G54MDP
Mobile Device Programming

Lecture 3 – Mobile Phone Architecture / Android Internals

ARM CPU

- 32-bit CPU (increasingly 64-bit)
  - Multiple instruction sets
  - Older phones made use of Jazelle DBX / embedded systems
    - Run Java bytecode directly, dedicated chips
- Sold as a design, not a physical device
  - Great for SoC usage
  - Chip manufactures can add ARM CPU to bespoke SoC vs buying a chip from Intel
- Aims
  - Fast and efficient
    - Good for mobile applications
    - Best use of battery – more instructions = more battery use
  - High code density
    - Less moving stuff around, best use of space

ARM and Thumb

- Every ARM instruction is 32bits long
  - Can take up a lot of memory
  - Memory accesses take time
  - Can slow things down
- Thumb
  - 16-bit version of the ARM instruction set
  - Variable length instruction set
  - Took the most popular ARM instructions used and encoded them as 16-bit values
  - Why?

ARM and Thumb

- Speed
  - ARM instructions require 32bits to be read every instruction
  - Memory reads take time
  - Not all memory is 32bit wide
    - On smaller RAM width, takes multiple reads to get an instruction
- By making the instructions smaller gain a speed up
  - 8-bit memory two reads per instruction
  - 16-bit memory, one read per instruction
  - 32-bit memory, one read gives two instructions
- iPhone SDK generates Thumb by default

Floating Point

- Thumb doesn't encode FPU instructions
- CPU would have to branch to ARM instructions to execute it
  - This takes time
- FPU heavy code is better compiled to ARM, not Thumb
  - Assuming the device has an FPU!

ARM big.LITTLE

- ARM's latest processor contains two cores
  - 2.0GHz quad core optimised for performance (big core)
  - 1.0GHz quad core optimised for energy efficiency (LITTLE core)
- Two Cores are architecturally consistent
- System can switch between the two as appropriate for the task in hand

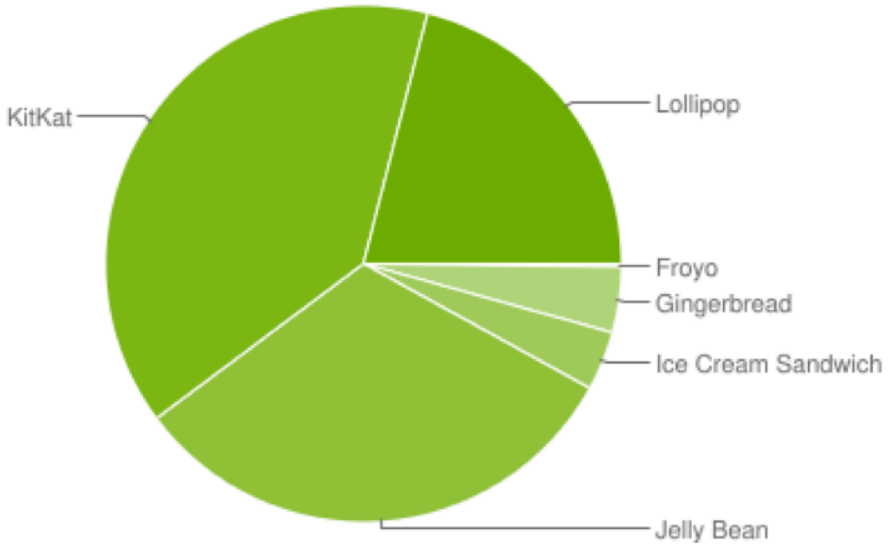
Android

- An operating system for mobile phones
- Purchased by Google in 2005
- Open (sort of)
  - Open source / Apache license eventually
  - (Bootloaders / rooting)
- Leverages existing technology
  - Linux (but not really Linux)
    - Linux kernel, custom middleware
  - Java (but not really Java)
- A different programming model

Android versions

- Several versions in the wild
- Android 1.5/1.6 – generally not seen anymore
- Android 2.2/2.3 - phones/cheap tablets
- Android 3.x (Honeycomb) - Tablets only
- Android 4.1/4.3 (Jelly Bean 16 - 18)
- Android 4.4 (KitKat 19)
- Lollipop 5.x
- Forks (e.g. Amazon Kindle Fire)

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	4.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	3.7%
4.1.x	Jelly Bean	16	12.1%
4.2.x		17	15.2%
4.3		18	4.5%
4.4	KitKat	19	39.2%
5.0	Lollipop	21	15.9%
5.1		22	5.1%

