EE5414 Development and Design in Embedded Systems

EE Department
Android Basic

images

- kernel-qemu
- ramdisk.img
- system.img
- userdata.img

tools/emulator.exe (QEMU + Goldfish)

Android Emulator
Android Platform

- kernel-qemu
- ramdisk.img
- system.img
- userdata.img

Android Emulator

tools/emulator.exe (QEMU + Goldfish)
Zygote and Dalvik VM

- **Zygote**
- **App. Framework**
  - System Services
  - Hardware Services
- **Dalvik VM**
- **Native Server**
  - Audio Flinger
  - Surface Flinger
- **System Server**
- **Home**
- **Dalvik VM**

**daemons**
- usbd
- adbd
- debuggerd
- rild

**libc**

**Service Manager**

**Kernel**

**Binder Driver**

**Init**

**Binder Driver**
Dalvik Virtual Machine

Considering the following environment for Bytecode Interpreter

- Slow CPU (250-500 MHz)
- RAM Usage: Low-level: 20M, High-level: 24M (system library: 10M)
- Little RAM (64MB): Available RAM: 20M
- Bus speed: 100MHz
- Data Cache: 16~32K
- No swap space, Battery power

Register Machine & Special Instruction

- More efficient interpreter
  - avoid instruction dispatch
  - avoid unnecessary memory access
- consume instruction stream efficiently
  - Higher semantic density of instructions
- Special instructions for problem cases
Register vs. Stack

* Stack, Accumulator, Register machine

**Stack machine**: 0-operand instruction set
**Accumulator machine**: 1-operand …
**Register machine**: 2 or 3-operand …

---

**Dalvik Virtual Machine**:  

```java
public static long sumArray(int[] arr) {
    long sum = 0;
    for (int i : arr) {
        sum += i;
    }
    return sum;
}
```

---

**Java VM(JVM) bytecode**

```
000b: iload 05
000d: iload 04
000f: if_icmpge 0024
0012: aload_3
0013: iload 05
0015: iaload
0016: istore 06
0018: lload_1
0019: iload 06
001b: i2l
001c: ladd
001d: lstore_1
001e: iinc 05, #+01
0021: goto 000b
```

---

**Dalvik VM(DVM) bytecode**

```
0007: if-ge v0, v2, 0010
0009: aget v1, v8, v0
000b: int-to-long v5, v1
000c: add-long/2addr v3, v5
000d: add-int/lit8 v0, v0, #int 1
000f: goto 0007
```

---

<table>
<thead>
<tr>
<th>Items</th>
<th>JVM</th>
<th>DVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>bytes</td>
<td>25</td>
<td>18</td>
</tr>
<tr>
<td>dispatches</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>reads</td>
<td>45</td>
<td>19</td>
</tr>
<tr>
<td>writes</td>
<td>16</td>
<td>6</td>
</tr>
</tbody>
</table>
Libraries

- **Bionic**: linker, libc.so, libdl.so, libm.so, libstdc++.so, libthead_db.so, libz.so, libGLES*.so  
  **NDK (Native Development Kit) Limitation** (for forward compatibility)
- **Function**: Many Library
- **Native Server**: Audio Flinger, Surface Flinger
- **Hardware Abstraction Layer**: Proprietary Area

![Diagram of Libraries and Hardware Abstraction Layer]

**Libraries**

- Surface Manager
- Media Framework
- OpenGL|ES
- FreeType
- SQLite
- SGL
- Libc
- WebKit
- SSL
- ...  

**Hardware Abstraction Layer**

- Graphics
- Audio
- Camera
- Bluetooth
- GPS
- Radio(RIL)
- WiFi
- ...
Libraries

- Manages all audio output devices
- Processes multiple audio streams into **PCM audio out** paths
- Handles audio routing to various outputs
Android is based on Linux but Android is not Linux. 
Android, such as X-Window does not include built-in Windows system. 
Android does not support glibc. 
Android includes a standard Linux Utility but not a full version.

Linux kernel versions 2.6.23, 2.6.24, 2.6.25, 2.6.27, 2.6.29, 2.6.32 are used by Android. 
In order to support Android, the Linux kernel includes a patch for the expansion.

The reason for using Linux on Android, it is because the memory and process managements are based on the model of the beam, and due to the advantages of open source, support of proven driver model and shared library.

The addtion modules to the Android Linux kernel are Alarm, Ashmem, Binder, Power Management, Low Memory Killer, Kernel Debugger and Logger.

