

액티비티와 생명주기

성신여자대학교 컴퓨터공학과
우 종 정 교수

애플리케이션, 프로세스, 태스크

■ 안드로이드 애플리케이션

- 액티비티, 서비스, 콘텐츠 공급자, 방송 수신자가 느슨하게 결합된 컴포넌트로 구성
- 결합된 컴포넌트에 대한 내역은 매니페스트 파일의 <application> 엘리먼트에 명시
- 새로운 액티비티가 시작할 때마다 이전 액티비티는 정지되지만 시스템은 이전 액티비티를 액티비티 스택 혹은 백스택이라는 스택에 보존

■ 프로세스

- 일반적 의미: 메모리에 적재되어 실행되고 있는 프로그램
- 안드로이드의 프로세스: 애플리케이션 컴포넌트가 실행되는 장소로써 가상머신으로 분리된 물리적인 실행 단위

애플리케이션, 프로세스, 태스크

■ 안드로이드 애플리케이션은 가상머신에서 다음과 같은 환경으로 수행

- 기본적으로 하나의 애플리케이션은 하나의 프로세스로 동작
- 애플리케이션이 시작될 필요가 있으면 안드로이드는 시스템 자원을 확보하여 프로세스를 시작하며, 더 이상 필요가 없거나 시스템 자원이 부족하면 프로세스를 종료 → 애플리케이션의 명시적 종료 불필요
- 모든 프로세스는 자신의 가상머신을 가지기 때문에 다른 프로세스와는 격리되어 실행
- 파일을 공유하기 위한 여러 가지 방법이 있지만 파일을 가진 애플리케이션은 일반적으로 자신만 파일에 접근 가능

■ 안드로이드 시스템은 메모리가 부족할 경우 중요하지 않은 순서대로 프로세스를 메모리에서 제거

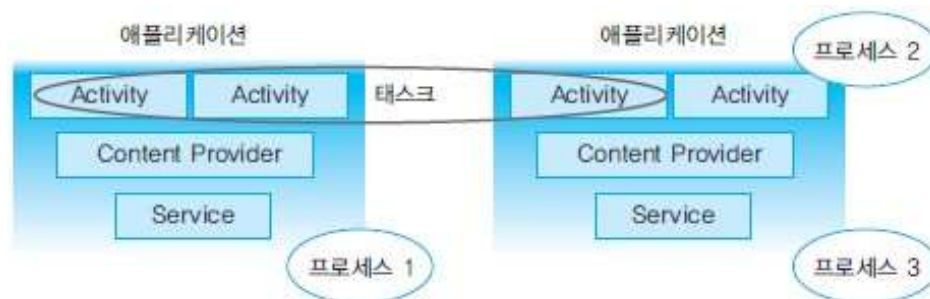
■ 중요도 순서에 의한 프로세스 종류

- 포그라운드 프로세스
- 비지블 프로세스
- 서비스 프로세스
- 백그라운드 프로세스
- 공백(empty) 프로세스

애플리케이션, 프로세스, 태스크

■ 태스크

- 액티비티는 다른 애플리케이션에 존재하는 액티비티까지 호출 가능 → 실행 측면에서 보면 애플리케이션 컴포넌트는 무경계
- 사용자와 상호작용하는 액티비티는 애플리케이션의 단위보다 태스크 단위로 관리



- 태스크는 동일한 목적을 위한 액티비티의 집합으로서 사용자에게 하나의 애플리케이션처럼 간주되는 논리적 실행 단위
- 애플리케이션과는 달리 태스크는 매니페스트 파일에 명시할 엘리먼트가 없고, 단지 액티비티의 스택에 불과
- [Back] 버튼을 누르면 현재 실행 중인 액티비티는 백스택에서 제거되며, 이전 액티비티가 실행 액티비티로 재개

