**Universidad Autónoma del Estado de México**

**Facultad de Ingeniería**

**División de Computación**

***Manual técnico***



***Integrantes:***

***Alvarez Porcayo Erick***

***Romero Escalona Luis Angel***

***David Josué Hernández Herrera***

***Indice***

***1.- Descripción del problema …………………………………………………………………..3***

***2.- Restricciones básicas ……………………………………………………………………….3***

***3.-Requerimientos de Hardware y Software ………………………………………………..3***

***4.- Recomendaciones …………………………………………………………………………..3***

***5.- Código fuente ……………………………………………………………………………….4***

***6- Referencias Bibliográficas y web usadas en fundamentos teóricos …………….***

**1.- Descripción del problema**

Por medio de la programación paralela, podemos optimizar el uso de procesadores con los que cuenta la computadora convencional, para llevar a acabo estas pruebas se tomaran en cuenta los algoritmos de Suma EREW, CREW, Búsqueda EREW, CECW, ordenamiento CRCW.

**2.-Restricciones básicas**

Los programas fueron hechos en el lenguaje de programación C# y se tomo en cuenta la programación de procesos en paralelo, haciendo uso de tareas para lograr el objetivo.

3**.-Requerimientos de Hardware y Software**

**Hardware**

* Procesador Intel Pentium 233 MHz o superior, o procesador AMD VISION E-204 1.50GHz inferior o superior. En OSX procesador 2.26 GHz Intel Core 2 Duo
* 64 MB RAM (se recomiendan 128MB)
* 400 MB de espacio disponible en disco duro
* Unidad grabadora de CD-ROM (para instalar el software base: compilador, S.O., y/o grabar los CD’s de usuario y técnico)
* Monitor con resolución SVGA o superior
* Mouse u otro dispositivo señalador

**Software**

* Microsoft Windows Vista**,** Windows XP, Windows 2000 o Windows 98
* Macintosh OSX snow leopard
* MATLAB version 7.10.0.499 (R2010a) o posterior

**4.-Recomendaciones**

El programa, en futuras versiones, puede optimarse con base en las siguientes sugerencias :

* Mejorar la interfaz de usuario para incrementar la usabalidad.
* Pedir al usuario que ingrese cualquier valor al vector que se va a trabajar.

