

Network Management and Administration

Chapter 6: System Administration

Study Notes for Engineering Students

Topics Covered
6.1 Keeping Your System Updated with up2date and Red Hat Network 6.2 Updating and Customizing the Kernel 6.3 Configuring the System at the Command Line 6.4 Administering Users and Groups Also Included: diagrams, tables, command cheat-sheet, 15 MCQs with answer key and 10 broad questions

Contents

Chapter Overview	4
Learning Objectives	4
6.1 Keeping Your System Updated with up2date and Red Hat Network	5
6.1.1 Why Systems Must Be Updated	5
6.1.2 Understanding Errata	5
6.1.3 Red Hat Network (RHN)	5
6.1.4 Registering the System	6
6.1.5 Using up2date	6
6.1.6 Configuring up2date	7
6.1.7 Updating from the RHN Web Site	8
6.1.8 RPM – The Package Manager Behind the Updates	8
6.1.9 Best Practices and Troubleshooting	8
6.2 Updating and Customizing the Kernel	10
6.2.1 The Role of the Kernel	10
6.2.2 Kernel Architecture and Versions	10
6.2.3 Important Kernel-Related Files	11
6.2.4 Why Update or Customize the Kernel?	11
6.2.5 Method 1: Installing a Pre-built Kernel RPM	11
6.2.6 Method 2: Compiling a Custom Kernel from Source	12
6.2.7 Configuring the GRUB Boot Loader	13
6.2.8 Working with Kernel Modules	14
6.2.9 Tuning the Running Kernel with sysctl	14
6.2.10 Safety Precautions and Troubleshooting	14
6.3 Configuring the System at the Command Line	16
6.3.1 Why Use the Command Line?	16
6.3.2 Where Configuration Is Stored	16
6.3.3 Editing Files with vi and nano	16
6.3.4 Configuration Tools: GUI and Text-Mode Equivalents	17
6.3.5 The Boot Process and Runlevels	17
6.3.6 Managing Services	18
6.3.7 Configuring the Network	18
6.3.8 Date and Time	19
6.3.9 File Systems and Mounting	19

6.3.10 Scheduling Jobs with cron and at	19
6.3.11 Process Management	20
6.3.12 System Logs	20
6.3.13 Remote Administration with SSH	20
6.4 Administering Users and Groups	21
6.4.1 Basic Concepts	21
6.4.2 Account Information Files	21
6.4.3 Creating User Accounts	22
6.4.4 Modifying and Deleting Accounts	23
6.4.5 Password Management and Ageing	23
6.4.6 Group Administration	23
6.4.7 File Ownership and Permissions	24
6.4.8 Delegating Administrative Privileges	24
6.4.9 Monitoring User Activity	25
6.4.10 Graphical Tool: User Manager	25
6.4.11 Automating Account Creation	25
6.4.12 Good Practices	25
Quick Revision Tables	26
Practice Questions	27
Part A: Multiple Choice Questions (15)	27
Answer Key for MCQs	29
Part B: Broad Questions (10)	29

Chapter Overview

System administration is the work of installing, configuring, securing, monitoring and maintaining a computer system so that it keeps running reliably for its users. On a Linux or UNIX-like machine this work is carried out by the **superuser** (the root account), who has unrestricted control over every file, process and device. In a networked environment an administrator may look after tens or hundreds of machines, so efficient, repeatable and well-documented procedures are essential.

This chapter covers four core administrative skills as they are practised on **Red Hat** based systems (Red Hat Linux, Fedora Core and Red Hat Enterprise Linux). Each skill answers a basic question: how do I keep software current, how do I control the heart of the operating system, how do I configure the machine without a mouse, and how do I control who can use it?

Table 6.1: Summary of the chapter

Section	Core question	Key tools and files
6.1	How do I keep installed software patched and up to date?	up2date, rhn_register, rpm, RHN web site
6.2	How do I upgrade or rebuild the kernel?	rpm, make, GRUB, modprobe, sysctl
6.3	How do I configure the system from a text terminal?	vi, chkconfig, service, ifconfig, cron, files in /etc
6.4	How do I manage user accounts and groups?	useradd, usermod, passwd, chage, groupadd, /etc/passwd

Learning Objectives

After studying this chapter the student should be able to:

- Register a system with Red Hat Network and apply errata using up2date and rpm.
- Explain the structure of the Linux kernel and describe how to install a new kernel or compile a custom one.
- Use the command line to edit configuration files, manage services and runlevels, configure the network, and schedule jobs.
- Create, modify, lock and delete users and groups, and explain the files that store account information.
- Apply file permissions, password ageing and privilege-delegation techniques to secure a system.

Version note: These notes follow the tools of the Red Hat Linux / Fedora Core / RHEL 3-4 generation (up2date, redhat-config-*, GRUB legacy). Newer releases replaced up2date with yum and later dnf, and redhat-config-* with system-config-*. The concepts remain identical, and a table of modern equivalents is given at the end of the notes.

6.1 Keeping Your System Updated with up2date and Red Hat Network

6.1.1 Why Systems Must Be Updated

No software is perfect. After a distribution is released, programmers keep finding security holes, crashes and compatibility problems. Vendors fix these defects and publish **updates** (also called **patches** or **errata**). A system that is never updated gradually becomes vulnerable, because attackers quickly learn how to exploit published flaws. Regular updating is therefore one of the most important duties of an administrator.

- **Security fixes:** close vulnerabilities in services such as OpenSSH, Apache, Sendmail or the kernel itself.
- **Bug fixes:** correct faults that cause crashes, data loss or wrong behaviour.
- **Enhancements:** add support for new hardware or improve performance and features.
- **Compatibility and compliance:** keep the system consistent with other machines and company policy.

6.1.2 Understanding Errata

Red Hat publishes each fix as an **erratum** (plural: **errata**). An erratum contains the corrected RPM packages together with a short description of the problem, the packages affected and the severity. Errata are identified by a code of the form RHSA-2005:123, where the prefix gives the type, the first number is the year and the second is a serial number.

Table 6.2: Types of Red Hat errata

Type	Prefix	Purpose
Security Advisory	RHSA	Fixes a security vulnerability; should be applied as soon as possible.
Bug Fix Advisory	RHBA	Fixes non-security defects in one or more packages.
Product Enhancement Advisory	RHEA	Adds new features or hardware support to an existing package.

6.1.3 Red Hat Network (RHN)

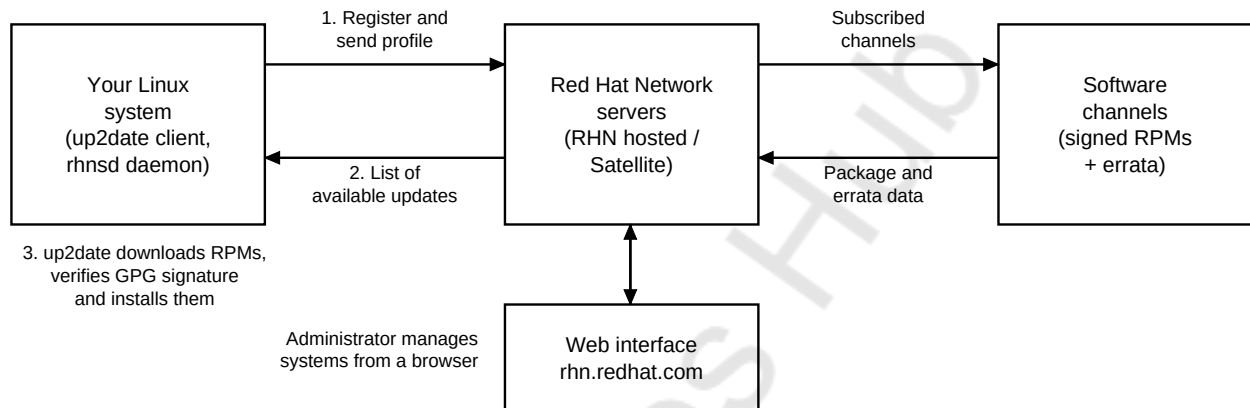
Red Hat Network is a subscription-based, Internet-delivered system-management service operated by Red Hat. A machine that is registered with RHN stores a **system profile** (hardware list and installed package list) on the Red Hat servers. RHN compares this profile with the latest errata and tells the administrator which updates apply. Because the comparison is done on the server, the administrator does not have to browse mailing lists or web sites to discover what is needed.

