### **Practical No. 9: To implement basic shell commands and shell scripts**

#### **A. Basic Shell Commands**

1. **Directory and File Operations:**
   * pwd: Show current working directory
   * ls: List contents of directory
   * touch file1.txt: Create a new file
   * cat file1.txt: Display file content
   * chmod 644 file1.txt: Change file permissions
   * rm file1.txt: Delete a file
   * rmdir "$dirname": Remove directory named by variable
2. **Loop Constructs:**

**For loop:** bash  
CopyEdit  
for i in {1..3}; do

echo "Number $i"

done

**While loop:** bash  
CopyEdit  
counter=1

while [ $counter -le 3 ]; do

echo "Count $counter"

((counter++))

done

**Until loop:** bash  
CopyEdit  
counter=1

until [ $counter -gt 3 ]; do

echo "Counter $counter"

((counter++))

done

**Function Use:** bash  
CopyEdit  
greet\_user() {

echo "Welcome, $1!"

}

name="Jhon"

greet\_user "$name"

#### **B. Custom Shell Script Execution**

**Script: myscript.sh**

bash

CopyEdit

#!/bin/bash

echo "Welcome to Basic Shell Script!"

echo "Current directory is:"

pwd

echo "List of files:"

ls -l

echo "Current date and time:"

date

echo "Content of a file (test.txt):"

cat test.txt

**To run:**

bash

CopyEdit

chmod +x myscript.sh

./myscript.sh

#### **C. Mini Shell in C**

c

CopyEdit

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#define MAX\_CMD\_LEN 1024

int main() {

char command[MAX\_CMD\_LEN];

while (1) {

printf("myshell> ");

fgets(command, MAX\_CMD\_LEN, stdin);

command[strcspn(command, "\n")] = 0;

if (strcmp(command, "exit") == 0) {

printf("Exiting shell...\n");

break;

}

pid\_t pid = fork();

if (pid == 0) {

execlp(command, command, (char \*)NULL);

printf("Command not found: %s\n", command);

exit(1);

} else {

wait(NULL);

}

}

return 0;

}