**Questions and Answers:**

**Day 1**

1. **What is the GNU project?**

GNU stands for *GNU's not Unix.* The GNU project is a mass collaborative initiative for the development of [free software](https://whatis.techtarget.com/definition/free-software). The original purpose of the GNU project was the creation of a free operating system. Free, in a software context, does not necessarily mean free of cost. The freedom referred to is the ability of anyone who wishes to run, copy, distribute, study, change and improve the software. According to the GNU Linux project, the [Linux](https://searchdatacenter.techtarget.com/definition/Linux-operating-system) operating system's [kernel](https://searchdatacenter.techtarget.com/definition/kernel) is Linux but all other elements of the system are GNU. According to the project, neither OS exists independently and, as such Linux would more appropriately be called GNU Linux.

**2. Difference between Unix and Linux?**

* Linux is an open-source operating system which is freely available to everyone. Whereas Unix is an operating system which can be only used by its copywriters.
* Linux is just the kernel, whereas Unix is a complete package of Operating systems.
* BASH is the Linux default shell. It offers support for multiple command interpreters. Unix Originally made it work in Bourne Shell. However, it is now compatible with many other software.
* Linux source is available to the general public whereas unix source code is not available to anyone.

**3. Integrity Check of BIOS?**

Drive encryption programs including TrueCrypt, PGP, and BitLocker) adopt a preboot authentication that is launched during the boot process as an extension of the BIOSbefore the operating system (such as Windows, Linux, Android, iOS, and so forth) is loaded.v

The problem is the lack of end-to-end protection. Most software solutions are available only after being loaded by the operating system. In other words, during the boot process—that is, from the moment a user presses the power button to when the operating system takes control and finishes loading the security solutions—the computer is not benefiting from the services offered by the security measures and is hence vulnerable. Drive encryption schemes that start during the boot do not depend on the operating system to function, but they do rely on the integrity of the boot loader that loads them.

Instead, the BIOS and boot loader are becoming more interesting targets. A number of such attacks were published in recent years. Here are two examples:

* ***Attacking BIOS*:** This type of attack replaces an authentic BIOS with an attacker’s BIOS that contains malicious code. There have been attacks against the UEFI (Unified Extensible Firmware Interface) secure boot.
* ***Attacking boot loader*:** This type of attack usually installs a boot kit (a variant of root kits that runs in the kernel mode) under an attacker’s control that infects the boot loader. The boot kit can be used to steal secrets during the boot path; for example, logging the user’s drive encryption password.

**4. What is UEFI? Difference between UEFI and BIOS?**

The BIOS (basic input/output system) is a firmware component stored in nonvolatile memory, usually a flash chip.The BIOS loads the boot loader, which is the first software component loaded during the boot process.

The boot loader is stored in the hard drive, together with the operating system and applications.

UEFI stands for Unified Extensible Firmware Interface. It does the same job as a BIOS, but with one basic difference: it stores all data about initialization and startup in an .efi file, instead of storing it on the firmware.

UEFI was designed to overcome many limitations of the old BIOS, including:

* UEFI supports drive sizes upto 9 zettabytes, whereas BIOS only supports 2.2 terabytes.
* UEFI has discrete driver support, while BIOS has drive support stored in its ROM, so updating BIOS firmware is a bit difficult.
* UEFI offers security like "Secure Boot", which prevents the computer from booting from unauthorized/unsigned applications. This helps in preventing rootkits, but also hampers dual-booting, as it treats other OS as unsigned applications.
* UEFI runs in 32bit or 64bit mode, whereas BIOS runs in 16bit mode. So UEFI is able to provide a GUI (navigation with mouse) as opposed to BIOS which allows navigation only using the keyboard.

**5.when should I go for Ubuntu & when for other systems?**

1. Free of Cost

Downloading and installing Ubuntu is free, and costs only time to install it. One can just download from the internet or can create an ISO bootable disk, and after it, the environment is ready to launch.

