Report on Round-Robin Scheduling Project

AIM :- About Round-Robin Schedule

Description :-

* The name of this algorithm comes from the round robin principle,where each person gets an equal share of something in turns
* Round Robin is one of the oldest,fairest and easiest algorithms and widely used scheduling methods in traditional OS.
* Round Robin is a pre-emptive algorithm.
* The biggest advantage of the round-robin scheduling method is that of you know the total number of processes on the run queue,then you can assume the worst-case response time for the same process.
* This method spends more time on context switching.
* Worst-cae latency is a term used for the maximum time taken for the execution of all the tasks.

Program Inputs :-

* Number of processes
* Arrival time of each process
* Burst time of each process
* Time quantum (a fixed time slice allotted to each process)

Program Outputs :-

* Waiting time of each process
* Turn-around time of each process
* Average waiting time
* Average turn-around time

Program Implementation :-

The program starts by taking the number of processes and the burst time of each process as input. The burst time is stored in an array for each process.The program then calculates the waiting time and turn-around time for each process using the Round Robin scheduling algorithm.The algorithm operates as follows :

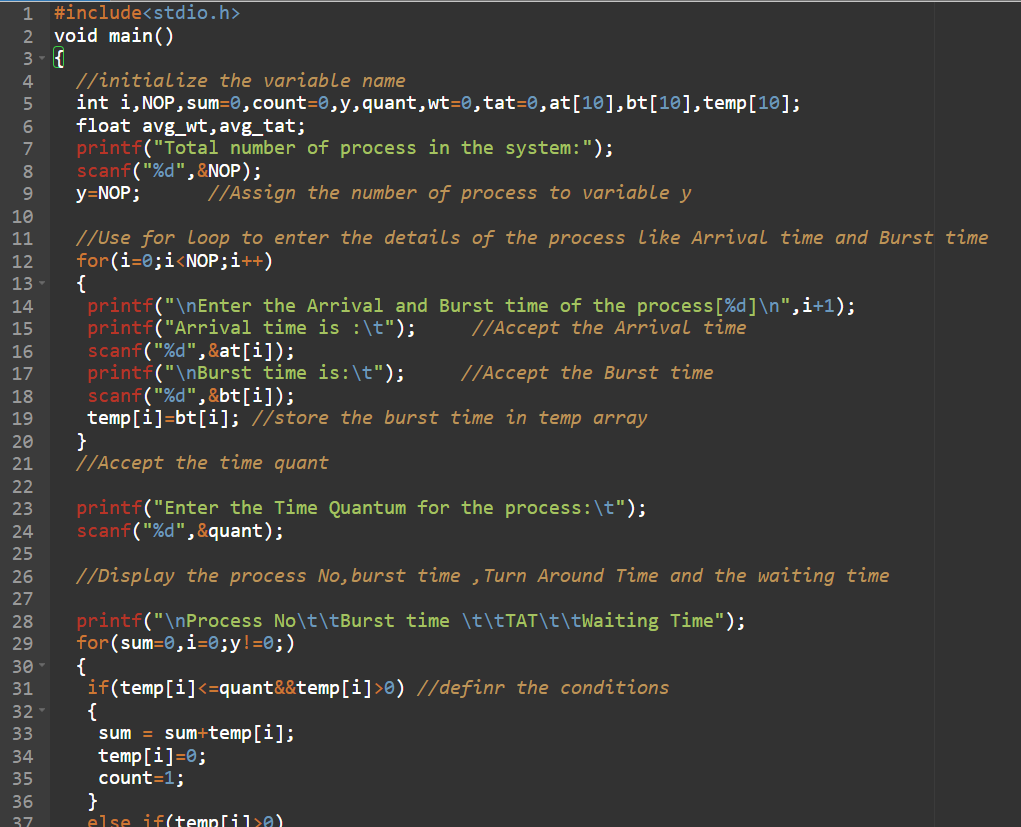
1. The CPU is assigned to the first process in the queue.
2. The CPU is allocated to the process for a fixed time quantum(time slice)
3. If the process completes its execution within the time quantum,it is removed from the queue.
4. If the process does not complete its execution within the time quantum,the remaining time is adde to the end of the queue and the CPU is assigned to the next process in the queue.
5. Steps 2 to 4 are repeated until all process are completed.

