

廈門大學



软件学院

《实用操作系统》实验报告

题 目 鸿蒙 LiteOS-a 内核移植——存储系统移植

姓 名 _____ 陈澄 _____

学 号 _____ 32420212202930 _____

班 级 _____ 软工三班 _____

实验时间 _____ 2023/12/6 _____

2023 年 12 月 6 日

1 实验目的

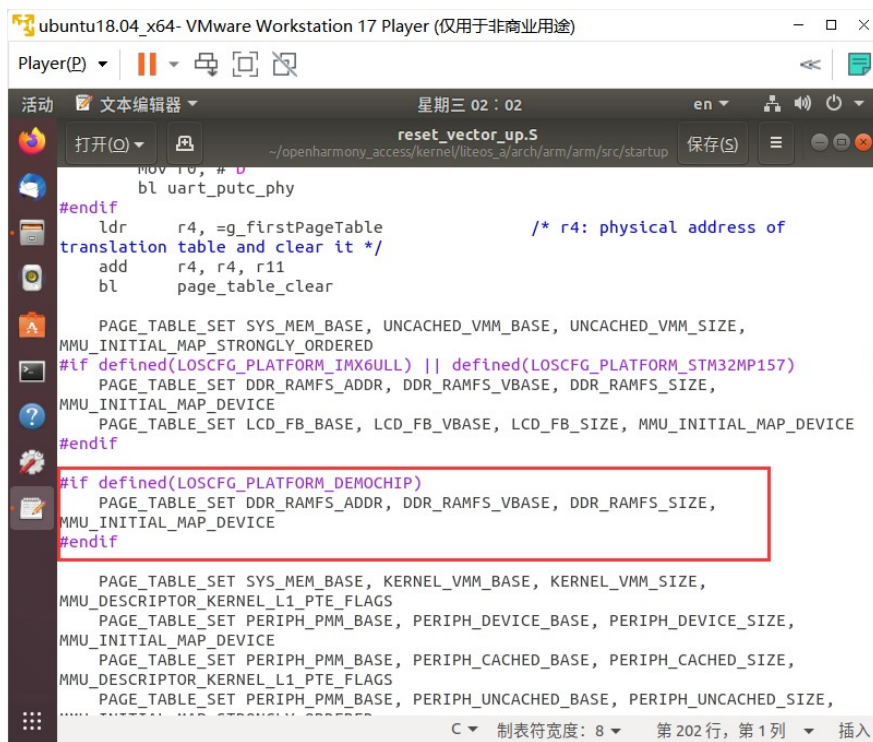
移植鸿蒙 Liteos 的存储系统

使用内存模拟 Flash

2 实验步骤

2.1 修改宏开关

打开 openharmony/kernel/liteos_a/arch/arm/arm/src/startup/reset_vector_up.修改



```
ubuntu18.04_x64- VMware Workstation 17 Player (仅用于非商业用途)
Player(P)  星期三 02:02  en  保存(S)  制表符宽度: 8  第 202 行, 第 1 列  插入

reset_vector_up.S
~/openharmony_access/kernel/liteos_a/arch/arm/arm/src/startup

    mov r0, #0
    bl uart_putc_phy
#endif
    ldr r4, =g_firstPageTable          /* r4: physical address of
translation table and clear it */
    add r4, r4, r11
    bl page_table_clear

    PAGE_TABLE_SET SYS_MEM_BASE, UNCACHED_VMM_BASE, UNCACHED_VMM_SIZE,
MMU_INITIAL_MAP_STRONGLY_ORDERED
    #if defined(LOSCFG_PLATFORM_IMX6ULL) || defined(LOSCFG_PLATFORM_STM32MP157)
    PAGE_TABLE_SET DDR_RAMFS_ADDR, DDR_RAMFS_VBASE, DDR_RAMFS_SIZE,
MMU_INITIAL_MAP_DEVICE
    PAGE_TABLE_SET LCD_FB_BASE, LCD_FB_VBASE, LCD_FB_SIZE, MMU_INITIAL_MAP_DEVICE
    #endif

    #if defined(LOSCFG_PLATFORM_DEMOCCHIP)
    PAGE_TABLE_SET DDR_RAMFS_ADDR, DDR_RAMFS_VBASE, DDR_RAMFS_SIZE,
MMU_INITIAL_MAP_DEVICE
    #endif

    PAGE_TABLE_SET SYS_MEM_BASE, KERNEL_VMM_BASE, KERNEL_VMM_SIZE,