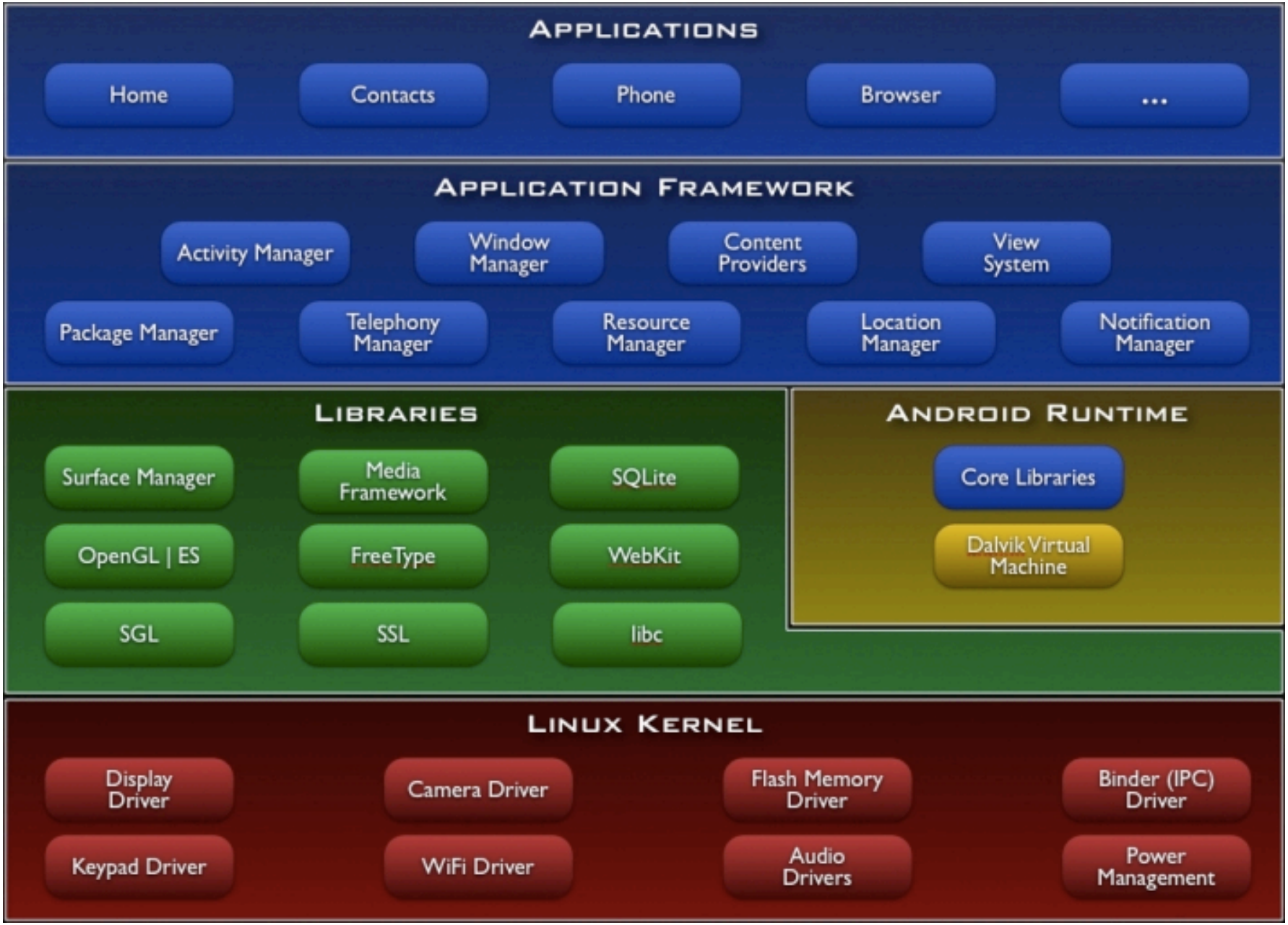


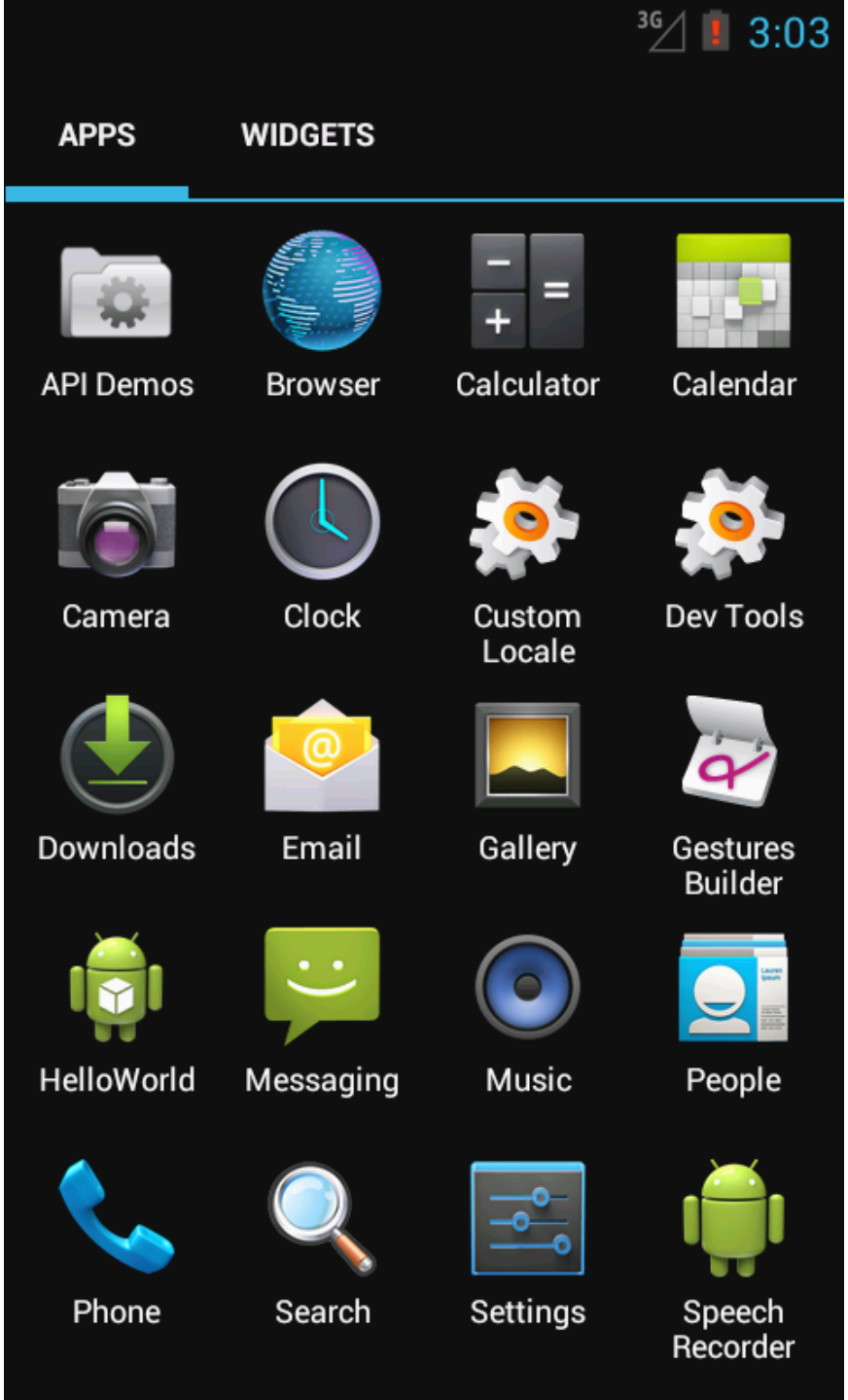
Android Compatibility

- Claims to be forwards / backwards compatible
  - An application built against 1.5 should work on the newest 5.* device
  - Some support for backwards compatibility
    - The newest features may have fallback equivalents in older APIs
    - Obviously cannot use an API that does not exist
    - Can restrict by specifying minimum API level
- The Android logo is CC licensed
- Can only call a device an “Android phone” if it passes compatibility tests / supports the API
  - “Android” the brand licensed to OMA members
    - Open Mobile Alliance

Android

- A software stack for mobile devices
  - Tablets, phones
- Operating system kernel
- Standard middleware
  - Android library support
- Key applications / user interfaces
  - Vendor specific modifications


Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager
Libraries
Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management


3:03
APPS
WIDGETS
API Demos
Browser
Calculator
Calendar
Camera
Clock
Custom Locale
Dev Tools
Downloads
Email
Gallery
Gestures Builder
HelloWorld
Messaging
Music
People
Phone
Search
Settings
Speech Recorder

```
1
2  import java.lang.System;
3
4  public class HelloWorld
5  {
6      public static void main(String args[])
7      {
8          System.out.println("hello world");
9      }
10 }
```

```
Code:
 0:   iconst_2
 1:   istore_1
 2:   iload_1
 3:   sipush  1000
 6:   if_icmpge       44
 9:   iconst_2
 10:  istore_2
 11:  iload_2
 12:  iload_1
 13:  if_icmpge       31
 16:  iload_1
 17:  iload_2
 18:  irem             # remainder
 19:  ifne    25
 22:  goto    38
 25:  iinc    2, 1
 28:  goto    11
 31:  getstatic       #84; //Field java/lang/System.out:Ljava/io/PrintStream;
 34:  iload_1
 35:  invokevirtual   #85; //Method java/io/PrintStream.println:(I)V
 38:  iinc    1, 1
 41:  goto    2
 44:  return
```

