

Programación Concurrente, Febrero 2023

Práctica 1

Martínez Buenrostro Jorge Rafael.
Universidad Autónoma Metropolitana
Unidad Iztapalapa, México
molap96@gmail.com

Soluciones

Problema 1

Para este problema usé como base el ejemplo *HilosNoSincronizados* el cual crea una cantidad N de hilos en los que su ejecución no está sincronizada de ninguna manera. Por lo que cada uno se ejecuta sin esperar al siguiente. Dentro de la función que ejecutan los hilos agregué al inicio la creación de las variables que acotan la búsqueda de cada hilo de acuerdo a su número de hilo. Una vez calculadas estas cotas el hilo recorre el arreglo de número desde su cota menor hasta su cota mayor, verificando si cada uno de los elementos es un número primo; en caso que el elemento sea un número primo escribe el dato en una localidad exclusiva de la memoria.

Al final el hilo padre suma los resultados puestos por los hilos hijos y los muestra en pantalla.

Problema 2

La base de este problema es el patrón Líder-Seguidor, el cuál le da un turno a cada hilo creado para que se puedan ejecutar en orden. Bajo esta premisa el hilo padre crea N cantidad de hilos hijo para que cada uno cuente la cantidad de número primos dentro de un subarreglo. Como cada hilo hijo tiene su turno aumenta un contador que es compartido para todos los hilos hijo.

Al final el hilo padre muestra en pantalla el valor del contador compartido

Programas fuente

Problema 1

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>

void Escribe( void *ptr);
int menorDivisor(int);
int esPrimo(int);
int * arreglo;
int * resultados;
int numHilos,m;

int main()
{
    printf("Indique el tamaño M del arreglo...");
    scanf("%d",&m);
    numHilos=menorDivisor(m);

    int semilla;
    int i, ids[numHilos],sumaTotal=0;
    pthread_t hilos[numHilos];

    /* Inicializamos el generador de numeros aleatorios */
    semilla = time(NULL);
    srand(semilla);

    //Reservamos memoria para el arreglo
    arreglo=(int*)calloc(m,sizeof(int));
    resultados=(int*)calloc(numHilos,sizeof(int));
    printf("\nLos valores de M y N son: %d y %d respectivamente\n\n",m,numHilos);
    //Se llena el arreglo con numero aleatorios entre 1 y 100
```

```

for(i=0;i<m;i++){
    arreglo[i]=(rand() % (100 - 1 + 1)) + 1;
}

for(i = 0; i < numHilos; i++){
    ids[i] = i;
    pthread_create(&hilos[i],NULL, (void *) &Escribe, (void *) (intptr_t)ids[i]);
}

for(i = 0;i < numHilos; i++){
    pthread_join(hilos[i], NULL);
    printf("\nNumeros primos del hilo %d: [%d]",i,resultados[i]);
    sumaTotal+=resultados[i];
}

printf("\n\nHilo padre: todos los hilos han terminado\n");
printf("\tSe encontraron %d primos en el arreglo\n\n\t",sumaTotal);
for(i=0;i<m;i++){
    printf("%d ",arreglo[i]);
}
printf("\n");
}

void Escribe(void *ptr){
    int numHilo = (intptr_t) ptr;
    int cotaMenor, cotaMayor,i,j;

    cotaMenor=(m/numHilos*numHilo)+1;
    cotaMayor=(m/numHilos)*(numHilo+1);

    printf("Rango de busqueda en el arreglo del hilo %d :
[%d,%d]\n",numHilo,cotaMenor,cotaMayor);
    for(i=cotaMenor-1;i<cotaMayor;i++){
        if(esPrimo(arreglo[i])==1)

```

```

        resultados[numHilo]++;
    }
    pthread_exit(0);
}

int menorDivisor(int m){
    int i;
    for(i = 2; i <= m; i++) {
        if((m%i) == 0){
            return i;
        }
    }
}

int esPrimo(int a){
    int i, esPrimo=0;
    for (i = 1; i <= a; i++) {
        if (a % i == 0) {
            esPrimo++;
        }
    }

