

实用操作系统课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 实验七  鸿蒙LiteOS-a内核移植——存储系统移植 |
| 实验日期： | 2023-12-8 |
| 实验地点： | 文宣楼B313 |
|  | |
| 学号： | 33920212204567 |
| 姓名： | 任宇 |
| 专业年级： | 软工2021级 |
| 学年学期： | 2023-2024学年第一学期 |

1. 实验目的

* 进行鸿蒙LiteOS-a内核的存储系统移植，实现：

• 用内存模拟Flash

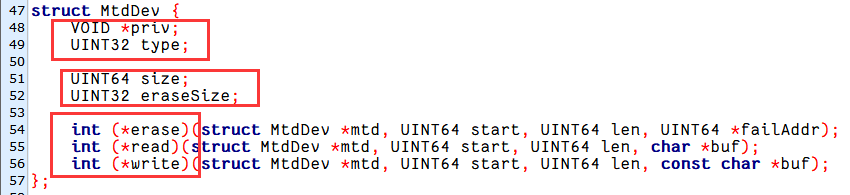
1. 实验内容和步骤

**理论知识学习：**

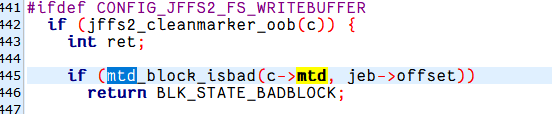
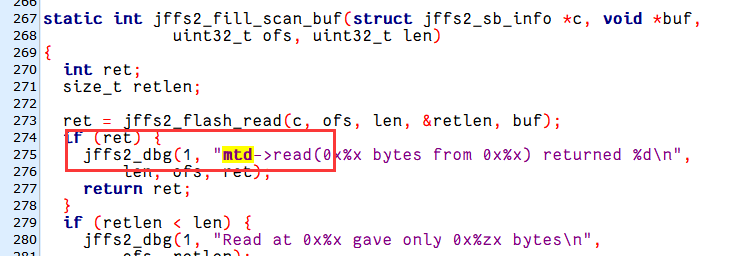
对于嵌入式设备，应用程序保存在Flash上面，内核如果想读取应用程序，首先得有能力读取存储设备。

在进行实践前，首先分析存储设备驱动程序，路径： openharmony/vendor/democom/demochip/driver/mtd/spi\_nor/src/common/spinor.c。

在鸿蒙内核中，存储设备可分为字符设备、块设备和网络设备三类。主要差别在于：字符设备驱动程序里可以读写任意长度的数据，而块设备驱动程序里读写数据以块为单位。无论对于什么设备，都要提供open/read/write/ioctl等函数。存储设备的类型是很多的，这些不同的存储设备访问方法各有不同，但肯定有读/写/擦除这三种操作，因此可以抽象出一个软件层：MTD(Memory Technology Device)，它封装了不同Flash的操作。查找LiteOS-a中相关定义：

这里的定义类似于面向对象设计中的抽象类，不同的Flash要提供它自己的MtdDev结构体，实现其中的函数。JFFS2文件系统直接使用MTD的，没有使用block\_operations，如图所示：

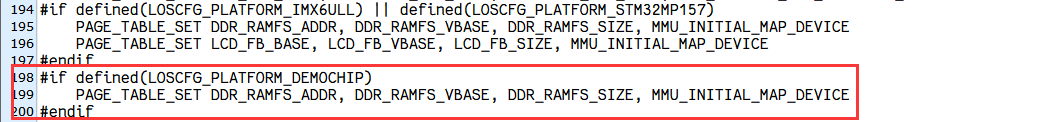


用内存模拟Flash需要指定要使用的内存地址和大小、实现MtdDev结构体。

**实践——用内存模拟Flash：**

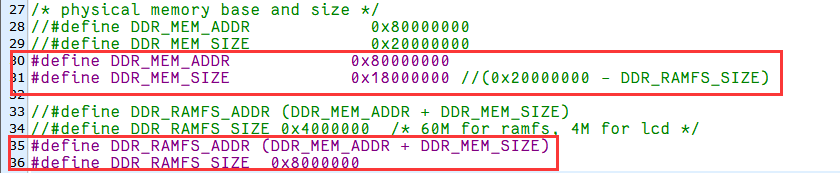
1. 修改 reset\_vector\_up.S 文件：

文件路径为： openharmony/kernel/liteos\_a/arch/arm/arm/src/startup/reset\_vector\_up.S，为 LOSCFG\_PLATFORM\_DEMOCHIP 定义新的内存映射配置：



