



BK Mobile

Bài 5

Lập trình Android

Intent

Nội dung

- Intents và intent filters
- Vòng đời Activity
- Ứng dụng với nhiều activities
- Explicit và implicit intents
- Bài tập

Android Intents

Intent

- Một đối tượng thông điệp để kích hoạt một thành phần khác của ứng dụng
 - Giao tiếp giữa một ứng với với một hệ thống host
- Tạo điều kiện thuận lợi cho việc tương tác và hợp tác giữa các thành phần
 - Start một activity
 - Start một service (dịch vụ)
 - Phân phối thông điệp rộng rãi
 - Phân phối dữ liệu

Các loại Intent

- *Explicit intents*
 - **Nêu rõ tên** thành phần được start
 - Thường sử dụng để start một thành phần bên trong ứng dụng
- *Implicit intents*
 - **Không có tên rõ ràng** thành phần được start
 - Chỉ ra 1 hành động chung sẽ được thực hiện
 - Cho phép ứng dụng khác, bên nhận lựa chọn một 1 thành phần thích hợp để thực hiện hành động
 - Ví dụ, yêu cầu một ứng dụng khác hiển thị 1 địa điểm xác định trên bản đồ.

Intent Filter

- Khai báo khả năng hay ý định của một ứng dụng nhằm xử lý với một loại hành động cụ thể
- Một biểu thức trong file **manifest**
 - Xác định kiểu của Intent mà thành phần của ứng dụng có thể nhận hoặc có khả năng đáp ứng
- Cho phép một ứng dụng khác có thể start một activity trong ứng dụng của bạn với ***implicit intent***.
- Nếu không có intent filter nào được khai báo cho 1 activity, thì activity đó chỉ có thể start với 1 ***explicit intent***.

Start một Activity

- Một đối tượng Intent mang kèm theo một số loại thông tin
 - Khi sử dụng như là một *explicit intent*
Component name – Tên của thành phần sẽ start
Extras – Cặp **key-value** chứa thông tin
- Start một activity với một *explicit intent*
 - Trong **context** của một activity

```
Intent intent =  
    new Intent(this, TargetActivity.class);  
startActivity(intent);
```

Android Activity Lifecycle

Android Activity

- Một cửa sổ đơn, tập trung vào những thứ user có thể thực hiện
 - Tương tác với user
- Thường hiển thị như là một cửa sổ **full-screen**
- Cũng có thể hiển thị như là
 - Cửa sổ floating, hoặc
 - Nhúng trong một activity khác
- Một trong những đơn vị nhỏ nhất có thể độc lập:
created, paused, resumed, và destroyed

Vòng đời trạng thái Activity

- Một activity có thể là một trong các vòng đời trạng thái sau:
- *Active (running, resumed)* – một activity là foreground, ở đầu (top) của activity stack.
- *Paused* – một activity đã mất focus nhưng vẫn visible
 - Foreground activity là trong suốt một phần hoặc không hiển thị full-screen
- *Stopped* – một activity đã hoàn toàn bị che khuất bởi activity khác
- *Destroyed* – một activity bị kill bởi hệ thống.

