**Assignment 1**

Operating Systems

Submitted by:

Logo, company name

Description automatically generatedGodala Sukumar Reddy(2020BTechCSE032)

Institute of Engineering and Technology (IET)

JK Lakshmipat University, Jaipur

August 2022

**Introduction to Linux:**

An open-source operating system modelled on UNIX where multi-users can operate system at same time. Linux is an O/S core originally written by Linus Torvalds.

Firstly, Linux came from UNIX which was developed by Bell Labs Developers in 1960.

This language was created specifically for the creation of the UNIX system. It was considerably easier to design an operating system that could operate on a wide range of hardware using this new method.

On May 14, 1991, the first Linux kernel, Version 0.01, was released to the public. It had no networking capabilities, could only run on 80386-compatible Intel CPUs and devices, and had extremely limited device driver support. The virtual memory subsystem was similarly quite primitive, with no support for memory-mapped files.

The history of this operating system's development heavily affected the inside aspects of Linux's design. Even though Linux operates on a variety of platforms, it was designed primarily for PC architecture. Because a large portion of that early work was done by people rather than well-funded research institutes, Linux aimed from the start to extract as much functionality as possible from limited resources.

Mostly it is a compatible OS where it can run on low end and low configuration Hardware, like **RISC** example “**Raspberry pi or Odroid or Pocket Beagle**”.

Linux can run contentedly on a multiprocessor computer with hundreds of megabytes of main memory and many gigabytes of disc space, but it can also run well with only **4 MB of RAM**. Speed and efficiency remain critical to the design goals attained by Linux, and much of the recent and ongoing work on Linux has an influence on the third and fourth fundamental design goals: At the same time, Linux has attained uniformity and security.

By the early 1990s, home PCs were finally strong enough to run a full-fledged UNIX system. Linus Torvalds, a young guy studying computer science at the University of Helsinki, believed it would be a good idea to create a publicly available academic version of UNIX, so he began coding right away.

In general, Linux resembles any other traditional, non-microkernel UNIX system. It is a multiuser, multitasking operating system that includes a comprehensive collection of UNIX-compatible utilities. Linux's file system adheres to classic UNIX format, and the standard UNIX networking concept is fully utilized.

He began to ask questions, hoping to find answers and solutions that would allow him to install UNIX on his PC.

**Some of Basic Features of Linux:**

Multi-access system (with time sharing) and multi-tasking

• Multiprocessor system, simple mechanisms to create hierarchy of processes, kernel pre-emption.

•Portable

•Open source

•Multiuser

•Multiprogramming

•Shell

•Security

• Available for many architectures

• Simple standard user interface that can be easily replaced (shell ◊ command interpreter)

• File systems (!) with a multi-level tree structure

• Files are seen as strings of bytes (easy to write filters)

• Loading programs on demand (fork with copy on write)

• Virtual memory with paging

• Dynamic hard disk cache

• Shared libraries, loaded into memory dynamically (one code used simultaneously by many processes)

• Compliance with the POSIX 1003.1 standard

• Different formats of executable files

The tendency to reduce the **kernel in Linux** is manifested through modules, which are individually constructed, loaded into memory on demand, and destroyed when no longer required.

• Examples include a device driver, a file system, and a file format that may be executed.

• Benefits include memory savings (the module occupies memory only when it is needed), the fault in the module does not suspend the system but just removes the module from memory, the ability to utilize competing drivers without having to restart the system, and so on.

• Much of what is wonderful about microkernels is borrowed by Linux: modular architecture, the ability to preempt itself, support for kernel threads, and the capacity to dynamically load distinct programmed (kernel modules).

**Linux System Components and Structure**

**Kernel**:

The kernel is the foundation of Linux. It is in charge of all important operations in this operating system. It is made up of numerous modules that interface with the underlying hardware. The kernel offers the necessary abstraction to system or application programmers to hide low level hardware specifics.

**System libraries** are special functions or programmers that allow application applications or system utilities to access Kernel functionality. These libraries implement the majority of the operating system's functionality and do not require kernel module code access permissions.

**System Utility**

programmers are in charge of performing specialized, individual-level activities.

**Hardware layer**

The Linux operating system has a hardware layer that includes many peripheral devices such as a CPU, HDD, and RAM.

**Shell**

It serves as a link between the kernel and the user. It can afford kernel services. It can accept commands from the user and execute kernel functions. The shell is available in a variety of operating systems. These operating systems are divided into two categories: graphical shells and command-line shells.

**Advantages:**

**Security:**

Linux is designed to be secure. Unlike Windows, Linux is less vulnerable to viruses. On rare cases, Linux is infected with a virus. That is why Linux users do not often install antivirus software. Furthermore, due of Linux's limited market penetration, hackers are uninterested in it. Even if it is, they will be unable to quickly compromise the system due to the locks on each tier.

**System Configuration:**

The Linux operating system can even operate on low-spec computers. It does not require as many resources as the Windows operating system. In this OS, all components such as memory, diskspace, and CPU are reduced. As a result, you do not need to update your existing hardware to install this OS. Because of this, Linux is a better choice for all systems.

**Source Code:**

Linux is a free and open-source operating system. This signifies that the code is open to modification. Users may simply modify the code to meet their specific requirements. As a result, you may personalize your operating system. You will, however, need to be familiar with programming.

**Cost:**

The GNU General Public License is included with the Linux operating system (GPL). Unlike Windows, it does not require any licensing. As a result, the operating system is free to use. On Linux, even some software is completely free. As a result, Linux is extremely cost effective. In the software manager, you may locate decent Windows alternative apps.

**Disadvantages:**

**Technical Support:**

Because Linux is an open-source operating system, it lacks technical support. The problem cannot be solved until you solve it yourself. Users must search online for solutions to their problems, potentially in a community forum. This does not ensure that you will have the same difficulty that you do. Aside from this, it is tough to locate someone who is knowledgeable with Linux. Otherwise, if you run into an issue, you will have to check at other Linux distributions.

**Compatibility with Hardware:**

Almost any piece of hardware may be linked to a Linux operating system. The issue, though, is with the drivers. The notion of drivers in Linux differs greatly from that of other operating systems. The drivers in this case are embedded within the kernel. Because the drivers are already preloaded, the newly connected hardware device may be missing the necessary drivers. As a result, the attached hardware device is incompatible.

**Software Compatibility**

Popular software designed for Windows and Mac is not available for Linux. Because of the low commercial value of Linux, many developers are unwilling to create applications for it. MS Office, iTunes, and Photoshop are examples of such applications. Even though there are substitutes, they cannot compete with the original level.

**Application Issues:**

Games, like software, do not support Linux natively. Because Linux is not a commonly utilized platform, gaming developers are not particularly interested in it. As a result, you should not anticipate your favorite game to operate on Linux. However, the number of games created for Linux has increased in recent years. Even if you utilize third-party software such as Steam to get any gaming title, the game will not operate as smoothly as it would on a native operating system.

**User Experience/Interface:**

Linux might be difficult to grasp for those with less computer knowledge. Because of the terminals utilized, most users find it difficult to transition to Linux. Terminals are command line interfaces where commands must be entered in order to execute tasks. Furthermore, the difficulty level changes depending on the distribution utilized. Learning to use Linux gets much more challenging with distributions such as Arch.

**Unified Installer:**

In contrast to Windows, which only has exe files, each Linux distribution has its own Package manager, making it difficult for developers to build and deliver Linux programmes. Fedora supports RPM, Debian supports DPKG, and Arch supports Pacman. By default, there is no unified package manager for all distributions.