**5.- Código Fuente**

**namespace SUMAEREW**

**{**

**class sumaErew**

**{**

**int[] vec = { 5, 2, 10, 1, 8, 12, 7, 3, 2};**

**int n = 8;**

**int i, j, x;**

**public void Suma(){**

**x = (int)(Math.Log(n, 2));**

**//x = (int)(Math.Log(n) / Math.Log(2));**

**imprimir();**

**for (i = 1; i < x+1; i++){**

**for (j = 1; j <= (n/2); j++){**

**Thread Hilo = new Thread(EREW);**

**Hilo.Start();**

**Hilo.Join();**

**//imprimir();**

**}**

**}**

**imprimir();**

**Console.Write("\n");**

**Console.WriteLine("La suma es:" + vec[n]);**

**Console.ReadKey(true);**

**}**

**public void EREW(){**

**if(((2\*j) % ((int)(Math.Pow(2, i)))) == 0){**

**vec[j\*2] = vec[j\*2] + vec[(j\*2) - (int)(Math.Pow(2, i-1))];**

**}**

**}**

**public void imprimir(){**

**Console.Write("[");**

**foreach (int z in vec){**

**Console.Write("{0} ", z);**

**}**

**Console.Write("]\n");**

**}**

**static void Main(string[] args){**

**Console.WriteLine("SUMA EREW");**

**sumaErew s = new sumaErew();**

**s.Suma();**

**}**

**}**

**}**

**namespace sumaCREW{**

**class sumaCrew{**

**int[] vec = {0,5,2,10,1,8,12,7,3};**

**int n = 8;**

**int i, j, x;**

**public void Suma(){**

**x = (int)(Math.Log(n, 2)); // Math.Log(2));**

**Console.Write("\n" + x);**

**imprimir();**

**for (i = 1; i <= x; i++){**

**for (j = 1; j <= n; j++){**

**if ((Math.Pow(2, i-1)) == 1){**

**Thread Hilo = new Thread(CREW);**

**Hilo.Start();**

**Hilo.Join();**

**//imprimir();**

**}**

**}**

**}**

**imprimir();**

**Console.Write("\n");**

**Console.WriteLine("La suma es:" + vec[n]);**

**Console.ReadKey(true);**

**}**

**public void CREW(){**

**vec[j] = vec[j] + vec[j - (int)(Math.Pow(2, i - 1))];**

**}**

**public void imprimir(){**

**Console.Write("[");**

**foreach (int z in vec){**

**Console.Write("{0} ", z);**

**}**

**Console.Write("]\n");**

**}**

**static void Main(string[] args){**

**Console.WriteLine("SUMA CREW");**

**sumaCrew s = new sumaCrew();**

**s.Suma();**

**}**

**}**

**}**

**namespace BUSQUEDAEREW**

**{**

**class BusquedaErew{**

**int[] L = { 0, 2, -1, 23, -4, 2, 55, -2, 0, 5, 1, 5, -5, 8, 5, 3, -2 };**

**int[] A = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };**

**int i, j, lg;**

**int n = 17;**

**int x = 55;**

**public void searchEREW(){**

**lg = (int)(Math.Log(n, 2));**

**broadCast();**

**for (i = 1; i < n; i++)**

**{**

**Thread Hilo = new Thread(hilo3);**

**Hilo.Start();**

**Hilo.Join();**

**}**

**min();**

**Console.WriteLine(" El numero 55 esta en la posicion:" + A[1]);**

**Console.ReadKey(true);**

**}**

**public void broadCast(){**

**A[1] = x;**

**int aux, aux2;**

**for(i=1; i<lg+1; i++){**

**aux = (int)(Math.Pow(2, i-1))+1;**

**aux2 = (int)(Math.Pow(2, i))+1;**

**for (j = aux; j < aux2; j++){**

**Thread Hilo = new Thread(hilo1);**

**Hilo.Start();**

**Hilo.Join();**

**}**

**}**

**}**

**public void min(){**

**i = 1;**

**int aux =((int)(n/(Math.Pow(2, i)))+1);**

**for(i=1; i<n; i++){**

**for (j = 1; j<aux ; j++){**

**Thread Hilo = new Thread(hilo2);**

**Hilo.Start();**

**Hilo.Join();**

**}**

**}**

**}**

**public void hilo1(){**

**A[j] = A[j - (int)(Math.Pow(2, (i-1)))];**

**}**

**public void hilo2(){**

**if(A[j\*2]>A[(j\*2)-1]){**

**A[j]=A[(j\*2)-1];**

**}**

**else{**

**A[j]=A[j\*2];**

**}**

**}**

**public void hilo3(){**

**if (L[i] == A[i]){**

**A[i] = i;**

**}**

**else{**

**A[i] = 100;**

**}**

**}**

**public void imprimir()**

**{**

**foreach (int z in L)**

**{**

**Console.Write("{0} ", z);**

**}**

**}**

**static void Main(string[] args){**

**Console.WriteLine("\n");**

**Console.WriteLine(" BUSQUEDA EREW");**

**BusquedaErew s = new BusquedaErew();**

**s.imprimir();**

**s.searchEREW();**

**}**

**}**

**}**

**namespace OrdenamientoCRCW{**

**class BusquedaCECW{**

**int[] L = { 95, 10, 6, 15 };**

**int[] win = { 1, 1, 0, 1 };**

**int n = 3;**

**int i, j, result;**

**public void Ordenar(){**

**imprimir();**

**for (i = 1; i <= n; i++){**

**Thread Hilo = new Thread(Hilo1);**

**Hilo.Start();**

**Hilo.Join();**

**}**

**for (i = 0; i <= n; i++){**

**for (j = 2; j <= n; j++){**

**if (i < j){**

**Thread Hilo = new Thread(Hilo2);**

**Hilo.Start();**

**Hilo.Join();**

**}**

**imprimir2();**

**}**

**}**

**for (i = 1; i <= n; i++){**

**for (j = 1; j <= n; j++){**

**Thread Hilo = new Thread(Hilo3);**

**Hilo.Start();**

**Hilo.Join();**

**}**

**}**

**Console.Write("\n");**

**Console.WriteLine("El menor valor es:" + L[result]);**

**Console.ReadKey(true);**

**}**

**public void Hilo1(){**

**win[i] = 0;**

**}**

**public void Hilo2(){**

**if (L[i] > L[j]){**

**win[i] = 1;**

**}**

**else{**

**win[j] = 1;**

**}**

**}**

**public void Hilo3(){**

**if (win[i] == 0)**

**{**

**result = i;**

**}**

**}**

**public void imprimir(){**

**Console.Write("L=[");**

**foreach (int z in L){**

**Console.Write("{0} ", z);**

**}**

**}**

**public void imprimir2()**

**{**

**Console.Write("win=[");**

**foreach (int z in win)**

**{**

**Console.Write("{0} ", z);**

**}**

**Console.Write("]\n");**

**}**

**static void Main(string[] args){**

**Console.WriteLine("ALGORITMO 4 (minCRCW)");**

**BusquedaCECW s = new BusquedaCECW();**

**s.Ordenar();**

**}**

**}**

**}**

**6.-Referencias bibliográficas y web usadas para fundamentos teóricos**

* C# Cómo programar, segunda edición, Prentice Hill
* C- Multithreaded and Parallel Programming, Rodney Ringler, Packt publishing
* https://msdn.microsoft.com/es-mx/library/dd460713(v=vs.110).aspx