
NETWORK MANAGEMENT AND ADMINISTRATION

Chapter 1

Introduction to Network Management and Administration

Detailed Study Notes for Engineering Students

With Diagrams, 15 Multiple Choice Questions with Answer Key and 10 Broad Questions

Poly Notes Hub

Contents

1.1 Duties of the System Administrator	4
Major Duties of a System Administrator	4
Linux as well as Other OS Administrator	4
Steps of Installing and Configuring Servers	5
1.2 Planning the Network	7
Describing the Topologies	7
Planning and Implementing the Security	9
1.3 Steps of Kickstart Installation	11
Installing the Kickstart Configurator	11
Boot Loader Option Screen	12
Partition	12
Network Configuration	13
Authentication	13
Firewall Configuration	13
Creating a Bootable CD-ROM	14
1.4 System Start-up and Shutdown	16
Examining the Boot Process	16
Boot Loader	16
The Kernel	17
1.5 The File System	19
Understanding the File System Structure	19
Different OS Supported File Systems	20
1.6 Examining the System Configuration Files	23
Important configuration files	23
Chapter Summary	25
Multiple Choice Questions (15)	27
Answer Key	29
Broad Questions (10)	30

Chapter 1: Introduction to Network Management and Administration

Computers in a modern institution rarely work alone. They are joined together by cables and wireless links, and they share files, printers, databases, e-mail and Internet access. A network of this kind does not run by itself: someone must install the servers, plan the cabling and addressing, create user accounts, protect the data, watch the performance and repair faults. This work is called **network management and administration**, and the person who performs it is the **system administrator** (sysadmin) or **network administrator**.

This chapter introduces the duties of the administrator, the way a network is planned and secured, how many servers are installed in one stroke with the Kickstart method, how a Linux system starts and stops, how data is organised in the file system and which configuration files control the behaviour of the machine. Red Hat based Linux (Red Hat Enterprise Linux, CentOS, Fedora) is used for most examples because it is the most common server platform and the one on which Kickstart was developed.

Topics covered in this chapter

Unit	Topic
1.1	Duties of the System Administrator; Linux and other OS administrator; steps of installing and configuring servers
1.2	Planning the network: topologies, planning and implementing security
1.3	Steps of Kickstart installation: Kickstart Configurator, boot loader options, partition, network configuration, authentication, firewall configuration, creating a bootable CD-ROM
1.4	System start-up and shutdown: boot process, boot loader, the kernel
1.5	The file system: structure and file systems supported by different operating systems
1.6	Examining the system configuration files

1.1 Duties of the System Administrator

A **system administrator** is the person responsible for the installation, configuration, security, maintenance and troubleshooting of computer systems and networks, so that users and services stay available, reliable and secure. In a large company the work is divided among specialists (network administrator, database administrator, security administrator, storage administrator). In a college, a small office or a start-up, one person often performs all these roles.

The administrator works with a special account called the **super user** (**root** on Linux and UNIX, **Administrator** on Windows). This account can read, change or delete anything, so it must be used carefully. A good practice is to log in as an ordinary user and gain extra privileges only when needed (with **su** or **sudo**). This follows the **principle of least privilege**.

Major Duties of a System Administrator

Duty area	What the administrator does	Common tools / commands
User and group management	Creates, modifies, locks and deletes accounts; sets passwords, quotas and file permissions	useradd, usermod, userdel, passwd, chage, groupadd
Software and patch management	Installs, upgrades and removes software; applies security patches and updates	rpm, yum, dnf, apt, dpkg
Hardware and storage management	Adds disks, creates partitions and file systems, mounts storage, configures RAID and LVM	fdisk, parted, mkfs, mount, lvm, mdadm
Network management	Configures IP addresses, routing, DNS and DHCP; monitors links and traffic	ip, ifconfig, route, netstat, ping, traceroute, nmap
Security management	Configures firewalls, access control and auditing; reviews logs; responds to incidents	iptables, firewalld, SELinux, sudo, last, /var/log
Backup and recovery	Schedules backups, tests restores and prepares a disaster recovery plan	tar, dump, rsync, cpio
Performance monitoring and tuning	Watches CPU, memory, disk and network usage; finds bottlenecks	top, vmstat, iostat, free, sar
Automation and scheduling	Writes shell scripts and schedules repeating jobs	cron, crontab, at, bash scripts
Documentation and user support	Maintains network diagrams, logs and change records; helps users with problems	help-desk system, wiki, log book
Policy and compliance	Enforces password policy, acceptable use policy and software licensing rules	written policies, audits

Qualities of a good administrator

- Strong knowledge of operating systems, networking and security.
- Logical thinking and patience in troubleshooting.
- Habit of documenting every change and taking a backup before making it.
- Ability to explain technical matters to non-technical users.
- Willingness to keep learning because tools and threats keep changing.

Linux as well as Other OS Administrator

Most of the ideas of administration are the same on every operating system: accounts, services, storage, networking, security and backups. The tools and file locations differ. A Linux administrator mainly works from the command line and edits plain text configuration files, while a Windows administrator mainly uses graphical consoles and PowerShell, with settings stored in the registry and in Active Directory.

Aspect	Linux	Windows Server	UNIX (Solaris, AIX, BSD)
Super user account	root	Administrator	root
Main admin interface	Shell, vi/nano, Webmin, Cockpit	Server Manager, MMC consoles, PowerShell	Shell, vendor tools (SMIT on AIX)
User database	/etc/passwd, /etc/shadow, LDAP	Local SAM and Active Directory	/etc/passwd, /etc/shadow, NIS
Package management	rpm, yum, dnf, dpkg, apt	MSI installers, Windows Update, winget	pkgadd (Solaris), installp (AIX), pkg
Service management	systemd, SysV init scripts	Services console, sc command	SMF (Solaris), rc scripts
Default file systems	ext4, XFS, Btrfs	NTFS, ReFS	UFS, ZFS, JFS2
Remote administration	SSH	RDP, WinRM, PowerShell remoting	SSH
Licence	Open source (GPL and others)	Commercial	Commercial or BSD licence

Additional tasks of administrators of other operating systems:

- **Windows administrator:** manages Active Directory domains, Group Policy, DNS, DHCP and file-server roles, and applies updates through WSUS.
- **UNIX administrator:** manages commercial UNIX servers, often with special hardware, logical domains and vendor-specific clustering.
- **Mixed-network administrator:** connects Linux and Windows systems, for example using Samba for file sharing or joining Linux hosts to Active Directory.

Steps of Installing and Configuring Servers

A server is a computer that provides a service (web, e-mail, DNS, files, printing) to other computers called clients. Installing a server in a planned, repeatable way reduces mistakes and security holes. The usual steps are:

1. **Define the requirements.** Decide the role of the server, the number of users, the expected load and the availability needed.
2. **Select the hardware or virtual machine.** Check CPU, RAM, disk space, network cards and RAID against the Hardware Compatibility List of the operating system.
3. **Choose the operating system and obtain the media.** Download the installation image and verify its checksum (for example with sha256sum).
4. **Install the operating system.** Use an attended installation or an automated Kickstart installation; create a sensible partition layout; install only the packages needed (minimal installation).
5. **Perform the initial configuration.** Set the host name, a static IP address, DNS servers, time zone and time synchronisation (NTP); set a strong root password and create a normal administrative user.
6. **Update the system.** Apply all available security patches before the server is placed on the network.
7. **Install the server software.** Install the required packages, for example Apache, BIND, Postfix or Samba.

