Android Process Management

Rajesh prodduturi (113050076)

Under the Guidance of
Prof. D.B.phatak

Department of Computer Science and Engineering,
Indian Institute Of Technology , Bombay

August 24, 2012

Outline

This presentation gives an overview of the following:

- Application Fundamentals in Android
  - Activity
  - services
  - Broadcast Drivers
  - Content Providers
- Process Life Cycle in Android
- Problem Statement

Application Fundamentals in Android

- Andriod is multi-user OS, each applicaiton has it's own user id(because of security reasons)
- Each applicaiton has it's own VM(virtual machine), to provide isolation from other process.
- Each applicaiton has it's own process.
- Application is a combination of components.
  - Activity
  - services
  - Broadcast Drivers
  - Content Providers

Application Fundamentals in Android

Application Components:

- Activity
  - Acivity is a single visual user interface(UI) on the screen.
  - Eg: List of applicaitons in home menu
- Services
  - Services is a component which runs on background, but it is visible user.
  - Eg: playing music
  - Eg: Downloading a file from internet

Application Fundamentals in Android

Application Components:

- Content Providers
  - It provides a shared data for different applicaitons.
  - Eg: contact information, calender information
- Broadcast Reciever
  - It sends broadcast anouncements system-wide
  - Eg: screen-off
  - Eg: Battery low

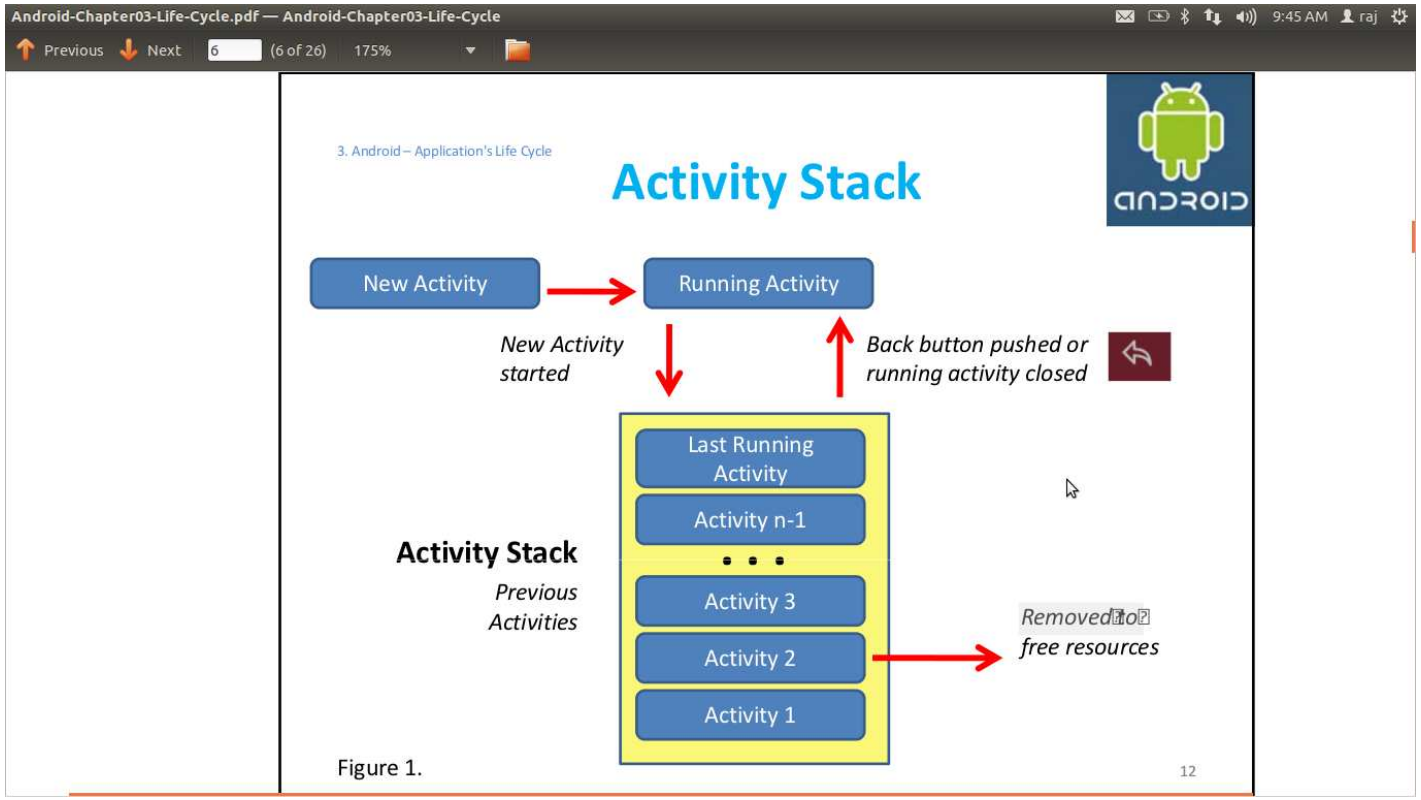
Each applicaiton contains a manifest file, it holds list of components available in an applicaitons.

