



INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 6.078

(Volume 6, Issue 2)

Available online at: www.ijariit.com

A survey on operating systems, LFS and BLFS

Kohinoor Chatterjee

kohinoor.chatterjee17@vit.edu

Vishwakarma Institute of Technology,
Pune, Maharashtra

Prafull Lad

prafull.lad17@vit.edu

Vishwakarma Institute of Technology,
Pune, Maharashtra

Kartik Kapgate

kartik.kapgate17@vit.edu

Vishwakarma Institute of Technology,
Pune, Maharashtra

Isha Bansod

isha.bansod17@vit.edu

Vishwakarma Institute of Technology,
Pune, Maharashtra

Pravin Khandare

pravin.khandare1@vit.edu

Vishwakarma Institute of Technology,
Pune, Maharashtra

ABSTRACT

Recently, government organizations, specialists have been attracted to the movement of open-source software and on top of them the Linux open-source and free operating system has caught the most attention. So much so that in this movement even the biggest companies like Google, Oracle have stepped forward and are developing an open-source software, particularly Linux. This paper states the step wise development of a Linux based operating system with the Linux from Scratch (LFS) method. First, we take a look at different operating systems, and the need to develop another Linux-based system. The advantages and disadvantages of these operating systems are presented. And further we analyze a step by step method from "scratch" to a complete Linux-based operating system.

Keywords— Linux, LFS, BLFS

1. INTRODUCTION

Recently, Linux open-source and free operating system has been on top of the movement of open-source software, attracting the attention of many specialists, and government organizations. So much so that in this movement even the biggest companies like Google, Oracle have stepped forward and are developing an open-source software, particularly Linux. There are many reasons for the popularity of an open-source software, particularly Linux. A few of them have been listed below:

- Stability
- Highly flexible
- Rapid development
- Strong support from the network
- Higher security
- Being free

Linux attracts attention as one of the safest available operating systems, hence in the present world of needing to secure data, governments and organizations tend to use more secured products. In Linux, the resource code is available for free, and most importantly personalization changes can be exerted in all the layers of the operating system. This allows a programmer to develop and design their own Linux-based operating system.

© 2020, www.IJARIIT.com All Rights Reserved

Linux does not contain any behavior code as it was written from scratch. Linux is a leading operating system on server and other big systems such as mainframe computers and fastest super computer. It also runs on the coordinated system which devices having OS is typically built into a microcode, it includes smart phones, tablets running android, Linux derivatives, network routers, televisions, video games and smart watches. The source code may be used, modified and distributed with respective license as GNU general public license. Some of the popular Linux distributions are DEBIAN, UBUNTU, LINUX MINT, FEDORA, ARCH LINUX, RED HAT and SUSE Linux organization server.

Having such strong capabilities, any countries and organizations have tried to design and implement their own operating systems to meet with their specific needs, without the interference of an external group, and at a minimal cost and high efficiency.

2. KERNEL ARCHITECTURE

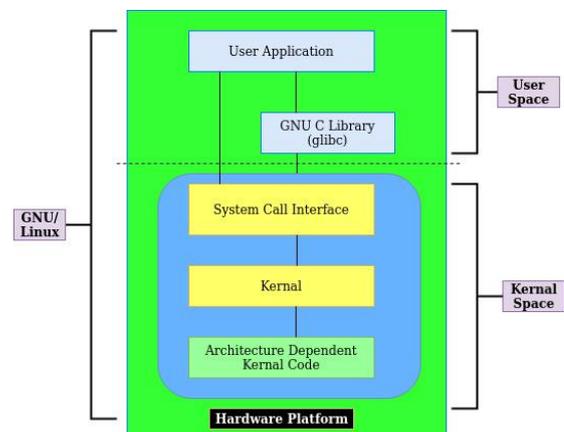


Fig. 1: Architecture

2.1 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.

2.2 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 call. When a system call is executed, a user process becomes a kernel process.