8. **Configure the service.** Edit its configuration files and set options such as ports, document roots, zones, users and access rules.
9. **Secure the server.** Enable the firewall and SELinux, disable unused services, harden SSH and restrict permissions.
10. **Start and test the service.** Start the daemon, enable it at boot and test from a client computer.
11. **Set up monitoring, logging and backup.** Make sure logs are rotated and that backups are scheduled and tested.
12. **Document the installation.** Record the configuration, passwords (in a secure store), IP addresses and the date of every change.

Common servers on a Linux system

Server	Package	Daemon (service)	Main configuration file	Port
Web	httpd (Apache)	httpd	/etc/httpd/conf/httpd.conf	80, 443
DNS	bind	named	/etc/named.conf	53
DHCP	dhcp	dhcpd	/etc/dhcp/dhcpd.conf	67, 68 (UDP)
FTP	vsftpd	vsftpd	/etc/vsftpd/vsftpd.conf	21
Mail (SMTP)	postfix	postfix	/etc/postfix/main.cf	25
File and print	samba	smb, nmb	/etc/samba/smb.conf	139, 445
NFS	nfs-utils	nfs-server	/etc/exports	2049
Secure shell	openssh-server	sshd	/etc/ssh/sshd_config	22

Example: installing and starting an Apache web server on a Red Hat based system.

```
# yum install httpd                (install the package)
# vi /etc/httpd/conf/httpd.conf    (edit ServerName, DocumentRoot, Listen)
# systemctl start httpd            (start the service now)
# systemctl enable httpd           (start it automatically at boot)
# firewall-cmd --permanent --add-service=http
# firewall-cmd --reload            (open port 80 in the firewall)
# curl http://localhost            (test the web server)
```

On older releases the equivalent commands are `service httpd start` and `chkconfig httpd on`.

1.2 Planning the Network

Network planning is the process of deciding, before any equipment is bought or installed, what the network must do and how it will be built. A network that grows without a plan becomes slow, costly, insecure and difficult to repair. Good planning saves money and makes later expansion easy.

Steps in planning a network

1. **Identify requirements:** number of users and computers, applications, bandwidth needs, services required and expected growth.
2. **Survey the site:** study the building layout, cable routes, power supply and location of the server room.
3. **Select the topology:** bus, star, ring, mesh, tree or hybrid.
4. **Select media and devices:** UTP Cat 6 or fibre cable, switches, routers, wireless access points, firewalls and servers.
5. **Design the IP addressing plan:** choose private address ranges, subnets, VLANs, static and DHCP pools.
6. **Plan the network services:** DNS, DHCP, file, print, mail, proxy and web servers.
7. **Plan the security:** policies, firewall zones, authentication and backups.
8. **Prepare the budget** and choose vendors.
9. **Implement and test:** build a pilot, test performance, then roll out fully.
10. **Document and maintain:** keep network diagrams and an inventory up to date.

Describing the Topologies

The **topology** of a network is the arrangement of its nodes (computers, printers, devices) and the links connecting them. The **physical topology** describes the actual layout of cables and devices, while the **logical topology** describes how data flows between nodes. The six common topologies are shown in Fig. 1.1.

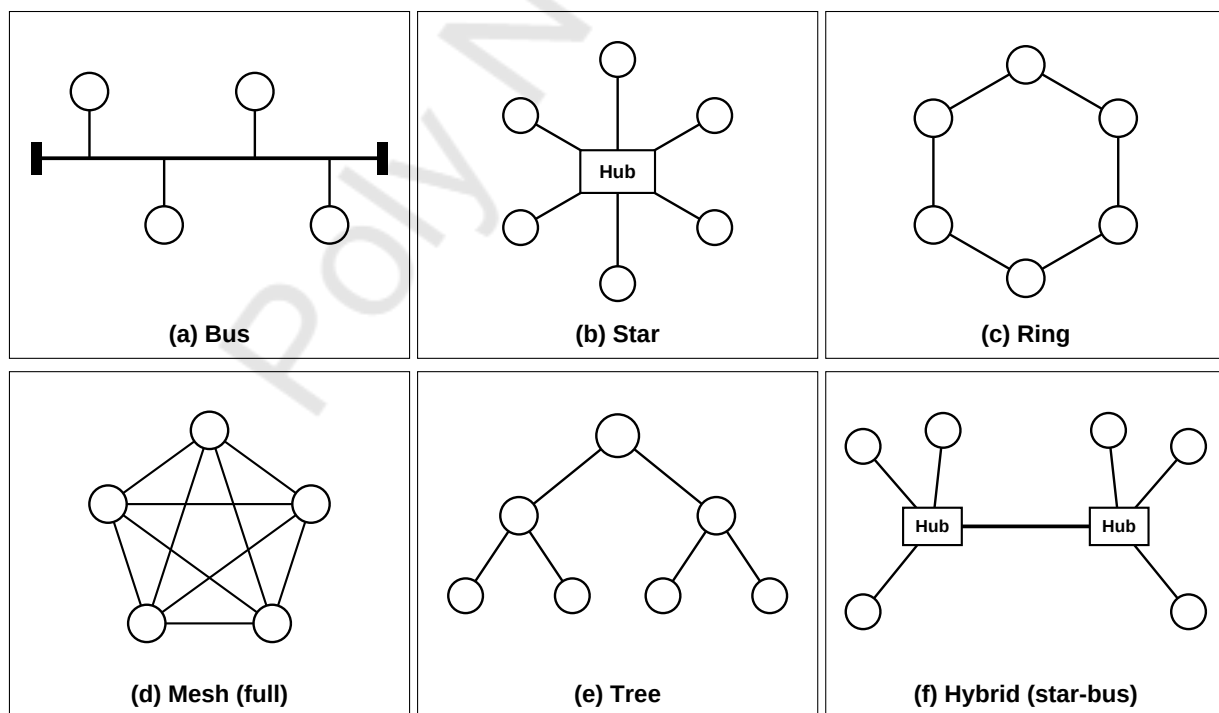


Fig. 1.1: Network topologies

Topology	Description	Advantages	Disadvantages	Typical use
Bus	All nodes share one backbone cable with terminators at both ends.	Cheap; easy to install; needs little cable.	Failure of the backbone stops the whole network; collisions; hard to troubleshoot.	Small, old Ethernet (10BASE2).
Star	Every node connects to a central hub or switch.	Easy to add or remove nodes; a cable fault affects only one node; easy management.	Central device is a single point of failure; needs more cable.	Almost all modern LANs.
Ring	Each node connects to exactly two neighbours forming a closed loop; data travels in one direction (token passing).	Orderly access; no collisions; good performance under load.	One broken link or node can stop the ring (unless dual ring); adding nodes disturbs the network.	Token Ring, FDDI, SONET.
Mesh	Nodes are connected to many or all other nodes (full or partial mesh).	Very reliable; many alternate paths; easy fault isolation.	Expensive; many links and ports; complex to install.	WAN backbones, Internet core.
Tree	Hierarchy of star networks linked to a root node.	Scalable; each segment can be managed separately.	Failure of a root or upper-level device cuts off the branches below.	Campus and college networks.
Hybrid	Combination of two or more topologies (for example star-bus).	Flexible; can be fitted to the site.	Complex design; higher cost.	Large organisations.