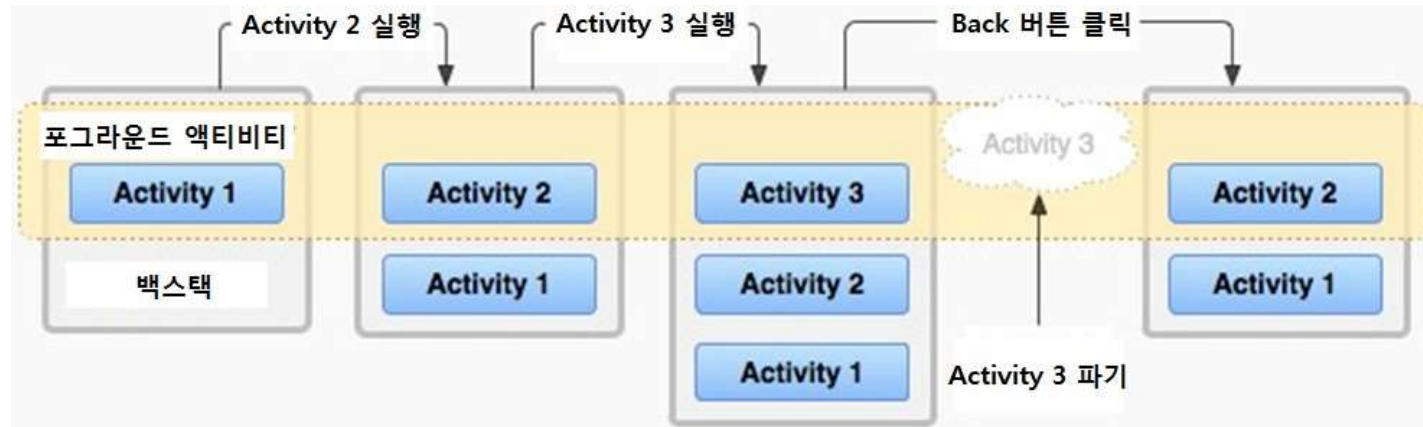
애플리케이션, 프로세스, 태스크

■ 액티비티와 태스크에 대한 디폴트 동작 방식

- 액티비티 A가 액티비티 B를 시작하면 액티비티 A는 중지 혹은 정지되고 시스템은 액티비티 A의 상태를 보유. 액티비티 B의 실행 도중에 [Back] 버튼을 누르면 액티비티 A의 상태를 복원하고 재시작.
- [Home] 버튼을 눌러 태스크를 떠나면 실행중인 액티비티가 정지되고 태스크는 백그라운드로 이동. 시스템은 태스크에 포함된 모든 액티비티의 상태를 보유. 태스크를 시작한 런처 아이콘을 선택하여 태스크를 재시작하면 태스크는 포그라운드로 돌아오고 백스택의 최상위에 있는 액티비티가 화면에 나타남.
- [Back] 버튼을 누르면 실행중인 액티비티가 백스택에서 제거 및 파기되며, 백스택의 차상위 액티비티가 재개. 시스템은 파기된 액티비티의 상태는 비보유.
- 태스크에서 동일 액티비티의 인스턴스를 여러 개 생성 가능.
- 태스크에 포함된 액티비티는 응집된 단위로 이동

