

**Usman Institute of Technology**

**Department of Computer Science Fall 2022**

Name: Muhammad Waleed

Roll no: 20B-115-SE

Course: Operating Systems (CS312)

Course Instructor: Ma’am Shabina Mushtaq

Date: 1-Dec-2022

# Round Robins:

import os

try:

    from rich.console import Console

    from rich.table import Table

except ImportError:

    os.system("pip install rich")

    from rich.console import Console

    from rich.table import Table

console = Console()

table = Table(show\_header=True, header\_style="bold magenta")

os.system("cls")

q = 4 # Quantum Time

t = 0 # Current Time

nprocess = int(input("Enter the number of processes: "))

bt\_rem = [] # Burst Time Remaining

for i in range(nprocess):

    bt = int(input("Enter the burst time for P[{}]: ".format(i+1)))

    bt\_rem.append(bt)

ct = [0 for i in range(nprocess)]

temp = bt\_rem.copy()

waiting\_time = []

turnaround\_time = []

while 1:

    done = True

    for i in range(0, 3):

        if bt\_rem[i] > 0:

            done = False

            if bt\_rem[i] > q:

                t += q

                bt\_rem[i] -= q

            else:

                t += bt\_rem[i]

                ct[i] = t

                bt\_rem[i] = 0

    if done == True:

        break

table.add\_column("PId", justify="center")

table.add\_column("Arrival Time", justify="center")

table.add\_column("BurstTime", justify="center")

table.add\_column("CompletionTime", justify="center")

table.add\_column("TurnAround Time", justify="center")

table.add\_column("Waiting Time", justify="center")

for i in range(0, 3):

    table.add\_row(str(i+1), str(0), str(temp[i]), str(ct[i]), str(ct[i]-0), str(ct[i]-temp[i]))

    waiting\_time.append(ct[i]-temp[i])

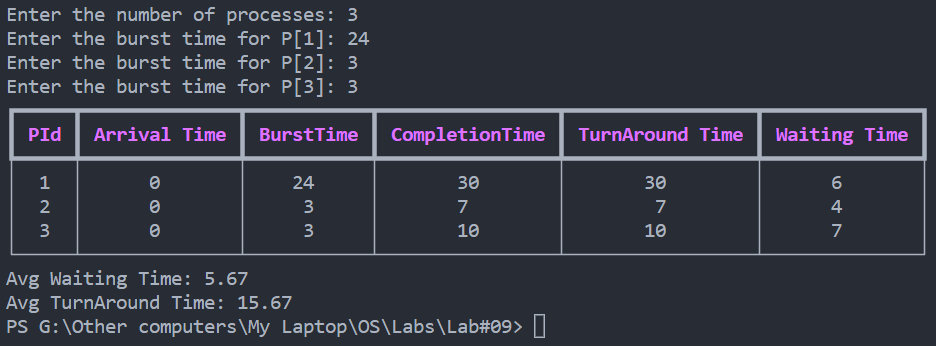
    turnaround\_time.append(ct[i]-0)

console.print(table)

print("Avg Waiting Time:", round(sum(waiting\_time)/3, 2))

print("Avg TurnAround Time:", round(sum(turnaround\_time)/3, 2))

Output:



# Priority Algorithm:

import os

try:

    from rich.console import Console

    from rich.table import Table

except ImportError:

    os.system("pip install rich")

    from rich.console import Console

    from rich.table import Table

console = Console()

table = Table(show\_header=True, header\_style="bold magenta")

os.system("cls")

n = int(input("Enter the number of processes: "))

processes = []

CT = []

TAT = []

WT = []

for i in range(n):

    b = int(input("Burst Time: "))

    pr = int(input("Priority no: "))

    processes.append(["P"+str(i+1), 0, b, pr])

# sort According to prioriry

processes.sort(key=lambda x: x[3])

# Calculting Completion time

for i in range(len(processes)):

    if i == 0:

        if processes[i][1] > 0:

            state\_idle = processes[i][1]

            CT.append(processes[i][2]+state\_idle)

        else:

            CT.append(processes[i][2])

    else:

        if CT[i-1] < processes[i][1]:

            idle\_state = processes[i][1] - CT[i-1]

            CT.append(CT[i-1]+processes[i][2]+idle\_state)

        else:

            CT.append(CT[i-1]+processes[i][2])

# Calculation Turn Around Time

for i in range(len(processes)):

    TAT.append(CT[i]-processes[i][1])

# Calculation Waiting Time

for i in range(len(processes)):

    WT.append(TAT[i]-processes[i][2])

table.add\_column("PId", justify="center")

table.add\_column("Arrival Time", justify="center")

table.add\_column("BurstTime", justify="center")

table.add\_column("Priority", justify="center")

table.add\_column("CompletionTime", justify="center")

table.add\_column("TurnAround Time", justify="center")

table.add\_column("Waiting Time", justify="center")

for i in range(len(processes)):

