Unix Programming Lab Choice Based Credit System

Laboratory Experiments:

(a) Explore Unix Environment.

(b) Explore vi- editor with Vim tutor. Perform the following operations using vi editor, but not limited to:

1. Insert character, delete character, replace character.
2. Save File and continue working.
3. Save File and exit editor.
4. Quit the editor.
5. Quit without saving the file.
6. Rename a file.
7. Insert lines, delete line.
8. Setline numbers.
9. Search for a pattern.
10. Move forward and backward.

1a. Write a shell script that takes a valid directory name as a argument recursively descend all the sub-directors, find the maximum length of any file in that hierarchy and writ the maximum value to the standard output.

1b. Write a shell script that accepts a path name and creates all the components in that path name as directories. For example, if the script is named as mpc, then the command mpc a/b/c/d should create sub-directories a, a/b, a/b/c, a/b/c/d.

2a. Write a shell script that accepts two filenames as arguments, checks if the permissions for these files are identical and if the permissions are identical, output common permissions otherwise output each filename followed by its permissions.

2b. Write a shell script which accepts valid log-in names as arguments and prints their corresponding home directories, if no arguments are specified, print a suitable error message.

3a. Create a script file called file properties that reads a filename entered and outputs it properties.

3b. Write a shell script to implement terminal locking (Similar to the lock command). It should prompt for the user for a password. After accepting the password entered by the user, it must prompt again for the matching password as confirmation and if match occurs, it must lock the keyword until a matching password is entered again by the user. Note the Script must be written to disregard BREAK, control-D. No time limit need be implemented for the lock duration.

4a. Write a shell script that accept one or more file names as argument and convert all of them to uppercase, provided they exists in current directory.

4b. Write a shell script that displays all the links to a file specified as the first argument to the script. The second argument, which is optional, can be used to specify in which the search is to begin. If this second argument is not present, the search is to begin in the current working directory. In either case, the starting directory as well as its subdirectories at all levels must be searched. The script need not include error checking.

5a. Write a shell script that accepts filename as argument and display its creation time if file exist and if does not send output error message.

5b. Write a shell script to display the calendar for the current month with current date replaced by * or ** depending whether the date is one digit or two digit.

6a. Write s a shell script to find a file/s that matches a pattern given as command line argument in the home directory, display the contents of the file and copy the file into the directory

~/mydir.

6b. Write a shell script to list all the files in a directory whose filename is at least 10 characters. (use expr command to check the length).

7a. Write a shell script that gets executed and displays the message either “Good Morning” or “Good Afternoon” or “Good Evening” depending upon time at which the user logs in.

7b. Write a shell script that accepts a list of filenames as its argument, count and report occurrence of each word that is present in the first argument file on other argument files.

8a. Write a shell script that determine the period for which as specified user is working on a system and display appropriate message.

8b. Write a shell script that reports the logging on of as specified user within one minute after he/she login. The script automatically terminates if specified user does not login during specified in period of time.

9a. Write a shell script that accepts the filename, starting and ending line number as an argument and display all the lines between the given line number.

9b. Write a shell script that folds long lines into 40 columns. Thus any line that exceeds 40 characters must be broken after 40th, a “/” is to be appended as the indication of folding and processing is to be continued with the residue. The input is to be supplied through a text file created by the user.

10a. Write an awkscript that accepts date argument in the form of dd-mm-yy and display it in the form month, day and year. The script should check the validity of the argument and in the case of error, display a suitable message.

10b. Write an awkscript to delete duplicated line from a text file. The order of the original lines must remain unchanged.

11a. Write an awk script to find out total number of books sold in each discipline as well as total book sold using associate array down table as given below.

Electrical	34
Mechanical	67
Electrical	80
Computer Science	43
Civil	98
Mechanical	65
Computer Science	64

11b. Write an awkscript to compute gross salary of an employee accordingly to rule given below.

If basic salary < 10000 then HRA=15% of basic & DA=45% of basic.

If basic salary is >=1000 then HRA=20% of basic & DA=50% of basic.