

Student Name: Rushikesh

Student ID: 11917234

Roll No.: RK18HAB60

Q. No. Assigned: 08, 19

Email Address: rushikesh.zirpe1234@gmail.com

GitHub Link: https://github.com/rushizirpe/OS_Project

1. Explain the problem in terms of operating system concept? (Max 200 word)

Problem 1: Round Robin Scheduling algorithm.

In the given problem 6 processes p1,p2,p3,p4,p5,p6 with Preemptive, RR Scheduling and pre-assigned priority, given Time Quantum set to 10 units. Idle task is scheduled if there is no other available process to run i.e P_idle.

TQ-10 units.

a.Scheduling order of the processes using Gantt chart

(Number with thread name is start time.)

0 -P1
10 -P1
20 -P idle
25 -P2
35 -P3
45 -P2
55 -P3
60 -P4
70 -P4
75 -P2
80 -P3
90 -P idle
100 -P5
105 -P6
115 -P5
120 - end

b.Turnaround time for each Processes

P1: $20-0 = 20$

P2: $80-25 = 55$

P3: $90 - 30 = 60$

P4: $75-60 = 15$

P5: $120-100 = 20$

P6: $115-105 = 10$

c. Waiting time for each processes

P1: 0

P2: 40

P3: 35

P4: 0

P5: 10

P6: 0

d. CPU utilization rate

$105/120 = 87.5 \%$

Problem 2: In the given problem 10 students i.e s1,s2,s3,s4,s5,s6,s7,s8,s9,s10 are purchasing the gift and arriving to billing counter randomly. The accountant gives the preference to that student who has maximum number of gifts.

If we refer this problem with respect to operating System concept, In priority scheduling algorithm, a priority is assigned to each process and CPU is allocated to the process which is having the highest priority. If two processes having same priority arrive then the CPU will be allocated according to first come first serve basis.

Write the algorithm for proposed solution of the assigned problem.

Problem 2:

I have used **Bubble Sort** algorithm in Problem 2 to sort the processes in descending order based on the arrival time of processes and updated their priority and sorted the order of process based on their current priority

Calculate complexity of implemented algorithm. (Student must specify complexity of each line of code along with overall complexity)

Step 1: Compare the first and the second element of the list and swap them if they are in wrong order. $O(n)$

Step 2: Compare the second and the third element of the list and swap them if they are in wrong order. $O(n)$

Step 3: Proceed till the last element of the list in a similar fashion.

Step 4: Repeat all of the above steps until the list is sorted.

The complexity of bubble sort is $O(n^2)$ in both worst and average cases, because the entire array needs to be iterated for every element.

Explain all the constraints given in the problem.

Problem 2: In this problem, we are given a constraint of 10 students. Whereas I applied a constraint of gifts a student can purchase, i.e. $1 \leq \text{gift} < 50$

This implies that one has to buy at least 1 gift or no student can buy more than 49 gifts.

If you have implemented any additional algorithm to support the solution, explain the need and usage of the same.
(Bubble Sort)

Bubble sort is (provably) the fastest sort available under a very specific circumstance. It originally became well known primarily because it was one of the first algorithms (of any kind) that was rigorously analyzed, and the proof was found that it was optimal under its limited circumstance.

```
begin BubbleSort(list)
```

```
for all elements of list
```

```
if list[i] > list[i+1]
```

```
4.           swap(list[i], list[i+1])
```

```
end if
```

```
end for
```

```
return list
```

```
8.end BubbleSort
```

Explain the boundary conditions of the implemented code:

There are no boundary conditions for given problems.

Have you made minimum 5 revisions of solution on GitHub?

GitHub Link: https://github.com/rushizirpe/OS_Project