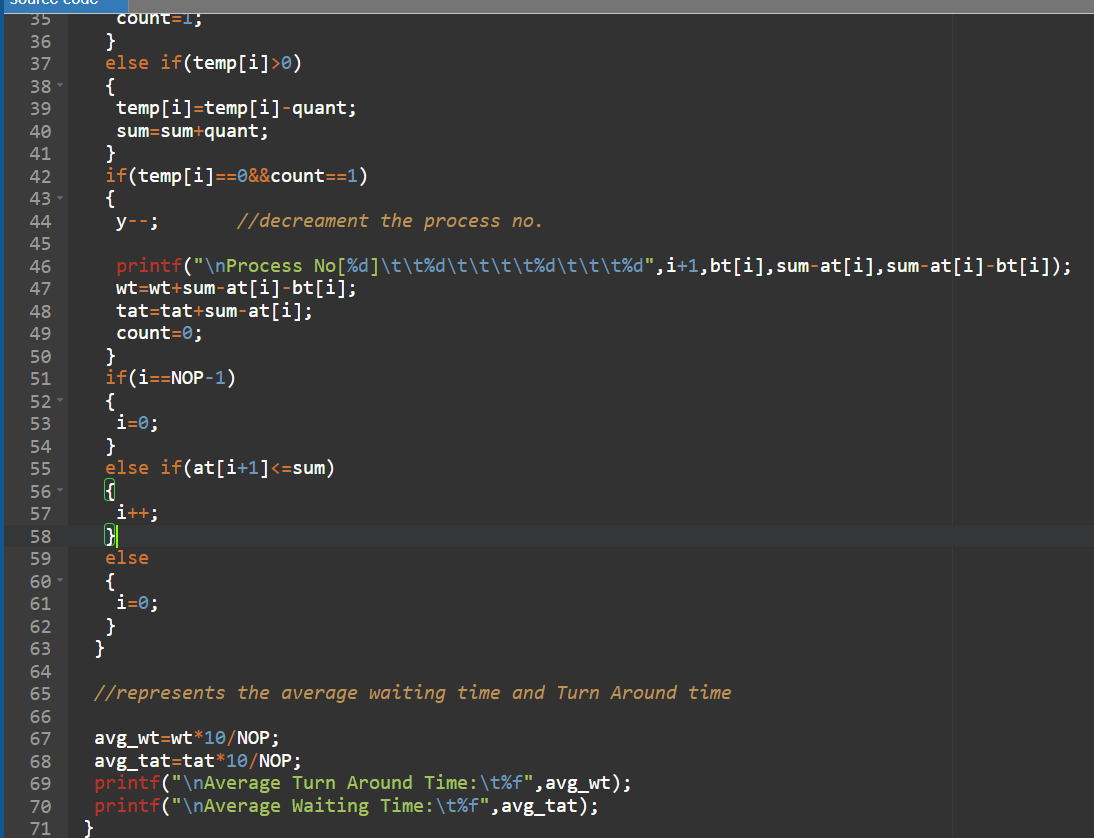
The waiting time for each process is calculated as the difference between the turn-around time and the burst time of the process.The average waiting time and average turn around time are calculated as the sum of waiting time and turn-around time divided by the number of processes.

