

● 태스크(Task) 관리: 대부분 시스템에서 액티비티를 알아서 잘 관리하고 수행해 주지만, 때에 따라 관리자가 시스템에 태스크 관리가 어떻게 되어야 한다고 명시해야 할 수도 있다.

▶ 시스템의 태스크 관리: 하나의 앱이 실행될 때 프로세스(process)와 함께 안드로이드에서는 태스크(스택)이 실행 되는데, 이는 일종의 앱 실행을 위한 정보 저장공간으로 안드로이드 시스템에서는 이 태스크 정보를 이용해 앱을 어떻게 수행할지 결정한다. 흔히 프로세스를 앱의 물리적인 실행 단위라고 표현하고, 태스크는 앱의 논리적인 실행 단위라고 표현한다. 태스크는 앱 내에서 실행 중인 액티비티의 각종 정보(액티비티의 프로그램적인 데이터가 아닌 시스템 내부 정보)를 저장하기 위한 공간이다.

안드로이드 시스템의 내부 규칙:

사용자가 실행한 앱이 두 개라면, 프로세스는 두 개가 실행되고, 안드로이드 시스템에서는 앱을 위한 태스크도 두 개 만든다. 태스크 목록에는 각 앱 마다, 각 액티비티의 컴포넌트 정보가 기록된다.

만약 하나의 앱의 액티비티에서 인텐트로 다른 앱의 액티비티를 실행한다면, 태스크 목록은 하나만 만들어진다. 두 앱 간의 연동이 발생하였더라도 사용자는 하나의 앱만 실행한 것이기 때문이다. 흔히 태스크를 사용자 관점을 고려한 프로그램의 실행단위라고 표현하는데, 사용자 관점에서는 하나의 앱만 실행한 것이므로 태스크는 하나만 만들어지는 것이다. 이처럼 태스크 관리는 사용자 관점을 고려한 일관된 앱 실행 상태를 유지하는데 목적이 있다. 이와 같은 상황이라면 첫번째 앱을 위한 태스크 목록에 두 개의 액티비티 정보가 유지된다.

만약 두 개의 앱이 실행되는데 한 액티비티가 이미 실행된 상태에서 다시 인텐트에 의해 실행되는 경우, 액티비티는 절대 싱글톤(Singleton - 어떤 클래스가 단 하나의 객체만 생성되는 경우; 개발자 알고리즘으로 하나의 객체만 생성되게 구현한다)으로 움직이지 않으며 이미 한 액티비티의 객체가 생성되어 이용되고 있더라도 다시 그 액티비티를 실행하기 위한 인텐트가 어디선가 발생하면 다시 생성된다. 즉, 하나의 액티비티 클래스는 인텐트가 발생하면 매번 객체가 생성되는 구조이다. 따라서 같은 액티비티가 두 번 실행된 경우 객체도 두 개 생성되어 각각 텍스트 목록에 저장된다.

▶ 태스크 제어: 시스템의 태스크 관리 규칙은 대부분 시스템에서 알아서 하지만 개발자가 직접 제어해야 할 때도 있다. 태스크 관리를 제어하는 방법은 두 가지이다.

1) 액티비티를 등록하는 AndroidManifest.xml 파일의 <activity> 태그의 launchMode 속성을 사용하여 해당 액티비티가 실행될 때 항상 태스크 목록에 어떻게 올라가야 한다고 명시하는 방법이다.

```
<activity
    android:name=".DetailActivity"
    android:exported="false"
    android:launchMode="singleTop"/>
```

2) 액티비티를 실행하는 인텐트에 플래그(Flag)를 이용해서 이번 실행 때는 태스크 목록에 어떻게 올라가야 한다고 명시하는 방법이다.

```
Intent intent = new Intent(this, DetailActivity.class);
intent.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
startActivity(intent);
```

☞ standard: 기본값이며, 설정되지 않았을 때다. 인텐트가 발생하면 매번 액티비티를 생성하고, 태스크 목록에 반복해서 올리겠다는 의미이다. 안드로이드에서는 얼마든지 자신이 자신을 인텐트로 실행할 수 있다. 액티비티 객체는 매번 생성되고, 생성된 객체 정보가 태스크에 매번 저장된다. 사용자가 뒤로가기 버튼을 누르면 같은 화면이 계속해서 나온다.

