Unix Shell Scripting Basics

Module 1

Agenda

1 Unix Operating System

2 Unix Commands

3 Files and Directories

4 User Management

1.Unix Operating System

Objectives

In this session, you learn about:

- The functions of OS
- The features of UNIX
- The Unix architecture
- File System
- Unix users
- Getting started
- User Management

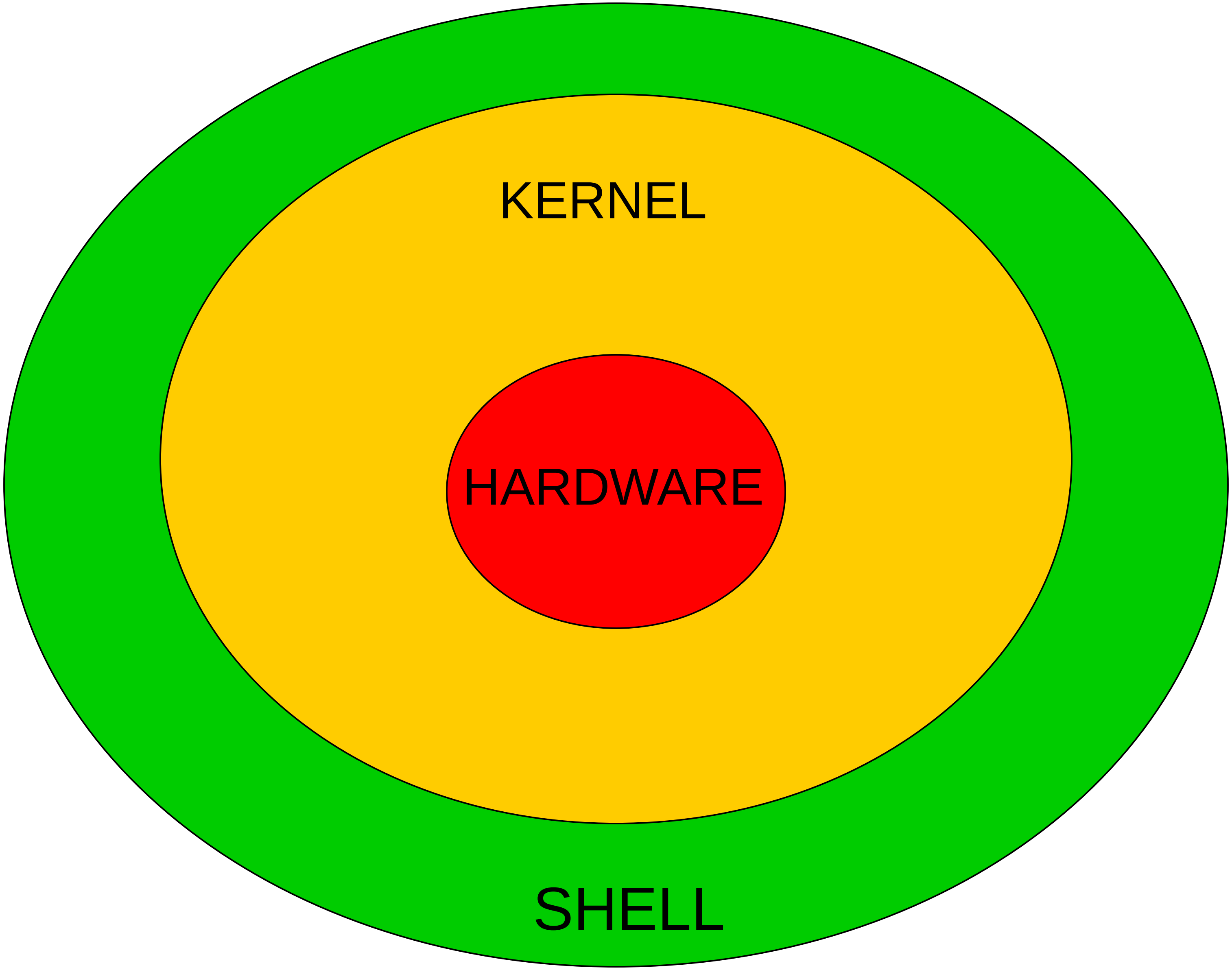
Operating System (OS)

- OS is a system software
- OS can be defined as an organized collection of software consisting of procedures for operating a computer
- OS provides an environment for execution of programs
- OS acts as an interface between the user and the hardware of the computer system.

