Android Kernel Compile And Run

- 系统环境: Ubuntu 9.10, Kernel 2.6.31-14-generic, i686
- Date： 12/16/2009

准备工作

- 下载Android内核源码及交叉工具链

  打开终端，进入用户主目录($ cd ~)并输入

  ```
  $ git clone git://android.git.kernel.org/kernel/common.git
  ```

  下载Android内核源码。下载完成后，Android内核源码将位于~/common目录下；然后，输入

  ```
  $ git clone git://android.git.kernel.org/kernel/platform/prebuilt.git
  ```

  下载交叉编译工具链。下载完成后，交叉编译工具链将位于~/prebuilt目录下。(cupcake版本的Android源码已经包含交叉工具链，位于源码根目录的prebuilt目录下。如果下载了cupcake版本的Android，此步可以省略)

- Build SDK

  进入Android源码根目录，输入$ make sdk即可。本机为：

  ```
  $ cd ~/yangdroid
  $ make sdk
  ```

  It takes 50 minutes or so..... 编译完成后，本机sdk位于"/home/yang/yangdroid/out/host/linux-x86/sdk/android-sdk_eng.yang_linux-x86"目录下。

提取内核配置文件

为简单起见，选用Android模拟器中的默认内核配置文件。打开终端，输入

```
$ export ANDROID_JAVA_HOME=$JAVA_HOME
$ export ANDROID_PRODUCT_OUT=/home/yang/yangdroid/out/target/product/generic
/* PATH for emulator and adb tool(这行是注释，别输终端里了) */
$ export PATH=$PATH:/home/yang/yangdroid/out/host/linux-x86/bin
$ emulator &
```

在后台运行Android模拟器；然后在该终端输入

```
$ adb pull /proc/config.gz .
```

从模拟器下载配置文件的压缩版本至当前目录。此时可能出现device offline错误，多试几次即可。如下所示：

```
yang@FallenAngel:~$ adb pull /proc/config.gz .
error: device offline
yang@FallenAngel:~$ adb pull /proc/config.gz .
error: device offline
yang@FallenAngel:~$ adb pull /proc/config.gz .
85 KB/s (6913 bytes in 0.079s)
```

此时，可以关闭模拟器。解压config.gz，然后将解压出来的config文件重命名为.config并移动至~/common目录(Android内核源码根目录)

```
$ mv config ~/common/.config
```

编译Android内核

进入Android内核源码根目录

```
$ cd ~/yangdroid
```

输入

```
$ git checkout origin/android-goldfish-2.6.27 -b goldfish
```

将当前内核转换为goldfish分支，以使编译出的内核可以在模拟器上运行。转换后输出如下：

```
yang@FallenAngel:~/common$ git checkout origin/android-goldfish-2.6.27 -b goldfish
Branch goldfish set up to track remote branch android-goldfish-2.6.27 from origin.
Switched to a new branch 'goldfish'
```

输入

```
$ export PATH=$PATH:/home/yang/prebuilt/linux-x86/toolchain/arm-eabi-4.4.0/bin
```

添加交叉编译工具路径至PATH环境变量。输入

添加交叉编译工具路径至PATH环境变量。输入

```
$ make ARCH=arm CROSS_COMPILE=arm-eabi- menuconfig
```

配置Android内核。在内核配置界面中选择"Load an Alternate Configuration File"，然后输入上一步移进~/common目录的.config内核配置文件。保存并退出。输入

```
$ make ARCH=arm CROSS_COMPILE=arm-eabi-
```

编译Android内核。这个编译得比较快，大概10分钟。默认配置文件里只保留了很小一部分的内核特性。编译完成后将在"common/arch/arm/boot"目录下生成内核镜像zImage，这个就是要用模拟器加载并运行的内核。

建立AVD(Android Virtual Devices)

- for more information about AVD, refer to Android Developers

打开终端，输入

```
$ export PATH=$PATH:/home/yang/yangdroid/out/host/linux-x86/sdk/android-sdk_eng.yang_linux-x86/tools
```

添加android工具路径至PATH环境变量；输入

```
$ android list
```

可以查看已存在的目标(targets)和AVD，如

```
yang@FallenAngel:~$ android list
Available Android targets:
id: 1 or "android-AOSP"
     Name: Android AOSP (Preview)
     Type: Platform
     API level: AOSP
     Revision: 1
     Skins: WQVGA432, WVGA800, HVGA (default), WVGA854, QVGA, WQVGA400
Available Android Virtual Devices:
```