Android Apps

- Applications are written in Java
  - But run on Google's own VM — Dalvik
  - Uses its own bytecode (DEX) format
- Code compiled using standard Java tools then convert to DEX format
  - Multiple class files in a single .dex file
- Code, data and resource files packed into a .apk file
  - Classes
  - Configuration
  - Resources

Dalvik

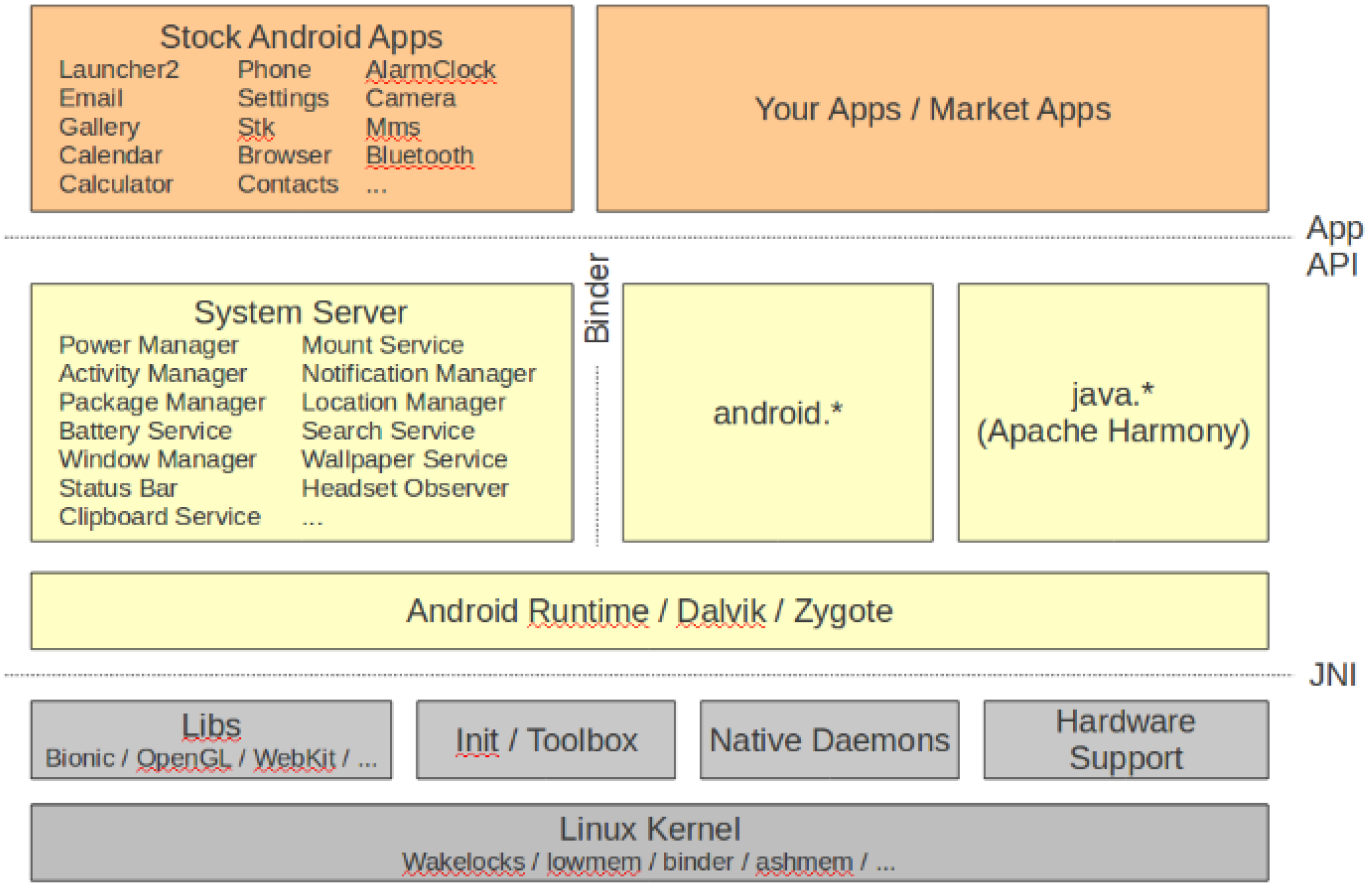
- A clean-room implementation of the JVM
- Sun Java
  - Java language, JDK compiler, JVM, JDK Libraries
  - java.*
- Android Java
  - Java language, JDK compiler, dx, Dalvik, Apache Harmony Libraries
  - Does not align to Java SE, Java ME
    - No AWT, Swing
  - Subset of java.*, android.*
- JVM interprets, executes .class files
- Dalvik interprets .dex files
  - Post-processes .class files
  - Size reduction, various optimisations
- Target slow cpu, no swap, low RAM, battery powered

