve installed Ubuntu using the standard installer, you’re probably using the Unity desktop right now.
2. GNOME was once the most popular Linux desktop environment. The GNOME 2.x series was used by default on Ubuntu, Fedora, Debian, and most other big Linux distributions.
3. At one point, KDE and GNOME were the two most popular Linux desktop environments. KDE has always been more complex than GNOME, packing in many more configuration options and features.
4. Xfce is a more lightweight desktop environment. It was once very similar to GNOME, but with GNOME 3 striking out in a different direction, Xfce now has its own identity as a more traditional Linux desktop environment that’s quite similar to GNOME 2
5. Cinnamon was developed for [Linux Mint](https://www.howtogeek.com/115041/htg-explains-whats-the-difference-between-ubuntu-linux-mint/). Cinnamon is based on GNOME 3, so it uses up-to-date libraries and other software — but it takes that software and tries to create a more traditional-looking desktop with it.
6. [MATE](https://www.howtogeek.com/110052/how-to-install-the-mate-desktop-go-back-to-gnome-2-on-ubuntu/) is a fork of the original GNOME 2 that aims to preserve GNOME 2, continually updating it so it will continue to work on modern Linux distributions. MATE has also seen some new features, but the main purpose of MATE is to give people who desperately miss GNOME 2 the opportunity to install it on new Linux distributions.
7. If you didn’t think Xfce was lightweight enough, try [LXDE](https://www.howtogeek.com/107368/how-to-install-the-lightweight-lxde-desktop-on-ubuntu/). LXDE is focused on being as lightweight as possible and is especially designed for older computers, netbooks, and other systems with low hardware resources
8. This isn’t a complete list — not by a long shot. There are many more niche desktop environments and window managers you could use, including [Xmonad, a tiling window manager](https://www.howtogeek.com/114728/how-to-use-xmonad-a-tiling-window-manager-for-linux/). Tiling window managers attempt to make your life easier by automatically arranging windows in tiles on your screen, saving you the trouble of dragging them around and allowing you to quickly rearrange them with keyboard shortcuts.

At some point in your career as a Linux administrator, you’re going to use Secure Shell (SSH) to remote into a Linux server or desktop. Chances are, you already have. In some instances, you’ll be SSH’ing into multiple Linux servers at once. In fact, Secure Shell might well be one of the most-used tools in your Linux toolbox. Because of this, you’ll want to make the experience as efficient as possible. For many admins, nothing is as efficient as the command line. However, there are users out there who do prefer a GUI tool, especially when working from a desktop machine to remote into and work on a server.

If you happen to prefer a good GUI tool, you’ll be happy to know there are a couple of outstanding graphical tools for SSH on Linux. Couple that with a unique terminal window that allows you to remote into multiple machines from the same window, and you have everything you need to work efficiently. Let’s take a look at these three tools and find out if one (or more) of them is perfectly apt to meet your needs.

I’ll be demonstrating these tools on [Elementary OS](https://elementary.io/), but they are all available for most major distributions.

### PuTTY

Anyone that’s been around long enough knows about [PuTTY](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html). In fact, PuTTY is the de facto standard tool for connecting, via SSH, to Linux servers from the Windows environment. But PuTTY isn’t just for Windows. In fact, from withing the standard repositories, PuTTY can also be installed on Linux. PuTTY’s feature list includes:

* Saved sessions.
* Connect via IP address or hostname.
* Define alternative SSH port.
* Connection type definition.
* Logging.
* Options for keyboard, bell, appearance, connection, and more.
* Local and remote tunnel configuration
* Proxy support
* X11 tunneling support

The PuTTY GUI is mostly a way to save SSH sessions, so it’s easier to manage all of those various Linux servers and desktops you need to constantly remote into and out of. Once you’ve connected, from PuTTY to the Linux server, you will have a terminal window in which to work. At this point, you may be asking yourself, why not just work from the terminal window? For some, the convenience of saving sessions does make PuTTY worth using.

**Topic C – Memory Allocation, Management,& Devices**

https://www.tecmint.com/command-line-tools-to-monitor-linux-performance/

* The definition of PC system tools are the programs used for software development or system maintenance. Virtually any program or utility that helps programmers or users develop applications or maintain their computers can be called a tool.
* Many system tools exist with the Linux OS.
* Linux Top, is a performance monitoring program which is used frequently by many system administrators to monitor Linux performance and it is available under many Linux/Unix like operating systems. The top command used to display all the running and active real-time processes in ordered list and updates it regularly. It display CPU usage, Memory usage, Swap Memory, Cache Size, Buffer Size, Process PID, User, Commands and much more. It also shows high memory and cpu utilization of a running processes. The top command is much useful for system administrator to monitor and take correct action when required.
* VMStat (Virtual Memory Statistics), Linux VmStat command is used to display statistics of virtual memory, kernerl threads, disks, system processes, I/O blocks, interrupts, CPU activity and much more. By default vmstat command is not available under Linux systems you need to install a package called sysstat that includes a vmstat program. The common usage of command format is.
* NetStat (Network Statistics), is a command line tool for monitoring incoming and outgoing network packets statistics as well as interface statistics. It is very useful tool for every system administrator to monitor network performance and troubleshoot network related problems.
* Monitorix, is a free lightweight utility that is designed to run and monitor system and network resources as many as possible in Linux/Unix servers. It has a built in HTTP web server that regularly collects system and network information and display them in graphs. It Monitors system load average and usage, memory allocation, disk driver health, system services, network ports, mail statistics (Sendmail, Postfix, Dovecot, etc), MySQL statistics and many more. It designed to monitor overall system performance and helps in detecting failures, bottlenecks, abnormal activities etc.

