

Bramha-The Supreme Os

Dr.Richa Sharma¹, Mr. Gaurav Kalge², Mr. Suraj Patil³, Ms. Mitali Ovhal⁴, Mr.Aman Pardeshi⁵

¹Guide, Dept of Information Technology

^{2, 3, 4, 5} Dept of Information Technology

^{1, 2, 3, 4, 5} Genba Sopanrao Moze College of Engineering, Balewadi, Pune, India

Abstract- In this paper, it states the step wise development of a Linux based operating system with the Linux from scratch method. The classifications of various existing systems are explained. OS Permits you to make exceptionally minimal Linux frameworks. The necessity to create a new Linux-based system is examined first, along with various operating systems. Benefits and drawbacks of various operating systems to the audience. Also, we examine a step-by-step process from "scratch" to a finished Linux-based operating system. Government agencies and experts have recently become interested in the open-source software movement, and the Linux open-source and free operating system has drawn the most interest. So much so that even the largest businesses, like Google and Oracle, have stepped forward and are creating open-source software, particularly Linux, in response to this movement. You can build Linux systems that are incredibly small with LFS.

Keywords- Linux, LFS, Framework, BLFS

I. INTRODUCTION

The open-source and free Linux operating system has recently dominated the open-source software movement, drawing the attention of several experts and governmental bodies. So much so that even the largest corporations, like Google and Oracle, have stepped forward and are creating open-source software, particularly Linux, in support of this trend. The popularity of open-source software, particularly Linux, can be attributed to a variety of factors. In PCs, Linux is being added to the BIOS framework. Since Linux is now available on portable Internet devices, this framework is essential for providing Internet [2]. The growth of distributed computing is a trait that fits with Linux, which currently powers a sizable fraction of the web servers on the internet. Linux makes it possible for the cloud to send clients' data and web apps. The extraordinary success of Linux on supercomputers, both in the high-accessibility (HA) and superior registering ranges is associated with Linux development in distributed computing. Here, academic research in physical science and bioengineering, as well as businesses in the financial and energy sectors need reliable and adaptable processing energy to accomplish their goals. Since Linux was created from scratch, it lacks any

behavior code. On servers and other large systems like mainframe computers and the fastest supercomputer, Linux is the leading operating system. It also functions on a coordinated system that is used by devices with OS, such as smart phones, tablets running Android, Linux-based operating systems, network routers, televisions, video games, and smart watches. The source code is available for use, modification, and distribution under the terms of the GNU general public license. DEBIAN, UBUNTU, LINUX MINT, FEDORA, ARCH LINUX, RED HAT, and SUSE Linux organization server are a few of the well-known Linux distributions.

1.1 Advantages of LFS

Following can be considered as the advantages of LFS:

(a) Higher Security:

Higher security is provided as resource codes are directly used to develop applications and for system configurations, hence the risk of destructive codes and security holes is minimized.

(b) Network Issues

- Security comes in three flavors: access, prevention, and detection. Users often log in to gain access. The spam module in blfs will change this. Applications like GnuPG help with breach prevention. The package that holds the "signatures" of crucial data is used to aid in detection.
- Ssh: With this package, we won't have to manually type every URL. Instead, we might link the existing system to another system to simplify our work.
- The pluggable authentication modules (PAM) in this package are utilized by the administrator.
- NFS is a bundle of libraries created to facilitate cross-platform development of server and client applications with security features.
- DHCP: The client, dhclient, is utilized to connect to a network that.

(c) A window or display manager All X Window Applications use the many libraries contained in the

XORG package. Display manager: These are graphical applications used to launch the X server for graphical displays. Desktop Environment: The designer can choose from a variety of environments and there are numerous libraries available. We chose the GNOME library since it has a lot of support. The Gnome process aids with the creation of software framework.

