



T.C.

ONDOKUZ MAYIS ÜNİVERSİTESİ

SİSTEM PROGRAMLAMA PROJE RAPORU

20060355

ALEYNA KAHRAMAN

Multi-threaded Matrix Multiplication

Ödevin konusu: 5*5'lik iki tane kare matrisin çarpımının thread kullanarak c programlama dilinde kodlanması.

Kullanılan platform: kali linux sanal makinası

Kullanılan dil: c programlama dili

Sistem programlama

Thread: Bir process'in birden fazla işi aynı anda yapmasını sağlayan yapılara **thread** denir. Bir process bünyesinde bir ya da birden fazla thread barındırabilir. Thread'ler aynı anda sadece tek bir iş yapabilir. Kısaca N adet thread N adet iş yapabilir diyebiliriz. Bir process içerisinde birden fazla thread çalıştırılmasına multi-threading diyoruz.

Matris çarpımında çarpım sonucu oluşan matrisin 1.satırındaki değerleri thread_1, 2.satırındaki değerleri thread_2, 3. satırındaki değerleri thread_3 , 4.satırındaki değerleri thread_4, 5.satırındaki değerleri thread_5 yapmalıdır. Her satırı bir thread hesaplar ve thread join yaparak her thread'in yapması gereken işlemleri ana thread beklemelidir.

KOD

```
1  #include<stdio.h>
2  #include<pthread.h>
3  #include<stdlib.h>
4  #define MAX 5
5
```

Kodumuzda threadleri kullanacağımız için ilk önce pthread kütüphanesini kodumuza eklememiz gerekir. Malloc fonksiyonu kullanıldığı için stdlib kütüphanesi eklenir. Ve MAX sabiti 5 olarak tanımlanır.

```

6   void *mult(void* arg)
7   {
8       int *data = (int *)arg;
9       int k = 0, i = 0;
10
11      int x = data[0];
12      for (i = 1; i <= x; i++)
13          k += data[i]*data[i+x];
14
15      int *p = (int*)malloc(sizeof(int));
16      *p = k;
17
18      pthread_exit(p);
19  }
20

```

Her iş parçacığı, elde edilen matristeki tek elemanı hesaplar. Ve bir iş parçacığını sonlandırmak için kullanılır ve dönüş değeri bir işaretçi olarak iletilir.

```

20
21  int main()
22  {
23      int matris_A[MAX][MAX];
24      int matris_B[MAX][MAX];
25
26      int r1=MAX,c1=MAX,r2=MAX,c2=MAX,i,j,k;
27

```

A, B matrisleri r1,c1,r2,c2,i,j,k değişkenleri tanımlanır. 5*5lik ilk matrisimizin adı A ikinci matrisimizin adı B'dir.

```

27
28      for (i = 0; i < r1; i++)
29          for (j = 0; j < c1; j++)
30              matris_A[i][j] = rand() % 10;
31
32      for (i = 0; i < r1; i++)
33          for (j = 0; j < c1; j++)
34              matris_B[i][j] = rand() % 10;
35

```

A ve B matrisleri random şekilde oluşturulur.

```

35
36     printf("MATRİS A \n");
37     for (i = 0; i < r1; i++){
38         for(j = 0; j < c1; j++)
39             printf("%d ",matris_A[i][j]);
40         printf("\n");
41     }
42
43     printf("MATRİS B \n") ;
44     for (i = 0; i < r2; i++){
45         for(j = 0; j < c2; j++)
46             printf("%d ",matris_B[i][j]);
47         printf("\n");
48     }

```

A ve B matrisleri ekrana bastırılır.

```

49
50     int max = r1*c2;
51
52     pthread_t *threads;
53     threads = (pthread_t*)malloc(max*sizeof(pthread_t));
54

```

$r1 \times c2$ boyutunda iş parçacığı dizisi bildiriliyor.

```

55     int count = 0;
56     int* data = NULL;
57     for (i = 0; i < r1; i++)
58         for (j = 0; j < c2; j++) {
59             data = (int *)malloc((20)*sizeof(int));
60             data[0] = c1;
61             for (k = 0; k < c1; k++)
62                 data[k+1] = matris_A[i][k];
63             for (k = 0; k < r2; k++)
64                 data[k+c1+1] = matris_B[k][j];
65             pthread_create(&threads[count++], NULL,
66                 mult, (void*)(data));
67         }

```

Veriler için alan tahsisi yapılır.Verilerin satır ve sütun öğeleri depolanır.

Thread yaratılır ve mult fonksiyonu çağrılır.

```

68
69     printf("A*B MATRİS : \n");
70     for (i = 0; i < max; i++)
71     {
72         void *k;
73         pthread_join(threads[i], &k);
74         int *p = (int *)k;
75         printf("%d ",*p);
76         if ((i + 1) % c2 == 0)
77             printf("\n");
78     }
79     return 0;
80 }

```

A matrisi ve b matrisinin çarpımları sonucu oluşan matris ekrana bastılır.

LINUX KOMUTLAR

```

(kali㉿kali)-[~/Desktop]
$ nano proje.c

```

Proje isimli c dosyası açılır.

```

(kali㉿kali)-[~/Desktop]
$ gcc proje.c -lpthread

```

Proje.c dosyası derlenir.

```

(kali㉿kali)-[~/Desktop]
$ ./a.out

```

Derleme sonucunda çıktı a.out'a yazılır. Ve bu konut ile çıktı görüntülenir.

ÇIKTI

```
(kali㉿kali)-[~/Desktop]
$ ./a.out
MATRİS A
3 6 7 5 3
5 6 2 9 1
2 7 0 9 3
6 0 6 2 6
1 8 7 9 2

MATRİS B
0 2 3 7 5
9 2 2 8 9
7 3 6 1 2
9 3 1 9 4
7 8 4 5 0

A*B MATRİS :
169 78 80 136 103
156 63 52 171 119
165 69 41 166 109
102 84 80 96 50
216 82 78 169 127
```

KAYNAKÇA

<https://www.geeksforgeeks.org/multiplication-of-matrix-using-threads/>

<https://devnot.com/2021/thread-nedir-detayli-bir-thread-incelemesi/>

<https://stackoverflow.com/questions/33224832/matrix-multiplication-with-multiple-threads-in-c>

[Matrix multiplication using threads in C programming](#)