Number of links in a full mesh: for n nodes, links = $n(n-1)/2$. For 6 nodes this is $6 \times 5 / 2 = 15$ links, and each node needs $n-1 = 5$ network ports. This is why full mesh is used only where reliability is more important than cost.

Factors for choosing a topology

- **Cost** of cable, devices and installation.
- **Size** of the network and distance between nodes.
- **Reliability** and the need for redundant paths.
- **Performance** (speed and traffic expected).
- **Ease of expansion** and troubleshooting.
- **Media** in use (twisted pair, fibre, wireless).

Planning an IP address scheme

A part of planning is dividing the address space into subnets for the departments. Example: the private network 192.168.10.0/24 is divided into four /26 subnets. Each subnet has $2^6 - 2 = 62$ usable host addresses (the network address and broadcast address cannot be assigned to hosts).

Department	Network address	Subnet mask	Usable host range
Administration	192.168.10.0/26	255.255.255.192	192.168.10.1 - 192.168.10.62
Computer Lab	192.168.10.64/26	255.255.255.192	192.168.10.65 - 192.168.10.126
Library	192.168.10.128/26	255.255.255.192	192.168.10.129 - 192.168.10.190
Servers and Wi-Fi	192.168.10.192/26	255.255.255.192	192.168.10.193 - 192.168.10.254

Planning and Implementing the Security

Network security means protecting the **confidentiality**, **integrity** and **availability** (the CIA triad) of data and services. Security must be planned together with the network and not added at the end.

Goal	Meaning	Example control
Confidentiality	Only authorised people can read the data.	Passwords, encryption, file permissions
Integrity	Data is not changed without permission.	Checksums, digital signatures, access control
Availability	Systems and data are usable when needed.	Backups, RAID, UPS, redundant links

Planning the security

1. **Risk assessment:** list the assets (servers, data, devices), the threats to them, the weak points (vulnerabilities) and the possible damage.
2. **Security policy:** write rules for acceptable use, passwords, remote access, backups, software installation and incident response.
3. **Security architecture:** divide the network into zones such as the internal LAN, the DMZ (public servers) and the outside Internet, and place a firewall between them.
4. **Select controls:** firewalls, antivirus, IDS/IPS, VPN, encryption, authentication methods.
5. **Training:** teach users about phishing, passwords and safe behaviour.

Common threats and countermeasures

Threat	Example	Countermeasure
Unauthorised access	Password guessing, brute-force login	Strong passwords, account lock-out, SSH keys, two-factor authentication
Malware	Viruses, worms, ransomware	Antivirus, regular patching, e-mail filtering, least privilege
Network attacks	Port scanning, sniffing, spoofing, denial of service	Firewall, IDS/IPS, VPN and SSL/TLS encryption, switch port security
Data loss	Disk failure, accidental deletion	Regular backups, RAID, off-site copy
Insider misuse	Staff abusing privileges	Least privilege, auditing, log review
Physical theft or damage	Stolen server, fire, power failure	Locked server room, CCTV, UPS, fire protection

Implementing the security

Security is applied in layers so that if one layer fails, the next still protects the system. This idea is called **defence in depth** (Fig. 1.2).

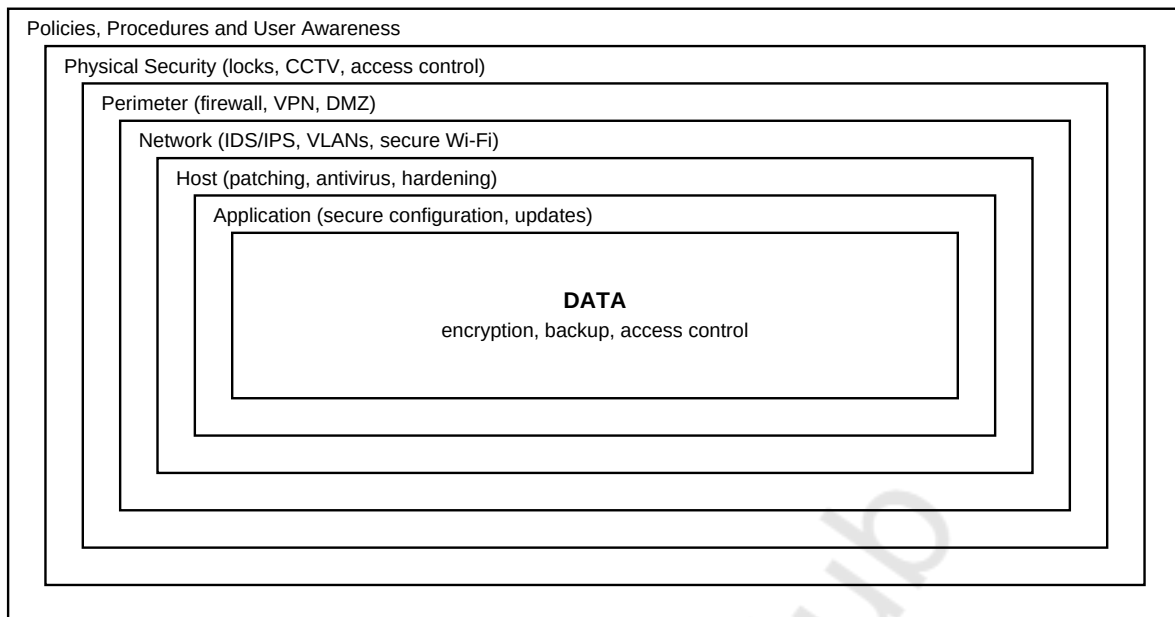


Fig. 1.2: Defence in depth (layers of security)

- **Physical security:** keep servers and network devices in a locked room with controlled access.
- **User and password security:** enforce password length and ageing (chage), disable unused accounts, disable direct root login over SSH, use `sudo`.
- **Host hardening:** remove unneeded packages and services, apply patches, enable SELinux, set correct file permissions.
- **Network security:** configure the firewall (iptables or firewalld), create a DMZ, use VLANs, use a VPN for remote access, use WPA2 or WPA3 for Wi-Fi.
- **Monitoring and logging:** collect logs with syslog, review them regularly, use an intrusion detection system such as Snort.
- **Encryption:** use SSH instead of Telnet, HTTPS instead of HTTP, SFTP instead of FTP.
- **Backup and recovery:** take regular backups, keep one copy off-site and test the restore.
- **Audit and review:** test the network periodically (vulnerability scan, penetration test) and update the policy.

Example: a simple iptables firewall that allows SSH and web traffic and blocks everything else. The allow rules are added before the default DROP policy so that the administrator is not locked out.

```
# iptables -A INPUT -i lo -j ACCEPT
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
# iptables -P INPUT DROP
# service iptables save
```

1.3 Steps of Kickstart Installation

Installing Linux by hand on many computers is slow and every installation may turn out a little different. **Kickstart** is the automated installation method of Red Hat based distributions. The administrator records all the answers to the installer's questions (language, partitions, network, packages, passwords and so on) in a text file called **ks.cfg**. The installer program, **Anaconda**, reads this file and installs the system without asking anything.

