

# Synchronized Threads

Pemrograman Berorientasi Objek  
S1 Informatika UNS

Pada suatu sistem multithreading, sinkronisasi adalah suatu proses pengendalian akses dari sumber daya terbagi pakai (shared resource) oleh banyak thread sedemikian sehingga hanya satu thread yang dapat mengakses suatu sumber daya tertentu pada satu waktu.

Penjelasan lebih detail silakan baca

<https://www.mahirkoding.com/threading-dan-synchronization/>

Pada modul kali ini akan dipraktekkan membuat thread tersinkronisasi.

## C++

Tulis code berikut

```
#include <iostream>
#include <thread>

using namespace std;

void func() {
    cout << "entered thread " << this_thread::get_id() << endl;
    this_thread::sleep_for(5s);
    cout << "leaving thread " << this_thread::get_id() << endl;
}

int main() {
    thread t1(func);
    thread t2(func);
    thread t3(func);

    t1.join();
    t2.join();
    t3.join();

    return 0;
}
```

Simpan file dengan nama Synchro.cpp, jelaskan output yang didapat dari source code tersebut!

Selanjutnya kita modifikasi source code dengan menambahkan library mutex untuk melakukan lock dan unlock resource, source code menjadi sebagai berikut

```
#include <iostream>
#include <thread>
#include <mutex>

using namespace std;

mutex g_lock;

void func() {
    g_lock.lock();

    cout << "entered thread " << this_thread::get_id() << endl;
    this_thread::sleep_for(5s);
    cout << "leaving thread " << this_thread::get_id() << endl;
}
```

```

        g_lock.unlock();
    }

int main() {
    thread t1(func);
    thread t2(func);
    thread t3(func);

    t1.join();
    t2.join();
    t3.join();

    return 0;
}

```

Jelaskan output yang didapat dari source code tersebut, apa yang berbeda dari hasil compile source code sebelumnya?

---

## Java

Tulis code berikut

```

class First {
    public void display(String msg) {
        System.out.print ("["+msg);
        try
        {
            Thread.sleep(1000);
        }
        catch (InterruptedException e)
        {
            e.printStackTrace();
        }
        System.out.println ("]");
    }
}

class Second extends Thread
{
    String msg;
    First fobj;
    Second (First fp,String str)
    {
        fobj = fp;
        msg = str;
        start();
    }
    public void run()
    {
        fobj.display(msg);
    }
}

public class Syncro
{
    public static void main (String[] args)
    {
        First fnew = new First();
        Second ss = new Second(fnew, "welcome");
    }
}

```

```

        Second ss1= new Second(fnew,"new");
        Second ss2 = new Second(fnew, "programmer");
    }
}

```

Simpan file dengan nama Syncro.java, compile dan jalankan program  
 Jelaskan output yang didapat setelah menjalankan program

Selanjutnya kita modifikasi source code tersebut dengan menambahkan keyword synchronized pada method display

synchronized void display (String msg)

sehingga kode akhirnya menjadi sebagai berikut

```

class First
{
    synchronized public void display(String msg)
    {
        System.out.print ("["+msg);
        try
        {
            Thread.sleep(1000);
        }
        catch(InterruptedException e)
        {
            e.printStackTrace();
        }
        System.out.println ("]");
    }
}

```

```

class Second extends Thread
{
    String msg;
    First fobj;
    Second (First fp,String str)
    {
        fobj = fp;
        msg = str;
        start();
    }
    public void run()
    {
        fobj.display(msg);
    }
}

```

```

public class Syncro
{
    public static void main (String[] args)
    {
        First fnew = new First();
        Second ss = new Second(fnew, "welcome");
        Second ss1= new Second(fnew,"new");
        Second ss2 = new Second(fnew, "programmer");
    }
}

```

Jelaskan output yang didapat setelah menjalankan program, apa yang membedakan dari eksekusi program sebelumnya?