MMU_DESCRIPTOR_KERNEL_L1_PTE_FLAGS
    PAGE_TABLE_SET PERIPH_PMM_BASE, PERIPH_DEVICE_BASE, PERIPH_DEVICE_SIZE,
MMU_INITIAL_MAP_DEVICE
    PAGE_TABLE_SET PERIPH_PMM_BASE, PERIPH_CACHED_BASE, PERIPH_CACHED_SIZE,
MMU_DESCRIPTOR_KERNEL_L1_PTE_FLAGS
    PAGE_TABLE_SET PERIPH_PMM_BASE, PERIPH_UNCACHED_BASE, PERIPH_UNCACHED_SIZE,
```

2.2 修改虚拟地址大小

打开 openharmony/vendor/democom/demochip/board/include/board.h

```
board.h
~/openharmy_access/vendor/democom/demochip/board/include

#ifndef __cplusplus
#ifdef __cplusplus
extern "C" {
#endif /* __cplusplus */
#endif /* __cplusplus */

/* physical memory base and size */
#define DDR_RAMFS_SIZE 0x8000000
#define DDR_MEM_ADDR 0x80000000
#define DDR_MEM_SIZE 0x18000000

#define DDR_RAMFS_ADDR (DDR_MEM_ADDR + DDR_MEM_SIZE)
// #define DDR_RAMFS_SIZE 0x4000000 /* 60M for ramfs, 4M for lcd */

#define LCD_FB_BASE (DDR_RAMFS_ADDR + DDR_RAMFS_SIZE)
#define LCD_FB_SIZE 0x400000 /* 4M */

/* Peripheral register address base and size */
#define PERIPH_PMM_BASE 0x00a00000 // GIC的基地址
#define PERIPH_PMM_SIZE 0x02300000 // 尽可能大一点,以后使用其他外设时就不用映射了

/* GIC base and size : 1M-align */
#define GIC_PHY_BASE 0xA0000000
#define GIC_PHY_SIZE 0xA0100000

#define KERNEL_VADDR_BASE 0x40000000

C/C++/ObjC 头文件 制表符宽度: 8 第 30 行, 第 43 列 插入
```

