Memory Management in Android

Android Builders Summit 2015

CC-BY-SA 3.0 - Attribution requirements and misc., **PLEASE READ**:

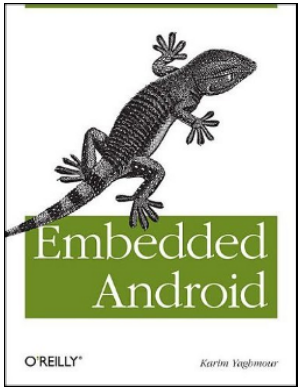
- This slide must remain as-is in this specific location (slide #1), everything else you are free to change; including the logo :-)
- Use of figures in other documents must feature the below "Originals at" URL immediately under that figure and the below copyright notice where appropriate.
- You are FORBIDDEN from using the default "About" slide as-is or any of its contents.

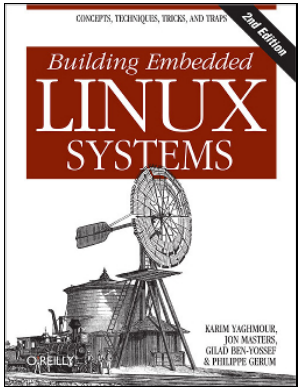
Copyright (C) 2014-2015, Opersys inc.

These slides created by: Karim Yaghmour

Originals at: **http://www.opersys.com/training/**

About


Embedded
Android
O'REILLY
Karim Yaghmour


CONCEPTS, TECHNIQUES, TRICKS, AND TRAPS
2nd Edition
Building Embedded
LINUX
SYSTEMS
KARIM YAGHMOUR,
JON MASTERS,
GILAD BEN-YOSSEF
& PHILIPPE GERUM
O'REILLY

- Introduced Linux Trace Toolkit in 1999
- Originated Adeos and relayfs (kernel/relay.c)
- Ara Android Arch Oversight
- Training, Custom Dev, Consulting, ...

"Note that memory usage on modern operating systems like Linux is an extremely complicated and difficult to understand area. In fact the chances of you actually correctly interpreting whatever numbers you get is extremely low. (Pretty much every time I look at memory usage numbers with other engineers, there is always a long discussion about what they actually mean that only results in a vague conclusion.)"

-- Dianne Hackborn, Feb 19, 2010, **Stackoverflow**

Agenda

1. Architecture Recap
2. Kernel's overall role
3. Kernel driver interfaces
4. Kernel user-space interfaces
5. Low-memory conditions
6. Bionic
7. App Dev Considerations
8. Tools
9. Misc.
10. References

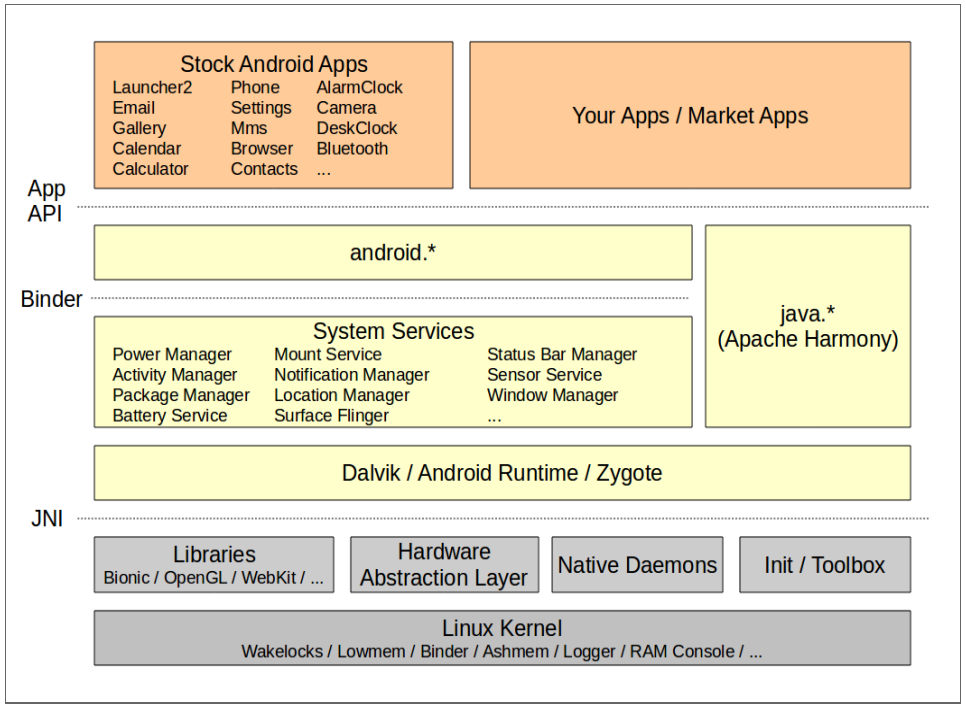
Demo Hardware

- IFC6410
- Qualcomm Snapdragon S4 Pro – APQ8064
- Krait CPU, 4-core, 1.7 GHz, 2MB L2 cache
- 2 GB on-board DDR3 (PCDDR 533MHz)
- 4 GB eMMC
- Separate power/usb
- Requires monitor/keyboard/mouse