Memory allocation in the Linux kernel is complex, because there are significant constraints involved—and different ways of allocating memory have different constraints. This means that anyone writing Linux kernel code needs to understand the various ways of allocating memory, including the tradeoffs involved. This makes for for more efficient use of memory and CPU time—you can specify exactly what you need—but it also makes for more demanding programming.

There are essentially five different ways of allocating memory in the kernel. That's a white lie, but it is close enough to the truth for anyone who needs to read this article to learn about kernel memory allocation. Three (which provide dynamic allocation) are generally useful, and two (which provide static allocation) are deprecated, and are mostly historical artifacts that should not be used. We will discuss the advantages and limitations of the useful ways first, and will only briefly mention the two deprecated ways at the end of this article so that you know what to avoid.

Memory Allocation Strategies

There are a few rules that apply no matter what form of dynamic kernel memory allocation you attempt to do. Whenever you attempt to allocate memory in kernel space, you **must** be prepared for an allocation error. Always check the value returned from the allocation function, and if it is 0, you will need to handle it cleanly, somehow. User-space code can be terminated with a segmentation violation if it ignores memory allocation errors, but the kernel can easily crash, bringing down the whole system.

There are several common error-handling strategies. One strategy is to attempt to allocate critical memory at the top of a function, where you are less likely to have committed yourself and can more likely return an error cleanly. This is usually the best way to handle the problem.

Another strategy, usually used together with allocation at the top of the function, is to allocate an “easy” amount of memory for the memory management system to provide, and then parcel it out for various purposes during the life of the function, effectively doing its own memory management. Several subsystems in the kernel do this, such as the high-level SCSI drivers and the network code. Both include special memory allocation functions which are only supposed to be used in those subsystem. These are not documented here, under the assumption that documentation for those subsystems should document subsystem-specific memory allocation routines.

Yet another strategy, which will only work if you are not in “critical” sections of code, is to allow the kernel to schedule another process by calling **schedule()**and then to try again later, when schedule returns. Note that some kinds of allocation are not safe to call even once from within critical code; that will be covered when we discuss the individual functions.

The fundamental rule is not to write algorithms that commit themselves to completing without having been guaranteed the resources they need in order to complete. Memory is one of the scarcest and most commonly needed of the resources that must be guaranteed, and the only way to guarantee that memory will be available is to allocate it.

Kmalloc

The **kmalloc()** function allocates memory at two levels: it uses a “bucket” system to allocate memory in units up to nearly a page (4Kb on the i86) in length, and uses a “buddy” system on lists of different sizes of contiguous chunks of memory to allocate memory in units up to 128Kb (on the i86) in length. Only in recent kernels has it been able to allocate memory in units over 4Kb in length, and allocating large amounts of memory with kmalloc is very likely to fail, especially in low-memory situations, and especially on machines with less memory.

**Topic D – Process / Task Scheduling and Management (System Startup)**

* **The Bootloader:** The software that manages the boot process of your computer. For most users, this will simply be a splash screen that pops up and eventually goes away to boot into the operating system.
* **The kernel:** This is the one piece of the whole that is actually called “Linux”. The kernel is the core of the system and manages the CPU, memory, and peripheral devices. The kernel is the “lowest” level of the OS.
* **Daemons:** These are background services (printing, sound, scheduling, etc) that either start up during boot, or after you log into the desktop.
* **The Shell:** You’ve probably heard mention of the Linux command line. This is the shell – a command process that allows you to control the computer via commands typed into a text interface. This is what, at one time, scared people away from Linux the most (assuming they had to learn a seemingly archaic command line structure to make Linux work). This is no longer the case. With modern desktop Linux, there is no need to ever touch the command line.
* **Graphical Server:** This is the sub-system that displays the graphics on your monitor. It is commonly referred to as the X server or just “X”.
* **Desktop Environment:** This is the piece of the puzzle that the users actually interact with. There are many desktop environments to choose from (Unity, GNOME, Cinnamon, Enlightenment, KDE, XFCE, etc). Each desktop environment includes built-in applications (such as file managers, configuration tools, web browsers, games, etc).
* **Applications:** Desktop environments do not offer the full array of apps. Just like Windows and Mac, Linux offers thousands upon thousands of high-quality software titles that can be easily found and installed. Most modern Linux distributions (more on this in a moment) include App Store-like tools that centralize and simplify application installation. For example: Ubuntu Linux has the Ubuntu Software Center (Figure 1) which allows you to quickly search among the thousands of apps and install them from one centralized location.

<https://www.linux.com/what-is-linux>

A typical Linux server runs in a command line interface (CLI) environment, preloaded with bare essential tools needed to install and configure various headless services. Compared to a full-blown GUI desktop image, such a minimal setup is advantageous in terms of security, resource consumption and speed.

<https://www.linux.com/learn/how-manage-linux-server-gui>

**Topic E – Software Security, Updates & System Tools**

1. [ClamAV](http://www.clamav.net/lang/en/)

My favorite antivirus software for Linux is Sourcefire's ClamAV, a free, [open source](http://www.pcworld.com/businesscenter/article/209891/10_reasons_open_source_is_good_for_business.html) package designed to detect Trojans, viruses, malware and other malicious threats. Included in the software, which now comes preinstalled in several [Linux distributions](http://www.pcworld.com/businesscenter/article/204767/a_guide_to_todays_top_10_linux_distributions.html), are a multithreaded scanning daemon, command line utilities for on-demand file scanning, and an intelligent tool for automatic signature updates. Of particular note for past or current Windows users is that the core ClamAV library is also used in [Immunet 3.0](http://www.immunet.com/), a sister solution for Microsoft's operating system.