2.3 为 demochip 添加宏

修改 openharmony/kernel/liteos_a/kernel/base/include/los_vm_zone.h

```
los_vm_zone.h
~/openharmy_access/kernel/liteos_a/kernel/base/include

#define KERNEL_VMM_SIZE KERNEL_VADDR_SIZE

#define KERNEL_VMM_BASE KERNEL_VMM_SIZE
#define KERNEL_VMM_SIZE KERNEL_VMM_SIZE

#ifdef LOSCFG_PLATFORM_IMX6ULL
#define DDR_RAMFS_VBASE (KERNEL_VMM_BASE + KERNEL_VMM_SIZE)
#define LCD_FB_VBASE (DDR_RAMFS_VBASE + DDR_RAMFS_SIZE)
#define UNCACHE_VMM_BASE (LCD_FB_VBASE + LCD_FB_SIZE)
#define DDR_RAMFS_PART0_SIZE (0xa000000)
#elif defined LOSCFG_PLATFORM_STM32MP157
#define DDR_RAMFS_VBASE (KERNEL_VMM_BASE + KERNEL_VMM_SIZE)
#define LCD_FB_VBASE (DDR_RAMFS_VBASE + DDR_RAMFS_SIZE)
#define UNCACHE_VMM_BASE (LCD_FB_VBASE + LCD_FB_SIZE)
#define DDR_RAMFS_PART0_SIZE (0xa000000)
#elif defined LOSCFG_PLATFORM_DEMOCHIP
#define DDR_RAMFS_VBASE (KERNEL_VMM_BASE + KERNEL_VMM_SIZE)
#define UNCACHE_VMM_BASE (DDR_RAMFS_VBASE + DDR_RAMFS_SIZE)
#define DDR_RAMFS_PART0_SIZE (0xa000000)
#else
/* Uncached vmm aspace */
#define UNCACHE_VMM_BASE (KERNEL_VMM_BASE + KERNEL_VMM_SIZE)
#endif
#define UNCACHE_VMM_SIZE DDR_MEM_SIZE

#define VMALLOC_START (UNCACHE_VMM_BASE + UNCACHE_VMM_SIZE)
#define VMALLOC_SIZE 0x08000000

正在保存文件"/home/book/o... C/C++/ObjC 头文件 制表符宽度: 8 第 56 行, 第 24 列 插入
```

2.4 修改 board.c 中的方法名

打开/openharmony_access/vendor/democom/demochip/board/board.c

修改两处方法名及其调用

```
static void demochip_mount_rootfs()
{
    #if 0
        int fd;
        dprintf("register partition ...\n");
        if (add_mtd_partition("spinor", 0, DDR_RAMFS_REAL_SIZE, 0))
        {
            PRINT_ERR("add_mtd_partition fail\n");
        }

        dprintf("mount /dev/spinorblk0 / ...\n");
        //if (mount("/dev/spinorblk0", "/", "jffs2", MS_RDONLY, NULL))
        if (mount("/dev/spinorblk0", "/", "jffs2", 0, NULL))
        {
            PRINT_ERR("mount failed\n");
        }
        fd = open("/bin/init", O_RDONLY);
        dprintf("open /bin/init, fd = %d\n", fd);
    //else
}

static void demochip_driver_init()
{
    #if 0
        extern int my_ramdisk_init(void);
        if (my_ramdisk_init())
        {
            PRINT_ERR("my_ramdisk_init failed\n");
        }
    #else
        extern int spinor_init(void);
        dprintf("spinor_init init ...\n");
        if (!spinor_init())
        {
            PRINT_ERR("spinor_init failed\n");
        }
    #endif
}

#ifdef LOSCFG_DRIVERS_VTDFO

#ifdef LOSCFG_DRIVERS_MEM
    dprintf("mem_dev init ...\n");
    extern int mem_dev_register(void);
    mem_dev_register();
#endif
    demochip_driver_init();
    demochip_mount_rootfs();
}

#ifdef LOSCFG_DRIVERS_HDF
```

