目录

[1 内核移植的整体思路概述（内核SIG已完成） 2](#_Toc82595733)

[1.1 Open Harmony内核态层和用户态层 2](#_Toc82595734)

[1.2 获得Open Harmony内核态层的两种方法 3](#_Toc82595735)

[1.3 借助三方Linux内核的快速移植流程简介 4](#_Toc82595736)

[2 整体构建环境的准备（软件所补充） 5](#_Toc82595737)

[2.1 将三方内核纳入Open Harmony编译环境 5](#_Toc82595738)

[2.2 编译器的选择 5](#_Toc82595739)

[3 OpenHarmony内核态基础代码的移植（华为内核团队补充） 5](#_Toc82595740)

[4 OpenHarmony内核态必选特性HDF的移植（华为HDF团队补充） 5](#_Toc82595741)

[5 移植完成后的基本功能测试 7](#_Toc82595742)

[5.1 运行HDF测试用例（华为HDF团队补充） 7](#_Toc82595743)

[5.1.1简介 7](#_Toc82595744)

[5.1.2预置条件 7](#_Toc82595745)

[5.1.3用例测试方法 7](#_Toc82595746)

[5.2 运行基本内核测试用例（华为内核团队补充） 10](#_Toc82595747)

# 内核移植的整体思路概述（内核SIG已完成）

## Open Harmony内核态层和用户态层



为了更好的解释整个内核移植，首先需要介绍一些概念：

我们可以把Open Harmony简单的分为

Open Harmony = Open Harmony 内核态层 + Open Harmony 用户态层

其中Open Harmony 内核层就是上图的紫色部分，可以看到，它主要由内核本身（如Linux Kernel，LiteOS），和一些运行在内核态的一些特性组成，比如HDF等。

而Open Harmony用户态层，在上图，就是紫色之外的部分。可以看到，由下往上看，它主要由系统服务层，框架层，应用层组成。在这儿我们将这三层整体称为“Open Harmony 用户态层”。

为什么这么区分呢？因为我们这篇文章主要是要讨论如何快速的把Open Harmony移植到三方芯片平台上。而Open Harmony的用户态层，整体来说和三方芯片平台的耦合度不高，移植较为方便。而内核态层中的内核本身以及HDF驱动框架等，和三方芯片平台的耦合度较高，是移植的重难点。我们先做这个区分，就是为了先把聚光灯打到我们最需要关注的Open Harmony内核态层上，开始分析和解题。另外说明，本文只包含Linux内核的移植，不包含LiteOS的移植。

## 获得Open Harmony内核态层的两种方法

将OH 内核态层继续分解

OH 内核态层 = OH Linux内核 + OH内核态特性（可选特性或者必选特性，如必选特性HDF，今后的可选特性HMDFS等）

而OH Linux内核 = 标准LTS Linux 内核 + 三方SOC芯片平台代码 + OH内核态基础代码（支撑OH用户态层运行的最基础代码）

因此OH 内核态层 = 标准LTS Linux 内核 + 三方SOC芯片平台代码 + OH内核态基础代码 + OH内核态特性（如HDF）

**三方Linux内核**

**Open Harmony Linux内核**

OH 内核态层 = 标准LTS Linux 内核 + 三方SOC芯片平台代码 + OH内核态基础代码 + OH内核态特性（如HDF）

而将前两项组合，标准LTS Linux 内核 + 三方SOC芯片平台代码，其实就是一个三方Linux内核的基础组成。从上面的推导可以看出，Open Harmony 内核态层其实能够由两种方法得到：

方法一：OH 内核态层 = 三方Linux内核 + OH内核态基础代码 + OH内核态特性（如HDF，今后的HMDFS等）

也就是直接借助三方Linux内核，再加上基础OH内核态基础代码、以及HDF等OH内核态特性。

方法二：OH 内核态层 = OH Linux内核 + OH内核态特性（如HDF，今后的可选特性HMDFS等）

也就是直接采用OH Linux内核，然后再加入OH的其他内核态特性。

方案二中的OH Linux内核，当前所支持的三方芯片平台还不够丰富。为了能够响应三方开发者快速移植Open Harmony到三方芯片平台的要求，下文会着重介绍方法一，即借助三方已有的Linux内核，来快速移植Open Harmony到三方芯片平台上。

## 借助三方Linux内核的快速移植流程简介

本节介绍以上的方案一，即借助三方Linux内核，快速的移植OpenHarmony到三方的开发板。

整个移植流程可以分为三步：

1. 准备整体构建环境，包括将三方内核拷贝到OpenHarmony的整体编译环境下等
2. OpenHarmony内核态基础代码的移植
3. OpenHarmony内核态必选特性（如HDF等）的移植、