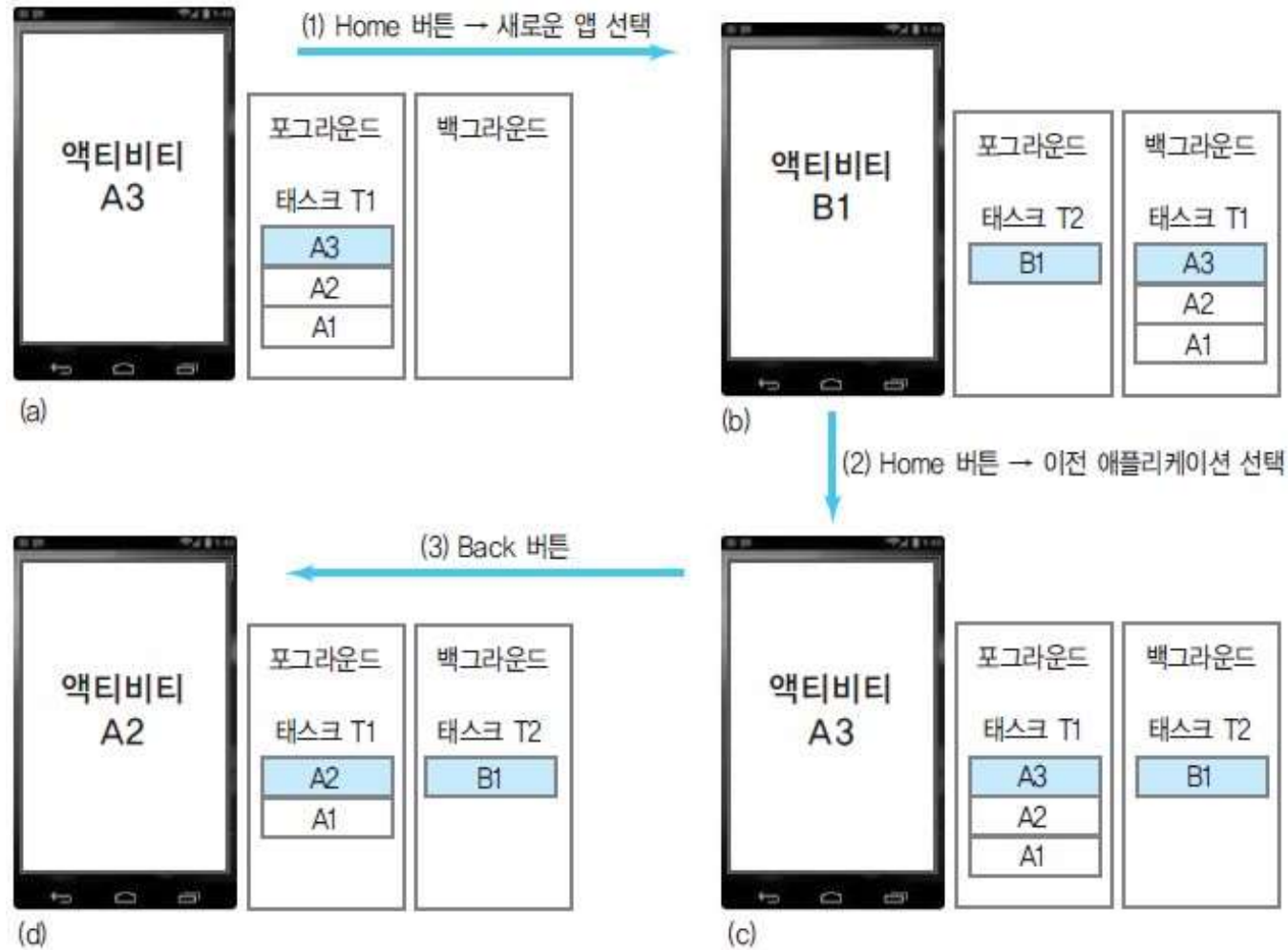
애플리케이션, 프로세스, 태스크

■ 백스택 예제 1



애플리케이션, 프로세스, 태스크

■ 백스택 예제 2



액티비티 생명주기

■ 생명주기

- 모바일 컴퓨팅의 부족한 자원 → 메모리와 같은 자원이 부족하더라도 걸려오는 전화 수신은 필수
→ 안드로이드는 시스템 자원이 부족하면 사용자의 동의 없이 프로세스 강제 종료
- 생명주기란 프로세스의 생성 및 소멸을 포함한 일련의 과정

