



Android Programing

Gavrilut Dragos

Administrative

Final grade for the Python course is computed using Gauss over the total points accumulated.

One can accumulate a maximum of 100 of points:

- Maximum 60 points from the laboratory examination
- Maximum 40 points at the final examination (course)

The laboratory examination consists in 3 test:

- 30 points → lab activity over the first 7 weeks
- 30 points → lab project

The minimum number of points that one needs to pass this exam:

- Minimum 30 points accumulated from the laboratory examination
- Minimum 15 points from the final examination (course)
- Course page:
<https://sites.google.com/site/fiiandroidprogramming/home>



History

- 2003 October – Android Inc, is founded
- 2005 August – Google Inc. buys Android Inc for 50.000.000 USD
- 2007 – Android becomes an open-source project under Apache license
- 2008 October – first device with Android OS is release

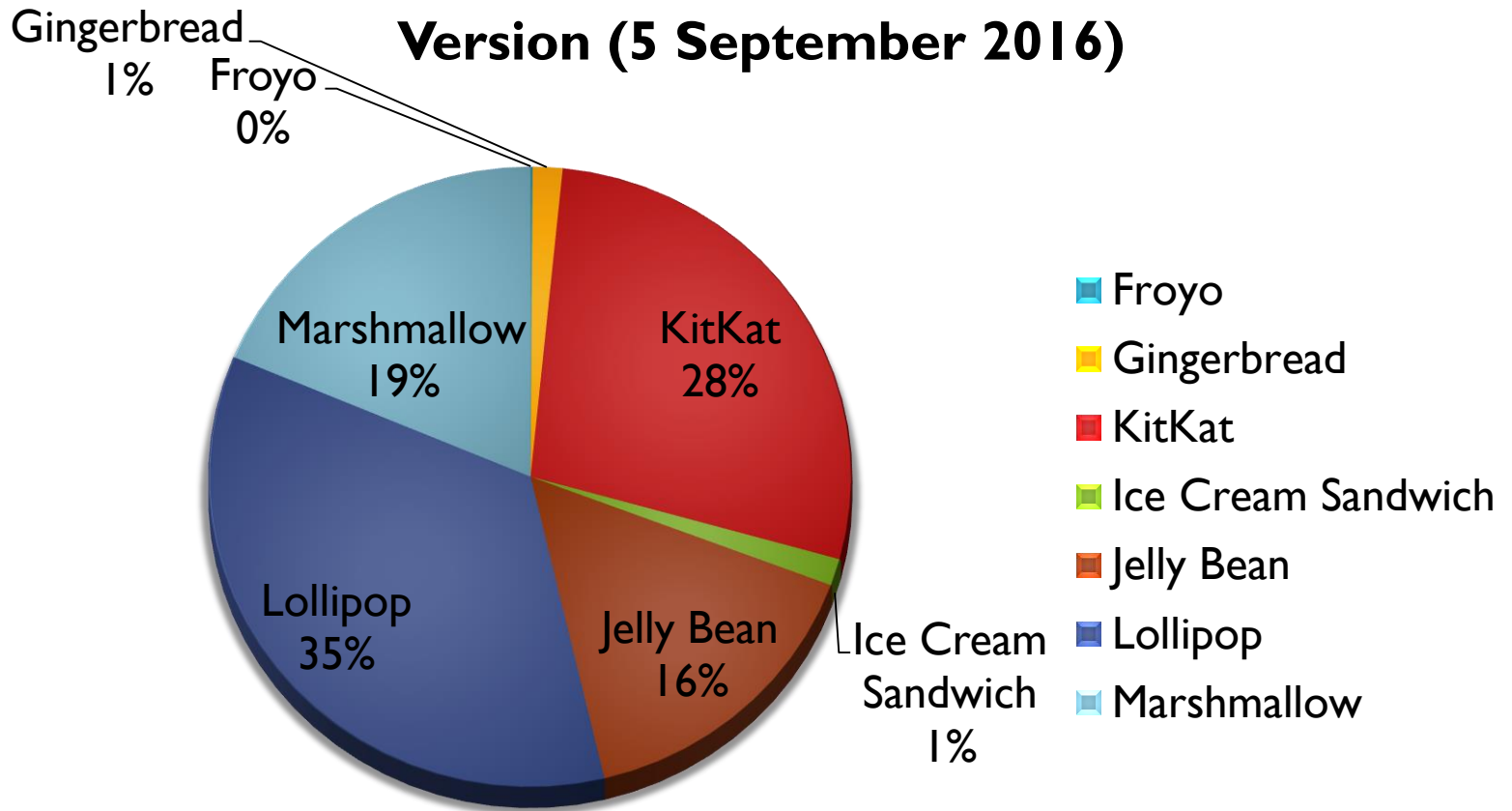
Android versions

Release date	Version	Code Name	API Level
2008.Sep	1.0	-	1
2009.Feb	1.3	Petite Four	2
2009.Apr	1.5	Cupcake	3
2009.Sep	1.6	Donut	4
2009.Oct	2.0	Eclair	5
2009.Dec	2.0.1	Eclair	6
2010.Ian	2.1	Eclair	7
2010.May	2.2	Froyo	8
2010.Dec	2.3 – 2.3.2	Gingerbread	9
2011.Feb	2.3.3 – 2.3.7	Gingerbread	10
2011.Feb	3.0	Honeycomb	11
2011.May	3.1	Honeycomb	12

Android versions

Release date	Version	Code Name	API Level
2011.July	3.2	Honeycomb	13
2011.Oct	4.0-4.0.2	Ice Cream Sandwich	14
2011.Dec	4.0.3 – 4.0.4	Ice Cream Sandwich	15
2012.Jul	4.1	Jelly Bean	16
2012.Nov	4.2	Jelly Bean	17
2013.Jul	4.3	Jelly Bean	18
2013.Sep	4.4	Kit Kat	19
2014.July	4.4W	Kit Kat	20
2014.Nov	5.0	Lollipop	21
2015.Mar	5.1	Lollipop	22
2015.Oct	6.0	Marshmallow	23
2016.Aug	7.0	Nougat	24

Android Distribution



<http://developer.android.com/about/dashboards/index.html>

Android architecture

Applications

Contacts, Browser, Phone, Games, ...