2.5 恢复宏的使用

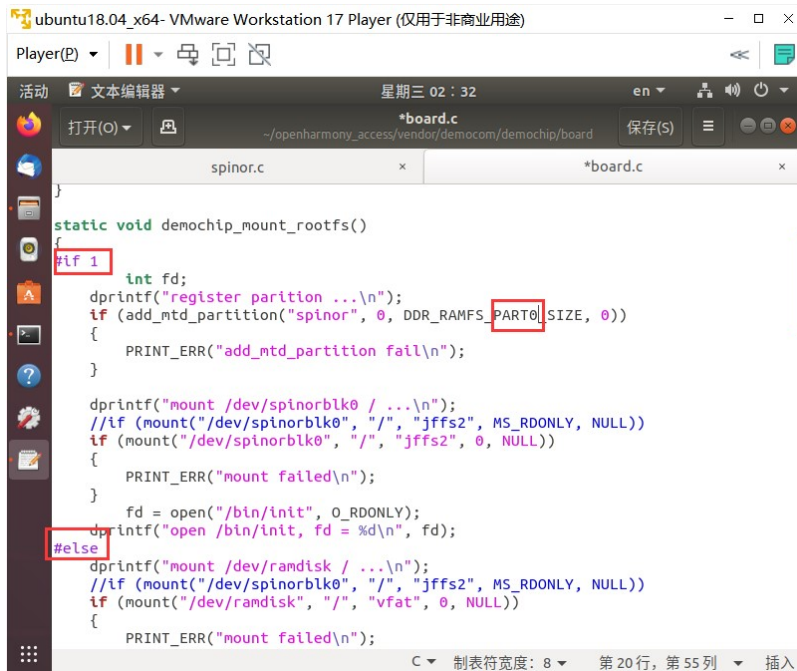
打开 openharmony/vendor/democom/demochip/driver/mtd/spi_nor/src/common/spinor.c

```
int spinor_init(void)
{
    spinor_mtd.priv = (void *)DDR_RAMFS_VBASE;
    spinor_mtd.size = DDR_RAMFS_SIZE;

    /* ramnor register */
    ramnor_register(&spinor_mtd);
    PRINT_RELEASE("%s %s %d\n", __FILE__, __FUNCTION__, __LINE__);
}
```

2.6 恢复 demochip_mount_rootfs()方法

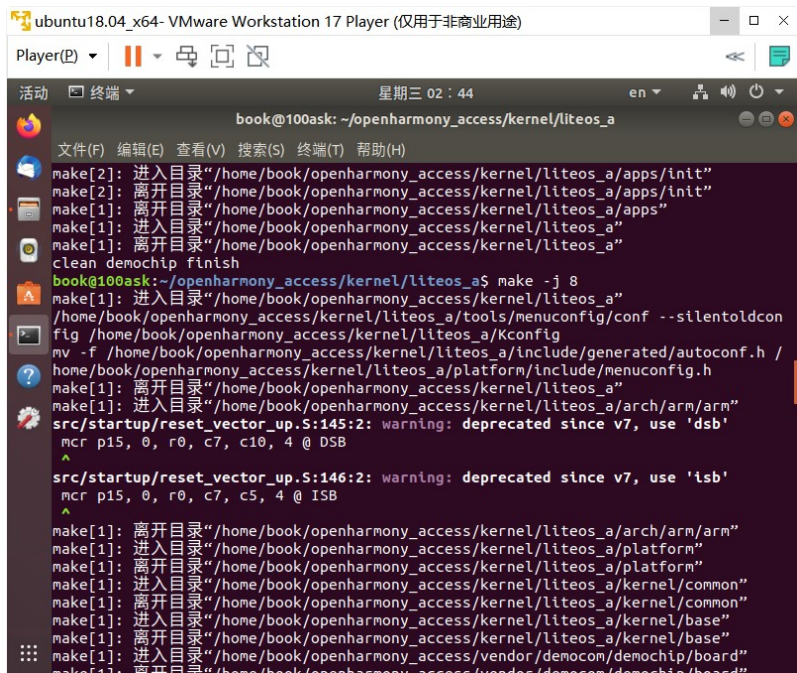
打开/openharmony_access/vendor/democom/demochip/board/board.c



```
static void demochip_mount_rootfs()
{
    if 1
    {
        int fd;
        dprintf("register partition ...\n");
        if (add_mtd_partition("spinor", 0, DDR_RAMFS_PART0, SIZE, 0))
        {
            PRINT_ERR("add_mtd_partition fail\n");
        }

        dprintf("mount /dev/spinorblk0 / ...\n");
        //if (mount("/dev/spinorblk0", "/", "jffs2", MS_RDONLY, NULL))
        if (mount("/dev/spinorblk0", "/", "jffs2", 0, NULL))
        {
            PRINT_ERR("mount failed\n");
        }
        fd = open("/bin/init", O_RDONLY);
        dprintf("open /bin/init, fd = %d\n", fd);
    }
    else
    {
        dprintf("mount /dev/ramdisk / ...\n");
        //if (mount("/dev/spinorblk0", "/", "jffs2", MS_RDONLY, NULL))
        if (mount("/dev/ramdisk", "/", "vfat", 0, NULL))
        {
            PRINT_ERR("mount failed\n");
        }
    }
}
```

