Laboratory Assignment 2 Subject: Design Principles of Operating Systems

Subject code: CSE 3249

**Assignment 2: Familiarization with basic Commands in Unix Operating System and Shell Programming**

**Objective of this Assignment:**

* To learn basic concepts of shell programming
* To lean concept of command line argument in shell script.

1. Write a shell script named as **prog** for merge the content of files a.txt, b.txt, and c.txt sort them and save the result in a file called **result** and display the sorted output on the screen.

(Note: a.txt, b.txt and c.txt file contain some numerical value. Make the script an executable file and run it as a command using its name only.)

#!/bin/bash

# Merge the content of a.txt, b.txt, and c.txt

cat a.txt b.txt c.txt > merged.txt

# Sort the merged content and save it to result

sort -n merged.txt > result

# Display the sorted output

cat result

# Clean up temporary merged file

rm merged.txt

1. Write a shell script named as **systeminfo** that will display the information about the login name of the user, name of the Unix system used by the user, type of the SHELL, Path of current working directory of the user and list of file contain in current working directory. (Make the script an executable file and run it as a command using its name only.)

#!/bin/bash

# Display the login name of the user

echo "Login Name of the User: $LOGNAME"

# Display the name of the Unix system used by the user

echo "Name of the Unix System: $(uname -s)"

# Display the type of the shell

echo "Type of Shell: $SHELL"

# Display the path of the current working directory

echo "Current Working Directory: $(pwd)"

# Display the list of files in the current working directory

echo "Files in the Current Working Directory:"

ls -l

chmod +x systeminfo (terminal)

./systeminfo(execute the script using this)

1. Write a shell script named as **dtcal** for displaying both the system date and calendar for specific month, say march 2022, in the given format:-

Date : specific date Calender : current calendar

(Make the script an executable file and run it as a command using its name only.)

#!/bin/bash

# Display the current system date

echo "Date: $(date)"

# Display the calendar for March 2022

echo "Calendar for March 2022:"

cal 3 2022

 **Make the script executable**:

bash

Copy code

chmod +x dtcal

 **Run the script**: Execute the script using:

bash

Copy code

./dtcal

1. Write a shell script named as **nvwc** which will display the filename and linecount, wordcount and char count of the file dtcal in the following format:

Filename: dtcal Line count: - Word count: - Charcout: -

(Make the script an executable file and run it as a command using its name only.)

#!/bin/bash

# Get the filename

filename="dtcal"

# Display the filename, line count, word count, and character count

echo -e "Filename: $filename"

echo -e "Line count: $(wc -l < $filename)"

echo -e "Word count: $(wc -w < $filename)"

echo -e "Char count: $(wc -m < $filename)"

 **Make the script executable**:

bash

Copy code

chmod +x nvwc

 **Run the script**: Execute the script using:

bash

Copy code

./nvwc

1. Write a shell script named as **nvwc2** which will display the filename and linecount, word count and char count of **any file** given as argument to nvwc2 in the following format:

|  |  |  |  |
| --- | --- | --- | --- |
| filename | linecount | wordcount | charcount |
| file1 | - | - | - |

(Make the script an executable file and run it as a command using its name only.)

#!/bin/bash

# Check if a filename is provided as an argument

if [ -z "$1" ]; then

echo "Usage: $0 <filename>"

exit 1

fi

# Get the filename from the argument

filename="$1"

# Display the filename, line count, word count, and character count

echo -e "Filename\tLinecount\tWordcount\tCharcount"

echo -e "$filename\t$(wc -l < $filename)\t$(wc -w < $filename)\t$(wc -m < $filename)"

 **Make the script executable**:

bash

Copy code

chmod +x nvwc2

 **Run the script**: To run the script on any file, use the following command:

bash

Copy code

./nvwc2 <filename>

1. Write a shell script named as **darg** to display the total number of command line arguments along with the first two arguments.

-Modify the script to display all the arguments.

(Make the script an executable file and run it as a command using its name only.)

#!/bin/bash

# Display the total number of command line arguments

echo "Total number of arguments: $#"

# Display the first two arguments, if provided

echo "First argument: $1"

echo "Second argument: $2"

# Display all the arguments

echo "All arguments: $@"

 **Make the script executable**:

bash

Copy code

chmod +x darg

 **Run the script**: To run the script, you can provide command-line arguments as follows:

bash

Copy code

./darg arg1 arg2 arg3 arg4

1. Write a shell script named as **ndisp** that will take three command line arguments specifying the value of n, m and a filename and display the first n number of lines and last m number of lines of the file given as argument.

(Make the script an executable file and run it as a command using its name only.)

#!/bin/bash

# Check if exactly three arguments are provided

if [ $# -ne 3 ]; then

echo "Usage: $0 <n> <m> <filename>"

exit 1

fi

# Assign command-line arguments to variables

n=$1

m=$2

filename=$3

# Display the first n lines of the file

echo "First $n lines of the file $filename:"

head -n $n $filename

# Display the last m lines of the file

echo "Last $m lines of the file $filename:"

tail -n $m $filename

 **Make the script executable**:

bash

Copy code

chmod +x ndisp

 **Run the script**: To run the script, use the following command:

bash

Copy code

./ndisp <n> <m> <filename>