

13장 스레드



스레드의 개념

◆ 프로세스(Process)

프로세스란 단순히 **실행 중인 프로그램**(Program)이라 할 수 있다.

사용자가 작성한 프로그램이 운영체제에 의해 메모리 공간을 할당 받아 실행 중인 것을 말한다.

이러한 프로세스는 프로그램에 사용되는 데이터와 메모리 등의 자원, 그리고 스레드로 구성된다.



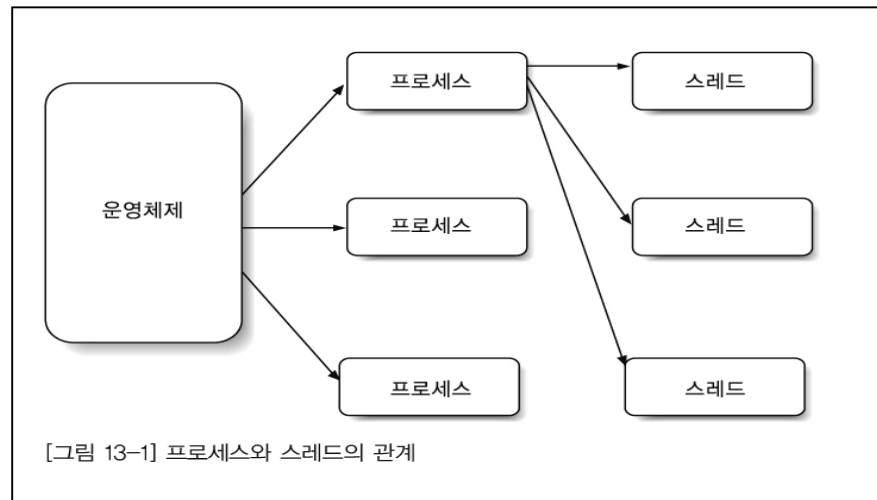
스레드의 개념

◆ 스레드(Thread)

스레드란 프로세스 내에서 실제로 작업을 수행하는 주체를 의미한다.

모든 프로세스에는 **한 개 이상의 스레드**가 존재하여 작업을 수행한다.

두 개 이상의 스레드를 가지는 프로세스를 멀티스레드 프로세스라고 한다.



멀티스레드 장점

프로그램 속도 증가

스레드 사용

◆ Thread 클래스를 상속하는 방법

start() : 스레드를 시작하는 메소드

run() : 실제 작업을 수행하는 메소드

sleep(time) : 주어진 시간만큼 실행 흐름을 중지 시키는 메소드

```
public class LecThread extends Thread{  
  
    @Override  
    public void run() {  
        // 현재 실행중인 스레드의 이름 출력  
        System.out.println(getName());  
    }  
  
}
```

```
public static void main(String[] args) {  
    // Thread를 상속받은 클래스 객체를 이용  
    LecThread lecThread = new LecThread();  
    lecThread.start(); // 재정의한 run 메소드가 실행된다.  
}
```

스레드 사용

◆ Runnable 인터페이스를 구현하는 방법

run() 메소드를 반드시 재정의(Override)해야 한다.

재정의한 클래스를 Thread 클래스 생성자의 매개변수로 입력하여 Thread 객체를 생성해서 사용한다.

```
public class LecRunnable implements Runnable{

    @Override
    public void run() {
        // 현재 실행중인 스레드의 이름 출력
        System.out.println(Thread.currentThread().getName());
    }

}
```

```
public static void main(String[] args) {
    // Runnable을 구현한 클래스 객체를 이용
    Thread lecThread = new Thread(new LecRunnable());
    lecThread.start(); // 재정의한 run 메소드가 실행된다.
}
```

스레드 사용

◆ 동기화(Synchronized)

만약 서로 다른 스레드에서 같은 데이터를 사용하게 되는 경우, 해당 데이터를 동기화 시켜주어야 한다.

동기화(Synchronized) 블록이나, 메소드 내에서는 스레드를 제어할 수 있는 메소드를 사용할 수 있다.

◆ 스레드의 제어를 위한 메소드들

setPriority(): 스레드가 Run 상태에 들어갈 수 있는 우선권을 결정

sleep(): 스레드를 일정시간 동안 멈추게 한다.

wait(): 스레드를 대기상태(NotRunnable)로 보낸다.

notify(): 대기상태에 있는 스레드들 중 하나를 Runnable 상태로 복귀시켜 작업을 재개하게 한다.

notifyAll(): 대기상태에 있는 모든 스레드를 Runnable 상태로 복귀시켜 작업을 재개하게 한다.