

已经有一些的文章介绍 Android 内核了，本系列篇将从 Linux 内核的角度来分析 Android 的内核，希望给初学者提够有用的信息。本章将简单的介绍 Android 内核的全貌，起到一个抛砖引玉的作用。从下一篇开始将详细介绍每一个 Android 内核驱动程序及其作用。

Android 内核是基于 Linux 2.6 内核的（目前最新开发版本是 2.6.31），它是一个增强内核版本，除了修改部分 Bug 外，它提供了用于支持 Android 平台的设备驱动，其核心驱动主要包括：

Android Binder，基于 OpenBinder 框架的一个驱动，用于提供 Android 平台的进程间通讯（IPC，inter-process communication）。

源代码位于 `drivers/staging/android/binder.c`

Android 电源管理（PM），一个基于标准 Linux 电源管理系统的轻量级的 Android 电源管理驱动，针对嵌入式设备做了很多优化。

源代码位于 `kernel/power/earlysuspend.c`

`kernel/power/consoleearlysuspend.c`

`kernel/power/fbearlysuspend.c`

`kernel/power/wakelock.c`

`kernel/power/userwakelock.c`

低内存管理器（Low Memory Killer），相对于 Linux 标准 OOM（Out Of Memory）机制更加灵活，它可以根据需要杀死进程来释放需要的内存。

源代码位于 `drivers/staging/android/lowmemorykiller.c`

匿名共享内存（ashmem），为进程间提供大块共享内存，同时为内核提供回收和管理这个内存的机制。

源代码位于 `mm/ashmem.c`

Android PMEM（Physical），PMEM 用于向用户空间提供连续的物理内存区域，DSP 和某些设备只能工作在连续的物理内存上。

源代码位于 `drivers/misc/pmem.c`

Android Logger，一个轻量级的日志设备，用于抓取 Android 系统的各种日志。

源代码位于 `drivers/staging/android/logger.c`

Android Alarm，提供了一个定时器用于把设备从睡眠状态唤醒，同时它也提供了一个即使在设备睡眠时也会运行的时钟基准，

源代码位于 `drivers/rtc/alarm.c`

USB Gadget 驱动，一个基于标准 Linux USB gadget 驱动框架的设备驱动，Android 的 USB 驱动是基于 gadget 框架的，

源代码位于 `drivers/usb/gadget/`

Android Ram Console，为了提供调试功能，Android 允许将调试日志信息写入一个被称为 RAM Console 的设备里，它是一个基于 RAM 的 Buffer。

源代码位于 drivers/staging/android/ram_console.c。

Android timed device 提供了对设备进行定时控制功能，目前支持 vibrator 和 LED 设备。

源代码位于 drivers/staging/android/timed_output.c(timed_gpio.c)。

Yaffs2 文件系统，Android 采用 Yaffs2 作为 MTD nand flash 文件系统，源代码位于 fs/yaffs2/目录下。Yaffs2 是一个快速稳定的应用于 NAND 和 NOR Flash 的跨平台的嵌入式设备文件系统，同其他 Flash 文件系统相比，Yaffs2 使用更小的内存来保存他的运行状态，因此它占用内存小；Yaffs2 的垃圾回收非常简单而且快速，因此能达到更好的性能；Yaffs2 在大容量的 NAND Flash 上性能表现尤为明显，非常适合大容量的 Flash 存储。

Android 内核添加或修改的文件很多，下面的列表描述了 Android Emulator 内核的文件：

- drivers/misc/kernel_debugger.c
- drivers/misc/pmem.c
- drivers/misc/qemutrace/qemu_trace_sysfs.c
- drivers/misc/qemutrace/qemu_trace.c
- drivers/misc/qemutrace/qemu_trace.h
- drivers/misc/uid_stat.c
- drivers/staging/android/lowmemorykiller.c
- drivers/staging/android/logger.c
- drivers/staging/android/timed_output.h
- drivers/staging/android/ram_console.c
- drivers/staging/android/timed_gpio.c
- drivers/staging/android/logger.h
- drivers/staging/android/binder.h
- drivers/staging/android/binder.c
- drivers/staging/android/timed_output.c
- drivers/staging/android/timed_gpio.h
- drivers rtc/alarm.c
- drivers/rtc/rtc-goldfish.c
- drivers/net/pppolac.c
- drivers/net/ppp_mppe.c
- drivers/net/pppopns.c
- drivers/video/goldfishfb.c
- drivers/switch/switch_class.c
- drivers/switch/switch_gpio.c
- drivers/char/dcc_tty.c
- drivers/char/goldfish_tty.c

