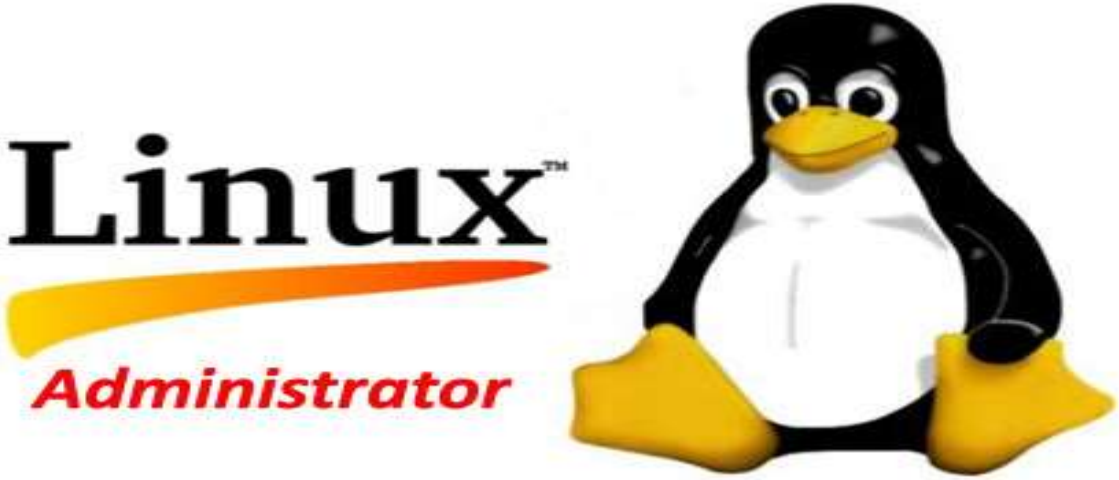


Linux System Administration

Prepared By : Soorya M U
Assistant Professor
Department of Computer Science
Al-Ameen College ,Edathala, Aluva

- Ultimate authority in a Linux environment.
- Has the responsibility to govern the system and care everything

- Under Linux, every file and program must be owned by a user. Each user has a unique identifier called a user ID. Each user must also belong to atleast one group,
- Group - a collection of users established by the system administrator.
- Like users, groups also have unique identifiers called group IDs.

r

- The administrator of Linux has all the rights required to control the working of the system. He is responsible for system maintenance.
  - Adding, removing, or updating user account information, resetting passwords, etc.
  - Applying operating system updates, patches, and configuration changes.
  - Allocates storage space
  - Assigns access permissions
  - Prevents unauthorized permissions and ensure security.
  - Installing and configuring new hardware and software.

- The utility to be used to get super user privileges is su
- After typing the password, the prompt for the regular user ($) will be changed to root user prompt #.
- The *su* (short for *switch user*) command makes it possible to change a login session's *owner.*
- Although su can be used to change the ownership of a session to any user, it is most commonly employed to change the ownership from an ordinary user to the *root*

The root user has complete control of the operation of the Linux system.

- The home directory of the root user is **/root**
- The home directory and other information associated with the root user account is located in the **/etc/passwd file**.
- An entry in the passwd file is as given

  root:x:0:0:root:/root:/bin/bash

- This shows that the user is named root, the user id is set to 0 and the group id is set to 0 , the home directory is /root and the shell for that user is /bin/bash
- /etc/passwd file – contains information required during login

Administrative Commands

- commands are intended only for root user.
- When a root user logs in , $PATH variable is set to include some directories that contain commands for the root user.
  - /sbin
  - Contains vital system utilities.
  - a standard subdirectory of the root directory in linux.
  - contains programs that are normally executed by the root user.

/usr/sbin – contains non-vital system utilities.

commands for managing user accounts and configuring your mouse. Commands that run as daemon processes are also contained in this directory.

Administrative Configuration files

- Most of the configuration files in Linux system are in the /etc directory .

- Commonly used configuration files found in the /etc are given below:

/etc/group	a list of groups with configurations for each
/etc/fstab	automatically mounts file systems when you start your system
/etc/motd	System administrator's message of the day
/etc/mtab	currently mounted file systems
/etc/passwd	user password and login configurations
/etc/shadow	User encrypted passwords
/etc/shells	shells installed on the system that users can use

/etc/shadow File

- ***/etc/shadow*** Contains the encrypted password information for users' accounts and optionally the password aging information. Included fields are:

smithj:Ep6mckrOLChF.:10063:0:99999:7:::

  - Login name or username
  - Encrypted password
  - No.of Days since the password was last changed
  - No.of Days before password may be changed
  - No.of Days after which password must be changed
  - The number of days to warn user of an expiring password
  - No.of Days after password expires, that account is disabled
  - No.Days that account has been disabled

- Shadow passwords are an enhancement to login security on Unix systems. Traditionally, passwords are kept in encrypted form in a world-readable table (/etc/passwd).
- To test a password, a program encrypts the given password with the same "key” that was used to encrypt the password stored in the /etc/shadow file.
- Because the encrypted passwords are not "decryptable", authentication takes place by comparison. If the /etc/shadow file password matches the encrypted login password, the user is granted access.

- Advantages of shadow passwords over the traditional way of storing passwords :
  - Improves system security by moving encrypted password from the world-readable /etc/passwd file to /etc/shadow, which is readable only by the root user.
  - Stores information about password aging.

- Logfile – a file that records either events that occur in an OS or other software runs, or messages between different users
- We can use log files to keep track of the system.
- Almost all logfiles are located under/var/log directory and its subdirectories on Linux.
- Only root user can view and access logfiles on linux.

- General system logging is done by syslogd.
- Logging that is specific to kernel activity is done by klogd.
- Logging is done according to the information in the /etc/syslog.cof file.
- Commands to see logfiles
- tail,more, cat,grep etc

Eg:

#cat /var/log/messages

- ***/etc/hosts* -** Contains a list of known hosts (in the local network).
- ***/etc/hosts.allow*** - lists host computers that are allowed to use certain TCP/IP services from the local computer.
- ***/etc/hosts.deny* -** lists host computers that are not allowed to use certain TCP/IP services from the local computer.

- Following is a sample /etc/hosts file:

IPAddress	Hostname	Alias
127.0.0.1	localhost	
208.164.186.1	deep.openna.com	deep
208.164.186.2	mail.openna.com	mail
208.164.186.3	web.openna.com	web

Managing User Accounts

Following are commands available on the majority of Unix systems to create and manage accounts and groups:

Command	Description
useradd	Adds accounts to the system.
usermod	Modifies account attributes.
userdel	Deletes accounts from the system.
groupadd	Adds groups to the system.
groupmod	Modifies group attributes.
groupdel	Removes groups from the system.

- Most widely available method for creating a new user from the shell is with the `useradd command`.
- The only required parameter to useradd is the login name of the user.
    - Syntax:

useradd [options] username

useradd chris –g mca1 -u 578

#passwd chris

Changing password for user chris

New UNIX password:

Retype new UNIX password:

Options

- -c “comment” provides a description of the new user account.