Features of UNIX

- Multi-user, multitasking, timesharing
- Portability
- Modularity
- File structure
- Security

Layered Architecture



Layered Architecture (Contd.).

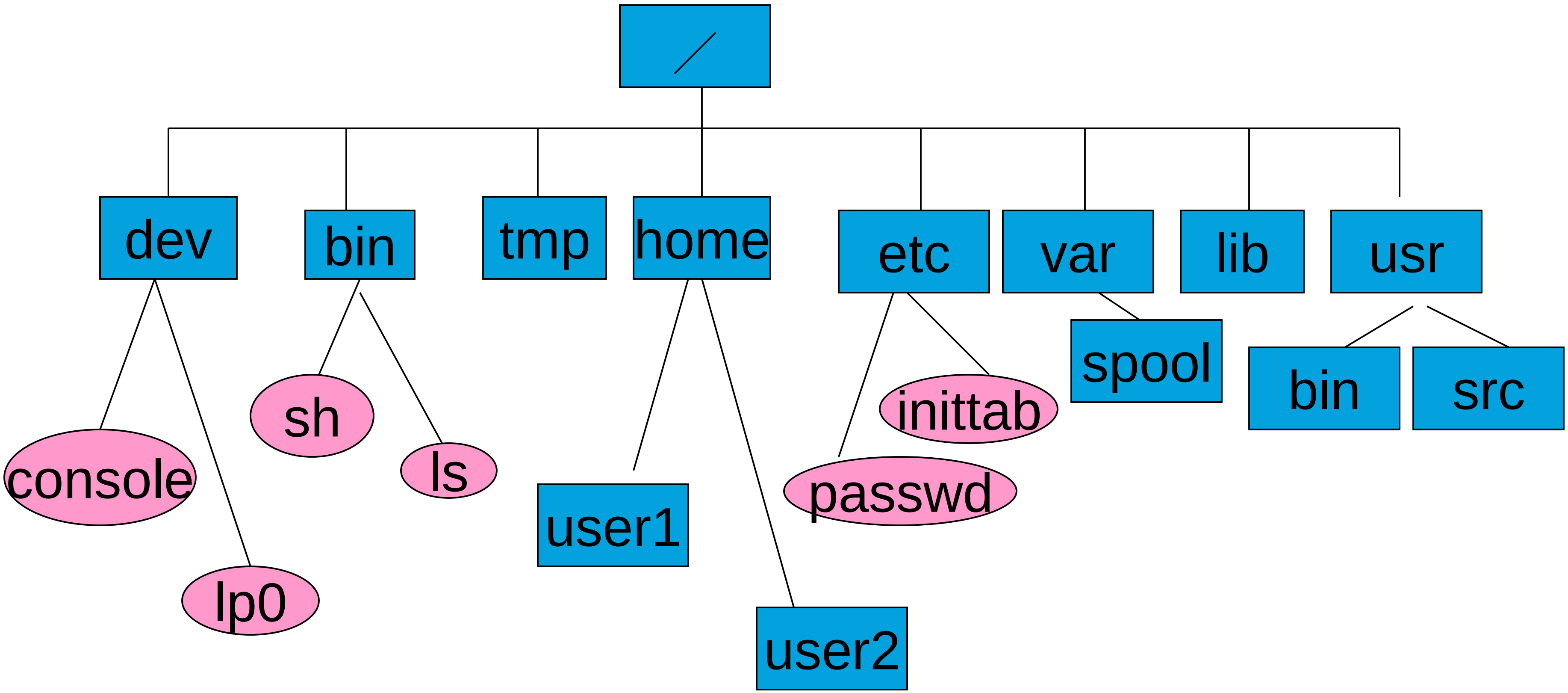
- Unix is a layered OS. The innermost layer is the hardware that provides the services for the OS.
-
- The OS, referred to as the Kernel interacts directly with the hardware and provides services to the user programs.
-
- User programs interacts with the kernel through a set of standard system calls.

