Multithreading

Thread

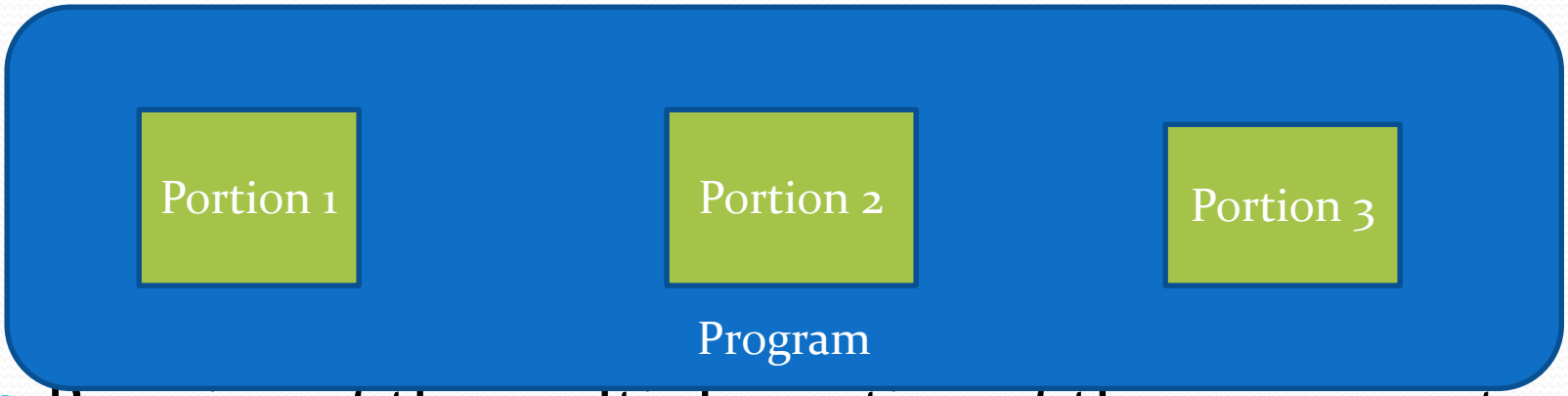
- A thread in Java is a lightweight subprocess, the smallest unit of processing. It is a separate path of execution.

Process

Processes are basically the programs that are dispatched from the ready state and are scheduled in the CPU for execution

Multithreading

- It is a process of executing multiple threads simultaneously.



- Running of the multiple portion of the program at a same time

Thread Synchronization

- It is a option where we want to allow only one thread to access the shared resource.
- It is used to prevent thread interference.
- It is used to prevent consistency problem
- It can be used with method, variables and blocks
- Example: Producer-Consumer

Multitasking vs Multithreading

Multitasking

- Multi-tasking is the ability of an operating system to run multiple processes or tasks concurrently, sharing the same processor and other resources.
- It is the feature of operating system

Multithreading

- Execution of the multi threads at a same time of a same program sharing the memory area
- It is the feature of programming language

Advantages of Multithreading

1. It doesn't block the user because threads are independent and you can perform multiple operations at the same time
2. You can perform many operations together so it saves time
3. Threads are independent, so it doesn't affect other threads if an exception occurs in a single thread

Applications of Multithreading

1. It is used in video games
2. It is used to create media players
3. It is used to build servers
4. It is used in web browsers to load multiple web pages at the same time
5. Used by the Databases


00:45
25.6 KM
04
COVER 2KM IN 60 SECONDS
234
KM/H
T2
T3


Cisdem Video Player
File
Convert
Edit
Playback
Audio
Subtitles
Window
Help
Dr.STRANGE_WEB_DL_1080p.mkv
00:35:12
01:55:15

Thread Creation in Java

- **Using Thread Class**
  We can use extends the Thread class provided by Java.lang package and write code in run() method

- **Using Runnable Interface**
  We can implements Runnable interface and implements the run() method

Geekyshows.java

```java
package mainn;
class Mythread extends Thread
{
    public void run()
    {
        for(int i =0; i<5; i++)
        {
            System.out.println("Child Thread");
        }
    }
}
public class Geekyshows
{
    public static void main(String args[])
    {
        Mythread t = new Mythread();
        t.start();
        for(int i=0; i<5; i++)
        {
            System.out.println("Main Thread");
        }
    }
```

