|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sinhgad Technical Education Society’s** | | | | | | | |
| **SINHGAD ACADEMY OF ENGINEERING, KONDHWA (BK) , PUNE -48** | | | | | | | |
|  | | | | | | | |
| **Department of Information Technology** | | | | | | | |
|  |  |  |  |  |  |  |  |

**Assignment 3**

* **SJF using Preemptive :**

In preemptive scheduling, the CPU is allocated to the processes for a limited time whereas, in Non-preemptive scheduling, the CPU is allocated to the process till it terminates or switches to the waiting state

* **Steps:**

1. Ask the user to enter the number of processes and store it in n.
2. We then accept the burst times from the user.  It is stored in the **bt array**.
3. the burst times are sorted in the next section so the shortest one can be executed first
4. Here selection sort is used to sort the array of burst time**bt**.
5. Waiting time of the first element is zero, the remaining waiting time is calculated by using two for loop that runs from 1 to in that controls the outer loop and the inner loop is controlled by another for loop that runs from j=0 to j<i.
6. Inside the loop, the waiting time is calculated by adding the burst time to the waiting time.
7. Total is the addition of all the waiting time together. The average waiting time is calculated: **avg\_wt=(float)total/n;** and it is printed.
8. Next, the turnaround time is calculated by adding the burst time and the waiting time
9. Again, here the for loop is used. And the total variable here holds the total turnaround time. After this the average turnaround time is calculated.

* **Program :**

#include <stdio.h>

  int main()

{

      int arrival\_time[10], burst\_time[10], temp[10];

      int i, smallest, count = 0, time, limit;

      double wait\_time = 0, turnaround\_time = 0, end;

      float average\_waiting\_time, average\_turnaround\_time;

      printf("Enter the Total Number of Processes:");

      scanf("%d", &limit);

      printf("nEnter Details of %d Processesn", limit);

      for(i = 0; i < limit; i++)

      {

            printf("nEnter Arrival Time:t");

            scanf("%d", &arrival\_time[i]);

            printf("Enter Burst Time:t");

            scanf("%d", &burst\_time[i]);

            temp[i] = burst\_time[i];

      }

      burst\_time[9] = 9999;

      for(time = 0; count != limit; time++)

      {

            smallest = 9;

            for(i = 0; i < limit; i++)

            {

                  if(arrival\_time[i] <= time && burst\_time[i] < burst\_time[smallest] && burst\_time[i] > 0)

                  {

                        smallest = i;

                  }

            }

            burst\_time[smallest]--;

            if(burst\_time[smallest] == 0)

            {

                  count++;

                  end = time + 1;

                  wait\_time = wait\_time + end - arrival\_time[smallest] - temp[smallest];

                  turnaround\_time = turnaround\_time + end - arrival\_time[smallest];

            }

      }

      average\_waiting\_time = wait\_time / limit;

      average\_turnaround\_time = turnaround\_time / limit;

      printf("nnAverage Waiting Time:t%lfn", average\_waiting\_time);

      printf("Average Turnaround Time:t%lfn", average\_turnaround\_time);

      return 0;

}