| **Fr. Conceicao Rodrigues College of Engineering**  **Department of Computer Engineering** | | | |
| --- | --- | --- | --- |
| **Student’s Roll No** | **9914** | **Students Name** | **Vivian Ludrick** |
| **Date of Performance** | **22/02/2024** | **SE Computer – Div** | **B Batch C** |

**Aim:** Study Process Scheduling

**Lab Outcome:**

**CSL403.2:** Implement various Process scheduling algorithm and evaluate their performance.

**Problem Statements:**

Batch (A): First Come First Serve (FCFS) ,Non Preemptive Shortest Job First (SJF)

Batch (B): Non Preemptive Shortest Job First (SJF) ,Shortest Remaining Time First (SRTF)

Batch (C ): Round Robin Algorithm (RR), Non Preemptive Priority (NPP)

Batch (D): Non Preemptive Priority (NPP), Premptive Priority (PP)

1. Calculate WT, AWT, TAT, ATAT.

2. Compare the result of algorithms for a problem and find which algorithm is performing better.

**Round Robin Algorithm(RR):**

CODE:

*# variable*

bucket\_list = [

{

"id": 1,

"new\_buckets\_added\_at": 10,

"no\_of\_bubbles": 5,

"temp\_no\_of\_bubbles": 5,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{*# variable*

bucket\_list = [

{

"id": 1,

"new\_buckets\_added\_at": 10,

"no\_of\_bubbles": 5,

"temp\_no\_of\_bubbles": 5,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 2,

"new\_buckets\_added\_at": 90,

"no\_of\_bubbles": 20,

"temp\_no\_of\_bubbles": 20,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 3,

"new\_buckets\_added\_at": 75,

"no\_of\_bubbles": 45,

"temp\_no\_of\_bubbles": 45,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 4,

"new\_buckets\_added\_at": 60,

"no\_of\_bubbles": 30,

"temp\_no\_of\_bubbles": 30,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 5,

"new\_buckets\_added\_at": 40,

"no\_of\_bubbles": 15,

"temp\_no\_of\_bubbles": 15,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 6,

"new\_buckets\_added\_at": 30,

"no\_of\_bubbles": 60,

"temp\_no\_of\_bubbles": 60,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

]

time\_quantum = 3

completed\_buckets = []

ongoing = []

time\_elapsed = 0

average\_bucket\_life\_span = 0

average\_wait\_time = 0

print("Gantt chart:\n")

*# logic*

while bucket\_list or ongoing:

*# add the new processes in the ongoing list*

for bucket in bucket\_list:

if bucket["new\_buckets\_added\_at"] <= time\_elapsed:

ongoing.append(bucket)

bucket\_list.remove(bucket)

*# time\_elapsed += time\_quantum*

if ongoing:

if ongoing[0]["temp\_no\_of\_bubbles"] > 0:

ongoing.append(ongoing.pop(0))

else:

*# print(f"{process['id']} : {time\_elapsed}")*

ongoing[0]["bucket\_empty\_time"] = time\_elapsed

*# print(f"|{process['id']} : {time\_elapsed}|", end="\t")*

completed\_buckets.append(ongoing.pop(0))

if ongoing:

process = ongoing[0]

time\_delta = min(time\_quantum, process["temp\_no\_of\_bubbles"])

*# print(f"{process['id']} : {time\_elapsed}")*

time\_elapsed += time\_delta

process["temp\_no\_of\_bubbles"] -= time\_delta

print(f"|{process['id']} : {time\_elapsed}|", *end*="\t")

else:

time\_elapsed += 1

for bucket in completed\_buckets:

bucket["bucket\_life\_span"] = (

bucket["bucket\_empty\_time"] - bucket["new\_buckets\_added\_at"]

)

bucket["wait\_time"] = bucket["bucket\_life\_span"] - bucket["no\_of\_bubbles"]

average\_bucket\_life\_span += bucket["bucket\_life\_span"]

average\_wait\_time += bucket["wait\_time"]

average\_bucket\_life\_span /= len(completed\_buckets)

average\_wait\_time /= len(completed\_buckets)

completed\_buckets.sort(*key*=lambda *bucket*: bucket["id"])

print("\n\nProcess List:\n")

print(

"-----------\t-------------\t-----------\t----------------\t-----------------\t----------\t-"