Main features

- Automatic notification when errata relevant to a registered system are released.
- Download of cryptographically signed packages from Red Hat channels.
- Web console for viewing and grouping many systems and scheduling updates remotely.
- Different **entitlement** levels (for example Update, Management and Provisioning), which determine the features available to a system.
- Optional on-site servers (**RHN Satellite** and **RHN Proxy**) that cache packages inside a company network and reduce bandwidth use.

Table 6.3: Components of Red Hat Network

Component	Role
up2date	Client program (GUI and command line) that retrieves and installs updates.
rhnc_register	Registers the system and uploads its first profile.
rhnsd	Daemon that wakes up periodically (default every two hours) and runs rhnc_check to see whether RHN has scheduled any actions.
Alert Notification Tool	Panel icon that changes colour when updates are available.
RHN web interface	Browser-based console at rhn.redhat.com for managing systems and errata.
Satellite / Proxy	Optional servers placed inside an organisation for local package distribution.

*Fig. 6.1: Interaction between a client system and Red Hat Network*

6.1.4 Registering the System

Before updates can be downloaded the system must be registered. Registration is done once, as root.

1. Log in as root (or use `su -`).
2. Start the registration tool with `rhnc_register` (use `rhnc_register --nox` on a text-only terminal).
3. Enter an existing RHN user name and password, or create a new account.
4. Give the system a profile name and choose whether to send the hardware and package lists.
5. Select the subscription/entitlement for the system.
6. Import Red Hat's public GPG key so that package signatures can be verified.

```
# rpm --import /usr/share/rhn/RPM-GPG-KEY
# rhnc_register
# up2date --configure          (optional: set proxy, skip lists, etc.)
```

A successful registration creates the file `/etc/sysconfig/rhn/systemid`, which uniquely identifies the machine to the RHN servers. Without it the system is treated as unregistered.

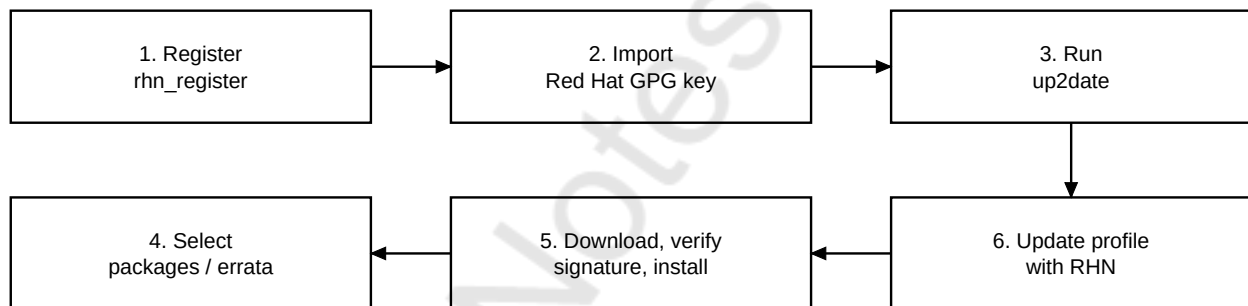
6.1.5 Using up2date

`up2date` is the client that talks to RHN. In a graphical session it presents a wizard that lists the available updates, lets the administrator select packages, downloads them, checks their signatures and installs them. On servers without X Windows, the same program is run with `--nox` or as `up2date-nox`, or simply from the command line with options.

Table 6.4: Commonly used up2date options

Command / option	Meaning
up2date	Start the graphical interface (or text mode if no X display).
up2date -l	List the updates available for this system.
up2date -u	Update every package that has a newer version available.
up2date package	Download and install or update the named package.
up2date -d package	Download only; do not install.
up2date -i package	Install after downloading (the default behaviour).
up2date -p	Update the package list stored in the RHN profile.
up2date -f	Force the update even when the package is on a skip list.
up2date --showall	List all packages available for the system, installed or not.
up2date --nox	Run in text mode without the X Window System.
up2date --configure	Open the configuration dialog for up2date.

# up2date -l	list pending updates
# up2date -u	install all updates
# up2date --nox bash openssh	update two packages in text mode
# up2date -d kernel	only download the kernel RPM

*Fig. 6.2: Typical workflow for updating a system with up2date*

6.1.6 Configuring up2date

The behaviour of up2date is controlled by the text file `/etc/sysconfig/rhn/up2date`, which can be edited by hand or through `up2date --configure`. Each line has the form `parameter=value`.

Table 6.5: Important parameters in `/etc/sysconfig/rhn/up2date`

Parameter	Purpose
serverURL	Secure URL of the RHN server (an https XML-RPC address).
noSSLServerURL	Fallback URL that does not use SSL.
enableProxy, httpProxy	Use an HTTP proxy and give its host:port when the machine is behind a firewall.
pkgSkipList	Packages never updated automatically. The default is <code>kernel*</code> ;
fileSkipList	Packages that are skipped if they contain files matching the given patterns.
useGPG	When 1, signatures of all packages are verified before installation.
keepAfterInstall	When 1, downloaded RPMs are kept after they are installed.
storageDir	Directory where RPMs are stored while downloading (default <code>/var/spool/up2date</code>).
systemIdPath	Location of the system identification file.
debug	Enables more detailed logging.

Why kernel*?: The kernel is placed on the default skip list because a bad kernel can stop the machine from booting. The administrator must deliberately choose to install a new kernel (see Section 6.2).

6.1.7 Updating from the RHN Web Site

Besides using `up2date` directly on the machine, an administrator can log in to the RHN web interface from any browser. This is especially useful for a group of servers.

- Browse the list of errata and read the advisory text and affected packages.
- Select one or more systems and **schedule** the errata to be applied.
- At the next check-in by `rhnsd` the client performs the scheduled action.
- Review the history of completed and failed actions for each system.

6.1.8 RPM – The Package Manager Behind the Updates

Almost all Red Hat software is distributed as **RPM** (Red Hat Package Manager) files. An RPM bundles program files, installation scripts, a list of dependencies and a digital signature. `up2date` is essentially a smart front end that downloads RPMs and calls `rpm` to install them. Knowing the `rpm` command is essential for troubleshooting.

Table 6.6: Frequently used rpm commands

Command	Function
<code>rpm -ivh pkg.rpm</code>	Install a new package (verbose, with hash marks).
<code>rpm -Uvh pkg.rpm</code>	Upgrade a package, or install it if absent; the old version is removed.
<code>rpm -Fvh pkg.rpm</code>	Freshen: upgrade only if an older version is already installed.
<code>rpm -e pkg</code>	Erase (uninstall) a package.
<code>rpm -qa</code>	Query: list all installed packages.
<code>rpm -qi pkg</code>	Show information (version, vendor, description) about an installed package.
<code>rpm -ql pkg</code>	List the files that belong to a package.
<code>rpm -qf /path/file</code>	Find which package owns a file.
<code>rpm -V pkg</code>	Verify installed files against the RPM database.
<code>rpm -K pkg.rpm</code>	Check the signature and digest of an RPM file.
<code>rpm --import keyfile</code>	Import a GPG public key.