输入

```
$ android create avd -n yangAVD01 -t 1
```

依照target id为1的目标生成AVD，如下所示

```
yang@FallenAngel:~$ android create avd -n yangAVD01 -t 1
Android AOSP (Preview) is a basic Android platform.
Do you wish to create a custom hardware profile [no]
Created AVD 'yangAVD01' based on Android AOSP (Preview), with the following hardware config:
hw.lcd.density=160
```

使用$ android list avd可以查看AVD的信息，如下：

```
yang@FallenAngel:~$ android list avd
Available Android Virtual Devices:
    Name: yangAVD01
    Path: /home/yang/.android/avd/yangAVD01.avd
  Target: Android AOSP (Preview) (API level AOSP)
    Skin: HVGA
```

关于AVD是什么，有什么作用？去AVD的目录(比如"/home/yang/.android/avd/yangAVD01.avd")，看看里面文件的内容就可以知道了。

使用编译的内核运行Android

打开终端，输入

```
/* your sdk root directory path */
$ export ANDROID_SDK_ROOT=/home/yang/yangdroid/out/host/linux-x86/sdk/android-sdk_eng.yang_linux-x86
/* path for Android emulator */
$ export PATH=$PATH:/home/yang/yangdroid/out/host/linux-x86/bin
```

最后，输入

```
$ emulator -kernel /home/yang/common/arch/arm/boot/zImage @yangAVD01 &
```

即可使用刚刚编译的Android内核运行Android操作系统。(这里是在后台运行Android)

系统信息查看

在上一步将Android放入后台运行的终端里输入

```
$ adb shell
```

即可使用终端操作Android的文件系统，其根目录信息如下

```
# ls -l
drwxrwxrwt root     root              1970-01-01 08:00 sqlite_stmt_journals
dr-x------ root     root              1970-01-01 08:00 config
drwxrwx--- system   cache             1970-01-01 08:00 cache
d--------- system   system            1970-01-01 08:00 sdcard
lrwxrwxrwx root     root              1970-01-01 08:00 d -> /sys/kernel/debug
lrwxrwxrwx root     root              1970-01-01 08:00 etc -> /system/etc
drwxr-xr-x root     root              2009-12-14 00:05 system
drwxr-xr-x root     root              1970-01-01 08:00 sys
drwxr-x--- root     root              1970-01-01 08:00 sbin
dr-xr-xr-x root     root              1970-01-01 08:00 proc
-rwxr-x--- root     root        12383 1970-01-01 08:00 init.rc
-rwxr-x--- root     root         1677 1970-01-01 08:00 init.goldfish.rc
-rwxr-x--- root     root       103164 1970-01-01 08:00 init
-rw-r--r-- root     root          118 1970-01-01 08:00 default.prop
drwxrwx--x system   system            1970-01-01 08:00 data
drwx------ root     root              1970-01-01 08:00 root
drwxr-xr-x root     root              1970-01-01 08:00 dev
```

输入

```
# cat /proc/version
```

有

```
# cat /proc/version
Linux version 2.6.27-00110-g132305e (yang@FallenAngel) (gcc version 4.4.0 (GCC) ) #1 Wed Dec 16 14:28:39 CST 2009
```

上为系统相关信息：内核版本、编译内核的主机和用户名、编译器版本和编译时间等。 输入

```
# cat /proc/cpuinfo
```

有

```
# cat /proc/cpuinfo
Processor       : ARM926EJ-S rev 5 (v5l)
BogoMIPS        : 246.57
Features        : swp half thumb fastmult vfp edsp java
CPU implementer : 0x41
CPU architecture: 5TEJ
CPU variant     : 0x0
CPU part        : 0x926
CPU revision    : 5
Cache type      : write-through
Cache clean     : not required
Cache lockdown  : not supported
Cache format    : Harvard
I size          : 4096
I assoc         : 4
I line length   : 32
I sets          : 32
D size          : 65536
D assoc         : 4
D line length   : 32
D sets          : 512

Hardware        : Goldfish
Revision        : 0000
Serial          : 0000000000000000
```

上为模拟器模拟的CPU信息，与项目的目标CPU还是有些区别的。更多的就不贴上来了，呵呵:-)

END