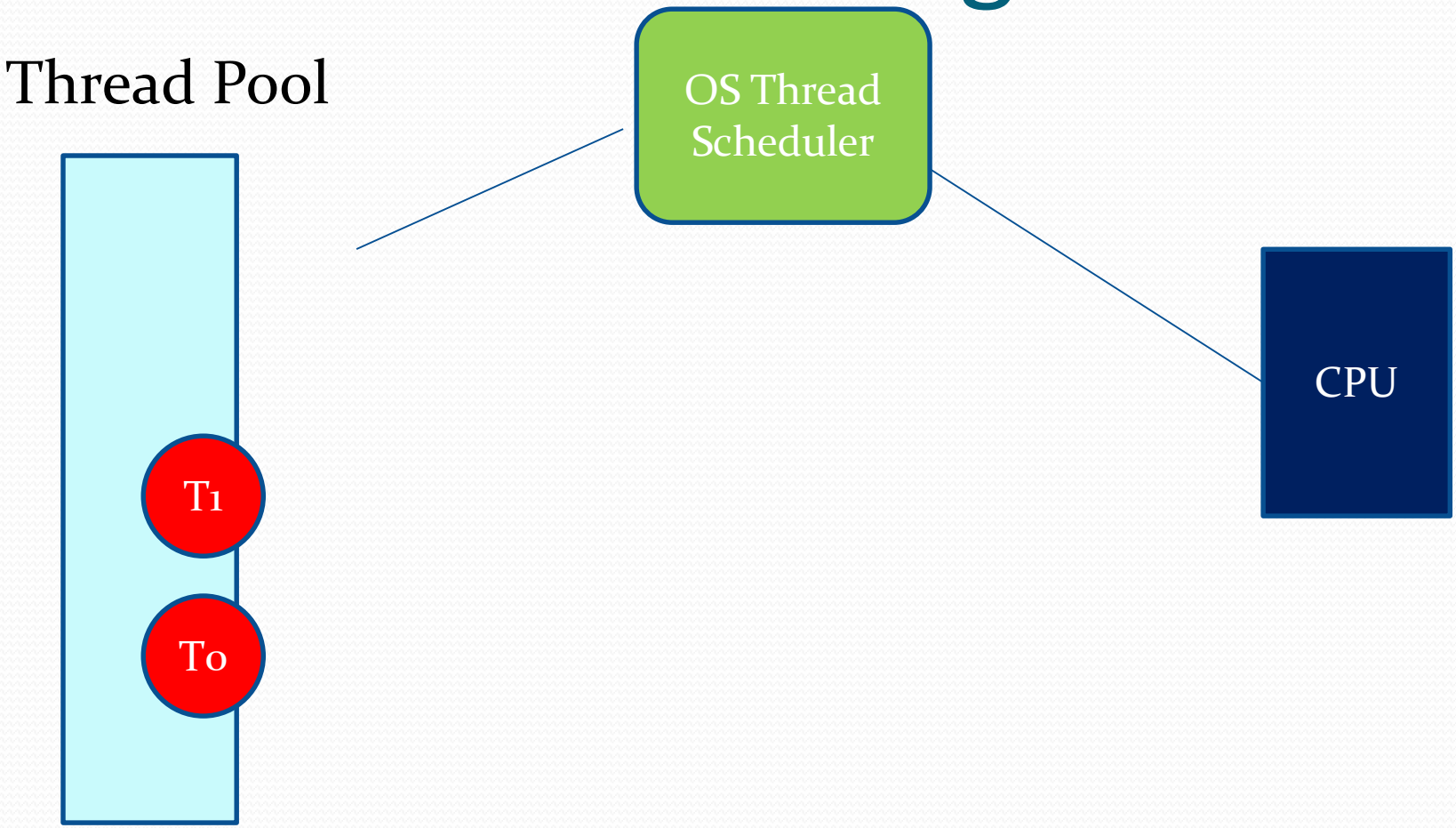
```
public class DemoThread  implements  Runnable{
        public void run() {
                for(int i=0;i<1000;i++) {
                System.out.println(" hey thread1 started");
                }
        }
        public static void main(String[] args) {
                DemoThread d=new DemoThread();
                Thread t1=new Thread(d);
                t1.start();

                DownloadThread down =new DownloadThread();
                Thread t2=new Thread(down);
                t2.start();
        }
}
```