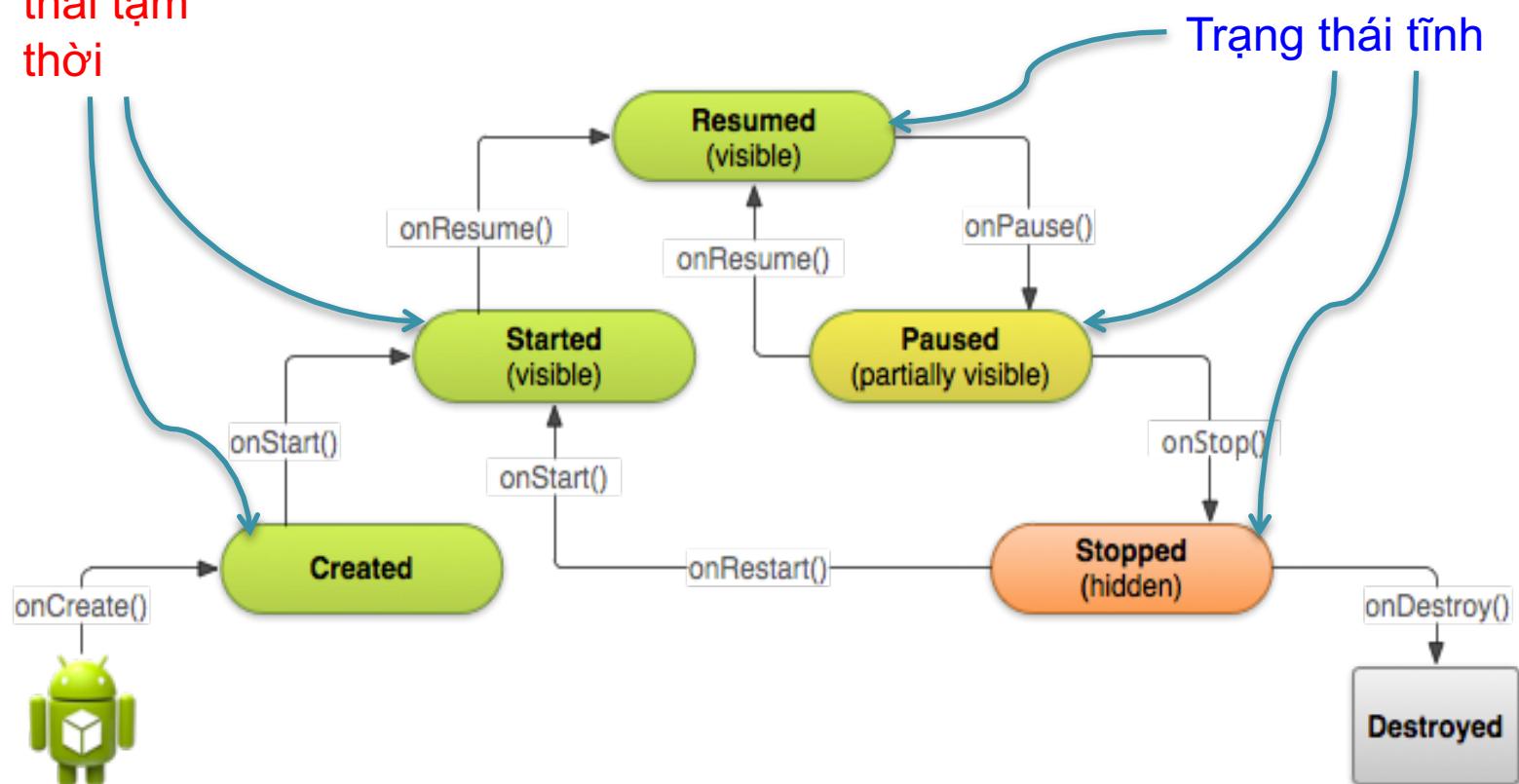
Vòng đời phương thức của Activity

- Vòng đời phương thức của Activity là gọi lại (callback) phương thức được gọi bởi hệ thống trong suốt quá trình dịch chuyển giữa vòng đời trạng thái của một activity

```