Android Kernel

- Linux kernel
  - Not the same as a linux distribution
    - Ubuntu et al.
    - No standard libraries, UI etc
  - Bionic
    - GNU libc library variant
    - Legal / efficiency
- Android specific modifications
  - wakelocks – keep the phone awake
  - binder – interprocess communication
  - ashmem – shared memory
  - oom – kills processes when memory is low
  - alarm manager – wakes up the phone when necessary

Android Apps

- Applications are sandboxed
  - A security mechanism for separating running applications and data
  - Each application gets its sandbox in which to play
  - Why?
- Android application sandbox
  - Linux is a multi-user system
    - How many people use your phone at once?
  - Makes use of Linux permissions and security
    - Own process, own VM, own UID
    - Cannot access other application files / data / processes
      - Owner not generally given access to the root user
      - Root can access the entire system, hence “rooting” or “jailbreaking” a phone
    - Root != supervisor mode



SDK, Eclipse, .apk

Manifest:
Perms / SDK ver.

.dex, ddms

NDK, rootfs, initrc, adb

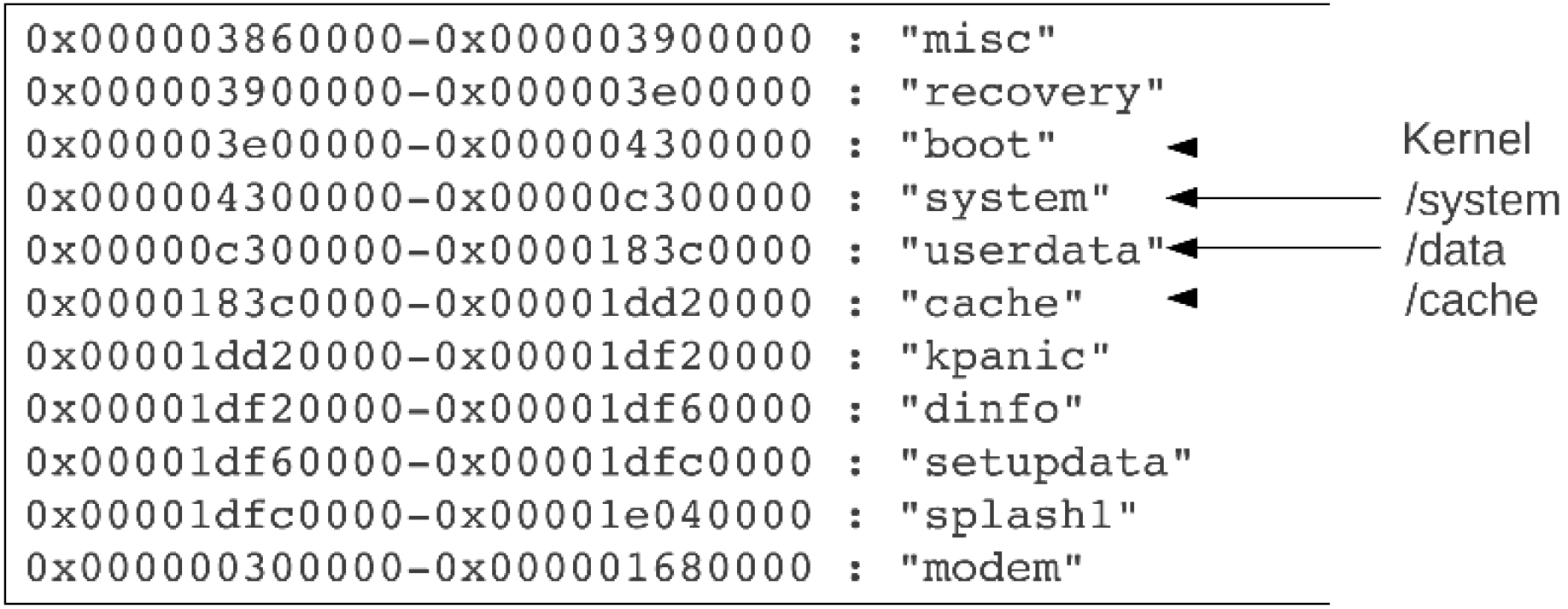
GNU toolchain

(fastboot)

Android Hardware Support

- Bluetooth - BlueZ
- GPS – Manufacturer provided libgps.so
- Wifi – wpa_supplicant
- Display – Standard framebuffer driver
- Keyboard – Standard input event
- Lights – Manufacturer provided liblights.so
- Audio – Manufacturer provided libaudio.so
- Camera – Manufacturer provided libcamera.so
- Power Management – “wakelocks” kernel patch
- Sensors – Manufacturer provided libsensors.so
- Radio – Manufacturer provided libril.so