Advantages of Kickstart

- Many identical computers can be installed quickly (a whole computer laboratory in one session).
- Every installation has the same partitions, packages and security settings.
- The file can be stored on a CD-ROM, DVD, USB disk, NFS, HTTP or FTP server.
- It can run pre-installation and post-installation scripts to customise the system.
- The file is plain text, so it can be edited, copied and kept under version control.

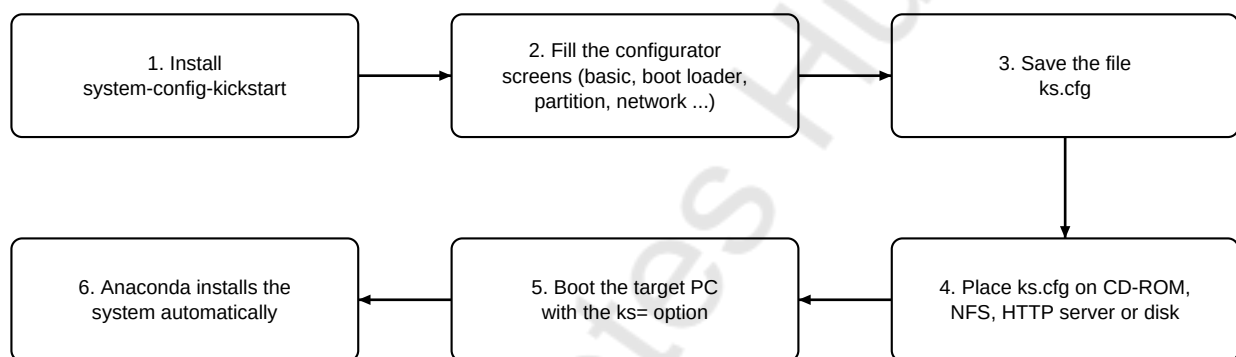


Fig. 1.3: Steps of a Kickstart installation

Installing the Kickstart Configurator

The **Kickstart Configurator** (system-config-kickstart) is a graphical tool that lets the administrator create a ks.cfg file by filling in screens instead of typing commands. It requires the X Window System, so it is installed on a workstation with a graphical desktop.

```
# yum install system-config-kickstart    (install the tool)
# system-config-kickstart                (start it from a terminal)
or: Applications -> System Tools -> Kickstart
```

After filling the screens, the file is saved with **File -> Save** as **ks.cfg**. The file can also be written by hand with any text editor. Note: newer Red Hat releases no longer include the graphical configurator; the installer itself saves a ready file named **/root/anaconda-ks.cfg** after every installation, and this can be edited and reused.

Screens of the Kickstart Configurator

Screen	Purpose	Matching ks.cfg command
Basic Configuration	Language, keyboard, time zone, root password, reboot after installation	lang, keyboard, timezone, rootpw, reboot
Installation Method	New installation or upgrade; source: CD-ROM, NFS, FTP, HTTP or hard drive	install, upgrade, cdrom, nfs, url, harddrive
Boot Loader Options	Boot loader type, password and location	bootloader

Screen	Purpose	Matching ks.cfg command
Partition Information	Clearing old partitions, creating new ones	clearpart, part, raid, volgroup, logvol
Network Configuration	Network interface and IP settings	network
Authentication	Password and user-database options	auth / authconfig
Firewall Configuration	Firewall and SELinux	firewall, selinux
Display Configuration	X Window System and video card settings	xconfig
Package Selection	Packages and groups to install	%packages
Pre-Installation Script	Commands run before installation starts	%pre
Post-Installation Script	Commands run after installation ends	%post

Boot Loader Option Screen

This screen decides how the installed system will start. A boot loader is the first program run after the BIOS; it loads the kernel (Section 1.4).

- **Install or upgrade boot loader:** install a new boot loader, update the existing one, or choose *Do not install a boot loader* if another loader (for example a network or third-party loader) will be used.
- **Boot loader type:** GRUB is the default; LILO is available on some releases.
- **GRUB password:** an optional password that must be typed to edit boot options; this stops unauthorised users from booting into single-user mode. The password can be stored encrypted.
- **Install location:** the *Master Boot Record (MBR)* or the *first sector of the boot partition*.
- **Kernel parameters:** extra options passed to the kernel, for example `console=ttyS0` or `quiet`.

```
boot loader --location=mbr --password=<password> --append="rhgb quiet"
```

Partition

The **Partition Information** screen tells the installer how to prepare the hard disk.

- **Clear Master Boot Record:** erase the existing MBR (`zerombr`).
- **Remove all existing partitions / remove existing Linux partitions / preserve existing partitions:** choose what happens to old data (`clearpart --all`, `--linux` or `--none`).
- **Initialize the disk label** (`--initlabel`): create a fresh partition table on a new disk.
- **Add partitions:** choose the mount point, file system type, size (fixed size, grow to fill the disk, or grow up to a maximum) and the drive.

A typical layout is shown below. Choose sizes according to the machine and its role.

Mount point	File system	Typical size	Purpose
/boot	ext4 (or ext3)	200 MB - 1 GB	Kernel and boot loader files; kept outside LVM
swap	swap	1 - 2 x RAM (small systems)	Virtual memory when RAM is full
/	ext4 or XFS	Remaining space	Root file system holding the operating system
/home	ext4 or XFS	As needed	User files (separate so that reinstalling does not erase data)

Mount point	File system	Typical size	Purpose
/var	ext4 or XFS	As needed	Logs, mail and spool (useful on servers)

```
zerombr
clearpart --all --initlabel
part /boot --fstype=ext4 --size=500
part swap --size=2048
part / --fstype=ext4 --size=1 --grow
```

Software RAID and LVM (logical volumes) can also be created from this screen (*raid*, *volgroup*, *logvol*).

Network Configuration

This screen sets up each network interface of the new system.

- **Network device:** eth0, eth1 and so on (names such as enp0s3 on newer releases).
- **Network type:** *DHCP* (address obtained automatically), *BOOTP*, or *static IP* where the IP address, netmask, gateway and name server are typed.
- **Activate on boot:** bring the interface up at start-up (*--onboot=yes*).
- **Host name:** the name of the computer on the network.

```
network --bootproto=dhcp --device=eth0 --onboot=yes
network --bootproto=static --device=eth0 --ip=192.168.10.20 --netmask=255.255.255.0 \
--gateway=192.168.10.1 --nameserver=192.168.10.5 --hostname=lab01.example.com
```

Authentication

The **Authentication** screen decides how users are verified at login.

- **Use shadow passwords:** store encrypted passwords in */etc/shadow*, which only root can read. Always enabled.
- **Password algorithm:** MD5 or SHA-512 (SHA-512 is stronger and is preferred).
- **NIS** (Network Information Service), **LDAP**, **Kerberos 5**, **Hesiod** and **SMB** (Windows domain): optional network-based authentication services where users are kept on a central server.
- **Name Service Switch cache:** caches lookups of user and group names to make logins faster.

```
auth --useshadow --passalgo=sha512
```

Firewall Configuration

This screen sets the basic firewall and SELinux policy of the new system.