2.3 Kernel Structure

2.3.1 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.

2.3.2 Process Management: The Unix OS is a time-sharing system. Every process is regular for a period of time, known as time slice. Kernel creates, succeed and deletes the process.

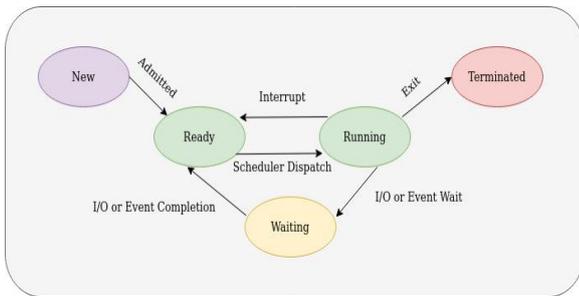


Fig. 1: Process

2.4 Types of Processes (Figure 1)

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

2.5 Device Driver

Related with each physical driver or essential driver is a piece of code called Device Driver, which carry off the device hardware. The main functions of driver: Setting up hardware on data formatting. Bringing the connected devices into and out services. Receiving data from hardware and passing it back to the kernel. Sending data from the kernel to the device. Detecting and manipulating device errors.

2.6 Memory Management

Physical memory is divided into portions called Pages. Types of memory management:

- Physical memory
- Virtual memory
- Swap memory

3. COMPARISON OF OPERATING SYSTEMS

Table 1: Comparison of operating systems

Qualities	Windows	MacOS	Linux
Ease of 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
Graphic Cards	Windows has the best graphic drivers so far	MacOS does well with a variety of graphics cards	Still a huge problem for Linux
Customization	Does not need any customization	Does not need any customization	Highly customizable
Reliability	A bit reliable, reboots are necessary as a normal part of using Windows	Very reliable, reboots might not be needed	Is somewhat better than Windows if you sort out all your customization issues
Error messages	Cryptic and require computer background to understand them	Typically clear	Might or might not have one, not very helpful
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

4. LINUX FROM SCRATCH

LFS, is a step by step method for developing a Linux based operating system, using a host system and the source code. This method was invented by Gerard Beekmans.

4.1 Implementation

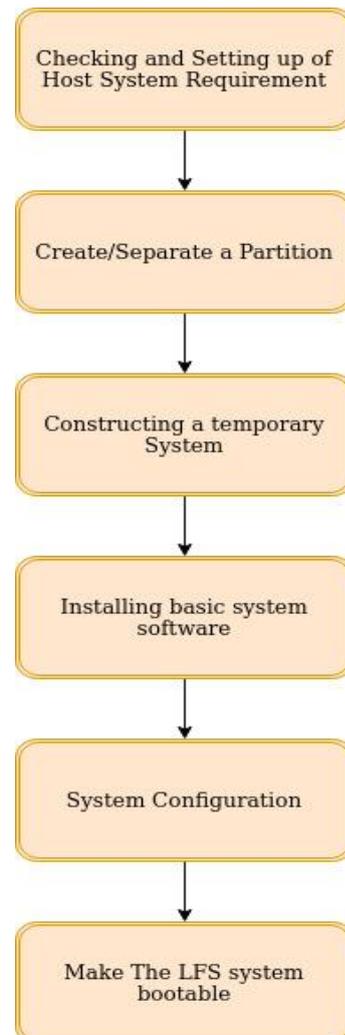


Fig. 2: Implementation process

(a) **Preparing the host system:** Checking the host system requirements by running a script "version-check.sh". Installing the programs that are not installed. Start the building process.

(b) Creating a new partition: There are approximately six partitions we need to create, a few are optional. The partitions are listed below:

- /boot
- /home
- /usr
- /opt
- /tmp
- /usr/src

Now we create a file system on the partition. It is of the type ext4, however ext2 and ext3 are also available options. Throughout we use /mnt/lfs directory to carry out the tasks. Example: /mnt/lfs/tools directory will contain all the tools, etc. Now we 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 contain all the basic tools which are required to make Linux operating system. There are two steps in developing this system, first is to build a new host independent toolchain, and second to use this toolchain and build other essential tools.

