Android Overview

1. 【**Android-introduction**】**Android** 介绍与前景
2. 【**Android-Architecture**】**Android** 核心模块及相关技术
3. 【**Android-kernel**】**Android** 内核改动

http://blog.csdn.net/magicyu2/archive/2009/10/26/4728521.aspx

1.【Android-introduction】Android 介绍与前景

Android 介绍

Android 一词的本义指“机器人”，2003 年美国有一家以 Android 为名的小公司成立，开发手机平台。Google 收购 Android 之后，于 07 年 11 月 5 日发布了开源的 Android 平台——一款包括操作系统（基于 Linux 内核）、中间件和关键应用的手机平台，并组建了开放手机联盟(Open Handset Alliance),包括 Google、中国移动、T-Mobile、宏达电、高通、摩托罗拉等领军企业。

2008 年 9 月 22 日，美国运营商 T-Mobile USA 在纽约正式发布第一款 Google 手机——T-Mobile G1。该款手机为宏达电制造，是世界上第一部使用 Android 操作系统的手机。

Android 项目的官方网站是：http://www.android.com/

Android 的前景

Android 手机系统的一个很大的优势在于其开放性和服务免费。Android 是一个对第三方软件完全开放的平台，开发者在为其开发程序时拥有更大的自由度，突破了 iPhone 等只能添加为数不多的固定软件的枷锁；同时与 Windows Mobile、Symbian 等厂商不同，Android 操作系统免费向开发人员提供，这样可节省近三成成本。

Android 项目目前正在从手机运营商、手机厂商、开发者和消费者那里获得大力支持。从下面列出的开放手机联盟成员可以看出其强大的实力：

开放手机联盟成员：

一、手机制造商：
台湾宏达国际电子（HTC)（Palm 等多款智能手机的代工厂）
摩托罗拉（美国最大的手机制造商）
韩国三星电子（仅次于诺基亚的全球第二大手机制造商）
韩国 LG 电子
中国移动（全球最大的移动运营商，有 3.5 亿用户）
日本 KDDI（2900 万用户）
日本 NTT DoCoMo(5200 万用户）
美国 Sprint Nextel（美国第三大移动运营商，5400 万用户）
意大利电信（Telecom Italia）（意大利主要的移动运营商，3400 万用户）
西班牙 Telefónica （在欧洲和拉美有 1.5 亿用户）
T-Mobile（德意志电信旗下公司，在美国和欧洲有 1.1 亿用户）

二、半导体公司：
Audience Corp（声音处理器公司）
Broadcom Corp（无线半导体主要提供商）
英特尔（Intel）
Marvell Technology Group
Nvidia （图形处理器公司）
SiRF（GPS 技术提供商）
Synaptics（手机用户界面技术）
德州仪器（Texas Instruments）
高通（Qualcomm ）

三、软件公司：

Aplix
Ascender
eBay 的 Skype
Esmertec
Living Image
NMS Communications
Noser Engineering AG
Nuance Communications
PacketVideo
SkyPop
Sonix Network
TAT-The Astonishing Tribe
Wind River Systems

Android 同时也获得了大量开发者的支持。据 Google 称，在其推出后的两个月内，程序员下载这个软件开发平台的次数已经高达 250000 多次。尽管在如 此庞大的下载数量中只有一小部分开发者真正开始编写实际的应用程序，但我们却由此可以看出惊人的开发商兴趣。相比之下，在 Symbian 操作系统推出一年 后，其操作指南的下载次数只有 70000 次左右。

