Concept: Basic commands in Unix, Filters, pipe

Objective: At the end of the assignment, participants will be able to run basic Unix commands and implement the concepts of Pipes and Filters.

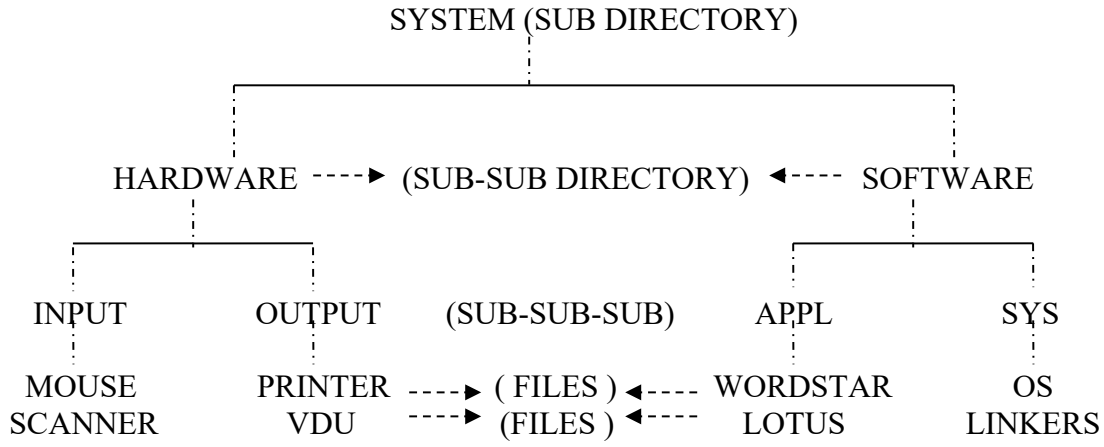
Problems :

Section 1 :

1. List all the files and directories of /bin with detail information from your current directory.
2. List all the files including hidden files in your parent directory.
3. List all the files starting with letter 'r' in your current directory.
4. List all the files which are three characters long in your current directory.
5. List all the files with extension .doc in your current directory.
6. List all the files starting with a range of letters from 'l' to 's' in your current directory.
7. List only the directory files in your current directory.
8. Create a file text1 by taking the input from keyboard.
9. Copy the contents of file text1 to another file text2.
10. Append the contents of file text2 to file text1.
11. Count the number of files in the current directory.
12. Display the output of command ls –l to a file and on the output screen.
13. From a sample text file print all lines starting from 10th line.
14. Find number of users currently working on the system.
15. Find out whether itp9 has been logged in or not?
16. Delete all the files starting with tmp.
17. Count the total number of words in file text1.
18. Create the file employee.txt having colon (:) separated fields. The fields of the record are: enumber, ename, eposition, esal, edoj, edept. Answer following queries:
    1. List all the employees working on project "CSS"
    2. List all the employees working as a SDE
    3. List all the employees joined before 1999
    4. List all the employees along with a row number
    5. Sort the file as per the names
    6. List top three salaried employees
    7. Remove duplicate records from the file
    8. List all the employees except "PL"s.
    9. List all the employees , except working as a Clerk.
    10. List dept. nos along with no. of employees working in each dept.
    11. Sort the file in descending order of salary.

Section 2 :

1. Display your current working directory.
2. Create following directory structure under your Home directory,



3. List all the files and directories of Hardware with detail information by remaining in HOME directory.

4. List all the files, directories, subdirectories along with their contents of HOME directory by remaining in SYS.

5. Copy file SCANNER under SYSTEM remaining in APPL.

6. Change the name of that file to SCAN.

7. Append some contents to the file SCAN.

8. List all the hidden files.

9. List all the files which are three characters long.

10. List all the files starting with a range of letters from L to S.

11. List all the files which are not starting with a range of letters from L to S.

12. List all the files starting with letter S.

13. List the contents of Is command page wise.

14. Display a long listing of only the directory files in your home directory.

15. List how many users currently logged on to the system.

16. Increase the no. of links to the file FILE2 TO 3. Check the inode numbers and link display for these files.

17. Using one single command display the output of WHO and PWD commands.

18. Display date in following format,

Today is Friday , 17 May 96

19. Display the following message on the monitor screen.

Please give me $ 100

20. Display the following message on the monitor.

The long listing of my home dir …… is ……

(Hint : Use ls – l and pwd commands)

21.Assign variable Black to var1. Display the following message on the monitor.