  #useradd –c “MarySmith” mary

-e expiry_date assign the expiration date for the account in MM/DD/YYYY format.

- -g group sets a group for the new user
- -d dir sets the home directory of the new user
- -s shell sets the login shell of the newuser

  default /bin/bash

- -u user_id -sets the user ID of the new user

- usermod command modifies user properties.

- Syntax: `usermod [options] *username*`
- Options are identical to those of useradd
- -l to change the user's login name

Changing user account properties

- There are a few commands for changing various properties of an account (i.e., the relevant field in /etc/passwd):
- **Chfn** Change the full name field.

  chfn –f "fullname" username
- **Chsh** Change the login shell.
- **passwd** Change the password.

- If we want to **disable a user account** , replace the password in the /etc/passwd file with a * or some other character. Since * is not a valid encrypted password there is no password that will allow you to log on to that account.

- To **<u>delete an account</u>** , remove the entire line from the /etc/passwd file.
- Deleting a user accounts can be done with either the userdel command or the User Manager window.
- Using userdel
  - userdel command takes a single argument, which is the login name of the account to delete.
  - -r option deletes the user's home directory and all the files in it.

  Eg: #userdel –r mary

Disabling & Deleting accounts-differences

- If we disable an account technically still it exists. The user can still receive mail, has a home directory and has all rights of any other account. The user just can't log in.
- When an account is deleted , the system will bounce any mail that comes to that user and also any access to the home directory will not work.

Suspending a User Account

To temporary disabling user account

- **Put a * as start of Password field in /etc/passwd**
- **Change login shell to /sbin/nologin**

- By default, permissions are set for the owner of the file, the group associated with the file, and everyone else who can access the file.
- Permissions are divided into 4 parts –
  - First part represents the special attribute (-, d or 1)
  - Second part indicates file owners permission
  - Third part indicates the group permission
  - Last part indicates the world permission.
- chmod command is used to set permission values.

R	Read	4
W	Write	2
X	Execute	1

- The chown command is used to change the ownership of a file to someone else.
- Only the root user can do this.
- Syntax

  #chown [-R] username filename

  username - the login name of the user to whom ownership ashould be assigned.

  filename-the name of the file in question. The filename may be a directory as well.

  -R option

  - tells the command that the specified filename is a directory name

  - recursively descend through the directory tree and apply the new ownership not only to the directory itself, but to all of the files and directories within it.

- Linux groups are a mechanism to manage a collection of system users. All Linux users have a user ID and a group ID.
- Users are members of a default group. The default group for a user is specified in the file /etc/passwd
- There are some main user administration files:
  - **/etc/passwd:** Keeps user account and password information. This file holds the majority of information about accounts on the system.
  - **/etc/shadow:** Holds the encrypted password of the corresponding account.
  - **/etc/group:** This file contains the group information for each account.
  - **/etc/gshadow:** This file contains secure group account information.

- You would need to create groups before creating any account otherwise you would have to use existing groups at your system. All the groups are listed in */etc/group* file.
- A sample entry in an etc/group file
- engines:x:100:chris,robert,valerie,aleena
- All the default groups would be system account specific groups and it is not recommended to use them for ordinary accounts.

Option	Description
-g GID	The numerical value of the group's ID.
-o	This option permits to add group with non-unique GID
-r	This flag instructs groupadd to add a system account
-f	This option causes to just exit with success status if the specified group already exists. With -g, if specified GID already exists, other (unique) GID is chosen
groupname	Actaul group name to be created.

Groups should be created before creating any account,otherwise use existing groups at your system. All the groups are listed in */etc/group* file.

- A sample entry in an etc/group file
- engines:x:100:chris,robert,valerie,aleena
- All the default groups would be system account specific groups and it is not recommended to use them for ordinary accounts.
- Syntax to create a new group account:
  - groupadd [-g gid [-o]] [-r] [-f] groupname

Option	Description
-g GID	The numerical value of the group's ID.
-o	This option permits to add group with non-unique GID
-r	This flag instructs groupadd to add a system account
-f	This option causes to just exit with success status if the specified group already exists. With -g, if specified GID already exists, other (unique) GID is chosen
groupname	Actaul group name to be created.

- To modify a group, use the **groupmod** syntax:

  $groupmod -n modified_group_name old_group_name

  -n to change the group name to the specified group name
- To change the developers_2 group name to developer, type:

  $ groupmod -n developer developer_2
- To change the GID to 545:
- **$ groupmod -g 545 developer**

- To delete an existing group, use the **groupdel** command and the group name.
- Eg: To delete the developer group, the command is:

  $ groupdel developer

- This removes only the group, not any files associated with that group. The files are still accessible by their owners.

Change the owner of a file

- # ls –l tmpfile

-rw-r--r-- 1 steeve family 0 2012-05-22 20:03 tmpfile

- # chown root tmpfile
- # ls -l tmpfile

-rw-r--r-- 1 root family 0 2012-05-22 20:03 tmpfile

- Every file is associated with an owner and a group.
- chown and chgrp commands can be used to change the owner or the group of a particular file or directory.

Using chown command, the group (that a file belongs to) can also be changed.

ls -l tmpfile

-rw-r--r-- 1 steeve **family** 0 2012-05-22 20:03 tmpfile

- # chown :friends tmpfile
- # ls -l tmpfile

  -rw-r--r-- 1 steeve **friends** 0 2012-05-22 20:03 tmpfile

By just adding a ‘:’ followed by the new group name, the group of the file can be changed.

- # ls -l tmpfile

-rw-r--r-- 1 root family 0 2012-05-22 20:03 tmpfile

- # chown steeve:friends tmpfile
- # ls -l tmpfile

-rw-r--r-- 1 steeve friends 0 2012-05-22 20:03 tmpfile

- using the syntax
- ‘<newOwner>:<newGroup>’, the owner as well as group can be changed in one go.

- **chgrp** (CHange GRouP) is one more command which is useful to change group associated to a file/folder from one group to other in a Linux box.
- This is sister command to **chown** which is used to change owner of the file/folder as well as group name associated with that file.

- **Example1:**

Change group name **sales** of file1 to other group name **hrgroup**.

chgrp hrgroup file1

- **Example2:** Change group ownership all the files located in /var/apache to group:apache

chgrp -R apache /var/apache

Difference between chown and chgrp

chown command is used to change ownership as well as group name associated to different one, where as **chgrp** can change only group associated to it.

Managing file systems and disk space

- Most of your hard disks are mounted automatically.
- We can use mount command to mount other type of devices and also other kinds of file systems on your Linux system.
- Ie we can store files from other OS or use file systems that are appropriate for certain kind of activities like writing large block sizes.