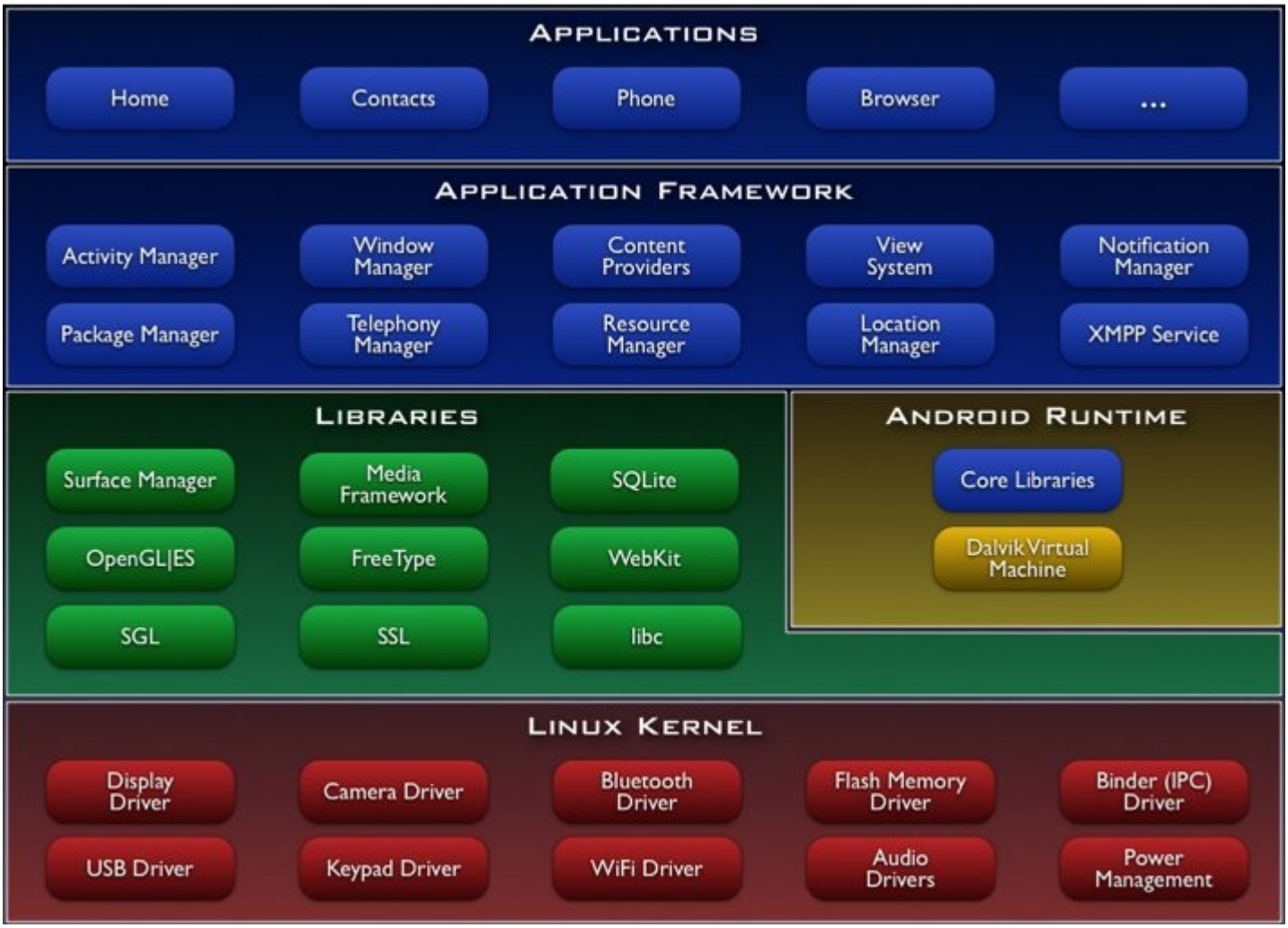
HTC CEO 周永明(Peter Chou)称，基于 Google Android 平台的 T-MobileG1 手机有望在今年底达到 100 万部的出货量，而此前预计的数量只有 60 万部。看起来“Google 手机”尽管宣传比较低调，但实际卖得还不错，前景也被人看好。

Android 在中国

为开放手机联盟的初始成员，中国移动早在去年就开始招兵买马，开发自己的 Android 手机。有消息人士表示，中国移动可能于 09 年推出 Android 手机（Ophone）。

另外，各山寨厂商也在大力进行 Android 的移植工作，更有所谓的首款中文 Android 手机琦基 i6 已经曝光，估计 09 年将会有多款 Android 手机推出。

2.【Android-Architecture】Android 核心模块及相关技术



Android 作为一个移动设备的平台，其软件层次结构包括了一个操作系统（OS），中间件（MiddleWare）和应用程序（Application）。

根据 Android 的软件框图，其软件层次结构自下而上分为以下几个层次:
第一、操作系统层（OS）
第二、各种库（Libraries）和 Android 运行环境（RunTime）
第三、应用程序框架（Application Framework）
第四、应用程序（Application）
以下分别介绍 Andoid 各个层次的软件的重点及其相关技术:

● 操作系统层（OS）
Android 使用 Linux2.6 作为操作系统，Linux2.6 是一种标准的技术，Linux 也是一个开放的操作系统。Android 对操作系统的使用包括核心和驱动程序两部分，Android 的 Linux 核心为标准的 Linux2.6 内核，Android 更多的是需要一些与移动设备相关的驱动程序。主要的驱动如下所示:
显示驱动（Display Driver）：常用基于 Linux 的帧缓冲（Frame Buffer）驱动。
Flash 内存驱动（Flash Memory Driver）
照相机驱动（Camera Driver）：常用基于 Linux 的 v4l（Video for ）驱动。

音频驱动（Audio Driver）：常用基于 ALSA（Advanced Linux Sound Architecture，高级 Linux 声音体系）驱动。
WiFi 驱动（WiFi Driver）：基于 IEEE 802.11 标准的驱动程序
键盘驱动（KeyBoard Driver）
蓝牙驱动（Bluetooth Driver）
Binder IPC 驱动： Andoid 一个特殊的驱动程序，具有单独的设备节点，提供进程间通讯的功能。
Power Management（能源管理）

● 各种库（Libraries）和 Android 运行环境（RunTime）
本层次对应一般嵌入式系统，相当于中间件层次。Android 的本层次分成两个部分一个是各种库，另一个是 Android 运行环境。本层的内容
大多是使用 C++实现的。
在其中，各种库包括：
■ C 库 ：C 语言的标准库，这也是系统中一个最为底层的库，C 库是通过 Linux 的系统调用来实现。
■ 多媒体框架（MediaFrameword）：这部分内容是 Android 多媒体的核心部分，基于 PacketVideo（即 PV）的 OpenCORE，从功能上本库一共
分为两大部分，一个部分是音频、视频的回放（PlayBack），另一部分是则是音视频的纪录（Recorder）。
■ SGL：2D 图像引擎。
■ SSL：即 Secure Socket Layer 位于 TCP/IP 协议与各种应用层协议之间,为数据通讯提供安全支持。
■ OpenGL ES 1.0：本部分提供了对 3D 的支持。
■ 界面管理工具（Surface Management）：本部分提供了对管理显示子系统等功能。
■ SQLite：一个通用的嵌入式数据库
■ WebKit：网络浏览器的核心
■ FreeType：位图和矢量字体的功能。
Android 的各种库一般是以系统中间件的形式提供的，它们均有的一个显著特点就是与移动设备的平台的应用密切相关。

Android 运行环境主要指的虚拟机技术——Dalvik。Dalvik 虚拟机和一般 JAVA 虚拟机（Java VM）不同，它执行的不是 JAVA 标准的字节码（bytecode ）而是 Dalvik 可执行格式（.dex）中执行文件。在执行的过程中，每一个应用程序即一个进程（Linux 的一个 Process）。二者最大的区别在于 Java VM 是以基于栈的虚拟机（Stack-based），而 Dalvik 是基于寄存器的虚拟机（Register-based）。显然，后者最大的好处在于可以根据硬件实现更大的优化，这更适合移动设备的特点。

● 应用程序框架（Application Framework）
Android 的应用程序框架为应用程序层的开发者提供 APIs，它实际上是一个应用程序的框架。由于上层的应用程序是以 JAVA 构建的，因此本
层次提供的首先包含了 UI 程序中所需要的各种控件：
例如： Views (视图组件)包括 lists(列表), grids(栅格), text boxes(文本框), buttons(按钮)等。甚至一个嵌入式的 Web 浏览器。
一个 Andoid 的应用程序可以利用应用程序框架中的以下几个部分：
Activity （活动）
Broadcast Intent Receiver （广播意图接收者）
Service （服务）
Content Provider （内容提供者）

● 应用程序（Application）
Android 的应用程序主要是用户界面（User Interface）方面的，通常以 JAVA 程序编写，其中还可以包含各种资源文件（放置在 res 目录中）JAVA 程序及相关资源经过编译后，将生成一个 APK 包。Android 本身提供了主屏

幕（Home），联系人（Contact），电话（Phone），浏览器（Browers）等众多的核心应用。同时应用程序的开发者还可以使用应用程序框架层的 API 实现自己的程序。这也是 Android 开源的巨大潜力的体现

3.【Android-kernel】Android 内核改动

Google 在内核里做了什么改动呢?