6.1.9 Best Practices and Troubleshooting

- Take a backup of important data and configuration files before large updates.
- Test updates on a non-production machine first when possible.
- Read the errata description; some updates need a service restart or reboot.
- Apply security advisories quickly, and schedule routine updates in off-peak hours.
- Check the log file `/var/log/up2date` after each run.
- Never disable GPG checking; unsigned packages may be tampered with.

Table 6.7: Common up2date problems

Problem	Likely cause	Remedy
Cannot connect to RHN	Firewall, proxy or DNS failure	Check network with <code>ping</code> ; set <code>enableProxy</code> and <code>httpProxy</code> .
Signature verification failed	Red Hat public key not imported	Run <code>rpm --import /usr/share/rhn/RPM-GPG-KEY</code> .
System not registered	Missing <code>systemid</code> file	Run <code>rhnc_register</code> again.
Package not updated	Package matches <code>pkgSkipList</code>	Edit the skip list or use the force option.
Dependency error	A required package is missing or conflicting	Let <code>up2date</code> resolve dependencies; check with <code>rpm -qa</code> .

6.2 Updating and Customizing the Kernel

6.2.1 The Role of the Kernel

The **kernel** is the core of the operating system. It is the first major program loaded at boot and it stays in memory until shutdown. It acts as the intermediary between application programs and hardware, and it provides the basic services on which everything else depends.

- **Process management:** creating, scheduling and terminating processes.
- **Memory management:** allocating RAM, virtual memory and swapping.
- **File systems:** presenting disks, partitions and network shares as directories and files.
- **Networking:** implementing protocols such as TCP/IP.
- **Device drivers:** controlling disks, keyboards, display adapters and network cards.

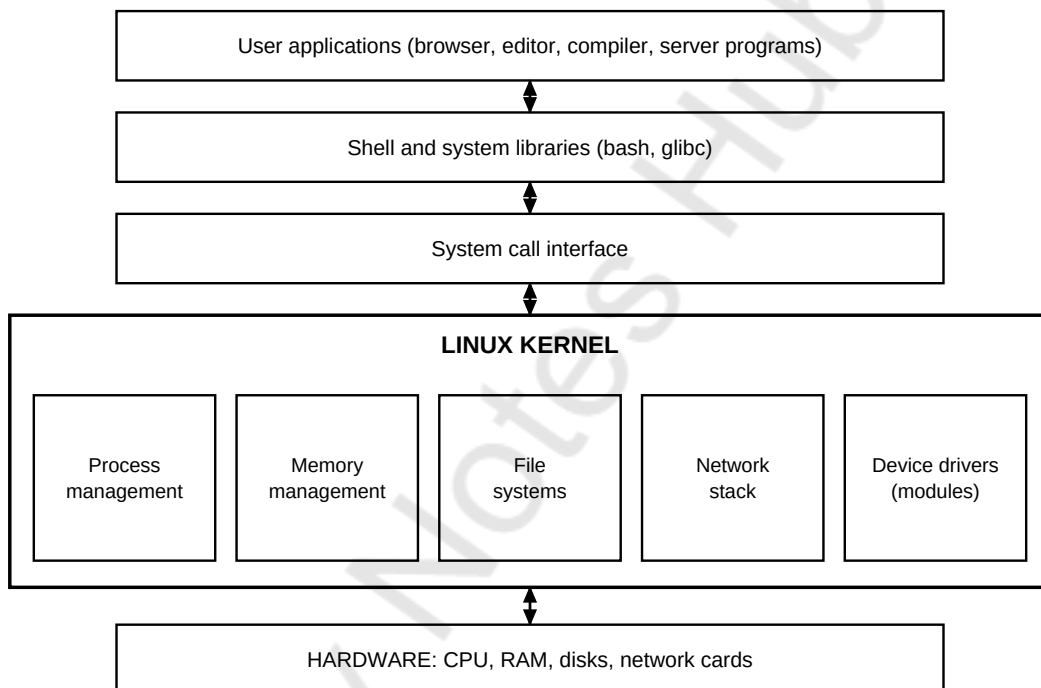


Fig. 6.3: Position of the kernel between applications and hardware

6.2.2 Kernel Architecture and Versions

Linux uses a **monolithic** kernel: the main services run together in one address space, which makes it fast. To avoid wasting memory and to avoid recompiling for every device, rarely used drivers are built as **loadable kernel modules** that can be inserted and removed while the system is running.

Table 6.8: Monolithic kernel versus microkernel

Feature	Monolithic (Linux)	Microkernel
Services run in	Kernel space together	User space as separate servers
Speed	Faster (fewer context switches)	Slower due to message passing
Size and complexity	Large single image plus modules	Small kernel, many servers
Fault isolation	A bad driver can crash the system	Better isolation

A kernel release is written like 2.6.9-34.EL. The command `uname -r` displays the version of the running kernel.

Table 6.9: Anatomy of a kernel version string

Part	Example	Meaning
Major version	2	Changes only with a fundamental redesign.
Minor version	6	Series number. In the 2.x series even numbers were stable and odd numbers were development.
Patch level	9	Incremented with each new kernel release in the series.
Vendor build	34	Red Hat's own release number including backported fixes.
Suffix	EL	Identifies Red Hat Enterprise Linux; may also indicate options such as <code>smp</code> .

6.2.3 Important Kernel-Related Files

Table 6.10: Files and directories related to the kernel

File or directory	Description
<code>/boot/vmlinuz-<version></code>	The compressed, bootable kernel image.
<code>/boot/initrd-<version>.img</code>	Initial RAM disk containing drivers needed to mount the root file system.
<code>/boot/System.map-<version></code>	Symbol table that maps kernel function names to memory addresses.
<code>/boot/config-<version></code>	The configuration options used to build that kernel.
<code>/lib/modules/<version>/</code>	Loadable modules for that kernel version.
<code>/usr/src/linux-<version>/</code>	Kernel source tree used for building a custom kernel.
<code>/boot/grub/grub.conf</code>	Boot-loader configuration listing the kernels that can be started.
<code>/proc/version</code> , <code>/proc/cmdline</code>	Run-time information about the running kernel and its boot options.

6.2.4 Why Update or Customize the Kernel?

- Install **security fixes** for vulnerabilities in the kernel.
- Add support for **new hardware** (network cards, disk controllers, graphics devices).
- Enable features such as new file systems or SMP support.
- **Remove unneeded drivers** to make the kernel smaller and faster, which is useful for embedded systems.
- Tune the kernel for a special workload such as a database or a firewall.
- Learn how the operating system works (a common exercise in engineering labs).

There are two ways to obtain a new kernel: install a **pre-built kernel RPM** (simple and recommended) or **compile the kernel from source** (flexible but more work).

6.2.5 Method 1: Installing a Pre-built Kernel RPM

The kernel RPM is obtained from RHN or from the distribution media. The important rule is to use `rpm -ivh` (install) instead of `rpm -Uvh` (upgrade). An upgrade removes the old kernel, but an install keeps the old kernel on disk so that the machine can still be booted if the new kernel fails. The package's installation script automatically builds the `initrd` image and adds a new entry to `grub.conf`.

```
# rpm -ivh kernel-2.6.9-34.EL.i686.rpm
# ls /boot                                confirm vmlinuz and initrd exist
# grep title /boot/grub/grub.conf
# reboot
# uname -r                                after reboot: check running version
```

If `up2date` is used to fetch the kernel, the `kernel*` entry in `pkgSkipList` has to be removed (or the `force` option used) because kernels are skipped by default.

