**LAB SHEET 4 – Operating Systems**

**Richu James**

**AM.SC.U4CYS23036**

**Question 1:**

Write a shell script to generate emails in the given format and write it into a file. Your script should accept sender and recipient email id’s and subject as command line arguments.

From: abc@domain1.com

To: xx@domain.com

Cc: yy@domain.com

Subject: Subject 1 This email is generated by my shell script.

Thanks and regards

S4 CSE student

Amritapuri

Answer:

#!/bin/bash

SENDER=$1

RECIPIENT=$2

SUBJECT=$3

cat <<EOM > email.txt

From: $SENDER

To: $RECIPIENT

Cc: yy@domain.com

Subject: $SUBJECT

This email is generated by my shell script.

Thanks and regards,

S4 CSE student,

Amritapuri

EOM

echo "Email content saved to email.txt"

**QUESTION 2:**

Modify Question 1 to allow user to enter text at the beginning of email content, by passing it as a command line argument.

ANSWER:

#!/bin/bash

# Usage: ./email\_gen.sh sender@example.com recipient@example.com "Subject" "Custom Text"

# Check if the correct number of arguments are provided

if [ "$#" -ne 4 ]; then

echo "Usage: $0 sender\_email recipient\_email subject custom\_text"

exit 1

fi

SENDER=$1

RECIPIENT=$2

SUBJECT=$3

CUSTOM\_TEXT=$4

# Write the email to a file

cat <<EOM > email.txt

From: $SENDER

To: $RECIPIENT

Cc: yy@domain.com

Subject: $SUBJECT

$CUSTOM\_TEXT

This email is generated by my shell script.

Thanks and regards,

S4 CSE student,

Amritapuri

EOM

echo "Email content saved to email.txt"

**QUESTION 3:**

Write a shell script to print all the primes below a given number.

ANSWER:

Code:

#!/bin/bash

# Usage: ./print\_primes.sh number

is\_prime() {

n=$1

if [ "$n" -le 1 ]; then

return 1

fi

for ((i = 2; i \* i <= n; i++)); do

if [ $((n % i)) -eq 0 ]; then

return 1

fi

done

return 0

}

if [ "$#" -ne 1 ]; then

echo "Usage: $0 number"

exit 1

fi

LIMIT=$1

for ((n = 2; n < LIMIT; n++)); do

if is\_prime "$n"; then

echo "$n"

fi

done

**QUESTION 4**

Write a shell script to print the first n Fibonacci numbers.

Answer:

#!/bin/bash

# Usage: ./fibonacci.sh number

if [ "$#" -ne 1 ]; then

echo "Usage: $0 number\_of\_fibonacci\_numbers"

exit 1

fi

n=$1

a=0

b=1

echo "$a"

if [ "$n" -gt 1 ]; then

echo "$b"

fi

for ((i = 2; i < n; i++)); do

temp=$b

b=$((a + b))

a=$temp

echo "$b"

done

**Question 5**

Write a shell script to generate a multiplication table.

a.Interactive version: The program should accept an integer n given by the user and should print the multiplication table of that n.

Code:

#!/bin/bash

# Multiplication table interactive version

read -p "Enter a number: " n

for i in {1..10}; do

echo "$n \* $i = $((n \* i))"

done

b. Command line arguments version

#!/bin/bash

# Usage: ./multiplication\_table.sh number

if [ "$#" -ne 1 ]; then

echo "Usage: $0 number"

exit 1

fi

n=$1

for i in {1..10}; do

echo "$n \* $i = $((n \* i))"

done

c.Redirection version

#!/bin/bash

# Usage: ./multiplication\_table\_redirection.sh < file\_with\_number.txt

read n

for i in {1..10}; do

echo "$n \* $i = $((n \* i))"

done

**Question 6**

Script to find GCD using a function

Answer:

#!/bin/bash

# Usage: ./gcd.sh number1 number2

gcd() {

a=$1

b=$2

while [ "$b" -ne 0 ]; then

temp=$b

b=$((a % b))

a=$temp

done

echo $a

}

if [ "$#" -ne 2 ]; then

echo "Usage: $0 number1 number2"

exit 1

fi

gcd "$1" "$2"

**Question 7**

Script to find factorial using recursion

Answer:

#!/bin/bash

# Usage: ./factorial.sh number

factorial() {

if [ "$1" -le 1 ]; then

echo 1

else

echo $(( $1 \* $(factorial $(( $1 - 1 )) ) ))

fi

}

if [ "$#" -ne 1 ]; then

echo "Usage: $0 number"

exit 1

fi

factorial "$1"