

Date: 13.06.2025

Complete Linux

Contents

1. Linux.....	2
2. Linux Distributions (Distros).....	3
3. Linux Boot Process.....	4
a. BIOS / UEFI Initialization.....	4
b. MBR / GPT.....	5
c. GRUB	5
d. Kernel	5
e. init / systemd.....	6
f. Runlevel Programs	6
4. Standard File System Hierarchy	7
5. Linux Commands	8
a. Basic Commands	9
b. File and Directory Commands.....	9
c. Text Processing Commands	9
d. User and Group Management.....	10
e. Process Management	10
f. Disk and File System Management	10
g. Networking Commands.....	11
h. Package Management	11
i. Process Management	11
j. Compression and Archiving	12
k. System Monitoring	12



1. Linux








In 1969, Ken Thompson and Dennis Ritchie of Bell Laboratories developed the UNIX operating system which was later rewritten in C.

A decade or so later, Richard Stallman started working on the GNU (GNU is Not UNIX) project, the GNU kernel called Hurd, which unfortunately never came to completion.

During this time other efforts such as BSD, MINIX, etc. were developed to be UNIX like-systems. However, one thing that all these UNIX like-systems had in common was the lack of a unified kernel.

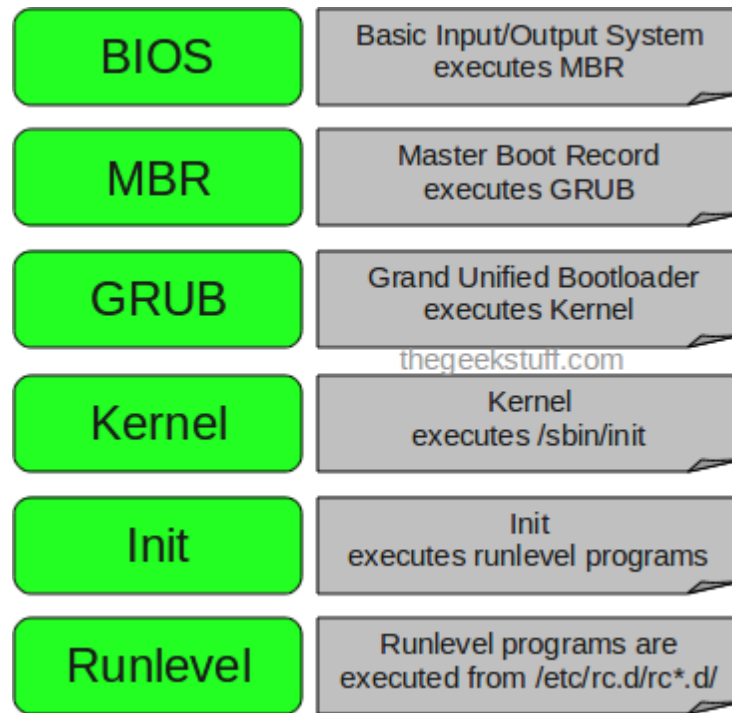
Then in 1991, Linus Torvalds started developing what we now know today as the Linux kernel.

2. Linux Distributions (Distros)

	Debian 	Ubuntu 	openSUSE 	Fedora 	RHEL 	Arch 	Gentoo 
Maintainer	Debian Project	Canonical Ltd.	openSUSE + SUSE	Fedora Project	Red Hat (IBM)	Arch Community	Gentoo Community
Release Model	Stable, Testing, Unstable	Regular + LTS	Tumbleweed (rolling), Leap (stable)	Regular (~6 months)	Long-Term Support (10+ yrs)	Rolling	Rolling (source-based)
Package Manager	APT (.deb)	APT (.deb) + Snap	Zypper (.rpm)	DNF (.rpm)	YUM/DNF (.rpm)	Pacman	Portage
Default File System	ext4	ext4	Btrfs (Leap), ext4 (Tumbleweed)	ext4	XFS	ext4	ext4
Stability	Very High	High	High (Leap), Medium (Tumble)	Moderate	Enterprise-grade	Moderate	Depends on user
Target Users	Advanced users, servers	Beginners, desktops, cloud	Developers, sysadmins	Developers, enthusiasts	Enterprises	Power users, enthusiasts	Experts, tinkerers
Pros	Very stable, open-source focused	User-friendly, strong hardware support	YaST tool, flexible models	Cutting-edge tech, GNOME default	Paid support, long lifecycle	Bleeding-edge, highly customizable	Full control, optimized for performance
Cons	Older packages, less beginner-friendly	Snap criticisms, heavier	Less software availability, slower Leap updates	Short support cycle, upgrade demands	Paid, closed-source elements	DIY install, fragile if misused	Very steep learning curve, time-intensive