Bootup


0x000003860000-0x000003900000 : "misc"
0x000003900000-0x000003e00000 : "recovery"
0x000003e00000-0x000004300000 : "boot"
0x000004300000-0x00000c300000 : "system"
0x00000c300000-0x0000183c0000 : "userdata"
0x0000183c0000-0x00001dd20000 : "cache"
0x00001dd20000-0x00001df20000 : "kpanic"
0x00001df20000-0x00001df60000 : "dinfo"
0x00001df60000-0x00001dfc0000 : "setupdata"
0x00001dfc0000-0x00001e040000 : "splash1"
0x000000300000-0x000001680000 : "modem"
Kernel
/system
/data
/cache

Bootup

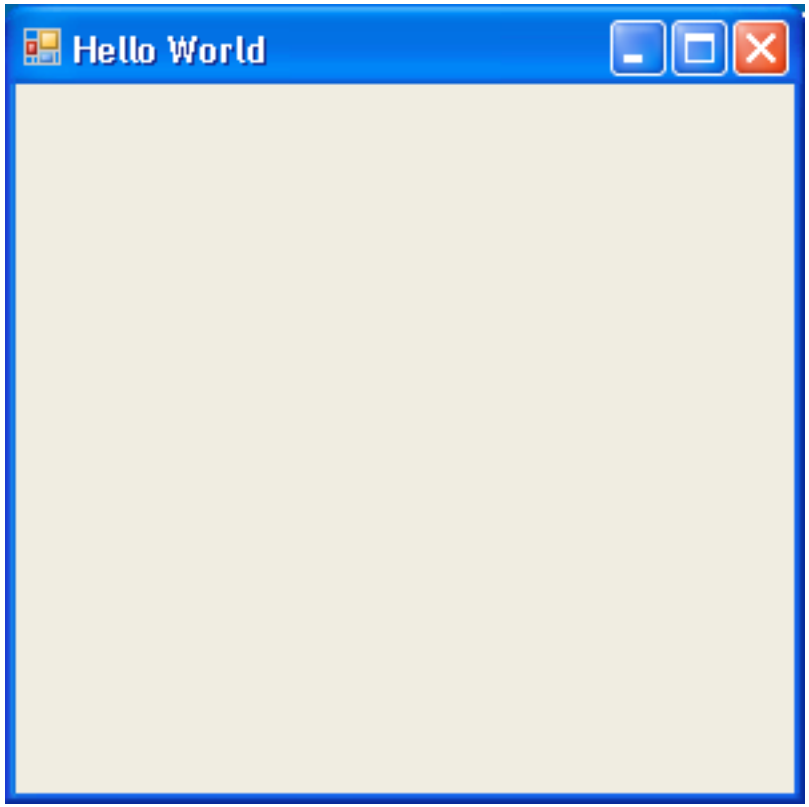
- Initialise the kernel
- Initialise device drivers
- Mount the root file system
- Execution of /init
  - Mount filesystems
  - Setup filesystem permissions
  - Set OOM properties
  - Start daemons
    - adbd
    - servicemanager (binder)
    - vold, netd, rild
    - app_process – Xzygote
    - …

