密级状态：绝密(　　)　　秘密(　)　　内部资料(　)　　公开(√)

RK1108_CVR 工程调试方法

文件状态：	文件标识：	RK1108_CVR 工程调试方法
[　] 草稿	当前版本：	V0.2
[√] 正式发布	作　　者：	廖华平
[　] 正在修改	完成日期：	2016-6-28

版 本 历 史

版本号	作者	修改日期	修改说明
V0.1	廖华平	2016/6/28	添加工程调试方法
V0.2	廖华平	2016/7/11	添加 adb 功能

目 录

1、 内存错误时调用堆栈

1) **编译环境：**更新服务器上最新的交叉编译工具，确保其中包含了“execinfo.h”头文件。
2) **参考代码：**库代码存于工程目录的 src/external/librk_backtrace 目录的 rk_backtrace.c
3) **信号注册：**在代码main函数最开始，需要注册dump信号，直接调用“init_dump()”注册多个信号处理函数（使用前要用“externl void init_dump();”先声明）。注意编译的时候，必须加上“-g -rdynamic -ldl -funwind-tables”指令，否则无法导出堆栈，并且链接上库“-lrk_backtrace”。
4) **错误解析：**如下图所示，当出现内存错误的时候便会出现，便会打印出错时相关的堆栈信息。这里可以看出出错时的机器码为 0x89e0。在本机找到相关的应用程序或库(注意 symbol 不能被移除，目标板上 symbol 是被移除掉了的)，使用命令“addr2line -e test 0x89e0”便可找到出错时的代码。

```
[root@arm-linux]#./sbin/test

 ***** start dump *****
Obtained 3 stack frames. The signo = 11

/sbin/test(stack_dump+0x30) [0x8874]
/lib/libc.so.1(__default_sa_restorer+0) [0xb6ebcfc4]
/sbin/test(main+0x20) [0x89e0]

 ***** end dump *****
```

2、 ADB 使用方法

1) **开启 ADB：**在系统目录下的“config.sh”配置文件中，将“enable_adb”配置为 yes。那么在系统起来的时候，可以直接运行“source /etc/init.d/run_adbd.sh”来启动 adb；或者修改脚本“out\root\etc\init.d\rcS”，去掉注释“#source /etc/init.d/run_adbd.sh”，那么 adb 就会开机运行。
   注意：adb 不能和 cvr 工程自带的 camera app 同时运行，不然会有冲突。
2) **配置 UBUNTU：**参考附录 A UBUNTU 下 adb 设备的配置。
3) 目前支持 adb shell、adb push、adb pull 等命令。

附录 A UBUNTU 下 ADB 设备的配置

在 Ubuntu14.04 下连接设备后并在机器上运行 adb 后，显示：device not found，解决方法如下：

1、 首先进入用户目录

```
cd  ~
ls –la
```

可以看到一个.android 目录

```
cd .android
```

应该有一个 adb_usb.ini 文件，没有则使用如下命令创建：

```
touch adb_usb.ini
```

2、 用 lsusb 看连接设备的 VID，先连接上设备，运行命令：

```
test@test:$lsusb
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 009: ID 0480:a200 Toshiba America Info. Systems, Inc.
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 005: ID 0461:4e35 Primax Electronics, Ltd
Bus 003 Device 004: ID 2207:0006
Bus 003 Device 002: ID 1a40:0101 Terminus Technology Inc. 4-Port HUB
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

拔掉设备，再次运行

```
test@test:$lsusb
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 009: ID 0480:a200 Toshiba America Info. Systems, Inc.
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 005: ID 0461:4e35 Primax Electronics, Ltd
Bus 003 Device 002: ID 1a40:0101 Terminus Technology Inc. 4-Port HUB
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

对比发现，Bus 003 Device 004: ID 2207:0006 就是连接上的设备。0x2207 就是 VID，后边的 ox0006 就是 PID，记下这两个值。

3、 打开刚才建立的 adb_usb.ini 文件执行：

```
sudo vim adb_usb.ini
```

最后一行加入 VID，也就是 0x2207。

4、 以 root 用户运行：

```
sudo su
adb shell
```

到此应该就能看到设备了。

5、 如果想以普通用户连接，可以在/etc/udev/rules.d/51-android.rules 里添加 owner 参数。

```
SUBSYSTEM=="usb", ATTR{idVendor}=="2207", ATTR{idProduct}=="0011", MODE="0666",
GROUP="plugdev"
```

保存退出，即可以用普通用户运行 adb shell 来连接了。