# A survey on implementation of a Linux-based operating system using LFS method

**Alireza Abed Masrurkhah[1], Amir Seyed Danesh[2] and Seyyedeh Narjes Ghiami Taklimi[3]**

**[1,3] Department of Software Engineering, University Of Guilan
Rasht, Guilan, Iran**

**[2] Department of Software Engineering
Faculty of Computer Science and information technology
University of Malaya, 50303, Kuala Lumpur, Malaysia**

## Abstract

In recent years, the movement of open-source software and on top of them Linux open-source and free operating system has attracted the attention of many specialists, governments, organizations, etc. so that the biggest software companies such as Google, Oracle, etc. stepped forward in this movement and are developing open- source software, particularly Linux. The present paper introduces and investigates developing a Linux-base operating system with LFS method. First we have a short look on the history of Linux and the reasons of its importance in the present world. Then, we describe different types of Linux Distributions and their internal structures and explain how to develop a Linux Distribution with different methods. LFS method is then compared with another method called "Remaster" and its advantages and disadvantages are presented. At last we analyze this method, step by step, from the beginning to achieving a Linux-based operating system.

***Keywords:*** *Linux distribution, LFS, Remaster.*

## 1. Introduction

In recent years, the movement of open- source software and on top of the Linux open- source and free operating system has attracted the attention of many specialists, governments, organizations, etc. so that the biggest software companies such as Google, Oracle, etc. stepped forward in this movement and are developing open- source software, particularly Linux. The reasons for popularity of this operating system can be summarized as follows:

- Stability
- Being free
- High flexibility
- Strong support from the network [1]
- Rapid development [2]
- Higher security (in comparison with available systems) [3, 4]

Considering the need to secure data in the present world, governments and organizations, naturally, tend to use more secured products and Linux attracts attention as one of the safest available operating systems (compared to Microsoft Windows and Mac OS, tow operating systems having the most users in the world) [5, 6]. In Linux, it is possible to exert changes and personalize in all layers of the operating system and its resource code is also available for no charge. This allows a programming specialist to design and implement his/her own Linux-based operating system.

Having mentioned capabilities, some countries, security organizations, and even groups and institutes have tried to design and implement their own operating systems to meet their needs with the lowest costs and in the shortest time, without any dependence to an external group and, also, with the highest efficiency. Linux-based operating systems of China and Russia are the examples such systems. These countries designed and implemented their own Linux-based operating systems for the sake of software and security independence of their information systems and also to prevent spending a great deal of money, annually, to buy and support their software. [7]

## 2. Linux-based Operating Systems

In fact, Linux is an application or program which plays the role of an operating system's kernel.[8] The first script of Linux was designed and written by a university student from Finland named "Linus Trowaldez" for Intel 80386 architecture. Another script of Linux was available on the Internet in 1991. Afterwards, many individuals played important role in developing and expanding Linux in different parts of the world. "Trowaldez" co-operated a person named "Richard Stalman" to develop his system. He used GNU project tools and thus an operating system developed through GNU/Linux project. [9]

### 2.1 Structures of Linux-based operating systems

As mentioned in previous section, despite the idea of people and many specialists, Linux is not an operating system per se. Although the most important part of an operating system is its kernel, but the kernel is not the only available part. [10]
An Linux-based operating system can be designed and implemented through combining Linux (as the kernel of an operating system) and other tools such as compilers, connectors, assemblers, and higher level tools such as text editors and graphic intermediates.
Figure 1 shows a simple and general schema of the Linux-based operating system. As can be seen, the kernel is put together with other applications and programs and has created a Linux Distribution. What makes distributions variable and different is the setup of structures and applications in each part. [11]

Table 1: Margin specifications

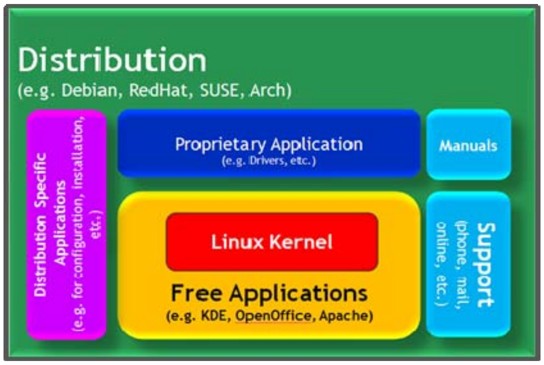

Distribution
(e.g. Debian, RedHat, SUSE, Arch)
Distribution Specific Applications
(e.g. for configuration, installation, etc.)
Proprietary Application
(e.g. Drivers, etc.)
Manuals
Linux Kernel
Free Applications
(e.g. KDE, OpenOffice, Apache)
Support
(phone, mail, online, etc.)