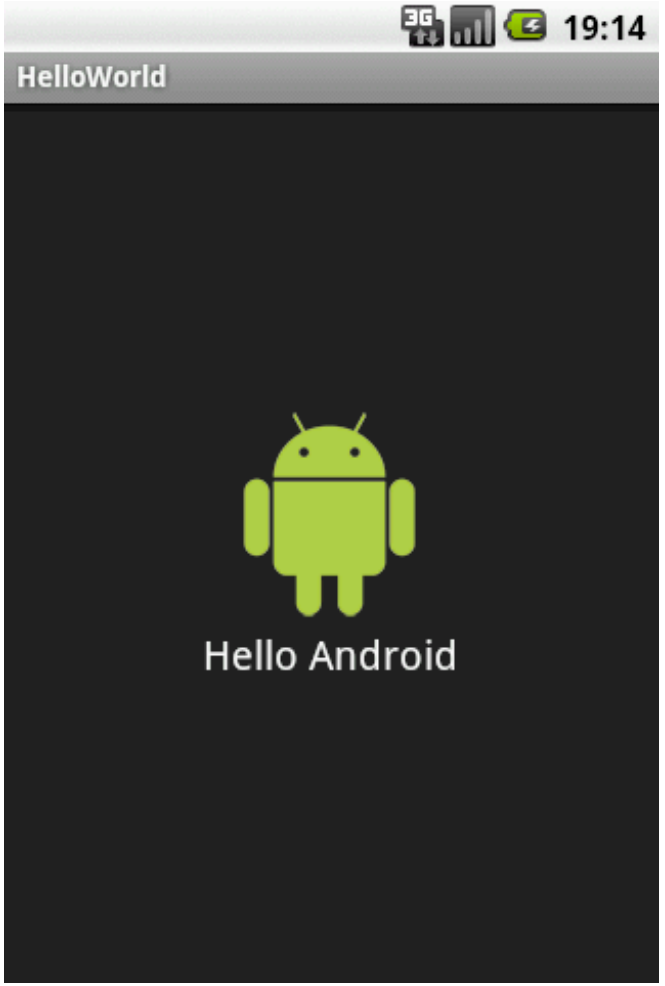
Zygote

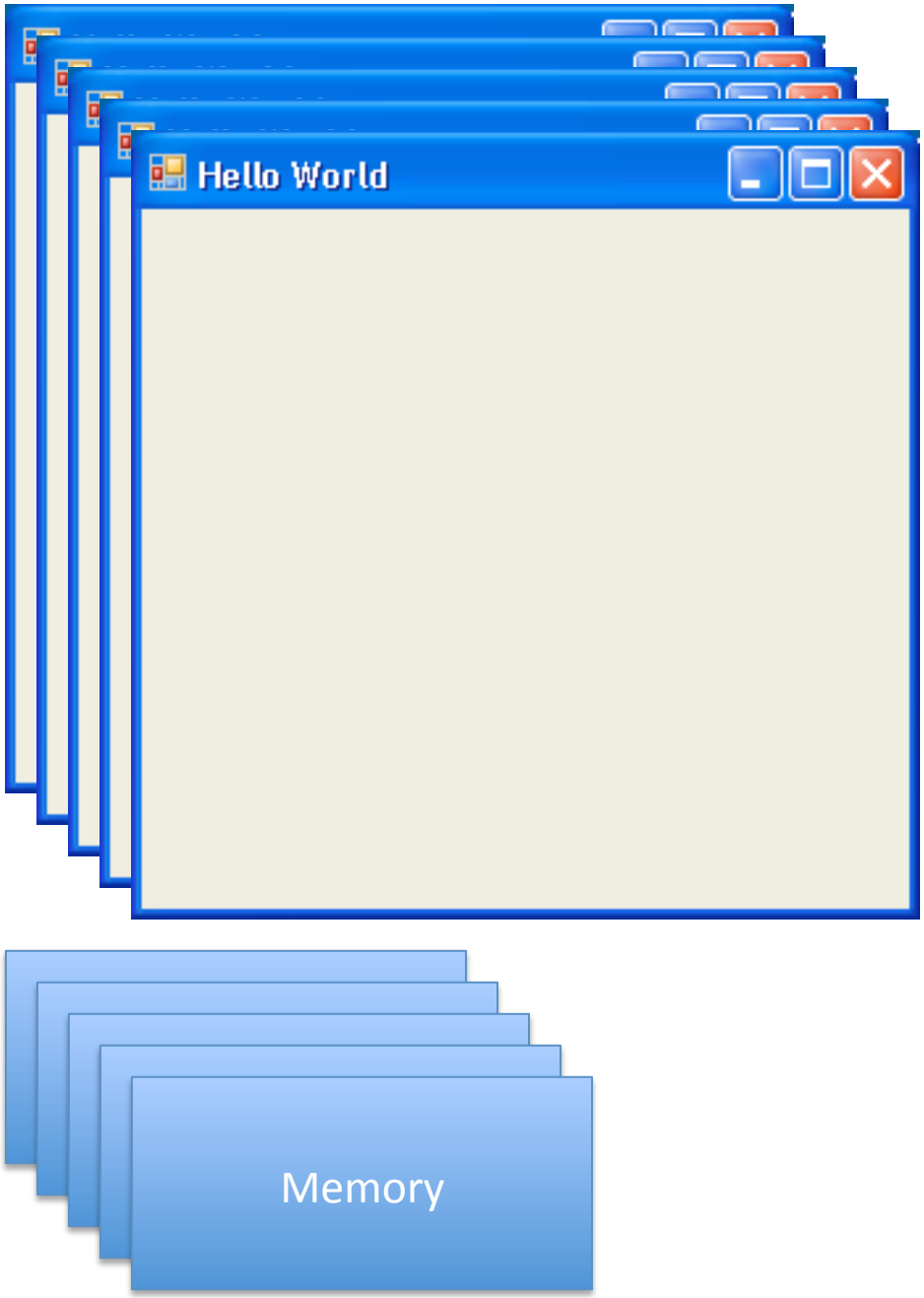
- “The first cell formed when an organism is produced.”
- app_main
  - Runtime.start(“com.android.internal.os.Zygote” …)
  - startVM()
  - Call Zygote’s main()
    - preloadClasses() ← what?
    - startSystemServer()
    - Call SystemServer’s run()
      - Start all system service/managers
      - Start ActivityManager
        - » Send Intent.CATEGORY_HOME
          - Launcher2