- Mounting: making a device accessible to users via the logical directory tree
- Attaching a file system on a storage device to the main directory tree
- Mount point: directory in the file structure to which the new filesystem is attached.
  - Any existing directory can be a mount point
- In order to prevent making files inaccessible, create empty directories used specifically for mounting devices

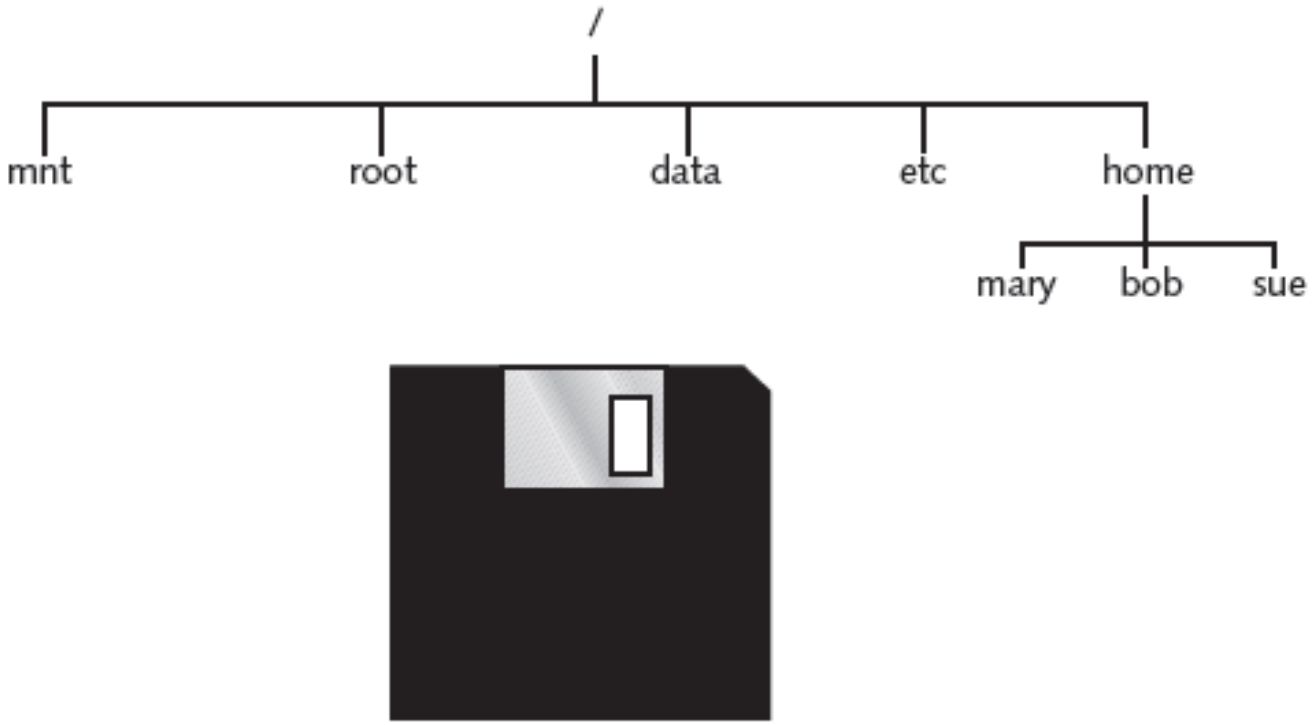


Figure 5-1: The directory structure prior to mounting

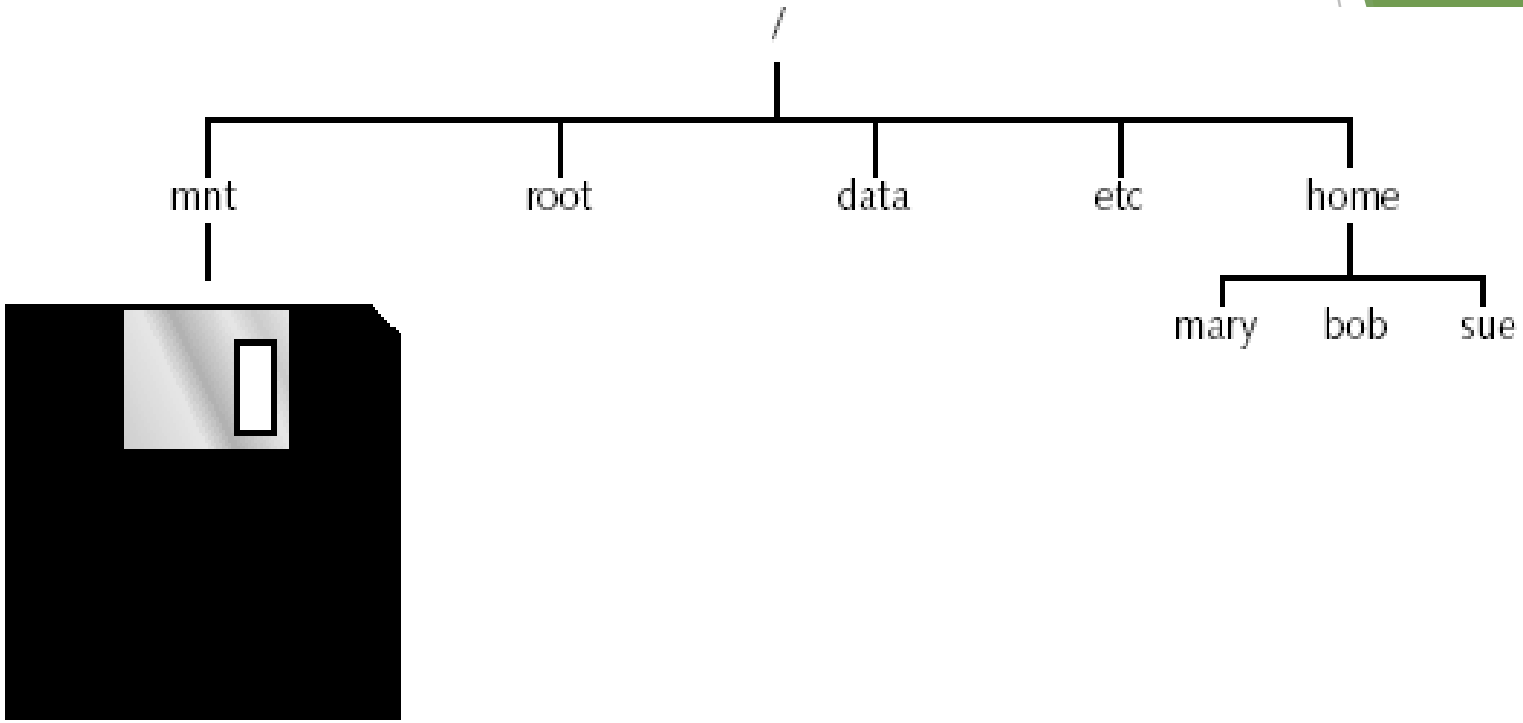


Figure 5-2: The directory structure after mounting a floppy device

- /mnt directory
- usually used as mount point for other mounted file systems such as windows partitions.

- Automatic mounting is handled by configuring
- **/etc/fstab** file.
- Main partitions holding Linux systems are automatically mounted while booting.

Mounting and unmounting a file system

Mount command

- The mount command is used to mount a file system to the Linux operating system.
- The syntax for mount command is

  Mount <[-t type] [options]> <device>[directory]
- When mount command is displayed without options a list of file systems mounted on the system is displayed.