详细步骤在接下来的第二、三、四章中分别介绍

# 整体构建环境的准备（软件所补充）

## 将三方内核纳入Open Harmony编译环境

## 编译器的选择

# OpenHarmony内核态基础代码的移植（华为内核团队补充）

可能包含：日志特性

# OpenHarmony内核态必选特性HDF的移植（华为HDF团队补充）

1、全量编译

2、下载三方内核

3、配置环境变量

export PROJ\_ROOT=`pwd`

export PRODUCT\_PATH=vendor/hisilicon/Hi3516DV300

export PATH=`pwd`/prebuilts/clang/ohos/linux-x86\_64/llvm/bin:`pwd`/prebuilts/gcc/linux-x86/arm/gcc-linaro-7.5.0-arm-linux-gnueabi/bin/:$PATH # 配置编译环境

MAKE\_OPTIONES="ARCH=arm CROSS\_COMPILE=arm-linux-gnueabi- CC=clang HOSTCC=clang" # 使用工程项目自带的clang环境

4、打HDF补丁

$PROJ\_ROOT/drivers/adapter/khdf/linux/patch\_hdf.sh $PROJ\_ROOT $PROJ\_ROOT/out/KERNEL\_OBJ/kernel/src\_tmp/linux-rpi3b /\*这儿是以树莓派为例\*/

$PROJ\_ROOT/kernel/linux/patches/linux-4.19/hi3516dv300\_patch/hdf.patch /\*这儿后续需要修改到目录，hdf patch不应放在3516的目录下\*/

5、配置config

用下面命令打开hdf相关配置

make ${MAKE\_OPTIONES} menuconfig #使用自带的默认hi3516dv300\_standard\_defconfig 构建内核

6、编译

make ${MAKE\_OPTIONS} -j33 zImage

# 移植完成后的基本功能测试

## 运行HDF测试用例（华为HDF团队补充）

## 5.1.1简介

​ HDF（Hardware Driver Foundation)自测试用例，用于测试HDF框架和外设的基本功能，提供用户态用例和内核态用例测试，本文主要介绍HDF内核态用例测试方法。

### 5.1.2预置条件

1. 按照Open Harmony标准系统环境配置，[快速入门](https://gitee.com/openharmony/docs/blob/master/zh-cn/device-dev/quick-start/quickstart-standard.md)，完成环境配置和代码下载，并打上HDF patch。
2. 测试前需要在menuconfig里打开测试开关CONFIG\_DRIVERS\_HDF\_TEST=y，代码全量编译通过。
3. 硬件环境以开源板HI3516DV300为例介绍。

### 5.1.3用例测试方法

通过[HDC工具](https://gitee.com/openharmony/developtools_hdc_standard)把用例执行文件推送到设备中，然后执行用例即可，操作步骤如下

* 步骤1 全量编译，并烧录版本
* 步骤2 编译hdf测试用例
* 步骤3 用HDC工具推送测试文件到设备中
* 步骤4 进入设备data/test目录，执行测试文件即可

**全量编译，并烧录版本**

编译命令如下：

./build.sh --product-name Hi3516DV300

**编译hdf测试用例**

编译hdf测试用例命令和文件路径如下：

./build.sh --product-name Hi3516DV300 --build-target hdf\_test

等待编译完成，生成的用例执行文件路径：

\out\ohos-arm-release\tests\unittest\hdf

此目录下的用例文件目录有devmgr、osal、sbuf、config

取用例文件推送到设备的data/test目录

**HDC工具推送文件到Hi3516DV300设备**

1.先在Hi3516DV300设备里新建data/test目录

mkdir -p data/test

2.推送依赖库和测试用吐送到到Hi3516DV300设备下

hdc file send XXX\out\ohos-arm-release\hdf\hdf\libhdf\_test\_common.z.so /system/lib

hdc file send XXX\out\ohos-arm-release\tests\unittest\hdf\config\hdf\_adapter\_uhdf\_test\_config /data/test

hdc file send XXX\out\ohos-arm-release\tests\unittest\hdf\devmgr\DevMgrTest /data/test

hdc file send XXX\out\ohos-arm-release\tests\unittest\hdf\osal\OsalTest /data/test

hdc file send XXX\out\ohos-arm-release\tests\unittest\hdf\sbuf\SbufTest /data/test

**进入设备data/test目录，执行测试文件即可**

进入目录执行测试文件目录data/test

 cd data/test

修改文件执行权限

chmod 777 hdf\_adapter\_uhdf\_test\_config DevMgrTest OsalTest SbufTest

./hdf\_adapter\_uhdf\_test\_config

./DevMgrTest

./OsalTest

./SbufTest

DevMgrTest用例结果显示：

./DevMgrTest

Running main() from gmock\_main.cc

[==========] Running 1 test from 1 test case.

[----------] Global test environment set-up.

[----------] 1 test from DevMgrTest

[ RUN ] DevMgrTest.DriverLoaderTest\_001

[ OK ] DevMgrTest.DriverLoaderTest\_001 (0 ms)

[----------] 1 test from DevMgrTest (0 ms total)

[----------] Global test environment tear-down

Gtest xml output finished

[==========] 1 test from 1 test case ran. (0 ms total)

[ PASSED ] 1 test.

## 运行基本内核测试用例（华为内核团队补充）