File System

The Unix file system looks like an inverted tree structure. You start with the root directory, denoted by /, at top and work down through sub-directories underneath it.

- Every non-leaf node of the tree is called as a directory file.
- Every leaf node can either be a file, or an empty directory

File System (Contd.).



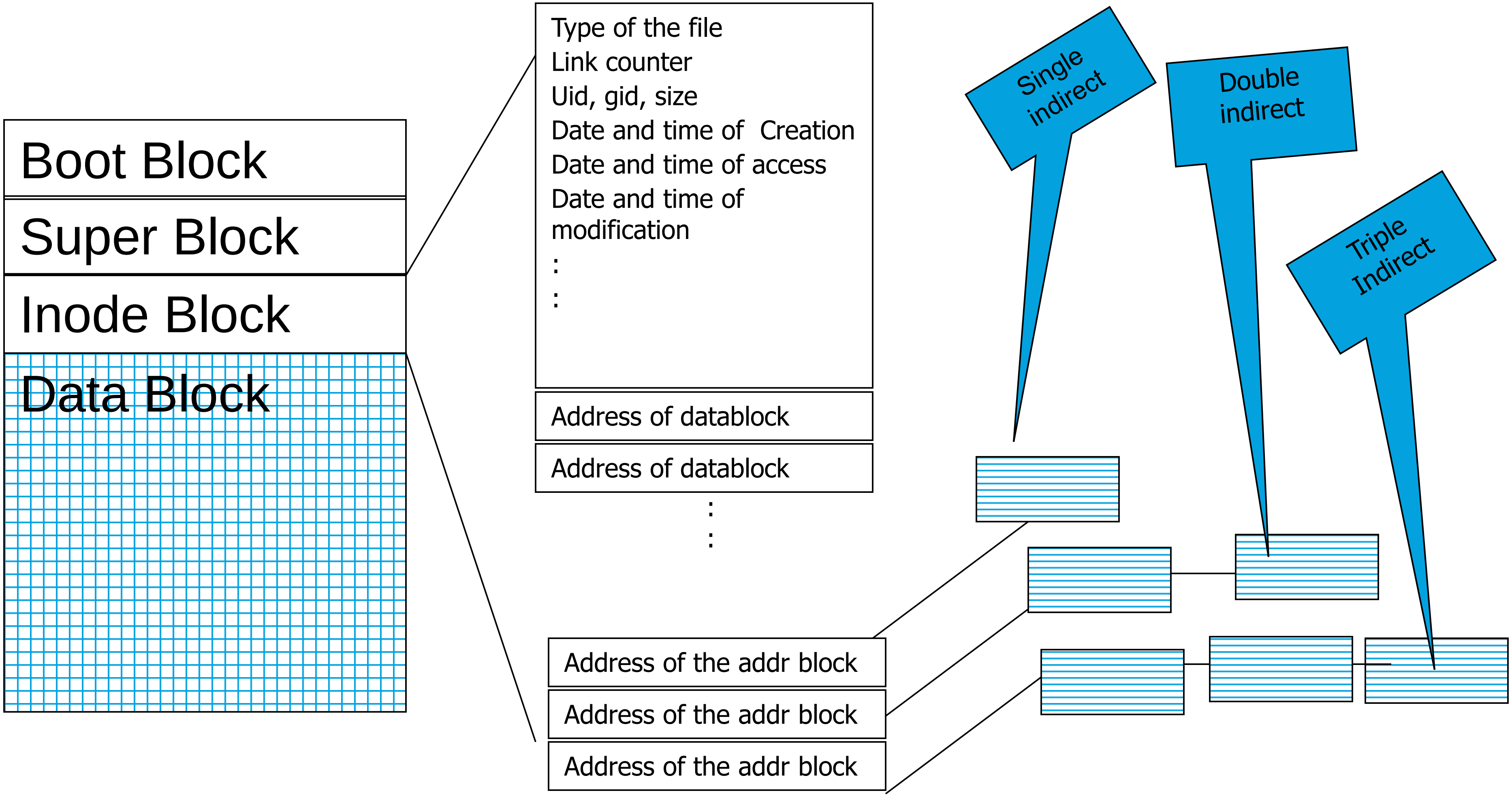
Unix File System - Directories

- / : Root directory. This is the parent of all the directories and files in the UNIX file system.
- /bin : Command-line executable directory. This directory contains all the UNIX native command executables.
- /dev : Device directory containing special files for character- and block-oriented devices such as printers and keyboards. A file called null existing in this directory is called the bit bucket and can be used to redirect output to nowhere.
- /etc : System configuration files and executable directory. Most of the administrative, command-related files are stored here.
- /lib : The library files for various programming languages such as C are stored in this directory.
- /home : Conventionally, all the user home directories are defined under this directory.
- /usr : This directory has a number of subdirectories (such as adm, bin, etc, and include. For example, /usr/include has various header files for the C programming language.

File System

- File system is the structure in which files are stored on disk
- File in UNIX is sequence of bytes organized in the form of blocks
- The size of each block is 512 bytes (depends on architecture)
- Block size can be decided while creating the file system structure

File System Structure



Boot Block

Super Block

Inode Block

Data Block

Type of the file
Link counter
Uid, gid, size
Date and time of Creation
Date and time of access
Date and time of modification
:
:

Address of datablock