2. Privacy

In comparison to Windows, [Ubuntu provides a better option](https://www.educba.com/install-ubuntu/) for privacy and security. The best advantage of having Ubuntu is that we can acquire the required privacy and additional security without having any third party solution. Risk of hacking and various other attacks can be minimized by using this distribution.

3. Working with Partitions of hard drives

Linux live CD and Gparted enable to divide the hard disk into partitions. This feature is also applicable when your system is a dual or triple boot. Otherwise, the partition is helpful in migration to solid state drive or more spacious drive. Apart from this, wiping of data can be easily done in Ubuntu.

4. Free Apps

Ubuntu provides a lot of free apps to explore the world of opportunities, refreshments, and enjoyment. Following are few apps of total listed below

* Spotify
* Skype
* Firefox
* Telegram

5. User-Friendly

Ubuntu is as easy as the [Windows operating system](https://www.educba.com/windows-operators/). There is a well-known myth about Ubuntu is that Ubuntu is developed for developers and coders only. But the actual fact is, Ubuntu is a beautiful desktop operating system and can be used by any common computer user.

6 Say Bye to Antivirus

Risk of having a virus or [malware is minimal](https://www.educba.com/what-is-malware/), that reduces the cost of anti-virus software. Antivirus is also a reason for the slowness of the computer system and affects performance badly

**6. What are the various operating systems & their uses?**

**MS-DOS :** which is short for Microsoft Disk Operating System is a non-graphical command line operating system developed for IBM compatible computers with x86 microprocessor. The operating system used a command line interface for the user to input commands to navigate, open and manipulate files on their computer.

* It is a single user operating system meaning only one user can operate at a time.
* It is a lightweight operating system allowing users to have direct access to the BIOS and its underlying hardware.
* Enables the computer to perform input and output operations such as taking commands from keyboard, printing information on the screen.
* It is very helpful in file management like creating, editing, deleting files, etc.
* It also controls and manages other external devices such as the printer, keyboard or external hard drive using various drive utilities.

**Windows Operating System:**

Windows is an operating system designed by Microsoft to be used on standard x86 Intel and AMD processors. It provides an interface, known as a graphical user interface(GUI) which eliminates the need to memorize commands for the command line by using a mouse to navigate through menus, dialog boxes, buttons, tabs, and icons. The operating system was named windows since the programs are displayed in the shape of a square. This Windows operating system has been designed for both a novice user just using at home as well as for professionals who are into development.

* It is designed to run on any standard x86 Intel and AMD hence most of the hardware vendors make drivers for windows like Dell, HP, etc.
* Windows has a very large user base so there is a much larger selection of available software programs, utilities.
* Windows is backward compatible meaning old programs can run on newer versions.

**LINUX Operating System:**

The Linux OS is an open source operating system project that is a freely distributed, cross-platform operating system developed based on UNIX. This operating system is developed by Linus Torvalds. The name Linux comes from the Linux kernel. It is basically the system software on a computer that allows apps and users to perform some specific task on the computer.

* Linux is free, can be downloaded from the Internet or redistribute it under GNU licenses and has the best community support.
* Linux OS is easily portable which means it can be installed on various types of devices like mobile, tablet computers.
* It is a multi-user, multitasking operating system.
* BASH is the Linux interpreter program which can be used to execute commands.
* Linux provides multiple levels of file structures i.e. hierarchical structure in which all the files required by the system and those that are created by the user are arranged.
* Linux provides user security using authentication features and also threat detection and solution is very fast because Linux is mainly community driven.