- **Security level:** *Enable firewall* blocks all incoming connections that are not explicitly allowed; *Disable firewall* allows all connections (use only on a fully trusted network).
- **Trusted devices:** interfaces (for example the internal LAN card) from which all traffic is accepted.
- **Trusted services:** services allowed through the firewall, such as SSH, WWW (HTTP), Secure WWW (HTTPS), FTP, SMTP and Telnet.
- **SELinux:** Enforcing, Permissive or Disabled.

```
firewall --enabled --ssh --http
selinux --enforcing
```

Creating a Bootable CD-ROM

To install a computer that has no network boot, the `ks.cfg` file can be placed on a bootable CD-ROM together with the boot files. The steps are:

1. Copy the `isolinux/` directory (boot files) from the first installation disc to a working directory, for example `/tmp/ksiso`.
2. Save the file as `ks.cfg` in the same working directory.
3. Edit `isolinux/isolinux.cfg` and add the option `ks=cdrom:/ks.cfg` to the append line (or type it at the boot prompt).
4. Create the ISO image with `mkisofs`.
5. Burn the image to a CD with `cdrecord/wodim` or any CD-writing program.
6. Boot the target computer from this CD; the installation proceeds automatically.

```
# mkisofs -o ks.iso -b isolinux/isolinux.bin -c isolinux/boot.cat \
          -no-emul-boot -boot-load-size 4 -boot-info-table -R -J -v -T isolinux/
# cdrecord -v dev=/dev/sr0 ks.iso
```

If the configuration file is not on the boot media, its location is given at the boot prompt:

Boot option	Meaning
linux ks=cdrom:/ks.cfg	ks.cfg is read from the CD-ROM
linux ks=hd:sda1:/ks.cfg	ks.cfg is read from a partition of a hard disk
linux ks=nfs:server:/path/ks.cfg	ks.cfg is read from an NFS server
linux ks=http://server/ks.cfg	ks.cfg is read from a web server
linux ks=floppy	ks.cfg is read from a floppy disk

On newer releases (RHEL 7 and later) the option is written `inst.ks=` and installation media can be a DVD or USB disk.

A sample `ks.cfg` file

```
install
cdrom
lang en_US.UTF-8
keyboard us
timezone Asia/Kolkata
rootpw --iscrypted <encrypted-password>
auth --useshadow --passalgo=sha512
firewall --enabled --ssh --http
selinux --enforcing
bootloader --location=mbr
zerombr
clearpart --all --initlabel
part /boot --fstype=ext4 --size=500
part swap --size=2048
part / --fstype=ext4 --size=1 --grow
network --bootproto=dhcp --device=eth0 --onboot=yes
reboot

%packages
@core
httpd
%end

%post
echo "Installed by Kickstart" > /etc/motd
%end
```

1.4 System Start-up and Shutdown

Examining the Boot Process

Booting is the sequence of steps from pressing the power button until the system is ready for users. A system administrator must understand it in order to repair a machine that does not start. The steps are shown in Fig. 1.4 and explained in the table.

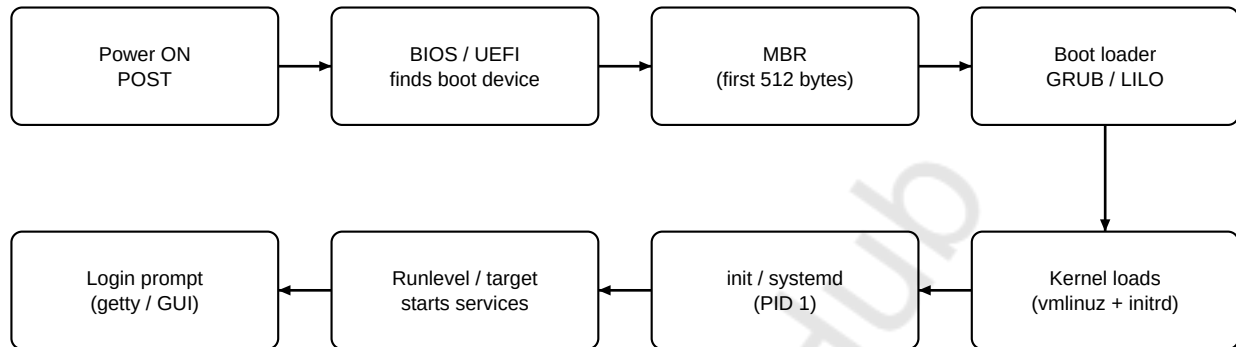


Fig. 1.4: Linux boot process

Stage	Component	What happens
1	Power-on and POST	The CPU starts executing firmware code. The Power-On Self-Test checks the memory, keyboard and other basic hardware.
2	BIOS / UEFI	The firmware looks for a bootable device in the configured boot order and reads its first sector.
3	MBR (Master Boot Record)	The first 512 bytes of the disk: 446 bytes of boot code, a 64-byte partition table and a 2-byte signature (0x55AA). The boot code starts the boot loader.
4	Boot loader (GRUB)	Shows a menu, lets the user choose a kernel or edit parameters, then loads the kernel and the initial RAM disk into memory.
5	Kernel	Decompresses itself, initialises memory and hardware, mounts the temporary root (initrd/initramfs), then mounts the real root file system (first read-only) and starts the first process.
6	init / systemd	The first process (PID 1). SysV init reads /etc/inittab to find the default runlevel; systemd reads its default target.
7	Runlevel scripts / targets	Services are started in order (for example networking, logging, sshd, httpd). SysV uses the S (start) and K (kill) scripts in /etc/rc.d/rcN.d.
8	Login	getty (or mingetty) shows the login prompt on each console; at runlevel 5 a graphical display manager is started.

Boot Loader

The **boot loader** is a small program that loads the operating system kernel into memory and passes control to it. When several operating systems are installed (dual boot), it offers a menu to choose one. Its main jobs are:

- to locate the kernel image and the initial RAM disk on the disk;
- to pass kernel parameters (for example the root device and the runlevel);
- to offer a menu and a command prompt for recovery;
- to chain-load other boot loaders, such as that of Windows.

Feature	LILO	GRUB (legacy)	GRUB 2
Full name	Linux Loader	GRand Unified Bootloader	GRUB version 2
Configuration file	/etc/lilo.conf	/boot/grub/grub.conf or menu.lst	/boot/grub2/grub.cfg (generated)
After a change	Must run the <code>lilo</code> command	No re-install needed; the file is read at boot	Run <code>grub2-mkconfig</code>
Interactive prompt	No	Yes	Yes
File system awareness	No (uses disk block addresses)	Yes	Yes
UEFI and GPT disks	No	Limited	Yes

A sample GRUB legacy configuration is given below.