Application Fundamentals in Android

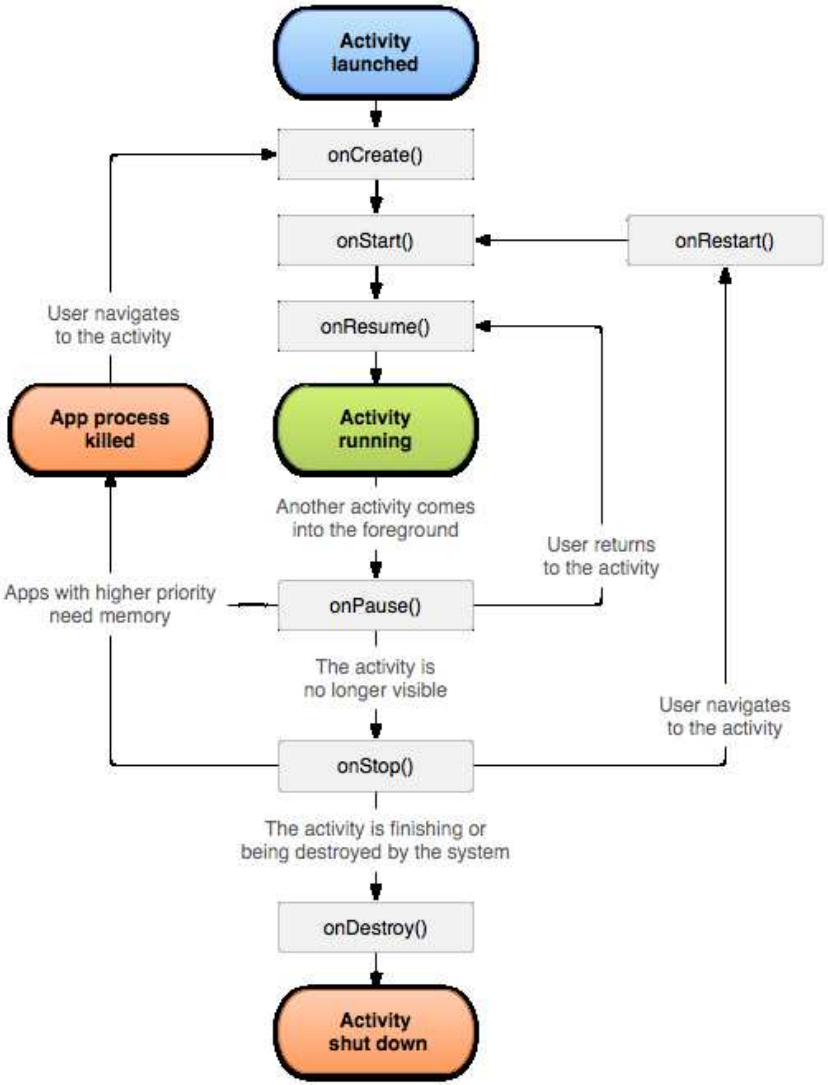
Acivity Stack:

- All the acivities in a system are placed in stack.
- Whenever a new activity starts, i.e placed on top of stack.
- whenever user press back button, activity on the top of stack would be removed.

