

Sprawozdanie

Zbieranie i Analiza Danych

Labolatorium nr 2 - Wielowątkowy dostęp do plików



Oskar Kwidziński

S156013

Fizyka Techniczna

Semestr zimowy 2022/2023

Wstęp

Wykonano zadanie mające na celu zapoznanie się z możliwością wykorzystywania trybu blokowania/współdzielenia pliku. W tym celu stworzono program wielowątkowy pozwalający na odczyt i zapis pliku tekstowego w różnych trybach ochrony. W celu blokowania plików skorzystano z biblioteki BlockEx i wrapperów platformy .NET.

Zadanie 1

Utworzono program zapisujący pojedyncze znaki char 'a' i 'b' przy pomocy poniższych funkcji:

```
1 reference
public static void WriteA()
{
    Console.WriteLine("Open by A");
    using (FileStream fs = File.Open(path, FileMode.Append, FileAccess.Write, FileShare.Write))
    {
        fs.LockEx(0, 10);
        Console.WriteLine("Locked by A");
        Byte[] info = new UTF8Encoding(true).GetBytes("A\r\n");
        fs.Write(info, 0, info.Length);
        fs.Flush();
        Console.WriteLine("Waiting in A ....");
        Thread.Sleep(100);
        fs.UnlockEx(0, 10);
        Console.WriteLine("Unlocked by A");
    }
    Console.WriteLine("Ended up by A");
}

1 reference
public static void WriteB()
{
    Console.WriteLine("Open by B");

    using (FileStream fs = File.Open(path, FileMode.Append, FileAccess.Write, FileShare.Write))
    {
        Console.WriteLine("Can lock B: " + fs.CanLockEx(0, 1000).ToString());
        fs.LockEx(0, 10);
        Console.WriteLine("Locked by B");
        Byte[] info = new UTF8Encoding(true).GetBytes("B\r\n");
        fs.Seek(0, SeekOrigin.End);
        fs.Write(info, 0, info.Length);
        fs.UnlockEx(0, 10);
        Console.WriteLine("Unlocked by B");
    }
    Console.WriteLine("Ended up by B");
}
```

W skutek wywołania ich w dwóch osobnych wątkach:

```
//Excercise 1
Console.WriteLine("-----");
Console.WriteLine("Excercise 1");
Console.WriteLine("Type enter to start writing single 'a' and 'b'");
Console.ReadKey();
Thread t1 = new Thread(WriteA);
Thread t2 = new Thread(WriteB);
t1.Start();
Thread.Sleep(1);
t2.Start();
t1.Join(); t2.Join();
```

Udało się uzyskać porządany wynik:

```
-----
Excercise 1
Type enter to start writing single 'a' and 'b'
Open by A
Open by B
Locked by A
Can lock B: False
Waiting in A ....
Unlocked by A
Locked by B
Unlocked by B
Ended up by A
Ended up by B
```

Początkowo zablokowany plik przez wątek A (dla wątku B), został odblokowany i dopiero w tym momencie doszło do zapisu tekstu do pliku poprzez wątek B.

Zadanie 2

Wykonano testy zapisu bloku znaków do pliku, o różnych długościach bloku. Zasymulowano czas trwania dokonania zapisu do pliku poprzez przyrównanie go do ilości znaków w bloku (w milisekundach) – zmienna blockLenght.

```
// Excercise 2 - FileAccess.Write
Console.WriteLine("-----");
Console.WriteLine("Excercise 2 with file access set to Write");
Console.WriteLine("Input lenght of the blocks");
int blocktLenght = Convert.ToInt32(CLRConsole.ReadLine());
Console.WriteLine("Input amount of blocks");
int blocksAmount = Convert.ToInt32(CLRConsole.ReadLine());
Console.WriteLine("Type enter to start writing block of a's and b's with file access set to Write");
Console.ReadKey();

Thread[] threads = new Thread[blocksAmount];
string A = new String(Convert.ToChar(65), blocktLenght);
string B = new String(Convert.ToChar(66), blocktLenght);

for (int i = 0; i < blocksAmount; i++)
{
    if (i % 2 == 0)
        threads[i] = new Thread(() => WriteBlock(A, FileAccess.Write));
    else
    {
        threads[i] = new Thread(() => WriteBlock(B, FileAccess.Write));
    }
    threads[i].Start();
    Thread.Sleep(100);
}

foreach (Thread t in threads)
{
    t.Join();
}

Console.WriteLine("Amount of errors: " + Error_Log.errors.ToString());
```

Zapis do pliku dokonano w ośmiu osobnych wątkach (zmienna blocksAmount), wywołując przez każdy z wątków funkcję WriteBlock w trybie FileAccess równym Write.

```
4 references
public static void WriteBlock(string block, FileAccess accessPrivilages)
{
    try
    {
        using (FileStream fs = File.Open(path, FileMode.Append, accessPrivilages, FileShare.Write))
        {
            Console.WriteLine("Open by " + block[0]);
            Console.WriteLine("Can lock " + block[0] + " : " + fs.CanLockEx(0, 1000).ToString());
            fs.LockEx(0, 1000000);
            Byte[] bytes = new UTF8Encoding(true).GetBytes(block + "\r\n");
            fs.Seek(0, SeekOrigin.End);
            fs.Write(bytes, 0, bytes.Length);
            fs.Flush();
            Console.WriteLine("Waiting in " + block[0] + " ....");
            Thread.Sleep(block.Length);
            fs.UnlockEx(0, 1000000);
            Console.WriteLine("Unlocked by " + block[0]);
            fs.Close();
        }
    }
    catch (Exception e)
    {
        Console.WriteLine("Error while writing file.");
        Error_Log.errors++;
    }
}
```

Takie działanie programu nie wyrzuciło żadnych błędów i pozwoliło zapisać każdemu z wątków blok znaków dowolnej wielkości do pliku, dopiero po odblokowaniu pliku przez poprzedni wątek.

```
-----
Excercise 2 with file access set to Write
Input lenght of the blocks
10000
Input amount of blocks
8
Type enter to start writing block of a's and b's with file access set to Write
Open by A
Can lock A : True
Waiting in A ....
Open by B
Can lock B : False
Open by A
Can lock A : False
Open by B
Can lock B : False
Open by A
Can lock A : False
Open by B
Can lock B : False
Open by A
Can lock A : False
Open by B
Can lock B : False
Unlocked by A
Waiting in B ....
Unlocked by B
Waiting in A ....
Unlocked by A
Waiting in B ....
Unlocked by B
Waiting in A ....
Unlocked by A
Waiting in B ....
Unlocked by B
Waiting in A ....
Unlocked by A
Waiting in B ....
Unlocked by B
Amount of errors: 0
```

Zadanie 3

Przetestowano różne tryby blokowania. Nie odnotowano zmiany działania programu, w przypadku zmiany trybu FileShare. Interesujący fakt odnotowano po zmianie trybu FileAccess na tryb ReadWrite w ww. funkcji Writeblock:

```
-----  
Excercise 3 with file access set to ReadWrite  
Type enter to start writing block of a's and b's with file access set to ReadWrite  
Error while writing file.  
Error while writing file.  
Error while writing file.  
Error while writing file.  
Error while writing file.  
Error while writing file.  
Error while writing file.  
Error while writing file.  
Amount of errors: 8
```

Podsumowanie

Wynik zapisu do pliku załączono do sprawozdania pod nazwą ab.txt.