Black belt is associated with karate.

22.Display the following messages in enlarge mode.

WELCOME TO UNIX
Current working directory.

23.Remove the directory structure that you have created.

Concept: Shell Scripting

Objective : At the end of the assignment, participants will be able to understand and implement shell scripting.

Problems :

1. Accept a login name from the user and check if the user has logged in.
2. Write a shell script which will generate the O/P as follows

*
**

***
****

3. Display the name and count of the directories in the current directory.
4. Write a shell script which will accept three nos. from the keyboard and will find out the largest of three.

   Note : Do not use logical operator -a for comparison.
5. Write a shell script which will calculate the factorial of a number entered from the keyboard.
6. Accept the first name, middle name, and last name of a person in variables fname, mname and lname respectively. Greet the person (take his full name) using appropriate message.

7. Accept a file name and a number (x). Display x lines from the top of the file. Check if the file exists and is readable. The value of x should not exceed the total number of lines in the files. Display suitable messages in case an error is encountered.
8. Write a shell script to modify the permissions of all the files in a given directory.

9. Display the name of files in the current directory. Also display the name of the file with maximum size and minimum size in the current directory. The size is considered in bytes.
10. Write a script which when executed checks out whether it is a working day or not?
11. Write a script that accept a member into some health club, if the weight of the person is more than 30kg or less than 250 kg.
12. Write a shell script that greets the user with an appropriate message depending on the time that the user logs into the system.
13. Write a shell script to copy one file to another.
    Accept 2 filenames from the keyboard. Filenames should not be NULL.
    If source file is directory and destination is directory, give appropriate error.
    If source file exist , ordinary and destination file exist , ordinary. Prompt user before copying source file to destination file.
    If source file exist , ordinary and destination file does not exist. Copy source to destination.
    If source file exist , ordinary and destination file exist, directory. Copy source under destination directory file.
    If source file exist , directory and destination file exist, directory. Copy source directory under destination directory file. After copying, flash successful copy message to user.
14. Write an interactive file-handling program. The user must be allowed to copy, move, or rename operations, depending on the choice entered. Generate the appropriate error messages.

15. Write a menu program which displays the following options :

    1. Make a file.
    2. Display contents
    3. Copy the file

4. Rename the file
5. Delete the file
6. Exit

Enter your option.

If the user selects option 1, accept a file name from the user. If the file exists, then display an error message pass the control to the menu. If the file does not exist, then allow the user to enter some data. Pressing ^D would save the contents and display the menu.

If the user selects option 2, then accept a file name from the user. If the file exists, then display the contents of the file. If the file does not exist, then display suitable error message. After this process, display the menu to accept another option.

Selecting Option 3 allows the user to accept the source file and target file. If the source file exists and is readable, then accept the target file name. If the source file does not exist, then display suitable error message. If the target file does not exist, then copy the contents of the source file to the target file. If the target file exists, then display suitable message and go back to the menu.

Option 4 is similar to option 3 but rename the file instead of copying.

Selecting option 5 allows the user to enter a file name. If the file exists, then check to see if it is writable. If so, then delete the file with confirmation from the user. If the file does not exist, then display suitable error message.

16. Accept roll number, name, marks in sub1, sub2 and sub3. Calculate total, percentage, grade & class and enter the record in a file “student”. The marks are out of 100 in each subject. Allow the user to enter any number of records.

Grade is found out as follows :

Percentage Grade

< 35 Fail

>= 35 & < 50 Third

>= 50 & < 60 Second

>= 60 & < 75 First

>= 75 Distn

If the student secures < 35 in any one of the subjects, then class = “fail”. Otherwise class=”pass”.

17. Accept roll number from the user at the command line. Search it in the "student" file. If it is present, then display name, percentage, grade and class of the student. If the roll number is not present, then display a message "not found". Allow the user to enter any number of queries.
18. Accept roll number from the user. Search it in the file. If the roll number is present, then allow the user to modify name and marks in 3 subjects. If the roll number is not present, then display a message "not found". Allow the user to modify one field at a time. Get the confirmation from the user before modifying a field.

19.Accept the roll number from the user. Search it in the file. If it is present, then delete the respective record from the "student" file. Get the confirmation from the user before deleting the record from the file. If the roll number is not present, then display suitable error message.

20.Write a program which accepts a password from the user. If the password is correct, then invoke the menu program. If the password is incorrect, then give him two more chances. If after two chances, the user is unable to enter the correct password, then display suitable message and exit from the program.