2. [Snort](http://www.snort.org/)

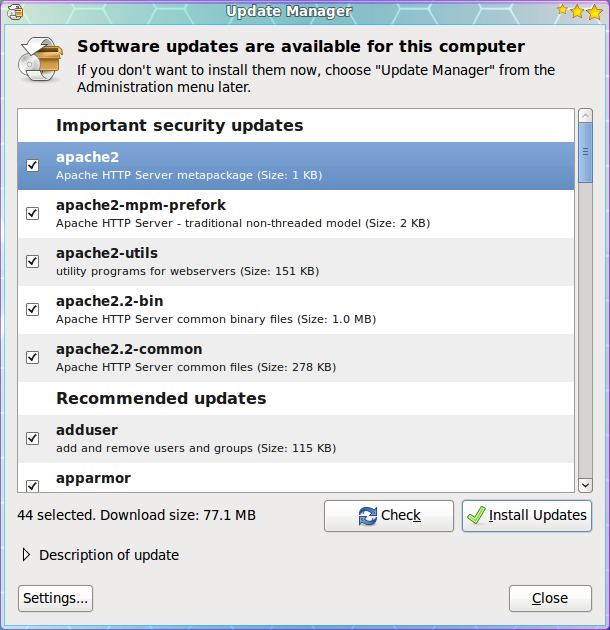
Also offered by Sourcefire is Snort, an open source network intrusion prevention and detection system that combines the benefits of signature, protocol and anomaly-based inspection. With millions of downloads and more than 300,000 registered users to its credit, Snort is the most widely deployed such technology worldwide, Sourcefire says.

**3. [Wireshark](http://www.wireshark.org/)**

Wireshark is a network protocol analyzer that lets you capture and interactively browse the traffic running on a computer network. The software runs not just on Linux but on Windows, OS X, Solaris, FreeBSD and NetBSD, as well. Captured network data can be browsed via GUI or via the TTY-mode TShark utility.

<https://www.pcworld.com/article/224955/7_free_security_tools_for_linux.html>

Ubuntu Linux has become one of the most popular of all the Linux distributions. And through the process of updating a system, you should be able to tell exactly why this is the case. Ubuntu is very user friendly. Ubuntu uses two different tools for system update:

* apt-get: Command line tool.
* Update Manager: GUI tool.

The Update Manger is a nearly 100% automatic tool. With this tool you will not have to routinely check to see if there are updates available. Instead you will know updates are available because the Update Manager will open on your desktop (see Figure 1) as soon as the updates depending upon their type:

* Security updates: Daily
* Non-security updates: Weekly

If you want to manually check for updates, you can do this by clicking the Administration sub-menu of the System menu and then selecting the Update Manager entry. When the Update Manager opens click the Check button to see if there are updates available.

Figure 1 shows a listing of updates for a Ubuntu 9.10 installation. As you can see there are both *Important Security Updates* as well as *Recommended Update*. If you want to get information about a particular update you can select the update and then click on the *Description of update* dropdown.

In order to update the packages follow these steps:

1. Check the updates you want to install. By default all updates are selected.
2. Click the Install Updates button.
3. Enter your user (sudo) password.
4. Click OK.

The updates will proceed and you can continue on with¬† your work. Now some updates may require either you to log out of your desktop and log back in, or to reboot the machine. There are is a new tool in development ([Ksplice](http://www.ksplice.com/))¬† that allow even the update of a kernel to not require a reboot.

Once all of the updates are complete the Update Manage main window will return reporting that *Your system is up to date*.

Now let's take a look at the command line tools for updating your system. The Ubuntu package management system is called *apt*. Apt is a very powerful tool that can completely manage your systems packages via command line. Using the command line tool has one drawback - in order to check to see if you have updates, you have to run it manually. Let's take a look at how to update your system with the help of Apt. Follow these steps:

1. Open up a terminal window.
2. Issue the command *sudo apt-get upgrade*.
3. Enter your user's password.
4. Look over the list of available updates (see Figure 2) and decide if you want to go through with the entire upgrade.
5. To accept all updates click the 'y' key (no quotes) and hit Enter.
6. Watch as the update happens.

That's it. Your system is now up to date.

<https://www.linux.com/learn/linux-101-updating-your-system>

**Topic F – File System & User Accounts**

Linux/Unix operating systems have the ability to multitask in a manner similar to other operating systems. However, Linux’s major difference from other operating systems is its ability to have multiple users. Linux was designed to allow more than one user to have access to the system at the same time. In order for this multiuser design to work properly, there needs to be a method to protect users from each other. This is where permissions come in to play.

* To view the permissions on a file or directory, issue the command ls -l <directory/file>. Remember to replace the info
* rmation in the < > with the actual file or directory name. Below is sample output for the ls command:
* -rw-r--r-- 1 root root 1031 Nov 18 09:22 /etc/passwd
* The first ten characters show the access permissions. The first dash (-) indicates the type of file (d for directory, s for special file, and - for a regular file). The next three characters (rw-) define the owner’s permission to the file. In this example, the file owner has read and write permissions only. The next three characters (r–) are the permissions for the members of the same group as the file owner (which in this example is read only). The last three characters (r–) show the permissions for all other users and in this example it is read only.