protected void onCreate(Bundle savedInstanceState);  
protected void onStart();  
protected void onRestart();  
protected void onResume();  
protected void onPause();  
protected void onStop();  
protected void onDestroy();
```

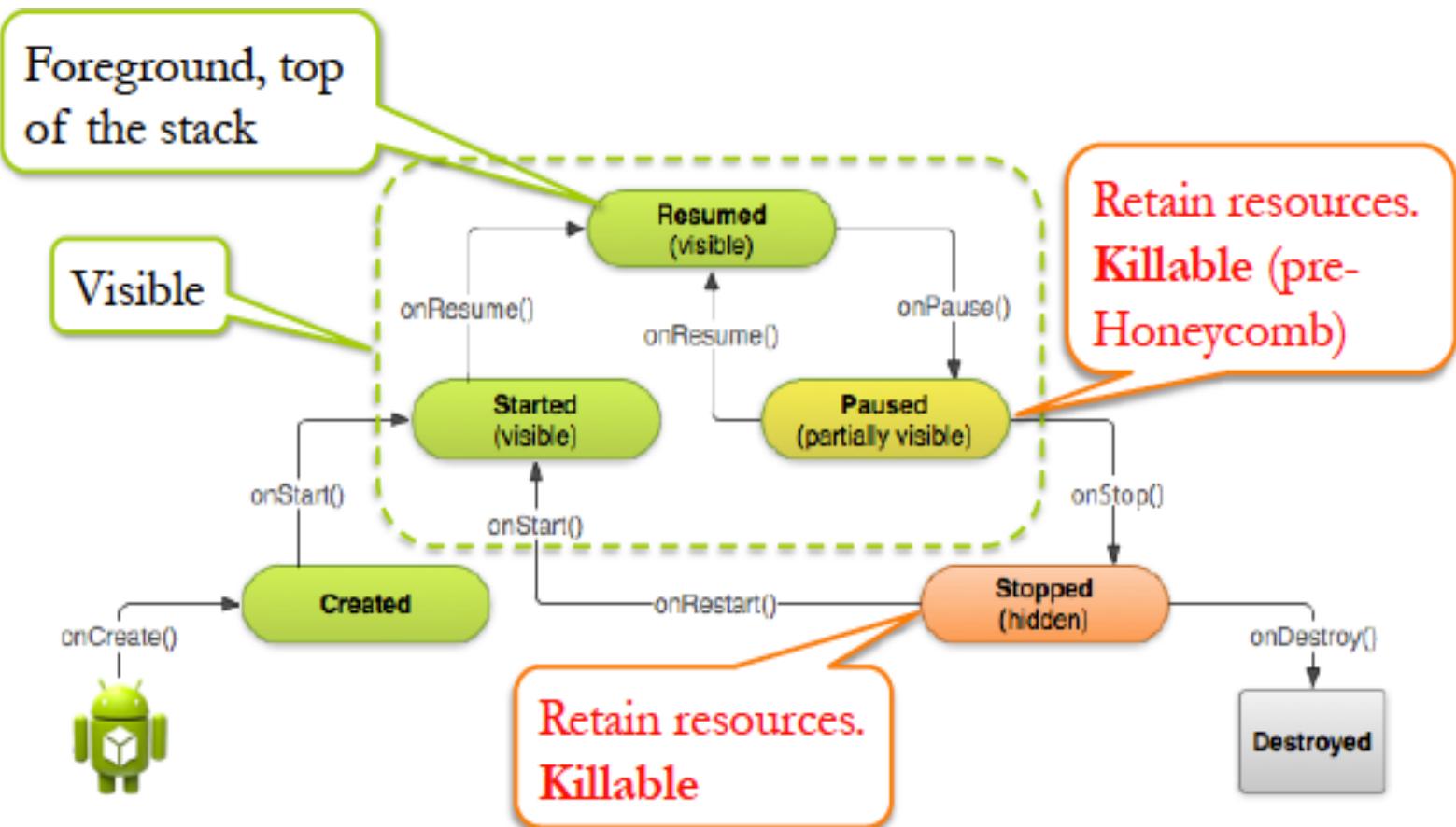
Vòng đời trạng thái (1)