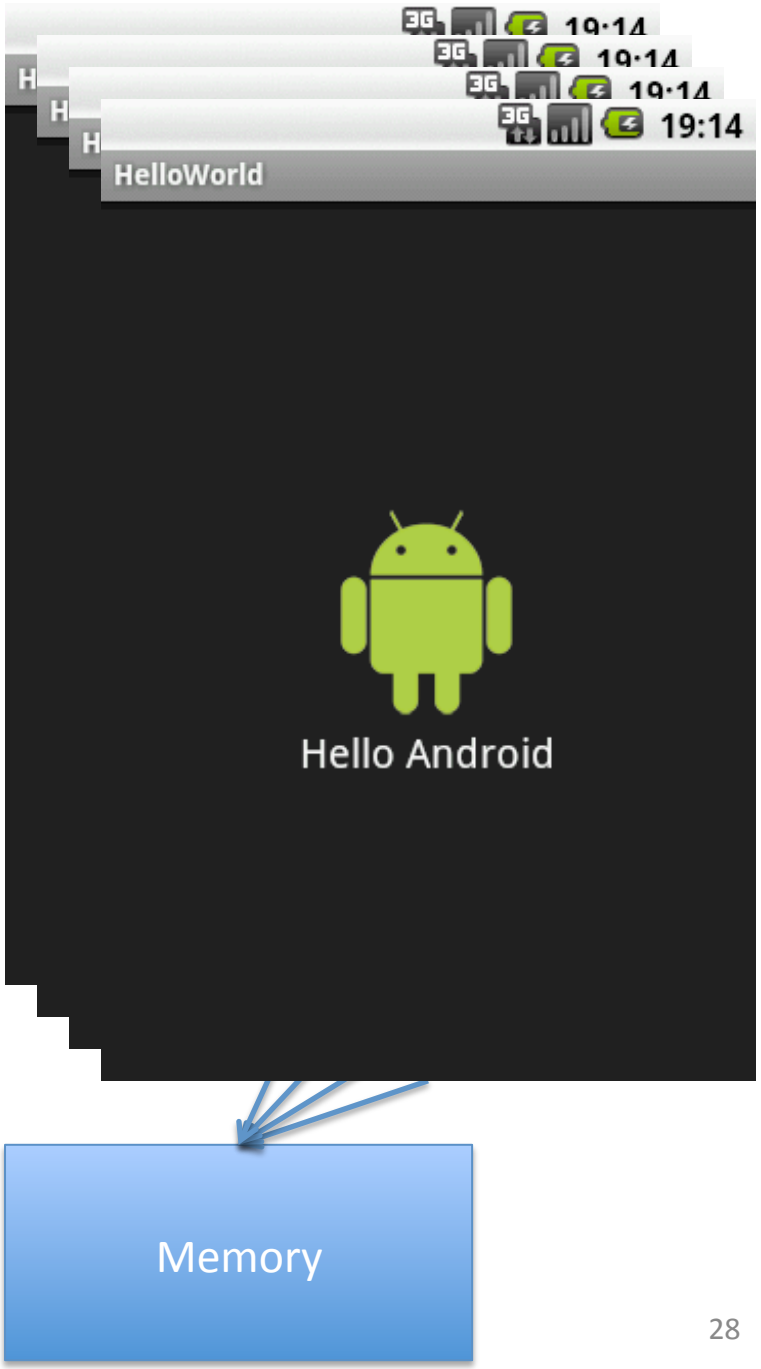
Shared Memory

- Load all java.*, android.* classes at boot time
- Initially create a single Dalvik VM process
  - Referencing classes loaded above
- When user runs an application…
  - onClick(Launcher)->startActivity(Activity.java)->Binder->ActivityManagerService->startViaZygote(Process.java)->Socket->Zygote)
  - **fork**
    - Creates a copy of itself in a separate address space
    - Does not copy **memory**, instead refers to original memory until modified
      - **Why?**


Hello World


19:14
HelloWorld
Hello Android


Hello World
Memory


19:14
HelloWorld
Hello Android
Memory

Android Programming Model

- Traditional OS applications
  - A single entry point
    - Main
  - OS loads the program into a process and executes it
- Java applications
  - A Java VM is instantiated
    - Loads all classes used by the application
    - Executes main

Android Programming Model

- Component based model
  - Multiple application entry points
    - The point through which the system can “enter” the application
  - Not all are entry points for the user
  - Each exists as a logical independent unique entity

Android Components

- Activities
  - UI components
- Services
  - Mechanism for doing something long-running in the background
- Broadcast Receivers
  - Respond to broadcast messages from the OS / other apps
- Content Providers
  - Make data available to / make use of data from other apps
  - No access to the file system
    - SD Card