<https://www.linode.com/docs/tools-reference/linux-users-and-groups/>

### Three files defines a user account

### /etc/passwd file

ram:x:1003:1003:ram kumar:/home/ram:/bin/bash  
charvi:x:1004:1004:charvi gite:/home/charvi:/bin/bash  
vivek:x:1005:1005:vivek gite:/home/vivek:/bin/bash

### /etc/group file

prouser:x:1001:vivek,charvi  
ram:x:1003:  
charvi:x:1004:  
Vivek:x:1005:

### /etc/shadow file

ram:$1$4tuS/iYO$CsmE8cD7j/96ca7K3gJ9Y/:13061:0:99999:7:::  
charvi:$1$4tuS/iYO$CsmE8cD7j/96ca7K3gJ9Y/:13066:0:99999:7:::  
vivek:$1$4tuS/iYO$CsmE8cD7j/96ca7K3gJ9Y/:13064:0:99999:7:::

<https://www.cyberciti.biz/faq/linux-what-defines-a-user-account/>

### **Structure**

It makes sense to explore the Linux filesystem from a terminal window, not because the author is a grumpy old man and resents new kids and their pretty graphical tools -- although there is some truth to that -- but because a terminal, despite being text-only, has better tools to show the map of Linux's directory tree.

In fact, that is the name of the first tool you'll install to help you on the way: *tree*. If you are using Ubuntu or Debian, you can do:

sudo apt install tree

On Red Hat or Fedora, do:

sudo dnf install tree

For SUSE/openSUSE use zypper:

sudo zypper install tree

For Arch-like distros (Manjaro, Antergos, etc.) use:

sudo pacman -S tree

... and so on.

Once installed, stay in your terminal window and run *tree* like this:

tree /

The / in the instruction above refers to the *root* directory. The root directory is the one from which all other directories branch off from. When you run tree and tell it to start with */*, you will see the whole directory tree, all directories and all the subdirectories in the whole system, with all their files, fly by.

If you have been using your system for some time, this may take a while, because, even if you haven't generated many files yourself, a Linux system and its apps are always logging, cacheing, and storing temporal files. The number of entries in the file system can grow quite quickly.

Don't feel overwhelmed, though. Instead, try this:

tree -L 1 /

And you should see what is shown in Figure 1.

#### ***/media***

The */media* directory is where external storage will be automatically mounted when you plug it in and try to access it. As opposed to most of the other items on this list, */media* does not hail back to 1970s, mainly because inserting and detecting storage (pendrives, USB hard disks, SD cards, external SSDs, etc) on the fly, while a computer is running, is a relatively new thing.

#### ***/mnt***

The */mnt* directory, however, is a bit of remnant from days gone by. This is where you would manually mount storage devices or partitions. It is not used very often nowadays.

#### ***/opt***

The */opt* directory is often where software you compile (that is, you build yourself from source code and do not install from your distribution repositories) sometimes lands. Applications will end up in the */opt/bin* directory and libraries in the */opt/lib* directory.

A slight digression: another place where applications and libraries end up in is */usr/local*, When software gets installed here, there will also be */usr/local/bin* and */usr/local/lib* directories. What determines which software goes where is how the developers have configured the files that control the compilation and installation process.

<https://www.linux.com/blog/learn/intro-to-linux/2018/4/linux-filesystem-explained>

**Topic G – Special Features of your OS**

* Kernel − Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
* System Library − System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries implement most of the functionalities of the operating system and do not requires kernel module's code access rights.
* System Utility − System Utility programs are responsible to do specialized, individual level tasks.

<https://www.tutorialspoint.com/operating_system/os_linux.htm>

**Customized Keyboards**

Linux OS has many users across the globe, so they provided it to be available in multiple languages and customer support. This feature includes customized keyboards that have the accessible languages for different nations. With this, it is a user – friendly and easy to use a system that meets everybody’s standards.

**Live CD Or USB**

Most of all the distributed Linux systems come with a Live CD or USB feature in which a user can use and run the operating system without the need of installing it to your computer or laptop. You can be assured that Linux can offer you a wide range of options based on your requirements.

**Application Support**

A Linux OS comes with a software repository in which a user can easily download or install a huge amount of applications by just providing a command to the terminal or shell of Linux. Linux OS can also be possible to run a Windows application.

**Graphical User Interface**

If you think Linux is only a command line operating system, you might be true but not really to its extent. You should know that Linux comes with packages in which it is possible to install to make its complete operating system graphics to be based on Windows.

**Good Security**

This feature can also be a good characteristic of Linux OS because it prioritizes the protection of the confidential works of the users from hackers or persons who are not authorized to enter their system. Linux provided some security concepts which include Authentication, Authorization, and especially Encryption. With these security concepts, it only goes to show that Linux OS is safe to use and prevent the access of any unauthorized people that might have bad intentions.

With all the information stated above, you now know what are the best features of Linux Operating System. You should also know that these features make it perform to its maximum and will surely amaze every user of it. In this way, you can be assured that you can have the excellent and high-quality operating system that you wish you can have. If you want to know more about Linux Operating System, you can ask an expert on this topic or visit their website.

<https://www.linuxchange.org/5-best-features-of-linux-operating-system-that-you-need-to-know/>

Linux was one of the first open-source technologies, but many programmers have contributed and added software that’s completely open-source for any user. This means that you can download the source code and change it any way you like. Some developers have restrictions on how you can distribute the code. For instance, some developers allow you to change the code, but you cannot distribute it for money.

One main advantage of [open-source](http://opensource.org/osd-annotated) technologies such as Linux is the wide range of options available to users and the increased security. With Linux being open-source, several distributions are available to the end-user. Debian, Fedora, Ubuntu and Mint are just a few of the distributions available to end users, and these distributions are completely free to download.