■ 액티비티 상태

- 활성 상태 혹은 실행 상태
- 중지 상태
- 정지 상태

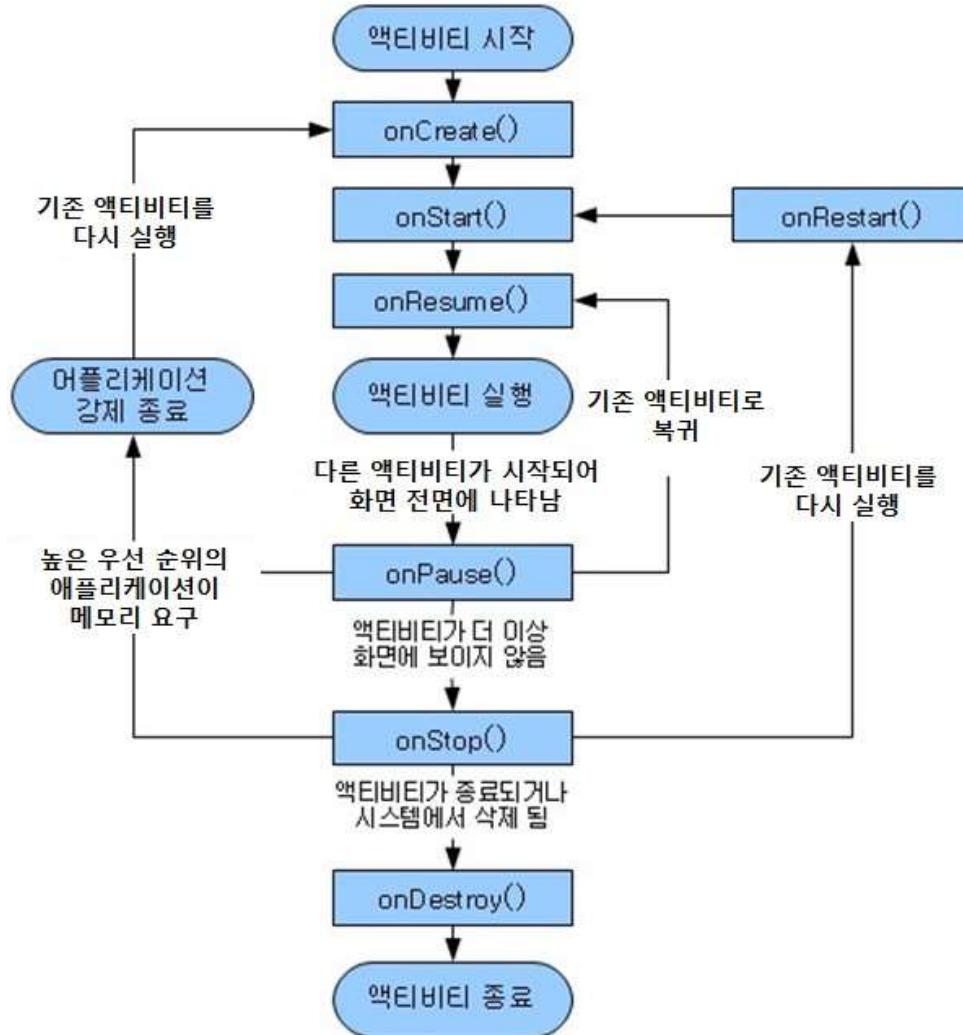
액티비티 생명주기

■ 생명주기 메서드

```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // 액티비티가 생성된다.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // 액티비티가 보이기 시작한다.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // 액티비티가 보이고 사용자와 상호작용할 수 있다.
    }
    @Override
    protected void onPause() {
        super.onPause();
        // 다른 액티비티에게 포커스를 빼앗긴다.
    }
    @Override
    protected void onStop() {
        super.onStop();
        // 액티비티가 더 이상 보이지 않는다.
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // 액티비티가 파괴되기 시작한다.
    }
}
```