有家公司专门比较了标准内核和 android 内核，发现 google 修改了 75 个文件，增加了 88 个文件。该公司还对这些被修改的和新增的文件做了注解。

Goldfish -- 44 Files

kernel/arch/arm/mach-goldfish
kernel/include/asm-arm/arch-goldfish

Android 模拟器运行了一个被 google 叫做“金鱼"的虚拟 CPU.金鱼运行 arm926t 指令集（arm926t 是属于 armv5 架构）；并且仿真了输入输出：比如键盘输入和 LCD 输出。这个模拟器其实是在 qemu 之上开发的，输入输出基于 libSDL.

内核里这个 Goldfish 接口实现了这个虚拟“金鱼”CPU 的一些接口，如果想在真实设备上运行 android,这些接口肯定要去掉的。

arm926ej 的介绍见 http://www.arm.com/products/CPUs/ARM926EJ-S.html

YAFFS2 -- 35 Files

kernel/fs/yaffs2

不同于 PC 机，文件是存储在硬盘上的；手机使用 FLASH 作为存储介质。HTC 的 G1 使用 NANDFLASH——这中存储目前已经相当普及了,而且种类也颇多，(SLC,MLC 等等），存储密度也越来越高（已经出现几十 G 大小的 NANDFLASH)，价格也越来越低。

YAFFS2 是专门用在 FLASH 上的文件系统，“YAFFS2”是"Yet Another Flash File System, 2nd edition"的缩写。YAFFS2 为 Linux 内核提供了一个高效访问 NANDFLASH 的接口。但是 NANDFLASH 的支持并不包含在标准的 2.6.25 内核中，所以 Google 在其中添加了对 NANDFLASH 的支持。

蓝牙 -- 10 files

在蓝牙通讯协议栈里 Google 修改了 10 个文件。这些改动解决了一些跟蓝牙耳机相关的明显的 bug，以及一些蓝牙调试和访问控制相关的函数。

调度器 -- 5 files

Android 内核还修改了进程调度和时钟相关策略，这个改动就比较深入了。其目的和效果估计在一段时间后才能找到。

为 android 新增的功能 -- 28 files

除了修正一些 bug 以及其他的改动，android 还增加了一些新的“子系统”，这些系统都比较重要。

IPC Binder

IPC Binder 是一种 IPC(进程间通信）机制。它使得进程能够为其他进程提供服务——还是通过标准的 linux 系统调用 api。IPC Binder 的概念起源于一家叫做“Be.Inc"的公司，在 Google 之前就已经被用到 Palm 软件里去了。

Low Memory Killer

其实内核里已经有一个类似的功能， 叫做"oom killer",就是 out of memory killer,当内存不够的时候，改策略会试图结束一个进程。不知道为什么 Google 重新实现了这个策略。

Ashmem

Ashmem,全程 Anonymous SHared MEMory,翻译成中文就是匿名共享内存。这个功能使得进程间能够共享大块的内存。比如说，系统可以使用 Ashmem 保存一些图标，多个应用程序可以访问这个共享内存来获取图标。

Ashmem 为内核提供了一种回收这些使用完的共享内存块的办法,如果一个进程试图访问这些已经被回收的内存块，它将会得到错误的返回值，以便它重新进行内存块分配和数据初始化。

RAM Console and Log Device

为了调试方便，Android 添加了一个功能，使得调试信息可以输入到一个内存块中。此外，Android 添加了一个独立的日志模块，这样用户空间的进程能够读写日志消息，调试打印信息等。

Android Debug Bridge

嵌入式设备的调试的确比较麻烦。为了便于调试，google 设计了这个调试工具，可以叫做"ADB"，使用 USB 作为连接方式,ADB 可以看作是链接 android 的设备和 PC 机的一套协议。

Android 还添加了其他的东西，比如 real-time clock, switch，timed GPIO。

增加了 Android 的相关 Driver,相应目录为:

kernel/drivers/android

主要分为:

Android IPC 系统: Binder (binder.c)
Android 日志系统: Logger (logger.c)
Android 电源管理: Power (power.c)
Android 闹钟管理: Alarm (alarm.c)
Android 内存控制台: Ram_console (ram_console.c)
Android 时钟控制的 gpio: Timed_gpio (timed_gpio.c)