☞ singleTop: 인텐트로 실행하려는 액티비티가 태스크의 최상단에 있으면 다시 생성하지 않겠다는 의미이다. 객체도 생성되지 않고 화면도 변경되지 않는다. 태스크 목록의 최상단에 있는 액티비티는 사용자 화면에 출력될 가능성이 있는데, 사용자가 보고 있는 액티비티의 객체를 다시 생성해서 화면을 전환하는 것이 비정상적으로 생각될 때 사용하면 된다. singleTop으로 선언한 액티비티는 intent가 발생할 경우 객체 생성은 되지 않고 그 액티비티의 onNewIntent() 함수만 반복 콜 되고, 이곳에서 화면의 데이터를 조정하면 된다. 다만 만약 singleTop으로 선언된 액티비티가 태스크 목록의 최상위에 있지 않으면, 액티비티가 다시 생성된다.

```
<activity
    android:name=".OneActivity"
    android:launchMode="SingleTop">
</activity>
```

실전에서 singleTop으로 선언해야 하는 대표적인 상황은 알림(Notification)으로 인해 액티비티가 실행되는 경우이다. 만약 standard로 설정할 경우 알림에 의해 실행될 때 새로운 액티비티가 생성되어 태스크가 저장되고, 뒤로가기 버튼을 누르면 같은 액티비티가 한 번 더 보이게 된다. 하지만 singleTop으로 설정하면 액티비티는 생성되지 않으며 태스크 목록에는 하나의 정보만 저장된다. 내용은 다르더라도 같은 액티비티가 두 번 실행되어 뒤로 가기 버튼에 의해 다시 나오는 게 이상하다고 생각되면 singleTop으로 선언하면 된다.

Ex) 카톡 앱으로 친구1과 대화를 하고 있던 도중 (chatActivity), 친구2에게 알림이 오고 그것을 클릭하면 친구2와의 대화창이 뜨게 된다. standard로 설정한 경우 알림을 클릭하는 순간 intent로 또 다른 chatActivity의 객체가 생성되고 태스크에 저장돼서 친구2와의 대화창이 뜨고, 이 상태에서 뒤로가기 버튼을 누르면 다시 친구1과의 대화창으로 돌아간다. 하지만 카카오톡은 이렇게 되지 않고, 뒤로가기 버튼을 누를 경우 앱이 종료된다. 이럴 경우 chatActivity을 SingleTop으로 설정하면 된다. 그러면 chatActivity가 태스크의 최상단에 있기 때문에 다시 생성하지 않는다. 다만 chatActivity의 내용을 친구1에서 친구2로 바꾸어야 한다.

이 경우 launchMode 설정만으로 객체 생성 없이 한 액티비티의 내용을 보여주다가 알림을 눌렀을 때 다른 내용으로 전환해야 하는 상황인데, 객체 생성 없이 객체에서 이 상황을 감지하기 위해 onNewIntent() 함수를 제공한다. singleTop으로 선언하면 태스크 목록에서 액티비티가 최상위에 있을 때 다시 생성 없이 위의 함수만 다시 호출된다. 이곳에서 화면의 데이터를 조정하면 된다.

```
@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    // 내용 변경
}
```

☞ singleTask: 액티비티가 실행되는 시점에 새로운 태스크 목록을 만들어 그곳에 액티비티 정보를 저장한다. 이 설정은 같은 앱에서는 적용되지 않고, 다른 앱의 액티비티를 인텐트로 실행할 때만 적용된다. 새로운 앱을 실행하지 않은 상태에서 사용자에게 새로운 앱이 실행되었다는 것을 알리고 싶을 때 사용한다. 한 앱에서 인텐트를 통해 다른 앱의 액티비티를 실행했을 때 해당 액티비티를 담은 새로운 태스크가 만들어진다는 (그리고 그 이후에 수행되는 컴포넌트들은 해당 태스크에 쌓인다).

☞ singleInstance: singleInstance가 설정된 액티비티는 혼자 새로 만들어진 하나의 태스크를 차지해 저장되며, 이후에 수행되는 컴포넌트들은 또 다른 새로 만들어진 태스크에 쌓인다. (액티비티가 앱 하나처럼 보이고 싶을 때 사용).