액티비티 생명주기

■ 액티비티의 상태 천이



액티비티 생명주기

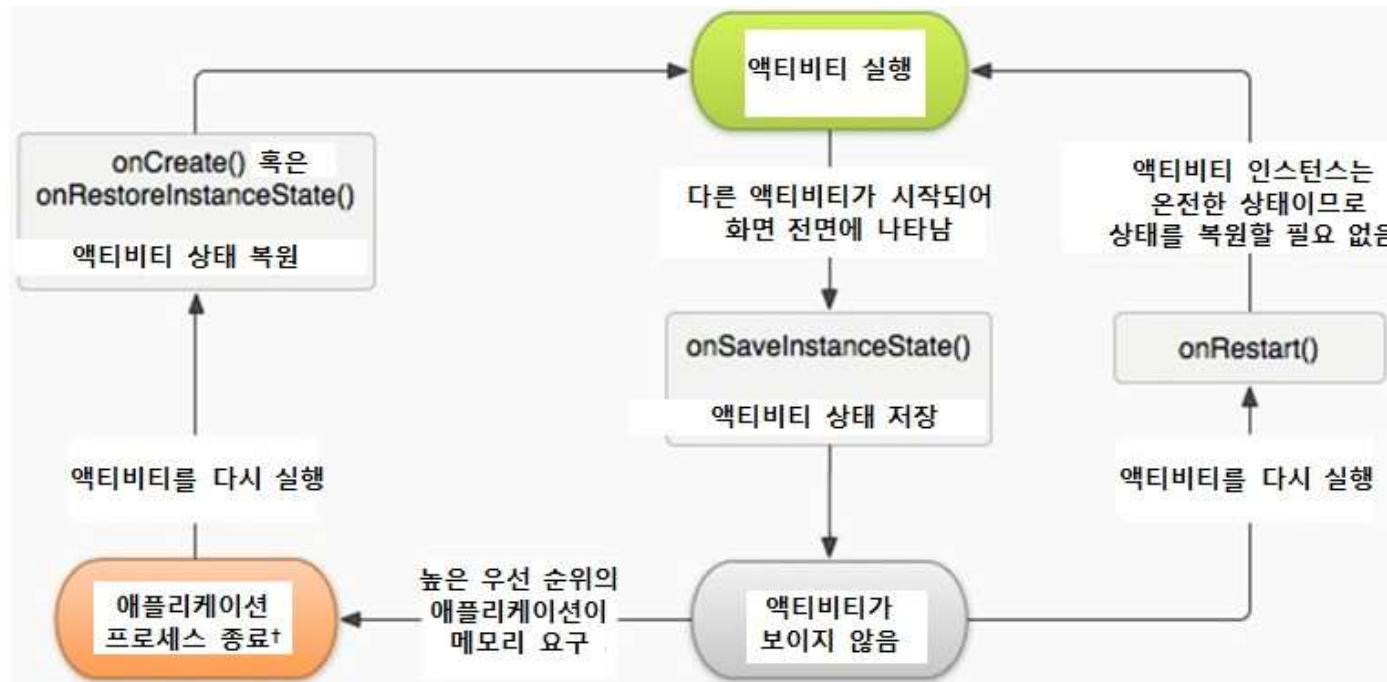
■ 액티비티의 상태 복원

- 생명주기 메소드에 의한 상태 복원
 - 생명주기 메소드를 사용하면 액티비티가 중지 혹은 정지되더라도 액티비티의 상태는 유지. 따라서 중지 혹은 정지된 액티비티가 복귀 혹은 다시 실행되어 포그라운드 상태가 되면 원래 상태로 복귀
 - 액티비티의 상태를 Bundle 형태로 저장해두면 다시 시작할 때 액티비티를 원래대로 화면에 표시. 액티비티가 처음 시작하는 경우라면 onCreate() 메소드의 매개변수인 Bundle 객체는 null을 포함
- 생명주기 메소드에 의한 상태 복원 문제점
 - 시스템이 메모리 부족 등의 이유로 액티비티를 파기 가능 → 액티비티 복원 불가. 따라서 액티비티를 다시 시작하면 시스템은 액티비티 인스턴스를 재생성
 - 사용자는 시스템에 의한 액티비티의 파기 여부를 알 수 없음

액티비티 생명주기

■ 액티비티의 강제 종료와 상태 복원

- 강제 종료의 경우에 상태를 저장하기 위하여 안드로이드는 개발자에게 다음 메소드를 제공
 - onSaveInstanceState(): 액티비티가 보이는 마지막 시점에 호출되어 상태를 저장
 - onRestoreInstanceState(): 액티비티가 파기된 후 다시 시작되는 시점에 호출되어 상태를 복원



액티비티 생명주기 응용

■ 액티비티의 생성과 소멸

- 액티비티 실행을 통한 생명주기 메소드 호출 현황
- 메인 액티비티 실행과 5가지 상황 ([액티비티](#))
 - 프로젝트의 실행

L...	Time	PID	TID	Application	Tag	Text
I	09-10 00:24:02.978	596	596	com.corea.activityldemo	Life Cycle	Main.onCreate(Bundle) invoked
I	09-10 00:24:02.978	596	596	com.corea.activityldemo	Life Cycle	Main.onStart() invoked
I	09-10 00:24:02.978	596	596	com.corea.activityldemo	Life Cycle	Main.onResume() invoked

■ 화면 방향 전환

L...	Time	PID	TID	Application	Tag	Text
I	09-10 00:36:59.848	596	596	com.corea.activityldemo	Life Cycle	Main.onPause() invoked
I	09-10 00:36:59.848	596	596	com.corea.activityldemo	Life Cycle	Main.onStop() invoked
I	09-10 00:36:59.848	596	596	com.corea.activityldemo	Life Cycle	Main.onDestroy() invoked
I	09-10 00:36:59.978	596	596	com.corea.activityldemo	Life Cycle	Main.onCreate(Bundle) invoked
I	09-10 00:36:59.978	596	596	com.corea.activityldemo	Life Cycle	Main.onStart() invoked
I	09-10 00:36:59.978	596	596	com.corea.activityldemo	Life Cycle	Main.onResume() invoked

액티비티 생명주기 응용

■ 메인 액티비티 실행과 5가지 상황

● [Home] 버튼 클릭

L...	Time	PID	TID	Application	Tag	Text
I	09-10 00:37:59.049	596	596	com.corea.activitydemo	Life Cycle	Main.onPause() invoked
I	09-10 00:38:00.619	596	596	com.corea.activitydemo	Life Cycle	Main.onStop() invoked

