



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: 24-Nov-2022

Muhammad Waleed
20b-115-se
SE-B
OS Lab#08
Ma'am Shabina Mushtaq

FCFS (with arrival time 0):

```
import os
try:
    from rich.console import Console
    from rich.table import Table
except ModuleNotFoundError:
    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")

nprocess = int(input("Enter number of processes: "))
processes = []
CT = []
TAT = []
WT = []
for i in range(nprocess):
    b = int(input("Burst Time: "))
    processes.append(["P"+str(i+1), 0, b])

# sort According to arrival time
processes.sort(key=lambda x: x[1])
# Calculating Completion time
for i in range(len(processes)):
    if i == 0:
        CT.append(processes[i][2])
    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("Process", justify="center")
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#08
Ma'am Shabina Mushtaq

```
table.add_column("Arrival Time", justify="center")
table.add_column("Burst Time", justify="center")
table.add_column("Completion Time", justify="center")
table.add_column("Turn Around Time", justify="center")
table.add_column("Waiting Time", justify="center")

for i in range(len(processes)):
    table.add_row(str(processes[i][0]), str(processes[i][1]), str(
        processes[i][2]), 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 number of processes: 4
Burst Time: 5
Burst Time: 4
Burst Time: 3
Burst Time: 2
```

Process	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	0	5	5	5	0
P2	0	4	9	9	5
P3	0	3	12	12	9
P4	0	2	14	14	12

```
Avarege TAT: 10.0
Avarege WT: 6.5
PS G:\Other computers\My Laptop\OS\Labs\Lab#08> 
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#08
Ma'am Shabina Mushtaq

SJF (with arrival time 0):

```
import os
try:
    from rich.console import Console
    from rich.table import Table
except ModuleNotFoundError:
    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")

nprocess = int(input("Enter number of processes: "))
processes = []
CT = []
TAT = []
WT = []
for i in range(nprocess):
    b = int(input("Burst Time: "))
    processes.append(["P"+str(i+1), 0, b])

# sort According to burst time
processes.sort(key=lambda x: x[2])

# Calculating 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])
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#08
Ma'am Shabina Mushtaq

```
# 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("Process", justify="center")
table.add_column("Arrival Time", justify="center")
table.add_column("Burst Time", justify="center")
table.add_column("Completion Time", justify="center")
table.add_column("Turn Around Time", justify="center")
table.add_column("Waiting Time", justify="center")

for i in range(len(processes)):
    table.add_row(str(processes[i][0]), str(processes[i][1]), str(
        processes[i][2]), 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 number of processes: 4
Burst Time: 2
Burst Time: 7
Burst Time: 1
Burst Time: 3
```

Process	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P3	0	1	1	1	0
P1	0	2	3	3	1
P4	0	3	6	6	3
P2	0	7	13	13	6

Avarege TAT: 5.75

Avarege WT: 2.5

PS G:\Other computers\My Laptop\OS\Labs\Lab#08>

Muhammad Waleed
20b-115-se
SE-B
OS Lab#08
Ma'am Shabina Mushtaq

Modified FCFS for different arrival time and idleness:

```
import os
try:
    from rich.console import Console
    from rich.table import Table
except ModuleNotFoundError:
    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")

nprocess = int(input("Enter number of processes: "))
processes = []
CT = []
TAT = []
WT = []
for i in range(nprocess):
    a = int(input("Arrival time: "))
    b = int(input("Burst Time: "))
    processes.append(["P"+str(i+1), a, b])

# sort According to arrival time
processes.sort(key=lambda x: x[1])
# Calculating 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])
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#08
Ma'am Shabina Mushtaq

```
# 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("Process", justify="center")
table.add_column("Arrival Time", justify="center")
table.add_column("Burst Time", justify="center")
table.add_column("Completion Time", justify="center")
table.add_column("Turn Around Time", justify="center")
table.add_column("Waiting Time", justify="center")

for i in range(len(processes)):
    table.add_row(str(processes[i][0]), str(processes[i][1]), str(
        processes[i][2]), 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 number of processes: 4
Arrival time: 0
Burst Time: 4
Arrival time: 2
Burst Time: 1
Arrival time: 3
Burst Time: 7
Arrival time: 5
Burst Time: 7
```

Process	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	0	4	4	4	0
P2	2	1	5	3	2
P3	3	7	12	9	2
P4	5	7	19	14	7

Avarege TAT: 7.5

Avarege WT: 2.75

PS G:\Other computers\My Laptop\OS\Labs\Lab#08> █

Muhammad Waleed
20b-115-se
SE-B
OS Lab#08
Ma'am Shabina Mushtaq

SJF (with different arrival time):

```
import os
try:
    from rich.console import Console
    from rich.table import Table
except ModuleNotFoundError:
    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")

nprocess = int(input("Enter number of processes: "))
processes = []
Sorted = []
CT = []
TAT = []
WT = []
for i in range(nprocess):
    a = int(input("Arrival time: "))
    b = int(input("Burst Time: "))
    processes.append(["P"+str(i+1), a, b])

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_bt = min(reach_pro, key=lambda x: x[2])
```


Muhammad Waleed
20b-115-se
SE-B
OS Lab#08
Ma'am Shabina Mushtaq

```
t = t + least_bt[2]
Sorted.append(least_bt)
processes.remove(least_bt)

# Calculating 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("Process", justify="center")
table.add_column("Arrival Time", justify="center")
table.add_column("Burst Time", justify="center")
table.add_column("Completion Time", justify="center")
table.add_column("Turn Around 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(CT[i]), str(TAT[i]), str(WT[i]))

console.print(table)
```

Muhammad Waleed
20b-115-se
SE-B
OS Lab#08
Ma'am Shabina Mushtaq

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

Output:

```
Enter number of processes: 3  
Arrival time: 1  
Burst Time: 2  
Arrival time: 3  
Burst Time: 4  
Arrival time: 4  
Burst Time: 7
```

Process	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	1	2	3	2	0
P2	3	4	7	4	0
P3	4	7	14	10	3

```
Avarege TAT: 5.33
```

```
Avarege WT: 1.0
```

```
PS G:\Other computers\My Laptop\OS\Labs\Lab#08> 
```