Address of datablock

:
:

Address of the addr block

Address of the addr block

Address of the addr block

Common UNIX Flavors

BSD:	Berkeley, BSD
Solaris:	Sun Microsystems, Sys 5/BSD
Ultrix:	Digital Equipment Corporation, BSD
OSF 1:	Digital Equipment Corporation, BSD/sys 5
HPUX:	Hewlett-Packard, Sys 5
AIX:	IBM, Sys 5 / BSD
IRIX:	Silicon Graphics, Sys 5
GNU/Linux:	GNU, BSD/Posix

Types of UNIX Users

- Broad classification of users
  - root (most privileged)
  - Non-root (less privileged)

- Group
  - UNIX allows user IDs to be grouped
  - A single user ID can be member of multiple groups

- Differentiating users w.r.to file access
  - Owner
  - Group
  - Others

Getting Started

After connecting with a Unix System, a user is prompted for a login username, then a password. The login username is the user's unique on the system. The password is a changeable code known only to the user. At the login prompt, the user should enter the username; at the password prompt, the current password should be typed.

Note: Unix is case sensitive. Therefore, the login and password should be typed exactly as issued; the login, at least, will normally be in lower case.

Summary

In this session, you have learnt:

- The functions of OS
- The history of Unix
- The features of UNIX
- The Unix architecture
- File System
- Unix users
- Getting started

2. Unix Commands

Objectives

In this session, you will learn to:

- Use the basic Unix commands
  - pwd
  - date
  - who
  - ls
  - man
- Use “man” pages

Simple Commands

- pwd
  - Displays the current working directory.
- date
  - Displays the current date and time

Simple Commands (Contd.).

- who
  - Displays the names of all the users who have currently logged in
- who am i
  - Displays the name of the current user.

Listing the Directory Contents

- ls

Syntax :ls [options] [file….]