6.2.6 Method 2: Compiling a Custom Kernel from Source

Prerequisites

- Kernel source (a tarball from kernel.org or the vendor's source RPM).
- Development tools: gcc, make, binutils, and ncurses-devel for menuconfig.
- Sufficient free disk space (several hundred MB) and root privileges.
- A backup of /boot and a working boot disk or rescue CD.

Steps

1. Unpack the source in /usr/src and enter the directory.
2. Run make mrproper to remove old configuration files and stale objects.
3. Configure the kernel with make menuconfig (or xconfig, oldconfig). The result is saved in the hidden file .config.
4. Build the kernel image and modules with make. (Kernel 2.4 requires the separate steps make dep, make bzImage, make modules.)
5. Install the modules with make modules_install (copies them to /lib/modules/<version>).
6. Install the kernel with make install, which copies the image, System.map and config to /boot, creates the initrd and updates grub.conf.
7. Reboot, select the new kernel in the GRUB menu and test it.

```
# cd /usr/src
# tar xjf linux-2.6.x.tar.bz2
# ln -s linux-2.6.x linux
# cd linux
# make mrproper
# make menuconfig
# make
# make modules_install
# make install
# reboot
```

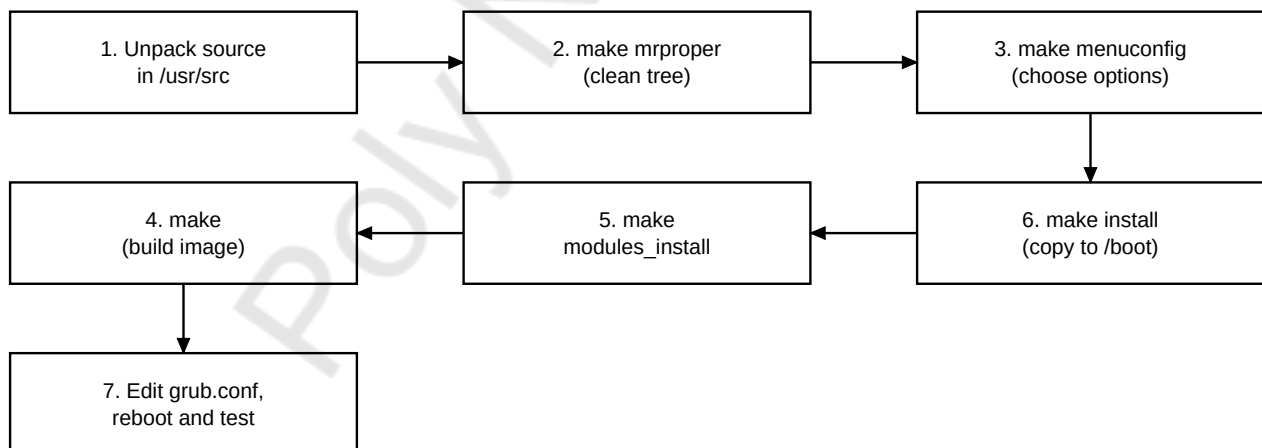


Fig. 6.4: Steps for compiling and installing a custom kernel

If make install is not used, the files can be installed manually:

```
# cp arch/i386/boot/bzImage /boot/vmlinuz-2.6.x-custom
# cp System.map /boot/System.map-2.6.x-custom
# cp .config /boot/config-2.6.x-custom
# mkinitrd /boot/initrd-2.6.x-custom.img 2.6.x-custom
(then add a new title entry to /boot/grub/grub.conf)
```

Tip: Edit the EXTRAVERSION line in the top-level Makefile (for example EXTRAVERSION = -custom). The new kernel and its modules then get a unique version string and do not overwrite the files of the existing kernel.

Table 6.11: Kernel configuration interfaces

Interface	Description
make config	Plain text question-and-answer session; tedious but works anywhere.
make menuconfig	Menu-driven interface using the ncurses library; suited to text terminals.
make xconfig	Graphical interface (Qt) available under X Windows.
make gconfig	Graphical interface using GTK.
make oldconfig	Reuses an existing .config and asks only about new options.
make defconfig	Creates a configuration with default options for the architecture.

Table 6.12: Meaning of options in kernel configuration

Choice	Symbol	Effect
Built-in	y	Code is compiled into the kernel image; always loaded.
Module	m	Code is compiled as a loadable module that is loaded only when needed.
Excluded	n	Feature is not built at all.

Table 6.13: Common make targets for the kernel

make target	Action
mrproper	Deep clean: deletes .config, object files and generated files.
clean	Removes most generated files but keeps .config.
bzImage	Builds the compressed bootable kernel image.
modules	Builds all options chosen as m.
modules_install	Installs modules under /lib/modules/<version>.
install	Installs kernel, map and config in /boot and updates the boot loader.

6.2.7 Configuring the GRUB Boot Loader

GRUB (GRand Unified Bootloader) is the default boot loader on Red Hat systems. It reads /boot/grub/grub.conf (also reachable as /etc/grub.conf) and presents a menu of kernels. Each entry starts with title and tells GRUB where the kernel and the initial RAM disk are located.

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
title Red Hat Enterprise Linux (2.6.9-34.EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-34.EL ro root=LABEL=/ rhgb quiet
    initrd /initrd-2.6.9-34.EL.img
title Previous kernel (2.6.9-22.EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-22.EL ro root=LABEL=/
    initrd /initrd-2.6.9-22.EL.img
```

Table 6.14: Important GRUB directives

Directive	Meaning
default	Number of the entry booted automatically (counting from 0).
timeout	Seconds to wait before booting the default entry.
title	Name shown in the boot menu.
root (hd0,0)	Partition that holds the kernel: first disk, first partition.
kernel	Path of the kernel image followed by kernel parameters such as <code>ro</code> and <code>root=</code> .
initrd	Path of the initial RAM disk image.

Parameters can also be typed at the GRUB prompt for a single boot. Pressing `e` edits an entry, and adding single (or 1) boots into single-user mode, which is how administrators recover a forgotten root password or repair a system.

6.2.8 Working with Kernel Modules

Modules let the kernel gain features at run time. Each module file ends in `.ko` (kernel 2.6) or `.o` (kernel 2.4) and lives under `/lib/modules/<version>/`. The tool `modprobe` is preferred over `insmod` because it automatically loads any modules on which the requested one depends.

Table 6.15: Module management commands

Command	Purpose
<code>lsmod</code>	List the modules currently loaded, with size and usage count.
<code>modinfo name</code>	Show author, description and parameters of a module.
<code>modprobe name</code>	Load a module and its dependencies.
<code>modprobe -r name</code>	Remove a module and unused dependencies.
<code>insmod file.ko</code>	Insert one module file; does not resolve dependencies.
<code>rmmod name</code>	Remove a single module.
<code>depmod -a</code>	Rebuild the module dependency list (<code>modules.dep</code>).

```
# lsmod | grep e1000
# modprobe e1000
# modinfo e1000
# modprobe -r e1000
```

Module aliases and options are defined in `/etc/modprobe.conf` (kernel 2.6) or `/etc/modules.conf` (kernel 2.4). For example, the line `alias eth0 e1000` tells the system to load the `e1000` driver for the first Ethernet card.

6.2.9 Tuning the Running Kernel with sysctl

Many kernel parameters can be changed without rebooting by writing to files under `/proc/sys` or by using the `sysctl` command. Changes made this way are lost at reboot unless they are added to `/etc/sysctl.conf`.