Security is the other main advantage. Several whitehat hackers have contributed to the overall security of Linux, and by making the source available to anyone, security experts can help identify any main security flaws in the operating system. The advantage over operating systems such as Windows is that security flaws are caught before they become an issue for the public.

<https://blog.storagecraft.com/linux-advantages-disadvantages-open-source-technology/>

**Topic H – Limitations of your OS**

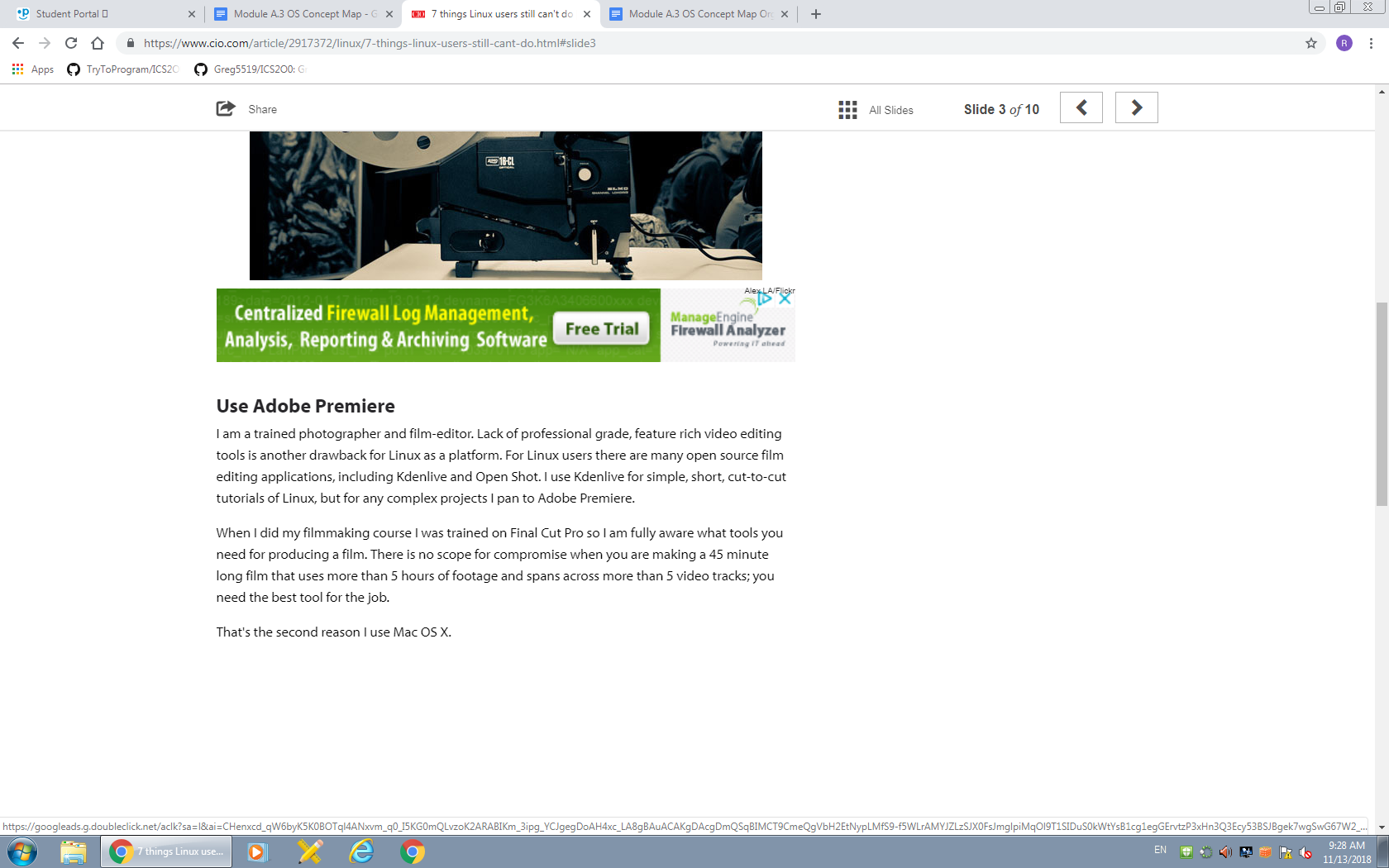
