|  |  |
| --- | --- |
| **Ex. No. 08** | **MULTITHREADING AND SYNCHRONIZATION** |
| **10.10.2023** |

**Aim**

To develop C# console application using Multithreading and Synchronization concepts.

**Description**

**Multithreading:**

* Allows a process to manage two or more concurrent threads
* Each can handle a task independently
* Implemented under System.Threading and Thread class.
* Executing a function passing it to thread class object created
* Explicitly call Start() function in order to run the thread
* Join() makes the other threads to wait for the running thread.

**Synchronization:**

Locks the shared object which makes to access by only one thread at a time until it gets completed.

Syntax:

Thread th= new Thread(<function\_name>);

th.Start();

lock(object){ //lock block of statements}

**Source Code**

**A 1.**

using System;

using System.Threading;

namespace Ex8{

internal class Question1{

public static string para;

public static void word\_printing(){

string[] li=para.Split(' ');

for(int i = 0; i < li.Length; i++){

Console.Write(li[i]+", ");}

Thread.Sleep(2000);}

public static void vowels\_printing(){

for(int i = 0; i < para.Length; i++){

if ("aeiouAEIOU".Contains(para[i])){

Console.Write(para[i]+", ");}}}

static void Main(string[] args){

Console.WriteLine("Enter any Paragraph");

para=Console.ReadLine();

Thread t1=new Thread(word\_printing);

Thread t2=new Thread(vowels\_printing);

t1.Start();

t1.Join();

t2.Start();

t2.Join();

Console.ReadKey();}}}

**A 2.**

using System;

using System.Threading;

namespace Ex8{

internal class Question2{

public static int[] arr;

public static void even\_printing(){

Console.WriteLine("Displaying Even Numbers:");{

if (arr[i]%2 == 0){

Console.Write(arr[i]+", ");}}

Thread.Sleep(2000);}

public static void odd\_printing(){

Console.WriteLine("\nDisplaying Odd Numbers:");

for (int i = 0; i < arr.Length; i++){

if (arr[i] % 2 != 0){

Console.Write(arr[i] + ", ");}}}

static void Main(string[] args){

arr=new int[10];

Console.WriteLine("Enter Array Elements: ");

for(int i = 0; i < arr.Length; i++){

arr[i]=Convert.ToInt32(Console.ReadLine());}

Thread t1=new Thread(even\_printing);

Thread t2=new Thread(odd\_printing);

t1.Start();

t1.Join();

t2.Start();

t2.Join();

Console.ReadKey();}}}

**B.**

using System;

using System.Threading;

namespace Ex8{

internal class Question3{

public static int num1, num2,num3;

public void table\_printing(){

lock (this){

Console.WriteLine("\nPrinting Table:");

for (int i = 1; i <= 10; i++){

if (Thread.CurrentThread.Name == "num1\_t") { Console.WriteLine(num1 + "x" + i + "=" + num1 \* i); }

if (Thread.CurrentThread.Name == "num2\_t") { Console.WriteLine(num2 + "x" + i + "=" + num2 \* i); }

if (Thread.CurrentThread.Name == "num3\_t") { Console.WriteLine(num3 + "x" + i + "=" + num3 \* i); }}

Thread.Sleep(2000);}}

static void Main(string[] args){

Console.Write("Enter Num1: ");

num1 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter Num2: ");

num2 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter Num3: ");

num3 = Convert.ToInt32(Console.ReadLine());

Question3 c=new Question3();

Thread num1\_t=new Thread(c.table\_printing);

Thread num2\_t = new Thread(c.table\_printing);

Thread num3\_t = new Thread(c.table\_printing);

num1\_t.Name = "num1\_t";

num2\_t.Name = "num2\_t";

num3\_t.Name = "num3\_t";

num1\_t.Start();

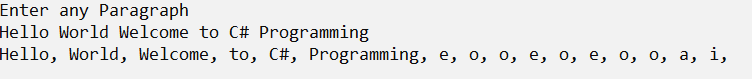
num2\_t.Start();

num3\_t.Start();

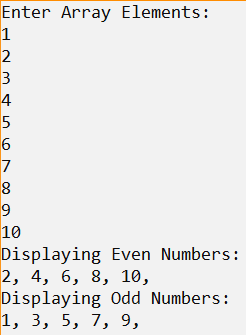
Console.ReadKey();}}}

**Output**

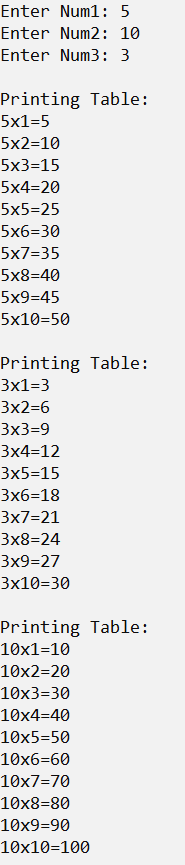
**A 1.**

****

**A 2.**

****

**B.**

****

**Result**

The C# console application using Multithreading and Synchronization concepts has been executed successfully and the desired output is displayed on the screen.