

实验 5 鸿蒙 LiteOS-a 内核移植——串口移植

22920212204396

黄子安

一、实验目的

1、移植串口，使其可以传送一个字符进而传输字符串

二、实验环境

- 1、物理机 Windows11
- 2、虚拟机 Ubuntu18.04
- 3、开发板 Imx6ull

三、实验内容

我们要实现的是 Liteos-a 的最小系统移植，因此不需要初始化串口，u-boot 已经完成了相关的初始化，此外也不需要动态配置串口，固定使用波特率 15200 即可，该操作已经在 u-boot 中完成设置

本次实验视频动作偏快，代码之间的跳转非常多，因此建议在虚拟机中安装一个 vscode 或者在本地物理机使用 ssh 和虚拟机建立连接

首先为了移植看起来比较舒服一些，将文件名从 stm32mp157-uart 修改成 uart，之后将里面的两个文件名也修改下



查看包含这个文件夹的 makefile 文件，可以看到在 demochip 中有三个

```
book@hza-virtual-machine:~/openharmony$ grep "stm32mp157-uart" * -nr
vendor/democom/demochip/board/Makefile:7:LOCAL_FLAGS := -I$(LITEOSTOPDIR)/../../vendor/st/stm32mp157/driver/stm32mp157-uart
vendor/democom/demochip/demochip.mk:18:LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/stm32mp157-uart
vendor/democom/demochip/demochip.mk:19:LITEOS_BASELIB += -lstm32mp157-uart
vendor/st/stm32mp157/board/Makefile:7:LOCAL_FLAGS := -I$(LITEOSTOPDIR)/../../vendor/st/stm32mp157/driver/stm32mp157-uart
vendor/st/stm32mp157/stm32mp157.mk:18:LIB_SUBDIRS += $(STM32MP157_BASE_DIR)/driver/stm32mp157-uart
vendor/st/stm32mp157/stm32mp157.mk:19:LITEOS_BASELIB += -lstm32mp157-uart
```

依次修改这 3 个 makefile 文件，这里视频中出现了一个问题，对于第一个 makefile 文件没有修改完文件目录，还是在 st 目录下没改成 democom，不过视频后续在编译内核的时候发现了这个问题

```
openharmy > vendor > democom > demochip > board > Makefile
1 include $(LITEOSTOPDIR)/config.mk
2
3 MODULE_NAME := $(notdir $(shell pwd))
4
5 LOCAL_SRCS := board.c bsd_board.c
6
7 LOCAL_FLAGS := -I$(LITEOSTOPDIR)/../../vendor/st/stm32mp157/driver/uart
8
9 include $(MODULE)
```

```
los_process.c 4, M Makefile
openharmy > vendor > democom > demochip > driver > uart > Makefile
1 include $(LITEOSTOPDIR)/config.mk
2
3 MODULE_NAME := $(notdir $(shell pwd))
4
5 LOCAL_SRCS := uart_core.c uart_dev.c uart_hardware.c
6
7 include $(MODULE)
8
```

```
openharmy > vendor > democom > demochip > demochip.mk
10 LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/mtd/spi_nor
11 LITEOS_BASELIB += -lspinor_flash
12
13 ifeq ($(LSCFG_DRIVERS_VIDEO), y)
14 LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/stm32mp157-fb
15 LITEOS_BASELIB += -lstm32mp157-fb
16 endif
17
18 LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/uart
19 LITEOS_BASELIB += -luart
20
21
22 LITEOS_MTD_SPI_NOR_INCLUDE += -I$(DEMOCHIP_BASE_DIR)/driver/mtd/common/include \
23                               -I$(DEMOCHIP_BASE_DIR)/driver/mtd/spi
```