Fig.1 Linux-based operating system structure.

### 2.2 Classification of Linux-based Operating Systems

After the submission of Linux on the Internet, user groups, individuals and some firms in different parts of the world started to use this kernel to implement and develop their specific systems. This resulted in development of different Linux distributions. Although these distributions share a kernel but they are significantly different from on another in other aspects. Differences between developed distributions are in various classes including: installing and configuring the operating system, used desktop, support from hardware available in the market, etc. [12]

Developing these distributions continues like a tree during the time and although there is no exact statistic in hand about their number but according to data in sites (such as "distrowatch") more several thousands of Linux distributions are developed so far. [13]
Among the large number of Linux distributions, only e few are mother and main ones and the remaining are alternatives of the main distribution and have its characteristics. [14]

## 3. Methods to develop a Linux-based Operating System

Generally, there are tow methods to develop a Linux-based operating system:

- Remaster method
- LFS (Linux from Scratch)

### 3.1 Remaster

Remaster means recommending a distribution of Linux installed on the system and does not mean to propagate and spread a new distribution. [14] After installing a distribution, users apply some settings on the system (like installing software). Many of these settings are common among users. Hence, recommending a distribution with Remaster is considered.

### 3.1.1 Reasons of Remaster

This method is used for providence in time, facilitating the user's task after installing needed distribution and meeting user's needs (respecting their feedback). [15] It is important to note that there is no need, in developing a distribution with this method, to have specialty or information about computer programming and the knowledge of working with Linux terminal.

### 3.1.2 Remastering Software

Some software is available to be used in Remastering that facilitates distribution development is if one is working with a common application. Examples of such software are Remastersys, MySLAX Creator, Linux Live, etc. [16]

### 3.1.3 Remastering procedures

A. Installing the distribution
B. Installing desired Remastering software
C. Introducing the CD based on which the recommended scripts is being developed.
D. Adding developed packages (like required software).

E. Deleting extra packages from the CD (like password).
F. Selecting desired settings to run the system.
G. Developing the virtual image of the recommended script.
H. Copying the recommended script.
I. Providing external memory to use the developed scripts by a USB. [17]

It is possible in the sixth step to determine the graphic or text system running type, password for the root administrator and hardware settings.

### 3.1.4 Some Points in Remastering

In developing a successful and valuable distribution for the user, it is better to:

- Choose a popular distribution
- Have a chose near to the mother distribution (which results in higher stability after Remastering).
- Recommend the distribution based on Users' feedback.

## 3.2 LFS (Linux From Scratch)

LFS, in fact, is a textured and step-by-step developing method during which a Linux-based operating system is developed using a host system and the resource code. The method was invented by "Gerard Beekmans" under a project with the same name.

### 3.2.1 Advantages of LFS

Advantages of LFS method can be described as follows:

- Higher security

In this method resource codes are directly used to develop applications and for system configuration, thus higher security is provided since the risk of destructive codes and security holes is minimized.

- Higher stability of the system

In this method, all system partitions are added to the operating system according to need definition and type of system usage. Thus, extra software that lowers the stability of the system through changing various partitions is deleted.

- More flexibility

As mentioned above, it is possible, in this method, to delete and add different partitions of the operating system. Hence, personalization and alteration is possible in all layers, even in the kernel so that the mass of kernel could be reduced to some megabytes. Also, the system may be personalized for various architectures such as ARM, x86, x64, etc. this characteristic is applied particularly in embedded systems. [18]

### 3.2.2 LFS (Linux From Scratch)

Some primary requirements must be met to develop a Linux-based operating system through LFS. These requirements are divided into software and hardware classes.