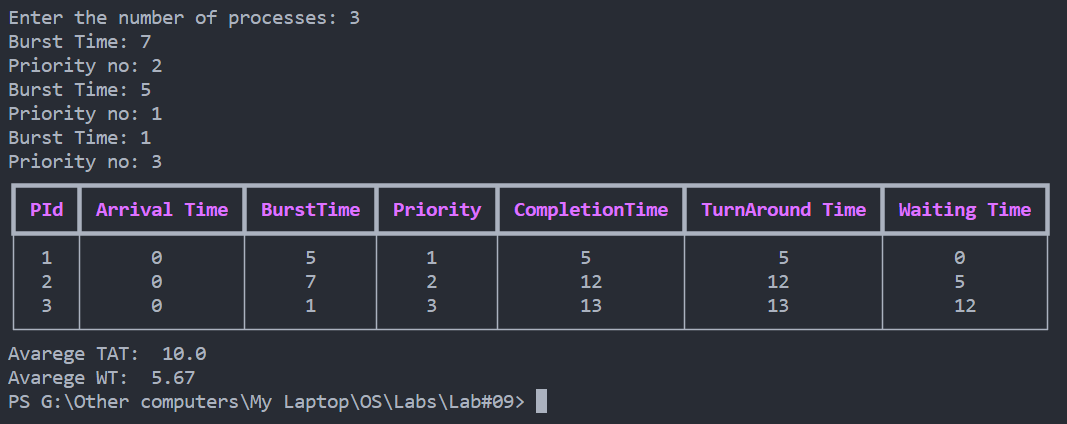
    table.add\_row(str(i+1), str(processes[i][1]), str(processes[i][2]), str(processes[i][3]), str(CT[i]), str(TAT[i]), str(WT[i]))

console.print(table)

print("Avarege TAT: ", round(sum(TAT)/len(TAT), 2))

print("Avarege WT: ", round(sum(WT)/len(WT), 2))

Output:



# Priority Algorithm (with different arrival time):

import os

try:

    from rich.console import Console

    from rich.table import Table

except ImportError:

    os.system("pip install rich")

    from rich.console import Console

    from rich.table import Table

console = Console()

table = Table(show\_header=True, header\_style="bold magenta")

os.system("cls")

n = int(input("Enter the number of processes: "))

processes = []

Sorted = []

CT = []

TAT = []

WT = []

for i in range(n):

    a = int(input("Arrival time: "))

    b = int(input("Burst Time: "))

    pr = int(input("Priority no: "))

    processes.append(["P"+str(i+1), a, b, pr])

n = len(processes)

# arranging

t = min(processes, key=lambda x: x[1])

t = t[1]

for i in range(n):

    reach\_pro = []

    flag = True

    while flag == True:

        for j in range(len(processes)):

            if processes[j][1] <= t:

                reach\_pro.append(processes[j])

        if len(reach\_pro) == 0:

            t += 1

        else:

            flag = False

    least\_p = min(reach\_pro, key=lambda x: x[3])

    t = t + least\_p[2]

    Sorted.append(least\_p)

    processes.remove(least\_p)

# Calculting Completion time

for i in range(len(Sorted)):

    if i == 0:

        if Sorted[i][1] > 0:

            state\_idle = Sorted[i][1]

            CT.append(Sorted[i][2]+state\_idle)

        else:

            CT.append(Sorted[i][2])

    else:

        if CT[i-1] < Sorted[i][1]:

            idle\_state = Sorted[i][1] - CT[i-1]

            CT.append(CT[i-1]+Sorted[i][2]+idle\_state)

        else:

            CT.append(CT[i-1]+Sorted[i][2])

# Calculation Turn Around Time

for i in range(len(Sorted)):

    TAT.append(CT[i]-Sorted[i][1])

# Calculation Waiting Time

for i in range(len(Sorted)):

    WT.append(TAT[i]-Sorted[i][2])

table.add\_column("PId", justify="center")

table.add\_column("Arrival Time", justify="center")

table.add\_column("BurstTime", justify="center")

table.add\_column("Priority", justify="center")

table.add\_column("CompletionTime", justify="center")

table.add\_column("TurnAround Time", justify="center")

table.add\_column("Waiting Time", justify="center")

for i in range(len(Sorted)):

    table.add\_row(str(Sorted[i][0]), str(Sorted[i][1]), str(Sorted[i][2]), str(Sorted[i][3]), str(CT[i]), str(TAT[i]), str(WT[i]))

console.print(table)

print("Avarege TAT: ", round(sum(TAT)/len(TAT), 2))

print("Avarege WT: ", round(sum(WT)/len(WT), 2))

Output:

