openEuler+libck

混合部署方案编译运行指南

文档版本　00B01

发布日期　2023-10-31

版权所有 © 海思技术有限公司2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

、HISILICON、海思和其他海思商标均为海思技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

海思技术有限公司

地址：　上海市青浦区虹桥港路2号101室　邮编：201721

网址：　https://www.hisilicon.com/cn/

客户服务邮箱：　support@hisilicon.com

前言

概述

本文档为使用海思Hi3093芯片的进行混合部署开发的程序员而写，目的是供您在开发过程中查阅混合部署所需环境以及编译步骤。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3093	-

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。

 版权所有 © 海思技术有限公司

符号	说明
须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

日期	版本	修订说明
2023-10-31	00B01	第1次临时版本发布。

 版权所有 © 海思技术有限公司

目 录

前言 i
1 概述 1
1.1 概述 1
2 编译参考 2
2.1 openEuler 编译 2
2.2 统一构建镜像 2
2.3 分模块编译 3
2.3.1 源码和编译环境准备 3
2.3.1.1 源码准备 3
2.3.1.2 编译环境准备 3
2.3.2 编译 Hi3093 SDK 4
2.3.3 mcs 编译 4
2.3.4 uboot 制作 5
2.3.5 uImage 的制作 5
2.3.6 Libck 编译 5
2.3.7 文件系统的制作 6
3 混合部署升级指南 7
3.1 混合部署升级指南 7

 版权所有 © 海思技术有限公司

插图目录

图 3-1 混合部署配置网络示意图 7
图 3-2 文件传输示意图 8
图 3-3 openEuler 侧输出示意图 9

 版权所有 © 海思技术有限公司

1 概述

1.1 概述

混合部署框架提供了以下功能：

- openEuler文件系统的编译制作。
- 支持Libck以及Libck用例编译。
- 支持混合部署中3核使用openEuler，1核使用Libck的场景。

用户可通过以上提供的三个功能实现混合部署框架的方案。

版权所有 © 海思技术有限公司

2 编译参考

2.1 openEuler 编译

参考文档：**容器环境下的快速构建指导 — openEuler Embedded在线文档 1.0.0 文档 (gitee.io)**。

由于openEuler Embedded构建过程需要基于openEuler操作系统，且需要安装较多系统工具和构建工具，为方便开发人员快速搭建构建环境，我们将构建过程所依赖的操作系统和工具封装到一个容器中， 使得开发人员可以快速搭建一个构建环境，进而投入到代码开发中，避免在准备环境阶段消耗大量时间。

2.2 统一构建镜像

编译环境：python 3.4+，OpenSSL 1.1.1f、CMake 3.20.5+。

具体步骤如下：

步骤1 下载release代码包mpu_solution.tgz并解压。

步骤2 将**2.1 openEuler编译**docker容器里面路径/usr1/openeuler/src/yocto-embedded-tools/config/arm64目录下的defconfig-kernel和编译生成的openeuler-image-qemu-aarch64-*.rootfs.cpio.gz、zImage拷贝到服务器docker所在服务器上mpu_solution/custom/open_euler。

步骤3 执行编译脚本。

```
cd mpu_solution/
./build.sh
```

在出现编译提示后，输入“oe_libck“。

⚠ 注意

本解决方案不包括openEuler系统，为提升用户构建效率，一键构建脚本中会通过网络下载openEuler-22.03-LTS-SP1版本，用户可通过修改build.sh指定openEuler版本，openEuler系统部分的网络安全问题由用户负责。

版权所有 © 海思技术有限公司

须知

1. 编译脚本需要添加管理员权限。
2. 打开虚拟串口功能需将src/real_time/libck/include/sre_common.h中的宏定义。
   ```
   #define CONFIG_VIRTUAL_SERIAL 1
   ```

步骤4 编译生成的文件在mpu_solution/build/build_hi3093/output目录下。

- u-boot_rsa_4096.bin：uboot。
- Hi3093_ext4fs_cms.bin：文件系统签名。
- Hi3093_ext4fs.img：文件系统，里面包含uImage和设备树。

----结束

2.3 分模块编译

描述：本章节是**2.2 统一构建镜像**中，模块编译过程。

2.3.1 源码和编译环境准备

2.3.1.1 源码准备

步骤1 下载release代码包mpu_solution.tgz并解压。

步骤2 将**2.1 openEuler编译**docker容器里面路径/usr1/openeuler/src/yocto-embedded-tools/config/arm64目录下的defconfig-kernel和编译生成的openeuler-image-qemu-aarch64-*.rootfs.cpio.gz、zImage拷贝到mpu_solution/custom/open_euler目录下。

----结束

2.3.1.2 编译环境准备

步骤1 从以下路径下载openEuler-22.03-LTS-SP1交叉编译链到~/hi3093_tool目录。

http://121.36.84.172/dailybuild/EBS-openEuler-22.03-LTS-SP1/embedded_img/arm64/aarch64-std/openeuler-glibc-x86_64-openeuler-image-aarch64-qemu-aarch64-toolchain-22.03.sh

步骤2 执行如下命令。

```
sh openeuler-glibc-x86_64-openeuler-image-aarch64-qemu-aarch64-toolchain-22.03.sh -d /home/${USER}/hi3093_tool/toolchain/ -y \
```

步骤3 配置openEuler交叉编译环境，执行如下命令。

```
. /home/${USER}/hi3093_tool/toolchain/environment-setup-aarch64-openeuler-linux
```

----结束

版权所有 © 海思技术有限公司

须知

- 路径前是“点(.) 空格( )”，点和空格都不能少。
- 在分模块编译过程中，编译模块所在服务器的CMake需要在3.20.5版本及以上。

2.3.2 编译 Hi3093 SDK

步骤1 修改编译脚本权限。

```
cd mpu_solution/
chmod u+x -R build
```

步骤2 从社区下载openEuler-22.03-LTS-SP1对应的内核放到mpu_solution/open_source目录下，并将目录名重命令成linux5.10，详细命令如下。

```
cd mpu_solution/open_source
wget --no-check-certificate https://gitee.com/openeuler/kernel/repository/archive/openEuler-22.03-LTS-SP1.zip
unzip openEuler-22.03-LTS-SP1.zip
mv kernel-openEuler-22.03-LTS-SP1/ linux5.10
```

步骤3 配置编译时所需要的环境。

```
cd mpu_solution/build
source build_prepare.sh
source build_exports
```

步骤4 更换内核默认配置。

```
cd mpu_solution
cp custom/open_euler/defconfig-kernel open_source/linux5.10/arch/arm64/configs/defconfig
```

步骤5 打patch。

```
cd mpu_solution/open_source/linux5.10
patch -p1 < ../../src/patches/openEuler/kernel-22.03-lts-sp1-mmc.patch
```

步骤6 编译内核。

```
cd mpu_solution/open_source/linux5.10
make ARCH=arm64 defconfig
make ARCH=arm64 CROSS_COMPILE=aarch64-openeuler-linux-gnu- -j32
```

步骤7 执行如下命令编译SDK。

```
cd mpu_solution/build/build_sdk/
./build_sdk.sh
```

----结束

SDK编译完成后会打包生成两个文件：hi3093sdk.tar.gz，securec.tar.gz存放在mpu_solution/build/output目录下。

2.3.3 mcs 编译

概述：混合关键性系统(MCS, Mixed Criticality System)，实现在一颗片上系统中部署多个OS，同时提供Linux的服务管理能力以及实时OS带来的高实时、高可靠的关键能力。软件架构如下所示：

- mcs_km：提供OpenAMP所需内核模块，支持Client OS启动、专用中断收发、管理保留内存等功能。
- rpmsg_pty_demo：提供OpenAMP用户态程序Linux端样例，支持在Linux上通过pty设备访问Client OS。

版权所有 © 海思技术有限公司

- library：提供OpenAMP样例必需的模块remoteproc、virtio、rpmsg、openamp。

具体编译步骤如下：

步骤1 完成2.2 统一构建镜像的编译后，如果需要重新开shell窗口运行mcs模块，则需要执行如下步骤。

```
source ~/hi3093_tool/toolchain/environment-setup-aarch64-openeuler-linux
cd mpu_solution/build
source build_exports
source build_prepare.sh
```

步骤2 进入build/build_hi3093目录，执行如下命令编译对应openAMP与Libmetal。

```
cd mpu_solution/build/build_hi3093
sh ./build_openamp.sh
```

步骤3 进入build/build_libck目录，执行如下命令。

```
cd mpu_solution/build/build_libck
sh ./build_mcs.sh
```

步骤4 生成文件：mcs_km.ko和rpmsg_main位于mpu_solution/build/build_libck/output/mcs_rpmsg。

----结束

2.3.4 uboot 制作

编译生成u-boot，执行如下命令。

```
cd mpu_solution/build/build_uboot
./build_uboot.sh openeuler
```

命令执行完成后，在mpu_solution/build/output目录下生成u-boot_rsa_4096.bin文件。

2.3.5 uImage 的制作

步骤如下：

步骤1 执行第2.3.4 uboot制作的uboot制作的步骤。

步骤2 执行如下命令。

```
cd mpu_solution/custom/open_euler
./mkimage -A arm -O linux -T kernel -C none -a 0x98280000 -e 0x98280040 -n linux-5.10.0 -d zImage
uImage
```

----结束

2.3.6 Libck 编译

按照2.3.3 mcs编译章节执行后，继续在mpu_solution/build/build_libck下执行如下命令。

```
./build_libck.sh
./build_sample.sh
```

生成的镜像路径在mpu_solution/build/build_libck/output 各自模块目录下。

版权所有 © 海思技术有限公司

2.3.7 文件系统的制作

步骤1 将mpu_solution/custom/open_euler目录下的openeuler-image-qemu-aarch64-*.rootfs.cpio.gz和**2.3.5 uImage的制作**生成的uImage放到mpu_solution/build/output目录下。

```
cd mpu_solution
cp custom/open_euler/openeuler-image-qemu-aarch64-*.rootfs.cpio.gz build/output/
cp custom/open_euler/uImage build/output/
```

步骤2 将mpu_solution/build/output目录下的openeuler-image-qemu-aarch64-*.rootfs.cpio.gz修改为initrd_boot.cpio.gz，命令如下。

```
cd mpu_solution/build/output
mv openeuler-image-qemu-aarch64-*.rootfs.cpio.gz initrd_boot.cpio.gz
```

步骤3 编译dtb。

```
cd mpu_solution/build/build_rtos
dtc -I dts -O dtb -o hi1711_asic.dtb hi3093_mcs_3with1.dts
```

命令执行完成后，在mpu_solution/build/build_rtos生成hi1711_asic.dtb文件。

步骤4 进入mpu_solution/build/build_fs目录下，使用有sudo权限的账号执行命令。

```
cd mpu_solution/build/build_fs
./build_hi3093_ext4.sh oe_libck
```

命令执行完成后，在mpu_solution/build/output目录下生成hi3093_ext4.tar.gz文件（压缩了**Hi3093_ext4fs_cms.bin**和**Hi3093_ext4fs.img**文件），在mpu_solution/build/build_fs/cpio_fs保存了文件系统压缩前的文件及目录结构。