Program Results :-

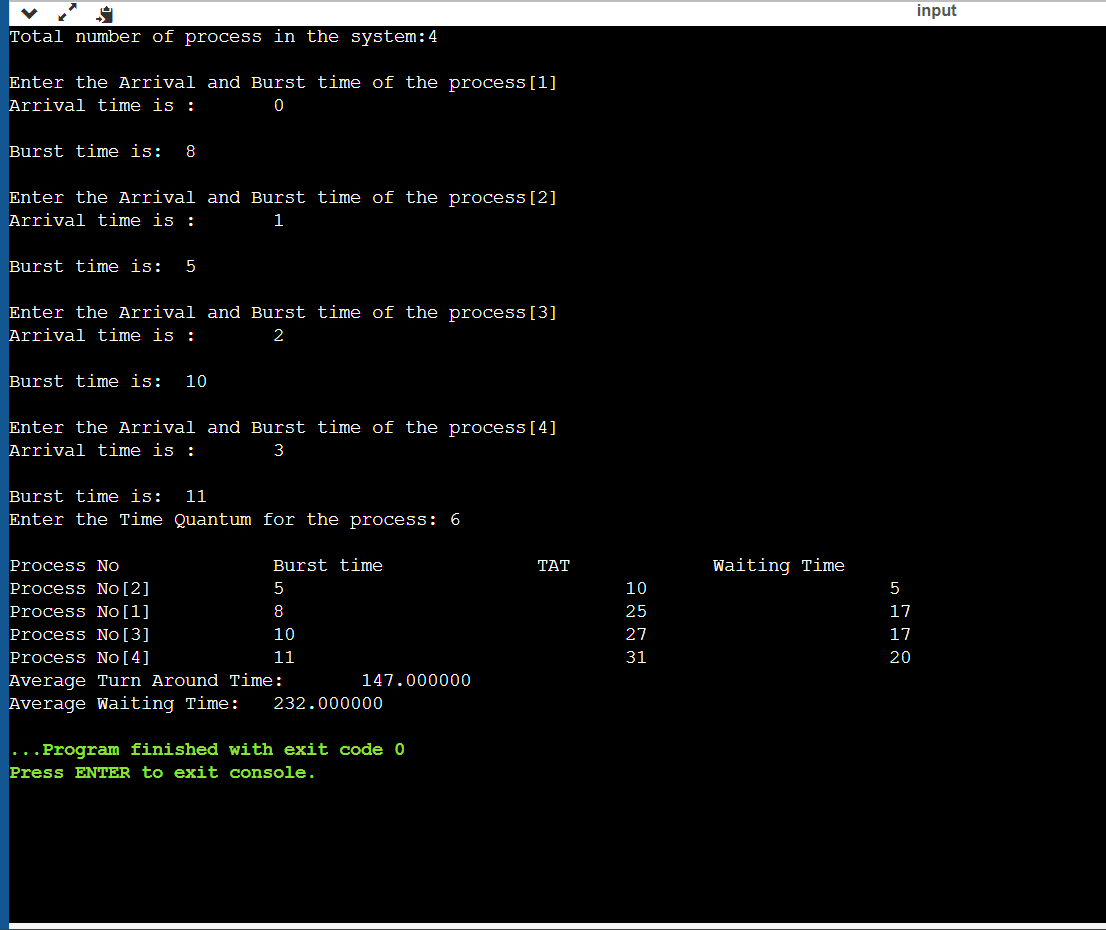
The program outputs the waiting time and turn-around time for each process and the average waiting time and average turn-around time.The results show the distribution of CPU time among processes and demonstrate the effectiveness of the Round Robin scheduling algorithm in providing a fair allocation of CPU resources.

Source Code :-





Program Output :-



Challenges and their possible solutions in Round-Robin scheduling

1. Complexity : Round Robin scheduling algorithms can be complex to implement, especially when taking into account the various factors that can impact the performance of the algorithm.

Solution :- Careful planning and design can help mitigate the complexity of the implementation, as well as the use of established libraries or existing code.

1. Concurrency : Managing concurrent processes can be challenging in round robin scheduling , as it requires proper synchronization between processes to ensure that shared resources are used correctly.

Solution: The use of semaphores, locks, or other synchronization mechanisms can help ensure proper resource utilization.

1. Time-Slice calculation : Determining the correct time slice for each process can be challenging, as it requires taking into account various factors such as the process’s priority , the system’s current load , and the overall performance requirements.

Solution: Dynamic time slice calculation, taking into account real-time data about the system and processes, can help optimize performance.

1. Context switching : Implementing context switching efficiently can be challenging, as it requires managing the process’s state , register values, and memory mappings.

Solution : Using a well-designed data structure to store process information and minimizing the amount of data stored can help improve context switching performance.

1. Queue Management : Managing the schedule queue can be challenging, as it requires proper insertion, deletion, and priority assignment for each process.

Solution : Using a priority queue data structure, along with efficient insertion and deletion algorithms, can help simplify queue management.

1. Load Balancing : Ensuring that processes are disturbed evenly across all available processors can be challenging , especially in multi-processor systems.

Solution : Using load balancing algorithms, such as work as work stealing or task migration, can help distribute processes evenly cross processors

Conclusion :-

The Round Robin scheduling algorithm is a widely used technique for allocating CPU resources to processes in a fair and efficient manner. The implementation of this algorithm in C language demonstrates the application of the algorithm in real-world scenarios and its ability to provide a balanced distribution of CPU time among processes