3. Linux Boot Process

There are 6 high level stages of a typical Linux Boot Process.



a. BIOS / UEFI Initialization

BIOS	UEFI
<ul style="list-style-type: none"> • BIOS stands for Basic Input/Output System • Performs some system integrity checks • Searches, loads, and executes the boot loader program. • It looks for boot loader in floppy, cd-rom, or hard drive. You can press the BIOS key during the BIOS startup to change the boot sequence. • Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it. • So, in simple terms BIOS loads and executes the MBR or GPT boot loader. 	<ul style="list-style-type: none"> • UEFI stands for Unified Extensible Firmware Interface. • Runs in 32-bit or 64-bit protected mode for faster and more secure booting. • Provides features like secure boot and graphical setup menus. • Has become the standard firmware on most modern Linux systems. • Many systems still support legacy BIOS mode for compatibility. • Enabling UEFI with secure boot is recommended to protect against boot-level

b. MBR / GPT

MBR	GPT
<ul style="list-style-type: none"> • MBR stands for Master Boot Record. • Located in the first sector (LBA 0) of the bootable disk (e.g., /dev/sda). • Exactly 512 bytes: <ul style="list-style-type: none"> ◦ 446 bytes: Primary bootloader code ◦ 64 bytes: Partition table (up to 4 entries) ◦ 2 bytes: Boot signature (0x55AA) • BIOS loads and executes the MBR to initiate the Stage 1 bootloader (e.g., GRUB or LILO). • Supports disks up to 2 TB and allows up to 4 primary partitions. 	<ul style="list-style-type: none"> • GPT stands for GUID Partition Table. • Used with UEFI firmware, it supports drives up to 9.4 ZB. • GPT stores partition data in LBA 1, with a backup header at the end of the disk. • Includes a protective MBR for legacy system compatibility. • Supports a dedicated EFI System Partition (ESP) to store bootloader files. • Offers better redundancy and structure compared to MBR.

c. GRUB

- GRUB stands for Grand Unified Bootloader.
- If you have multiple kernel images installed on your system, you can choose which one to be executed.
- GRUB displays a splash screen, waits for few seconds, if you don't enter anything, it loads the default kernel image as specified in the grub configuration file.
- GRUB has the knowledge of the filesystem (the older Linux loader LILO didn't understand filesystem).
- Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this).
- It contains kernel and initrd image.
- So, in simple terms GRUB just loads and executes Kernel and initrd images.

d. Kernel

- Mounts the root file system as specified in the "root=" in grub.conf
- Kernel executes the /sbin/init program
- Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1.
- initrd stands for Initial RAM Disk.
- initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted.
- It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

e. init / systemd

init	systemd
<ul style="list-style-type: none"> Looks at the /etc/inittab file to decide the Linux run level. Following are the available run levels <ul style="list-style-type: none"> 0 – halt 1 – Single user mode 2 – Multiuser, without NFS 3 – Full multiuser mode 4 – unused 5 – X11 6 – reboot Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program. Identify the default run level: `grep initdefault /etc/inittab` You can set the default run level from 0 to 6 but it is advised to set it to either 3 or 5. 	<ul style="list-style-type: none"> systemd manages Linux startup by tracking service dependencies via unit files. Enables parallel and ordered service initialization for faster boots. Uses boot targets (like graphical.target) similar to traditional runlevels. Parses unit files (e.g., /etc/systemd/system/*.service) to build a dependency graph. The [Install] section links services to boot targets. systemd improves boot speed and management but requires adapting to new init concepts.

f. Runlevel Programs

- During boot, runlevel programs execute from directories based on the runlevel:
 - Runlevel 0: /etc/rc.d/rc0.d/
 - Runlevel 1: /etc/rc.d/rc1.d/
 - ... up to Runlevel 6: /etc/rc.d/rc6.d/
- Symbolic links like /etc/rc0.d point to /etc/rc.d/rc0.d/ for convenience.
- Programs in these directories start with **S** (startup) or **K** (kill/shutdown).
- Numbers following S/K indicate execution order (sequence number).
- Example: S12syslog starts syslog before S80sendmail starts sendmail.