● [Back] 버튼으로 프로젝트로 복귀

L...	Time	PID	TID	Application	Tag	Text
I	09-10 00:39:15.429	596	596	com.corea.activitydemo	Life Cycle	Main.onRestart() invoked
I	09-10 00:39:15.429	596	596	com.corea.activitydemo	Life Cycle	Main.onStart() invoked
I	09-10 00:39:15.429	596	596	com.corea.activitydemo	Life Cycle	Main.onResume() invoked

● 프로젝트 종료

L...	Time	PID	TID	Application	Tag	Text
I	09-10 00:40:21.249	596	596	com.corea.activitydemo	Life Cycle	Main.onPause() invoked
I	09-10 00:40:21.909	596	596	com.corea.activitydemo	Life Cycle	Main.onStop() invoked
I	09-10 00:40:21.909	596	596	com.corea.activitydemo	Life Cycle	Main.onDestroy() invoked

액티비티 생명주기 응용

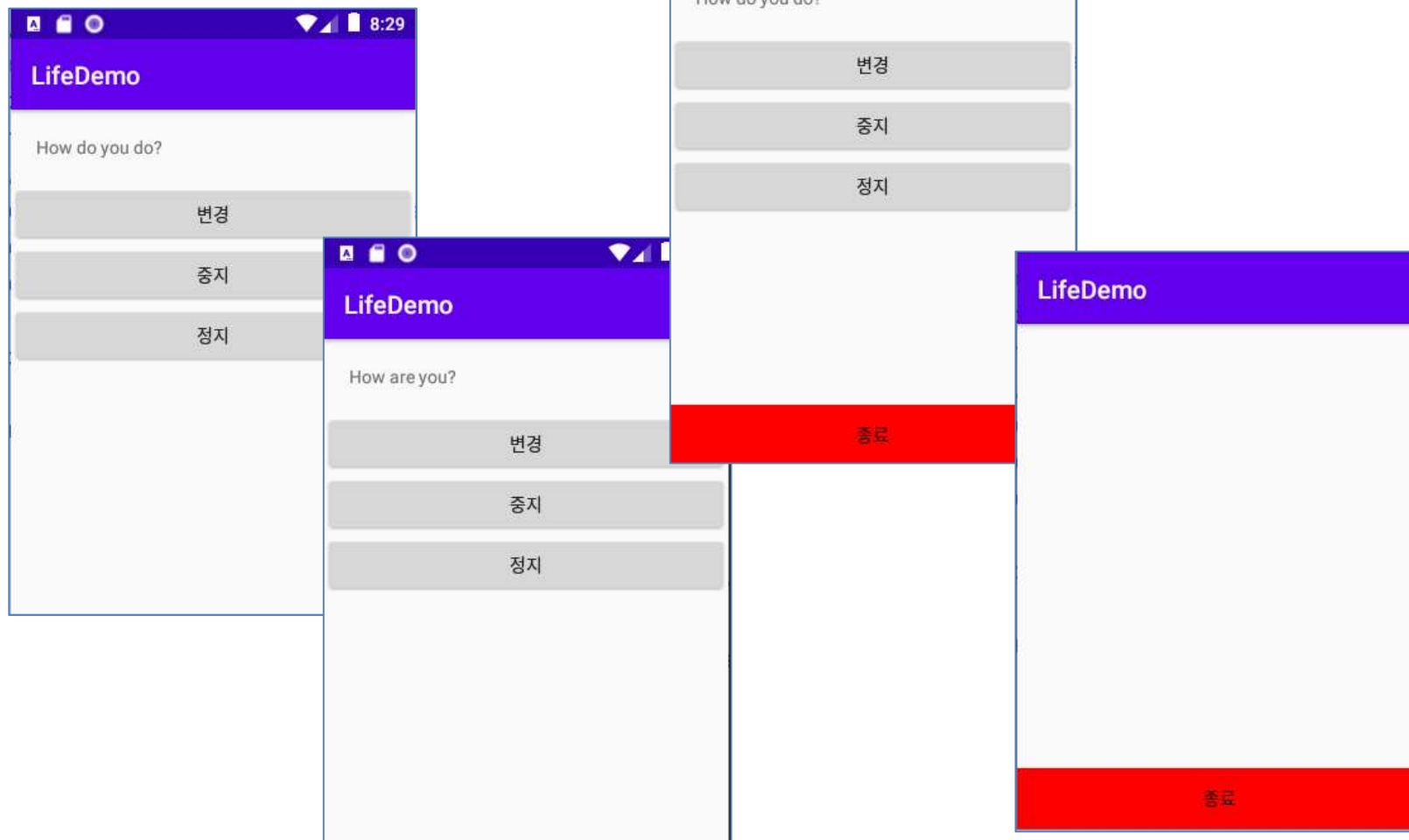
■ 다른 액티비티 호출

- 메인 액티비티가 두 가지 종류의 서브 액티비티를 호출
 - 메인 ([액티비티](#), [레이아웃](#))
 - 불투명 ([액티비티](#), [레이아웃](#))
 - 반투명 ([액티비티](#), [레이아웃](#))
 - 기타 ([스타일](#), [매니페스트](#))

액티비티 생명주기 응용

■ 다른 액티비티 호출

- LifeDemo 실행



액티비티 생명주기 응용

■ 다른 액티비티 호출 : 메인 액티비티 실행과 4가지 상황

- 불투명 액티비티(Full)로 이동

L...	Time	PID	TID	Application	Tag	Text
I	09-10 01:07:36.518	852	852	com.corea.activity2demo	Life Cycle	Main.onPause() invoked
I	09-10 01:07:36.558	852	852	com.corea.activity2demo	Life Cycle	Full.onCreate(Bundle) invoked
I	09-10 01:07:36.738	852	852	com.corea.activity2demo	Life Cycle	Full.onStart() invoked
