**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA)

****

**LEARNING ASSESSMENT 1 REPORT**

High Performance Computing

(18CS71)

*Submitted in partial fulfilment of the requirement for the award of Degree of*

*Bachelor of Engineering*

*in*

*Computer Science and Engineering*

*Submitted by:*

|  |  |
| --- | --- |
| Nagarigeri Saicharan Reddy | 1NT18CS104 |



**Department of Computer Science and Engineering**

2021-22

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| Topic | Page no. |
| Problem Statement | 3 |
| Solution Code | 4 |
| Output | 5 |

**PROBLEM STATEMENT**

MTR runs a set of eight restaurants in Bangalore geographically distributed (labelled M0 ... M7 ). Restaurants M0 ... M3 are in the Northern region and M4 ... M7 in the southern region. In order to profitably run the chain the main management located in M0 collects information on an hourly basis from all the locations. The following data set collected on an hourly basis: ( Number of Customers, Number of Rice Idlis sold, Number of Vadas sold, Number of Dosas sold, Number of Rava Idlis sold). Based on the data received from each restaurant they determine a running total of the number of customers and all the items sold in all the restaurants every hour. The Management uses this information to decide on the dishes that need to be prepared for the next one hour in each of the restaurants. This information specific to each restaurant is then sent across from the central site to the destinations.

Determine the costs of transferring information.

Note: Concurrent communications can take place in a time slot as long as there is no conflict on the physical layer. Assume each “number” is of type integer and is one “word” long. The message setup time is ts and the word transfer time is tw.

ii. What will be cost if the management is interested in only knowing about the total number of customers and the total of each of the types of dishes that are sold and is not bothered about associating it with specific locations.

**Solution Code**

|  |
| --- |
| #include <stdio.h> |
|  |

|  |
| --- |
| #include <time.h> |
|  |

|  |
| --- |
| #include <math.h> |
|  |

|  |
| --- |
| #include <omp.h> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| double s1, s2, e1, e2, clk1, clk2; |
|  |

|  |
| --- |
| int n; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // struct for the restaurant |
|  |

|  |
| --- |
| struct R\_data{ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int no\_of\_customer; |
|  |

|  |
| --- |
| int no\_of\_idlis; |
|  |

|  |
| --- |
| int no\_of\_vada; |
|  |

|  |
| --- |
| int no\_of\_dosas; |
|  |

|  |
| --- |
| int no\_of\_rava\_idli; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // for total calculation |
|  |

|  |
| --- |
| struct Total\_Data{ |
|  |

|  |
| --- |
| int no\_of\_customer; |
|  |

|  |
| --- |
| int no\_of\_idlis; |
|  |

|  |
| --- |
| int no\_of\_vada; |
|  |

|  |
| --- |
| int no\_of\_dosas; |
|  |

|  |
| --- |
| int no\_of\_rava\_idli; |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
| // sending specific restaurant data |
|  |

|  |
| --- |
| struct Receive\_data{ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int no\_of\_customer; |
|  |

|  |
| --- |
| int no\_of\_idlis; |
|  |

|  |
| --- |
| int no\_of\_vada; |
|  |

|  |
| --- |
| int no\_of\_dosas; |
|  |

|  |
| --- |
| int no\_of\_rava\_idli; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int main() |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| struct R\_data r\_data[8]; |
|  |

|  |
| --- |
| struct Total\_Data t\_data; |
|  |

|  |
| --- |
| struct Receive\_data R\_data[8]; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| for(int i = 0; i < 8; i++){ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| r\_data[i].no\_of\_customer = rand() % 10; |
|  |

|  |
| --- |
| r\_data[i].no\_of\_idlis = rand() % 10; |
|  |

|  |
| --- |
| r\_data[i].no\_of\_vada = rand() % 10; |
|  |

|  |
| --- |
| r\_data[i].no\_of\_dosas = rand() % 10; |
|  |

|  |
| --- |
| r\_data[i].no\_of\_rava\_idli = rand() % 10; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| printf("Enter the restaurant number\n\n"); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| scanf("%d", &n); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #pragma omp parallel for |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| printf("first part of the question\n"); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| s1 = clock(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| R\_data[6].no\_of\_customer = r\_data[6].no\_of\_customer; |
|  |

|  |
| --- |
| R\_data[6].no\_of\_idlis = r\_data[6].no\_of\_idlis; |
|  |

|  |
| --- |
| R\_data[6].no\_of\_vada = r\_data[6].no\_of\_vada; |
|  |

|  |
| --- |
| R\_data[6].no\_of\_dosas = r\_data[6].no\_of\_dosas; |
|  |

|  |
| --- |
| R\_data[6].no\_of\_rava\_idli = r\_data[6].no\_of\_rava\_idli; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| e1 = clock(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| printf(" restaurant number : M(%d)\n", n-1); |
|  |

|  |
| --- |
| printf(" customer %d \n",R\_data[6].no\_of\_customer); |
|  |

|  |
| --- |
| printf(" idlis %d \n", R\_data[6].no\_of\_idlis ); |
|  |

|  |
| --- |
| printf(" vada %d\n", R\_data[6].no\_of\_vada); |
|  |

|  |
| --- |
| printf(" dosa %d\n", R\_data[6].no\_of\_dosas); |
|  |

|  |
| --- |
| printf("rava\_idli %d\n\n", R\_data[6].no\_of\_rava\_idli); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| clk1 = double(e1 - s1) / CLOCKS\_PER\_SEC; |
|  |

|  |
| --- |
| printf("time : %lf\n\n", clk1); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| s2 = clock(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| for(int i = 0; i<8; i++) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| t\_data.no\_of\_customer += r\_data[i].no\_of\_customer; |
|  |

|  |
| --- |
| t\_data.no\_of\_idlis += r\_data[i].no\_of\_customer; |
|  |

|  |
| --- |
| t\_data.no\_of\_vada += r\_data[i].no\_of\_vada; |
|  |

|  |
| --- |
| t\_data.no\_of\_dosas += r\_data[i].no\_of\_dosas; |
|  |

|  |
| --- |
| t\_data.no\_of\_rava\_idli += r\_data[i].no\_of\_rava\_idli; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| printf("second part of question\n\n"); |
|  |

|  |
| --- |
| printf("total customer %d \n",t\_data.no\_of\_customer); |
|  |

|  |
| --- |
| printf("total idlis %d \n",t\_data.no\_of\_idlis ); |
|  |

|  |
| --- |
| printf("total vada %d\n",t\_data.no\_of\_vada ); |
|  |

|  |
| --- |
| printf("total dosa %d\n",t\_data.no\_of\_dosas); |
|  |

|  |
| --- |
| printf("total rava\_idli %d\n",t\_data.no\_of\_rava\_idli ); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| e2 = clock(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| clk2 = double(s2 - e2) / CLOCKS\_PER\_SEC; |
|  |

|  |
| --- |
| printf("time : %lf\n\n", clk2); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

}

**Output:**

Text

Description automatically generated