4. Standard File System Hierarchy

Directory Type	Directory	Purpose
Top level Directory	/	Root directory — top of the filesystem tree
Essential System Directories	/bin	Essential user binaries (commands needed in single-user mode)
	/boot	Boot loader files, kernel, initrd
	/dev	Device files (hardware devices like disks, terminals)
	/etc	Host-specific system configuration files
	/lib	Essential shared libraries and kernel modules
	/sbin	System binaries (commands for system administration)
User Directories	/root	Home directory for the root user
	/home	User home directories
	/media	Mount points for removable media (USB, CD-ROM)
	/mnt	Temporary mount point for mounting filesystems
Virtual or Temporary File Systems	/proc	Virtual filesystem providing process and kernel info
	/sys	Virtual filesystem for kernel device and system info
	/run	Runtime variable data, information about running system
	/tmp	Temporary files (cleared on reboot)
Optional or Variable Components	/opt	Optional application software packages
	/srv	Data for services provided by the system
	/usr	Secondary hierarchy for read-only user data; contains binaries, libraries, documentation
	/var	Variable files like logs, spool files, mail, cache

5. Linux Commands

Category	Key Linux Commands / Utilities
Basic Commands	echo, pwd, date, whoami, hostname, uptime
File and Directory Management	ls, cd, pwd, mkdir, rmdir, rm, cp, mv, find, stat, file
Text Processing	cat, tac, head, tail, grep, sed, awk, cut, sort, uniq, tr, wc
Process Management	ps, top, htop, kill, killall, nice, renice, jobs, fg, bg
User Management and Permissions	useradd, usermod, userdel, passwd, groups, id, chmod, chown, chgrp, sudo
Networking	ping, traceroute, ifconfig (deprecated), ip, netstat, ss, scp, ssh, wget, curl, dig
Disk and Filesystem Management	df, du, mount, umount, fsck, blkid, lsblk, parted, fdisk, tune2fs
Package Management	apt, dpkg (Debian/Ubuntu), yum, dnf (Fedora/RHEL), pacman (Arch), zypper (openSUSE), rpm
Shell and Scripting	bash, sh, zsh, dash, source, alias, export, env, cron, at
System Information and Monitoring	uname, top, vmstat, iostat, free, uptime, dmesg, lscpu, lsusb, lspci
Job Scheduling and Automation	cron, crontab, at, batch, systemd timers
Security and Encryption	iptables, firewallld, ufw, openssl, gpg, ssh-keygen, fail2ban, selinux
Archiving and Compression	tar, gzip, gunzip, bzip2, xz, zip, unzip, 7z
System Services and Daemon Management	systemctl, service, chkconfig, init, journalctl
Hardware Information and Configuration	lshw, lsusb, lspci, hwdm, dmidecode, modprobe
Development Tools (compilers, debuggers)	gcc, make, gdb, strace, valgrind, ld, objdump
Virtualization and Containers	docker, podman, kubect, virsh, qemu, vboxmanage
Backup and Recovery	rsync, tar, dd, dump, restore, timeshift
Logging and Auditing	journalctl, logger, auditctl, ausearch, syslog, rsyslog
Localization and Internationalization	locale, localectl, timedatectl, gettext

a. Basic Commands

Command	Description	Basic Syntax
echo	Display a line of text	echo "Hello World"
pwd	Print current working directory	pwd
date	Show or set system date/time	date
whoami	Show current user	whoami
clear	Clear terminal screen	clear
man	Display manual pages	man ls

b. File and Directory Commands

Command	Description	Basic Syntax
ls	List directory contents	ls -l /home
cd	Change directory	cd /path/to/dir
mkdir	Create new directories	mkdir new_folder
rmdir	Remove empty directories	rmdir empty_folder
rm	Remove files/directories	rm file.txt
touch	Create empty file / update timestamp	touch file.txt
cp	Copy files/directories	cp file1 file2
mv	Move or rename files/directories	mv oldname newname
stat	Show file/directory status	stat file.txt

c. Text Processing Commands

Command	Description	Basic Syntax
cat	Concatenate and display files	cat file.txt
head	Display first lines of a file	head -n 10 file.txt
tail	Display last lines of a file	tail -n 10 file.txt
grep	Search text with patterns	grep "pattern" file.txt
sed	Stream editor for filtering text	sed 's/old/new/g' file.txt
awk	Pattern scanning and processing	awk '{print \$1}' file.txt
sort	Sort lines of text files	sort file.txt
cut	Extract sections of lines	cut -d',' -f1 file.txt
wc	Count lines, words, bytes	wc -l file.txt
uniq	Remove duplicate lines	uniq filename

d. User and Group Management

Command	Description	Basic Syntax
who	Show logged-in users	who
id	Show user and group IDs	id username
useradd	Add a new user	useradd username
userdel	Delete a user	userdel username
usermod	Modify user account	usermod -aG group username
groupadd	Add new group	groupadd groupname
passwd	Change user password	passwd username
groups	Show groups user belongs to	groups username

e. Process Management

Command	Description	Basic Syntax
ps	Show current processes	ps aux
top	Real-time process monitoring	top
htop	Interactive process viewer	htop
kill	Send signal to terminate process	kill PID
killall	Kill processes by name	killall process_name
nice	Start process with modified priority	nice -n 10 command
renice	Change priority of running process	renice 5 -p PID
pgrep	Find process IDs by name	pgrep process_name

f. Disk and File System Management

Command	Description	Basic Syntax
df	Show disk space usage	df -h
du	Estimate file/directory space usage	du -sh /path
mount	Mount a filesystem	mount /dev/sda1 /mnt
umount	Unmount a filesystem	umount /mnt
fsck	Filesystem consistency check	fsck /dev/sda1
blkid	Display block device attributes	blkid
Fdisk	Disk partitioning tool for MBR	Fdisk /dev/sda
parted	Disk partitioning tool for GPT	parted /dev/sda print
lsblk	List information about block devices	lsblk
mkfs	Create filesystem	mkfs.ext4 /dev/sda1

g. Networking Commands

Command	Description	Basic Syntax
ifconfig	Configure network interfaces (deprecated)	ifconfig eth0 up
ip	Modern network configuration tool	ip addr show
ping	Send ICMP echo requests	ping google.com
tracert	Trace network route	tracert google.com
netstat	Network connections, routing tables (deprecated)	netstat -tulnp
ss	Display socket statistics	ss -tuln
wget	Download files from the web	wget http://example.com/file
curl	Transfer data from or to a server	curl http://example.com
nslookup	DNS query tool	nslookup google.com

h. Package Management

Command	Description	Basic Syntax
apt	Debian/Ubuntu package manager	apt install package
dpkg	Debian package manager tool	dpkg -i package.deb
yum	RHEL/CentOS package manager (deprecated)	yum install package
dnf	Fedora/RHEL/CentOS package manager	dnf install package
zypper	openSUSE package manager	zypper install package
pacman	Arch Linux package manager	pacman -S package
portage	Gentoo package manager	emerge package

i. Process Management

Command	Description	Basic Syntax
ps	Show current processes	ps aux
kill	Send signals to processes	kill PID
killall	Kill processes by name	killall processname
nice	Run process with a specific priority	nice -n 10 command
renice	Change priority of a running process	renice 10 -p PID
pgrep	Find process IDs by name	pgrep processname

j. Compression and Archiving

Command	Description	Basic Syntax
tar	Archive files	tar -cvf archive.tar files
gzip	Compress files	gzip file
gunzip	Decompress gzip files	gunzip file.gz
bzip2	Compress files	bzip2 file
bunzip2	Decompress bzip2 files	bunzip2 file.bz2
xz	Compress files	xz file
unxz	Decompress xz files	unxz file.xz
zip	Create zip archives	zip archive.zip files
unzip	Extract zip archives	unzip archive.zip

k. System Monitoring

Command	Description	Basic Syntax
top	Real-time system monitor	top
htop	Interactive system monitor	htop
vmstat	Virtual memory stats	vmstat 1
iostat	CPU and I/O stats	iostat
free	Memory usage	free -h
sar	System activity report	sar -u 1 3
uptime	System uptime and load average	uptime
pidstat	Process statistics	pidstat 1