

**ONDOKUZ MAYIS ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**İŞLETİM SİSTEMLERİ**  
**2023-2024 BAHAR DÖNEMİ**  
**Proje#1 Multi-threaded Matrix Multiplication**

**Mustafa ÇETİN**  
**20060331**

**Kodların Açıklamaları:**

1)

```
1 import java.io.*;
```

Giriş/çıkış işlemleri için gerekli sınıfları içeren kütüphane

2)

```
2 import java.util.Random;
```

Rastgele sayı üretmek için gerekli sınıfı içeren kütüphane

3)

```
6 class MatrixMultiplication {  
7     static final int SATIR1 = 4;  
8     static final int ORTAK = 1;  
9     static final int SUTUN2 = 7;
```

Matris boyutları için sabitler

4)

```
13 static int[][] A = new int[SATIR1][ORTAK];  
14 static int[][] B = new int[ORTAK][SUTUN2];  
15 static int[][] C = new int[SATIR1][SUTUN2];  
16 static int satirSayisi = 0;
```

Matrislerin tanımlanması ve thread sayısını belirlemek için kullanılan sayaç

5)

```
18 // Ana metod: main  
19 public static void main(String[] args) {  
20     randomDosyayaYaz();  
21     dosyadanAktar();
```

Rastgele A ve B matrislerini oluşturup dosyaya yazma işlemini gerçekleştirir .  
Dosyadan oluşturulan A ve B matrislerini okuma işlemini gerçekleştirir.

6)

```
24 Thread[] threadler = new Thread[SATIR1];  
25 for (int i = 0; i < SATIR1; i++) {  
26     threadler[i] = new Thread(new Carpim());  
27     threadler[i].start();  
28 }
```

Her bir satır için bir thread oluşturur ve oluşturulan threadi başlatır.

7)

```
31 for (int i = 0; i < SATIR1; i++) {  
32     try {  
33         threadler[i].join();  
34     } catch (InterruptedException e) {  
35         e.printStackTrace();  
36     }  
37 }
```

Threadlerin tamamlanmasını bekler.

8)

```

40 ~         for (int i = 0; i < SATIR1; i++) {
41 ~             for (int j = 0; j < SUTUN2; j++) {
42 ~                 if (C[i][j] < 10)
43 ~                     System.out.print(C[i][j] + " ");
44 ~                 else if (C[i][j] < 100)
45 ~                     System.out.print(C[i][j] + " ");
46 ~                 else if (C[i][j] < 1000)
47 ~                     System.out.print(C[i][j] + " ");
48 ~                 else if (C[i][j] < 10000)
49 ~                     System.out.print(C[i][j] + " ");
50 ~                 else
51 ~                     System.out.print(C[i][j] + " ");
52 ~             }
53 ~             System.out.println();
54 ~         }
55 ~     }

```

Sonuç matrisini ekrana yazdırma

9)

```

58 ~     static class Carpim implements Runnable {
59 ~         @Override
60 ~         public void run() {
61 ~             long baslangic = System.currentTimeMillis(); // İş parçacığının çalışma süresinin başlangıç zamanını alır
62 ~             int threadNo;
63 ~             synchronized (MatrixMultiplication.class) {
64 ~                 threadNo = satirSayisi++; // İş parçacığının hangi satırda çalıştığını belirler
65 ~             }

```

Thread'in çalışma süresinin başlangıç zamanını alır ve Thread'in hangi satırda çalıştığını belirler.

10)

```

67 ~         for (int i = 0; i < SUTUN2; i++) {
68 ~             for (int j = 0; j < ORTAK; j++) {
69 ~                 C[threadNo][i] += (A[threadNo][j] * B[j][i]);
70 ~             }
71 ~         }
72 ~         long son = System.currentTimeMillis();
73 ~         double calismaZamani = (son - baslangic) / 1000.0;
74 ~         System.out.printf("Çarpımın sonuç matrisinin %d. satırı %f saniyede hesaplandı!\n", threadNo + 1, calismaZamani);
75 ~     }
76 ~ }

```

Çarpma işlemini gerçekleştirir ve sonucu C matrisine kaydeder.

Thread'in çalışma süresinin sonunu alır.

Thread'in çalışma süresini hesaplar.

Thread'in çalışma süresini ekrana yazdırır.

11)

```

79 ~ static void randomDosyayaYaz() {
80 ~     try (BufferedWriter writer = new BufferedWriter(new FileWriter("matrisler.txt"))) {
81 ~         Random rand = new Random();
82 ~         // A matrisini dosyaya yazma
83 ~         for (int i = 0; i < SATIR1; i++) {
84 ~             for (int j = 0; j < ORTAK; j++) {
85 ~                 int sayi = rand.nextInt(100);
86 ~                 writer.write(String.format("%03d", sayi));
87 ~             }
88 ~         }
89 ~         // B matrisini dosyaya yazma
90 ~         for (int i = 0; i < ORTAK; i++) {
91 ~             for (int j = 0; j < SUTUN2; j++) {
92 ~                 int sayi = rand.nextInt(100);
93 ~                 writer.write(String.format("%03d", sayi));
94 ~             }
95 ~         }
96 ~     } catch (IOException e) {
97 ~         e.printStackTrace();
98 ~     }
99 ~ }

```

Rastgele oluşturulan A ve B matrisini dosyaya yazar.

12)

```

102 ~ static void dosyadanAktar() {
103 ~     try (BufferedReader reader = new BufferedReader(new FileReader("matrisler.txt"))) {
104 ~
105 ~         for (int i = 0; i < SATIR1; i++) {
106 ~             for (int j = 0; j < ORTAK; j++) {
107 ~                 char[] okuma = new char[3];
108 ~                 reader.read(okuma);
109 ~                 A[i][j] = Integer.parseInt(new String(okuma));
110 ~             }
111 ~         }
112 ~
113 ~         for (int i = 0; i < ORTAK; i++) {
114 ~             for (int j = 0; j < SUTUN2; j++) {
115 ~                 char[] okuma = new char[3];
116 ~                 reader.read(okuma);
117 ~                 B[i][j] = Integer.parseInt(new String(okuma));
118 ~             }
119 ~         }
120 ~     } catch (IOException e) {
121 ~         e.printStackTrace();
122 ~     }
123 ~ }

```

Dosyadan oluşturulan A ve B matrisini okur.

**Kodumuzun çıktısı :**

```

Çarpımın sonuç matrisinin 1. satırı 0.000000 saniyede hesaplandı!
Çarpımın sonuç matrisinin 3. satırı 0.000000 saniyede hesaplandı!
Çarpımın sonuç matrisinin 4. satırı 0.000000 saniyede hesaplandı!
Çarpımın sonuç matrisinin 2. satırı 0.000000 saniyede hesaplandı!
5376 6624 6144 5472 9408 7584 1920
3976 4899 4544 4047 6958 5609 1420
2464 3036 2816 2508 4312 3476 880
2464 3036 2816 2508 4312 3476 880

** Process exited - Return Code: 0 **

```