II. LITERATURE SURVEY

By today's standard of OS (operating system) and the requirements of applications, security, and laws it will be very difficult to build a good functioning securely, and evolving OS. This may be why we have only three major OS in the world today: Linux, Windows, and macOS. OS is a platform that involves hardware, software, and people so this requires a massive continuous collaboration to build, maintain and evolve the OS. An OS needs the collaboration of good designers and programmers as many different functionalities and services must be designed, coded, and integrated to work together such as boot loader, system calls, file system, ability to execute a file, a shell, and multiprocessing, etc. (Black, 2009). The challenges and difficulties will also vary depending on the type of OS being built. For example, for Internet ware OS “technical challenges still need to be resolved, including the architecture of Internet ware OSs, performance and applicability issues, as well as security and privacy considerations” (Mei & Guo, 2018). There are several aspects that will be difficult to build and evolve. For example, we know that OS should allow multiple computations to take place concurrently, divide the hardware's time between the computations, keep track of where each one leaves off so that it can later correctly resume, control the interactions between the concurrent computations, forbid computations from modifying data structures, provide isolated areas of memory for private use by the different computations and provide support for controlled interaction of computations across time and space (Halpern, 2019). These give rise to the issues of concurrency, deadlock, synchronization, virtual memory, and security. Any programmer of a complicated system requiring these functionalities is likely to introduce race bugs and security loopholes because concurrency and synchronization for example are hard to reason about (Halpern, 2019). Designing fast GUI for example can be difficult that does not crash or hang often or slows down. Also choosing a good shell interface and implementing its functionalities will require careful analysis of existing shells and perhaps backward compatibility. These issues also make designing and implementing OS very difficult. An important but difficult aspect is how theory transitions to actual coding, if there is any use of assembly language or rather which programming languages may be used to implement some of these OS

functionalities, Also there may be context or situation when concurrency fails and considering the implications and remedy for such a situation become difficult*.

❖ Linux

The Linux operating system is used on a wide range of devices, including smartphones, automobiles, supercomputers, home appliances, personal PCs, and business servers. Since its introduction in the middle of the 1990s, Linux has grown to have a global user base (Linux, 2018). Early in the 1990s, Linus Torvalds developed the open-source Linux Operating System. Linux, which is based on UNIX, enables users to alter the existing code and create various distributions or versions of it, which they can then use and even sell. The majority of websites on the internet are served by systems powered by one of Linux's distributions since it has emerged as the leading choice for putting up servers. The OS is also widely used in computers, smartphones, gaming consoles, eBook readers, and many more devices (Bhattia, 2022).

Features

The Linux OS consists of several components: the bootloader, the kernel, the init system, daemons, graphical server, desktop environment, and various applications (Linux, 2018). Linux shows how much total system memory is currently used, cached, buffered, or free, shows how much time the CPU spends performing system and user duties, stealing jobs from virtual operating systems, waiting for I/O to finish, and processing pleasant tasks, shows the amount of work that a computer system performs like Windows and macOS, tracks growth or reduction in the amount of available disk space used like windows and macOS and other features such as disk utilization, queuing, latency and throughput (Operating System Performance Metrics, n.d.).

Strength

Linux is an OS where you would battle less with obstacles like viruses, malware, slowdowns, crashes, costly repairs, and licensing fees (Linux, 2018). Linux is also open source which means it is flexible to use, changes can be made easily, and distributions can be shared among peers and updated easily. Every type of user can utilize one of the many versions of Linux. You can select a "variant" of Linux to suit your needs, whether you're a novice user or a die-hard. Distributions (or "distros" in the short form) are the names for these variations. Almost all Linux distributions are available for free download, burning to disc (or USB thumb drive), and installation (on as many machines as you like) (Linux, 2018).

❖ Windows

Microsoft began developing a tool named "Interface Manager" in 1981, which is when Windows first appeared on the scene. Under the moniker "Windows," it was first introduced in November 1983 (after the Apple Lisa but before the Macintosh), but Windows 1.0 wasn't made public until November 1985. (A History of Windows - Microsoft Windows, 2016).

Features

Windows have some similar features to Linux but some features that are special to Windows are Windows shows the available physical memory, the cached operating system code, and the allocated pool-paged-resident and pool-nonpaged memory, shows how much time the CPU spends on user processes, and system processes, has a start menu, has tablet mode in Windows 10, has a game center which can be connected with XBOX, has multiple desktops and task views, has Cortana assistant manager, unified settings and action center (Low, 2014; Operating System Performance Metrics, n.d.).