21.Write a shell script that takes 3 command line arguments. The first argument is the name of a destination file, and the other two arguments are names of files to be placed in the destination file. Copy the contents of files to the destination file.

22.Write a script that takes a command line argument and reports on its file type (regular file, directory file, etc.). For more than one arguments generate error message.

23.Display a list of files that have read, write or execute permissions depending on the choice entered on the command line.

24.Write a script that will take in a directory name at the command line, counts the number of subdirectories and the number of ordinary files in it. If the user does not enter a directory name then the script should terminate after displaying an appropriate message.

25.Write a program that will accept even number of filenames on the command line. It should then copy the first file in the second, third file in the fourth, and fifth file in the sixth and so on. The program should check for appropriate conditions before carrying out the copy operation.

26.Enhance program #1 to ensure that the new roll number being added in the file does not exist in the file. The marks in the 3 subjects should be from 1 to 99. In case of an error, then display a suitable message and ask the user to re-enter the marks. Also check whether the file exists and is writable before writing the record in the file. If the file does not exist or is not writable, then display suitable error message.

27.Modify program #2 to execute it using command line arguments. Accept the roll number to be searched along with the program name at the command line.

28.Create a data file “details” and enter some roll number in the file. Write a program which matches the roll number from the details file with the “student” file. If the roll number matches then copy the record from the student file in “valid” file. If the roll number does not exist, then write a record which comprises of roll number and a message ”not found” in “invalid”. If either “valid” or “invalid” file does not contain any record at the end of the program, then display a suitable error message to notify the user.

29.Add some records in the “student file” manually. Write a program which does the following

1. If the roll number is already existing, then write the record followed by a message “roll number exists” in “log1” file.
2. If the marks in the subjects is not in the range of 1 – 99 then add such a record followed by a message “marks out of range” in “log1” file
3. If the data is valid, the calcuate total, percentage, grade and class as specified in program #1 and update the student file.

30.Accept a grade along with the program name at the command line and display the records of all students who have secured the specified grade. If there are no students who have secured the specified grade then display a message “There are no students with this grade”. If the grade has been specified incorrectly, then display a message “grade is incorrect”. Also check to see that only 1 argument has been supplied at the command line.

31. Create a data file “master” and enter some records manually. The record comprises of batch code, faculty name and number of students. Enter number of students as 0 initially which typing the record. Write a data entry program which accepts the batch code from the participant. Search this batch in the “master” file. If it is present, then allow the user to enter any number of records. Ensure that the records are written in a file which has the batchcode as the filename. Consider the data as specified in program #1 above. Take care to see that the roll number is not duplicated and the marks in each subject are entered in the range from 1 to 99. Once is user has finished entering the records in the file, update the number of students field in “master” file with the total number of participants in the respective batch code file.

32.Accept the batch code along with the program name at the command line and display the name and the rank of the top three rank holders. In case of a tie, then display all the names.

33.Create a library of shell functions to do the followings:
Function to concatenate 2 strings. Pass 2 strings as parameters to function.
Function to find out the length of a given string, Pass string as a parameter to function.
Function to compare the two strings. Pass 2 strings as parameters to function.
Function to check if string is palindrome or not. Pass string as a parameter to function.

Function to print given string in reverse order. Pass string as a parameter to function.

34.Consider a data file “names” with two fields : name, dept and code. Enter some records in the file manually. Enter dept as “mkt”, “per” or “sys”. Enter the code as 0 for all the records. Consider the following coding scheme. Alphabets a to z are having the codes from 1 to 26. For each name, add the code of each alphabet of the name and update the code field of “names” file with the result. Ensure that the names are entered in lower case. In case, the data is entered in upper case, then convert the data into lower case before processing the file.

35.Write a program which sorts the file department-wise (ascending), code-wise (descending) and changes the department name “sys” of “names” file to “edp”.

36.Write a program which prints one line description of all

- Linux commands
- System calls
- Library API calls

(Hint: Man pages of each of the above categories are in a separate directory.)

Concept: Awk Scripting and Case Study

Objective : At the end of the assignment, participants will be able to understand and implement awk scripting and complete a case study by implementing the concepts of Shell Scripting learned in this course.

Problems :

1. Consider the results are stored in following format:

EmpID	Name	Subject	Marks(/50)
E001	Nilesh	Unix	30
E002	Nilesh	DSA	20

Like these you have 10 records (5 of DSA and 5 of Unix)

Calculate the avg score secured in Unix and DSA and the first 2 topers in Unix and DSA each.