The current Android Linux kernel source distribution is at the URL below. 
http://git.android.com
# Kernel

<table>
<thead>
<tr>
<th>Key Features</th>
<th>2.6.23</th>
<th>2.6.25</th>
<th>2.6.27</th>
<th>2.6.29</th>
<th>2.6.32</th>
<th>2.6.33</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Binder</strong></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Alarm</strong></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Logger</strong></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Low Memory Killer</strong></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Power Management</strong></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>USB Gadget</strong></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Kernel Debugger</strong></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ASHMEM</strong></td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PMEM</strong></td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>x86 Support</strong></td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>/drivers/staging/Android</strong></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Module</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Binder</strong></td>
<td>an OpenBinder-based driver to facilitate inter-process communication (IPC) in the Android platform.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Power Management</strong></td>
<td>a light weight power management driver built on top of standard Linux power management but optimized for embedded systems.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Low Memory Killer</strong></td>
<td>Based on hints from the userspace, the low memory killer can kill off processes to free up memory as necessary. It is designed to provide more flexibility than the Out Of Memory (OOM) killer in the standard kernel.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Logger</strong></td>
<td>A light weight logging device used to capture system, radio, logdata, etc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>USB Gadget</strong></td>
<td>Uses the USB function framework.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ashmem</strong></td>
<td>Anonymous SHared MEMory, or ASHMEM, is a named memory block that is shared between processes that the kernel is allowed to free. This is notable as the kernel is not allowed to free standard shared memory.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PMEM</strong></td>
<td>The PMEM (physical memory) driver is used to provide contiguous physical memory regions to userspace libraries that interact with the digital signal processor (DSP) and other hardware that cannot cope with scatter-gather.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Alarm</strong></td>
<td>A driver which provides timers that can wake the device up from sleep and a monotonic timebase that runs while the device is asleep.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Embedded Bootloader, Kernel and File System

**Bootloader & Kernel**
- ARM based on OMAP3530 Target Board
- Android bootloader
  - USB loader
- Load Linux
- Start kernel - /init/main.c

**Android Open Source**
- Initialization process - /etc/init.rc
- Start Android Services -
  - Console, adbd, service manager, mountd, debuggerd, ril-daemon, zygote, mediaserver, installd, flash_recovery
## Android processes

<table>
<thead>
<tr>
<th>USER</th>
<th>PID</th>
<th>PPID</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>1</td>
<td>0</td>
<td>/init</td>
</tr>
<tr>
<td>system</td>
<td>49</td>
<td>1</td>
<td>/system/bin/servicemanager</td>
</tr>
<tr>
<td>root</td>
<td>49</td>
<td>1</td>
<td>/system/bin/debuggerd</td>
</tr>
<tr>
<td>radio</td>
<td>50</td>
<td>1</td>
<td>/system/bin/rild</td>
</tr>
<tr>
<td>root</td>
<td>51</td>
<td>1</td>
<td>zygote</td>
</tr>
<tr>
<td>media</td>
<td>52</td>
<td>1</td>
<td>/system/bin/mediaserver</td>
</tr>
<tr>
<td>root</td>
<td>54</td>
<td>1</td>
<td>/system/bin/installld</td>
</tr>
<tr>
<td>root</td>
<td>58</td>
<td>1</td>
<td>/sbin/adbd</td>
</tr>
<tr>
<td>system</td>
<td>76</td>
<td>51</td>
<td>system_server</td>
</tr>
<tr>
<td>app_5</td>
<td>137</td>
<td>51</td>
<td>com.android.inputmethod.voic</td>
</tr>
<tr>
<td>radio</td>
<td>139</td>
<td>51</td>
<td>e com.android.phone</td>
</tr>
<tr>
<td>app_29</td>
<td>148</td>
<td>51</td>
<td>com.amazon.mp3</td>
</tr>
<tr>
<td>app_1</td>
<td>153</td>
<td>51</td>
<td>android.process.acore</td>
</tr>
<tr>
<td>app_38</td>
<td>167</td>
<td>51</td>
<td>com.google.android.apps.uploader</td>
</tr>
<tr>
<td>app_6</td>
<td>181</td>
<td>51</td>
<td>com.google.process.gapps</td>
</tr>
<tr>
<td>app_17</td>
<td>183</td>
<td>51</td>
<td>android.process.media</td>
</tr>
<tr>
<td>app_5</td>
<td>235</td>
<td>51</td>
<td>com.google.android.voicesearch</td>
</tr>
<tr>
<td>system</td>
<td>249</td>
<td>51</td>
<td>com.android.settings</td>
</tr>
<tr>
<td>app_30</td>
<td>265</td>
<td>51</td>
<td>com.google.android.apps.googlevoice</td>
</tr>
<tr>
<td>app_9</td>
<td>271</td>
<td>51</td>
<td>e com.android.voicedialer</td>
</tr>
<tr>
<td>app_36</td>
<td>289</td>
<td>51</td>
<td>com.android.mms</td>
</tr>
<tr>
<td>app_2</td>
<td>309</td>
<td>51</td>
<td>com.google.android.apps.maps:FriendService</td>
</tr>
<tr>
<td>app_37</td>
<td>319</td>
<td>51</td>
<td>e com.google.android.partnersetup</td>
</tr>
<tr>
<td>app_33</td>
<td>325</td>
<td>51</td>
<td>com.android.email</td>
</tr>
<tr>
<td>app_60</td>
<td>361</td>
<td>51</td>
<td>com.cyandroid.wakeupcallalarm</td>
</tr>
<tr>
<td>app_27</td>
<td>376</td>
<td>51</td>
<td>com.cooliris.media</td>
</tr>
<tr>
<td>app_18</td>
<td>560</td>
<td>51</td>
<td>com.android.vending</td>
</tr>
<tr>
<td>app_8</td>
<td>619</td>
<td>51</td>
<td>android.process.acore</td>
</tr>
</tbody>
</table>
zygote, system_server, apps, binder ipc studies