之后修改串口的物理基地址，根据相关文件查询可以知道物理基地址为 0x2020000（这里视频里明显韦东山老师手误了，说的 4 个打了 5 个，不过后续也发现）

```
openharmy > vendor > democom > demochip > board > include > asm > platform.h > I2C2_REG_PBASE
79 #define I2C0_REG_PBASE 0x12060000
80 #define I2C1_REG_PBASE 0x12061000
81 #define I2C2_REG_PBASE 0x12062000
82 #define I2C0_REG_BASE IO_DEVICE_ADDR(0x12060000)
83 #define I2C1_REG_BASE IO_DEVICE_ADDR(0x12061000)
84 #define I2C2_REG_BASE IO_DEVICE_ADDR(0x12062000)
85
86 #define UART1_REG_PBASE 0x02020000 /* imx8ull_uart1 */
87
88 #define UART4_REG_BASE IO_DEVICE_ADDR(UART4_REG_PBASE)
```

| | |
|-----------|-----------|
| 0203_4000 | 0203_7FFF |
| 0203_0000 | 0203_3FFF |
| 0202_C000 | 0202_FFFF |
| 0202_8000 | 0202_BFFF |
| 0202_4000 | 0202_7FFF |
| 0202_0000 | 0202_3FFF |
| 0201_C000 | 0201_FFFF |

AIPS-1 (via SPBA)
Glob,Module ENABLE

| | |
|-----------------------------------------|-------|
| ASRC | 16 KB |
| SAI3 | 16 KB |
| SAI2 | 16 KB |
| SAI1 | 16 KB |
| ESAI | 16 KB |
| UART1 | 16 KB |
| Reserved for SDMA internal registers | 16 KB |

之后修改 `uart_add_device` 函数，将原来的串口 4 修改为串口 1

```
openharmy > vendor > democom > demochip > board > C bsd_board.c > UART_ADD_DEVICE(dev, unit)
44     device_set_softc(dev, &uart ## unit ## _softc); \
45     } \
46 } while (0)
47
48 // callback never be null pointer
49 static void uart_add_device(add_res_callback_t callback)
50 {
51     device_t uart_dev;
52     UART_ADD_DEVICE(uart_dev, 0);
53     callback("uart", SYS_RES_MEMORY, 0, UART1_REG_PBASE,
54             UART1_REG_PBASE + UART_IOMEM_COUNT, UART_IOMEM_COUNT);
55     callback("uart", SYS_RES_IRQ, 0, NUM_HAL_INTERRUPT_UART4,
56             NUM_HAL_INTERRUPT_UART4, 1);
57 }
58 }
59 //endif
```

修改串口 1 的中断号，查询文档可以发现这个中断号是 32+26

| | | | |
|----|--------|---|-----------------------------------------|
| 23 | usart2 | - | USART2 Enhanced USART interrupt request |
| 24 | sai3 | - | SAI interrupt |
| 25 | sai3 | - | SAI interrupt |
| 26 | uart1 | - | UART-1 ORed interrupt |
| 27 | uart2 | - | UART-2 ORed interrupt |
| 28 | uart3 | - | UART-3 ORed interrupt |
| 29 | uart4 | - | UART-4 ORed interrupt |
| 30 | uart5 | - | UART-5 ORed interrupt |

```
openharmy > vendor > st > stm32mp157 > board > include > asm > C hal_platform_ints.h > NUM_HAL_INTERRUPT_UART1
61 #define NUM_HAL_INTERRUPT_TIMER0      37
62 #define NUM_HAL_INTERRUPT_TIMER1      37
63 #define NUM_HAL_INTERRUPT_TIMER2      38
64 #define NUM_HAL_INTERRUPT_TIMER3      38
65
66 #define NUM_HAL_INTERRUPT_UART0       58
67 #define NUM_HAL_INTERRUPT_UART1       (32+26)
68 #define NUM_HAL_INTERRUPT_UART2       41
69
70 #define NUM_HAL_INTERRUPT_UART4       84
71
```

```
openharmy > vendor > democom > demochip > board > C bsd_board.c > UART_ADD_DEVICE(dev, unit)
44     device_set_softc(dev, &uart ## unit ## _softc); \
45     } \
46 } while (0)
47
48 // callback never be null pointer
49 static void uart_add_device(add_res_callback_t callback)
50 {
51     device_t uart_dev;
52     UART_ADD_DEVICE(uart_dev, 0);
53     callback("uart", SYS_RES_MEMORY, 0, UART1_REG_PBASE,
54             UART1_REG_PBASE + UART_IOMEM_COUNT, UART_IOMEM_COUNT);
55     callback("uart", SYS_RES_IRQ, 0, NUM_HAL_INTERRUPT_UART1,
56             NUM_HAL_INTERRUPT_UART1, 1);
57 }
58 }
59 //endif
60
61 //if defined (LOSCFG_DRIVERS_MTD_SPT_NOR_SUNXT)
```

