

# **Scheduling Algorithms Using Excel File**

**Sardar Muhammad  
Zeeshan Khan**

11/11/2022

—

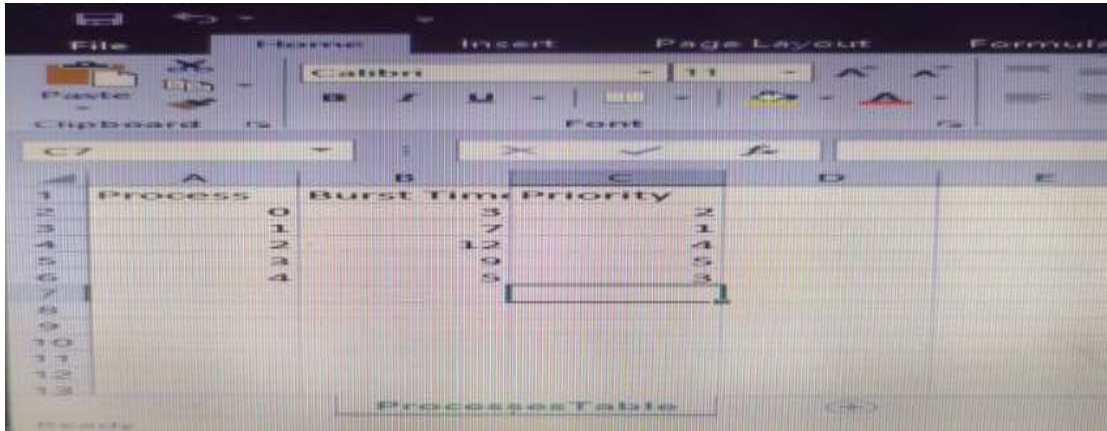
Operating Systems

—

Sir Waqas Ahmed

---

## Excel File



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E
1	Process	Burst Time	Priority		
2	0	3	2		
3	1	7	1		
4	2	12	4		
5	3	9	5		
6	4	5	3		

The spreadsheet is titled 'ProcessesTable' and is located in the 'ProcessesTable' worksheet.

## Shortest Job First

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help SchedulingAlgorithms - main.py
SchedulingAlgorithms - main.py
main.py
1 import pandas as pd
2 path = r"D:\Pycharm\SchedulingAlgorithms\OS-ProcessesTable.csv"
3 functionn = pd.read_csv(path)
4 print("Shortest Job First")
5
6 process = functionn["Process"].tolist()
7
8 Execution_Time = functionn["Burst Time"].tolist()
9
10 Execution_Time.sort()
11
12 WT = 0
13 ET = 0
14 for x in range(len(process)):
15     print("Waiting Time Of Process :", process[x], "=", ET)
16     ET = ET + Execution_Time[x]
17     print("Execution Time of Process :", process[x], "=", ET)
18     if x < len(process)-1:
19         WT = WT + ET
20
21
22 print(WT)
23 print("Average Waiting Time = ", WT/len(process))
```

## Output

```
*****Shortest Job First*****
Waiting Time Of Process : 0 = 0
Execution Time of Process : 0 = 3
Waiting Time Of Process : 1 = 3
Execution Time of Process : 1 = 8
Waiting Time Of Process : 2 = 8
Execution Time of Process : 2 = 15
Waiting Time Of Process : 3 = 15
Execution Time of Process : 3 = 24
Waiting Time Of Process : 4 = 24
Execution Time of Process : 4 = 36
50
Average Waiting Time = 10.0
```

# First Come First Served

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help SchedulingAlgorithms - main.py
SchedulingAlgorithms main.py
1 import pandas as pd
2 path = r"D:\Pycharm\SchedulingAlgorithms\OS-ProcessesTable.csv"
3 functionn = pd.read_csv(path)
4 print("First Come First Serve")
5
6 process = functionn["Process"].tolist()
7
8 Execution_Time = functionn["Burst Time"].tolist()
9
10
11 WT = 0
12 ET = 0
13 for x in range(len(process)):
14     print("Waiting Time Of Process   :", process[x], "=", ET)
15     ET = ET + Execution_Time[x]
16     print("Execution Time of Process :", process[x], "=", ET)
17     if x < len(process)-1:
18         WT = WT + ET
19
20
21 print(WT)
22 print("Average Waiting Time = ", WT/len(process))
```