Trạng
thái tạm
thời

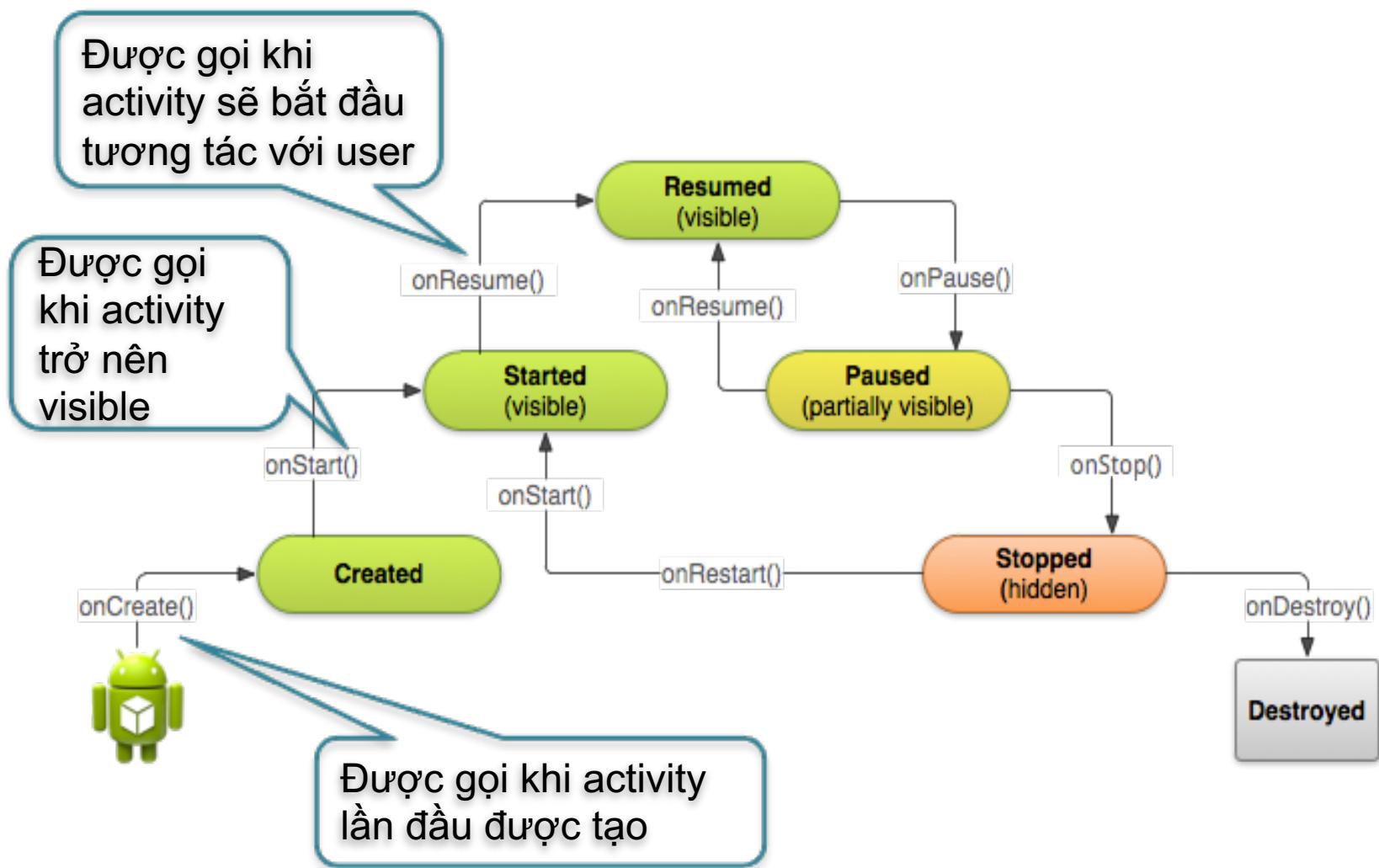


Trạng thái tĩnh

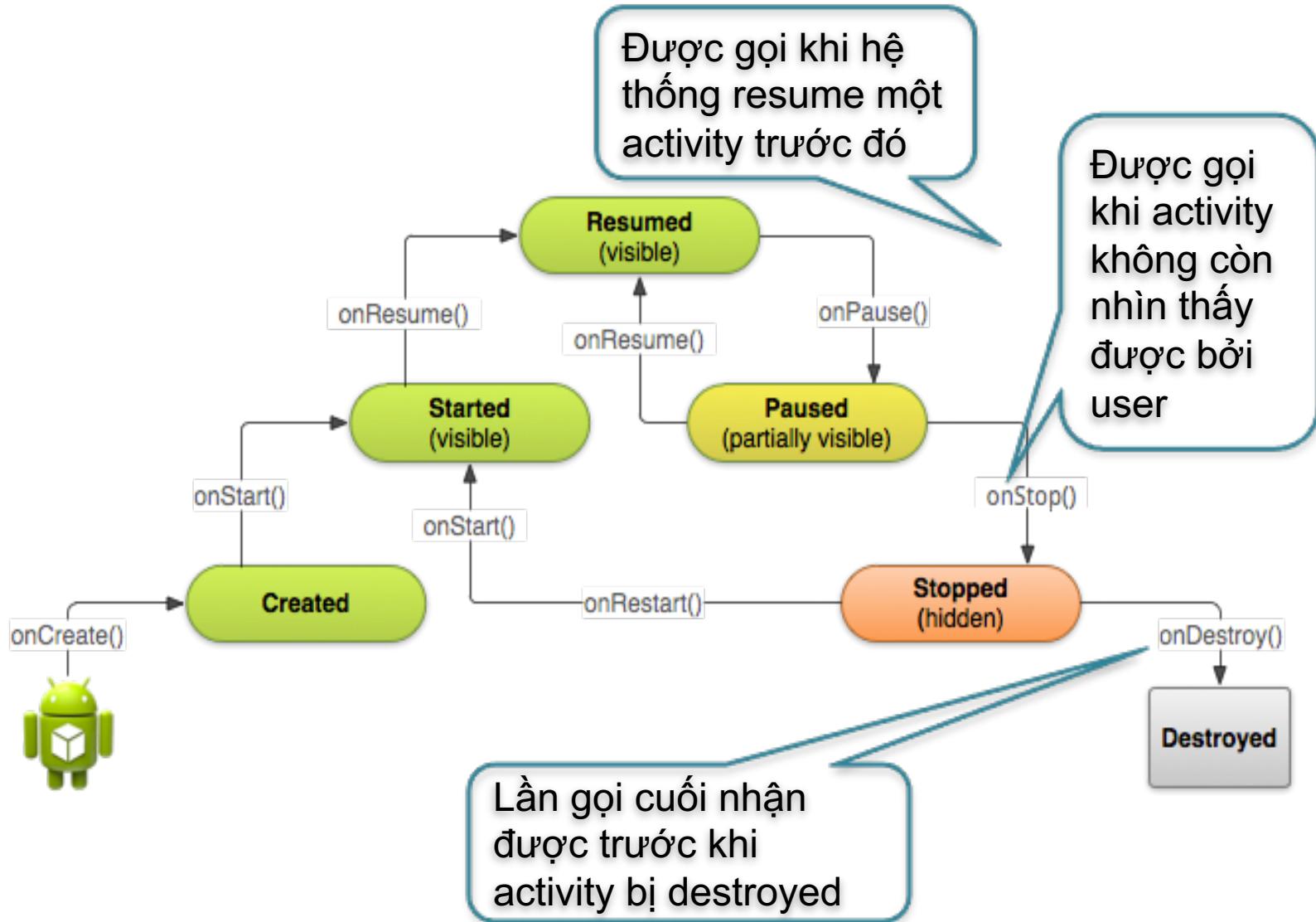
Vòng đời trạng thái (2)