增加了 switch 处理, 相应的目录为:

kernel/drivers/switch/

增加了一种新的共享内存处理方式, 相应增加的文件为:

kernel/mm/ashmem.c

其他为 Linux-2.6.25 内核所做的补丁等等，例如 BlueTooth, 在此不做详细分析

另外 GoldFish 平台相关的驱动文件如下:

1. 字符输出设备:
   kernel/drivers/char/goldfish_tty.c
2. 图象显示设备: (Frame Buffer)
   kernel/drivers/video/goldfishfb.c
3. 键盘输入设备:
   kernel/drivers/input/keyboard/goldfish_events.c
4. RTC 设备: (Real Time Clock)
   kernel/drivers/rtc/rtc-goldfish.c
5. USB Device 设备:
   kernel/drivers/usb/gadget/android_adb.c
6. SD 卡设备:
   kernel/drivers/mmc/host/goldfish.c
7. FLASH 设备:
   kernel/drivers/mtd/devices/goldfish_nand.c
   kernel/drivers/mtd/devices/goldfish_nand_reg.h
8. LED 设备:
   kernel/drivers/leds/ledtrig-sleep.c
9. 电源设备:
   kernel/drivers/power/goldfish_battery.c
10. 音频设备:

kernel/arch/arm/mach-goldfish/audio.c

11. 电源管理:
    kernel/arch/arm/mach-goldfish/pm.c
12. 时钟管理:
    kernel/arch/arm/mach-goldfish/timer.c

Power Management -- 5 files

电源管理对于移动设备来说相当重要，也是最复杂，开发难度最高的一个功能。Google 添加了一个新的电源管理系统，并没有原先 apm,dpm 等。

其他修改 -- 36 files

除了上述改动之外，还有一些小改动，比如新增的额外的调试功能， 键盘背光控制，TCP 网络管理等等，共涉及 36 个文件。

根据上述，google 对标准的内核做了很大的改动。相比其他的项目，比如 Nokia N810,Openmoko 等项目中，内核的改动仅仅是增加了某个平台的支持。所以移植最快也是最可能的办法是在 google 使用的 kernel 上增加平台支持。

也有一些开发人员将 google 对 2.6.25 内核的改动做成补丁，直接打在自己开发的内核上——当然，自己的内核也应该是 2.6.25，否则会出问题。

模拟器上 kernel 的信息

运行 emulator –show-kernel –avd <your avd name>

```
Uncompressing Linux..... done, booting the kernel.
emulator: emulator window was out of view and was recentred

Linux version 2.6.25-00350-g40fff9a (android-build@apa27.mtv.corp.google.com) (gcc version 4.2.1) #1 Wed Jul 23 18:10:44 PDT 2008
CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00003137
Machine: Goldfish
Memory policy: ECC disabled, Data cache writeback
CPU0: D VIVT write-through cache
CPU0: I cache: 4096 bytes, associativity 4, 32 byte lines, 32 sets
CPU0: D cache: 65536 bytes, associativity 4, 32 byte lines, 512 sets
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 24384
Kernel command line: qemu=1 console=ttyS0 android.checkjni=1 android.qemud=ttyS1 android.ndns=2
Unknown boot option `android.checkjni=1': ignoring
Unknown boot option `android.qemud=ttyS1': ignoring
Unknown boot option `android.ndns=2': ignoring
PID hash table entries: 512 (order: 9, 2048 bytes)
Console: colour dummy device 80x30
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 96MB = 96MB total
Memory: 94268KB available (2380K code, 445K data, 100K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
net_namespace: 152 bytes
android_power_init
android_power_init done
NET: Registered protocol family 16
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 3, 32768 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
TCP: Hash tables configured (established 4096 bind 4096)
TCP reno registered
checking if image is initramfs... it is
Freeing initrd memory: 136K
goldfish_new_pdev goldfish_interrupt_controller at ff000000 irq -1
goldfish_new_pdev goldfish_device_bus at ff001000 irq 1
```