drivers/watchdog/i6300esb.c
drivers/input/misc/gpio_event.c
drivers/input/misc/gpio_input.c
drivers/input/misc/gpio_output.c
drivers/input/misc/keychord.c
drivers/input/misc/gpio_axis.c
drivers/input/misc/gpio_matrix.c
drivers/input/keyreset.c
drivers/input/keyboard/goldfish_events.c
drivers/input/touchscreen/synaptics_i2c_rmi.c
drivers/usb/gadget/android.c
drivers/usb/gadget/f_adb.h
drivers/usb/gadget/f_mass_storage.h
drivers/usb/gadget/f_adb.c
drivers/usb/gadget/f_mass_storage.c
drivers/mmc/host/goldfish.c
drivers/power/goldfish_battery.c
drivers/leds/ledtrig-sleep.c
drivers/mtd/devices/goldfish_nand_reg.h
drivers/mtd/devices/goldfish_nand.c
kernel/power/earlysuspend.c
kernel/power/consoleearlysuspend.c
kernel/power/fbearlysuspend.c
kernel/power/wakelock.c
kernel/power/userwakelock.c
kernel/cpuset.c
kernel/cgroup_debug.c
kernel/cgroup.c
mm/ashmem.c
include/linux/ashmem.h
include/linux/switch.h
include/linux/keychord.h
include/linux/earlysuspend.h
include/linux/android_aid.h
include/linux/uid_stat.h
include/linux/if_pppolac.h
include/linux/usb/android.h
include/linux/wifi_tiwlan.h
include/linux/android_alarm.h
include/linux/keyreset.h
include/linux/synaptics_i2c_rmi.h
include/linux/android_pmem.h
include/linux/kernel_debugger.h
include/linux/gpio_event.h

```
include/linux/wakelock.h
include/linux/if_pppops.h
net/ipv4/sysfs_net_ipv4.c
net/ipv4/af_inet.c
net/ipv6/af_inet6.c
net/bluetooth/af_bluetooth.c
security/commoncap.c
fs/proc/base.c
```

Android 内核配置

Android 是基于 Linux 的，对于一个新的设备，我们首先要编译一个支持 Android 的 Kernel，那么如何使你的 Kernel Android 化呢？除了要移植前面提到的驱动之外，就是如何配置你的 Kernel 来支持 Android 平台，这可以参考 Goldfish 的内核配置文件 - arch/arm/configs/goldfish_defconfig。

一般来说，我们会基于一个平台标准内核配置选项来配置 Android 内核，你可以根据具体的硬件平台来选择 Android 内核的配置选项，可以参考下面的 Android 内核配置列表：

一般需要支持的内核选项

```
ANDROID_PARANOID_NETWORK
ASHMEM
CONFIG_FB_MODE_HELPERS
CONFIG_FONT_8x16
CONFIG_FONT_8x8
CONFIG_YAFFS_SHORT_NAMES_IN_RAM
DAB
EARLYSUSPEND
FB
FB_CFB_COPYAREA
FB_CFB_FILLRECT
FB_CFB_IMAGEBLIT
FB_DEFERRED_IO
FB_TILEBLITTING
HIGH_RES_TIMERS
INOTIFY
INOTIFY_USER
INPUT_EVDEV
INPUT_GPIO
INPUT_MISC
LEDS_CLASS
LEDS_GPIO
LOCK_KERNEL
LkLOGGER
```

LOW_MEMORY_KILLER
MISC_DEVICES
NEW_LEDS
NO_HZ
POWER_SUPPLY
PREEMPT
RAMFS
RTC_CLASS
RTC_LIB
SWITCH
SWITCH_GPIO
TMPFS
UID_STAT
UID16
USB_FUNCTION
USB_FUNCTION_ADB
USER_WAKELOCK
VIDEO_OUTPUT_CONTROL
WAKELOCK
YAFFS_AUTO_YAFFS2
YAFFS_FS
YAFFS_YAFFS1
YAFFS_YAFFS2
但是 Android 推荐不要支持下面两个功能：
CONFIG_YAFFS_DISABLE_LAZY_LOAD
DNOTIFY

配置好后，就可以用 Toolchain 来编译内核了。编译内核比较简单，以 Emulator 的 kernel 为例：

- git clone git://android.kernel.org/kernel/common.git kernel-emulator
- cd kernel-emulator
- export ARCH=arm
- export CROSS_COMPILE=arm-eabi-
- export PATH=<CROSS_COMPILE_TOOLCHAIN_PATH_DIR>PATH
- make goldfish_defconfig
- make