**7. getty** [ [ [**-r**](https://www.ibm.com/docs/en/aix/7.1?topic=g-getty-command#getty__row-d3e140985) | [**-u**](https://www.ibm.com/docs/en/aix/7.1?topic=g-getty-command#getty__row-d3e140997) | [**-U**](https://www.ibm.com/docs/en/aix/7.1?topic=g-getty-command#getty__row-d3e141006) ] [ [**-d**](https://www.ibm.com/docs/en/aix/7.1?topic=g-getty-command#getty__row-d3e140934) ] [ [**-H**](https://www.ibm.com/docs/en/aix/7.1?topic=g-getty-command#getty__row-d3e140940) *HeraldString* ] [ [**-M**](https://www.ibm.com/docs/en/aix/7.1?topic=g-getty-command#getty__row-d3e140955) *motdFile* ] [ [**-N**](https://www.ibm.com/docs/en/aix/7.1?topic=g-getty-command#getty__row-d3e140967) ] ] *PortName*

The getty command sets and manages terminal lines and ports. The getty command is run by the init command. The getty command is linked to the Terminal State Manager program. The Terminal State Manager program provides combined terminal control and login functions.

You can configure the getty command to create your home directory at your login if you do not have a home directory already. The getty command calls the mkuser.sys command to create the home directory and customize the account. To enable this capability, set the mkhomeatlogin attribute of the usw stanza in the /etc/security/login.cfg file to true. The getty command is not entered on the command line.

The file /usr/sbin/getty contains the getty command.

**8. Uname commands?**

The command ‘*uname*‘ displays the information about the system.

Syntax: uname [OPTION]

1. -a option: It prints all the system information in the following order: *Kernel name*, *network node hostname*, *kernel release date*, *kernel version*, *machine hardware name*, *hardware platform*, *operating system.*

2. -s option: It prints the kernel name.

3. -n option: It prints the hostname of the network node(current computer).

4. -r option: It prints the kernel release date.

5. -v option: It prints the version of the current kernel.

6. -m option: It prints the machine hardware name.

7. -p option: It prints the type of the processor.

8. -i option: It prints the platform of the hardware.

9. -o option: It prints the name of the operating system.

**9. What is systemd.unit(5)?**

A unit configuration file whose name ends in ".service" encodes information about a process controlled and supervised by systemd. This man page lists the configuration options specific to this unit type. See [systemd.unit(5)](https://man7.org/linux/man-pages/man5/systemd.unit.5.html) for the common options of all unit configuration files. The common configuration items are configured in the generic [Unit] and [Install] sections. The service specific configuration options are configured in the [Service] section.

Additional options are listed in [systemd.exec(5)](https://man7.org/linux/man-pages/man5/systemd.exec.5.html), which define the execution environment the commands are executed in, and in [systemd.kill(5)](https://man7.org/linux/man-pages/man5/systemd.kill.5.html), which define the way the processes of the service are terminated, and in [systemd.resource-control(5)](https://man7.org/linux/man-pages/man5/systemd.resource-control.5.html), which configure resource control settings for the processes of the service. If a service is requested under a certain name but no unit configuration file is found, systemd looks for a SysV init script by the same name (with the .service suffix removed) and dynamically creates a service unit from that script. This is useful for compatibility with SysV. Note that this compatibility is quite comprehensive but not 100%. For details about the incompatibilities, see the **Incompatibilities with SysV**[1] document. The number basically corresponds to the section of the manual page.

Section 1 for General Commands , Section 2 for System Calls, Section 3 for Library functions, covering in particular the C standard library, Section 4 for Special files (usually devices, those found in /dev) and drivers, Section 5 for File formats and conventions, Section 6 for Games and screensavers, Section 7 for Miscellanea, Section 8 for System administration commands and daemons.

**10. Difference between Systemd and initd?**

The init process forces services to be launched in a particular sequence. It makes each process dependent on another process which can lead to delay.

Whereas systemd is used to run the system services in parallel. It is helpful in removing unnecessary delays and boosting up the initialization process.

**11. whichis and whereis command?**

Whereis is a command line utility that allows you to find the location of the binary, source and manual pages files.

Ex: $ whereis bash

Whichis command in linux used to locate the executable file.

Ex: $ which cpp python java