Strength

Windows looks good for GUI, is easy to learn, has lots of productivity apps, has good gaming functionalities and support, has an app store, has some decent diagnostic and administrative tools, and setting up is easier than Linux, and has a large support base among other strengths.

❖ macOS

Steve Jobs launched NeXT after leaving Apple in 1985, and it was there that the roots of what would eventually become macOS could be found. Before being released in 1989, the Unix-like NeXTSTEP operating system was created there. With extra kernel layers and low-level user space code inherited from portions of BSD, NeXTSTEP's kernel is based on the Mach kernel, which was first created at Carnegie Mellon University. (01_SysOver, 1995).

Features

macOS uses proprietary hardware. With OS X, you get a completely integrated system in which hardware and software interact flawlessly to provide you with the finest Mac experience possible. The fact that your Mac's apps can communicate with those on your iOS devices is due to the practical iCloud. To give you an example, anything you do in

apps on your Mac will also be reflected on your other iOS devices. (Poole, 2021).

Strength

macOS has fewer virus attacks than Windows, has good customer support, consistent GUI for all its products, provides great speed and availability and due to its superior hardware provides long life. It also supports the Windows file system and can run Windows. (Rehman, 2019b).

III. OBJECTIVE OF STUDY

Kernel Architecture:

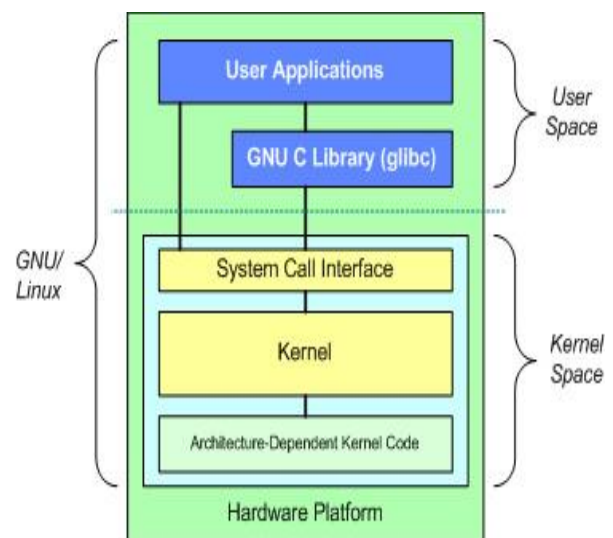


Fig-1Architecture

3.1 User Manner: The user manner is the space allocated in the memory where user processes run. This is located above the kernel space. The system prevents one process from interfering with another process. Only kernel processes can access a user process and not vice versa.

3.2 Kernel Manner: The kernel manner is the space in the memory where all the kernel servers are provided via kernel process. The user can access kernel only through a system

3.3 Kernel Structure

3.3.1 File System: Is generally is a layer which is under the operating system which handles the allocation of data, without which the system won't know where a file starts and ends.

3.3.2 Process Management: A time-sharing system, Unix OS. Every process has a time slice during which it is consistent. The process is created, completed, and deleted by the kernel.

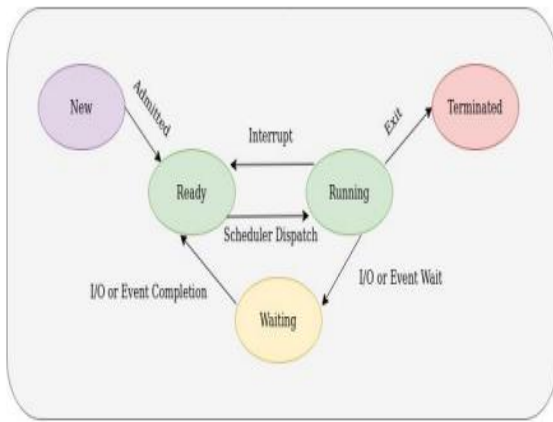


Fig-2 Process

Types of Processes :

- (1) Running: The process is either ready to run or running
- (2) Waiting: The process is waiting for an event to occur
- (3) Stopped: The process has been stopped by a signal.
- (4) Zombie: This is an unfit process which for some reason still has a task affected data structure.

