Name: Mudit Mohit Assignment 8

Enrolment Number: BT21GCS057

Section: B2

Q.1) WAP to simulate preemptive and non-preemptive longest job first.

Ans.

import matplotlib.pyplot as plt

comp = ["Process", "TAT", "BT", "WT", "AT", "RD"]

import matplotlib.pyplot as plt

def ljf\_nonpreemptive(filename):

    with open(filename, 'r') as f:

        lines = f.readlines()

        n = int(lines[0].strip())

        processes = []

        for i in range(1, n + 1):

            pid, at, bt = map(int, lines[i].split())

            processes.append({'pid': pid, 'at': at, 'bt': bt})

    processes.sort(key=lambda x: x['at'])

    t = 0

    completed = 0

    gantt\_chart = []

    start\_times = {}

    responses = []

    while completed < n:

        eligible\_processes = [p for p in processes if p['at'] <= t and p['bt'] > 0]

        if len(eligible\_processes) == 0:

            t += 1

            continue

        longest\_process = max(eligible\_processes, key=lambda x: x['bt'])

        longest\_process['bt'] = 0

        pid = longest\_process['pid']

        if pid not in start\_times:

            start\_times[pid] = t

        gantt\_chart.append(longest\_process['pid'])

        completed += 1

        longest\_process['ct'] = t + 1

        longest\_process['tat'] = longest\_process['ct'] - longest\_process['at']

        longest\_process['wt'] = longest\_process['tat'] - longest\_process['bt']

        rt = start\_times[pid] - longest\_process['at']

        responses.append(rt)

        t += 1

    print("+-----+--------------+------------+--------------+---------------+------------------+----------------+")

    print("| PID | Arrival Time | Burst Time | Waiting Time | Response Time | Turn Around Time | Relative Delay |")

    print("+-----+--------------+------------+--------------+---------------+------------------+----------------+")

    total\_tat = 0

    total\_wt = 0

    relative\_delay = []

    for p in range(0, len(processes)):

        f"| {'AVG':3} | {'':12} | {'':10} | {total\_wt/n:12} | {sum(responses) / len(processes): 13} | {total\_tat/n:16} | {round(sum(relative\_delay) / len(processes), 2):14} |"

    print("+-----+--------------+------------+--------------+---------------+------------------+----------------+")

    fig, ax = plt.subplots()

    ax.set\_title("Gantt Chart - LJF Preemptive")

    ax.set\_xlabel("Time")

    ax.set\_ylabel("Process")

    ax.grid(True)

    for i in range(len(gantt\_chart)):

        if gantt\_chart[i] == 1:

            color\_str = '#f49097'

        elif gantt\_chart[i] == 2:

            color\_str = '#dfb2f4'

        elif gantt\_chart[i] == 3:

            color\_str = '#bad4aa'

        elif gantt\_chart[i] == 4:

            color\_str = '#ff6b6b'

        else:

            color\_str = '0.5'

        ax.broken\_barh([(i, 1)], (0.5, 0.4), facecolors=color\_str)

        ax.text(i + 0.5, 0.7, str(gantt\_chart[i]), ha='center', va='center', color='white', fontsize=10)

    ax.set\_ylim(0.1, 1)

    ax.set\_xlim(0, len(gantt\_chart))

    plt.show()

def ljf\_preemptive(filename):

    with open(filename, 'r') as f:

        lines = f.readlines()

        n = int(lines[0].strip())

        processes = []

        for i in range(1, n + 1):

            pid, at, bt = map(int, lines[i].split())

            processes.append({'pid': pid, 'at': at, 'bt': bt, 'rd': bt})

    processes.sort(key=lambda x: x['at'])

    t = 0

    completed = 0

    gantt\_chart = []

    start\_times = {}

    responses = []

    while completed < n:

        eligible\_processes = [p for p in processes if p['at'] <= t and p['rd'] > 0]

        if len(eligible\_processes) == 0:

            t += 1

            continue

        longest\_process = max(eligible\_processes, key=lambda x: x['rd'])