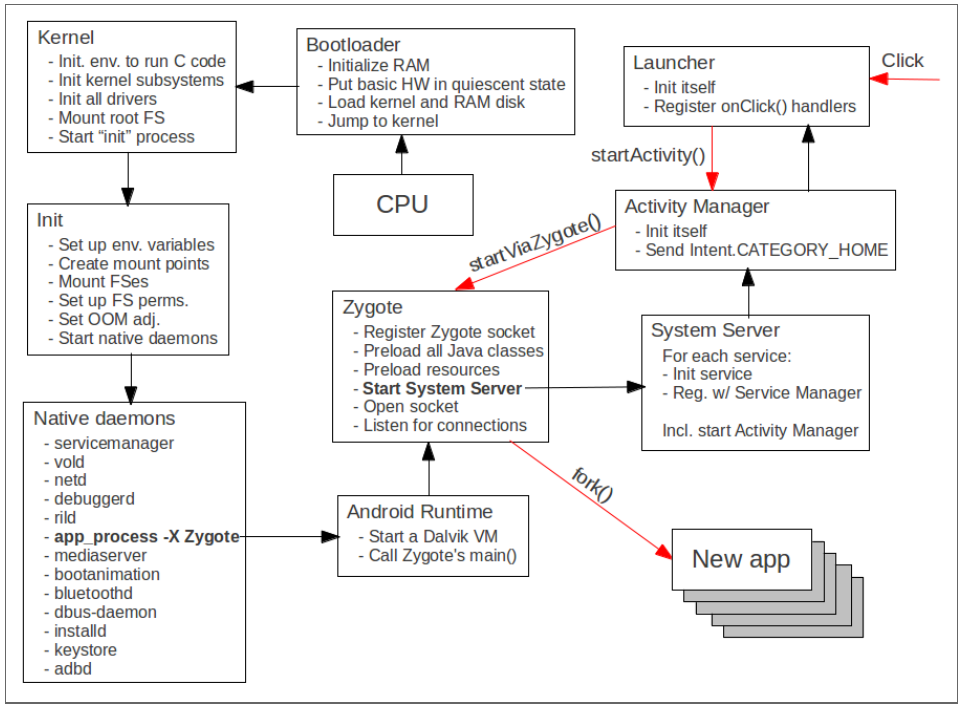
Architecture Recap

- AOSP
- System startup
- Memory layout

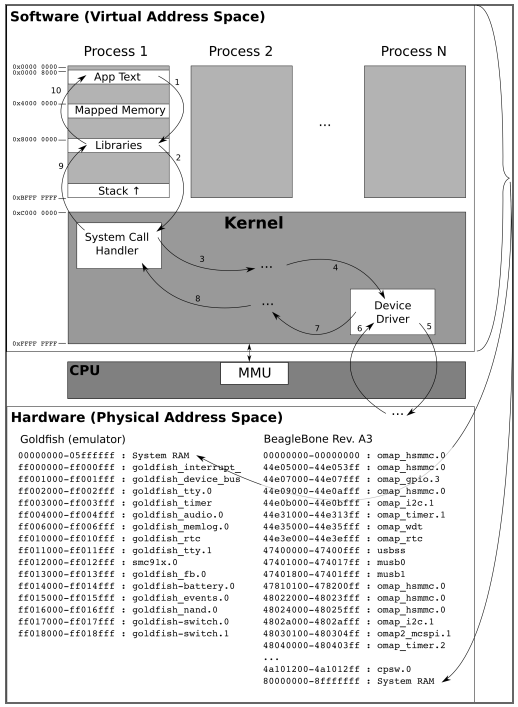
1. AOSP


Stock Android Apps
Launcher2 Phone AlarmClock
Email Settings Camera
Gallery Mms DeskClock
Calendar Browser Bluetooth
Calculator Contacts ...
Your Apps / Market Apps
App API
android.*
Binder
System Services
Power Manager Mount Service Status Bar Manager
Activity Manager Notification Manager Sensor Service
Package Manager Location Manager Window Manager
Battery Service Surface Flinger ...
java.*
(Apache Harmony)
Dalvik / Android Runtime / Zygote
JNI
Libraries
Bionic / OpenGL / WebKit / ...
Hardware Abstraction Layer
Native Daemons
Init / Toolbox
Linux Kernel
Wakelocks / Lowmem / Binder / Ashmem / Logger / RAM Console / ...

