School of Computer Science Engineering and Technology
Bennett University (The Times Group)

Program- BTech-3rd Semester	Type- Sp. Core-I
Course Code- CSET213	Course Name-Linux and Shell Programming
Year- 2024	Semester- Odd
Date- 28/10/2024	Batch- Cyber Security

Lab Assignment 9

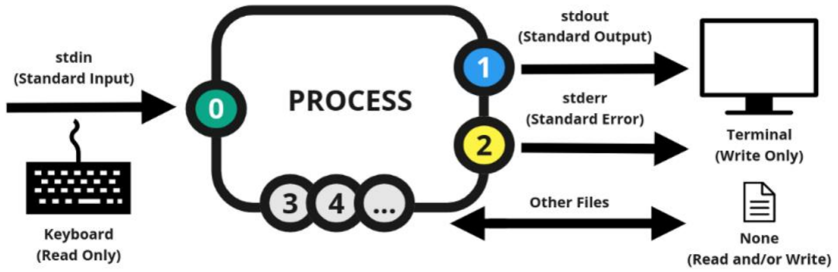
Exp No	Name	CO1	CO2	CO3
9	Shell programming File handling	✓	✓	

Objective: To use shell scripting for processing the files, and user management.

Outcomes: After executing this assignment, the students will be able to

1. understand the file creation,
2. reading and writing data from/into files,
3. modifying file permissions,
4. sharing files
5. user management (adding, modifying, and deleting users)

Hands-on Learning on operators (40 minutes)



1. **Shell I/O redirection:** There are three redirectors to work with: >, >>, and <. The following information describes each one:
   a. Redirection with >
      i. command > file: Sends standard output to <file>
      ii. command 2> file: Sends error output to <file>
      iii. command 2>&1: Sends error output to standard output
      iv. command > file 2>&1: Sends standard output and the error output to a file
      v. command &> file: Sends standard output and the error output to a file
      vi. command 2>&1 > file: Sends error output to standard input and the standard input to a file
   b. append with >>
      i. command >> file: Appends standard output to a file
      ii. command 2>> file: Appends error output to a file
      iii. command >> file 2>&1: Appends standard output and error output to a file
      iv. command &>> file: Appends standard output and error output to a file
      v. command 2>&1 >> file: Sends error output to standard input and appends standard input to a file
   c. Redirect with <
      i. command < input: Feeds a command input from <input>

ii. command << input: Feeds a command or interactive program with a list defined by a delimiter; this is known as a here-document (heredoc)

iii. command <<< input: Feeds a command with <input>; this is known as a here-string

2. **Reading and writing data from/into files:** Using read and redirection operators
   Example1: Script to read file character by character

```
#!/bin/bash

read -p "Enter file name : " filename

while read -n1 character

do

echo $character

done < $filename
```

   Example2: Script to read file line by line

```
#!/bin/bash

read -p "Enter file name : " filename

while read line

do

echo $line

done < $filename
```

3. **Files Sharing:** when you create a new file, Linux assigns the file permissions of the new file using your default UID and GID. To allow others access to the file, you need to either change the security permissions for the everyone security group or assign the file a different default group that contains other users.
   Use $chmod permission filename

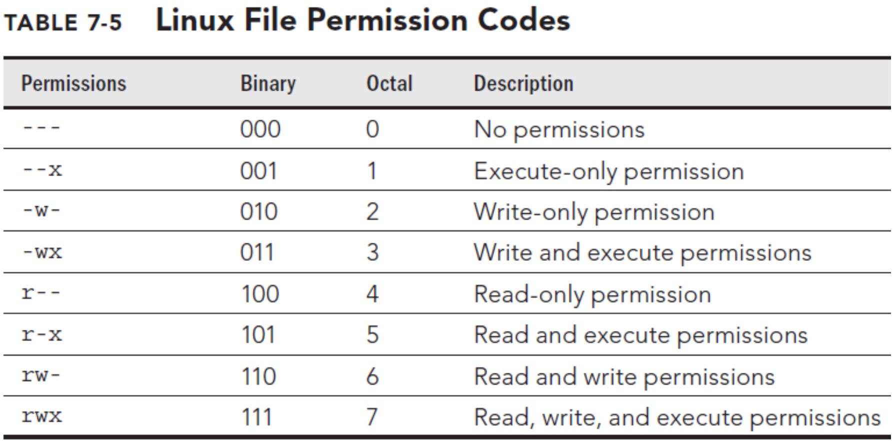


TABLE 7-5 Linux File Permission Codes

Permissions	Binary	Octal	Description
`---`	000	0	No permissions
`--x`	001	1	Execute-only permission
`-w-`	010	2	Write-only permission
`-wx`	011	3	Write and execute permissions
`r--`	100	4	Read-only permission
`r-x`	101	5	Read and execute permissions
`rw-`	110	6	Read and write permissions
`rwx`	111	7	Read, write, and execute permissions

4. **User Management:** using commands- id, useradd, passwd, userdel, usermod
   Example:

```
# for u in bob joe ; do
useradd $u
echo '$u:Password1' | chpasswd
passwd -e $u
      done
```

5. **File Handling API:** The File Handling API enables web apps to register their ability to handle file types with the Linux. This enables the file manager of the Linux or other OS flows to use the web app to open the file, just like with a native app.

Scripting Problems for Assessment (30 Minutes)

1. Write a shell script to create the users Ram, Shyam, Swasti, Bhoomi, Nutan, Virat, Aayush, Udbhav, Yashi, Kulwinder, Rahim, Bob, Alice using for loop and store the information of these users in the file /etc/shadow. Also print the content of /etc/shadow. (15 Minutes) **(1 Mark) (Even)**
2. Write a script that counts the total number of lines across a set of files. **(1 Mark) (Odd Batch)** (15 Minutes)
3. Write a shell script to list all files in your home directory with their inodes and delete a file using its inode number. **(1 Mark) (Odd Batch) (15 Minutes)**
4. Write a shell script to change the permissions on the file created in Q1 as read, write, and execute (7) permission for owner, read, and execute permission (5) for group and read permission for others. **(15 Minutes)** (1 **Mark) (Even Batch)**

Submission Instructions:

1. Submission requires the screen shots of all the incurred steps to execute a shell script or a video showing the whole process.
2. All these files are in single zip folder.
3. Use the naming convention: Prog_CourseCode_RollNo_LabNo.docx (Example: BCA3rdSem_CBCA221_ E21BCA002_Lab1.1)
4. Submission is through LMS only