**Ex No 3**

**Date**

**SYSTEM CALLS PROGRAMMING**

**Problem Statement:**

Develop a menu-driven program using the exec() system call to execute Linux commands, including ls, cat, cp, echo, ps, rm, mv, man, chmod, and clear.

**Problem Description:**

Design a menu-driven program using the execl() system call to execute Linux commands like ls, cat, cp, echo, ps, rm, mv, man, chmod, and clear, providing an interactive interface for users to conveniently run these commands. Ensure secure handling of user input to prevent vulnerabilities like command injection.

**Synopsis:**

1. **ls:** List files and directories.
2. **cat:** Concatenate and display file contents.
3. **cp:** Copy files and directories.
4. **echo:** Display text.
5. **ps:** List running processes.
6. **rm:** Remove files and directories.
7. **mv:** Move or rename files and directories.
8. **man**: View manual pages.
9. **chmod:** Change file permissions.
10. **clear:** Clear the terminal screen.

**Code:**

import os

command\_path =['/bin/ls','/bin/cat','/bin/cp','/bin/echo','/bin/ps','/bin/rm','/bin/mv','/usr/bin/man','/bin/chmod','/usr/bin/clear']

print('''

1.ls

2.cat

3.cp b

4.echo

5.ps

6.rm

7.mv

8.man

9.chmod

10.clear

11.exit

'''

)

x=int(input("enter choice"))

while x!=11:

    i=x

    if i==11:

        break

    elif i==1:

        os.execl(command\_path[0], 'ls', '-l')

    elif i==2:

        os.execl(command\_path[1], 'cat', '1.txt')

    elif i==3:

        os.execl(command\_path[2], 'cp ', '1.txt' '2.txt')

    elif i==4:

        os.execl(command\_path[3], 'echo', 'Hello World')

    elif i==5:

        os.execl(command\_path[4], 'ps', '-aux')

    elif i==6:

        os.execl(command\_path[5], 'rm', '2.txt')

    elif i==7:

        os.execl(command\_path[6], 'mv', '1.txt','2.txt')

    elif i==8:

        os.execl(command\_path[7], 'man', 'ls')

    elif i==9:

        os.execl(command\_path[8], 'chmod', '755','2.txt')

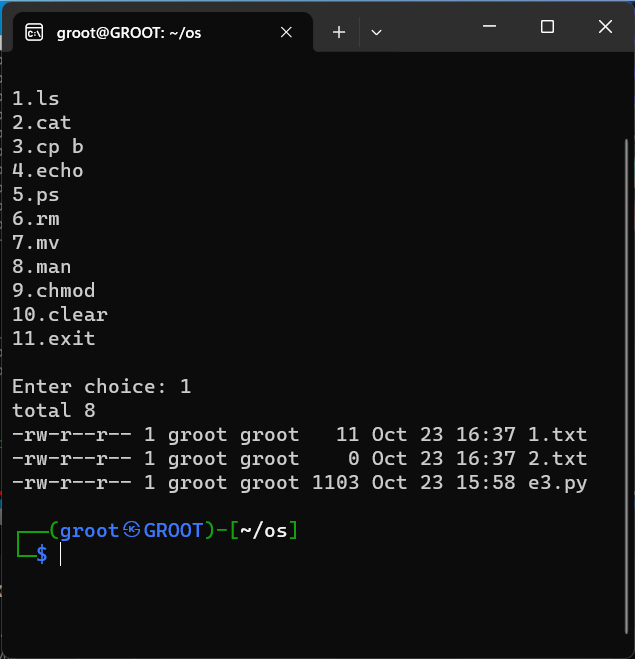
    elif i==10:

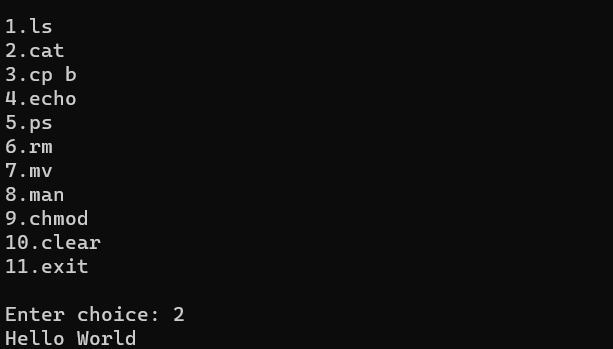
        os.execl(command\_path[9], 'clear')