Application Framework

Activity Manager, Views, Telephony Manager, Package Manager, Resource Manager,

Libraries

Surface manager, Media, SQL, OpenGL, WebKit,

Android
Runtime
Dalvik VM

Linux Kernel

Display drivers, Camera drivers, WiFi drivers, Audio Drivers, Power management drivers, ...



Android architecture

- Hardware platform
 - ARM
 - X86 (Google TV, ...)
 - i.MX
 - Intel



Mobile architecture

- Touch Screens
- SMS & Phone
- GPS
- Flash Drives
- NFC
- WiFi
- DLNA
-



Legal issues

- Google vs Oracle
 - Java structure
- Apple vs Samsung
 - Design
- Apple & Microsoft vs HTC & Samsung
- Google buys Motorola

File Hierarchy

- Different file systems (YAFFS, EXT_x, proprietary (Samsung RFS), F2FS, F2FS, JFFS2, ...)
- Partitions:
 - /cache
 - /system
 - /sdcard
 - /mnt
 - /sys
 - /data
 - /root
 - /dev
 - ...

File Hierarchy

- /cache
 - Cached files and data
- /sdcard
 - Contains application data, pictures, etc
 - Some applications use this partition as a way to record different data regarding their installation (install date, etc)
- /system
 - Includes Android OS files (except for kernel) such as libraries, fonts, default applications (email, browser, phone, ...), system sounds, linux executables for different commands (ls, rm, su, ...), ...



File Hierarchy

- /data
 - User specific data (contacts, messages, settings, ...)
 - Private data and libraries for every application installed (by package)
 - Installed applications

APK format

- APK = **A**pplication **P**ackage **F**ile
- ZIP archive
 - **\META-INF**
 - MANIFEST.MF
 - CERT.RSA
 - CERT.SF (SHA-1 digest for MANIFEST.MF)
 - **\lib**
 - \armeabi
 - \armeabi-v7a
 - \x86
 - \mips
 - **\res**
 - drawable-{xxx}
 - raw-{xxx}
 - layout —{xxx}
 - menu
 - **\assets**
 - classes.dex
 - resources.arsc
 - AndroidManifest.xml



Dalvik

- Register-based VM
- Sandbox
- A **DEX** file contains all the information required for the Dalvik VM to execute the code (libraries, endianness, ...)
- Can run native code.
- “Every Android application runs in its own process, with its own instance of the Dalvik virtual machine.”



DEX files

- Header (Checksum, SHA, ...)
- String Indexes
- Type Indexes
- Prototype Indexes
- Field Indexes
- Method Indexes
- Class Definitions
- Data (dex code , strings, classes, ...)

Zygote process

- Use to increase the start time of a dalvik VM process
- Shares constant data (libraries) between instances of VM processes

```
def ZygoteStart:  
    while (true)  
        if (new app is requested)  
            fork()  
        endif  
    endwhile  
enddef
```

- Uses “copy-on-write” to copy modified memory to a spawn child

Applications

- Each Android Application runs as a linux process
- Each Android Application has multiple components:
 - Activities
 - Services
 - Content Providers
 - Broadcast Receivers
- Each Android Application can start another Android Application components (use an activity from email application to send email). This can be done using **Intent** object.

Applications

- Each Android Application has its own process. Each process has a rank (importance) in Android. The more important a process is, the less is the chance that it will be killed by the system.
- There are 5 ranks for processes:
 1. **Foreground process**
 - Has an Activity that users interact with
 - Has a Service that interacts with a Foreground process
 - Has a Service that runs in foreground
 - Has an active Broadcast Receiver
 2. **Visible process**
 - Has an Activity that is in background (paused)
 - Has a Service that is linked to an background activity
 3. **Service process**
 - Has a Service
 4. **Background process**
 - Has an Activity with a process that was stopped
 5. **Empty process**
 - Does not have any components. It is maintain for caching purposes.

Applications

- To execute an Android Application the system checks **AndroidManifest.xml** file
- AndroidManifest file contains:
 - Permissions
 - List of activities
 - List of services
 - List of receivers
 - List of providers

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
    <application android:icon="@drawable/app_icon.png">
        <activity android:name="com.myapp.myActivity"
            android:label="@string/HelloWorld">
            </activity>
        ...
    </application>
</manifest>
```

Applications

```
<application android:allowTaskReparenting=["true" | "false"]  
    android:description="string resource"  
    android:hasCode=["true" | "false"]  
    android:hardwareAccelerated=["true" | "false"]  
    android:icon="drawable resource"  
    android:label="string resource"  
    android:name="string"  
    android:permission="string"  
    android:process="string"  
    android:theme="resource or theme"  
  
    . . .  
</application>
```

Activity Life Cycle