(d) Installing basic system software: All the tools that we have downloaded in the /mnt/lfs/tools directory will now be installed. While installing we will be maintaining dependencies to avoid circular dependencies. The number of packages to be installed are less and hence we can use “it’s all in my head” technique and start the installation process.

(e) System configuration: In this, the boot time scripts must be set which will determine the kind of configuration used in the system while loading in the main memory. A special booting facility named “SysVinit”, which is based on run-levels, is used. The different run levels are:

- 0: halt the computer
 - 1: single-user mode
 - 2: multi-user mode without networking
 - 3: multi-user mode with networking
 - 4: reserved for customization, otherwise does the same as 3
 - 5: same as 4, it is usually used for GUI login (like X’s xdm or KDE’s kdm)
 - 6: reboot the computer
- The default run level is 3.

The next thing to be configured in this is the general network configuration. Here we modify three files, hardcoded according to our network preferences. This configuration will help us access the internet via the Linux operating system that we built. Further we have to make a few changes on the kernel level, such as adding custom rules. There are a few more configurations such as the system clock, syslogd script, etc.

(f) Making the LFS system Bootable: In this section we will create fstab file, and building a kernel for the new LFS system. Installing the GRUB boot loader is also an option. While installing the kernel we have to keep in mind the changes that we have to make for the next step that is BLFS. To install the boot loader we need to make changes in /boot/grub which is optional as we can change the boot sequence manually.

4.2 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) Higher stability of the system: Only according to the need definitions and type of the system usage, system partitions are added. Thus, extra software which reduces the stability of the system is deleted.

(c) More flexibility: In this method it is possible to add and delete different partitions of the operating system. Hence, customization and personalization are possible in all layers, including the kernel layer.

5. BEYOND LINUX FROM SCRATCH

5.1 Implementation

(a) Configuration after LFS

- About Firmware: /lib/firmware directory is where the kernel drivers look for firmware images. In order to get multiple firmware, we simply download them from git.
- About devices: Devices such as multiple sound cards, USB device issues, new dev device attribute, server, and DVD driver have to be manually configured.
- Configuring for adding user: As we cannot do everything in the kernel space, we need to add a user which can operate in the user space. Hence, we start adding user using the following command:
- useradd -m <username>
- Bash shell start up file: /bin/bash uses a collection of startup files to create an environment. Each file has a specific use. Starting from /bin/login to ~/.bash_logout, there are a magnitude of files which handle the startup, such as /etc/profile, /etc/profile.d/dircolors.sh, etc.