2. System startup



3. Memory layout



Kernel's overall role

- Manage physical memory
- Manage virtual-to-physical mappings
- Process isolation:
  - Protect the hardware from user-space
  - Protect processes from each other
- Driver capabilities:
  - Memory-mapped registers
  - DMA
  - MTD maps
- Filesystem cache
- File mapping
- Swapping

Internals

- Architecture-independent:

```
mm/
```

- Architecture-specific:

```
arch/arm/mm/
```

- Per-task struct (include/linux/sched.h):

```
struct task_struct {
...
    struct mm_struct *mm, *active_mm;
...
```

- The memory structures (include/linux/mm_types.h):

```
        struct mm_struct {
struct vm_area_struct * mmap;        /* list of VMAs */
struct rb_root mm_rb;
struct vm_area_struct * mmap_cache;    /* last find_vma result */
```

Kernel driver interfaces

- Memory-mapped registers
- DMA
- MTD maps
- ION

Kernel user-space interfaces

- brk
- mmap/munmap

Low-memory conditions

- OOM killer
- oom_adj
- Android low-mem driver
- oom_adj set during init
- Modifications to oom_adj at runtime by framework

Bionic

- malloc()/free()
  - Comes from Doug Lea's dlmalloc
  - Public Domain
  - See bionic/libc/upstream-dlmalloc/
  - Tutorial/doc: **http://g.oswego.edu/dl/html/malloc.html**
  - **https://en.wikipedia.org/wiki/C_dynamic_memory_allocation**
  - Dates back to 1987
  - Uses CALL_MORECORE() macro do allocations
    - Based on sbrk()
- dlopen()/dlsym()

- Flags to debug/observe malloc/free Linux

```
$ adb shell setprop libc.debug.malloc 1
$ adb shell stop
$ adb shell start
```

- Enable native monitoring by DDMS:
  - Open ~/.android/ddms.cfg
  - Add line stating: "native=true"

App Dev Considerations

- Recommendations given by Google **https://developer.android.com/training/articles/memory.html**
- Measuring app mem usage **https://developer.android.com/reference/android/app/ActivityManager.html#getProcessMemoryInfo%28int[]%29**

```
public MemoryInfo[] getProcessMemoryInfo (int[] pids)
```

- Getting system mem usage **https://developer.android.com/reference/android/app/ActivityManager.html#getMemoryInfo%28android.app.ActivityManager.MemoryInfo%29**

```
public void getMemoryInfo (ActivityManager.MemoryInfo outInfo)
```

- android.os.Debug
- android:largeHeap="true"

Tools

- Kernel
- Native
- Dalvik/ART
- Framework

8.1 Kernel

- Overall memory use
- Physical-mapped reg address ranges
- FS cache
- "fragmentation"
- /proc interface
  - RSS, VSS, USS, PSS, etc.
  - /proc/[pid]/maps, semantics of

8.2 Native

- dumpstate
- librank
- procrank
- procmem
- showmap
- tombstones
- debuggerd
- core files

8.3 Dalvik/ART

- Heap size measurement
- API in apps to get access to heap size from Runtime **https://developer.android.com/reference/java/lang/Runtime.html#maxMemory%28%29**

```
public long maxMemory ()
```

- MAT/Eclipse
- dalvik.vm.heapsize

8.4. Framework

- dumpsys meminfo
- dumpsys procinfo

Misc.

- DDOS on memory
- KitKat efforts for low-mem
- Security aspects
- HAL use of mmap
- Swap space?
  - zMAP
- ION
  - CMA

References

- developer.android.com
- source.android.com
- **http://source.android.com/devices/native-memory.html**
- **http://source.android.com/devices/low-ram.html**
- **https://developer.android.com/tools/debugging/debugging-memory.html**
- **https://developer.android.com/training/articles/memory.html**
- **http://android-developers.blogspot.com/2011/03/memory-analysis-for-android.html**
- **http://android-developers.blogspot.com/2009/02/track-memory-allocations.html**
- **https://stackoverflow.com/questions/2298208/how-to-discover-memory-usage-of-my-application-in-android**
- **http://elinux.org/Android_Memory_Usage**
- **https://www.youtube.com/watch?v=_CruQY55HOk**
- **http://blog.yojimbocorp.com/2012/10/03/view-android-application-memory-usage-with-eclipse-ddms-plugin/**
- **https://lwn.net/Articles/480055/**
- **https://lwn.net/Articles/565469/**

strong/weak/soft/phantom references wrt Java GC: - **https://weblogs.java.net/blog/2006/05/04/understanding-weak-references** - **http://docs.oracle.com/javase/7/docs/api/java/lang/ref/package-summary.html** -->

Thank You!

karim.yaghmour@opersys.com