**OS experiment no. 07**

**Name :** Devansh Shah

**Roll No:** 1914078

**Batch :** B1

**Exp No:** 07

**Date:**

**Q1. Students have to study different non pre-emptive, pre-emptive scheduling algorithms and implement any one of them.**

**Q2.Calculate average waiting time and average turnaround time of the algorithm.**

**Solution:**

1. **FCFS-First Come First Serve Scheduling Algorithm- Non-Preemptive**

**CODE(PYTHON):**

import pandas as pd

noOfProcessors = int(input("Enter Number Of Processors: "))

columns = ["Processor", "Arrival Time",

           "Burst Time", "Waiting Time", "Turn-around Time"]

df = pd.DataFrame(columns=columns, index=[i for i in range(noOfProcessors)])

for i in range(noOfProcessors):

    arrival = int(input("Enter Arrival Time of Processor "+str(i)+": "))

    burst = int(input("Enter Burst Time of Processor "+str(i)+": "))

    df.loc[i] = ["P"+str(i), arrival, burst, 0, 0]

df.sort\_values(by=["Arrival Time"], inplace=True)

start\_time = df["Arrival Time"].tolist()[0]

waiting\_time = [0]

for i in range(noOfProcessors-1):

    time = start\_time

    for j in range(i+1):

        time += df["Burst Time"].tolist()[j]

    time -= df["Arrival Time"].tolist()[i+1]

    waiting\_time.append(time)

df["Waiting Time"] = waiting\_time

turn\_time = []

for i in range(noOfProcessors):

    time = start\_time

    for j in range(i+1):

        time += df["Burst Time"].tolist()[j]

    time -= df["Arrival Time"].tolist()[i]

    turn\_time.append(time)

df["Turn-around Time"] = turn\_time

print(df)

mean\_wt = df['Waiting Time'].mean()

mean\_tt = df['Turn-around Time'].mean()

print("Average waiting time = " + str(mean\_wt))

print("Average turn around time = "+str(mean\_tt))

**OUTPUT:**

Enter Number Of Processors: 5

Enter Arrival Time of Processor 0: 0

Enter Burst Time of Processor 0: 5

Enter Arrival Time of Processor 1: 1

Enter Burst Time of Processor 1: 2

Enter Arrival Time of Processor 2: 2

Enter Burst Time of Processor 2: 4

Enter Arrival Time of Processor 3: 3

Enter Burst Time of Processor 3: 3

Enter Arrival Time of Processor 4: 4

Enter Burst Time of Processor 4: 6

Processor Arrival Time Burst Time Waiting Time Turn-around Time

0 P0 0 5 0 5

1 P1 1 2 4 6

2 P2 2 4 5 9

3 P3 3 3 8 11

4 P4 4 6 10 16

Average waiting time = 5.4

Average turn around time = 9.4

**Average Waiting Time :** 5.4 ms

**Average Turn around Time:** 9.4ms

1. **RR-Round Robin Scheduling Algorithm-Preemptive**

**CODE(PYTHON):**

import pandas as pd

noOfProcessors = int(input("Enter Number of Processors: "))

columns = ["Processor", "Arrival Time", "Burst Time",

           "Completion Time", "Waiting Time", "Turn-around Time"]

df = pd.DataFrame(columns=columns, index=[i for i in range(noOfProcessors)])

for i in range(noOfProcessors):

    arrival = int(input("Enter Arrival Time of Processor "+str(i)+": "))

    burst = int(input("Enter Burst Time of Processor "+str(i)+": "))

    df.loc[i] = ["P"+str(i), arrival, burst, 0, 0, 0]

df.sort\_values(by=["Arrival Time"], inplace=True)

timestamp = int(input("Enter Timestamp: "))

print("Timestamp: "+str(timestamp))

start\_time = df["Arrival Time"].tolist()[0]

dictionary = df[['Processor', 'Arrival Time', 'Burst Time']].to\_dict("records")

processor1 = df["Processor"].tolist()[0]

queue = [processor1]

df.set\_index("Processor", inplace=True)

t = start\_time

total\_time = df[['Burst Time']].sum()

burst\_time = df[["Burst Time"]]

time = start\_time+total\_time

while(t < time).bool():

    current = queue.pop(0)

    if(df.loc[current, 'Burst Time'] >= timestamp):

        t += timestamp

        df.loc[current, 'Burst Time'] -= timestamp

        for pr in dictionary:

            if (pr['Arrival Time'] > t-timestamp and pr['Arrival Time'] <= t):

                queue.append(pr['Processor'])

        if(df.loc[current, 'Burst Time'] == 0):

            df.loc[current, 'Completion Time'] = t

        else:

            queue.append(current)

    else:

        t += df.loc[current, 'Burst Time']

        df.loc[current, 'Completion Time'] = t

        for pr in dictionary:

            if (pr['Arrival Time'] > t-timestamp and pr['Arrival Time'] <= t):

                queue.append(pr['Processor'])

df[['Burst Time']] = burst\_time

df['Turn-around Time'] = df['Completion Time']-df['Arrival Time']

df['Waiting Time'] = df['Turn-around Time']-df['Burst Time']

print()

print(df.reset\_index())

mean\_wt = df['Waiting Time'].mean()

mean\_tt = df['Turn-around Time'].mean()

print("Average waiting time = " + str(mean\_wt))

print("Average turn around time = "+str(mean\_tt))

**OUTPUT:**

Enter Number of Processors: 4

Enter Arrival Time of Processor 0: 0

Enter Burst Time of Processor 0: 5

Enter Arrival Time of Processor 1: 1

Enter Burst Time of Processor 1: 4

Enter Arrival Time of Processor 2: 2

Enter Burst Time of Processor 2: 2

Enter Arrival Time of Processor 3: 4

Enter Burst Time of Processor 3: 1

Enter Timestamp: 2

Timestamp: 2

Processor Arrival Time Burst Time Completion Time Waiting Time \

0 P0 0 5 12 7

1 P1 1 4 11 6

2 P2 2 2 6 2

3 P3 4 1 9 4

Turn-around Time

0 12

1 10

2 4

3 5

Average waiting time = 4.75

Average turn around time = 7.75

**Average Waiting Time :** 4.75 ms

**Average Turn around Time:** 7.75ms

**Outcome:** CO2: Demonstrate use of inter process communication

**Conclusion:** We learnt and implemented one Preemptive and one Non-Preemptive scheduling algorithms and calculated the average waiting time as well as the average turn-around time.