Callbacks - Creation



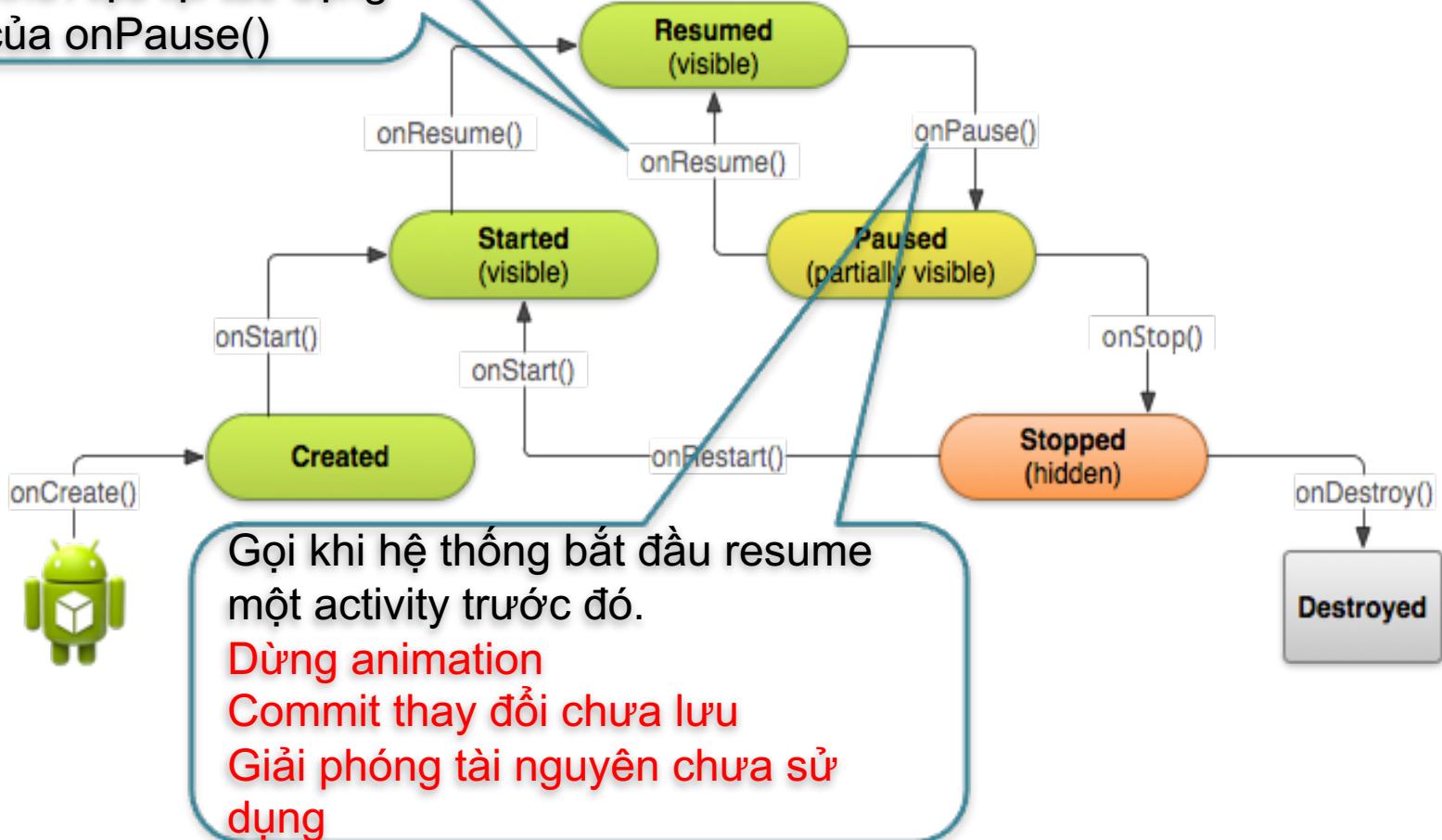
Callbacks – Đi tới Background



Callbacks – Pausing và Resumption

Gọi khi activity bắt đầu tương tác với user.

Khởi tạo lại tác dụng của onPause()



Callbacks – Stopping và Restarting

Được gọi khi activity bắt đầu tương tác với user

Được gọi khi activity trở bên visible.
Khởi tạo lại



onCreate()

Created

onStart()

Started
(visible)

onStart()

Resumed
(visible)

onResume()

Paused
(partially visible)

onPause()

onStop()

Stopped
(hidden)

Được gọi sau khi activity đã dừng, trước khi nó bắt đầu trở lại

Được gọi khi activity không nhìn thấy được nữa bởi user.

Giải phóng tài nguyên chưa dùng

onDestroy()