On the part of hardware resources, we need a system to act as the host. The significant point about the host is that its processor architecture determines the architecture of the operating system which we are trying to develop. Also, a Linux distribution must be installed on the host system.

Software resources include some software packages, a number of software patches and the kernel of Linux operating system.

In our sample, the number of software packages reached 300.

### 3.2.3 Implementation Algorithm of LFS Method

Figure 2 shows the procedures of developing a Linux-based operating system.

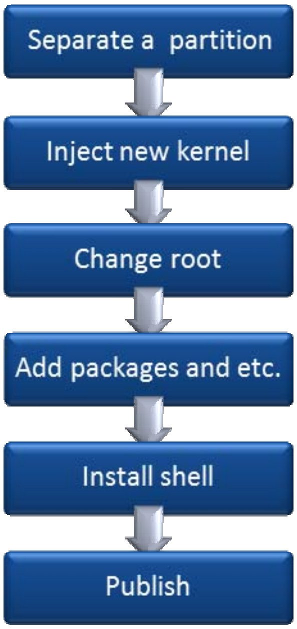


Fig.2 Linux-based operating system development flowchart.

In the first step, a partition of the hard disk is separated and formatted with file systems accepted by Linux (like file systems of ext, JFS, XFS, RaiserFS family). It is noteworthy that this procedure can be performed in simulator software such as SUN virtual machine.

The new kernel is injected to the provided partition in the second step. In fact, the new kernel is installed and the Linux file system is developed virtually. Of course, you

must notice that there is no need to complete development of this structure in this step (Figure 3) and only main partition are developed.
The following command could be used to do this task:

```
-v $LFS/{dev,proc,sys} Mkdir
```

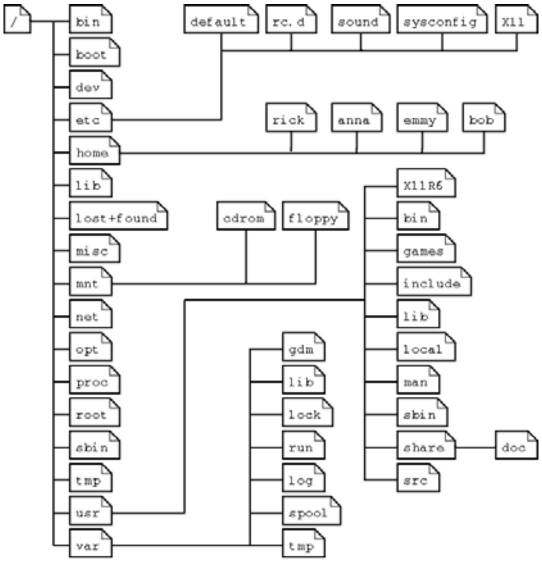


Fig. 3 File system tree structure in linux.

Then, the relation between the shell and the kernel f the host system is cut and the shell of host system is connected to the new kernel. Thus, we can exert changes on the new kernel or install new software packages on it. To do this, the current system's shell (which is the host system) must identify our installed kernel as the operating system kernel. The following command, indeed, does this task:

```
chroot "$LFS" /tools/bin/env -i \
HOME=/root TERM="$TERM" PS1='\u:\w\$ ' \
PATH=/bin:/usr/bin:/sbin:/usr/sbin:/tools/bin \
/tools/bin/bash --login +h
```

This command, in fact, introduces LFS variable to the system shell as the root directory (explanation of all used switched is not presented here). Next, the system standard directory tree is developed (Figure 3).
In the fourth step, have a system including a kernel and the standard directory tree structure the current system shell of which is installed through "chroot" command and it is controlled. Now, required packages are installed and the system is personalized. Packages installed in this step are responsible for some tasks of the system and we use the term of "essential packages" for them. Some of these packages include:

Autoconf
Automake
Bash
Binutils
Bison
Bzip2
Check
Coreutils
DejaGNU
Diffutils
E2fsprogs
Expect
File
Findutils
Flex
Gawk
GCC
GDBM
Gettext
Glibc
GMP
Grep
Groff
....