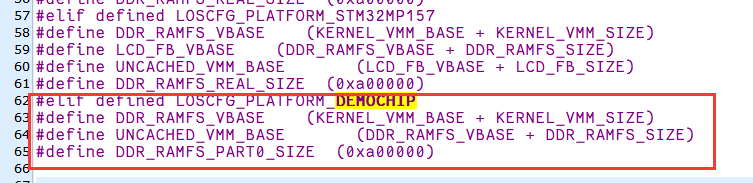
1. 修改board.h文件：

文件路径为：openharmony/vendor/democom/demochip/board/include/board.h，定义 DEMOCHIP 平台的物理内存基址和大小以及 RAM 文件系统的地址和大小，如图所示：



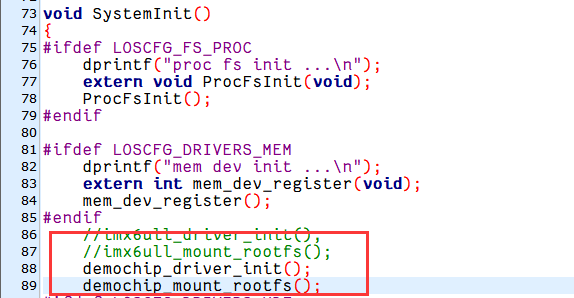
1. 修改 los\_vm\_zone.h 文件：

文件路径为：openharmony/kernel/liteos\_a/kernel/base/include/los\_vm\_zone.h，为DEMOCHIP平台添加虚拟内存的宏定义，仿照其他平台：

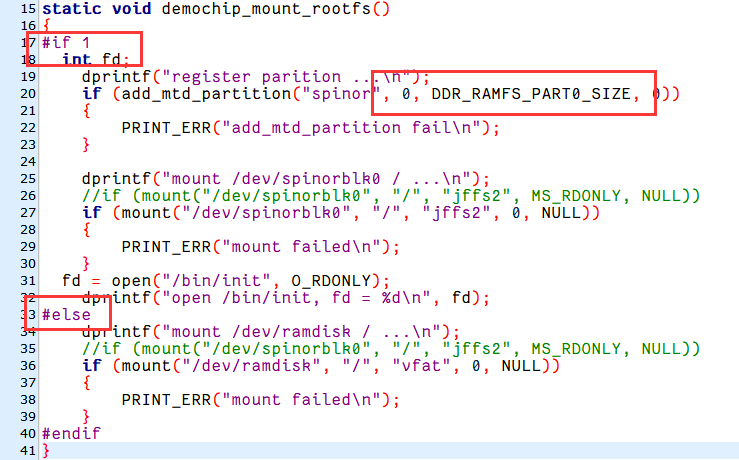


1. 修改board.c文件：

文件路径为：openharmony/vendor/democom/demochip/board/board.c，需要将原有针对 imx6ull 的函数名称更改为适用于 DEMOCHIP 的名称。将 imx6ull\_mount\_rootfs 更改为 demochip\_mount\_rootfs，并将 imx6ull\_driver\_init 更改为 demochip\_driver\_init。

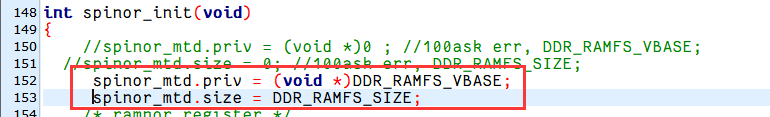


在 demochip\_mount\_rootfs 函数中，根据条件编译 (#if 1) 更改了挂载根文件系统的方式，包括添加 MTD 分区和打开初始化文件。

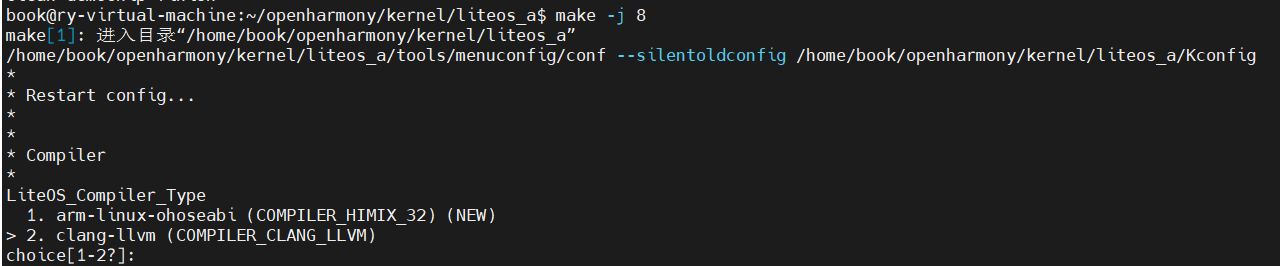


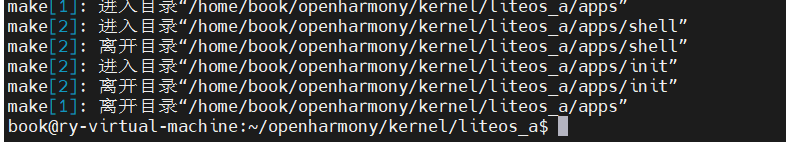
1. 修改spinor.c文件：、

文件路径为：openharmony/vendor/democom/demochip/driver/mtd/spi\_nor/src/common/spinor.c，在 SPINOR 驱动的初始化函数 spinor\_init 中，将 spinor\_mtd.priv 和 spinor\_mtd.size 设置为 DEMOCHIP 平台特定的值：

