

## CPU Scheduling Programs

### 1) FCFS Scheduling Algorithm (First come First Serve):

CODE:

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

processes = []
for i in range(n):
    p = int(input("Enter Pid: "))
    at = eval(input("Enter Arrival time: "))
    bt = eval(input("Enter the burst time: "))
    processes.append({'pid': p, 'at': at, 'bt': bt})

processes = sorted(processes, key=lambda x: x['at'])

ct_l = []
tat_l = []
wt_l = []

c = 0
x = 0
for i in range(n):
    p = processes[i]
    bt = p['bt']
    at = p['at']
    if c == 0:
        ct = at + bt
        x = ct
    else:
        y = at - x
        ct = x + bt + y
    x = ct

    tat = ct - at
    wt = tat - bt

    ct_l.append(ct)
    tat_l.append(tat)
    wt_l.append(wt)

    c += 1

avg_tat = sum(tat_l) / n
```

## CPU Scheduling Programs

```

avg_wt = sum(wt_l) / n

for i in range(n):
    processes[i]['ct'] = ct_l[i]
    processes[i]['tat'] = tat_l[i]
    processes[i]['wt'] = wt_l[i]

print("Pid\tAT\tBT\tCT\tTAT\tWT")
for i in range(n):
    p = processes[i]
    print(f"{p['pid']}\t\t{p['at']}\t\t{p['bt']}\t\t{p['ct']}\t\t{p['tat']}\t\t{p['wt']}\t\t{p['wt']}\t\t{p['wt']}")

print("Average Turn Around Time: ", avg_tat)
print("Average Waiting Time: ", avg_wt)

```

### OUTPUT:

```

PS C:\Users\venka> & C:/Users/venka/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/venka/OneDrive/Desktop/fcfs.py
Enter the number of processes: 4
Enter Pid: 1
Enter Arrival time: 2
Enter the burst time: 3
Enter Pid: 2
Enter Arrival time: 1
Enter the burst time: 4
Enter Pid: 3
Enter Arrival time: 0
Enter the burst time: 2
Enter Pid: 4
Enter Arrival time: 4
Enter the burst time: 1
  Pid  Arrival Time  Burst Time
2   3           0           2
1   2           1           4
0   1           2           3
3   4           4           1
Pid      3
Name: 3, dtype: int64
  Pid  Arrival Time  Burst Time  Completion time  Turn Around Time  \
2   3           0           2             2             2
1   2           1           4             5             4
0   1           2           3             5             3
3   4           4           1             5             1

  Waiting time
2           0
1           0
0           0
3           0
Average Turn Around Time:  2.5

```

## CPU Scheduling Programs

### 2) SJF (Shortest job first):

CASE 1: Without Arrival Time

CODE:

```
n = int(input("Enter number of process: "))
bt = []
p = []
wt = []
tat = []
total = 0
totalT = 0

print("\nEnter Burst Time:")
for i in range(n):
    burst_time = int(input("p{}:".format(i+1)))
    bt.append(burst_time)
    p.append(i+1)

for i in range(n):
    pos = i
    for j in range(i+1, n):
        if bt[j] < bt[pos]:
            pos = j

    temp = bt[i]
    bt[i] = bt[pos]
    bt[pos] = temp

    temp = p[i]
    p[i] = p[pos]
    p[pos] = temp

wt.append(0)

for i in range(1, n):
    wt.append(0)
    for j in range(i):
        wt[i] += bt[j]

    total += wt[i]
```

## CPU Scheduling Programs

```
avg_wt = total / n

print("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time")
for i in range(n):
    tat.append(bt[i] + wt[i])
    totalT += tat[i]

    print("p{}\t{}\t{}\t{}".format(p[i], bt[i], wt[i], tat[i]))

avg_tat = totalT / n

print("\nAverage Waiting Time =", avg_wt)
print("Average Turnaround Time =", avg_tat)
```

### OUTPUT:

```
PS C:\Users\venka> & C:/Users/venka/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/venka/OneDrive/Desktop/sjf.py
Enter number of process: 4

Enter Burst Time:
p1:5
p2:4
p3:12
p4:7

Process Burst Time    Waiting Time    Turnaround Time
p2          4           0             4
p1          5           4             9
p4          7           9            16
p3         12          16            28

Average Waiting Time = 7.25
Average Turnaround Time = 14.25
PS C:\Users\venka> |
```

### CASE 2: With Arrival Time

#### CODE:

```
print('Enter number of processes')
n, proc_in, time, proc_queue = int(input()), [], 0, []

for i in range(n):
    proc = []
    print('Enter PID')
    proc.append(input())
    print('Enter arrival time')
    proc.append(int(input()))
    print('Enter execution time')
    proc.append(int(input()))
    proc_in.append(proc)
```

## CPU Scheduling Programs

```
proc_in.sort(key=lambda proc_in: proc_in[1])
proc_queue = []

foo = []
i = 0
while i < n:
    at = proc_in[i][1]
    while i < n and proc_in[i][1] == at:
        foo.append(proc_in[i])
        i = i + 1
    foo.sort(key=lambda foo: foo[2])
    for proc in foo:
        proc_queue.append(proc)
    foo = []

for proc in proc_queue:
    time = time if time > proc[1] else proc[1]
    time = time + proc[2]
    proc.append(time)
    proc.append(proc[3] - proc[2] - proc[1] if time > proc[1] else 0)

print('\n')
print('{:<4} {:<12} {:<12} {:<12} {:<12}'.format(*'PID_Arrival Time_Burst
Time_Waiting Time_Finish Time'.split('_')))
for proc in proc_queue:
    print('{:<4} {:<12} {:<12} {:<12} {:<12}'.format(proc[0], proc[1],
proc[2], proc[4], proc[3]))
```

## CPU Scheduling Programs

### Output:

```
PS C:\Users\venka> & C:/Users/venka/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/venka/OneDrive/Desktop/sjfl.py
Enter number of processes
Enter arrival time
1
Enter execution time
6
Enter PID
2
Enter arrival time
2
Enter execution time
3
Enter PID
3
Enter arrival time
3
Enter execution time
2
Enter PID
4
Enter arrival time
4
Enter execution time
3
Enter PID
5
Enter arrival time
5
Enter execution time
4

PID Arrival Time Burst Time Waiting Time Finish Time
1 1 6 0 7
2 2 3 5 10
3 3 2 7 12
4 4 3 8 15
5 5 4 10 19
PS C:\Users\venka> |
```

### 3) Round Robin:

#### CODE:

```
n = int(input("Enter Total Process:\t "))
at = []
bt = []
rt = []
for i in range(n):
    arrival_time, burst_time = map(int, input(f"Enter Arrival Time and Burst Time for Process Process Number {i+1}: ").split())
    at.append(arrival_time)
    bt.append(burst_time)
    rt.append(burst_time)

tq = int(input("Enter Time Quantum:\t"))
print("\n\nProcess\t|Burst Time|Turnaround Time|Waiting Time\n\n")
```

## CPU Scheduling Programs

```
wt = 0
tat = 0
remain = n
flag = 0
t = 0
i = 0

while remain != 0:
    if rt[i] <= tq and rt[i] > 0:
        t += rt[i]
        rt[i] = 0
        flag = 1
    elif rt[i] > 0:
        rt[i] -= tq
        t += tq
    if rt[i] == 0 and flag == 1:
        remain -= 1
        print(f"P[{i+1}]\t|\t{bt[i]}\t|\t{t-at[i]}\t|\t{t-at[i]-bt[i]}")
        wt += t-at[i]-bt[i]
        tat += t-at[i]
        flag = 0
    if i == n-1:
        i = 0
    elif at[i+1] <= t:
        i += 1
    else:
        i = 0

print(f"\nAverage Waiting Time = {wt/n:.2f}")
print(f"Average Turnaround Time = {tat/n:.2f}")
```

## CPU Scheduling Programs

### OUTPUT:

#### Case-1: With Zero Arrival Time

```
PS C:\Users\venka\onedrive\desktop> python rr.py
Enter Total Process:      3
Enter Arrival Time and Burst Time for Process Process Number 1: 0 4
Enter Arrival Time and Burst Time for Process Process Number 2: 0 3
Enter Arrival Time and Burst Time for Process Process Number 3: 0 5
Enter Time Quantum:      2

Process |Burst Time|Turnaround Time|Waiting Time

P[1]    |      4    |      8    |      4
P[2]    |      3    |      9    |      6
P[3]    |      5    |     12    |      7

Average Waiting Time = 5.67
Average Turnaround Time = 9.67
PS C:\Users\venka\onedrive\desktop> █
```

#### Case-2: With Sequential Arrival Time

```
PS C:\Users\venka\onedrive\desktop> python rr.py
Enter Total Process:      4
Enter Arrival Time and Burst Time for Process Process Number 1: 0 8
Enter Arrival Time and Burst Time for Process Process Number 2: 1 5
Enter Arrival Time and Burst Time for Process Process Number 3: 2 10
Enter Arrival Time and Burst Time for Process Process Number 4: 3 11
Enter Time Quantum:      6

Process |Burst Time|Turnaround Time|Waiting Time

P[2]    |      5    |     10    |      5
P[1]    |      8    |     25    |     17
P[3]    |     10    |     27    |     17
P[4]    |     11    |     31    |     20

Average Waiting Time = 14.75
Average Turnaround Time = 23.25
PS C:\Users\venka\onedrive\desktop> █
```



## CPU Scheduling Programs

### 4) Priority Queue:

CODE:

```
def swap(a, b):
    temp = a
    a = b
    b = temp
    return a, b

n = int(input("Enter Number of Processes: "))
b = [0]*n
p = [0]*n
index = [0]*n
for i in range(n):
    burst, priority = map(int, input(f"Enter Burst Time and Priority Value for Process {i+1}: ").split())
    b[i] = burst
    p[i] = priority
    index[i] = i+1

for i in range(n):
    a = p[i]
    m = i
    for j in range(i, n):
        if p[j] > a:
            a = p[j]
            m = j
    p[i], p[m] = swap(p[i], p[m])
    b[i], b[m] = swap(b[i], b[m])
    index[i], index[m] = swap(index[i], index[m])

t = 0
print("Order of process Execution is")
for i in range(n):
    print(f"P{index[i]} is executed from {t} to {t+b[i]}")
    t += b[i]

print("\nProcess Id      Burst Time    Wait Time    TurnAround Time")
wait_time = 0
for i in range(n):
```

## CPU Scheduling Programs

```
print(f"P{index[i]}\t\t\t| \t{b[i]}\t\t\t| \t{wait_time}\t\t\t| \t{wait_t  
ime + b[i]}\n")  
wait_time += b[i]
```

### OUTPUT:

```
PS C:\Users\venka> & C:/Users/venka/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/venka/OneDrive/Desktop/pq.py
Enter Number of Processes: 4
Enter Burst Time and Priority Value for Process 1: 5 1
Enter Burst Time and Priority Value for Process 2: 7 6
Enter Burst Time and Priority Value for Process 3: 2 4
Enter Burst Time and Priority Value for Process 4: 3 5
Order of process Execution is
P2 is executed from 0 to 12
P3 is executed from 10 to 12
P1 is executed from 12 to 17
```

Process Id	Burst Time	Wait Time	TurnAround Time
P4	3	7	10
P3	2	10	12
P1	5	12	17

```
PS C:\Users\venka>
```