)

print(

"|Process ID\t|Arrival Time\t|Burst Time\t|Completion Time\t|Turn Around Time\t|Wait Time\t|"

)

print(

"|----------\t|------------\t|----------\t|---------------\t|----------------\t|---------\t|"

)

for bucket in completed\_buckets:

print(

f"|{bucket['id']}\t\t|{bucket['new\_buckets\_added\_at']}\t\t|{bucket['no\_of\_bubbles']}\t\t|{bucket['bucket\_empty\_time']}\t\t\t|{bucket['bucket\_life\_span']}\t\t\t|{bucket['wait\_time']}\t\t|"

)

print(f"\nAverage turn around time : {average\_bucket\_life\_span:.3f}")

print(f"Average wait time : {average\_wait\_time:.3f}")

"id": 2,

"new\_buckets\_added\_at": 90,

"no\_of\_bubbles": 20,

"temp\_no\_of\_bubbles": 20,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 3,

"new\_buckets\_added\_at": 75,

"no\_of\_bubbles": 45,

"temp\_no\_of\_bubbles": 45,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 4,

"new\_buckets\_added\_at": 60,

"no\_of\_bubbles": 30,

"temp\_no\_of\_bubbles": 30,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 5,

"new\_buckets\_added\_at": 40,

"no\_of\_bubbles": 15,

"temp\_no\_of\_bubbles": 15,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 6,

"new\_buckets\_added\_at": 30,

"no\_of\_bubbles": 60,

"temp\_no\_of\_bubbles": 60,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

]

time\_quantum = 3

completed\_buckets = []

ongoing = []

time\_elapsed = 0

average\_bucket\_life\_span = 0

average\_wait\_time = 0

print("Gantt chart:\n")

*# logic*

while bucket\_list or ongoing:

*# add the new processes in the ongoing list*

for bucket in bucket\_list:

if bucket["new\_buckets\_added\_at"] <= time\_elapsed:

ongoing.append(bucket)

bucket\_list.remove(bucket)

*# time\_elapsed += time\_quantum*

if ongoing:

if ongoing[0]["temp\_no\_of\_bubbles"] > 0:

ongoing.append(ongoing.pop(0))

else:

*# print(f"{process['id']} : {time\_elapsed}")*

ongoing[0]["bucket\_empty\_time"] = time\_elapsed

*# print(f"|{process['id']} : {time\_elapsed}|", end="\t")*

completed\_buckets.append(ongoing.pop(0))

if ongoing:

process = ongoing[0]

time\_delta = min(time\_quantum, process["temp\_no\_of\_bubbles"])

*# print(f"{process['id']} : {time\_elapsed}")*

time\_elapsed += time\_delta

process["temp\_no\_of\_bubbles"] -= time\_delta

print(f"|{process['id']} : {time\_elapsed}|", *end*="\t")

else:

time\_elapsed += 1

for bucket in completed\_buckets:

bucket["bucket\_life\_span"] = (

bucket["bucket\_empty\_time"] - bucket["new\_buckets\_added\_at"]

)

bucket["wait\_time"] = bucket["bucket\_life\_span"] - bucket["no\_of\_bubbles"]

average\_bucket\_life\_span += bucket["bucket\_life\_span"]

average\_wait\_time += bucket["wait\_time"]

average\_bucket\_life\_span /= len(completed\_buckets)

average\_wait\_time /= len(completed\_buckets)

completed\_buckets.sort(*key*=lambda *bucket*: bucket["id"])

print("\n\nProcess List:\n")

print(

"-----------\t-------------\t-----------\t----------------\t-----------------\t----------\t-"

)

print(

"|Process ID\t|Arrival Time\t|Burst Time\t|Completion Time\t|Turn Around Time\t|Wait Time\t|"

)

print(

"|----------\t|------------\t|----------\t|---------------\t|----------------\t|---------\t|"