- In the syntax for the mount command - type is optional and is used to specify the type of file system.
- Examples of types of file systems
- ISO9660 the standard file system used for CD-ROMS
- MSDOS the standard file system used by MS-DOS
- Vfat the standard file system used by windows 3x and 9x

Some examples of device files

- /dev/fd0 the primary floppy drive
- /dev/fd1 the secondary floppy drive
- /dev/cdrom refers to CDROM Drive

Options

- rw specifies that the file system can be both read from and written to.

- Device name for Floppy drive -fd
- IDE hard drive – hd
- SCSI hard drive – sd

- hda2 -2nd partition on first(represented by a) IDE hard drive
- sdb3 – 3rd partition on 2nd(represented by b) SCSI hard drive

mount /dev/fd0 /mnt/floppy

- This command will connect the device "/dev/fd0" (usually the floppy drive) to the directory "/mnt/floppy" so that you can access the files and directories (folders) on the floppy disk in the floppy drive under the "/mnt/floppy" directory. The directory "/mnt/floppy" is also called the "mount point", which must already exist when this command is executed.

- ro specifies that the file system can be read from but cannot be written to.
- Example of mount command to mount a floppy
- # mount –t vfat /dev/fd0 /mnt/floppy

Unmount command

Syntax

Unmount [device][directory]

Example

unmount /mnt/floppy

unmount /dev/fd0

- This command unmounts the floppy drive. After this command is executed the files and directories on the floppy will no longer be accessible from the directory tree of the Linux system.

- Removable storage devices are handled by udev and HAL(Hardware Abstract Layer)

- udev(user devices)
- – the tool used to detect and generate device files
- Capable of managing all removable devices.

HAL – allows removable devices like a USB to be recognised.

Files and File Structure

What is File System?

- It is responsible for storing information on disk and retrieving and updating this information.
- Example :
  - FAT16, FAT32, NTFS
  - ext2, ext3
  - …
- In Linux everything is file.

Type of File System

- Network File System
  - NFS (Network File System)
  - SMB(Server Message Block)
- Disk File System
  - ext2 (second extended filesystem)
  - ext3
  - etc…

File System

- Managing file system is basic task for system administrators
- Partition
  - Distinct area of hard disk
  - Has been prepared to store particular type of data
- File system
  - Arrangement of information on device such as hard disk
  - The whole area is divided into several block

General Structure

Boot block
Super block
Inode table
Data blocks

Boot Block

- An area of a disk having information for loading the operating system that is needed to start a computer.
- located in the first few sectors of a file system.
- Contains initial bootstrap program used to load the operating system.
- Blocks on a Linux filesystem are 1024 bytes in length
- Some systems use 512 bytes ,but 2048 and 4096 are also seen

Super Block

- The boot blocks are followed by the superblock
- Each file system has one super block
- It contains information about the geometry of the physical disk
- it contains
  - 1. type of file system (ext2, ext3...)
  - 2. the block size
  - 3. pointers to a list of free blocks
  - 4. the inode number of the root directory

inodes

- Disk space allocation is managed by the inodes (**i**nformation **node**)
- Inodes cannot be manipulated directly, but are changed by many commands

- Inode: Data Block Addressing
  - Every Inode has a table of block addresses
  - Addressing: direct, one-level indirect, two-levels indirect, ...
  - Each inode corresponds to one file, and it stores file's primary metadata, such as file's size, ownership, and temporal information.
  - Inode is typically 128 bytes in size and is allocated to each file and directory

Structure of a directory entry

Inode number	File name

Contents of an inode

Link Count
File Mode
User ID
Time Created
Time Last Updated
Access Permissions

.
.
.

File's Location on Disk

Data Blocks

- This is where the file data itself is stored
- An allocated data block can belong to one and only one file in the system.
- If a data block is not allocated to a file, it is free and available for the system to allocate when needed.

The Linux file System

- In Linux, file is defined as simply the thing that deals with a sequence of bytes
- Hence everything are files
  - An ordinary file is a file; a directory is also file; (Files containing list of files within them)
  - a network card, a hard disk, any device are also files since they deal with a sequence of bytes

The Linux file System

- **The Linux file system looks like an inverted tree structure.**
- **We start with the root directory, denoted by /, at the top and work down through sub-directories below it.**
- **Each node is either a file or a directory of files, where the latter can contain other files and directories**