    if (esPrimo == 2) {
        return 1;
    }
    else {
        return 0;
    }
}

```

Problema 2

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void EscribeLiderSeguidor( void *ptr);
int menorDivisor(int);
int esPrimo(int);
int numHilos, m;
int * arreglo;
int contador;
int turno; /* VARIABLE COMPARTIDA */

int main()
{
    printf("Indique el tamaño M del arreglo...");
    scanf("%d",&m);
    numHilos=menorDivisor(m);

    int i, ids[numHilos],semilla;
    pthread_t hilos[numHilos];

    //Inicializamos el generador de numeros aleatorios
    semilla=time(NULL);
    srand(semilla);
    turno = 0;

    //Reservamos memoria para el arreglo
    arreglo=(int*)calloc(m,sizeof(int));

    printf("\nLos valores de M y N son: %d y %d respectivamente\n\n",m,numHilos);
    //Se llena el arreglo con numero aleatorios entre 1 y 100
    for(i=0;i<m;i++){
        arreglo[i]=(rand() % (100 - 1 + 1)) + 1;
    }
```

```

    for(i = 0; i < numHilos; i++){
        ids[i] = i;
        pthread_create(&hilos[i],NULL,(void *) &EscribeLiderSeguidor, (void
*) (intptr_t)ids[i]);
    }

    for(i = 0;i < numHilos; i++){
        pthread_join(hilos[i], NULL);
    }

    printf("\n\nHilo padre: todos han terminado\n");
    printf("\tSe encotraron %d primos en el arreglo\n\n\t",contador);
    for(i=0;i<m;i++){
        printf("%d ",arreglo[i]);
    }
    printf("\n");
}

void EscribeLiderSeguidor(void *ptr){
    int j, numHilo = (intptr_t) ptr;
    int cotaMenor, cotaMayor,i;

    while(numHilo!=turno){ };

    cotaMenor=(m/numHilos*numHilo)+1;
    cotaMayor=(m/numHilos)*(numHilo+1);
    printf("Rango de busqueda en el arreglo del hilo %d :
[%d,%d]\n",numHilo,cotaMenor,cotaMayor);
    for(i=cotaMenor-1;i<cotaMayor;i++){
        if(esPrimo(arreglo[i])==1)
            contador++;
    }

    /* LIBERA EL DERECHO DE ACCESO A LA PANTALLA CEDIENDOLO AL SIGUIENTE HILO*/
    turno = (turno + 1) % numHilos;

```

```
pthread_exit(0);
}

int menorDivisor(int m){
    int i;
    for(i = 2; i <= m; i++) {
        if((m%i) == 0){
            return i;
        }
    }
}

int esPrimo(int a){
    int i,esPrimo=0;
    for (i = 1; i <= a; i++) {
        if (a % i == 0) {
            esPrimo++;
        }
    }

    if (esPrimo == 2) {
        return 1;
    }
    else {
        return 0;
    }
}
```

Salidas

Problema 1

```
Indique el tamaño M del arreglo...10
Los valores de M y N son: 10 y 2 respectivamente
Rango de búsqueda en el arreglo del hilo 1 : [6,10]
Rango de búsqueda en el arreglo del hilo 0 : [1,5]

Numeros primos del hilo 0: [2]
Numeros primos del hilo 1: [1]

Hilo padre: todos los hilos han terminado
          Se encontraron 3 primos en el arreglo

        60 42 2 87 43 41 58 9 91 21
```

```
Indique el tamaño M del arreglo...15
Los valores de M y N son: 15 y 3 respectivamente
Rango de búsqueda en el arreglo del hilo 2 : [11,15]
Rango de búsqueda en el arreglo del hilo 1 : [6,10]
Rango de búsqueda en el arreglo del hilo 0 : [1,5]

Numeros primos del hilo 0: [0]
Numeros primos del hilo 1: [2]
Numeros primos del hilo 2: [1]

Hilo padre: todos los hilos han terminado
          Se encontraron 3 primos en el arreglo

        18 77 78 32 58 31 42 67 15 94 49 10 77 29 38
```

```
Indique el tamaño M del arreglo...16
Los valores de M y N son: 16 y 2 respectivamente
Rango de búsqueda en el arreglo del hilo 0 : [1,8]
Rango de búsqueda en el arreglo del hilo 1 : [9,16]

Numeros primos del hilo 0: [1]
Numeros primos del hilo 1: [2]

Hilo padre: todos los hilos han terminado
          Se encontraron 3 primos en el arreglo

        99 26 68 87 34 48 97 86 8 89 54 28 28 90 64 97
```


Problema 2

```
Indique el tamaño M del arreglo...10
Los valores de M y N son: 10 y 2 respectivamente
Rango de búsqueda en el arreglo del hilo 0 : [1,5]
Rango de búsqueda en el arreglo del hilo 1 : [6,10]

Hilo padre: todos han terminado
           Se encontraron 2 primos en el arreglo

           9 72 83 9 54 90 39 42 47 27
```

```
Indique el tamaño M del arreglo...15
Los valores de M y N son: 15 y 3 respectivamente
Rango de búsqueda en el arreglo del hilo 0 : [1,5]
Rango de búsqueda en el arreglo del hilo 1 : [6,10]
Rango de búsqueda en el arreglo del hilo 2 : [11,15]

Hilo padre: todos han terminado
           Se encontraron 5 primos en el arreglo

           56 84 5 57 77 45 63 67 6 61 86 93 36 71 2
```

```
Indique el tamaño M del arreglo...10
Los valores de M y N son: 10 y 2 respectivamente
Rango de búsqueda en el arreglo del hilo 0 : [1,5]
Rango de búsqueda en el arreglo del hilo 1 : [6,10]

Hilo padre: todos han terminado
           Se encontraron 3 primos en el arreglo

           3 5 64 55 96 34 7 39 69 51
```

Enlace GDB online

- <https://onlinegdb.com/250Q3q24x>

Profesora GDB online no me deja compartir enlaces de carpetas nuevas, le comparto el enlace a la carpeta de la práctica 1, en el cual están los proyectos para la Práctica 2.

Se llaman Actividad 3_B y Actividad 3_C. Le pido una disculpa por los inconvenientes, veré que puedo hacer para la siguiente práctica