```
# sysctl -a | less           show all parameters
# sysctl -w net.ipv4.ip_forward=1  change now
# echo 1 > /proc/sys/net/ipv4/ip_forward  same effect
# vi /etc/sysctl.conf         add: net.ipv4.ip_forward = 1
# sysctl -p                  reload the file
```

6.2.10 Safety Precautions and Troubleshooting

- Always keep at least one known-good kernel entry in `grub.conf`.

- Test the new kernel before removing the old one.
- Keep a copy of `/boot`, and record the hardware configuration.
- Do not delete `/lib/modules/<version>` of a kernel that is still in use.

Table 6.16: Kernel troubleshooting guide

Symptom	Probable cause	Solution
Kernel panic: unable to mount root fs	Missing <code>initrd</code> or missing disk driver	Boot the old kernel; rebuild <code>initrd</code> with <code>mkinitrd</code> .
Module not found when loading	<code>depmod</code> not run or modules not installed	Run <code>make modules_install</code> and <code>depmod -a</code> .
New kernel not shown in menu	No entry in <code>grub.conf</code>	Add the <code>title</code> section manually.
Network card not detected	Driver not compiled	Rebuild kernel with the driver as <code>m</code> and load with <code>modprobe</code> .

6.3 Configuring the System at the Command Line

6.3.1 Why Use the Command Line?

Graphical configuration tools are convenient, but professional administrators rely heavily on the shell. Servers are often managed remotely through `ssh`, where no graphical desktop exists. Commands can be recorded in scripts, repeated on many machines, and used over slow network links.

- Works over a low-bandwidth remote connection and on systems without a GUI.
- Allows automation with shell scripts and `cron` jobs.
- Gives access to every option, including those omitted from GUI tools.
- Uses less memory and CPU, which is important on servers.

6.3.2 Where Configuration Is Stored

UNIX stores configuration in plain text files, mostly inside `/etc`. Editing the right file and then restarting the corresponding service is the basis of command-line administration. Red Hat keeps many system settings in the subdirectory `/etc/sysconfig`.

Table 6.17: Important configuration files

File	Purpose
<code>/etc/passwd, /etc/shadow, /etc/group</code>	User and group databases.
<code>/etc/fstab</code>	File systems mounted at boot.
<code>/etc/inittab</code>	Default runlevel and init process table.
<code>/etc/hosts</code>	Static host name to IP address mapping.
<code>/etc/resolv.conf</code>	DNS servers and search domain.
<code>/etc/sysconfig/network</code>	Host name, default gateway and networking on/off.
<code>/etc/sysconfig/network-scripts/ifcfg-eth0</code>	Settings of the first Ethernet interface.
<code>/etc/crontab, /etc/cron.*</code>	Scheduled system jobs.
<code>/etc/syslog.conf</code>	Rules for system logging.
<code>/etc/ssh/sshd_config</code>	Configuration of the SSH server.
<code>/etc/exports</code>	Directories shared through NFS.

Caution: Before editing a file, make a copy (for example `cp /etc/fstab /etc/fstab.bak`). A single typing error in files such as `fstab` or `inittab` may prevent the system from booting.

6.3.3 Editing Files with `vi` and `nano`

`vi` is available on every UNIX system and is the editor administrators are expected to know. It has two main modes: **command mode** (keys are commands) and **insert mode** (keys are text). The easier editor `nano` shows its key shortcuts at the bottom of the screen.

Table 6.18: Basic vi commands

vi key	Action
i / a / o	Insert before cursor / append after cursor / open a new line below.
Esc	Return to command mode.
:w :q :wq :q!	Save / quit / save and quit / quit without saving.
dd yy p	Delete line / copy line / paste.
/text n	Search forward / repeat search.
:set nu	Show line numbers.
:%s/old/new/g	Replace all occurrences in the file.

6.3.4 Configuration Tools: GUI and Text-Mode Equivalents

Red Hat supplies a family of system-config-* programs (named redhat-config-* in older releases). Most can run either in X or have a console counterpart, and all of them simply read and write the same text files described above.

Table 6.19: GUI tools and their command-line equivalents

Task	Graphical tool	Text-mode / command-line
Network	system-config-network	netconfig, edit ifcfg-eth0, ifup, ifdown
Date and time	system-config-date	date, hwclock, ntpdate
Services	system-config-services	ntsysv, chkconfig, service
Firewall	system-config-securitylevel	lokkit, iptables
Users and groups	system-config-users	useradd, usermod, groupadd
Authentication	authconfig-gtk	authconfig
Keyboard / language	system-config-keyboard	edit /etc/sysconfig/keyboard, /etc/sysconfig/i18n

6.3.5 The Boot Process and Runlevels

Understanding the boot sequence helps in locating configuration problems. After the BIOS tests the hardware, it loads GRUB from the disk. GRUB loads the kernel and initrd. The kernel mounts the root file system and starts init, the first process (PID 1). init reads /etc/inittab to decide the **runlevel** and then runs the startup scripts of that level.

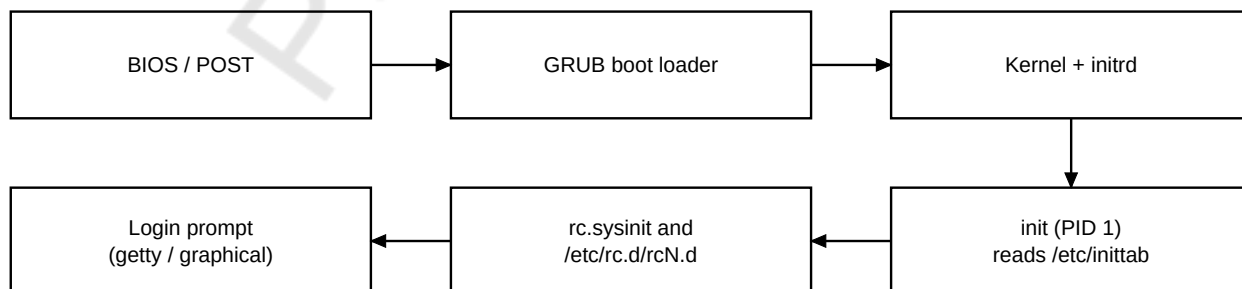
*Fig. 6.5: Linux boot sequence*

Table 6.20: Red Hat runlevels

Runlevel	State
0	Halt (shutdown).
1	Single-user mode, used for maintenance and recovery.
2	Multi-user without network file system (NFS).
3	Full multi-user, text mode login (typical for servers).
4	Unused / user-definable.
5	Full multi-user with graphical (X11) login.
6	Reboot.

```
# grep initdefault /etc/inittab      shows id:5:initdefault:
# runlevel                          previous and current level
# telinit 3                         switch to runlevel 3 now
# init 6                            reboot
```

6.3.6 Managing Services

A **service** (daemon) is a background program such as httpd or sshd. For each service a control script exists in /etc/rc.d/init.d/. Each runlevel directory /etc/rc.d/rcN.d/ contains symbolic links to these scripts. Links beginning with **S** (Start) are run in numeric order when the level is entered, and links beginning with **K** (Kill) are run to stop the service. For example, S85httpd starts Apache late in the boot, while K15httpd stops it early at shutdown.