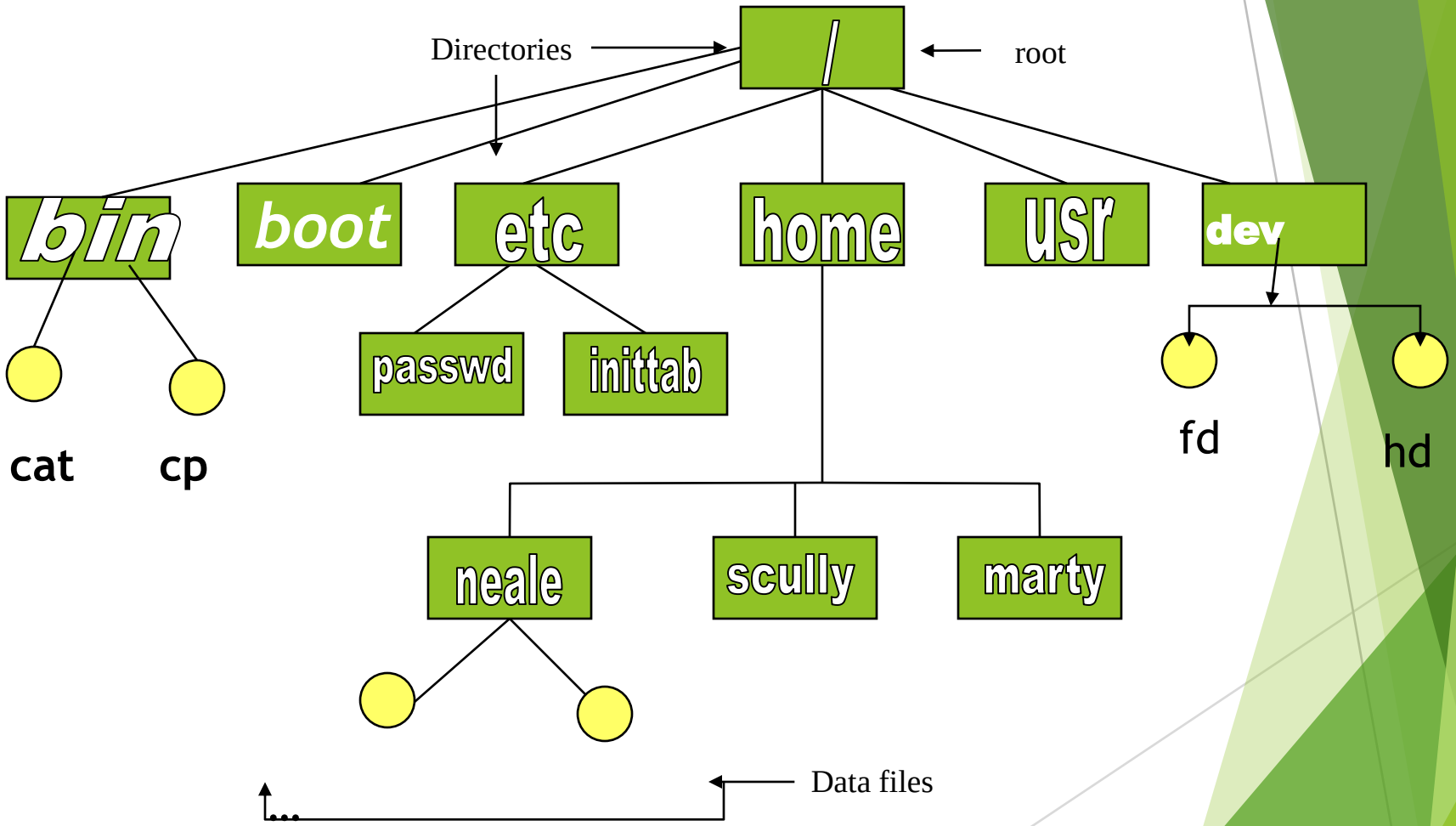
The Linux ile System

- **We specify a file or directory by its path**

- **The full path name starts with the root, /, and follows the branches of the file system, each separated by /, until you reach the desired file, e.g.:**

- **/home/ckc/desktop/stud**

The Linux file System



Directory Structure of Linux File System

Directory	Content
/bin	Common programs, shared by the system, the system administrator and the users.
/boot	The startup files and the kernel,
/dev	Contains references to all the CPU peripheral hardware, which are represented as files with special properties.
/etc	Most important system configuration files are in /etc, this directory contains data similar to those in the Control Panel in Windows
/home	Home directories of the common users.
/lib	Library files, includes files for all kinds of programs needed by the system and the users.

Directory	Content
/lost+found	Every partition has a lost+found in its upper directory. Files that were saved during failures are here.
/misc	For miscellaneous purposes.
/proc	A virtual file system containing information about system resources.
/root	The administrative user's home directory. the difference between /, the root directory and /root, the home directory of the *root* user.
/sbin	Programs for use by the system and the system administrator.
/tmp	Temporary space for use by the system.
/usr	Programs, libraries, documentation etc. for all user-related programs.
/var	Storage for all variable files and temporary files created by users, such as log files, the mail queue, the print spooler area, space for temporary storage of files downloaded from the Internet.

File naming Conventions

Some applications require certain extensions, but LINUX does not use any standard convention for file extensions

In Linux the file name

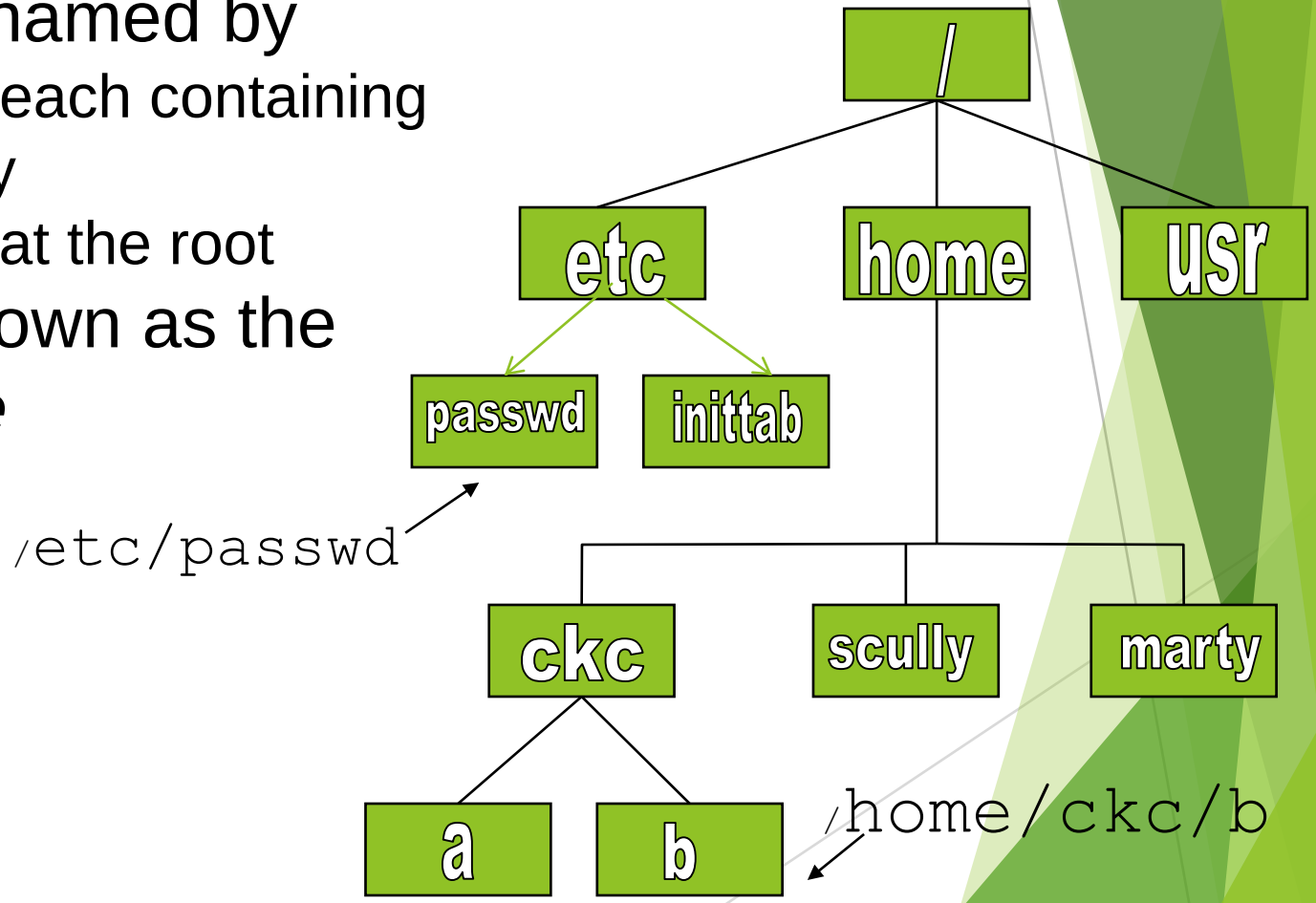
- can be up to 256 characters long
- can contain special characters, except ‘/’
- case sensitive
- should not contain a blank or a tab
  - Can contain both uppercase and lowercase alphabets.
    - …