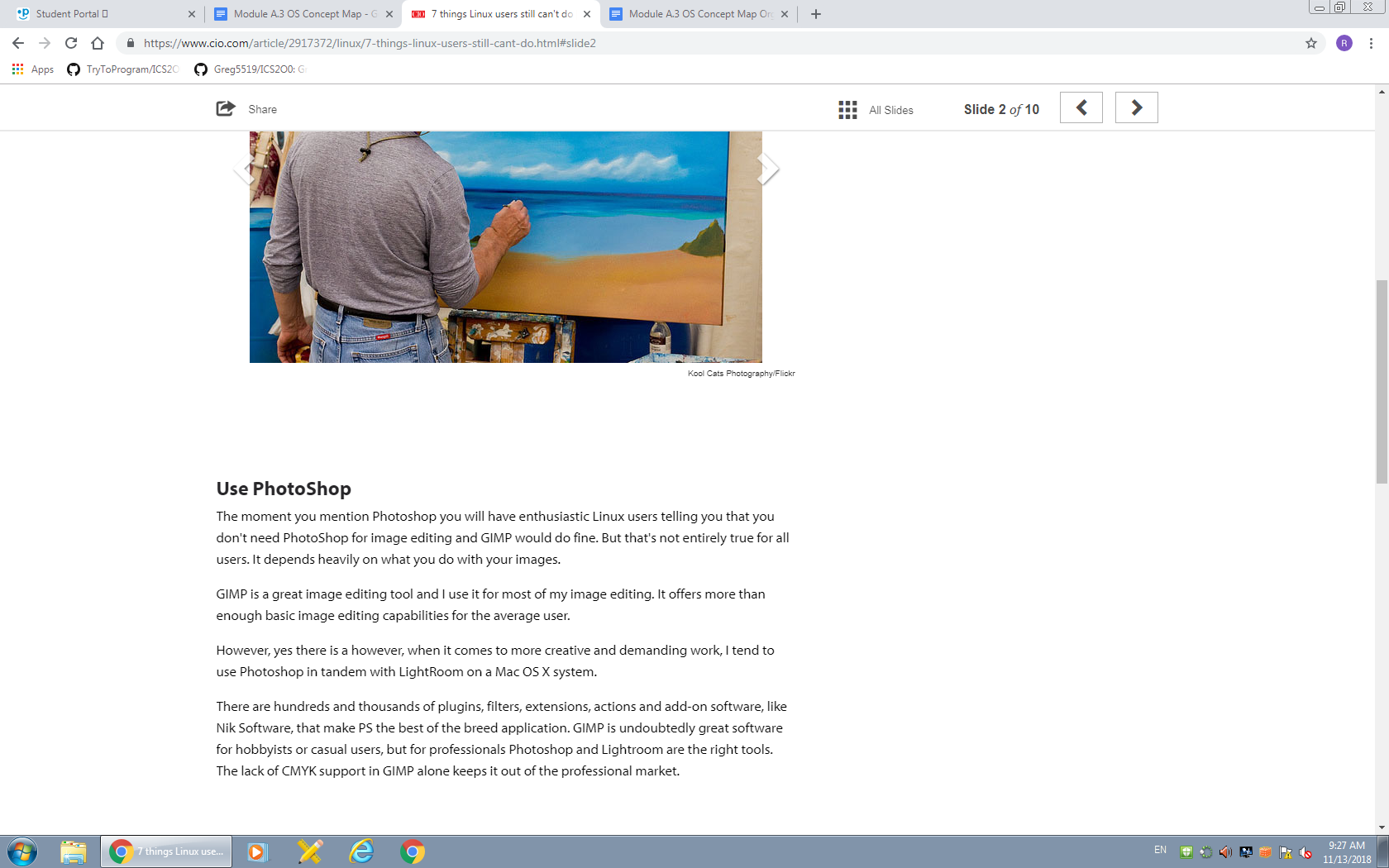
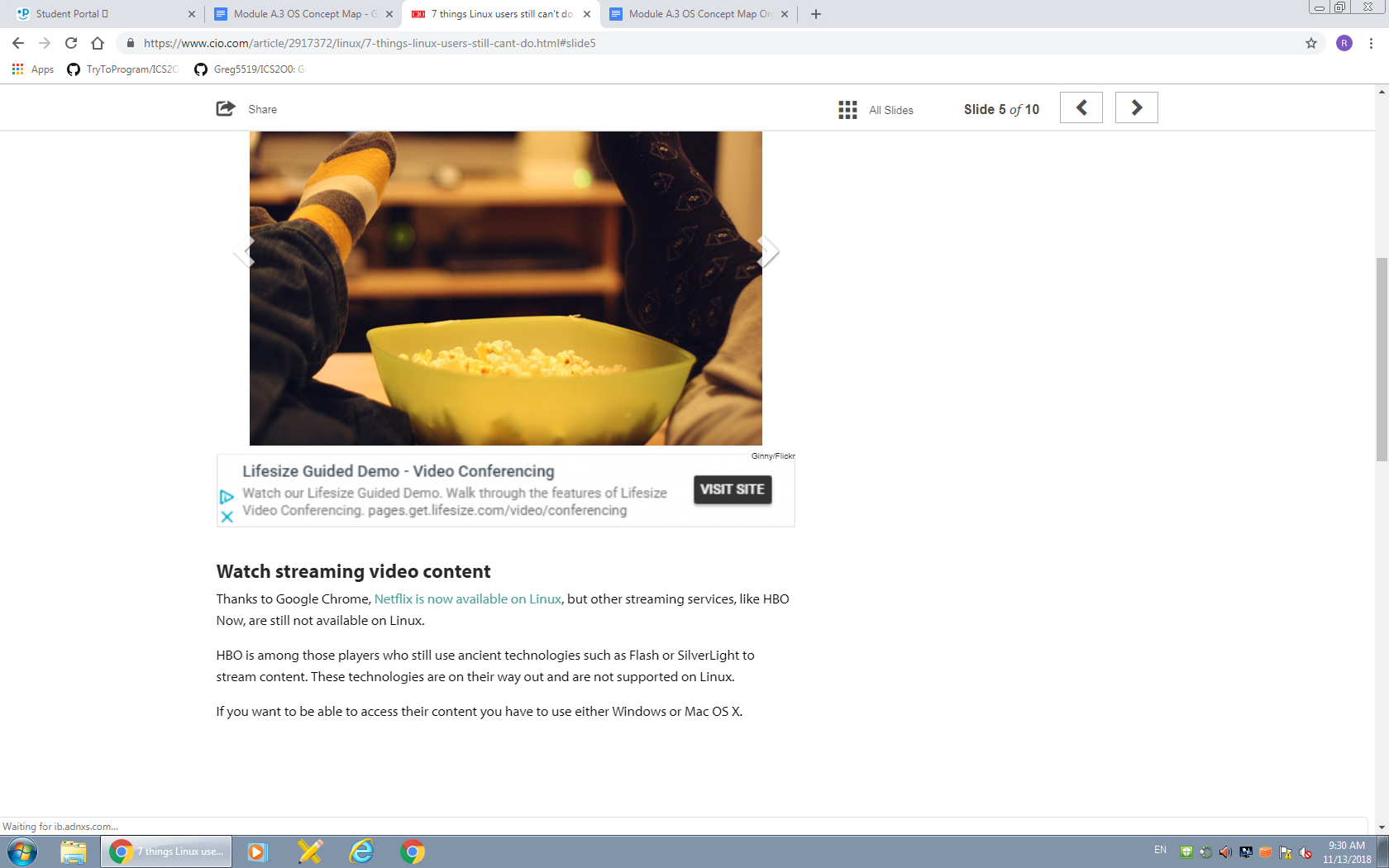
Because Linux does not dominate the market like Windows, there are some disadvantages to using the operating system. First, it’s more difficult to find applications to support your needs. This is an issue for mostly businesses, but more programmers are developing applications that are supported by Linux. Many more applications are available for the working world compared to what was available a decade ago.

