**UIT2512---Operating Systems Practices Lab**

# 4) Simulation of Linux Commands in Python

# Name: Vasundhara.B

# Roll no: 3122 21 5002 119

1. **ls**

**CODE:**

import os

# Function : List all files, including hidden files (ls -a)

def ls\_a():

    files = os.listdir('.')

    for file in files:

        print(file)

# Function: List files in a detailed format (ls -l)

def ls\_l():

    files = os.listdir('.')

    for file in files:

        file\_info = os.stat(file)

        print(f"File: {file}, Size: {file\_info.st\_size} bytes")

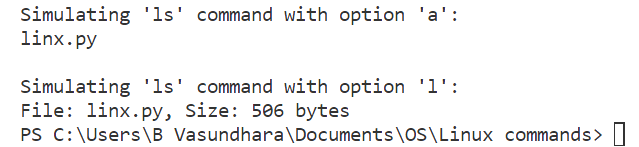
print("Simulating 'ls' command with option 'a':")

ls\_a()

print("\nSimulating 'ls' command with option 'l':")

ls\_l()

**OUTPUT:**



1. **cat**

**CODE:**

# Function: Display line numbers (cat -n)

def cat\_n(filename):

    with open(filename, 'r') as file:

        lines = file.readlines()

        for i, line in enumerate(lines, start=1):

            print(f"{i}: {line.strip()}")

# Function: Display the last 'n' lines of the file (cat -tail <num\_lines>)

def cat\_tail(filename, num\_lines):

    with open(filename, 'r') as file:

        lines = file.readlines()

        last\_lines = lines[-num\_lines:]

        for i, line in enumerate(last\_lines, start=len(lines) - num\_lines + 1):

            print(f"{i}: {line.strip()}")

print("Simulating 'cat' command with option 'tail' lines:")

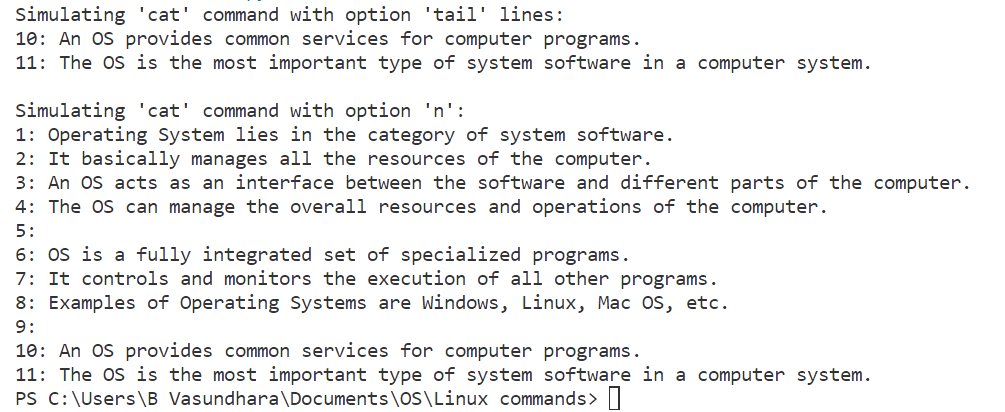
cat\_tail('file1.txt', 2)  # Display the last 5 lines

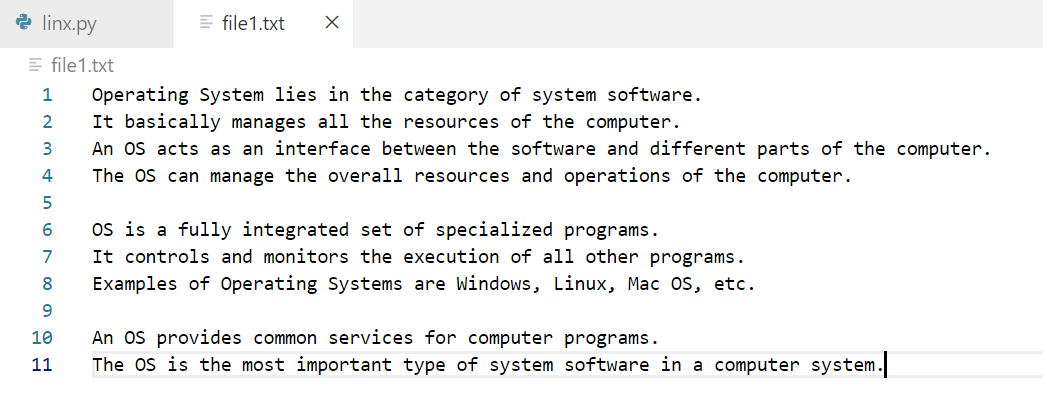
print()

print("Simulating 'cat' command with option 'n':")

cat\_n('file1.txt')