Implements Runnable interface

Creating instance of the class

Creating instance of thread class and starting the thread

Execution of the Program



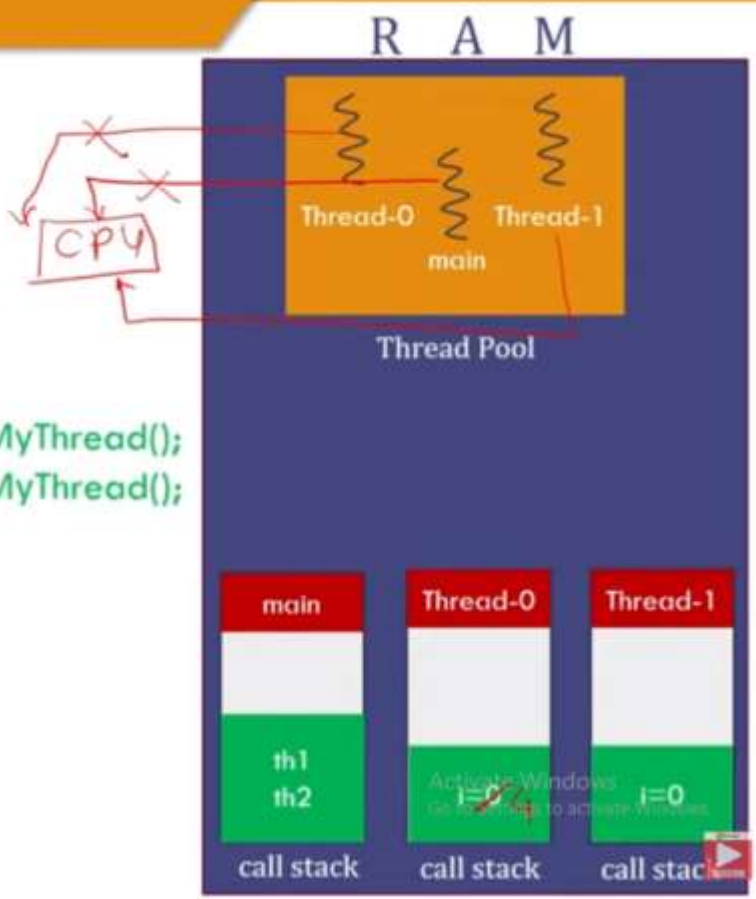
Working of Start method

- Working of start() method

```
class MyThread
{
    public void run()
{
  for(int i=0;i<=10;i++)
   {
     s.o.pln(i);
   }
}

}
```

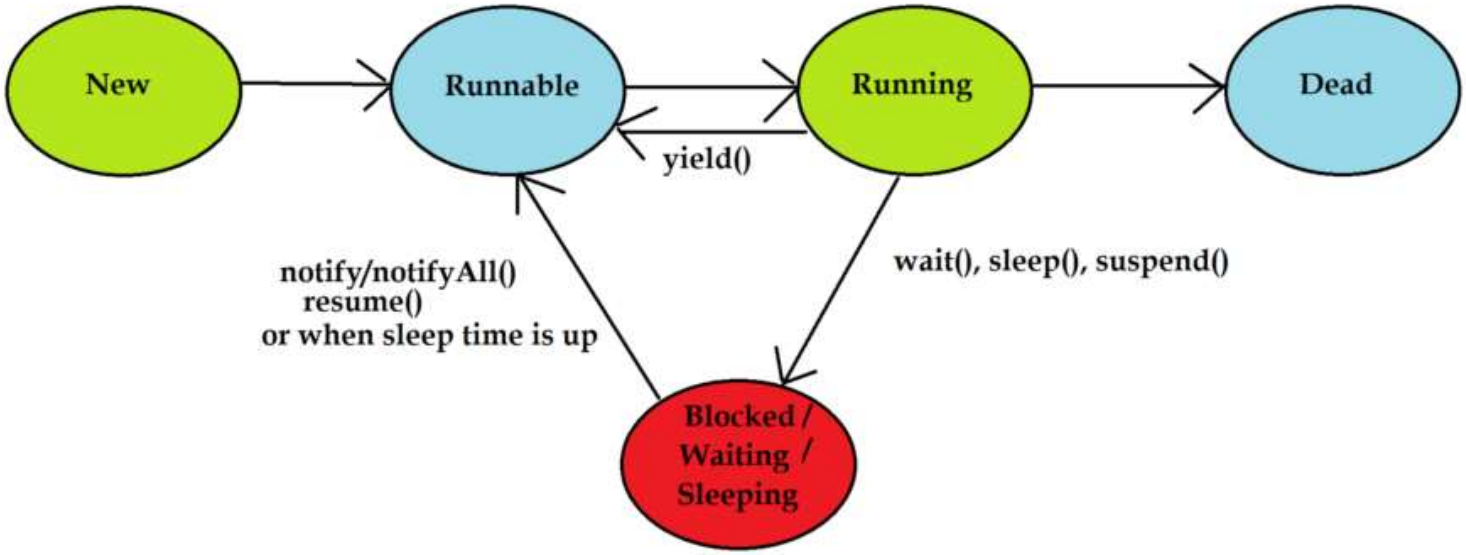
```
class ThreadDemo
{
  public static void
     main(String [] args)
{
    MyThread th1=new MyThread();
    MyThread th2=new MyThread();
    th1.start();
    th2.start();
}

}
```

0 1 2 3



- What if we call run instead of start method

Life Cycle of Thread



Thread Lifecycle using Thread states

Life Cycle method of a Thread

1. **New**
   When a new thread is created using new keyword
2. **Runnable**
   Thread is created and present in thread pool to get execute
3. **Running**
   Thread is currently executing by the CPU
4. **Waiting**
   Thread is waiting or timed waiting for other threads to gets complete their task
5. **Dead**
   Execution of the thread is done by the processor

Different Methods

- Sleep()
- Join()
- Notify()
- notifyAll()
- Wait()

Some other concepts

- Thread Name
- Thread Priority

What are the daemons thread

What is race condition

Thread Synchronization

- What is Synchronization, critical section
- Synchronization keyword
- Why it is required, explain with example
- Mutual Exclusive synchronization
- Conditional synchronization
- Producer consumer problem