

Ch11 쓰레드

객체지향프로그래밍(기본) 2019
경상대학교 항공우주및소프트웨어공학전공

11.1 쓰레드란

쓰레드는 하나의 흐름으로 프로세스 내에 하나이상 존재한다.

쓰레드는 프로세스와 자원을 공유함으로써 공유된 자원의 문제가 있을시 쓰레드에 영향을 미친다.

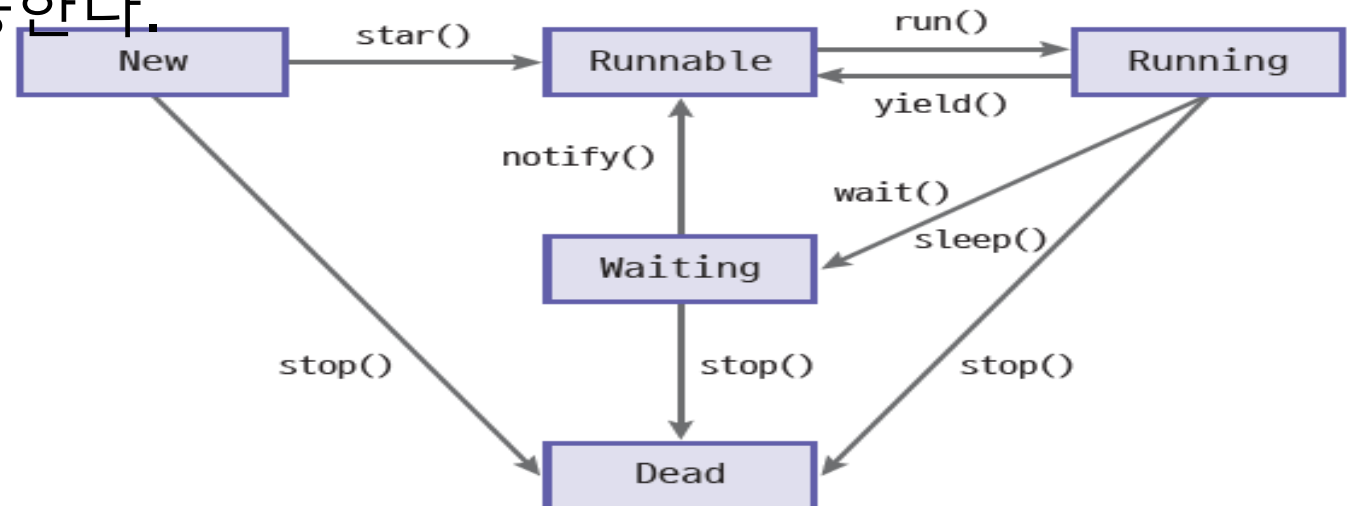
쓰레드가 프로세스에 하나 있을 때 단일 스레드 프로세싱이라 하며 쓰레드가 프로세스 내에 여러 개 존재하면 멀티 쓰레드 프로세싱이라 한다.

쓰레드의 일생

쓰레드가 객체를 생성하고 `start()` 메소드를 이용하여 `Runnable` 상태로 변경한다.

`Runnable` 상태는 실행되는 것이 아니라 실행 대기 상태를 의미한다. 실행 대기 상태에서 `run()` 메소드를 사용하면 쓰레드가 실행되고 쓰레드의 실행 상태에서 `stop()` 메소드를 사용하면 쓰레드는 없어진다.

쓰레드의 실행 중에 쓰레드를 없애지 않고 잠시 대기 상태로 만들기 위해 `sleep()`나 `wait()` 메소드를 사용한다.



11.2 쓰레드의 생성

자바에서 쓰레드는 여러 개 사용 할 수 있지만 Thread 클래스는 하나이다. 새로운 쓰레드를 생성 할 때 Thread 객체를 생성해야 한다.

쓰레드를 시작하는 방법은 Runnable인터페이스를 이용하는 방법과 Thread 클래스를 상속 받는 두 가지 방법이 있다.

Runnable 인터페이스

Runnable 인터페이스를 이용한 방법은 먼저 Runnable 인터페이스를 가지는 객체를 Runnable 레퍼런스를 이용하여 생성한다.

Runnable 인터페이스를 가진 객체에 스레드에서 할 작업을 구현 한다.