2.7 编译



```
book@100ask: ~/openharmy_access/kernel/liteos_a
make[2]: 进入目录 "/home/book/openharmony_access/kernel/liteos_a/apps/init"
make[2]: 离开目录 "/home/book/openharmony_access/kernel/liteos_a/apps/init"
make[1]: 进入目录 "/home/book/openharmony_access/kernel/liteos_a/apps"
make[1]: 离开目录 "/home/book/openharmony_access/kernel/liteos_a/apps"
make[1]: 进入目录 "/home/book/openharmony_access/kernel/liteos_a"
make[1]: 离开目录 "/home/book/openharmony_access/kernel/liteos_a"
clean demochip finish
book@100ask:~/openharmy_access/kernel/liteos_a$ make -j 8
make[1]: 进入目录 "/home/book/openharmony_access/kernel/liteos_a"
/home/book/openharmony_access/kernel/liteos_a/tools/menuconfig/conf --silentoldcon
fig /home/book/openharmony_access/kernel/liteos_a/Kconfig
mv -f /home/book/openharmony_access/kernel/liteos_a/include/generated/autoconf.h /
home/book/openharmony_access/kernel/liteos_a/platform/include/menuconfig.h
make[1]: 离开目录 "/home/book/openharmony_access/kernel/liteos_a"
make[1]: 进入目录 "/home/book/openharmony_access/kernel/liteos_a/arch/arm/arm"
src/startup/reset_vector_up.S:145:2: warning: deprecated since v7, use 'dsb'
mcr p15, 0, r0, c7, c10, 4 @ DSB
src/startup/reset_vector_up.S:146:2: warning: deprecated since v7, use 'isb'
mcr p15, 0, r0, c7, c5, 4 @ ISB
make[1]: 离开目录 "/home/book/openharmony_access/kernel/liteos_a/arch/arm/arm"
make[1]: 进入目录 "/home/book/openharmony_access/kernel/liteos_a/platform"
make[1]: 离开目录 "/home/book/openharmony_access/kernel/liteos_a/platform"
make[1]: 进入目录 "/home/book/openharmony_access/kernel/liteos_a/kernel/common"
make[1]: 离开目录 "/home/book/openharmony_access/kernel/liteos_a/kernel/common"
make[1]: 进入目录 "/home/book/openharmony_access/kernel/liteos_a/kernel/base"
make[1]: 离开目录 "/home/book/openharmony_access/kernel/liteos_a/kernel/base"
make[1]: 进入目录 "/home/book/openharmony_access/vendor/democom/demochip/board"
make[1]: 离开目录 "/home/book/openharmony_access/vendor/democom/demochip/board"
```



```
book@100ask:~/openharmy_access/kernel/liteos_a
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
hi_mci_init -upl011_init -uhinfc620_init -uhisnfc100_init -uregulator_machine_init
-uhisimeidia_regulator_init -ucpufreq_init -uhisilicon_cpufreq_init -ucpufreq_mac
hine_init -udevfreq_init -umedia_devfreq_init -udevfreq_machine_init -uhieth_machi
ne_init -uhigmac_machine_init -umachine_init -Map=/home/book/openharmony_access/k
ernel/liteos_a/out/demochip/liteos.map -o /home/book/openharmony_access/kernel/lit
eos_a/out/demochip/liteos --start-group -lclang_rt.builtins -lunwind --no-depende
nt-libraries -lcortex-a7 -lbsp -lrootfs -lbase -lboard -lmtcommon -lspinor_flash
-luart -lcpup -ldynload -lvds0 -ltickless -lliteipc -lpipes -lc -lsec -lscrew -lc
++ -lc++abi -lcppsupport -lz -lpostx -lbsd -llinuxkpi -lvfs -lmulti_partition -lbc
h -lfat -lvirtpart -ldisk -lbcache -lramfs -lnfs -lproc -ljffs2 -llwip --whole-arch
ive -lhdf -lhdf_config -lhello --no-whole-archive -lhievent -lmem -lmtcommon -lh
ilog -lshell -ltelnet -lsyscall -lsecurity --end-group
/home/book/llvm/bin/.../bin/llvm-objcopy -R .bss -O binary /home/book/openharmony_
access/kernel/liteos_a/out/demochip/liteos /home/book/openharmony_access/kernel/li
teos_a/out/demochip/liteos.bin
/home/book/llvm/bin/.../bin/llvm-objdump -t /home/book/openharmony_access/kernel/l
iteos_a/out/demochip/liteos |sort >/home/book/openharmony_access/kernel/liteos_a/o
ut/demochip/liteos.syn.sorted
/home/book/llvm/bin/.../bin/llvm-objdump -d /home/book/openharmony_access/kernel/l
iteos_a/out/demochip/liteos >/home/book/openharmony_access/kernel/liteos_a/out/dem
ochip/liteos.asm
make[1]: 进入目录"/home/book/openharmony_access/kernel/liteos_a/apps"
make[2]: 进入目录"/home/book/openharmony_access/kernel/liteos_a/apps/shell"
make[2]: 离开目录"/home/book/openharmony_access/kernel/liteos_a/apps/shell"
make[2]: 进入目录"/home/book/openharmony_access/kernel/liteos_a/apps/init"
make[2]: 离开目录"/home/book/openharmony_access/kernel/liteos_a/apps/init"
make[1]: 离开目录"/home/book/openharmony_access/kernel/liteos_a/apps"
book@100ask:~/openharmy_access/kernel/liteos_a$
```