One main issue with Linux is drivers. Before you can install any hardware component in your computer, you must make sure the hardware has drivers available. Hardware manufacturers usually write drivers for Windows, but not all brands write drivers for Linux. This means that some of your hardware might not be compatible with Linux if you decide to switch.

Support for open-source can also be an issue. While there are plenty of Windows support people, Linux is not supported out-of-the-box. The way Linux distribution companies make money is through their support channels. This means that companies must pay fees for support, if they cannot solve an issue. However, there are plenty of forums and blogs that support Linux issues. If your company has a good Linux administrator, the administrator can typically find answers through one of these free channels without paying for support.

Before you decide on open-source technology, make sure you have the resources and personnel to support the software.

<https://blog.storagecraft.com/linux-advantages-disadvantages-open-source-technology/>

[****](https://www.cio.com/article/2917372/linux/7-things-linux-users-still-cant-do.html#slide2)

<https://www.cio.com/article/2917372/linux/7-things-linux-users-still-cant-do.html#slide2>

**Level 2 – Organized Research**

Organize your rough research information to provide more stricture and meaning.

* Re-read your rough research to identify (highlight) important sub-topics and facts
* Rearrange (cut–and-paste) your rough research so that related sub topics and facts are next to each other.
* Your finished organization should look like the template provided below.
* Upload your rough research notes to your repository when you are done.

Suggested organization template:

**Topic A – Productivity,  Entertainment & Other Software Applications**

* **Sub- Topic: Simplenote**
  + You can now filter notes by tag in the search box, including multiple tags. For example, if you wanted to search for notes tagged ‘travel’ and ‘poetry’, you would enter ‘tag:travel tag:poetry’ in the search bar.
  + A security fix related to cross-site scripting in the markdown preview
  + We’ve made some UI improvements to the app, including better support for running the app at smaller screen sizes and a new placeholder view that shows when there are no notes to display in the app.
* **Sub- Topic: Focus Writer**
  + A text processor that creates a distraction-free environment for writers, it supports popular text formats and uses a hide-away interface to block out all distractions
  + You can select any visual and sound theme that works best for your productivity, and focus on your work
  + Also allows you to set daily goals, use timers, alarms, and look into statistics
* **Sub- Topic: Osmo**
  + A personal organizer, includes various modules: calendar, notes, tasks list and reminder, and contacts. It is a lightweight and easy to use tool for managing all important personal information
  + App can run both in an open window or in the background mode, and it doesn’t need an Internet connection
  + Offers various configuration and formatting options for different types of information you record in it: addresses, birthdays, ideas,

**Topic B – User Interface (Window Management & Input Devices)**

* **Sub- Topic: Variety of Desktop Environments**
  + Unity
  + GNOME
  + KDE
  + Xfce
  + Cinnamon
  + MATE
  + LXDE
  + Xmonad
* **Sub- Topic: User Friendly**
  + Saved sessions
  + Proxy support
  + Server connection via IP address or host name

**Topic C – Memory Allocation, Management,& Devices**

* **Sub- Topic: Tools**
  + Linux Top, can show performance of PC
  + VMStat, displays statistics of memory, disks and processors
  + NetStat, displays data to incoming and outgoing network
* **Sub- Topic: Memory Allocation**
  + Dynamic Kernel Memory Allocation
  + Error-handling
  + High-level SCSI drivers

**Topic D – Process / Task Scheduling and Management (System Startup)**

* **Sub- Topic: The Shell**
  + a command process that allows you to control the computer via commands typed into a text interface
  + at one time, scared people away from Linux the most (assuming they had to learn a seemingly archaic command line structure to make Linux work)
  + With modern desktop Linux, there is no need to ever touch the command line
* **Sub- Topic: Desktop Environment** 
  + There are many desktop environments to choose from (Unity, GNOME, Cinnamon, Enlightenment, KDE, XFCE, etc)
  + Each desktop environment includes built-in applications (such as file managers, configuration tools, web browsers, games, etc)
  + the piece of the puzzle that the users actually interact with
* **Sub- Topic: Applications**
  + offers thousands upon thousands of high-quality software titles that can be easily found and installed
  + Most modern Linux distributions (more on this in a moment) include App Store-like tools that centralize and simplify application installation
  + allows you to quickly search among the thousands of apps and install them from one centralized location