Destroyed

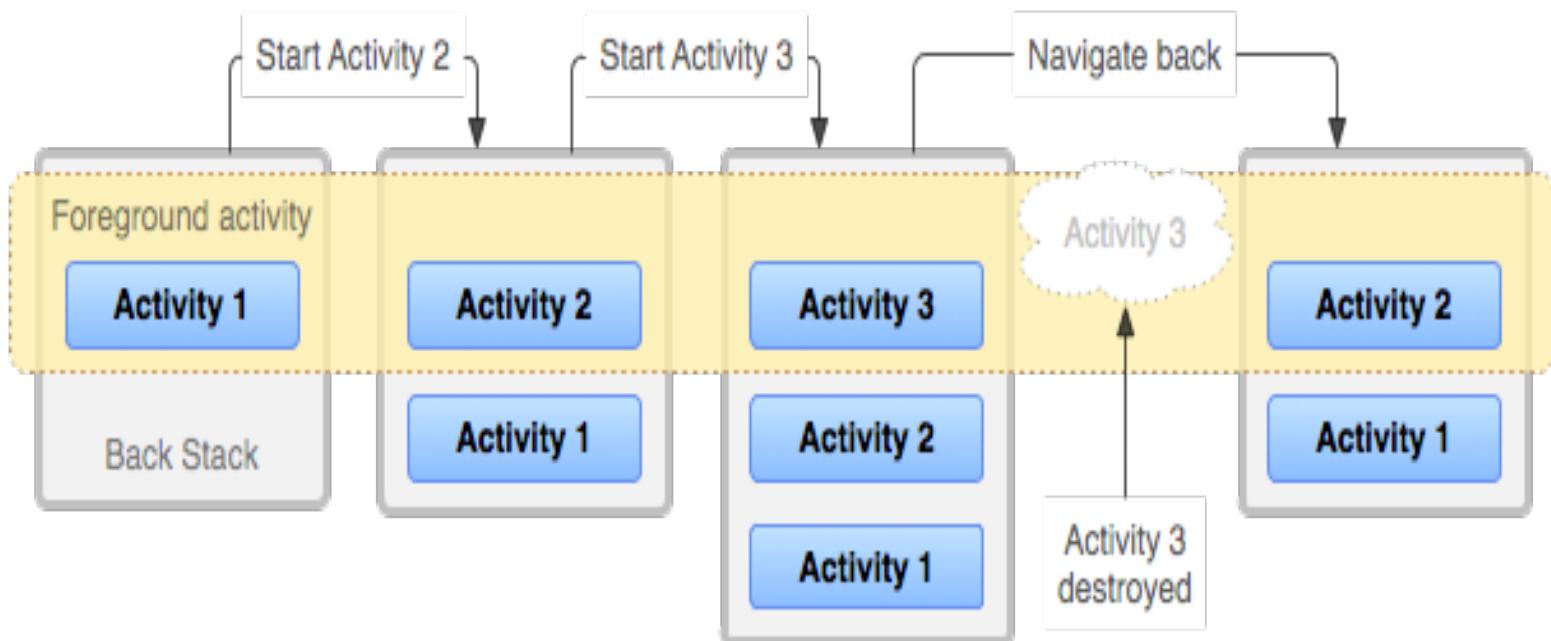
Callbacks – Tạo lại một Activity



Tasks và Back Stack

- Một ứng dụng thường chứa nhiều activities.
 - Thậm chí start activities trong ứng dụng khác
- Một *task* là một tập các activities mà users tương tác khi thực hiện một công việc xác định.
- Các activities được sắp xếp trong một stack (“*back stack*”), theo thứ tự mỗi activity được mở.
 - *LIFO* (“last in, first out”)
 - **Push**: chạy 1 ứng dụng từ màn hình Home, hoặc bắt đầu một activity từ activity khác
 - **Pop**: nhấn nút back, activity hiện tại sẽ được đưa ra khỏi stack

Back Stack



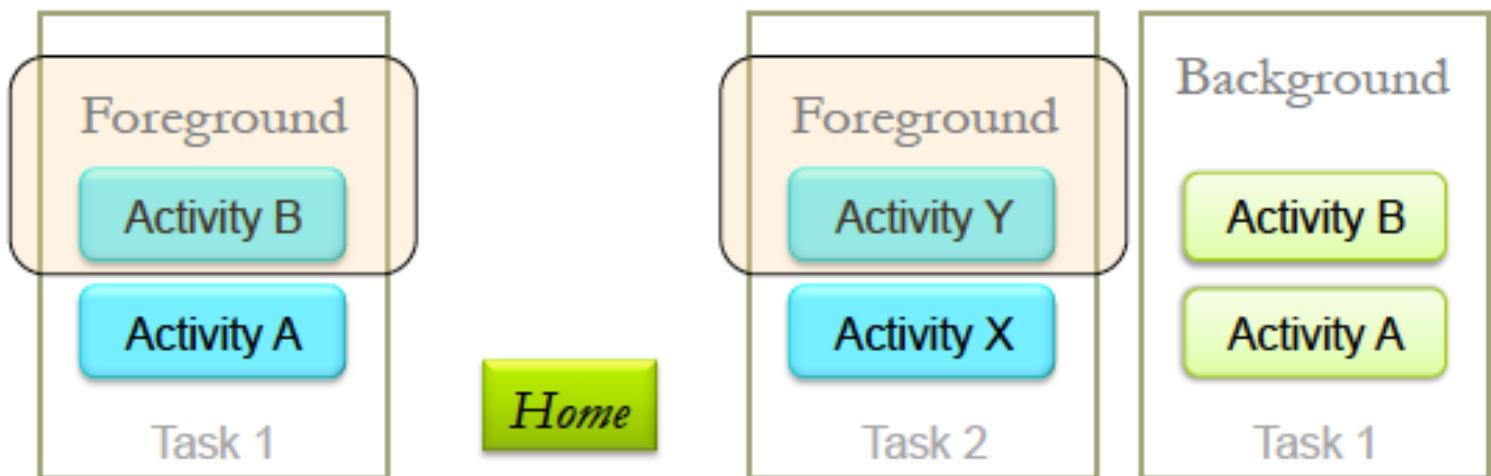
Activities và Tasks

- **Activity A** khởi động **Activity B**
 - Activity A dừng (*stopped*), trạng thái của nó được lưu lại.
 - Activity B được đẩy vào trong đỉnh stack.
- Nhấn nút *Back*
 - **Activity B** được lấy ra khỏi stack và bị *destroyed*, trạng thái của nó không được lưu lại.
 - **Activity A** khôi phục với trạng thái đã được lưu trước đó.

Background Tasks (1)