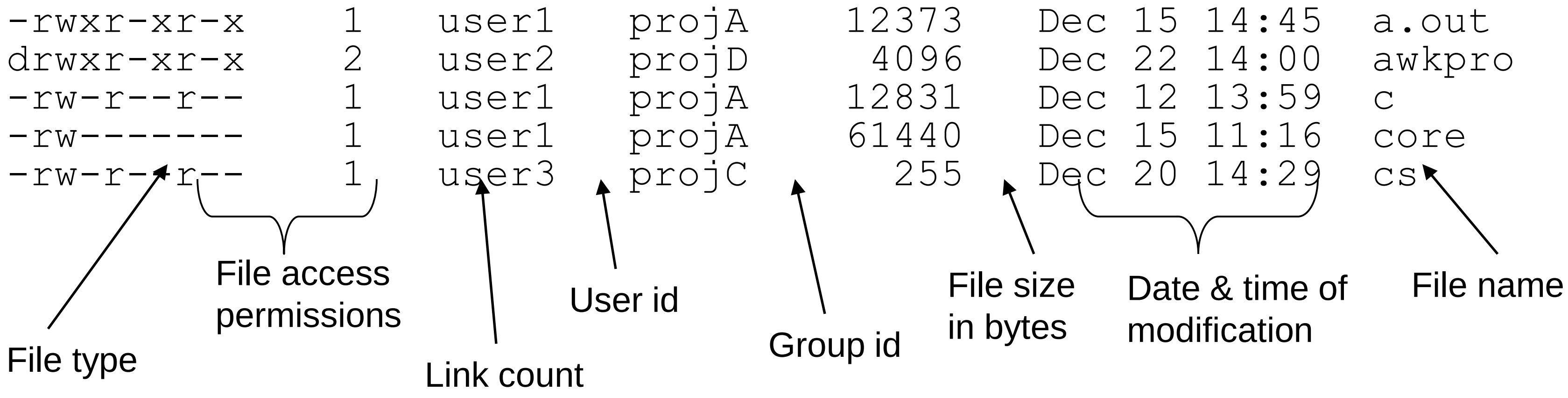
options: -l list in long format
-a list all files including hidden files
-i list inode no of file in first column
-s reports disk blocks occupied by file
-F mark type of each file
-C display files in columns
-R recursively list all sub directories

Meta Characters

*	Match with one or more characters or none	$ls –l *.c file*	abc.c , filepro
?	Match any Single character	$ ls –l file?	filea
[..]	Match with any single character with in the bracket	$ls –l file[abc]	filea, fileb , filec
;	Command seperator	$cat filea; date	displays the content of filea and displays the current date and time
|	Pipe two commands	$ cat filea|wc -l	Prints the number of lines of filea
()	Group commands, used when the output of the command group has to be redirected	$(echo “***x.c***”;cat x.c)>out	Redirects the content of x.c with a heading ***x.c*** to the file out

Listing the Directory Contents

```
 $ ls –l

 total 6
-rwxr-xr-x    1   user1   projA    12373   Dec 15 14:45  a.out
drwxr-xr-x    2   user2   projD     4096   Dec 22 14:00  awkpro
-rw-r--r--    1   user1   projA    12831   Dec 12 13:59  c
-rw-------    1   user1   projA    61440   Dec 15 11:16  core
-rw-r--r--    1   user3   projC      255   Dec 20 14:29  cs
```



File access permissions

User id

File size in bytes

Date & time of modification

File name

Group id

File type

Link count

Getting Help on Commands

- The Unix manual, usually called man pages, is available on-line to explain the usage of the Unix system and commands.

Syntax:

- man [options] command_name

Common Options

-k keyword	list command synopsis line for all keyword matches
-M path	path to man pages
-a show	all matching man pages (SVR4)

- help -–command_name– gives command synatx
- info command_name - help for commands

Summary

In this session, you have learned to …

- use the basic Unix commands like
  - pwd
  - date
  - who
  - ls
  - man
- use “man” pages

3. Files & Directories

Objectives

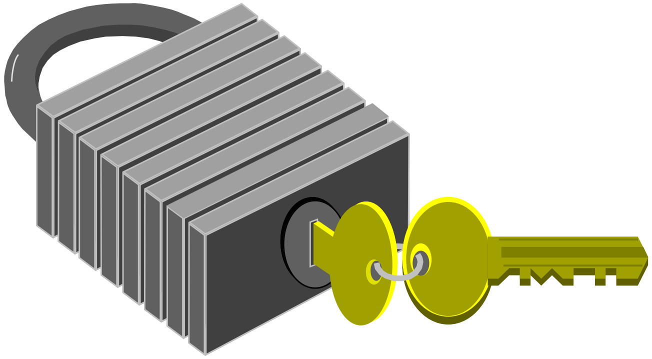
- In this session, you will learn to:
  - set file permissions using the chmod command
  - use directory-related commands namely mkdir, rmdir, cd commands
  - use file-related commands namely cp, mv, rm commands
  - create and edit files using the vi editor