I	09-10 01:07:36.738	852	852	com.corea.activity2demo	Life Cycle	Full.onResume() invoked
I	09-10 01:07:37.328	852	852	com.corea.activity2demo	Life Cycle	Main.onStop() invoked

- 불투명 액티비티(Full)에서 메인 액티비티로 복귀

L...	Time	PID	TID	Application	Tag	Text
I	09-10 01:08:55.958	852	852	com.corea.activity2demo	Life Cycle	Full.onPause() invoked
I	09-10 01:08:55.998	852	852	com.corea.activity2demo	Life Cycle	Main.onRestart() invoked
I	09-10 01:08:55.998	852	852	com.corea.activity2demo	Life Cycle	Main.onStart() invoked
I	09-10 01:08:55.998	852	852	com.corea.activity2demo	Life Cycle	Main.onResume() invoked
I	09-10 01:08:56.488	852	852	com.corea.activity2demo	Life Cycle	Full.onStop() invoked
I	09-10 01:08:56.488	852	852	com.corea.activity2demo	Life Cycle	Full.onDestroy() invoked

액티비티 생명주기 응용

■ 다른 액티비티 호출 : 메인 액티비티 실행과 4가지 상황

● 투명 액티비티(Semi)로 이동

L...	Time	PID	TID	Application	Tag	Text
I	09-10 01:10:24.028	852	852	com.corea.activity2demo	Life Cycle	Main.onPause() invoked
I	09-10 01:10:24.089	852	852	com.corea.activity2demo	Life Cycle	Semi.onCreate(Bundle) invoked
I	09-10 01:10:24.128	852	852	com.corea.activity2demo	Life Cycle	Semi.onStart() invoked
I	09-10 01:10:24.128	852	852	com.corea.activity2demo	Life Cycle	Semi.onResume() invoked