编译成功

2.8 测试运行

成功启动，也可以正常运行

```
COM5 (USB-Enhanced-SERIAL CH9102 (COM5))
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
User sessions
COM5 (USB-Enhanced-SERIAL CH
main core booting up...
cpu 0 entering scheduler
proc fs init ...
Mount procs finished.
mem dev init ...
spinor_init init ...
src/common/spinor.c spinor_init 155
register partition ...
mount /dev/spinorblk0 / ...
open /bin/init, fd = 3
DeviceManagerStart start ...
[ERR][HDF:E/hcs_blob_if]CheckHcsBlobLength: the blobLength: 76, byteAlign: 1, to
talSize: -56
[ERR][HDF:E/HDF_LOG_TAG]HdfAttributeManagerGetHostList get hdf manager node is n
ull
[ERR]No drivers need load by hdf manager!DeviceManagerStart end ...
[ERR]No console dev used.
[ERR]No console dev used.
OHOS # ls
Directory /:
drwxr-xr-x 0      u:0    g:0    dev
dr-xr-xr-x 0      u:0    g:0    proc
drwxrwxr-x 0      u:1001  g:1001  etc
drwxrwxr-x 0      u:1001  g:1001  bin
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

3 实验遇到的问题及其解决方法

无

4 我的体会

通过本次实验，我深入了解了鸿蒙 LiteOS 的存储系统，并学会了如何进行移植和测试。其次，通过使用内存模拟 Flash 的方式，我对嵌入式系统中存储器件的工作原理有了更加直观的认识。在实验中，我深入理解了 Flash 存储器件的特点和工作机制，并学会了如何利用内存来模拟 Flash 的行为。这让我对嵌入式系统的存储管理有了更清晰的认识，也为我以后在实际项目中的应用打下了坚实的基础。