        longest\_process['rd'] -= 1

        pid = longest\_process['pid']

        if pid not in start\_times:

            start\_times[pid] = t

        gantt\_chart.append(longest\_process['pid'])

        if longest\_process['rd'] == 0:

            completed += 1

            longest\_process['ct'] = t + 1

            longest\_process['tat'] = longest\_process['ct'] - longest\_process['at']

            longest\_process['wt'] = longest\_process['tat'] - longest\_process['bt']

            rt = start\_times[pid] - longest\_process['at']

            responses.append(rt)

        t += 1

    print("+-----+--------------+------------+--------------+---------------+------------------+----------------+")

    print("| PID | Arrival Time | Burst Time | Waiting Time | Response Time | Turn Around Time | Relative Delay |")

    print("+-----+--------------+------------+--------------+---------------+------------------+----------------+")

    total\_tat = 0

    total\_wt = 0

    relative\_delay = []

    for p in range(0, len(processes)):

        f"| {'AVG':3} | {'':12} | {'':10} | {total\_wt/n:12} | {sum(responses) / len(processes): 13} | {total\_tat/n:16} | {round(sum(relative\_delay) / len(processes), 2):14} |"

    print("+-----+--------------+------------+--------------+---------------+------------------+----------------+")

    fig, ax = plt.subplots()

    ax.set\_title("Gantt Chart - LJF Preemptive")

    ax.set\_xlabel("Time")

    ax.set\_ylabel("Process")

    ax.grid(True)

    for i in range(len(gantt\_chart)):

        if gantt\_chart[i] == 1:

            color\_str = '#f49097'

        elif gantt\_chart[i] == 2:

            color\_str = '#dfb2f4'

        elif gantt\_chart[i] == 3:

            color\_str = '#bad4aa'

        elif gantt\_chart[i] == 4:

            color\_str = '#ff6b6b'

        else:

            color\_str = '0.5'

        ax.broken\_barh([(i, 1)], (0.5, 0.4), facecolors=color\_str)

        ax.text(i + 0.5, 0.7, str(gantt\_chart[i]), ha='center', va='center', color='white', fontsize=10)

    ax.set\_ylim(0.1, 1)

    ax.set\_xlim(0, len(gantt\_chart))

    plt.show()

program = ["Non Premptive LJF", "Premptive LJF"]

print('Choose a program to run')

for i in range(0,2):

    print(f'{i+1}: {program[i]}')

choice = int(input("> "))

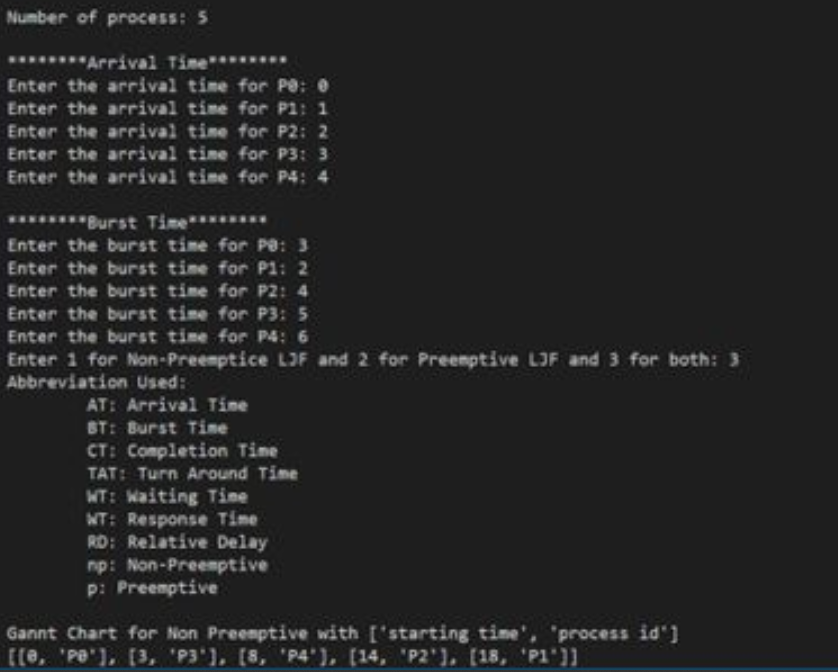
if choice == 1:

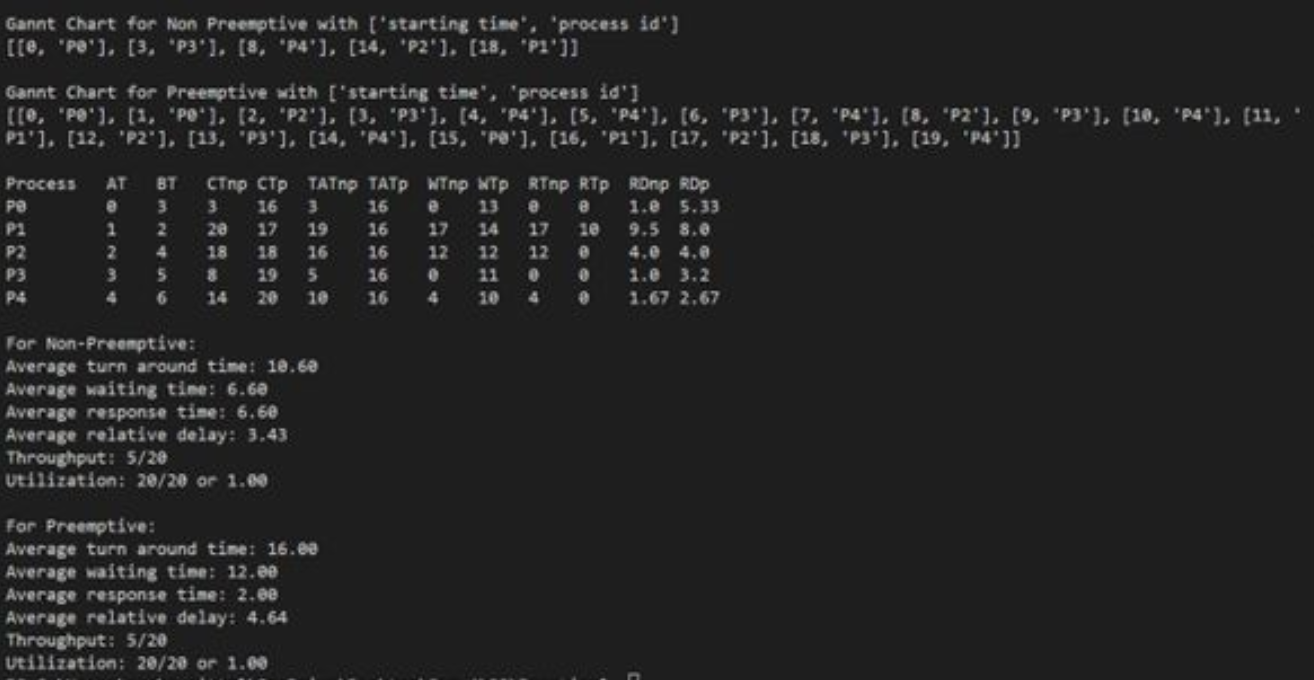
    ljf\_nonpreemptive('Input.txt')

else:

    ljf\_preemptive('Input.txt')

Output:





Question 2 (a)

#!/bin/bash

echo -e "Enter Number : \c"

read n

while [ $n -gt 2 ]

do

for((i=2; i<=$n/2; i++))

do

res=$(( n%i ))

if [ $res -eq 0 ]

then

echo "$n is not a prime number."

exit 0

fi

done

done

echo "$n is a prime number."

Question 2 (b)

#! /bin/bash

echo "Enter the value of n"

read n

a=0

b=1

count=2

echo "Fibonacci series:"

echo $a

echo $b

while [ $count -le $n ]

do

fib=`expr $a + $b`

a=$b

b=$fib

echo $fib

count=`expr $count + 1`

done