```
# service httpd start | stop | restart | status
# chkconfig --list                all services and their runlevels
# chkconfig --list httpd
# chkconfig --level 35 httpd on    start at boot in levels 3 and 5
# chkconfig httpd off             disable in all levels
# chkconfig --add myservice       register a new script
# ntsysv --level 35               text-menu alternative
```

6.3.7 Configuring the Network

Each Ethernet interface has a file in /etc/sysconfig/network-scripts. A static IP configuration for eth0 looks like this:

```
# /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.1.10
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
ONBOOT=yes
```

For a DHCP client the line BOOTPROTO=dhcp is used and the address lines are omitted. The general host name and gateway are kept in /etc/sysconfig/network, name servers in /etc/resolv.conf (nameserver 192.168.1.1), and local name mappings in /etc/hosts. After editing, apply the changes with service network restart or ifdown eth0; ifup eth0.

Table 6.21: Network administration commands

Command	Use
ifconfig	Show or set interface address, mask and state.
ifup eth0 / ifdown eth0	Bring an interface up or down using its ifcfg file.
route -n	Display or modify the routing table.
netstat -tuln	List listening TCP/UDP ports.
ping host	Test reachability using ICMP echo.
traceroute host	Show the path taken by packets.
hostname	Show or set the host name.
dig / nslookup	Query DNS servers.

6.3.8 Date and Time

Correct time is vital for logs, security certificates and cron. The kernel maintains the system clock, while the hardware clock is a battery-powered chip. Time can be synchronised with an NTP server.

# date	show current time
# date -s "2026-10-22 14:30:00"	set system time
# hwclock --systohc	copy system time to hardware clock
# ntpdate pool.ntp.org	one-time synchronisation
# service ntpd start	continuous synchronisation

6.3.9 File Systems and Mounting

Before use, a disk partition must have a file system and must be **mounted** on a directory (mount point). Permanent mounts are listed in /etc/fstab, which has six fields per line.

# device	mount point	type	options	dump	fsck-order
/dev/sda2	/home	ext3	defaults	1	2

Table 6.22: Disk and file-system commands

Command	Function
mount /dev/sdb1 /mnt	Attach a file system.
umount /mnt	Detach a file system.
df -h	Free and used space per file system.
du -sh dir	Space used by a directory.
fdisk -l	List disk partitions.
mkfs -t ext3 /dev/sdb1	Create a file system.
fsck /dev/sdb1	Check and repair a (unmounted) file system.

6.3.10 Scheduling Jobs with cron and at

cron runs commands repeatedly at fixed times, whereas **at** runs a command once. Every user may have a personal crontab, edited with crontab -e. A crontab line has five time fields followed by the command.

Table 6.23: Fields of a crontab entry

Field	Minute	Hour	Day of month	Month	Day of week
Range	0-59	0-23	1-31	1-12	0-7 (0 and 7 = Sunday)

```

30 2 * * 1-5    /usr/local/bin/backup.sh      02:30 on weekdays
0 */6 * * *    /usr/sbin/ntpdate pool.ntp.org  every 6 hours
# crontab -l          list own jobs      # crontab -r  remove all
# at 10:30 tomorrow   then type command, finish with Ctrl-D
# atq                list pending jobs   # atrm 3      cancel job 3

```

6.3.11 Process Management

Table 6.24: Process control commands

Command	Purpose
ps aux	Snapshot of all running processes.
top	Live display of CPU and memory usage.
kill PID/kill -9 PID	Ask a process to end / force it to end.
killall name	Kill processes by name.
nice -n 10 cmd, renice	Start or change a process's priority.
jobs, bg, fg	Control background and foreground jobs in the shell.

6.3.12 System Logs

Daemons record events through the syslog service, configured in /etc/syslog.conf. Log files are kept in /var/log and are rotated by logrotate so that they do not fill the disk. Reading logs is the first step in diagnosing a fault.

Table 6.25: Common log files

Log file	Contents
/var/log/messages	General system messages.
/var/log/secure	Authentication and security events (logins, su, sshd).
/var/log/cron	Records of cron job execution.
/var/log/maillog	Mail server activity.
/var/log/boot.log	Services started or failed during boot.
/var/log/up2date	Package update history.
dmesg	Command showing kernel ring-buffer messages from boot.

```

# tail -f /var/log/messages      watch new messages live
# grep "Failed password" /var/log/secure

```

6.3.13 Remote Administration with SSH

Secure Shell (SSH) gives an encrypted login to a remote machine and replaces insecure tools such as telnet. The server daemon sshd is configured in /etc/ssh/sshd_config. A common hardening step is setting PermitRootLogin no, so that administrators log in as normal users and then use su or sudo.

```

$ ssh student@192.168.1.10      remote login
$ scp report.txt student@192.168.1.10:/tmp/  secure copy

```

6.4 Administering Users and Groups

6.4.1 Basic Concepts

Linux is a **multi-user** system: many people may use the machine at once, and each is given an **account** with a private home directory and a limited set of rights. Every account has a numeric **User ID (UID)**, and each group has a numeric **Group ID (GID)**. The kernel works with these numbers; names are for human convenience. A **group** is a collection of users that share access to files or resources.

Table 6.26: Types of user accounts

Account type	UID range (Red Hat)	Description
Superuser (root)	0	Unrestricted access to the whole system.
System accounts	1 to 499	Used by services such as <code>apache</code> , <code>mail</code> and <code>daemon</code> ; usually have no login shell.
Regular users	500 and above	Ordinary people; limited to their own files and permitted resources.

Each user has one **primary group** (the GID in `/etc/passwd`) and may belong to several **secondary groups**. Red Hat uses the **User Private Group** scheme: when a user is created, a new group with the same name is created and made the primary group. This lets users share directories safely without risking accidental exposure of their files to everyone. (Newer distributions start regular UIDs at 1000.)

6.4.2 Account Information Files

User information is stored in plain text files, each line holding one record with fields separated by colons.

student	:	x	:	500	:	500	:	Student User	:	/home/student	:	/bin/bash	:
1		2		3		4		5		6		7	
Username		Password		UID		GID		Comment (GECOS)		Home directory		Login shell	

Fig. 6.6: Fields of one line in `/etc/passwd`

Table 6.27: Fields in `/etc/passwd`

No.	Field	Description
1	Username	Login name, unique on the system.
2	Password	Usually <code>x</code> , meaning the encrypted password is stored in <code>/etc/shadow</code> .
3	UID	Numeric user ID.
4	GID	Primary group ID.
5	Comment (GECOS)	Full name or other descriptive text.
6	Home directory	Directory the user starts in.
7	Login shell	Program run at login; <code>/sbin/nologin</code> prevents interactive login.

Early UNIX stored the password hash in `/etc/passwd`, which every user can read. This made password cracking easy. **Shadow passwords** move the hash to `/etc/shadow`, readable only by root. Besides the hash, this file also holds the password-ageing information.