goldfish_new_pdev goldfish_timer at ff003000 irq 3
goldfish_new_pdev goldfish_rtc at ff010000 irq 10
goldfish_new_pdev goldfish_tty at ff002000 irq 4
goldfish_new_pdev goldfish_tty at ff011000 irq 11
goldfish_new_pdev smc91x at ff012000 irq 12
goldfish_new_pdev goldfish_fb at ff013000 irq 13
goldfish_new_pdev goldfish_audio at ff004000 irq 14
goldfish_new_pdev goldfish_memlog at ff006000 irq -1
goldfish_new_pdev goldfish-battery at ff014000 irq 15
goldfish_new_pdev goldfish_events at ff015000 irq 16
goldfish_new_pdev goldfish_nand at ff016000 irq -1
goldfish_new_pdev goldfish-switch at ff017000 irq 17
goldfish_new_pdev goldfish-switch at ff018000 irq 18
goldfish_pdev_worker registered goldfish-switch
goldfish_pdev_worker registered goldfish-switch
goldfish_pdev_worker registered goldfish_nand
goldfish_pdev_worker registered goldfish_events
goldfish_pdev_worker registered goldfish-battery
goldfish_pdev_worker registered goldfish_memlog
goldfish_audio_probe
goldfish_pdev_worker registered goldfish_audio
goldfish_pdev_worker registered goldfish_fb
goldfish_pdev_worker registered smc91x
goldfish_pdev_worker registered goldfish_tty
goldfish_pdev_worker registered goldfish_tty
goldfish_pdev_worker registered goldfish_rtc
goldfish_pdev_worker registered goldfish_timer
goldfish_pdev_worker registered goldfish_device_bus
goldfish_pdev_worker registered goldfish_interrupt_controller
ashmem: initialized
Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
yaffs Jul 23 2008 18:10:35 Installing.
io scheduler noop registered
io scheduler anticipatory registered (default)
io scheduler deadline registered
io scheduler cfq registered
allocating frame buffer 320 * 480, got ffc10000
Console: switching to colour frame buffer device 40x30
console [ttyS0] enabled
brd: module loaded
loop: module loaded
nbd: registered device at major 43
smc91x.c: v1.1, sep 22 2004 by Nicolas Pitre <nico@cam.org>
No IRQF_TRIGGER set_type function for IRQ 12 (goldfish)
eth0: SMC91C11xFD (rev 1) at c6800000 IRQ 12 [nowait]
eth0: Ethernet addr: 52:54:00:12:34:56
goldfish nand dev0: size 4000000, page 2048, extra 64, erase 131072
goldfish nand dev1: size 4000000, page 2048, extra 64, erase 131072
goldfish nand dev2: size 4000000, page 2048, extra 64, erase 131072
mice: PS/2 mouse device common for all mice
*** events probe ***
events_probe() addr=0xc6804000 irq=16
events_probe() keymap=qwerty2
input: qwerty2 as /class/input/input0
goldfish_rtc goldfish_rtc: rtc core: registered goldfish_rtc as rtc0
logger: created 64K log 'log_main'
logger: created 64K log 'log_events'
logger: created 64K log 'log_radio'
IPv4 over IPv4 tunneling driver
GRE over IPv4 tunneling driver
TCP cubic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
802.1Q VLAN Support v1.8 Ben Greear <greearb@candelatech.com>
All bugs added by David S. Miller <davem@redhat.com>
VFP support v0.3: implementor 41 architecture 1 part 10 variant 9 rev 0
goldfish_rtc goldfish_rtc: setting system clock to 2009-01-02 07:43:01 UTC (1230882181)
Freeing init memory: 100K
init: cannot open '/initlogo.rle'

```
yaffs: dev is 32505856 name is "mtdblock0"
yaffs: passed flags ""
yaffs: Attempting MTD mount on 31.0, "mtdblock0"
yaffs: dev is 32505857 name is "mtdblock1"
yaffs: passed flags ""
yaffs: Attempting MTD mount on 31.1, "mtdblock1"
yaffs: dev is 32505858 name is "mtdblock2"
yaffs: passed flags ""
yaffs: Attempting MTD mount on 31.2, "mtdblock2"
sh: can't access tty; job control turned off
# init: cannot find '/system/bin/playmp3', disabling 'bootsound'
eth0: link up
warning: `rild' uses 32-bit capabilities (legacy support in use)
init: sys_prop: mis-match msg size recieved: -1 expected: 128
```

从启动信息可以看出，其主 CPU 为 ARM926EJ-S,并非 ARM11 CPU, 说明下载的 Emulator 内核并非是针对 G1 手机的，估计只是实现了对 ARM926EJ-S CPU 的模拟。