Relative path names

Files are named by

- naming each containing directory
- starting at the root

This is known as the *pathname*

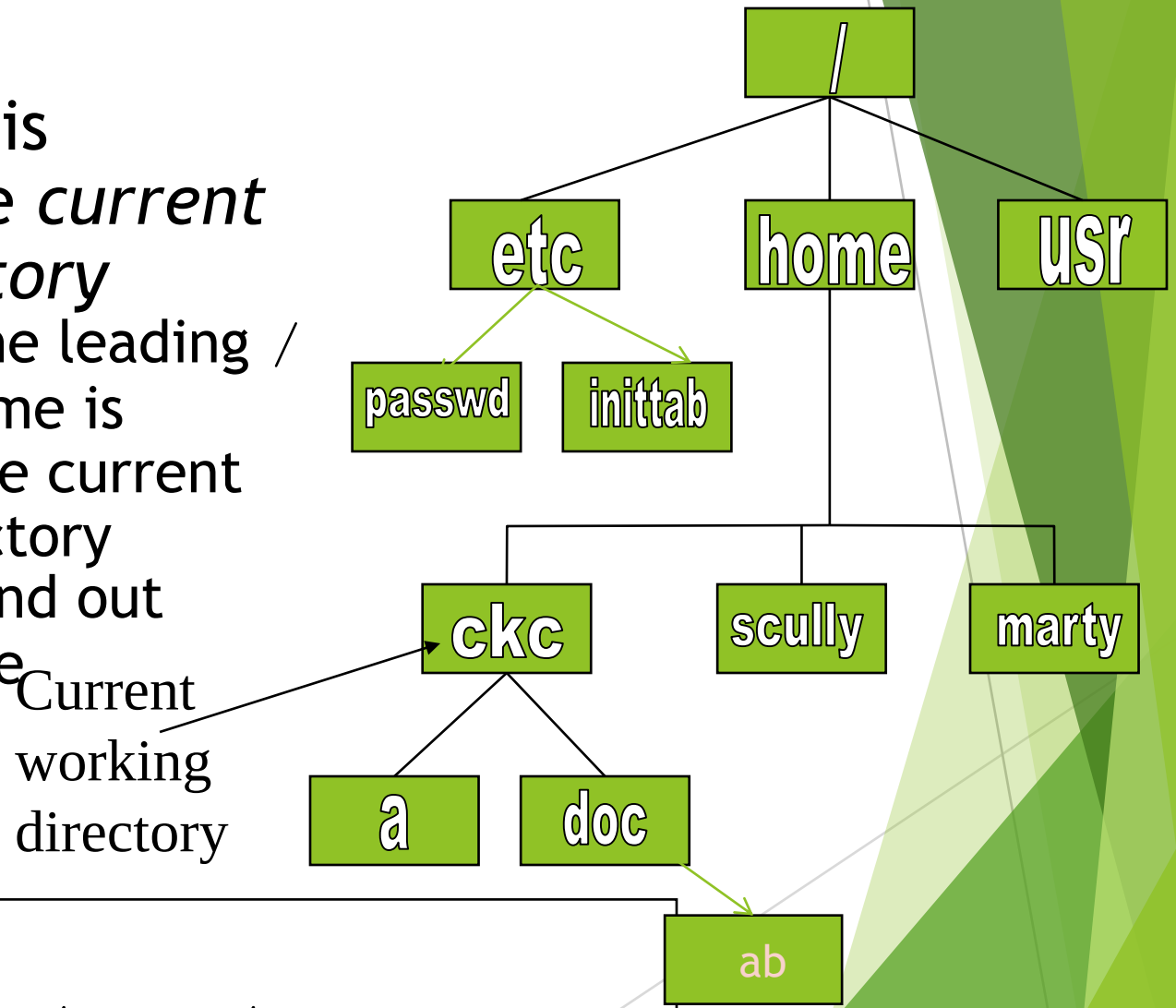


…

The Working Directory(Home Directory)

One directory is designated the *current working directory*

- if you omit the leading / then path name is relative to the current working directory
- Use pwd to find out where you are




```
doc/ab
~/doc/ab
/home/ckc/doc/ab
```

Some Special File Names

Some file names are special:

/ The root directory (not to be confused with the root user)

. The current directory

.. The parent (previous) directory

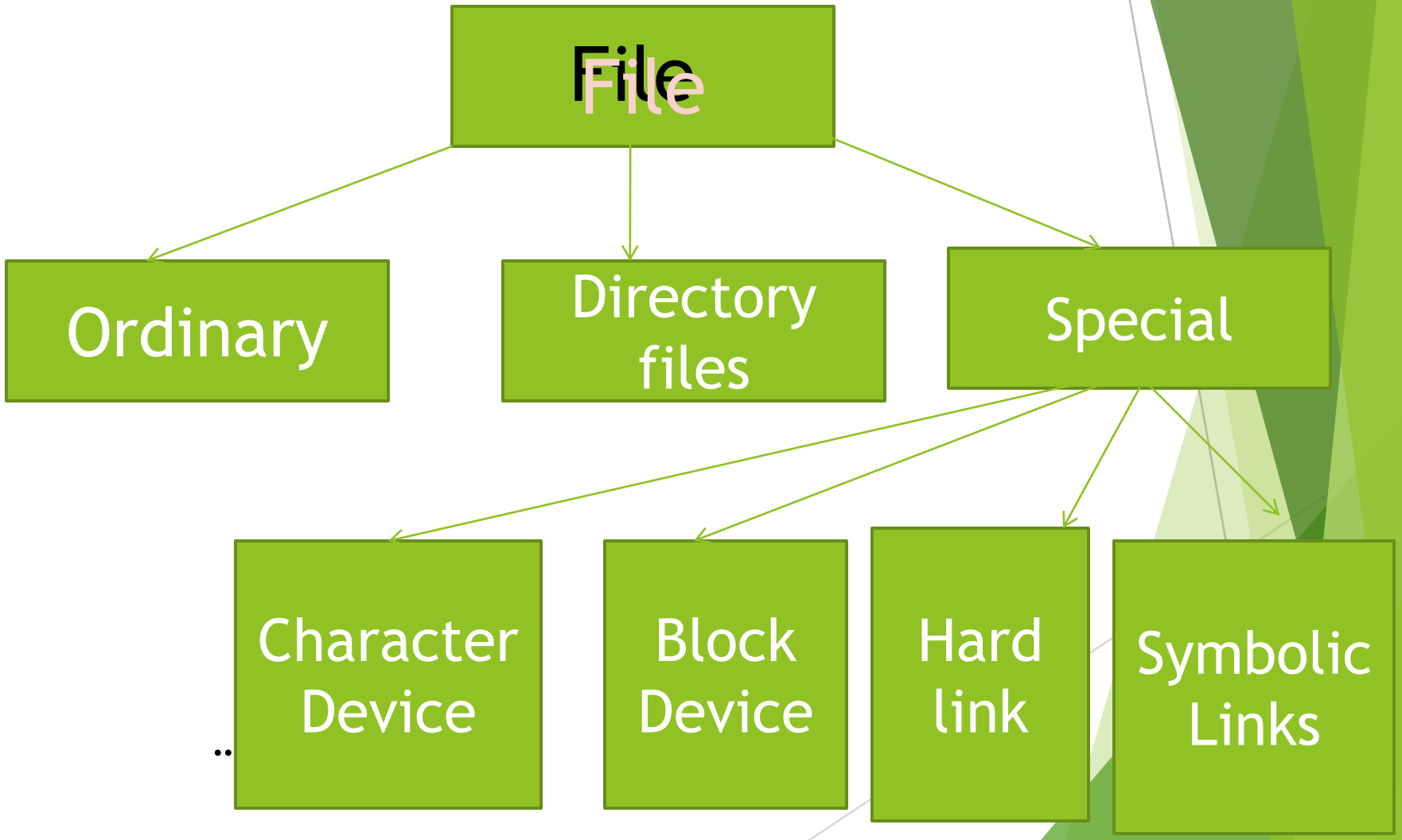
~ My home directory

Examples:

./a same as a

../stud/x go up one level then look in directory stud for x

Types of Files in Linux



Types of Files in Linux

- Linux supports 3 types of files
  - simple/ordinary file (text file, c++ file, etc)
  - Directory
  - special file (device)

➢ **simple/ordinary file**
  - ➢ All files created by a user
  - ➢ Include all data ,pgm, object and executable files
  - ➢ A user can modify it

…

Types of Files in Linux

- **Directory Files**
  - File systems typically have **directories** which allow the user to group files into separate collections
  - This may be implemented by associating the file name with an an inode in a Linux-like file system.
  - The directory file is automatically created by the OS.

…

- The directory file has the same name as the directory and contains information about the files stored in the directory.
- Eg: the directory /home/steve contains a directory file called steve in the directory /home.
- This directory file contains information about all the files and directories in the directory steve.
- A user can not modify a directory file.

Types of Files in Linux

- **Special file (device)**
  - Each hardware device, e.g. keyboard, hard disk, CD-ROM, etc is associated with at least one file
  - Usually store in /dev directory
  - Applications can read and write any devices by reading and writing their associate file – hence the access method is known as device independent
  - Special files are stored in */dev and /etc*
  - A user can not modify special files.

…

Types of Files in Linux

- Divide into 4 types:
  - character device files,
  - block device files, e.g. disk
  - Hard links
  - Symbolic link

- **character device files**
  - Read and write one character at a time
  - Also known as sequentially accessed devices
  - e.g. keyboard,modem

Types of Files in Linux

- **Block Device Files**
  - Access one block of data at a time
  - A block of data cpmprises either 512 or 1024 bytes.
  - The kernel collects data in the memory and then makes the data available to a user
  - Block devices allow random access, which makes i/o operation faster
  - Eg; hard disk

…

- Many devices can act as either character or block devices, depending on the command used to access the device.

Types of Files in Linux

- **Hard links**
  - Hard links are special files that allow a single file to have multiple names
  - Hard link only for a file not for a directory
  - These links are known as hard links bcoz they create direct link to an inode
  - Each file system has its own inode information database.
  - Therefore, you can specify hard links for files only when they are on the same file system

- **Symbolic (soft) link**

  similar to hard links, but you can specify symbolic links for files across different file systems

  - Not a real file, just a link to another file
  - Allow giving another name to a file without actually duplicates it – hence save memory space

Types of Users in Linux

- **System Administrator**
  - Root
    - Administrative account
    - Also called super user
    - Can perform any operation on Linux system
    - Do not log in as root for normal work
    - Change temporarily to root user

…

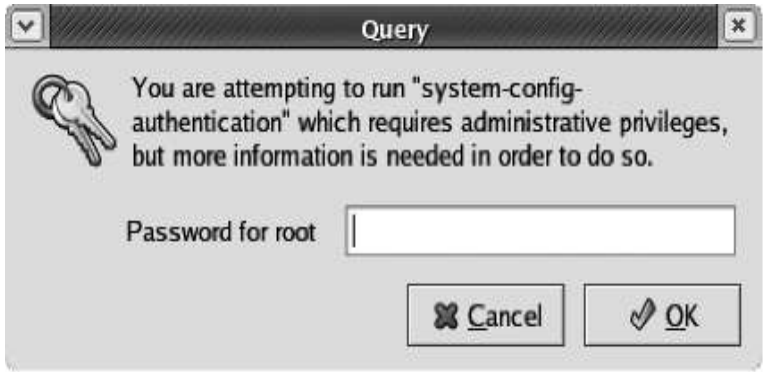


Figure 4-1 Entering the root password to run a restricted program

File Owner(User(u))

- The user who create the file known as the owner of the file.
- Owner can perform any operation on that file, such as copying, deleting or editing.

Group Owner(User group(g))

- Users who belong to a group
- we can specify a name for the group of users
- Collection of user accounts
- Can be collectively granted access to files and directories

Other Users (o)

- Do not belong to a particular group

Text Editors

Text editors

- A text editor is a program that enables us to create and modify text files
- A text editor provides a screen with a fixed line length and line numbers. we can type text line by line, navigate through the documents by using different commands and save the document.

A text editor is used to

- Create and edit documents
- Create programs and utilities
- Write e-mail messages

Functions of a Text editor

- Create files
- Open files
- Copy text
- Search for text and replace it

Editors in Linux

- vi Editor
- vim Editor
- emacs editor
- ed editor
- red editor
- joe editor
- pico editor

Vi Editor

- The visual editor is the most widely used UNIX based text editors.
- It offers a compact interface and enables us to control the system by using the keyboard.
- The vi editor is used
  - to enter and edit source code
  - write short notes such as email messages

Vim editor

- *visual editor improved* or vim editor
- an enhanced version of the vi editor .The vim editor includes various enhancements over the vi editor such as
- Syntax highlighting
- Command-line editing
- Online help

emacs editor

- The emacs editor is the *edit macros* editor.
- The emacs editor provides a much larger set of commands than the vi editor .
- we can use emacs editor to format a set of source code for programming languages such as c, c++.
- emacs editor displays an online help screen.

ed editor

- ed editor or line editor is used to create, display, modify and save text files.
- When we start the ed editor with a filename as an argument , a copy of the file is created in the editor's buffer .we can make changes only to the copy of the file and not directly to file.
- To save the changes to the file use the **w** command.

red editor

- The restricted editor is used to edit files in the current directory .
- we cannot execute shell commands by using the red editor.

joe editor

- Another popular editor is the *Joe's own editor* or joe.
- joe editor can be invoked by typing joe at the command prompt.
- File name can be provided along with joe command to open the file in the joe editor.
- The joe editor is a full screen editor that lets us edit both programs and text.

pico editor

- Pico editor is based on pine messaging system.
- Pine messaging system is a character based interface in Linux. It can be used to send and receive email messages.
- Pico editor offers features like paragraph justification, spelling checker , file browser and search features.
- In pico, editor command are displayed at the bottom of the screen.
- pico provides context sensitive help.

vi Editor

Vi editor

- Vi editor is used to
  - Create files
  - Edit files, documents & programs
  - Displays contents of files on the screen.
- Getting started with vi Editor
  - $ vi filename<Enter>

- Vi editor works in two modes

  1.Edit Mode

  2.Command Mode

  In edit mode we can add text to a file.

  The vi editor has different commands to add, change and delete text. In the command mode we can move the cursor, save the file and give various commands to manage files.

- To switch from edit mode to command mode press the **<Esc>** key. To switch from command mode to edit mode press a, A, o, O, i, or I keys.