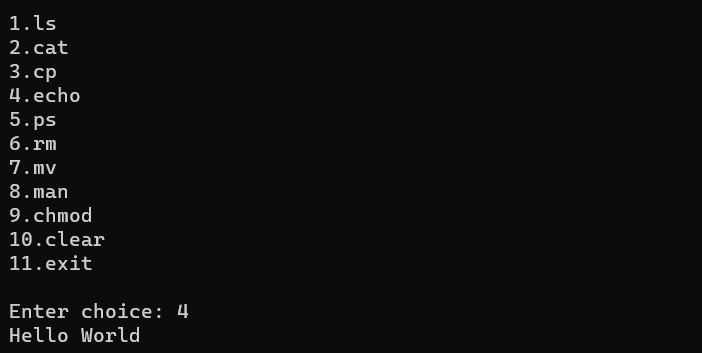
    else:

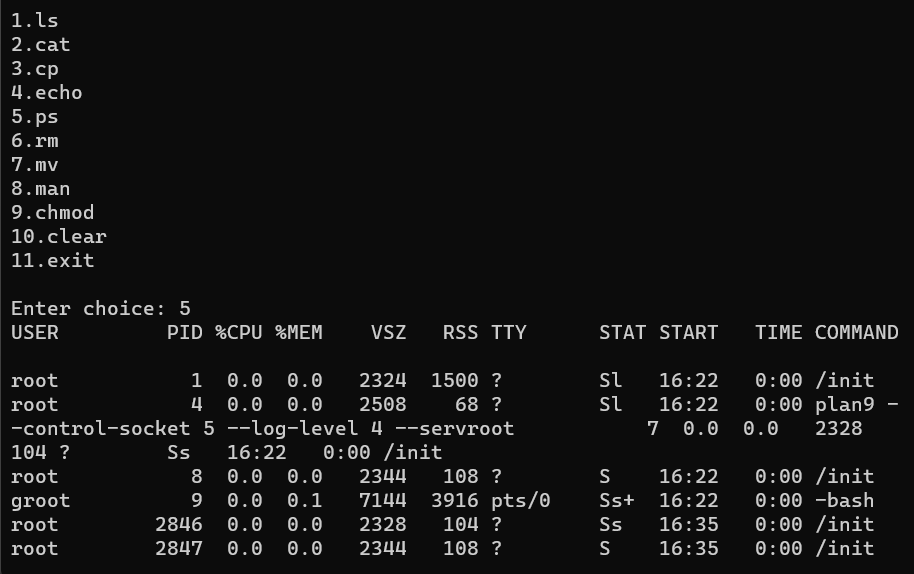
        print("enter a valid choice")

**Output:**

****

****

****

****

**Problem Statement:**

Use exec() system call for process 1 to call process 2 and process 2 to call process 1 to create an infinite loop.

**Problem Description:**

This exercise demonstrates an infinite loop created between two processes, Process 1 and Process 2, through the use of the `exec()` system call. Process 1 invokes Process 2 with `exec()`, and Process 2 subsequently triggers Process 1 using the same system call, resulting in an endlessly repeating execution pattern.

**Synopsis:**

**exec():** replaces the current process with a new one, loading and executing a different program.

**Code:**

**Process 1:**

def main():

    i=int(input("Process 1; Input: "))

    if i!=0:

        import p2

        p2.main(i)

if \_\_name\_\_ == '\_\_main\_\_':

    exec("main()")

**Process 2:**

def main(i):

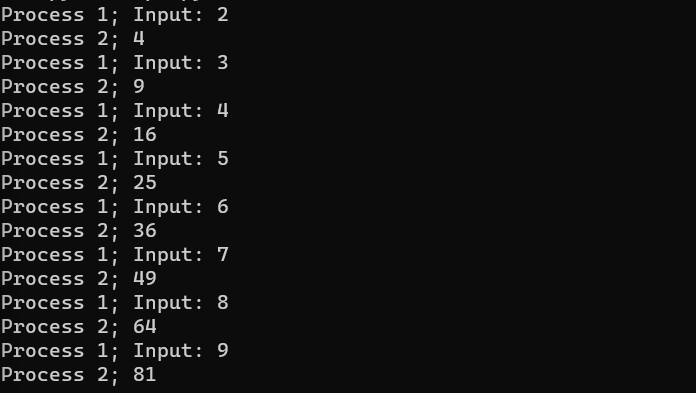
    if i!=0:

        import p1

        print("Process 2;",(i)\*(i))

        exec('p1.main()')

**Output:**

****

**Result:**

Thus, the menu driven program using execl() and calling one process from another process using exec() was executed successfully.