**OUTPUT:**





1. **cp**

**CODE:**

import shutil

#Function: Prompt before overwrite (cp -i)

def cp\_option\_i(src, dest):

    shutil.copy2(src, dest)

# Function: Copy directories recursively (cp -r)

def cp\_option\_r(src, dest):

    shutil.copytree(src, dest)

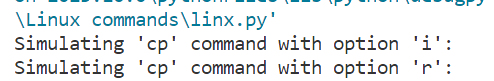
print("Simulating 'cp' command with option 'i':")

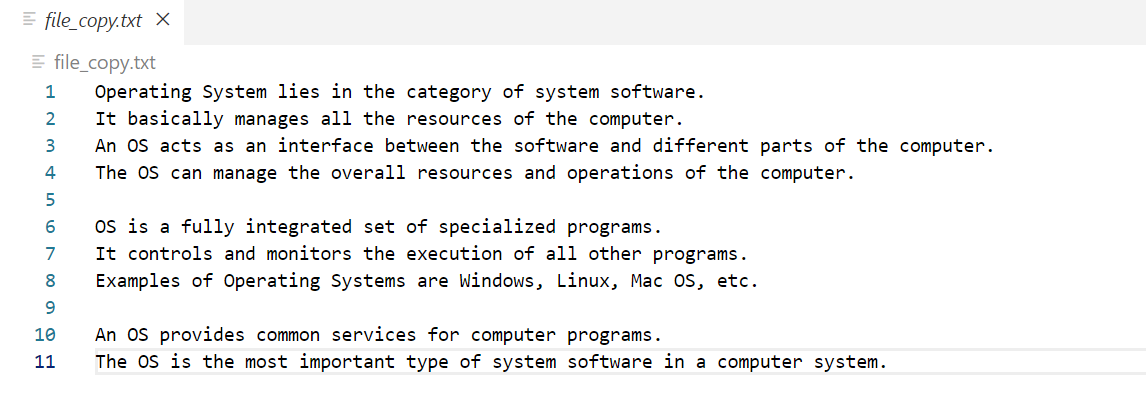
cp\_option\_i('file1.txt', 'file\_copy.txt')

print("Simulating 'cp' command with option 'r':")

cp\_option\_r('newdirectory', 'directorycopy')

**OUTPUT:**





1. **mv**

**CODE:**

import shutil

# Option: Move only when source is newer or missing (mv -u)

def mv\_option\_u(src, dest):

    shutil.move(src, dest)

# Option: Make a backup of existing destination files (mv -b)

def mv\_option\_b(src, dest):

    shutil.move(src, f"{dest}.bak")

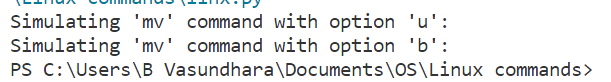
print("Simulating 'mv' command with option 'u':")

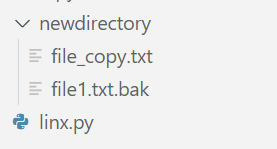
mv\_option\_u('file1.txt', 'newdirectory/file1.txt')

print("Simulating 'mv' command with option 'b':")

mv\_option\_b('file1.txt', 'newdirectory/file1.txt')

**OUTPUT:**





1. **grep**

**CODE:**

import re

# Option: Ignore case (grep -i)

def grep\_option\_i(pattern, filename):

    with open(filename, 'r') as file:

        for line in file:

            if re.search(pattern, line, re.IGNORECASE):

                print(line.strip())

# Option: Count of matching lines (grep -c)

def grep\_option\_c(pattern, filename):

    count = 0

    with open(filename, 'r') as file:

        for line in file:

            if re.search(pattern, line):

                count += 1

    print(f"Count of lines matching the pattern: {count}")

print("Simulating 'grep' command with option 'i':")

grep\_option\_i('OS', 'file.txt')

print()

print("Simulating 'grep' command with option 'c':")

grep\_option\_c('OS', 'file.txt')

**OUTPUT:**