File Access Permissions

- Refers to the permissions associated with a file with respect to the following

- Permission Levels
  - User (owner) (u)
  - Group (wheel, staff, daemon, etc.) (g)
  - World (guest, anonymous and all other users) (o)

- Permission Settings
  - Read (r)
  - Write (w)
  - Execute (x)



Changing Permissions - chmod

- chmod u+x file_name

Syntax:

chmod <category> <operation> <permission> <filename(s)>

or

chmod <octal number> filename

Octal Number

4 - for read

2 - for write

1 - for execution

$ chmod 744 xyz

this sets read, write and execute permissions for owner, read permission for group and others

Directory Creation

Command Syntax

```
    mkdir [OPTION] DIRECTORY
$ mkdir <path>/<directory>
$ mkdir –m <directory>
$ mkdir –p <directory1>/<directory2>/<directory3>
```

Example:

```
      $ mkdir  project1
```

This creates a directory project1 under current directory

Note: Write and execute permissions are needed for the directory in which user wants to create a directory

Directory Removal

rmdir command removes directory

Syntax

- rmdir <directory name>

Example

Removes project1 directory in the current directory

- rmdir project1

Remove multiple directories

```
rmdir  pos1 pos2
```

Remove the directory recursively

```
rmdir –p dir1/dir2/dir3
```

rmdir removes a directory if it is empty and is not the current directory

Command – cd

cd command is used to change the directory

- cd   - take to the home directory
- cd ..   - takes to the parent directory
- cd /   - takes to the root directory

File-Related Commands

File Operation	Command
Copying a file	cp
Moving a file	mv
Removing a file	rm
Displaying a file and concatenating files	cat

Command – cp (Contd.).

Used to copy files across directories

Syntax

```
cp <source file> <new file name>
```

Example

```
cp file1 file2
```

Command – cp (Contd.).

Options to cp

- -p
    - Copies the file and preserves the following attributes
        - owner id
        - group id
        - permissions
        - last modification time

  -r

    - recursive copy; copy subdirectories under the directory if any

    -i

    - interactive; prompts for confirmation before overwriting the target file, if it already exists

Command - mv

Used to move a file, or rename a file

Preserves the following details

- owner id
- group id
- permissions
- Last modification time

-f   suppresses all prompting (forces overwriting of target)

-i   prompts before overwriting destination file

Command - rm

Used to remove a file

- Syntax : rm file(s)

-f   suppresses all prompting

-i   prompts before deleting destination file

-r   will recursively remove the file from a directory (can be used to delete a directory along with the content)