Commands used in vi editor

h	Moves the cursor to the previous character
l	Moves the cursor to the next character
k	Moves the cursor to the line above the current line
j	Moves the cursor to the line below the current line.
x	Deletes the character at the current cursor position
:wq<Enter>	Saves all changes and quits the vi editor
:w<Enter>	Saves the file

:q!<Enter>	Quits vi without saving changes
:e<filename>>Enter>	Opens the specified file
:r<filename><Enter>	Reads and inserts the contents of the file after the current line.
:w<filename><Enter>	Writes to the specified file.(From Vi file to another)
:W!<filename><Enter>	Forcefully writes to the specified file.
:!<Command_name> <Enter>	Executes a shell command

Insert and Replace Commands

a	Enables to append text after the current character
A	Enables to append text at the end of the line
i	Enables to insert text before the current character
I	Enables to insert text at the beginning of the file
o	inserts a blank line below the current line and allows to insert text
O	inserts a blank line above the current line and allows to insert text
rx	Replaces the current character with the specified character
Rtext<esc>	Replaces characters at the current cursor position with the specified text , until <Esc> Key is pressed

Cursor Movement Commands

<Ctrl>d or D	Scrolls down half-screen
<Ctrl>u or U	Scrolls up half-screen
<Ctrl>F	Moves a page forward
<Ctrl>B	Moves a page backward
0(zero)	Moves The cursor to the beginning of the line
$	Moves The cursor to the end of the line

Word Movement Commands

w	Moves the cursor to the next word
b	Moves the cursor to the previous word
e	Moves the cursor to the end of the word

Deletion and modification commands

dw	Deletes a word
dd	Deletes a line
cw	Changes a word
cc	Changes a line
X	Deletes the character before the current cursor position
J	Joins two lines
u	Undoes last change
.(dot)	Repeats last change made

Command to Copy Lines

yy or Y	Yanks the current line, which is to be copied
Nyy or NY	Yanks N lines to be copied, including the current line
p	places yanked text after the current line
P	places yanked text before the current line

Pattern finding commands

fx	Finds the specified character x on the current line **after** the current cursor position
FX	Finds the specified character x on the current line **before** the current cursor position a line
/pattern<enter>	find the next line containing the specified pattern
?pattern<enter>	find the previous line containing the specified pattern

n Repeat the last search command

N Repeats the search command in the opposite direction

Substitution

- :s -search and replace
- :adress s/source_pattern/target_pattern/flags

- Eg:
- :1,$s/director/member/g
- :1,50s/unsigned//g
- :3,10s/director/member/g
- :.s/director/member/g only the current line
- :$s/director/member/g only the last line

- :1,$s/director/member/gc -interactive substitution

Absolute Movement(G)

- Ctrl+g –to know the current line number
- 40G –go to line number 40
- 1G -go to line number 1
- G -go to end of file

Pattern	Meaning
^part	search for all **lines** which begin with the word 'part'
part$	search for all **lines** which end with the word 'part'
\<part	search for all **strings** which begin with the word 'part'
part\>	search for all **strings** which end with the word 'part'
\<part\>	only the whole word 'part' will qualify
[m-s]ing	will search for strings which contain any character in the range ‘m’ to 's' and is followed by 'ing'
[^p]art	search for all strings which contain the characters 'art', with 'art' being preceded by any character other than 'p'
wing*	All words which begin with characters 'wing' and ends with any character

Regular Expressions

Agarwal

aggarwal

Agrawal

Similar ,but not identical patterns

An expression used to match a group of similar patterns with the help of an elaborate metacharacter set is termed as regular Expressions.

eg: [aA]g[ar][ar]wal

2 Categories of Regular expressions are there

1. Basic Regular expressions(BRE)
2. Extended Regular expressions(ERE)

ERE

The ERE set includes 2 special characters

+ and ?

+ - Matches one or more occurrences of the previous character.

? - Matches zero or one occurrence of the previous character.

Eg: b+ matches b, bb,bbb etc

b? matches b or nothing

Eg: grep -E “[aA]gg?arwal” emp.lst

egrep “[aA]gg?arwal” emp.lst

```
#include <stdio.h>
#include    <stdio.h>
#include          <stdio.h>
```

```
#include +<stdio.h>
```

Expression	Significance
ch+	Matches one or more occurrences of the character ch.
ch?	Matches zero or one occurrence of the character ch.
exp1|exp2	Matches exp1 or exp2
GIF|JPEG	Mathces GIF or JPEG
(x1|x2)x3	Matches x1x3 or x2x3
(holy|bolly)wood	matches holywood or bolltwood

grep -E ‘senguptha|dasguptha’ emp.lst

grep -E ‘(sen|das)guptha’ emp.lst

-f - to take the pattern from the file

sed –The stream Editor

A non interactive text editor used to change or manipulate streams of text.

- Combines works of several filters
- Derived from ed editor.
- Syntax

sed options ‘ address action ’ file(s)

Addressing in sed can be done in two ways:

1. By one or two line numbers(like 3,7) **line addressing**
2. By specifying a /-enclosed pattern which occurs in a line. (like /pattern/) **Context addressing**

Actions are referred to as commands

```
$ sed '3q' emp.lst

$ sed -n '1,2p' emp.lst

$ sed -n '$p' emp.lst

$ sed -n '9,11p' emp.lst
```

To select Multiple groups of Lines

sed –n ‘1,2p

7,9p

$p’ emp.lst

Negating the action(!)

sed –n ‘3,$!p’ emp.lst don’t print lines3 to end

Using multiple instructions

sed –n -e ‘1,2p’ -e '7,9p’ -e ‘$p’ emp.lst

$cat pattern1

1,2p

7,9p

$p

sed –n –f pattern1 emp.lst

sed –n –f pattern1 –f pattern2 emp.lst

Context Addressing

$ sed –n ‘/director/p’ emp.lst

$ sed –n ‘/dasguptha/,/saksena/p’ emp.lst

$ sed –n ‘/[aA]gg*[ar][ar]wal/p’ emp.lst

Writing selected lines to a file(w)

sed –n ‘/director/**w dlist**’ emp.lst

sed –n ‘/director/w dlist

/manager/w mlist

/executive/w elist emp.lst

sed –n ‘1,500w file1

501,$w file2’ myfile

Deleting Lines(d)

Sed ‘/director/d’ emp.lst >olist

sed –n ‘/director/!p’ emp.lst >olist

select all lines except those containing director , and save them in olist.

Substitituion(s)

Syntax:

[address]s/expression1/expression2/flags

Eg:

sed ‘s/ | /:/’ emp.lst |head –n 2

sed ‘s/ | /:/g’ emp.lst

sed ‘1,3s/ | /:/g’ emp.lst

sed ‘1,5s/director/member /g’ emp.lst

sed ‘s/[Aa]gg*[ar][ar]wal/Agarwal/g’ emp.lst

sed ‘s/ ^ / 2 /g’ emp.lst

sed ‘s/ $ /.00 /g’ emp.lst

- pr
- - Prepares a file for printing by adding suitable headers,footers and formatted text.