**Ex. No: 4**

**Date:**

**SIMULATION OF LINUX COMMANDS**

**Problem Statement:**

To simulate ls, cp, cat, mv, grep Linux commands using subprocess.

**Problem Description:**

Simulate Linux commands like ls, cp, cat, mv, and grep using Python's subprocess module, allowing you to execute these commands from within a Python script, mimicking their behavior and functionality on a Linux system.

**Synopsis:**

* **ls:** list files and directories
* **cp:** copy files and directories
* **cat:** concatenate and display file contents
* **mv:** move or rename files and directories
* **grep:** Search for patterns in text within files

**Code:**

import subprocess

# Command 1: ls (List files in a directory)

option1 = subprocess.run(["ls", "-l"], stdout=subprocess.PIPE, text=True)

option2 = subprocess.run(["ls", "-a"], stdout=subprocess.PIPE, text=True)

print("Option 1 - List files in long format:")

print(option1.stdout)

print("Option 2 - List all files (including hidden):")

print(option2.stdout)

# Command 2: cp (Copy a file)

source\_file = "source.txt"

destination\_file = "destination.txt"

subprocess.run(["cp", source\_file, destination\_file])

print(f"File '{source\_file}' copied to '{destination\_file}'")

# Command 3: cat (Display file contents)

file\_to\_display = "source.txt"

option1 = subprocess.run(["cat", file\_to\_display], stdout=subprocess.PIPE, text=True)

option2 = subprocess.run(["cat", "-n", file\_to\_display], stdout=subprocess.PIPE, text=True)

print("Option 1 - Display file contents:")

print(option1.stdout)

print("Option 2 - Display file contents with line numbers:")

print(option2.stdout)

# Command 4: mv (Move or rename a file)

source\_file = "source.txt"

destination\_file = "newfile.txt"

subprocess.run(["mv", source\_file, destination\_file])

print(f"File '{source\_file}' moved/renamed to '{destination\_file}'")

# Command 5: grep (Search for a pattern in a file)

file\_to\_search = "newfile.txt"

pattern = "Hello"

option1 = subprocess.run(["grep", pattern, file\_to\_search], stdout=subprocess.PIPE, text=True)

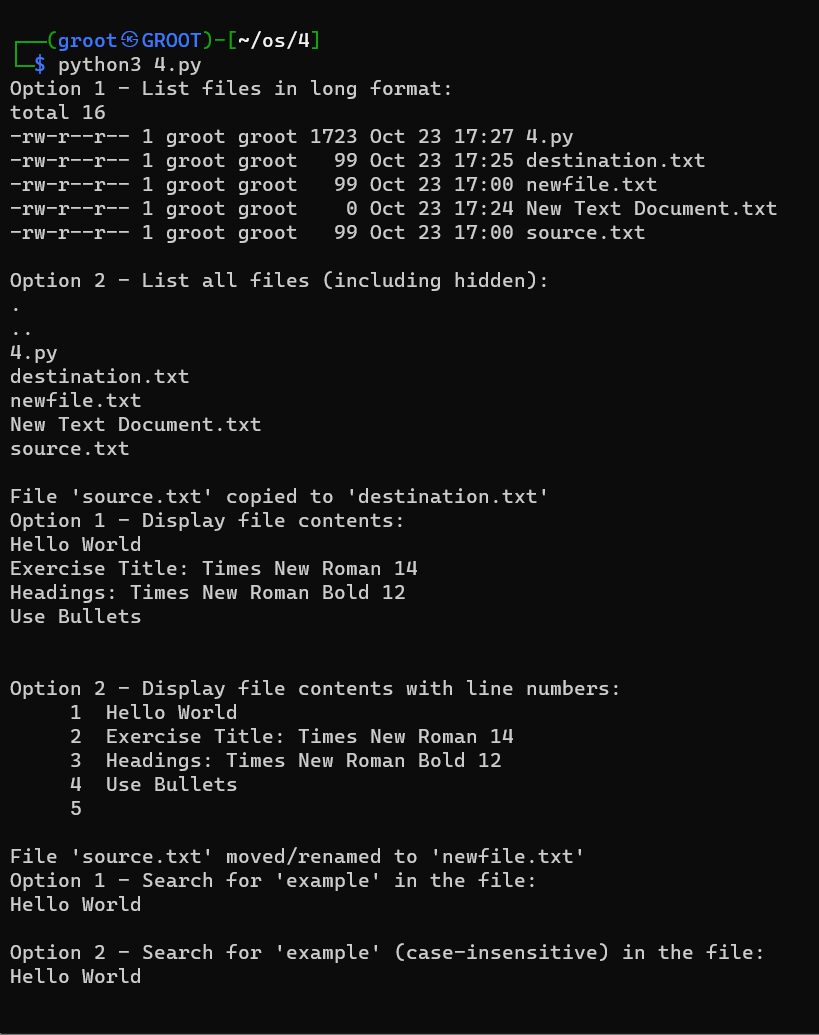
option2 = subprocess.run(["grep", "-i", pattern, file\_to\_search], stdout=subprocess.PIPE, text=True)

print("Option 1 - Search for 'example' in the file:")

print(option1.stdout)

print("Option 2 - Search for 'example' (case-insensitive) in the file:")

print(option2.stdout)

****

**Result:**

Thus, all the Linux commands were simulated using subprocess from a python script.