## Output

```
First Come First Serve
Waiting Time Of Process   : 0 = 0
Execution Time of Process : 0 = 3
Waiting Time Of Process   : 1 = 3
Execution Time of Process : 1 = 10
Waiting Time Of Process   : 2 = 10
Execution Time of Process : 2 = 22
Waiting Time Of Process   : 3 = 22
Execution Time of Process : 3 = 31
Waiting Time Of Process   : 4 = 31
Execution Time of Process : 4 = 36
66
Average Waiting Time =      13.2
```

# Priority Scheduling

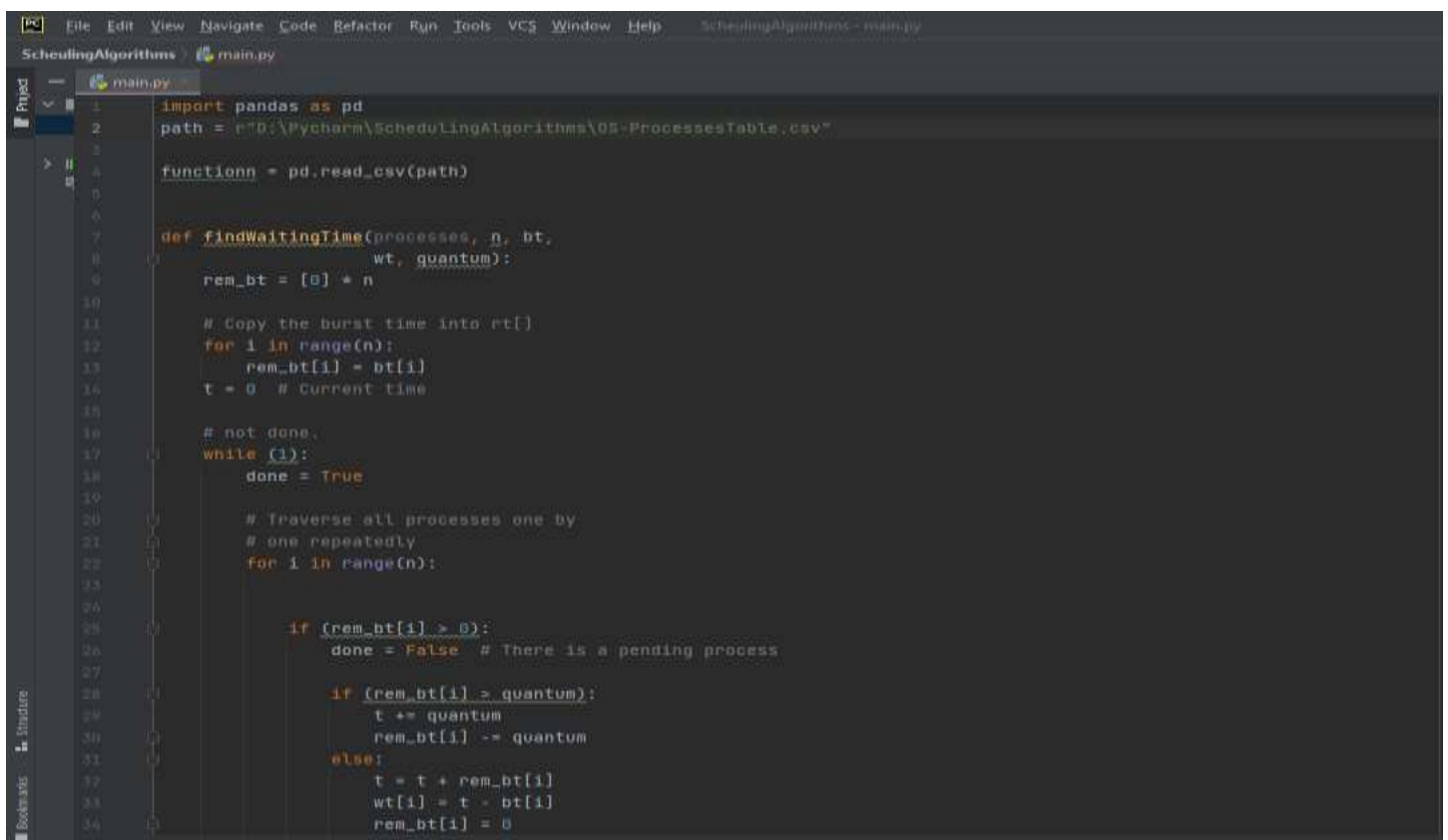
```
SchedulingAlgorithms - main.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
SchedulingAlgorithms main.py
Project main.py
1 #To read excel file
2 import pandas as pd
3 path = r"D:\Pycharm\SchedulingAlgorithms\05-ProcessesTable.csv"
4 functionn = pd.read_csv(path)
5
6
7 def findWaitingTime(processes, n, wt):
8     wt[0] = 0
9
10    # calculating waiting time
11    for i in range(1, n):
12        wt[i] = processes[i - 1][1] + wt[i - 1]
13
14    # Function to calculate turn around time
15    def findTurnAroundTime(processes, n, wt, tat):
16        # Calculating turnaround time by
17        # adding bt[i] + wt[i]
18        for i in range(len(processes)):
19            tat[i] = processes[i][1] + wt[i]
20
21    # Function to calculate average waiting
22    # and turn-around times.
23    def findavgTime(processes, n):
24        wt = [0] * n
25        tat = [0] * n
26
27        # Function to find waiting time
28        # of all processes
29        findWaitingTime(processes, n, wt)
30
31        # Function to find turn around time
32        # for all processes
33        findTurnAroundTime(processes, n, wt, tat)
34
35
36    # Display processes along with all details
37    print("\nProcesses| Burst Time | Waiting",
38          "Time")
39    total_wt = 0
40
41    for i in range(n):
42        total_wt = total_wt + wt[i]
43
44        print(" ", processes[i][0], "\t\t",
45              processes[i][1], "\t\t", wt[i])
46
47    print("\nAverage waiting time = " + str(total_wt / n))
48
49
50 def priorityScheduling(proc, n):
51     # Sort processes by priority
52     proc = sorted(proc, key=lambda proc: proc[2],
53                  reverse=True)
54
55     print("Order in which processes gets executed")
56     for i in proc:
57         print(i[0], end=" ")
58     findavgTime(proc, n)
59
60
61 # Driver code
62 if __name__ == "__main__":
63     # Process id's
64     proc = [functionn["Process"].tolist(),
65            functionn["Burst Time"].tolist(),
66            functionn["Priority"].tolist()]
67     n = 3
68     priorityScheduling(proc, n)
```