(The number of essential packages is very large and we named just some of them). In this step, also, boot time scripts must be set. These scripts determine the kind of configuration used in the system while loading in the main memory. Here, we name some driver scripts and their function:

Checkfs:
Checks the integration of file system before the system connects them (except for JFS).

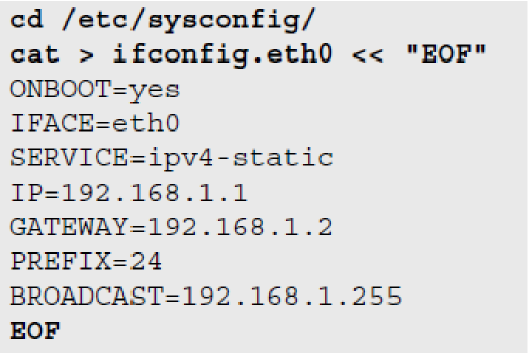
Console:
Loads the table keyboard keys for the demanded setting and is responsible for determining the font of the display and monitor.
The reason for the presence of driver scripts is that Linux uses a special driving option called "sysVinit" which allows the system to run in different run-levels.
Six main runs are described briefly:

0: halt status
1: single-user status
2:multi-user status without access to network
3:multi-user status with access to network
4: reserve status for personalization
5: as before (usually for entering graphic environments).
6: re-run the system (The default status is run-level 3).

It is clear that developing various run-levels is directly related to driving scripts. Among other important settings of this step is network setting. Files to set and configure the network relation must be developed in this step. Which network relations run by the system depends on scripts available in the /etc/sysconfig/ path. There must be a configuration file for each network relation. For example:

ifconfig.xyz

where, xyz is the name of the network relation, like eth0.



```
cd /etc/sysconfig/
cat > ifconfig.eth0 << "EOF"
ONBOOT=yes
IFACE=eth0
SERVICE=ipv4-static
IP=192.168.1.1
GATEWAY=192.168.1.2
PREFIX=24
BROADCAST=192.168.1.255
EOF
```

In the fifth step, the shell of new operating system is installed and configured. Various shells may be used. Then, the old shell disconnects from new kernel. Also, in this step, files relevant to boot ability of the operating system are set and the creation process is accomplishes and finished. Thus, the host system and the new system are separated from each other.

In the last step, the new born system is ready to publish and can be used as a complete operating system.

## 4. Conclusions

In the present world, which is called "communication era", information security is undoubtedly one of the most important principles in communications in all levels. The only way to achieve this goal is the software independence of countries, security organizations and even small and big business agencies. But first some infrastructures must be provided to realize this independence.

Operating system is one of the most significant components of communication infrastructures as the most essential software for every computer. It has been tried to design and develop a local system through LFS and taking some steps toward this big goal by preserving Remastering advantages.

Of course, the developed system is still in its first steps and must pass a long way to be completed and be proper for public usage.

## References

[1] Haas, J. *Linux, the Ultimate Unix*. 2004; Available from: http://linux.about.com/cs/linux101/a/linux_2.htm.

[2] G. K. Hartman, J.C., A. McPherson (2008) *Linux Kernel Development*.

[3] Lee (2011) *Is Linux Still The Safest Operating System?*

[4] West, M., *Linux Fundamentals A Practical Guide to Learning Linux*. 2005. 190.

[5] Wikipedia (2012) *Usage share of operating systems*.

[6] *OS Platform Statistics*.

[7] Runjorel (2011) *Why Russia, China, and Iran Love Linux and Open Source*.

[8] (2009) *What Is Linux: An Overview of the Linux Operating System*

[9] Mathieson, J., *Linux Overview*. 2009, IEEE.

[10] Vermeulen, S., *Configuring a Linux Kernel*, in *Linux Sea*. 2011.

[11] Wikipedia (2012) *Linux distribution*.

[12] Matzan, J. (2006) *The differences between Linux distributions*.

[13] Distrowatch. *Linux distrobitions*. Available from: www.Distrowatch.com.

[14] Wikipedia (2011) *Software remastering*.

[15] Hartley, M., *The Joy of Remastering your Own Linux Distro*. 2011: p. 2.

[16] Wikipedia (2012) *List of remastering software*.

[17] Farizluqman (2011) *MyOS*.

[18] Beekmans, G., *Linux From Scratch*.