Thread 객체를 생성하여 Runnable 레퍼런스를 전달하고 Thread 객체를 실행 한다.

```
public class MyRunnable implements Runnable{  
  
    public void run() { ← Runnable 인터페이스의 run 메소드를 오버라이드 함  
        System.out.println("동작을 실행 합니다.");  
    }  
}
```

```
public class ThreadTest {  
    public static void main(String[] args) {  
  
        Runnable threadRun = new MyRunnable();  
        Thread myThread = new Thread(threadRun);  
  
        myThread.start();  
    }  
}
```

Thread 클래스

Thread 클래스를 상속 받아 스레드를 생성하는 할 때는 상속받은 자식 클래스에 `run()` 메소드를 오버라이드 하여 스레드의 동작을 구현 한다.

```
public class MyThread extends Thread{  
  
    public void run() { ← Thread 클래스의 run 메소드를 오버라이드 함  
        System.out.println("동작을 실행 합니다.");  
    }  
}
```

```
public class ThreadTest {  
    public static void main(String[] args) {  
  
        Thread myThread = new MyThread();  
  
        myThread.start();  
    }  
}
```

11.3 쓰레드의 실행 및 제어

Thread 클래스는 쓰레드의 상태를 제어하기 위해 다음과 같은 메소드를 제공한다.

메소드	기능
<code>static void sleep(long msec) throws InterruptedException</code>	지정된 시간동안 쓰레드를 대기시킴, 사용 시 예외처리를 해주어함
<code>void start()</code>	쓰레드를 시작 run() 메소드를 호출함
<code>void join() throws InterruptedException</code>	쓰레드가 끝날 때 까지 대기함, 사용 시 예외처리를 해주어함
<code>void suspend()</code>	쓰레드를 일시 정지시킴
<code>void resume()</code>	일시 정지된 쓰레드를 다시 실행시킴
<code>void yield()</code>	다른 쓰레드에게 실행 상태를 양보하고 준비 상태로 돌아감
<code>void stop()</code>	쓰레드를 종료함.

sleep()

sleep() 메서드를 사용하여 스레드의 동작을 지정된 시간 동안 정지시킬 수 있다.

```
try {  
    Thread.sleep(1000);  
}catch (InterruptedException e) {  
}
```

마이크로세컨즈 단위

11.4 멀티 쓰레드

여러 개의 쓰레드를 실행하기 위해서는 쓰레드 객체를 여러 개 생성하면 된다.

```
public class MyThread extends Thread{

    public void run() {
        System.out.println(this.getName()+"가 동작을 합니다.");
    }
}
```

```
public class ThreadTest {
    public static void main(String[] args) {

        Thread myThread1 = new MyThread();
        Thread myThread2 = new MyThread();

        myThread1.setName("Thread One");
        myThread2.setName("Thread Two");

        myThread1.start();
        myThread2.start();
    }
}
```

Thread One가 동작을 합니다.

Thread Two가 동작을 합니다.

멀티 쓰레드의 동작

쓰레드의 작동 순서는 무작위로 진행된다.

쓰레드는 스케줄러에 의해 진행 순서가 달라지며 쓰레드의 작동 순서를 정확하게 확정 할 수는 없다.

```
public class MyThread extends Thread{  
  
    public void run() {  
        for(int i = 0; i < 5 ;i++)  
            System.out.println(this.getName()+"가 동작을 합니다.");  
    }  
}
```

```
Thread myThread1 = new MyThread();  
Thread myThread2 = new MyThread();  
  
myThread1.setName("Thread One");  
myThread2.setName("Thread Two");  
  
myThread1.start();  
myThread2.start();
```

Thread Two가 동작을 합니다.
Thread One가 동작을 합니다.
Thread Two가 동작을 합니다.
Thread Two가 동작을 합니다.
Thread One가 동작을 합니다.
Thread Two가 동작을 합니다.
Thread One가 동작을 합니다.
Thread Two가 동작을 합니다.
Thread One가 동작을 합니다.
Thread One가 동작을 합니다.

11.5 동기화

자바에서는 한 스레드가 먼저 선점한 자원을 작업이 끝날 때 까지 다른 스레드가 점유 하지 못하도록 하여 이런 문제를 방지할 수 있는데 이것을 동기화라 한다.

하나의 스레드가 공유하는 데이터 값을 변경하기 전에 잠깐 쉬고 있을 때 다른 스레드가 끼어들어 값을 변경했기 때문이다.

이러한 문제점을 해결하는 방법으로 `synchronized` 키워드를 사용하여 동기화 시킨다.