----结束

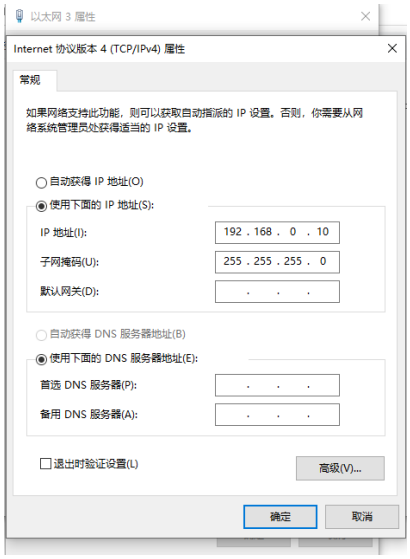
版权所有 © 海思技术有限公司

3 混合部署升级指南

3.1 混合部署升级指南

步骤1 如图3-1更改对应以太网的TCP/IPv4属性 ip地址：192.168.0.10 子网掩码：255.255.255.0。

图 3-1 混合部署配置网络示意图



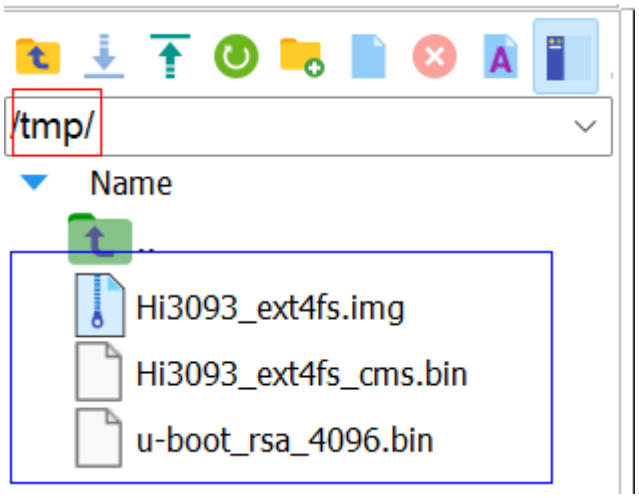
步骤2 成功进入openEuler系统，执行命令。

```
sh /hi3093_init.sh (如果在系统启动阶段已自动执行过此命令，则不用再次执行)
ifconfig eth2 192.168.0.11
```

步骤3 通过MobaXterm建立ssh连接，remote host设为192.168.0.11，把**2.2 统一构建镜像**中制作完成的Hi3093_ext4fs.img、Hi3093_ext4fs_cms.bin和u-boot_rsa_4096.bin通过拖曳方式放在/tmp目录下，如图3-2所示。

版权所有 © 海思技术有限公司

图 3-2 文件传输示意图



步骤4 执行如下命令。

```
cd /tmp
sh /hi3093_upgrade.sh 4096
reboot
```

步骤5 升级成功后，进入系统后，进行初始化和网络配置。

```
sh /hi3093_init.sh (如果在系统启动阶段已自动执行过此命令，则不用再次执行)
ifconfig eth2 192.168.0.11
```

步骤6 进入lib/modules目录，执行如下命令。

```
cd /lib/modules
insmod mcs_km.ko
```

步骤7 将编译好的rpmsg_main、bin文件（如mcs_core3.bin）通过网络传输到单板上的同一文件夹下（排除tmp目录）。

```
执行如下命令：
./rpmsg_main -c 3 -t mcs_core3.bin -a 0x93000000
```

须知

- rpmsg_main位于mpu_solution/build/build_libck/output/mcs_rpsmg中。
- mcs_core3.bin位于mpu_solution/build/build_libck/output/mcs中。
- 其余bin文件也都在mpu_solution/build/build_libck/output对应目录下。

步骤8 执行完成**步骤7**后，在openEuler侧的串口打印hello,Libck!：虚拟串口不会输出任何信息，需要主核在虚拟串口输入信息；如果没有出现**图3-3**打印时，请注释须知部分中只验证普通端点中的代码内容，再重新执行如下命令生成rpmsg_main，再执行**步骤7**。

```
source ~/hi3093_tool/toolchain/environment-setup-aarch64-openeuler-linux
cd mpu_solution/build/build_libck
./build_mcs.sh
```


版权所有 © 海思技术有限公司

图 3-3 openEuler 侧输出示意图

----结束

须知

- 改变rpmsg_main和mcs_core3.bin的执行权限。
- 下电功能仅支持核间通信mcs。
- 如果只需要验证普通端点时，请在mpu_solution/open_source/mcs/rpmsg_pty_libck/rpmsg_main.c下，rpmsg_app_master函数中注释如下代码。

```
int register_endpoint_pty = openamp_create_pty_endpoint(RPMSG_ENDPOINT_PTY, pty_endpoint_cb);
if (register_endpoint_pty < 0) {
    return -1;
}
tick_count = 15000;
while (!openamp_endpoint_ready(RPMSG_ENDPOINT_PTY)) {
    if (tick_count < 0) {
        printf("The %s endpoint isn't ready\n", RPMSG_ENDPOINT_PTY);
         return -1;
       }
       usleep(usleep_master);
       tick_count--;
}
printf("The %s endpoint is ready\n", RPMSG_ENDPOINT_PTY);
```


版权所有 © 海思技术有限公司