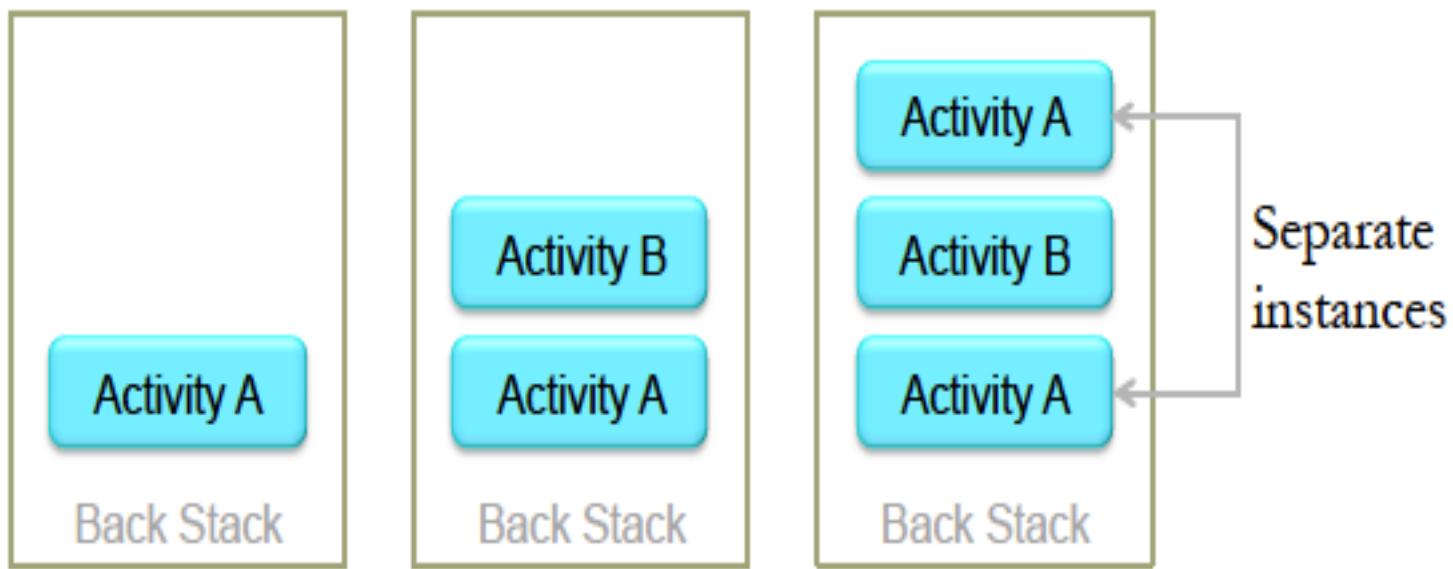
- Nhấn nút *Home*, rời khỏi 1 task
 - Activity hiện tại bị dừng (*stopped*)
 - Task hiện tại đi tới *background*
 - Trạng thái của mỗi activity trong task được lưu lại
- Task trên có thể được khôi phục (*resumed*) thông qua
 - Chạy icon trong màn hình *Home*, hoặc
 - *Recent Tasks*

Background Tasks (2)



Đa thể hiện của Activities

- Activities có thể được **khởi tạo nhiều lần**
 - Nhiều thể hiện của cùng một activity có thể cùng tồn tại trong back stack





Ứng dụng với nhiều activities

Intent Demo App

- Một ứng dụng có 2 màn hình, tức là 2 activities
- **Activity A** có thể
 - Start *Activity B* với **explicit** intent
 - Finish chính nó
- **Activity B** có thể
 - Start *Activity A* với **explicit** intent
 - Finish chính nó



Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://...
    package="cse.intentdemo">
    <application ... >
        <activity android:name=".ActivityA"
            android:label="@string/title_a">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity android:name=".ActivityB"
            android:label="@string/title_b">
        </activity>
    </application>
</manifest>
```

Tên activity nên
được đặt gợi nhắc
đầy đủ.

‘.’ chỉ package hiện
tại.

Layout Screens

- Layout file: **activity_a.xml**

```
<LinearLayout ...>
    <TextView ...
        android:id="@+id/title"
        android:text="Activity A"/>
    <Button ...
        android:id="@+id/button1"
        android:text="Start Activity B"/>
    <Button ...
        android:id="@+id/button2"
        android:text="Finish"/>
</LinearLayout>
```



- Layout file: **activity_b.xml**, gần tương tự

onCreate và onStart

```
public class ActivityA extends Activity {  
    private static int INSTANCE_COUNTER = 0; # of instances  
    private int instanceID; # of starts  
    private int counter = 0; # of starts  
    protected void onCreate(Bundle savedInstanceState) {  
        instanceID = ++INSTANCE_COUNTER;  
        Log.d(TAG, "onCreate() instanceID=" + instanceID);  
        ...  
    }  
    protected void onStart() {  
        super.onStart();  
        Log.d(TAG, "onStart() counter=" + ++counter);  
    }  
    ...
```

Start một activity mới

- ActivityA.java

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_a);  
    Button button1 = (Button) findViewById(R.id.button1);  
    button1.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            Intent intent =  
                new Intent(ActivityA.this, ActivityB.class);  
            startActivity(intent);  
        }  
    });  
    ...  
}
```

Finish activity hiện tại

- ActivityA.java

```
protected void onCreate(Bundle savedInstanceState) {  
    ...  
    Button button2 = (Button) findViewById(R.id.button2);  
    button2.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            finish();  
        }  
    });  
}
```

Finish activity hiện tại.
Activity sẽ được lấy ra khỏi
stack và bị destroyed.
Not a callback.