Caution: Use “i” option along with “r” to get notified on deletion

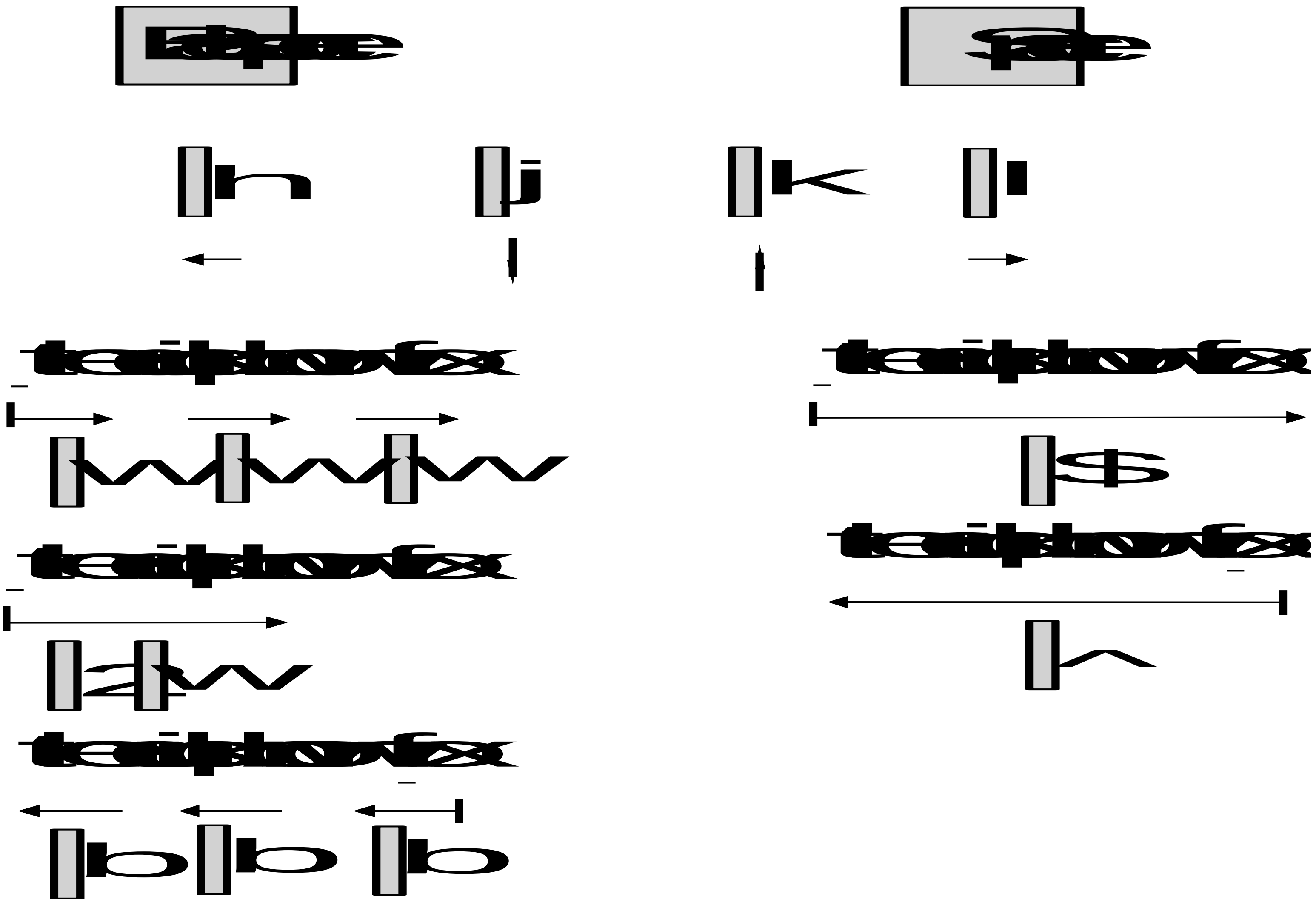
Command – chown & chgrp

```
$ ls –l
-rwxr-xr-x    1 user1   training    12373 Dec 15 14:45 a.out
-rwxr-xr-x    3 user1   faculty      4096 Dec 24 11:56 awkpro

$chown user2 a.out

$ls –l
-rwxr-xr-x    1 user2   training    12373 Dec 15 14:45 a.out
-rwxr-xr-x    3 user1   faculty      4096 Dec 24 11:56 awkpro

$ chgrp training awkpro

$ls –l
-rwxr-xr-x    1 user2   training    12373 Dec 15 14:45 a.out
-rwxr-xr-x    3 user1   training     4096 Dec 24 11:56 awkpro
```

The Unix Text Editor - vi editor

- vi is a visual editor used to create and edit text files.
  - A screen-oriented text editor
  - Included with most UNIX system distributions
  - Command driven
- Categories of commands include
  - Cursor movement
  - Editing commands
  - Search and replace commands
- The vi editor is invoked by the following command:
  - $ vi filename

Navigation



Editing Commands

Text insertion / replacement

- i - inserts text to the left of the cursor
- a - inserts text to the right of the cursor
- I - inserts text at the beginning of the line
- A - appends text at end of the line
- o - opens line below
- O - opens line above
- R - replaces text from cursor to right
- s - replaces a single character with any number of characters
- S - replaces entire line

Editing Commands (Contd.).