3.4 Device Driver

A piece of code known as a device driver, which manages the hardware of the device, is associated with each physical driver or essential driver. The primary duties of a driver are: hardware setup for data formatting. Integrating and removing connected devices from services. Returning to the kernel after receiving data from hardware. Sending information to the device from the kernel. Manipulating and detecting device faults.

3.5 Memory Management

Physical memory is divided into portions called Pages.

Types of memory management:

- Physical memory
- Virtual memory
- Swap memory

IV. COMPARISON OF OPERATING SYSTEM

Qualities	Windows	MacOS	Linux
Easy to use	Easy to use	Easy to use	Recent distributions are easier to use than previous variants.
Power	Has power but still working on it.	Better than windows	Highest power
Privacy	Very Intrusive	A bit intrusive	Non-intrusive
Maintenance	Requires a lot of maintenance	Extremely low maintenance	Claims to have low maintenance
Security	Worst security even on the newest versions	Best Security	Claims to have high security
Package manager	Does not need a package manager	Does not need a package manager	Use it to get new software or update existing software
Ease of recovery	Installation DVD	Built in backup software	Recover the system the hard way

V. LINUX FROM SCRATCH:

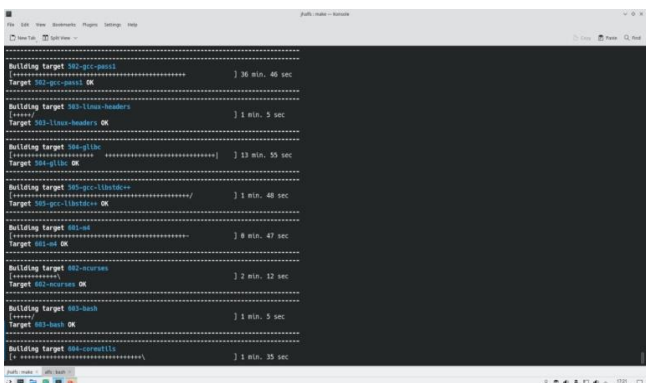
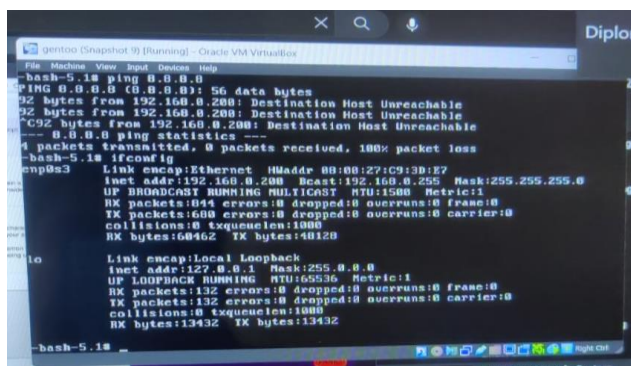
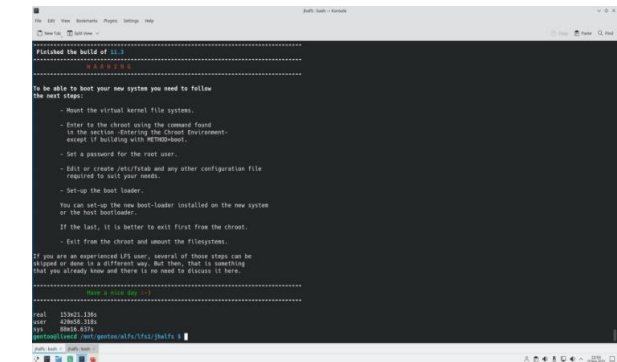