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
title Red Hat Enterprise Linux (kernel x.y.z)
    root (hd0,0)
    kernel /vmlinuz-x.y.z ro root=/dev/VolGroup00/LogVol00 rhgb quiet
    initrd /initrd-x.y.z.img
```

- **default=0:** the first title entry is booted by default.
- **timeout=5:** the menu waits 5 seconds before booting the default entry.
- **root (hd0,0):** the first partition of the first disk holds /boot.
- **kernel:** path of the kernel image and its parameters; **ro** mounts the root file system read-only at first.
- **initrd:** the initial RAM disk that contains the drivers needed to reach the real root file system.

The Kernel

The **kernel** is the core of the operating system. It runs in privileged mode, controls the hardware and gives programs a safe and fair way to use it. Linux uses a **monolithic kernel** with **loadable modules**: drivers and file systems can be added or removed while the system is running.

Main functions of the kernel

- **Process management:** creates processes, schedules the CPU and handles signals.
- **Memory management:** allocates RAM, virtual memory and swap.
- **Device management:** controls hardware through device drivers.
- **File system management:** provides a uniform interface to many file systems (Virtual File System).
- **Networking:** implements the TCP/IP protocol stack and sockets.
- **Security:** enforces permissions, user IDs and SELinux rules.

Item	Description
/boot/vmlinuz-version	Compressed kernel image that the boot loader loads
/boot/initrd-version.img (initramfs)	Initial RAM disk with early drivers
/lib/modules/version/	Loadable kernel modules (drivers)
uname -r	Shows the running kernel version
dmesg	Shows the kernel messages produced during boot
lsmod, modprobe, rmmod	List, load and remove kernel modules

Item	Description
/etc/sysctl.conf	Run-time kernel parameters (for example IP forwarding)

Runlevels

A **runlevel** is a state of the system that decides which services are running. Systems using **systemd** use *targets* in place of runlevels.

Runlevel	systemd target	Meaning
0	poweroff.target	Halt (shutdown) the system
1	rescue.target	Single-user mode for maintenance and password recovery
2	multi-user.target	Multi-user without network file sharing (NFS)
3	multi-user.target	Full multi-user, text (command line) login
4	multi-user.target	Not used; can be defined by the administrator
5	graphical.target	Full multi-user with graphical login (X11)
6	reboot.target	Reboot the system

The command `runlevel` shows the current level and `init 3` or `systemctl isolate multi-user.target` changes it. The default level is set in `/etc/inittab` (line `id:5:initdefault:`) or with `systemctl set-default graphical.target`.

Shutdown

A Linux system must never be switched off abruptly because files held in memory buffers would not be written to disk and the file system could become damaged. The `shutdown` command warns the users, stops services in an orderly way and then halts or reboots the machine.

Command	Action
<code>shutdown -h now</code>	Halt the system immediately
<code>shutdown -h +10 "Maintenance"</code>	Halt after 10 minutes and send the message to users
<code>shutdown -r now</code>	Reboot immediately
<code>shutdown -c</code>	Cancel a scheduled shutdown
<code>halt, poweroff, reboot</code>	Halt, power off or reboot
<code>init 0 / init 6</code>	Go to runlevel 0 (halt) or 6 (reboot)
<code>systemctl poweroff / reboot</code>	Shutdown or reboot on systemd systems

Steps during shutdown

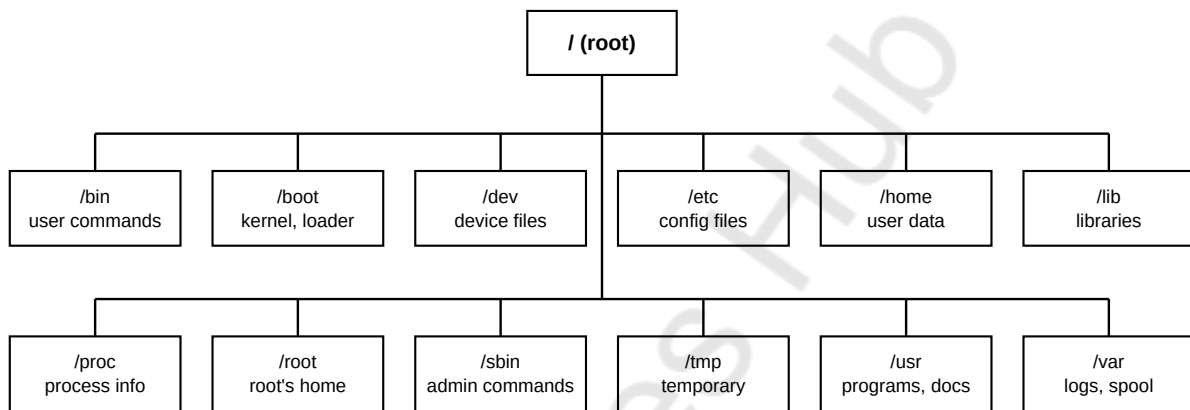
1. All logged-in users are notified and new logins are blocked.
2. The system moves to runlevel 0 (or 6 for reboot).
3. Services are stopped by the K (kill) scripts; running processes receive SIGTERM, and any that remain receive SIGKILL.
4. Buffers are flushed to disk (`sync`) and file systems are unmounted.
5. The machine is halted, powered off or restarted.

1.5 The File System

Understanding the File System Structure

A **file system** is the method an operating system uses to name, store, organise and retrieve data on a storage device. Linux follows the philosophy that *everything is a file*: ordinary documents, directories, hardware devices and even running processes are all accessed through the file system. All files are arranged in a single inverted tree that starts at the **root directory** (/). Other disks and partitions are attached to this tree at a **mount point**.

The layout of directories follows the **Filesystem Hierarchy Standard (FHS)**, so the same directory has the same purpose on every Linux distribution.



Other top-level directories: /media, /mnt, /opt, /srv, /sys, /run, /lost+found

Fig. 1.5: Linux file system hierarchy

Directory	Contents and purpose
/	Root of the entire tree
/bin	Essential user commands (ls, cp, cat) needed in single-user mode
/sbin	Essential system administration commands (fdisk, ifconfig, reboot)
/boot	Kernel, initrd and boot loader files
/dev	Device files (disks, terminals, printers), for example /dev/sda
/etc	System-wide configuration files
/home	Home directories of normal users
/root	Home directory of the super user
/lib, /lib64	Shared libraries and kernel modules needed by /bin and /sbin
/media, /mnt	Mount points for removable media (CD, USB) and temporary mounts
/opt	Optional add-on application software
/proc	Virtual file system showing processes and kernel information (stored in memory)
/sys	Virtual file system showing devices and drivers
/srv	Data for services such as web and FTP servers
/tmp	Temporary files (may be cleared at reboot)
/usr	User programs, libraries, documentation and shared data (/usr/bin, /usr/share)

Directory	Contents and purpose
/var	Variable data: logs (/var/log), mail, print spool, databases

Main parts of a Linux file system

Part	Function
Boot block	Holds boot code, if the partition is bootable
Superblock	Stores information about the whole file system: type, size, block size, number of free blocks and inodes
Inode table	An inode stores the details of one file: owner, permissions, size, time stamps and the addresses of its data blocks. The file name is not stored in the inode but in the directory entry.
Data blocks	Hold the actual contents of files and directories

File types and permissions

Symbol	Type	Example
-	Regular file	/etc/passwd
d	Directory	/home
l	Symbolic link	/bin/sh
c	Character device	/dev/tty
b	Block device	/dev/sda
p	Named pipe (FIFO)	created by mkfifo
s	Socket	/var/run sockets

The output of `ls -l` shows the type and permissions: `-rwxr-xr--` means a regular file where the owner has read, write and execute (7), the group has read and execute (5) and others have read only (4); the numeric mode is 754.

Important file system commands

Command	Use
<code>df -h</code>	Show used and free space of mounted file systems
<code>du -sh dir</code>	Show the space used by a directory
<code>lsblk, blkid</code>	List block devices and their UUIDs and types