```
student:$1$xyz$Qr8s...:13500:0:90:7:14::
```

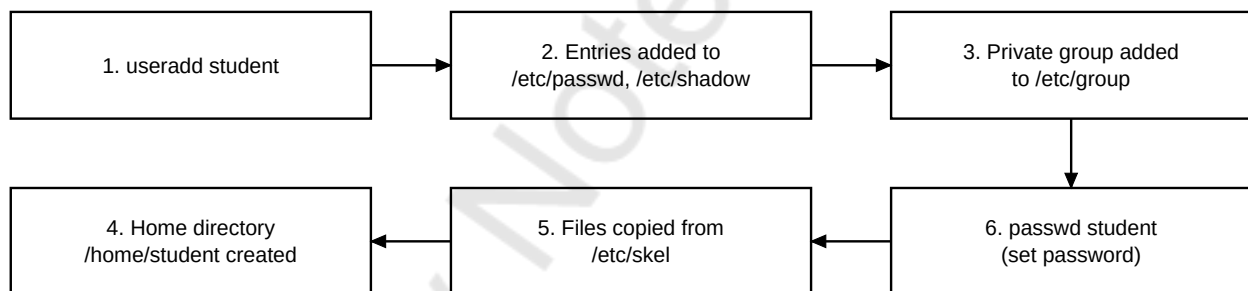
Table 6.28: Fields in /etc/shadow

No.	Field	Description
1	Username	Matches the entry in /etc/passwd.
2	Encrypted password	Hash of the password. A leading ! or * means the account is locked or has no password.
3	Last change	Days since 1 January 1970 when the password was last changed.
4	Minimum age	Days before the password may be changed again.
5	Maximum age	Days after which the password must be changed.
6	Warning period	Days before expiry that the user is warned.
7	Inactivity period	Days after expiry before the account is disabled.
8	Expiration date	Date (in days since 1970) on which the account is disabled.
9	Reserved	Reserved for future use.

Group data is in /etc/group, which has four fields: group name, password placeholder, GID and a comma-separated member list. An example line is `developers:x:600:alice,bob`. The file /etc/gshadow holds group passwords and group administrators.

6.4.3 Creating User Accounts

The `useradd` command adds a user. It reads defaults from /etc/default/useradd and /etc/login.defs, and copies the template files from /etc/skel (such as `.bash_profile` and `.bashrc`) into the new home directory.

*Fig. 6.7: What happens when a user is created***Table 6.29: Options of useradd**

Option	Meaning
-c "comment"	Full name or description (GECOS field).
-d dir	Home directory path.
-m	Create the home directory if it does not exist.
-g group	Primary group (name or GID).
-G g1,g2	Secondary groups.
-s shell	Login shell, e.g. /bin/bash.
-u UID	Specify the UID manually.
-e date	Account expiry date (YYYY-MM-DD).
-f days	Days after password expiry before the account is disabled.

```
# useradd -m -c "Student One" -s /bin/bash student1
# passwd student1
# useradd -u 1200 -g developers -G wheel -m -e 2027-12-31 dev1
```

6.4.4 Modifying and Deleting Accounts

Table 6.30: Modifying and removing users

Command	Function
<code>usermod -l new old</code>	Rename the login name.
<code>usermod -d /newhome -m user</code>	Change and move the home directory.
<code>usermod -s /sbin/nologin user</code>	Change login shell (here, disable login).
<code>usermod -G g1,g2 user</code>	Set secondary groups (replaces the existing list).
<code>usermod -aG group user</code>	Add to an additional group, keeping the old ones.
<code>usermod -L user / -U user</code>	Lock / unlock the account.
<code>userdel user</code>	Delete the account but keep the home directory.
<code>userdel -r user</code>	Delete the account and its home directory and mail spool.

6.4.5 Password Management and Ageing

Only the superuser can set another user's password, while ordinary users can change their own with `passwd`. Strong passwords are the first line of defence. A good password is at least eight characters, mixes letters, digits and symbols, and is not a dictionary word, name or birth date. Password ageing forces periodic changes.

Table 6.31: Password-related commands

Command	Function
<code>passwd user</code>	Set or change a password.
<code>passwd -l user / -u user</code>	Lock / unlock the password.
<code>passwd -d user</code>	Delete the password (account has no password).
<code>passwd -S user</code>	Display status of the password.
<code>passwd -e user</code>	Expire the password; user must change it at next login.
<code>chage -l user</code>	List ageing information.
<code>chage -M 90 user</code>	Maximum password age of 90 days.
<code>chage -m 7 user</code>	Minimum age of 7 days.
<code>chage -W 14 user</code>	Warn 14 days before expiry.
<code>chage -I 30 user</code>	Disable account 30 days after expiry.
<code>chage -E date user</code>	Set the account expiry date.
<code>chage -d 0 user</code>	Force a password change at the next login.

6.4.6 Group Administration

Table 6.32: Commands for groups

Command	Function
<code>groupadd [-g GID] name</code>	Create a group.
<code>groupmod -n new old</code>	Rename a group (or change its GID with <code>-g</code>).
<code>groupdel name</code>	Delete a group (not allowed if it is a user's primary group).
<code>gpasswd -a user group</code>	Add a user to a group.
<code>gpasswd -d user group</code>	Remove a user from a group.
<code>id user, groups user</code>	Show UID, GID and group memberships.
<code>newgrp group</code>	Start a shell with a different primary group.

```
# groupadd -g 600 developers
# gpasswd -a alice developers
# gpasswd -a bob developers
# id alice
uid=501(alice) gid=501(alice) groups=501(alice),600(developers)
```

6.4.7 File Ownership and Permissions

Access control relies on ownership and permission bits. Every file has an **owner**, a **group** and three sets of permissions: for the owner (u), the group (g) and others (o). Each set contains read (r), write (w) and execute (x). In octal notation r = 4, w = 2 and x = 1, and the values are added.

Table 6.33: Permission values

Octal	Symbolic	Meaning
7	rwX	Read, write and execute.
6	rw-	Read and write.
5	r-X	Read and execute.
4	r--	Read only.
0	---	No access.

```
# ls -l report.txt
-rw-r----- 1 alice developers 2048 Oct 22 10:00 report.txt
# chmod 640 report.txt          owner rw, group r, others none
# chmod u+x script.sh          add execute for owner
# chown alice:developers report.txt
# chmod 2775 /srv/project      shared directory with setgid bit
```

Table 6.34: Special permission bits

Special bit	Octal	Effect
SUID (set user ID)	4000	Executable runs with the privileges of the file's owner (e.g. passwd).
SGID (set group ID)	2000	Executable runs with the group's privileges; on a directory, new files inherit the directory's group.
Sticky bit	1000	On a directory, only the file's owner may delete or rename it (e.g. /tmp).

6.4.8 Delegating Administrative Privileges

Logging in permanently as root is dangerous because every mistake has full effect. Better practice is to work as a normal user and gain privileges only when necessary.

- su switches to another user. su - gives a full login environment of root.
- sudo command runs one command as root (or another user), after checking the file /etc/sudoers. The file must be edited with visudo, which checks syntax.
- Members of the group wheel may be allowed to use su or sudo for administration.

```
# visudo
student1  ALL=(ALL)  /sbin/service, /sbin/ifup, /sbin/ifdown
%wheel    ALL=(ALL)  ALL
```

6.4.9 Monitoring User Activity

Table 6.35: Commands for watching users

Command / file	Information shown
who, w	Users currently logged in and what they are doing.
last	History of logins from /var/log/wtmp.
lastb	Failed login attempts.
lastlog	Most recent login of every user.
/var/log/secure	Detailed authentication messages.

6.4.10 Graphical Tool: User Manager

The program `system-config-users` (`redhat-config-users` in older releases) offers a graphical way to add, edit and delete users and groups, and to set shell, home directory, password ageing and group membership. It does not hide anything different: it updates the same files, `/etc/passwd`, `/etc/shadow` and `/etc/group`, using the same commands.

6.4.11 Automating Account Creation

When many accounts are required (for example for a lab with 60 students) a shell script is used. The `chpasswd` command reads `user:password` pairs, and `chage -d 0` forces each user to select a new password at first login.