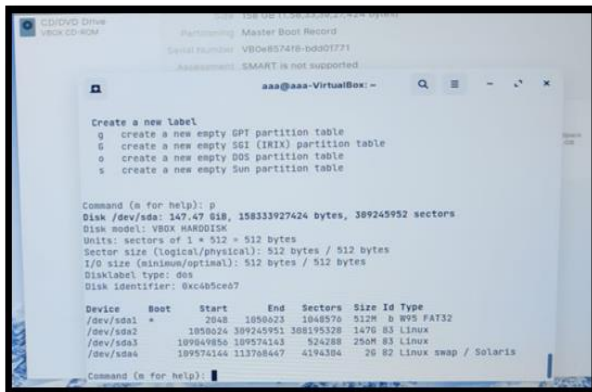
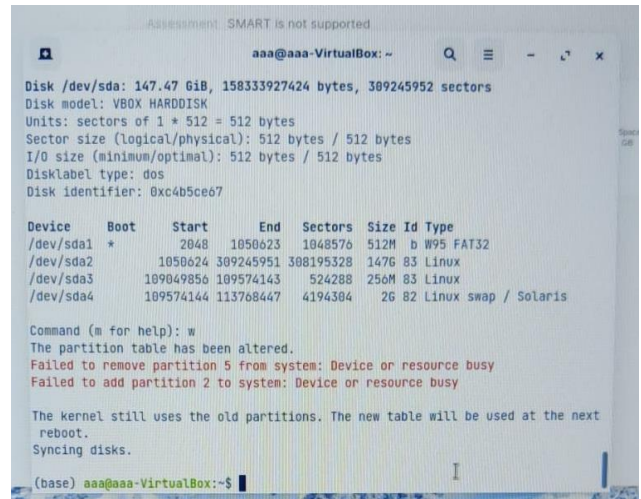
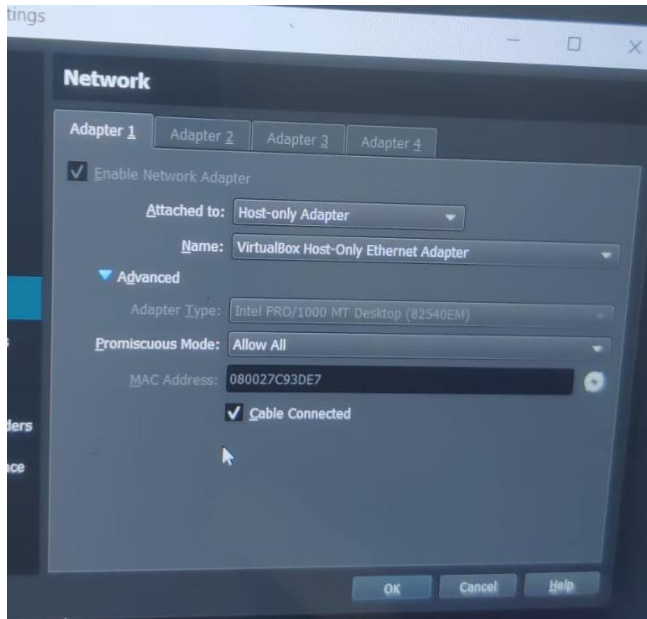
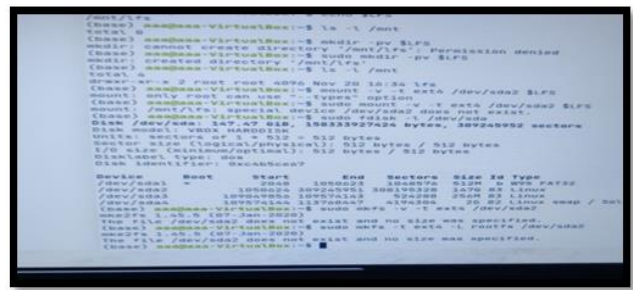
LFS is a step-by-step process that uses a host system and the source code to create an operating system based on Linux. Gerard Beekman developed this process.



Fig-3Implementation process

- (a) **Preparing the host system:** Executing the script "version-check.sh" to determine the host system requirements. putting in the programs that aren't already there. Start the construction process.
- (b) **Creating a new partition:**We need to establish about six partitions, some of which are optional. Here is a list of the partitions:/boot, /home, /usr, /opt, /tmp, and /usr/src are all included.On the partition, we are now creating a file system. Although ext2 and ext3 are also choices, it is of the type ext4.We perform all of the actions using the /mnt/lfs directory.As an illustration, the directory /mnt/lfs/tools will house all the tools, etc.It's time to mount the partition.
- (c) **Constructing a temporary system:** We download packages and patches available on: <http://www.linuxfromscratch.org/lfs/view/stable/wget-list>. These packages include every essential tool needed to create linux operating system. This system is being developed in two stages: first, a new host independent toolchain is being created, and then, using this toolchain, other crucial tools are being created.