Deletion

- x - to delete character at cursor position
- 3x - to delete 3 characters at cursor position
- dw - to delete word
- 2dw - to delete 2 word
- dd - to delete a line
- 2dd - to delete 2 lines

Editing Commands (Contd.).

- Yanking
  - Y - copy line into buffer
  - 3Y - copy 3 lines into buffer
  - p - paste buffer below cursor
  - P - paste buffer above cursor
  - u - undo
- Save and quit
  - :w - to save
  - :w! - to name a file (:w! filename -> save as)
  - :x - save and quit
  - :q - cancel changes
  - :q! - cancel and quit

Search & Replace Commands

Commands for vi editor in Linux:

/pat searches for the pattern pat and places cursor where pattern occurs.

/ repeat last search

:%s/old/new/g to change every occurrence in the whole file.

:#,#s/old/new/g where #,# should be replaced with the numbers of the two lines (say between line no.'s 2 and 5).

Example - :**2,5s/am/was/g**

Summary

In this session, you have learned how to:

- use file permissions using the chmod command
- use directory-related commands namely mkdir, rmdir, cd commands
- use file-related commands namely cp, mv, rm commands
- create and edit files using the vi editor

User Management

Objectives

In this session, you learn about:

- Creating users/groups
- Modifying users/groups
- Deleting users/groups

User Management

User ID and password is created by the system administrator using commands / GUI Tool.

- Command to create a user

  useradd

  adduser
- Command to create a group

  groupadd

  addgroup
- Command to modify a user & group
  - usermod
  - groupmod
- Command to delete a user & group
  - userdel
  - groupdel
- Command to set/reset password

  passwd <username>

passwd

- To change a user password
  - $ passwd
  - Changing password for username
  - New UNIX password:
  - Reenter UNIX password:

User Related Files

/etc/passwd This file stores all the user information except the password

/etc/shadow This file stores the encrypted passwords and all password related information

/etc/group This file stores all the groups in the system along with the secondary group members

/etc/passwd

Find below the first two lines of an /etc/passwd file from a linux system

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

Fields in the /etc/passwd file

loginID user's login name

x represents a placeholder for the user's encrypted password

UID user ID number used by the system to identify the user

GID GID number which identifies the user's primary group

comment comment about the user – normally a user's full name

home_directory specifies the full path name to the user's home directory

login_shell defines the user's login shell

/etc/shadow

Find below the first two lines of an /etc/shadow file from a linux system

```
root:5RiJS.yvdGBkU:13255:0:99999:7:::
bin:*:13255:0:99999:7:::
```

Fields in the /etc/passwd file

loginID user’s login name

encrypted_passwd encrypted password

lastchg the number of days between Jan 1, 1970 and the last password modification date

min minimum no. of days between password changes

max maximum number of days the password is valid

warn number of days the user is warned before the password expires

inactive number of inactive days allowed for the user before the account getting locked

expire The date when the user account expires

reserved field For future use

/etc/group

Find below the first two lines of an /etc/group file from a linux system

```
root:x:0:root
bin:x:1:root,bin,daemon
```

Fields in the /etc/passwd file

goupname name assigned to the group

group-passwd allows a non group member to work as group member on supply of this password

gid group id

userlist list of user names which represent the user’s secondary group membership

su **command**

- The su command is used to switch the user to another user
- System administrator should not login as “root” instead login as a normal user and run su command to switch user

Command

su	Profile will not change
su –	Profile will change to the root’s
su - <username>	Switch to a user with profile change

- **id**

$ id

Summary

- In this session, you have learned how to:
  - Create users/groups
  - Modify users/groups
  - Delete users/groups

References

- 1. Richard Petersen, Linux The Complete Reference. Ed 6. New Delhi: McGraw-Hill Education India, 2007.
- 2. Eric Foster-Johnson, John C. Welch, and Micah Anderson, Beginning Shell Scripting, Indianapoli: Wiley Publishers Inc, 2005.
- 3. Sander Van Vugt, Beginning the Linux Command Line, Wrox Press, 2005.

Thank you