```
#!/bin/bash
for u in alice bob carol
do
    useradd -m -s /bin/bash $u
    echo "$u:Temp#1234" | chpasswd
    chage -d 0 $u
done
```

6.4.12 Good Practices

- Create one account per person; never share accounts.
- Remove or lock accounts of users who leave: `usermod -L` and set the shell to `/sbin/nologin`; archive the home directory before deleting.
- Use groups to give access to shared resources instead of changing permissions to 777.
- Enforce password ageing and complexity rules.
- Check `/etc/passwd` for entries with UID 0 other than root, and for empty password fields (the tools `pwck` and `grpck` check file consistency).
- Edit account files only with the proper commands or `vipw` / `vigr`, which lock the file during editing.

Quick Revision Tables

Table 6.36: Task-to-command cheat sheet

Task	Red Hat command
Register with RHN	<code>rhn_register</code>
List / apply all updates	<code>up2date -l/up2date -u</code>
Install a new kernel keeping the old one	<code>rpm -ivh kernel-*.rpm</code>
Clean kernel source tree	<code>make mrproper</code>
Build and install modules	<code>make; make modules_install</code>
Load / unload a module	<code>modprobe name/modprobe -r name</code>
Show running kernel version	<code>uname -r</code>
Enable service at boot	<code>chkconfig --level 35 httpd on</code>
Restart a service	<code>service httpd restart</code>
Change runlevel	<code>telinit 3</code>
Edit schedule	<code>crontab -e</code>
Create user with home dir	<code>useradd -m user</code>
Set password	<code>passwd user</code>
Force password change at login	<code>chage -d 0 user</code>
Add user to group	<code>gpasswd -a user group</code>
Delete user and home directory	<code>userdel -r user</code>

Table 6.37: Older tools and their modern equivalents

Older Red Hat tool	Modern equivalent
<code>up2date</code>	<code>yum</code> (RHEL 5/6, with <code>yum-rhn-plugin</code>), <code>dnf</code> (RHEL 8+, Fedora)
<code>rhn_register</code>	<code>rhnreg_ks</code> , later <code>subscription-manager</code>
GRUB legacy (<code>grub.conf</code>)	GRUB 2 (<code>/etc/default/grub</code> , <code>grub2-mkconfig</code>)
<code>chkconfig</code> , <code>service</code> , <code>runlevels</code>	<code>systemctl</code> and <code>systemd</code> targets
<code>ifconfig</code> , <code>route</code> , <code>netstat</code>	<code>ip addr</code> , <code>ip route</code> , <code>ss</code>
<code>redhat-config-*</code>	<code>system-config-*</code> , now mostly <code>nmtui</code> , <code>cockpit</code>
Regular UID starts at 500	Regular UID starts at 1000

Practice Questions

Part A: Multiple Choice Questions (15)

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

- Q1.** Which Red Hat client utility checks Red Hat Network for updated RPM packages and installs them?
- (a) rpmbuild
 - (b) up2date
 - (c) mkinitrd
 - (d) chkconfig
- Q2.** Which command is used to register a system with Red Hat Network?
- (a) rhn_sync
 - (b) rhn_login
 - (c) rhn_register
 - (d) rpm --register
- Q3.** By default, which package group appears in the up2date package skip list?
- (a) kernel*
 - (b) glibc*
 - (c) bash*
 - (d) openssh*
- Q4.** Which command installs a new kernel RPM while keeping the old kernel for fallback?
- (a) rpm -Uvh kernel.rpm
 - (b) rpm -e kernel.rpm
 - (c) up2date -u
 - (d) rpm -ivh kernel.rpm
- Q5.** Which command removes old configuration files and stale objects before configuring a kernel source tree?
- (a) make clean
 - (b) make mrproper
 - (c) make install
 - (d) make bzImage
- Q6.** While configuring a kernel, the option letter 'm' means that the feature is:
- (a) Excluded from the kernel
 - (b) Compiled into the kernel image
 - (c) Built as a loadable module
 - (d) Mandatory
- Q7.** Which command copies the compiled kernel modules to /lib/modules/<version>?
- (a) make modules_install
 - (b) make modules
 - (c) make dep
 - (d) make install_boot

Q8. Which file normally contains the GRUB boot menu on Red Hat systems?

- (a) /etc/lilo.conf
- (b) /boot/vmlinuz
- (c) /etc/inittab
- (d) /boot/grub/grub.conf

Q9. Which runlevel provides full multi-user mode with a graphical login?

- (a) 3
- (b) 5
- (c) 1
- (d) 6

Q10. Which command is used to enable or disable services in particular runlevels?

- (a) service
- (b) ifup
- (c) chkconfig
- (d) modprobe

Q11. In /etc/passwd, which field contains the numeric user ID (UID)?

- (a) Third
- (b) Second
- (c) Fifth
- (d) Seventh

Q12. Which file stores the encrypted user passwords and password ageing information?

- (a) /etc/passwd
- (b) /etc/group
- (c) /etc/skel
- (d) /etc/shadow

Q13. Which command deletes a user account together with the user's home directory?

- (a) userdel user
- (b) userdel -r user
- (c) usermod -L user
- (d) groupdel user

Q14. Which command displays the password-ageing information of a user?

- (a) passwd -x user
- (b) useradd -l user
- (c) chage -l user
- (d) id user

Q15. Which directory contains the template files copied to a new user's home directory?

- (a) /etc/skel
- (b) /etc/default
- (c) /var/spool
- (d) /usr/src

Answer Key for MCQs

Q.	Answer	Reason
1	(b)	up2date is the RHN client that downloads and installs updates.
2	(c)	rhncerttool creates the system profile on RHN.
3	(a)	pkgSkipList=kernel*; prevents unattended kernel updates.
4	(d)	-i installs, so the old kernel is retained; -U would remove it.
5	(b)	make mrproper performs a deep clean, including .config.
6	(c)	m builds the feature as a loadable module.
7	(a)	make modules_install copies modules to /lib/modules.
8	(d)	GRUB reads /boot/grub/grub.conf.
9	(b)	Runlevel 5 is multi-user with X11 (graphical).
10	(c)	chkconfig manages service start-up per runlevel.
11	(a)	Format: name:x:UID:GID:comment:home:shell, so UID is the third field.
12	(d)	/etc/shadow holds hashes and ageing data.
13	(b)	-r removes the home directory and mail spool.
14	(c)	chage -l lists the ageing details.
15	(a)	/etc/skel supplies default startup files for new users.

Part B: Broad Questions (10)

1. What is Red Hat Network? Explain the role of up2date and describe the complete procedure of registering a system and updating it. Draw a diagram of the RHN architecture.
2. Describe the different types of errata. Explain the important parameters of the file /etc/sysconfig/rhn/up2date and why the kernel is placed on the skip list.
3. Explain the functions of the Linux kernel with a neat diagram. Distinguish between a monolithic kernel and a microkernel, and explain the kernel version numbering scheme.
4. Compare the two methods of obtaining a new kernel. Why is rpm -ivh preferred over rpm -Uvh when installing a kernel? Describe the role of initrd and grub.conf.
5. Explain step by step how a custom kernel is compiled from source and installed. Mention the purpose of each make target and the three choices (y, m, n) in kernel configuration.
6. What are loadable kernel modules? List and explain the commands used to manage them and describe how sysctl is used to tune the running kernel.
7. Describe the Linux boot sequence and the runlevels. Explain how services are started and stopped using the scripts in /etc/rc.d, and how chkconfig and service are used.
8. Explain how the network is configured from the command line in Red Hat Linux. Write the contents of ifcfg-eth0 for a static address and list the commands used to test the configuration.
9. Describe the structure of /etc/passwd, /etc/shadow and /etc/group with examples. Why were shadow passwords introduced?
10. Explain the commands used to create, modify, lock and delete users and groups. Discuss password ageing with chage, file permissions including SUID, SGID and sticky bit, and the use of sudo.

End of Notes