(d) **Installing basic system software:**The /mnt/lfs/tools directory will now contain all of the utilities that we have downloaded.To prevent circular dependencies, we shall preserve dependencies while installation. Since there are fewer items to install, we may employ the "it's all in my head" strategy and begin the installation process.



VI. FUTURE SCOPE

Enhancing the kernel code for better cpu performance and disk scheduling. Integrating module for improving overall performance of the operating system. Better handling of I/O interrupt. For now, essential target architectures of OS are AMD/INTEL x86_64 CPUs. In future, it will support all essential architectures and will be able to run on any machine. This works will improve the Linux Operating System. OS Permits you to make exceptionally minimal Linux frameworks. Preference of a custom manufactured Linux framework is security. By accumulating the whole framework from source code, we are engaged to review everything and apply all the security patches sought. In this paper we will attempt to make Open Operating by modifying the source code for better security. We will attempt enhance the kernel module to enhance the CPU execution too driver integration. New binary is not always correctly built so we will take into consideration to improve the performance. Keyword: Framework, Kernel, Partition, SuSE, API headers, LFS, Virtual file.

VII. CONCLUSION

Following the above procedure for LFS, we get the output as a terminal. And the BLFS system, continued on LFS, which we have built can be considered as a flavor of Linux operating system. Importing the new kernel file system decreases many perform issue and we find that OS execution has enhanced significantly since its creation. Workloads that used to be viewed as exceptionally difficult, in the same way as line-rate 10 Gbps systems administration, are currently conceivable utilizing just a solitary center utilizing 2013-time equipment and programming. Actually, utilizing the speediest accessible manifestations of OS still adds some overhead to each WE/O operation; this overhead ranges from noteworthy when performing little WE/O to immaterial when it is amortized over expansive, WE/Os. By and large, we discovered the execution of Ext2 to be superior to that of Ext3 with data=writeback to be the same as with data=ordered. In specific cases, we discovered Ext3 with data=ordered to be speedier; in different cases data=writeback is quicker.

REFERENCES

- [1] Maurice J. Bach, "The Design of the UNIX Operating System. Prentice Hall", EngleWoodClis, New Jersey, 1986.
- [2] Daniel P. Bovet and MarcoCesati, "Understanding the Linux Kernel", O'Reilly and Associates (2001), ISBN 0-596-00002-2.

- [3] Remy Card, "The Second Extended File System - Current State Future Development", 2nd International Linux and Internet Conference, Berlin, 1995.
- [4] Marshall Kirk McKusick, "The Virtual Filesystem Interface in 4.4BSD", The USENIX Association, Computing Systems, 8(1):3{25, Winter 1995.
- [5] Mauerer W. Professional Linux kernel architecture: Wiley Publishing, Inc, 2008.
- [6] A. Sweeney, D. Doucette, W. Hu, C. Anderson, M. Nishimoto, and G. Peck. Scalability in the XFS File System, ATC, pages 1-14, San Diego, CA, USA, 1996.
- [7] SEPPANEN, E., O'KEEFE, M., AND LILJA, D. Highperformance solid state storage under Linux. In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on (2010).
- [8] YU, Y. J., SHIN, D. I., SHIN, W., SONG, N. Y., EOM, H., AND YEOM, H. Y. Exploiting peak device throughput from random access workload. In Proceedings of the 4th USENIX conference on Hot Topics in Storage and File Systems (Berkeley, CA, USA, 2012), HotStorage'12, USENIX Association.
- [9] <http://www.linuxfromscratch.org/lfs/view/stable/>
- [10] <http://www.linuxfromscratch.org/blfs/view/stable/>
- [11] <https://ijcsi.org/papers/IJCSI-9-2-3-170-174.pdf>
- [12] www.ijtrd.com/papers/IJTRD1374.pdf