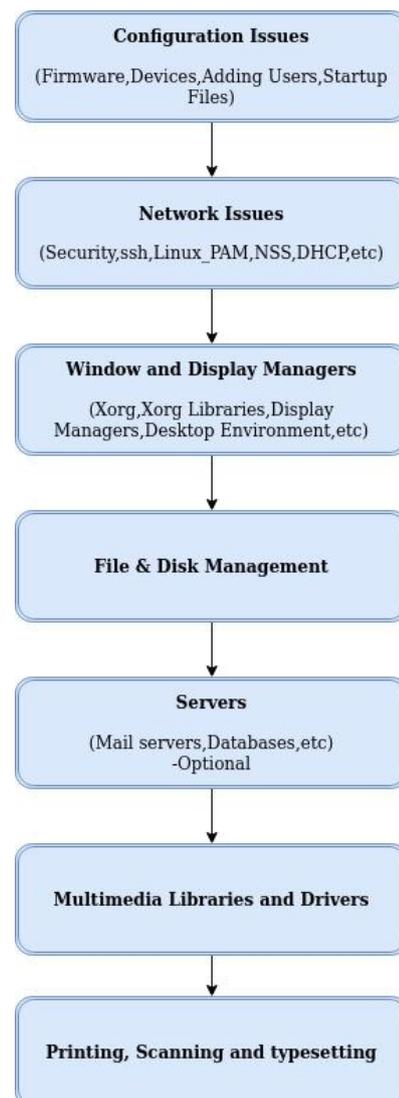


Fig. 3: Implementation Process

(b) Network Issues

- Security: There are three types of security- access, prevention, and detection. Access for users is usually handled by login. This will be changed in blfs in spam module. Prevention of breaches are assisted by applications like- GnuPG. Detection is done with the help of the package that store “signatures” of critical files.
- Ssh: Using this package we will not have to type all the URL manually. Instead we can connect the current system to another system to make our work easier.
- PAM: This package contains pluggable authentication modules used by the administrator.
- NFS: This package is a set of libraries designed to support cross
- platform development of security enabled lines and server applications.
- DHCP: dhclient (the client) is used for connecting to a network which uses DHCP to assign network addresses. dhcpd (the server) is used for assigning network addresses on private networks.

(c) Window and display manager

XORG: This package contains multiple libraries that are used with all X Window Applications.

Display manager: These are graphical programs for starting the graphical display- X server.

Desktop Environment: There are multiple libraries available and is completely dependent on the designer on which environment they select. We have gone for gnome library as there is lot of support available. Gnome is process that helps develop software framework.

(d) File systems and Disk managements

Most of them are tools which help in making of the file system and disk management. Initramfs is used to mount the root file system at boot time the boot loader loads kernel and initramfs into memory and starts the kernel. The kernel will mount intiramfs as / and start /init. The storage technology used is RAID and LVM manages disk drives. LVM handles physical volumes, volume groups and logical volumes. Whereas RAID is used to combine multiple physical disks into a logical unit.

(e) Server

This section is entirely optional.

- Major server: They are the programs that provide content or services to users or other programs. Servers such as Apache HTTPD, BIND, ProFTPD, etc. are completely possible on our system.
- Mail Server Software: This program basically transports mail from one machine to another. Usually Sendmail is used,

however there are several other choices. We can also have SMTP server or a POP3 server or an IMAP server.

- Databases: There are a variety of databases available from single-user read/write to industrial database servers with transaction support. Building a SQL server on this system is entirely possible.

(f) Multimedia

In this section we handle multimedia libraries, audio utilities, video utilities and CD/DVD-Writing Utilities. There are more than sixty multimedia libraries and drivers that need to be installed so that any multimedia program can function properly. In order to have any audio on the system we need to have packages which can handle audio file manipulation. If you want to have a text to speech renderer, then you need to have sound driver installed on kernel level. Till now all the libraries installed and a few more (like VLC) on the BLFS system act as a prerequisite for the playing videos on the system. Nowadays, CDs and DVDs are no longer used, hence this section is optional, but with six more packages we can start writing to CDs and DVDs.

(g) Printing, Scanning and Typesetting

In order to print we need to have a printer management system and ghostscript applications to render PostScript for display on terminals or paper. There are two ways to install typesetting- there is a direct binary installer or you could build from source.

5.2 Advantages of BLFS

- (a) It is Modular that is we control what goes in. We can control the boot time by changing the startup files directory, etc.
- (b) Graphic User Interface: We can design our own GUI which makes the operating system we just created unique.
- (c) Increased security: Using PAM we can define several rules and add users, which can be completely controlled by administrator.
- (d) All the advantages of LFS are inherited by BLFS.

6. 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.

7. REFERENCES

- [1]<http://www.linuxfromscratch.org/lfs/view/stable/>
- [2]<http://www.linuxfromscratch.org/blfs/view/stable/>
- [3]<https://ijcsi.org/papers/IJCSI-9-2-3-170-174.pdf>
- [4]<http://www.ijtrd.com/papers/IJTRD1374.pdf>