2. Write a script to get the report of the users logged on to the System in the following formats. (Records should be sorted on logging time.)

   Header must include company name and Date

   Records in the format

Username Logged-in-time Terminal

Tailor should include total number of the users logged in.

3. Consider a text file containing the records (colon separated fields) in the format:
   EmpName:EmpId:Subject:ObtMarks:TotMarks:Result

   Write a script to get the result of "UNIX" Subject in the format (Considering the data file has TotMarks=50 for UNIX)

   EmpName:ObtMark:MarksOutof35

   The header of the report must contain total marks and the tailor must specify the percentage result for that subject.

   Also generate another summary result containing total number of participants appeared, total number of participants passed, and Name of the participants ranked Ist IInd, IIIrd with their total score.

4. Create an emp_mast and dept_mast files containing following details,

   empno:name:job:deptno
   deptno:deptname
   empno:basic:hra:conveyance:medical:prof. tax: PF:TDS

   Generate the pay slip as shown below by passing the empno as parameter to the awk script.

```
Empno           1001
Deptno          10
Job             MGR
Curency         INR
-------------------------------------------------------------------------------------------
Earnings
-------------------------------------------------------------------------------------------
BASIC           10,000.00
HRA              4,000.00
CONVEYANCE       9,600.00
MEDICAL          5,000.00
-------------------------------------------------------------------------------------------
Total Earnings :    28,000.00
-------------------------------------------------------------------------------------------
Deductions
-------------------------------------------------------------------------------------------
PF               1200.00
PROF_TAX          200.00
TDS              1600.00
-------------------------------------------------------------------------------------------
```

Net Pay : 25000.00

Note : Check empno before passing to awk script.

5. Generate the report as given below by passing the deptno to awk script

Deptno : 10 Dname : Admin

Empno	Name	Job
1001	Ketan	MGR
1002	Sachin	CLERK

Note : Check deptno before passing to awk script

6. With respect to question no 16 in previous assignment write an awk script which displays a report of all the participants in a suitable format. At the end of the report, display the following information :

(a) Total number of participants.

(b) Total number of participants who have passed.

(c) Total number of participants who have failed.

(d) Total number of participants who have secured distinction.

7. Write an awk program which prints the record followed by the length of each record of "names" file. (record [length]). At the end, display the maximum and minimum length of the record.

8. A log file contains tags to delimit different information. We are interested in all the lines from tag "<workorder>" to tag "</workorder>", if the keyword "customer" is present on any of these lines. The line containing the starting tag and the line containing ending tag should be extracted alongwith all lines in between to a separate file, only if the keyword is present in this set of lines. The tag set may occour multiple times in our large file.

What would happen if we use a shell script containg grep, etc, instead of this awk script?

Case Study

Write a menu driven program for Store Mgt using shell script:

MENU-LIST

2. Add Record
3. Delete Record

4. Modify Record
5. Generate Report
6. Exit

1. Add record option contains three sub options:
    a. Add New Item
    b. Add an Order
    c. Back to main menu

(a) Write a shell script to add new item detail in “ITEM.DAT” file. While adding record in file make sure that:

Item Number should be numeric and unique.

Item name should not be null

Total Quantity should be numeric

Reorder Value should be numeric and greater than zero.

(b) Write a shell script to add a new order in “ORDER.DAT” file. While adding record in file, make sure that:

Order Number should be numeric and unique

Item Number should be numeric and should exist in “ITEM.DAT”

Quantity ordered should be numeric, greater than zero and less than the quantity in “ITEM.DAT” for that item.

2.Delete record option contains three sub options :

a. Delete an Item

b. Delete an Order

c. Back to main menu

(a)Accept an item no from the user which should be numeric and not null. Delete a record from “ITEM.DAT” and all related records from “ORDER.DAT” accordingly.

(b) Accept an order no from the user which should be numeric and not null. Delete a record from “ORDER.DAT” and update the related record in “ITEM.DAT” accordingly.

3.Accept an order no from the user which should be numeric and not null. If this order no exists in "ORDER.DAT", accept new values for the other fields of that order and update it.

4.Generate a report in the following format :

ITEM ORDER DETAILS

Item Number : _____ Item Name :_____

Quantity Available : _____

Order Number	Quantity Ordered	Order Date
_____	_____	_____
_____	_____	_____

Total Quantity Ordered: _____