**Topic E – Software Security, Updates & System Tools**

* **Sub- Topic: ClamAV**
  + a free, [open source](http://www.pcworld.com/businesscenter/article/209891/10_reasons_open_source_is_good_for_business.html) package designed to detect Trojans, viruses, malware and other malicious threats included in software
  + a multithreaded scanning daemon, command line utilities for on-demand file scanning, and an intelligent tool for automatic signature updates
  + particular note for past or current Windows users is that the core ClamAV library is also used in [Immunet 3.0](http://www.immunet.com/), a sister solution for Microsoft's operating system
* **Sub- Topic: Snort**
  + an open source network intrusion prevention and detection system
  + that combines the benefits of signature, protocol and anomaly-based inspection
  + ith millions of downloads and more than 300,000 registered users to its credit, Snort is the most widely deployed such technology worldwide, Sourcefire says
* **Sub- Topic: Manually Checking**
  + If you want to manually check for updates, you can do this by clicking the Administration sub-menu of the System menu and then selecting the Update Manager entry.
  + shows a listing of updates for a Ubuntu 9.10 installation. As you can see there are both *Important Security Updates* as well as *Recommended Update*
  + If you want to get information about a particular update you can select the update and then click on the *Description of update* dropdown.

**Topic F – File System & User Accounts**

* **Sub- Topic: File System**
  + Linux’s major difference from other operating systems is its ability to have multiple users
  + Linux was designed to allow more than one user to have access to the system at the same time. In order for this multiuser design to work properly, there needs to be a method to protect users from each other. This is where permissions come in to play.
  + To view the permissions on a file or directory, issue the command ls -l <directory/file>.
* **Sub- Topic: Tree**
  + but because a terminal, despite being text-only, has better tools to show the map of Linux's directory tree
  + In fact, that is the name of the first tool you'll install to help you on the way: *tree*. If you are using Ubuntu or Debian, you can do
  + The / in the instruction above refers to the *root* directory. The root directory is the one from which all other directories branch off from. When you run tree and tell it to start with */*, you will see the whole directory tree, all directories and all the subdirectories in the whole system, with all their files, fly by.
* **Sub- Topic: /opt**
  + is often where software you compile (that is, you build yourself from source code and do not install from your distribution repositories) sometimes lands
  + Applications will end up in the */opt/bin* directory and libraries in the */opt/lib* directory.
  + A slight digression: another place where applications and libraries end up in is */usr/local*, When software gets installed here, there will also be */usr/local/bin* and */usr/local/lib* directories. What determines which software goes where is how the developers have configured the files that control the compilation and installation process.

**Topic G – Special Features of your OS**

* **Sub- Topic: Good Security**
  + This feature can also be a good characteristic of Linux OS because it prioritizes the protection of the confidential works of the users from hackers or persons who are not authorized to enter their system.
  + Linux provided some security concepts which include Authentication, Authorization, and especially Encryption. With these security concepts, it only goes to show that Linux OS is safe to use and prevent the access of any unauthorized people that might have bad intentions.
  + Security is the other main advantage. Several whitehat hackers have contributed to the overall security of Linux, and by making the source available to anyone, security experts can help identify any main security flaws in the operating system
* **Sub- Topic: Application Support**
  + A Linux OS comes with a software repository in which a user can easily download or install a huge amount of applications
  + Linux OS can also be possible to run a Windows application.
  + providing a command to the terminal or shell of Linux
* **Sub- Topic: Graphical User Interface**
  + If you think Linux is only a command line operating system, you might be true but not really to its extent.
  + You should know that Linux comes with packages in which it is possible to install to make its complete operating system graphics to be based on Windows.
  + **Live CD Or USB:** Most of all the distributed Linux systems come with a Live CD or USB feature in which a user can use and run the operating system without the need of installing it to your computer or laptop. You can be assured that Linux can offer you a wide range of options based on your requirements.

**Topic H – Limitations of your OS**

* **Sub- Topic: Domination**
  + it’s more difficult to find applications to support your needs
  + an issue for mostly businesses, but more programmers are developing applications that are supported by Linux
  + Many more applications are available for the working world compared to what was available a decade ago.
* **Sub- Topic: Drivers**
  + Before you can install any hardware component in your computer, you must make sure the hardware has drivers available.
  + Hardware manufacturers usually write drivers for Windows, but not all brands write drivers for Linux.
  + This means that some of your hardware might not be compatible with Linux if you decide to switch.
* **Sub- Topic: Support for open-source**
  + While there are plenty of Windows support people, Linux is not supported out-of-the-box.
  + The way Linux distribution companies make money is through their support channels. This means that companies must pay fees for support, if they cannot solve an issue. However, there are plenty of forums and blogs that support Linux issues.
  + If your company has a good Linux administrator, the administrator can typically find answers through one of these free channels without paying for support.

**Level 3 – Concept Map**

Create a “concept map” as a final report of your organized research.

Use the PowerPoint template provided as a starting point.

You can use PowerPoint or another concept mapping tool of your choice.

Select the best and most interesting information from your organized research.

Summarize and edit your information to fit on the concept map.

Share your finished concept map with Mr. Nestor at p0079141@pdsb.net

A concept map can be created using the “Smart Ideas” application or PowerPoint or other applications. A concept map template can be downloaded from the “Topic A” folder on the class GitHub repository



LINK TO CONCEPT MAP:

<https://atlas.mindmup.com/2019/05/ab2719907b4f11e98d947db8aab6a980/linux_os/index.html>