Activity onStart()

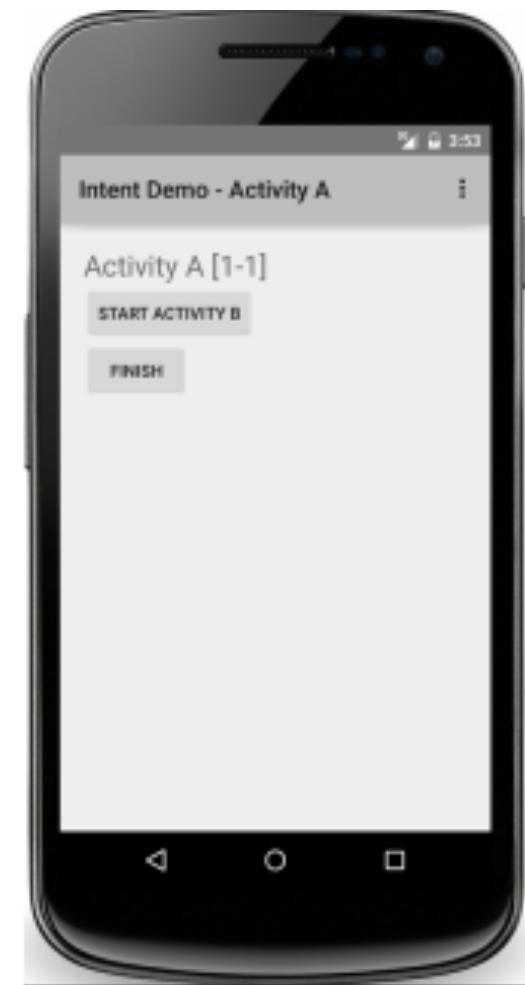
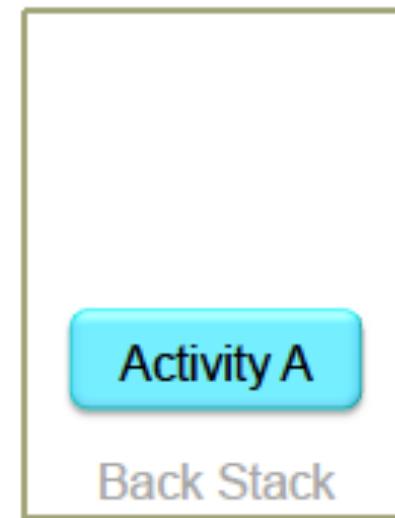
- ActivityA.java

Sử dụng log để hiểu
cụ thể về vòng đời
Android.

```
@Override  
protected void onStart() {  
    super.onStart();  
    Log.d(TAG, "onStart() counter=" + ++counter);  
    TextView title = (TextView) findViewById(R.id.title);  
    title.setText("Activity A [" + instanceID +  
        "-" + counter + "]");  
}
```

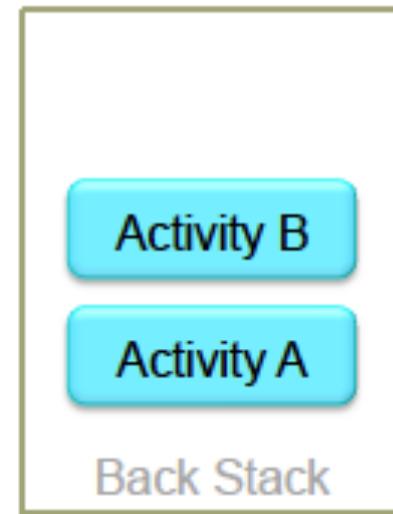
Intent Demo – Step 1

- Chạy demo app
 - *ActivityA*



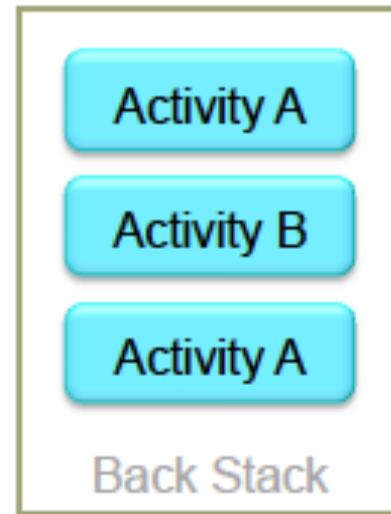
Intent Demo – Step 2

- Chạy demo app
 - *ActivityA*
- Start Activity
 - *ActivityB*



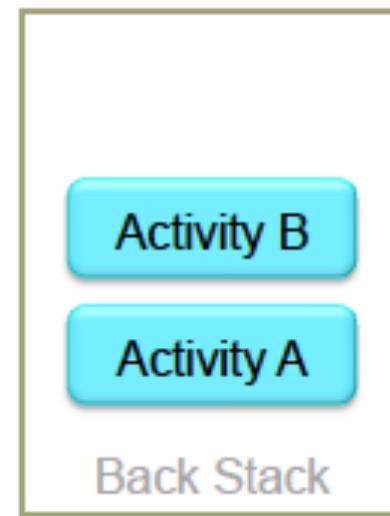
Intent Demo – Step 3

- Chạy demo app
 - *ActivityA*
- Start Activity
 - *ActivityB*
- Start Activity
 - Một thẻ hiện mới của *ActivityA*



Intent Demo – Step 4

- Chạy demo app
 - *ActivityA*
- Start Activity
 - *ActivityB*
- Start Activity
 - Một thẻ hiện mới của *ActivityA*
- Finish
 - Pop *ActivityA*



Intent Demo – Step 5

- Chạy demo app
 - *ActivityA*
- Start Activity
 - *ActivityB*
- Start Activity
 - Một thẻ hiện mới của *ActivityA*
- Finish
 - Pop *ActivityA*
- Finish
 - Pop *ActivityB*

