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

Muhammad Waleed
20b-115-se
SE-B
OS Lab#09
Ma’am Shabina Mushtaq

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
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#09
Ma’am Shabina Mushtaq

```
            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:

```
Enter the number of processes: 3
Enter the burst time for P[1]: 24
Enter the burst time for P[2]: 3
Enter the burst time for P[3]: 3
```

PId	Arrival Time	BurstTime	CompletionTime	TurnAround Time	Waiting Time
1	0	24	30	30	6
2	0	3	7	7	4
3	0	3	10	10	7

```
Avg Waiting Time: 5.67
Avg TurnAround Time: 15.67
PS G:\Other computers\My Laptop\OS\Labs\Lab#09>
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#09
Ma’am Shabina Mushtaq

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]
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#09
Ma’am Shabina Mushtaq

```
            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:

Muhammad Waleed
20b-115-se
SE-B
OS Lab#09
Ma’am Shabina Mushtaq

```
Enter the number of processes: 3
Burst Time: 7
Priority no: 2
Burst Time: 5
Priority no: 1
Burst Time: 1
Priority no: 3
```

PId	Arrival Time	BurstTime	Priority	CompletionTime	TurnAround Time	Waiting Time
1	0	5	1	5	5	0
2	0	7	2	12	12	5
3	0	1	3	13	13	12

```
Avarege TAT:  10.0
Avarege WT:  5.67
PS G:\Other computers\My Laptop\OS\Labs\Lab#09>
```

Priority Algorithm (with different arrival time):

```python
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])
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#09
Ma’am Shabina Mushtaq

```
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])
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#09
Ma’am Shabina Mushtaq

```
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:

```
Enter the number of processes: 3
Arrival time: 1
Burst Time: 7
Priority no: 1
Arrival time: 1
Burst Time: 5
Priority no: 3
Arrival time: 2
Burst Time: 1
Priority no: 2
```

PId	Arrival Time	BurstTime	Priority	CompletionTime	TurnAround Time	Waiting Time
P1	1	7	1	8	7	0
P3	2	1	2	9	7	6
P2	1	5	3	14	13	8

```
Avarege TAT:  9.0
Avarege WT:  4.67
```