<code>mkfs.ext4 /dev/sdb1</code>	Create (format) an ext4 file system
<code>mount /dev/sdb1 /mnt, umount /mnt</code>	Attach and detach a file system
<code>fsck /dev/sdb1</code>	Check and repair a file system (on an unmounted partition)
<code>tune2fs, e2label</code>	Adjust parameters or set the label of an ext file system

Permanent mounts are listed in `/etc/fstab` so that they are attached automatically at boot (see Section 1.6).

Different OS Supported File Systems

Each operating system has its own native file systems, but it can often use others through drivers. Linux supports a very large number of file systems because of its Virtual File System layer.

Operating system	File system	Main features
Linux	ext2	Simple, fast, no journaling; still used for small /boot partitions
Linux	ext3	ext2 with a journal for faster recovery after a crash
Linux	ext4	Default on many distributions; journaling, extents, large files (up to 16 TiB with 4 KiB blocks) and volumes (up to 1 EiB)
Linux	XFS	High-performance journaling file system for very large files and volumes; default on RHEL 7 and later
Linux	Btrfs	Copy-on-write, snapshots, checksums, built-in volume management
Linux	JFS, ReiserFS	Other journaling file systems
Windows	FAT16, FAT32	Simple and widely compatible; no permissions; FAT32 allows files of at most about 4 GB
Windows	exFAT	For flash drives and SD cards; no 4 GB file limit
Windows	NTFS	Journaling, access control lists, encryption (EFS), compression, quotas
Windows	ReFS	Resilient file system for servers; protects against data corruption
macOS	HFS+ (Mac OS Extended)	Journalized file system of older Macs
macOS	APFS	Designed for SSDs; snapshots, encryption, space sharing
UNIX	UFS	Traditional UNIX File System (Solaris, BSD)
UNIX	ZFS	Storage pools, checksums, snapshots (Solaris, FreeBSD, available on Linux)
UNIX (AIX)	JFS2	Journalized file system of IBM AIX
Network	NFS	Sharing of files between UNIX and Linux computers
Network	SMB / CIFS	Windows file sharing; Samba provides it on Linux
Optical disc	ISO 9660, UDF	CD-ROM and DVD file systems
Virtual	proc, sysfs, tmpfs	Kept in memory; give access to kernel information and temporary storage

Comparison of FAT32, NTFS and ext4

Feature	FAT32	NTFS	ext4
Journaling	No	Yes	Yes
File permissions	None	ACLs (access control lists)	Owner/group/other and ACLs
Maximum file size	About 4 GB	Very large (terabytes)	16 TiB (4 KiB blocks)
Case-sensitive names	No	Optional	Yes
Typical use	Pen drives, cameras	Windows disks	Linux disks

Journaling

A **journaling** file system first writes the planned changes to a special log (the journal) and only then to the main file system. After a power failure the system replays the journal and recovers in seconds instead of running a long `fsck` check of the whole disk. ext3, ext4, XFS, NTFS and JFS are journaling file systems.

Poly Notes Hub

1.6 Examining the System Configuration Files

Most of the behaviour of a Linux system is controlled by plain text **configuration files**, which are kept mostly in the `/etc` directory. An administrator must know where they are and what they contain, because changing a setting usually means editing a file and restarting the service.

Important configuration files

File	Purpose
Users and groups	
<code>/etc/passwd</code>	User account information (name, UID, GID, home directory, shell)
<code>/etc/shadow</code>	Encrypted passwords and password ageing; readable only by root
<code>/etc/group</code>	Group names, GIDs and members
<code>/etc/gshadow</code>	Encrypted group passwords
<code>/etc/sudoers</code>	Rules for who may use sudo (edit with visudo)
Boot and system	
<code>/etc/inittab</code>	Default runlevel and init settings (SysV init)
<code>/etc/fstab</code>	File systems to mount at boot
<code>/etc/grub.conf</code> or <code>/boot/grub2/grub.cfg</code>	Boot loader configuration
<code>/etc/crontab</code> , <code>/etc/cron.d/</code>	Scheduled jobs
<code>/etc/rsyslog.conf</code> (syslog.conf)	What messages are logged and where
<code>/etc/sysctl.conf</code>	Kernel parameters
Network	
<code>/etc/hosts</code>	Local table of host names and IP addresses
<code>/etc/resolv.conf</code>	DNS servers and search domain
<code>/etc/hostname</code>	Name of the computer
<code>/etc/sysconfig/network</code>	Global network settings on Red Hat systems
<code>/etc/sysconfig/network-scripts/ifcfg-eth0</code>	Settings of the eth0 interface
<code>/etc/nsswitch.conf</code>	Order of sources (files, DNS, NIS) used to look up names
<code>/etc/services</code>	Names and port numbers of network services
Services and security	
<code>/etc/ssh/sshd_config</code>	SSH server settings
<code>/etc/httpd/conf/httpd.conf</code>	Apache web server settings
<code>/etc/samba/smb.conf</code>	Samba file-sharing settings
<code>/etc/exports</code>	Directories shared through NFS
<code>/etc/hosts.allow</code> , <code>/etc/hosts.deny</code>	TCP wrapper access control
<code>/etc/selinux/config</code>	SELinux mode (enforcing, permissive, disabled)
Shell and login	
<code>/etc/profile</code> , <code>/etc/bashrc</code>	Global shell start-up settings for all users

File	Purpose
/etc/shells	List of valid login shells
/etc/motd, /etc/issue	Message of the day and pre-login banner
/etc/yum.conf, /etc/yum.repos.d/	Package manager and repository settings

The /etc/passwd file

Each line describes one user and has seven fields separated by colons.

student:x:1001:1001:Student User:/home/student:/bin/bash
--

Field	Example	Meaning
1	student	Login name
2	x	Password placeholder; the real password is in /etc/shadow
3	1001	User ID (UID); 0 is root, small numbers are system accounts, ordinary users usually start at 500 or 1000
4	1001	Primary group ID (GID)
5	Student User	Comment or full name (GECOS field)
6	/home/student	Home directory
7	/bin/bash	Login shell (/sbin/nologin blocks login)

The /etc/shadow file

Each line has nine fields separated by colons. Example:
student:\$6\$salt\$hash:19000:0:99999:7:::

Field	Meaning
1	Login name
2	Encrypted (hashed) password; ! or * means the account is locked
3	Date of last password change (days since 1 January 1970)
4	Minimum days between password changes
5	Maximum days the password stays valid
6	Days of warning before the password expires
7	Days after expiry before the account is disabled
8	Account expiry date
9	Reserved

The /etc/fstab file

Each line has six fields that describe one file system to be mounted at boot.

#	device	mount point	type	options	dump	fsck-order
	/dev/sda2	/	ext4	defaults	1	1
	/dev/sda1	/boot	ext4	defaults	1	2
	/dev/sda3	swap	swap	defaults	0	0
	192.168.10.5:/data	/data	nfs	defaults	0	0

Field	Meaning
Device	Partition, UUID, label or network share
Mount point	Directory where it is attached
Type	File system type (ext4, xfs, nfs, swap, vfat ...)
Options	defaults, ro, rw, noexec, nosuid, user, noauto ...
Dump	1 = back up with the dump command, 0 = do not
fsck order	Order of checking at boot: 1 for root, 2 for others, 0 for none

Network configuration files

```
# /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.10.10
NETMASK=255.255.255.0
GATEWAY=192.168.10.1
ONBOOT=yes

# /etc/hosts
127.0.0.1        localhost
192.168.10.5     server1.example.com    server1

# /etc/resolv.conf
search example.com
nameserver 192.168.10.5
```

The /etc/crontab file

Scheduled jobs have six fields: minute, hour, day of month, month, day of week and command (the system crontab has an extra user field before the command). For example `30 2 * * 1 root /usr/local/bin/backup.sh` runs a backup at 2:30 a.m. every Monday.

Good practices when editing configuration files

- Take a backup copy before editing (`cp file file.bak`).
- Use the special editors `viw`, `vigr` and `visudo` for the `passwd`, `group` and `sudoers` files because they lock the file and check the syntax.
- Test the syntax before restarting a service (for example `apachectl configtest`).
- Restart or reload the service so that the change takes effect.
- Keep permissions strict: `/etc/shadow` must be readable only by root.
- Record every change in the administrator's log book.

Chapter Summary

- The system administrator installs, configures, secures, monitors and repairs systems and networks and works with the super user account.
- A server is installed by defining requirements, installing the OS, configuring and securing the service, testing it, and documenting the work.
- Network planning covers topology, addressing, services and security; the main topologies are bus, star, ring, mesh, tree and hybrid.
- Security is applied in layers (defence in depth) according to a written policy.

- Kickstart automates installation through the ks.cfg file created with the Kickstart Configurator; it can be delivered on a bootable CD-ROM.
- Boot order: POST, BIOS/UEFI, MBR, boot loader, kernel, init, runlevel services, login. Use shutdown for a safe stop.
- The Linux file system is one tree starting at /, organised by the FHS; many OS specific file systems (ext4, XFS, NTFS, FAT32, APFS, ZFS) exist.
- Configuration files in /etc (passwd, shadow, fstab, hosts, inittab and others) control the behaviour of the system.

Poly Notes Hub

Multiple Choice Questions (15)

Choose the correct option. The answer key is given after the questions.

- Q1.** Which file is used by the Kickstart method to automate a Linux installation?
- (a) ks.conf
 - (b) ks.cfg
 - (c) kick.cfg
 - (d) install.cfg
- Q2.** Which command starts the graphical Kickstart Configurator?
- (a) kickstart-gui
 - (b) ks-config
 - (c) system-config-kickstart
 - (d) anaconda-gui
- Q3.** In the boot sequence, the code stored in the MBR starts the:
- (a) boot loader
 - (b) kernel
 - (c) init process
 - (d) login shell
- Q4.** Which SysV runlevel is used to reboot the system?
- (a) 0
 - (b) 1
 - (c) 5
 - (d) 6
- Q5.** Which runlevel gives a multi-user system with a graphical login?
- (a) Runlevel 3
 - (b) Runlevel 5
 - (c) Runlevel 1
 - (d) Runlevel 6
- Q6.** Which directory holds the system-wide configuration files?
- (a) /var
 - (b) /usr
 - (c) /etc
 - (d) /opt
- Q7.** Which topology connects every node to a central hub or switch?
- (a) Star
 - (b) Bus
 - (c) Ring
 - (d) Mesh

- Q8.** How many links are needed for a full mesh of n nodes?
- (a) n
 - (b) $n - 1$
 - (c) $2n$
 - (d) $n(n - 1)/2$
- Q9.** Which file stores the encrypted passwords of Linux users?
- (a) `/etc/passwd`
 - (b) `/etc/shadow`
 - (c) `/etc/group`
 - (d) `/etc/hosts`
- Q10.** Which of the following is a journaling file system?
- (a) FAT32
 - (b) ISO 9660
 - (c) ext4
 - (d) ext2
- Q11.** What is the maximum size of a single file on a FAT32 file system?
- (a) About 4 GB
 - (b) About 2 GB
 - (c) About 16 GB
 - (d) About 1 TB
- Q12.** Which command halts the system immediately after warning users?
- (a) `init 6`
 - (b) `shutdown -r now`
 - (c) `init 3`
 - (d) `shutdown -h now`
- Q13.** Which directory contains the device files such as `/dev/sda`?
- (a) `/proc`
 - (b) `/dev`
 - (c) `/home`
 - (d) `/tmp`
- Q14.** How many colon-separated fields does each entry of `/etc/passwd` have?
- (a) 5
 - (b) 6
 - (c) 7
 - (d) 8
- Q15.** Which boot option tells the installer to read `ks.cfg` from the CD-ROM?
- (a) `ks=cdrom:/ks.cfg`
 - (b) `ks=floppy`
 - (c) `linux text`
 - (d) `linux rescue`

Answer Key

Q. No.	Answer	Reason
1	(b) ks.cfg	The answers to the installer's questions are stored in ks.cfg.
2	(c) system-config-kickstart	system-config-kickstart is the configurator tool.
3	(a) boot loader	The MBR boot code loads the first stage of the boot loader.
4	(d) 6	Runlevel 0 is halt and runlevel 6 is reboot.
5	(b) Runlevel 5	Runlevel 5 corresponds to graphical.target.
6	(c) /etc	/etc contains the configuration files.
7	(a) Star	In a star topology all nodes connect to a central device.
8	(d) $n(n-1)/2$	Each pair of nodes needs one link: $n(n-1)/2$.
9	(b) /etc/shadow	/etc/shadow holds the hashed passwords and is readable only by root.
10	(c) ext4	ext4 keeps a journal; the others do not.
11	(a) About 4 GB	FAT32 limits a file to roughly 4 GB.
12	(d) shutdown -h now	shutdown -h now halts; init 6 and shutdown -r reboot.
13	(b) /dev	Device files are kept in /dev.
14	(c) 7	Name, password, UID, GID, comment, home directory and shell: seven fields.
15	(a) ks=cdrom:/ks.cfg	ks=cdrom:/ks.cfg points to the file on the CD-ROM.

Broad Questions (10)

Answer each question in detail with diagrams, tables and examples wherever needed.

1. What are the duties of a system administrator? Explain the differences between administering a Linux system and administering other operating systems.
2. Describe the steps of installing and configuring a server. Show the installation of a web server on Linux with the commands used.
3. What is network planning? Describe the bus, star, ring, mesh, tree and hybrid topologies with neat diagrams, giving advantages and disadvantages of each.
4. Explain how security is planned and implemented in a network. Describe the common threats and the countermeasures against them.
5. What is Kickstart installation? Explain how the Kickstart Configurator is installed and the steps of creating a ks.cfg file. Write a sample ks.cfg.
6. Explain the Boot Loader Option, Partition, Network Configuration, Authentication and Firewall Configuration screens of the Kickstart Configurator.
7. How is a bootable CD-ROM for Kickstart installation created? Write the commands and the options used at the boot prompt to locate ks.cfg.
8. Explain the Linux boot process with a neat diagram. Describe the roles of the boot loader and the kernel, and explain runlevels and the shutdown procedure.
9. Explain the structure of the Linux file system (FHS) with a diagram. Compare the file systems supported by Linux, Windows, macOS and UNIX.
10. Describe important system configuration files such as /etc/passwd, /etc/shadow, /etc/fstab, /etc/hosts and /etc/inittab, and explain the fields of /etc/passwd.

End of Chapter 1