IT332: Mobile Application Development

Lecture # 05: More on Intents

Muhammad Imran

Outline

- Activity state and ejection from memory
- Tasks and Back Stack
- Parcelables and Bundles with Intents

Activity State and Ejection from Memory

- The android system kills processes when it needs to free up RAM; the likelihood of the system killing a given process depends on the process state at the time.
- Process state, in turn, depends on the activity state running in the process.

Likelihood of being killed	Process state	Activity state
Least	Foreground (having or about to get focus)	Created Started Resumed
More	Background (lost focus)	Paused
Most	Background (not visible)	Stopped
	Empty	Destroyed

Table 1. Relationship between process lifecycle and activity state

Tasks and Back Stack

- A task is a collection of activities that users interact with when performing a certain job.
- The activities are arranged in a stack—the back stack)—in the order in which each activity is opened.
- For example,
  - an email app might have one activity to show a list of new messages. When the user selects a message, a new activity opens to view that message. This new activity is added to the back stack.
  - If the user presses the Back button, that new activity is finished and popped off the stack.

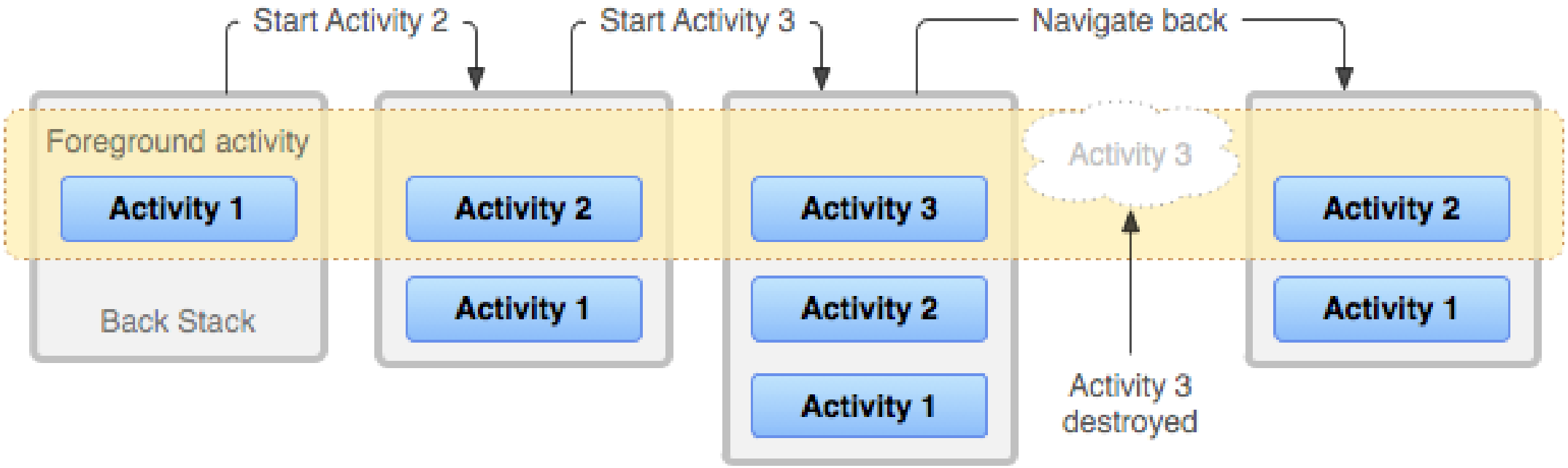
Tasks and Back Stack

- When the user touches an icon in the app launcher (or a shortcut on the Home screen), that app's task comes to the foreground
- If no task exists for the app (the app has not been used recently), then a new task is created and the "main" activity for that app opens as the root activity in the stack.
- When the current activity starts another, the new activity is pushed on the top of the stack and takes focus.
- The previous activity remains in the stack but is stopped.
- When an activity stops, the system retains the current state of its user interface.

Tasks and Back Stack

- When the user presses the Back button, the current activity is popped from the top of the stack (the activity is destroyed) and the previous activity resumes (the previous state of its UI is restored).
- Activities in the stack are never rearranged, only pushed and popped from the stack
- The back stack operates as a "last in, first out" object structure.

Tasks and Back Stack



Parcelables and Bundles with Intents

- Parcelable and Bundle objects are intended to be used across process boundaries between activities with intents, and to store transient state across configuration changes.
- We can use them as a mechanism to send composite or complex objects across activities
- The custom class should implement Parcelable and provide the definition of writeToParcel(android.os.Parcel, int) method
- It must also provide a non-null field called CREATOR that implements the Parcelable.Creator interface
- When sending data via an intent, you should be careful to limit the data size to a few KB.
- Sending too much data can cause the system to throw a TransactionTooLargeException exception.

Parcelables and Bundles with Intents

In MainActivity (Send the Parcelable Object)

```
public void sendPerson(View view) {
    Person p = new Person( mProfileImage: 1, strName: "Imran", strAddress: "IIUI");
    Intent intent = new Intent( packageContext: this,ActivityReceiveObject.class);
    intent.putExtra( name: "person1",p);

    startActivity(intent);
}
```

In SecondActivity (Get the Parcelable Object)

```
private void ReceiveObjectFromIntent() {
    Intent intent = getIntent();
    Person p = intent.getParcelableExtra( name: "person1");

    String name = p.getStrName();
    String address = p.getStrAddress();
    int profileImageID = p.getmProfileImage();
}
```

Parcelables and Bundles with Intents

- Read more on using Parcelables and How we can Pass Custom Objects to an activity

More on Intents and Intent Filters

- Intents and Intent Filters
- https://developer.android.com/guide/components/intents-filters

- Common Intents
- https://developer.android.com/guide/components/intents-common

- Documentation Reference
- https://developer.android.com/reference/android/content/Intent

Processes and Application Lifecycle

- https://developer.android.com/guide/components/activities/process-lifecycle

Fragments

- https://developer.android.com/guide/components/fragments

- Then Fragments from the Wrox Book page 99 and Wrox Teacher Slides

Look the Google Course

- Android Developer Fundamentals (Version 2) - In Chapters Format
- https://google-developer-training.github.io/android-developer-fundamentals-course-concepts-v2/

- Android Developer Fundamentals (Version 2) - SLIDES FORMAT
- https://drive.google.com/drive/folders/1eu-LXxiHocSktGYpG04PfE9Xmr_pBY5P

- Android Developer Fundamentals (Version 2) - Code LABS Format
- https://developer.android.com/courses/fundamentals-training/toc-v2

- MAIN COURSES PAGE
- https://developer.android.com/courses/fundamentals-training/toc-v2

Recommended Readings

- Page # 47 to 53, Chapter # 03: Activities, Fragments, and Intents from Beginning Android Programming with Android Studio, 4th Edition by J. F. DiMarzio, Wrox, 2017
- Page # 61 to 75, Chapter # 03: Activities, Fragments, and Intents from Beginning Android Programming with Android Studio, 4th Edition by J. F. DiMarzio, Wrox, 2017

- User Guide: developer.android.com/studio/intro
- Android Vocabulary Glossary https://developers.google.com/android/for-all/vocab-words