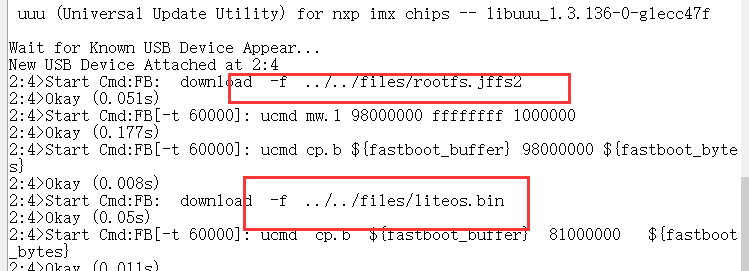


1. 编译并运行：

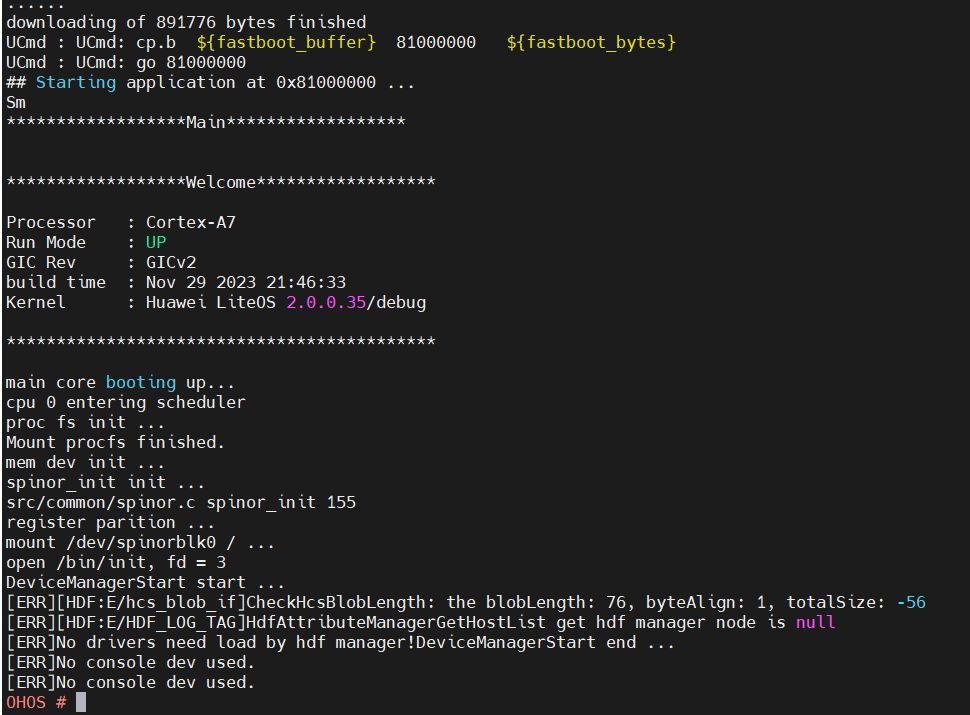


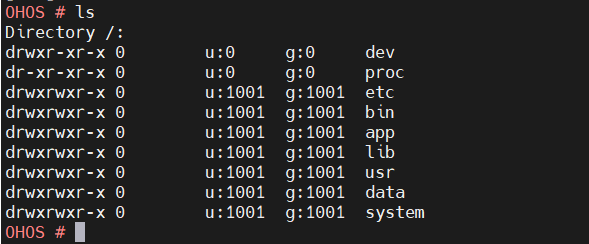


将得到的liteos.bin文件放入烧写工具中运行：



可以正常执行指令，实现使用内存模拟Flash。





1. 实验总结

在本次实验中，我成功实现了鸿蒙LiteOS-a内核的存储系统移植，特别是通过内存模拟Flash的功能。这一过程涉及了对鸿蒙内核存储设备驱动程序的深入理解和修改。通过这次实验，我不仅加深了对鸿蒙操作系统内核存储系统的理解，而且提升了在实际嵌入式环境中进行系统移植和驱动程序开发的能力。实验结果表明，我们能够有效地利用内存来模拟Flash存储，从而为后续的移植提供强有力的基础。

1. 遇到的困难及解决方法

无