)

for bucket in completed\_buckets:

print(

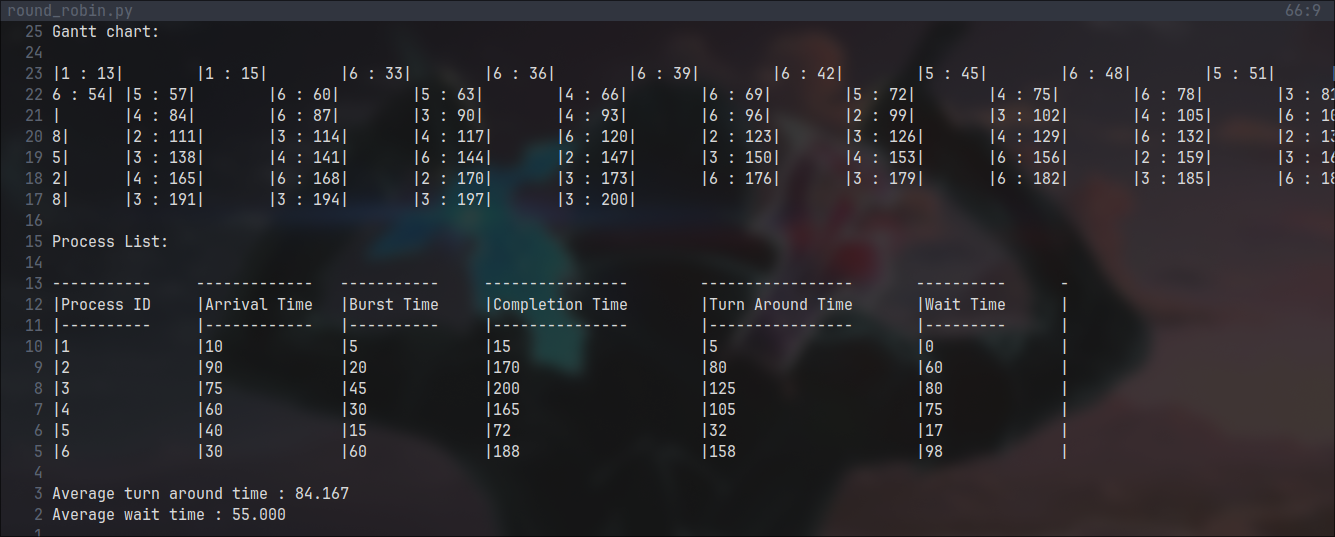
f"|{bucket['id']}\t\t|{bucket['new\_buckets\_added\_at']}\t\t|{bucket['no\_of\_bubbles']}\t\t|{bucket['bucket\_empty\_time']}\t\t\t|{bucket['bucket\_life\_span']}\t\t\t|{bucket['wait\_time']}\t\t|"

)

print(f"\nAverage turn around time : {average\_bucket\_life\_span:.3f}")

print(f"Average wait time : {average\_wait\_time:.3f}")

OUTPUT:



**Non Premptive Priority algorithm:**

CODE:

*# variable*

bucket\_list = [

{

"id": 1,

"priority": 1,

"new\_buckets\_added\_at": 10,

"no\_of\_bubbles": 5,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 2,

"priority": 6,

"new\_buckets\_added\_at": 90,

"no\_of\_bubbles": 20,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 3,

"priority": 4,

"new\_buckets\_added\_at": 75,

"no\_of\_bubbles": 45,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 4,

"priority": 2,

"new\_buckets\_added\_at": 60,

"no\_of\_bubbles": 30,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 5,

"priority": 9,

"new\_buckets\_added\_at": 40,

"no\_of\_bubbles": 15,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

{

"id": 6,

"priority": 5,

"new\_buckets\_added\_at": 30,

"no\_of\_bubbles": 60,

"bucket\_empty\_time": 0,

"bucket\_life\_span": 0,

"wait\_time": 0,

},

]

*# time\_quantum = 3*

completed\_buckets = []

ongoing = []

time\_elapsed = 0

average\_bucket\_life\_span = 0

average\_wait\_time = 0

print("Gantt chart:\n")

*# logic*

while bucket\_list or ongoing:

*# add the new processes to the execution sequence*

for bucket in bucket\_list:

if bucket["new\_buckets\_added\_at"] <= time\_elapsed:

ongoing.append(bucket)

bucket\_list.remove(bucket)

if ongoing:

ongoing.sort(*key*=lambda *bucket*: bucket["priority"])

time\_elapsed += ongoing[0]["no\_of\_bubbles"]

ongoing[0]["bucket\_empty\_time"] = time\_elapsed

print(f"|{ongoing[0]['id']} : {time\_elapsed}|", *end*="\t")

completed\_buckets.append(ongoing.pop(0))

else:

time\_elapsed += 1

for bucket in completed\_buckets:

bucket["bucket\_life\_span"] = (

bucket["bucket\_empty\_time"] - bucket["new\_buckets\_added\_at"]

)

bucket["wait\_time"] = bucket["bucket\_life\_span"] - bucket["no\_of\_bubbles"]

average\_bucket\_life\_span += bucket["bucket\_life\_span"]

average\_wait\_time += bucket["wait\_time"]

average\_bucket\_life\_span /= len(completed\_buckets)

average\_wait\_time /= len(completed\_buckets)

completed\_buckets.sort(*key*=lambda *bucket*: bucket["id"])

print("\n\nProcess List:\n")

print(

"-----------\t-------------\t-------------\t-----------\t----------------\t-----------------\t----------\t-"

)

print(

"|Process ID\t|Priority\t|Arrival Time\t|Burst Time\t|Completion Time\t|Turn Around Time\t|Wait Time\t|"

)

print(

"|----------\t-------------\t|------------\t|----------\t|---------------\t|----------------\t|---------\t|"

)

for bucket in completed\_buckets:

print(

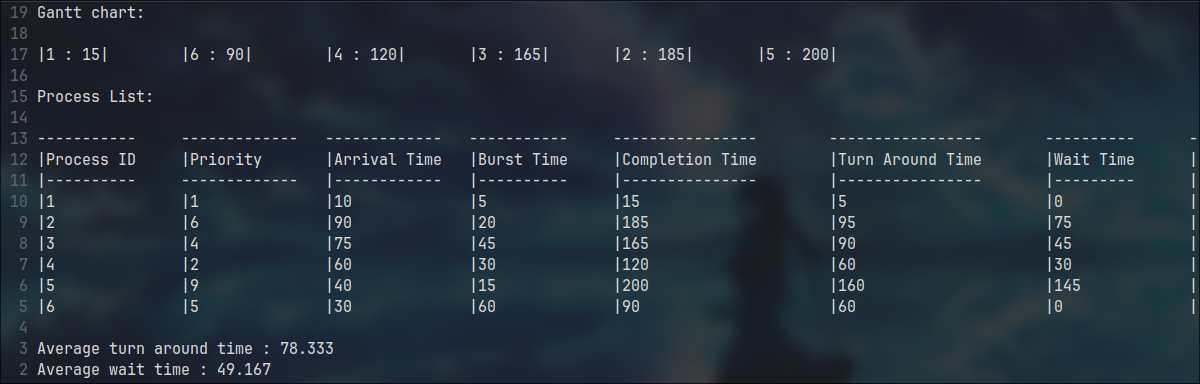
f"|{bucket['id']}\t\t|{bucket['priority']}\t\t|{bucket['new\_buckets\_added\_at']}\t\t|{bucket['no\_of\_bubbles']}\t\t|{bucket['bucket\_empty\_time']}\t\t\t|{bucket['bucket\_life\_span']}\t\t\t|{bucket['wait\_time']}\t\t|"

)

print(f"\nAverage turn around time : {average\_bucket\_life\_span:.3f}")

print(f"Average wait time : {average\_wait\_time:.3f}")

OUTPUT:

****

**Conclusion:**

Process ID Priority Arrival Time Burst Time

1 1 10 5

2 6 90 20

3 4 75 45

4 2 60 30

5 9 40 15

6 5 30 60

For the above inputs:

1. Round Robin:

* Average TAT = 84.167
* Average WT =55.008

1. Round Robin:

* Average TAT = 78.333
* Average WT = 49.167

As we can see from the above data the average turn around time and the wait time are both larger in case of round robin algorithm. Therefore in this case the round robin algorithm is a more suitable algorithm.

**References:**

[**https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/?ref=lbp**](https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/?ref=lbp)

| **On time Submission(2)** | **Knowledge of Topic(4)** | **Implementation and Demonstraion(4)** | **Total (10)** |
| --- | --- | --- | --- |
|  |  |  |  |
| **Signature of Faculty** |  | **Date of Submission** |  |