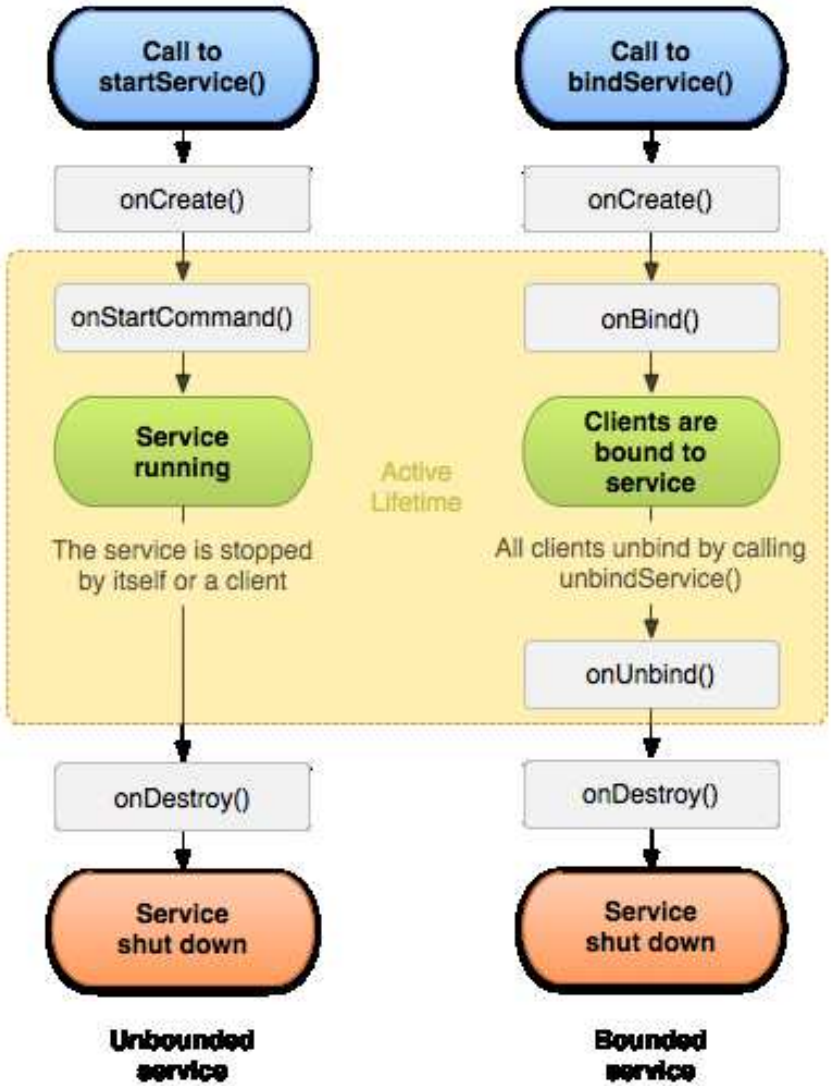
Acivity Stack:



Acivity LifeCycle:



Service LifeCycle:



Process lifecycle in Android

Android Process:

- For each applicaiton android creates a new process with single thread of execution(i.e main thread)
- By deafalut all the components run in same thread(main thread). If not developer explicitly creates a new thread.
- In low memory situations a process is decide to kill based upon the number of inactive components of that process.
- A process which have more number of inactive components should be killed first.

Process lifecycle in Android

Foreground Process:

- A process is treated as foreground if any of components of process holds following states.
- An acivitiy which is interacting with user. When an acivity calls methods like onstart(), onresume().
- When a service is interacting with foreground acitvity.
- When a service is started using startForeground().
- when a service is executing any of the methods like onCreate(), onStart(), onDestroy().

Process lifecycle in Android

Visible Process:

- A process is treated as visible if any of components of process holds following states.
- An acivitiy which is not interacting with user. But it is still visible to user, when an acivity running method like onPause().
- When a service is interacting with visible acitvity.

Service Process:

- A process is treated as Service if any of components of process holds following states.
- When a service is started using startService().
- Eg: playing music in background, downloading file from internet.

Process lifecycle in Android

Background Process:

- A process is treated as background if any of components of process holds following states.
- When an acivity running method like onStop(). i.e currently user is not interacting with that activity.
- System maintains LRU list of background process.
- Whenever a process decided to kill, it stores the state of activity. So whenever next user wants that activity, we can restore the previous state of activity.

Empty Process:

- A process is treated as Service if any of components of process holds following states.
- When a process does not have any active component.
- Caching of empty process inside memory, reduces relaunching of applicaiton once again if user wants that applicaiton in future.

Problem Statement:

Objective:
Tunning minfree parameters of lowmemorykiller and dynamically changing oom_adj values of process.

- Effective memory usage
- Improve speedness of system(i.e better performance)
- Low power consumption

I am planing to implement a tool to support above features. It contains different modes. Based upon the user interest he can choose any of the mode.

Mode 01 And 02:

Default Mode:
Based upon hardware configurations choose best set of minfree thresholds.
The following factors will be considered.

- Size of RAM
- Footprint of OS

By using benchmarks we can decide minfree thresholds, which give better performance.
User Specified Mode:
Provide different levels of minfree thresholds.

- Large(aggressive)
- Medium
- small

User can choose one of the above levels based upon his/her interest.

Mode 03,04 And 05:

partially User Interactive Mode:
In low memory situations, Whenever we need to kill a foreground, visible and service process. This tool generates a notification message to user to select which applicaiton he wants to kill.
Without User Interaction Mode:
Estimate the users current interesting applicaiton, based upon his previous log history. Do not kill frequently used applications.
Dynamic Mode:
Dynamically choosing minfree thresholds level, based upon currently running applicaitons. Examples are given bellow

- Database app, Browser - small
- media player -medium
- contacts,calender -large

REFERENCES I

- http://developer.android.com/guide/components/processes-and-threads.html

Thank You