PRACTICAL 4

import os

import sys

import stat

import time

def get\_file\_details(file\_path):

    try:

        # Get file stat information

        file\_stat = os.stat(file\_path)

        # Extract owner access permissions

        permissions = stat.filemode(file\_stat.st\_mode)

        # Get file access time

        access\_time = time.ctime(file\_stat.st\_atime)

        # Print file details

        print(f"File: {file\_path}")

        print(f"Owner access permissions: {permissions}")

        print(f"File access time: {access\_time}")

    except FileNotFoundError:

        print(f"Error: File '{file\_path}' not found.")

    except Exception as e:

        print(f"Error: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

    # Check if a file name is provided as a command-line argument

    if len(sys.argv) != 2:

        print("Usage: python file\_details.py <file\_name>")

        sys.exit(1)

    file\_name = sys.argv[1]

    get\_file\_details(file\_name)

PRACTICAL 6

class Process:

    def \_\_init\_\_(self, process\_id, arrival\_time, burst\_time):

        self.process\_id = process\_id

        self.arrival\_time = arrival\_time

        self.burst\_time = burst\_time

def fcfs\_scheduler(processes):

    processes.sort(key=lambda x: x.arrival\_time)  # Sort processes based on arrival time

    completion\_time = [0] \* len(processes)

    turnaround\_time = [0] \* len(processes)

    waiting\_time = [0] \* len(processes)

    completion\_time[0] = processes[0].burst\_time

    turnaround\_time[0] = completion\_time[0] - processes[0].arrival\_time

    waiting\_time[0] = turnaround\_time[0] - processes[0].burst\_time

    for i in range(1, len(processes)):

        completion\_time[i] = completion\_time[i - 1] + processes[i].burst\_time

        turnaround\_time[i] = completion\_time[i] - processes[i].arrival\_time

        waiting\_time[i] = turnaround\_time[i] - processes[i].burst\_time

    print("Process\t Arrival Time\t Burst Time\t Completion Time\t Turnaround Time\t Waiting Time")

    for i in range(len(processes)):

        print(f"{processes[i].process\_id}\t\t{processes[i].arrival\_time}\t\t{processes[i].burst\_time}\t\t"

              f"{completion\_time[i]}\t\t\t{turnaround\_time[i]}\t\t\t{waiting\_time[i]}")

if \_\_name\_\_ == "\_\_main\_\_":

    # Example processes

    processes = [

        Process(1, 0, 6),

        Process(2, 2, 4),

        Process(3, 4, 8),

        Process(4, 6, 5),

    ]

    fcfs\_scheduler(processes)