- **zygote**
  - Runtimelnit
  - AndroidRuntime
  - ZygoteInit
  - Dalvik VM

- **system_server**
  - ApplicationContext
  - Application
  - ApplicationThread
  - ActivityThread
  - Instrumentation
  - WindowManager
  - ActivityManager
  - PackageManager

- **process_name (eg. android.process.acore)**
  - ApplicationContext
  - Application
  - ApplicationThread
  - ActivityThread
  - Instrumentation
  - Activity
  - ContentProvider
  - Service

The flow involves:
- **fork** from zygote to system_server
- **socket** from system_server to process_name
- **fork** from system_server to process_name
- **binder ipc** from system_server to process_name
File system Overview

- Android system and Embedded system
File System Abstraction Layer

- Application
- C library (Libc)
- Virtual File System (VFS)
  - ext2, ext3
  - JFFS2, YAFFS2
- FTL or NFTL
- MTD => Flash memory

User Space
Abstraction Layer
Kernel Space
Hardware
File System Abstraction Layer

- Interplay of VFS components
Network File System

NFS Mapping
Task & Scheduling & Preemption

CPU2

CPU1

device

interrupt

userspace

kernel

Interrupt kernel

interrupt

Interrupt handler

tasklet

Kernel thread

scheduler

critical region

Virtual memory (resource)

3G

1G

System call

User application
Context Switching

Process A (ex: mediaserver)

main()
{
...
}

task_struct

CPU status

1) Registers for process address space
2) PC (Program Counter)
3) SP (Stack Pointer)
4) Some registers

CPU

kernel stack

Process B (ex: system_server)

main()
{
...
}

task_struct

CPU status

save
restore
Linux Kernel

Syscall from userspace (SWI – Software Interrupt)

main routine
1) Initialize the system
2) forever loop
   while (1) {
       schedule();
   }

Syscall handling routine

Memory management ...

Interrupt routine

Hardware interrupt from CPU

Scheduler

Process dispatcher

Process (task)

Kernel thread

task list

Scheduling request

Schedule
Yield ...

asynchronously

asynchronously
Course Contents

- Describe the embedded Linux Kernel, Device Driver, and Device Driver Modules
- Apply Android and port it to an embedded device, and to configure the Linux kernel and drivers to support Android.
- Implement and design hardware interfacing with Embedded Linux.
- Mini Projects and Laboratories for embedded devices

The link for the course:

http://www.cityu.edu.hk/pg/current/course/EE5414.htm
Who should take this course?

- For those interested in embedded system design.
- This course will teach you how to port a Linux or Android OS to an embedded device.
- Hand on experience will be provided in this course.
Related Courses

- EE5806 Topics in Image Processing
- EE5413 Advanced Internet Technologies
- EE5415 Mobile Applications Design and Development