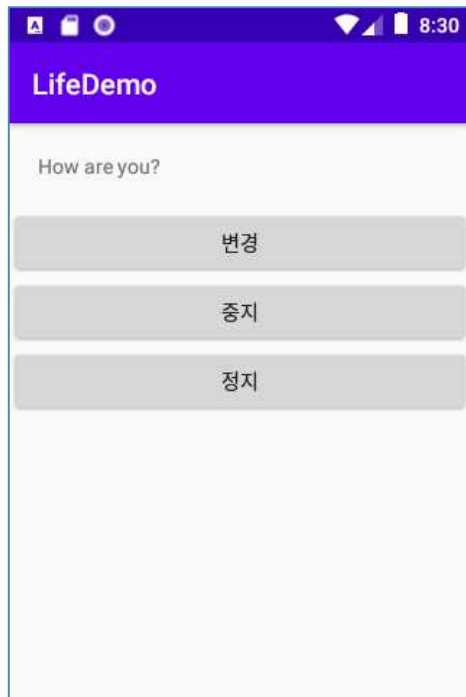
● 투명 액티비티(Semi)에서 메인 액티비티로 복귀

L...	Time	PID	TID	Application	Tag	Text
I	09-10 01:11:16.359	852	852	com.corea.activity2demo	Life Cycle	Semi.onPause() invoked
I	09-10 01:11:16.418	852	852	com.corea.activity2demo	Life Cycle	Main.onResume() invoked
I	09-10 01:11:16.478	852	852	com.corea.activity2demo	Life Cycle	Semi.onStop() invoked
I	09-10 01:11:16.478	852	852	com.corea.activity2demo	Life Cycle	Semi.onDestroy() invoked

액티비티 생명주기 응용

■ 액티비티 강제 종료와 상태 복원

- 생명주기 메서드만 사용할 경우
 - 화면 방향을 전환하거나 [Back] 버튼을 클릭한 후 다시 실행할 때마다 완전한 상태 복원 불가
- LifeDemo 실행 ([액티비티](#))



액티비티 생명주기 응용

■ 액티비티 강제 종료와 상태 복원 : 액티비티 실행과 5가지 상황

- 메인 액티비티 최초 실행과 [변경] 버튼 클릭

L...	Time	PID	TID	Application	Tag	Text
I	09-10 01:31:06.459	978	978	com.corea.activity3demo	Life Cycle	Main.onCreate(Bundle) invoked
I	09-10 01:31:06.489	978	978	com.corea.activity3demo	Life Cycle	Main.onStart() invoked
I	09-10 01:31:06.489	978	978	com.corea.activity3demo	Life Cycle	Main.onResume() invoked

- [Home] 버튼 클릭

L...	Time	PID	TID	Application	Tag	Text
I	09-10 01:32:25.029	978	978	com.corea.activity3demo	Life Cycle	Main.onPause() invoked
I	09-10 01:32:26.629	978	978	com.corea.activity3demo	Life Cycle	Main.onSaveInstanceState(Bundle) invoked
I	09-10 01:32:26.629	978	978	com.corea.activity3demo	Life Cycle	Main.onStop() invoked

- 메인 액티비티로 복귀

L...	Time	PID	TID	Application	Tag	Text
I	09-10 01:33:16.308	978	978	com.corea.activity3demo	Life Cycle	Main.onRestart() invoked
I	09-10 01:33:16.308	978	978	com.corea.activity3demo	Life Cycle	Main.onStart() invoked
I	09-10 01:33:16.308	978	978	com.corea.activity3demo	Life Cycle	Main.onResume() invoked

액티비티 생명주기 응용

■ 액티비티 강제 종료와 상태 복원 : Activity3Demo 실행과 5가지 상황

● 화면 방향 전환

L...	Time	PID	TID	Application	Tag	Text
I	09-10 01:34:06.969	978	978	com.corea.activity3demo	Life Cycle	Main.onPause() invoked
I	09-10 01:34:06.969	978	978	com.corea.activity3demo	Life Cycle	Main.onSaveInstanceState(Bundl
I	09-10 01:34:06.969	978	978	com.corea.activity3demo	Life Cycle	Main.onStop() invoked
I	09-10 01:34:06.969	978	978	com.corea.activity3demo	Life Cycle	Main.onDestroy() invoked
I	09-10 01:34:07.129	978	978	com.corea.activity3demo	Life Cycle	Main.onCreate(Bundle) invoked
I	09-10 01:34:07.129	978	978	com.corea.activity3demo	Life Cycle	Main.onStart() invoked
I	09-10 01:34:07.149	978	978	com.corea.activity3demo	Life Cycle	Main.onRestoreInstanceState(Bu
I	09-10 01:34:07.149	978	978	com.corea.activity3demo	Life Cycle	Main.onResume() invoked

● [Back] 버튼으로 메인 액티비티 종료

L...	Time	PID	TID	Application	Tag	Text
I	09-10 01:34:55.449	978	978	com.corea.activity3demo	Life Cycle	Main.onPause() invoked
I	09-10 01:34:56.698	978	978	com.corea.activity3demo	Life Cycle	Main.onStop() invoked
I	09-10 01:34:56.698	978	978	com.corea.activity3demo	Life Cycle	Main.onDestroy() invoked