

V3s 项目

CamDroid 介绍及编译系统
/ V1.0

文档履历

| 版本号 | 日期 | 制/修订人 | 内容描述 |
|------|-----------|-------|------|
| V1.0 | 2015-03-9 | | 正式版本 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

confidential

目 录

| | |
|---------------------------------|---|
| 1. CamDroid 介绍..... | 3 |
| 2. CamLinux 主要目录结构..... | 4 |
| 3. CamDroid 编译..... | 5 |
| 4. 移植运行在 Android 上的库或可执行文件..... | 6 |
| 5. 重新写 mk 文件编译新库或可执行文件..... | 7 |

confidential

1. CamDroid 介绍

CamDroid 由 Android 系统经过裁剪，保留了 Native 层的运行机制，并最大程度减少体积而形成的高度可定制的 Android 精简系统。其目的是为了最大程度复用 Android 成果。提升开发效率。

主要特点：

（1）基本系统（kernel+ramdisk+rootfs）小于 4M，其中含 libc, libm, 属性, binder, servicemanager, init.rc, busybox, 调试系统, utils 库, netlink 框架, vold, crypto, ssl, 时区等。

（2）保留并扩展了 Android 编译系统。在 Android 上能运行的库，可执行文件和框架，无需做任何改动，即可移植到 CamDroid 平台。通过产品 feature 定义可以灵活的选择所需要的库，从而实现根据方案资源来裁剪固件大小。

（3）支持 NDK 开发，支持第三方 makefile 移植。

confidential

2. CamLinux 主要目录结构

```
|---external
|---Framework
|   |---cmds
|   |---include
|   |---libs
|   |---prebuilts
|---hardware
|---device
|   |---softwinner
|---system
```

external 目录用来存放第三方库。

Framework 目录用来存放自己开发的框架，库和可执行文件。

其中 **cmds** 目录用来存放自己开发的可执行文件。

Libs 目录用来存放自己开发的库文件。

Prebuilts 目录用来存放已经编译成功，不方便透漏源码的库或可执行文件。

Hardware 目录用来存放硬件抽象层相关的代码。

Device 目录下设计的目录和各方案配置相关。

System 目录下存放系统调用的命令。

3. CamDroid 编译

```
cd camdroid
source build/envsetup.sh
lunch      //选择合适的方案
mklichee   //编译 kernel
extract-bsp //提取 kernel 文件
make -j8    //编译
pack        //打包
```

confidential

4. 移植运行在 Android 上的库或可执行文件

备注：只需将原 Android 上的源码拷贝到对应位置，将目录下的 Android.mk 文件改名为 CamLinux.mk 文件即可。

confidential

5. 重新写 mk 文件编译新库或可执行文件

编写 CamLinux.mk 文件的规则与 Android.mk 文件的规则完全一致。

公共规则

- CamLinux.mk 文件首先需要指定 LOCAL_PATH 变量,用于查找源文件。由于一般情况下 CamLinux.mk 和需要编译的源文件在同一目录下,所以定义成如下形式: LOCAL_PATH:=\$(call my-dir)上面的语句的意思是将 LOCAL_PATH 变量定义成本文件所在目录路径。
- CamLinux.mk 中可以定义多个编译模块,每个编译模块都是以 include \$(CLEAR_VARS)开始以 include \$(BUILD_XXX)结束。

include \$(CLEAR_VARS) 由编译系统提供,指定让 GNU MAKEFILE 为你清除除 LOCAL_PATH 以外的所有 LOCAL_XXX 变量,如 LOCAL_MODULE, LOCAL_SRC_FILES, LOCAL_SHARED_LIBRARIES, LOCAL_STATIC_LIBRARIES 等。

include \$(BUILD_STATIC_LIBRARY)表示编译成静态库

include \$(BUILD_SHARED_LIBRARY)表示编译成动态库。

include \$(BUILD_EXECUTABLE)表示编译成可执行程序

- 举例:

编译静态库

```
LOCAL_PATH:= $(call my-dir)

Include $(CLEAR_VARS)

LOCAL_SRC_FILES:=AudioHardwareGeneric.cpp \
                AudioHardwareStub.cpp \
                AudioHardwareInterface.cpp \

LOCAL_SHARED_LIBRARIES:=libcutils libutils libbinder libmedia libhardware_legacy

LOCAL_CFLAGS += -DGENERIC_AUDIO

LOCAL_MODULE:=libaudiointerface

ifdef $(BOARD_HAVE_BLUETOOTH),true)
    LOCAL_SRC_FILES+=A2dpAudioInterface.cpp
    LOCAL_SHARED_LIBRARIES += liba2dp
    LOCAL_CFLAGS += -DWITH_BLUETOOTH -DWITH_A2DP
    LOCAL_C_INCLUDES+=$(call include-path-for,bluez)
endif

include $(BUILD_STATIC_LIBRARY)
```

编译动态库

```
include$(CLEAR_VARS)

LOCAL_SRC_FILES:=AudioFlinger.cpp \
                AudioMixer.cpp \
                AudioResampler.cpp \
                AudioResamplerSinc.cpp \
                AudioResamplerCubic.cpp \
                AudioPolicyService.cpp

LOCAL_SHARED_LIBRARIES :=libcutils libutils libbinder libmedia libhardware_legacy

ifdef$(TARGET_SIMULATOR),true)
    LOCAL_LDLIBS+=-ldl
else
```



```

LOCAL_SHARED_LIBRARIES+=libdl

endif

LOCAL_MODULE:=libaudioflinger

ifeq ($(BOARD_HAVE_BLUETOOTH),true)

    LOCAL_CFLAGS+=-DWITH_BLUETOOTH -DWITH_A2DP

    LOCAL_SHARED_LIBRARIES+=liba2dp

endif

ifeq ($(AUDIO_POLICY_TEST),true)

    LOCAL_CFLAGS+= -DAUDIO_POLICY_TEST

endif

include $(BUILD_SHARED_LIBRARY)

```

编译可执行文件

```

include $(CLEAR_VARS)

LOCAL_SHARED_LIBRARIES := liblog

LOCAL_SRC_FILES := service_manager.c binder.c

LOCAL_MODULE := servicemanager

include $(BUILD_EXECUTABLE)

```

预编译第三方库（不方便公开源码的）

```

include $(CLEAR_VARS)

LOCAL_MODULE := libcvbsdisp.so

LOCAL_MODULE_TAGS := optional eng

LOCAL_MODULE_PATH := $(TARGET_OUT)/lib

LOCAL_SRC_FILES := $(LOCAL_MODULE)

LOCAL_MODULE_CLASS := lib

include $(BUILD_PREBUILT)

```

预编译第三方可执行程序

```

include $(CLEAR_VARS)

LOCAL_MODULE := libcvbsdisp.so

LOCAL_MODULE_TAGS := optional eng

LOCAL_MODULE_PATH := $(TARGET_OUT)/bin

LOCAL_SRC_FILES := $(LOCAL_MODULE)

LOCAL_MODULE_CLASS := EXECUTABLE

include $(BUILD_PREBUILT)

```

预编译一个配置到 etc 目录

```

include $(CLEAR_VARS)

LOCAL_MODULE := zoneinfo.dat

LOCAL_SRC_FILES := $(LOCAL_MODULE)

LOCAL_MODULE_CLASS := ETC

LOCAL_MODULE_TAGS := eng

LOCAL_MODULE_PATH := $(TARGET_OUT)/etc

include $(BUILD_PREBUILT)

```

Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

confidential