# Output

```
PS C:\Users\adell> python3 .\main.py
Order in which processes gets executed
3 2 0
Processes| Burst Time | Waiting Time
3         7         0
2         1         7
0         1         8

Average waiting time = 5.0
```

## Round Robin



```
1 import pandas as pd
2 path = r"D:\Pycharm\SchedulingAlgorithms\OS-ProcessesTable.csv"
3
4 functionn = pd.read_csv(path)
5
6
7 def findWaitingTime(processes, n, bt,
8                     wt, quantum):
9     rem_bt = [0] * n
10
11     # Copy the burst time into rt[]
12     for i in range(n):
13         rem_bt[i] = bt[i]
14     t = 0 # Current time
15
16     # not done.
17     while (1):
18         done = True
19
20         # Traverse all processes one by
21         # one repeatedly
22         for i in range(n):
23
24             if (rem_bt[i] > 0):
25                 done = False # There is a pending process
26
27                 if (rem_bt[i] > quantum):
28                     t += quantum
29                     rem_bt[i] -= quantum
30                 else:
31                     t = t + rem_bt[i]
32                     wt[i] = t - bt[i]
33                     rem_bt[i] = 0
34
```



```
PC File Edit View Navigate Code Refactor Run Tools VCS Window Help SchedulingAlgorithms : main.py
SchedulingAlgorithms : main.py
Project
main.py
35 for _ in range(1000000):
36     if (done == True):
37         break
38
39
40 def findTurnAroundTime(processes, n, bt, wt, tat):
41     for i in range(n):
42         tat[i] = bt[i] + wt[i]
43
44
45 def findavgTime(processes, n, bt, quantum):
46     wt = [0] * n
47     tat = [0] * n
48     findWaitingTime(processes, n, bt,
49                     wt, quantum)
50     findTurnAroundTime(processes, n, bt,
51                       wt, tat)
52     print("Processes Burst Time    Waiting",
53           "Time")
54     total_wt = 0
55
56     for i in range(n):
57         total_wt = total_wt + wt[i]
58
59         print(" ", i + 1, "\t\t", bt[i],
60               "\t\t", wt[i])
61
62     print("\nAverage waiting time = ", (total_wt / n))
63
64
65 if __name__ == "__main__":
66     # Process id's
67     proc = functionn["Process"].tolist()
68     n = 4
69
70     # Burst time of all processes
71     burst_time = functionn["Burst Time"].tolist()
72
73     # Time quantum
74     quantum = 2;
75     findavgTime(proc, n, burst_time, quantum)
76
```

## Output

Processes	Burst Time	Waiting Time
1	3	6
2	7	15
3	12	19
4	9	20
Average waiting time =		15.0