将串口 1 的物理地址映射成虚拟地址

```
openharmy > vendor > st > stm32mp157 > board > include > asm > C platform.h > UART1_REG_BASE
83 #define I2C1_REG_BASE IO_DEVICE_ADDR(0x12061000)
84 #define I2C2_REG_BASE IO_DEVICE_ADDR(0x12062000)
85
86 #define UART1_REG_PBASE 0x02020000 /* imx6ull uart1 */
87
88 #define UART4_REG_BASE IO_DEVICE_ADDR(UART4_REG_PBASE)
89
90
91 #define UART0_REG_BASE IO_DEVICE_ADDR(UART0_REG_PBASE)
92 #define UART1_REG_BASE IO_DEVICE_ADDR(UART1_REG_PBASE)
93 #define UART2_REG_BASE IO_DEVICE_ADDR(0x12042000)
94
95 // #define UART0_REG_PBASE 0x12040000
96 #define UART1_REG_PBASE 0x12041000
97 #define UART2_REG_PBASE 0x12042000
98
```

再使用这个宏赋值给串口的基地址

```
openharmy > vendor > democom > demochip > board > include > C uart.h > UART_NUM
78
79
80 #define TTYS0 "/dev/ttyS0"
81
82 #define UART4 0
83 #define CONSOLE_UART UART4
84
85 #define CONSOLE_UART_BAUDRATE 115200
86 #define UART_NUM 4
87
88 #define TTY_DEVICE "/dev/uartdev-0"
89 #define UART_REG_BASE UART1_REG_BASE
90 #define NUM_HAL_INTERRUPT_UART NUM_HAL_INTERRUPT_UART1 /* TODO, 100ask */
91
92
93 #ifdef LOSCFG_PLATFORM_HISI_AMP
94 #undef TTY_DEVICE
95 #define TTY_DEVICE "/dev/virt-tty"
```

删除该部分代码，之后将串口修改成串口 1

```
demochip.mk Makefile ~/stm32mp157/... bsd_board.c 3 uart.h 3 platform.h ~/demochip/... 2 x hal_platform_int
openharmy > vendor > democom > demochip > board > include > asm > C platform.h > ...
94
95 // #define UART0_REG_PBASE 0x12040000
96 #define UART1_REG_PBASE 0x12041000
97 #define UART2_REG_PBASE 0x12042000
98
99 #if (CONSOLE_UART == UART0)
100 #define UART_BASE UART0_REG_BASE
101 #define UART0_INT_NUM NUM_HAL_INTERRUPT_UART0
102 #elif (CONSOLE_UART == UART1)
103 #define UART_BASE UART1_REG_BASE
104 #define UART0_INT_NUM NUM_HAL_INTERRUPT_UART1
105 #elif (CONSOLE_UART == UART2)
106 #define UART_BASE UART2_REG_BASE
107 #define UART0_INT_NUM NUM_HAL_INTERRUPT_UART2
108 #endif
109
110 #define MISC_REG_BASE IO_DEVICE_ADDR(0x12028000)
111 #define SYS_CTRL_REG_BASE IO_DEVICE_ADDR(0x12020000)
```

```
demochip.mk  Makefile ~/stm32mp157/...  bsd_board.c 3  uart.h 3  platform.h ~/.../demochip/... 2  hal...
openharmy > vendor > democom > demochip > board > include > asm > C platform.h > UART2_REG_PBASE
94
95 // #define UART0_REG_PBASE 0x12040000
96 #define UART1_REG_PBASE 0x12041000
97 #define UART2_REG_PBASE 0x12042000
98
99
100 #define UART_BASE UART0_REG_BASE
101 #define UART0_INT_NUM NUM_HAL_INTERRUPT_UART1
102
103 #define MISC_REG_BASE IO_DEVICE_ADDR(0x12028000)
104 #define SYS_CTRL_REG_BASE IO_DEVICE_ADDR(0x12020000)
105 #define CRG_REG_BASE IO_DEVICE_ADDR(0x12010000)
106
107 #define TIMER0_ENABLE BIT(16)
108 #define TIMER1_ENABLE BIT(17)
109 #define TIMER2_ENABLE BIT(18)
110 #define TIMER3_ENABLE BIT(19)
111
```

这样子修改后这里串口的基地址就是串口 1 的基地址，串口的中断号就是串口 1 的中断号

```
openharmy > vendor > democom > demochip > board > include > C uarth > ...
81
82 #define UART4 0
83 #define CONSOLE_UART UART4
84
85 #define CONSOLE_UART_BAUDRATE 115200
86 #define UART_NUM 4
87
88 #define TTY_DEVICE "/dev/uartdev-0"
89 #define UART_REG_BASE UART1_REG_BASE
90 #define NUM_HAL_INTERRUPT_UART NUM_HAL_INTERRUPT_UART1 /* TODO,100ask */
91
92
93 #ifdef LOSCFG_PLATFORM_HISI_AMP
94 #undef TTY_DEVICE
95 #define TTY_DEVICE "/dev/virt-tty"
96 #endif
97
```

使用 imx6ull_uart.h 文件替换整个 uart_hardware.h 文件

```
demochip.mk  Makefile ~/stm32mp157/...  bsd_board.c 3  uart.h 3  imx6ull_uart.h 2  platform.h ~/.../demochip/... 2  hal_platform_ints.h 1  C
openharmy > vendor > nxp > imx6ull > driver > imx6ull-uart > C imx6ull_uart.h
33 #define __AMBA_PL011_UART_H
34
35 #include "los_typedef.h"
36
37 #ifdef __cplusplus
38 #if __cplusplus
39 extern "C" {
40 #endif /* __cplusplus */
41 #endif /* __cplusplus */
42
43
44 /*根据IMX6ULL芯片手册<<55.15>UART-Memory-Map/Register-Definition>>的3608页,定义UART的结构体,*/
45 typedef struct {
46     volatile unsigned int URXD; /*<UART Receiver Register, offset: 0x00> 串口接收寄存器, 偏移地址0x00*/
47     unsigned char RESERVED_0[60];
48     volatile unsigned int UTXD; /*<UART Transmitter Register, offset: 0x40> 串口发送寄存器, 偏移地址0x40*/
49     unsigned char RESERVED_1[60];
50     volatile unsigned int UCR1; /*<UART Control Register 1, offset: 0x80> 串口控制寄存器1, 偏移地址0x80*/
51     volatile unsigned int UCR2; /*<UART Control Register 2, offset: 0x84> 串口控制寄存器2, 偏移地址0x84*/
52     volatile unsigned int UCR3; /*<UART Control Register 3, offset: 0x88> 串口控制寄存器3, 偏移地址0x88*/
53     volatile unsigned int UCR4; /*<UART Control Register 4, offset: 0x8C> 串口控制寄存器4, 偏移地址0x8C*/
54     volatile unsigned int UFCR; /*<UART FIFO Control Register, offset: 0x90> 串口FIFO控制寄存器, 偏移地址0x90*/
55     volatile unsigned int USR1; /*<UART Status Register 1, offset: 0x94> 串口状态寄存器1, 偏移地址0x94*/
56     volatile unsigned int USR2; /*<UART Status Register 2, offset: 0x98> 串口状态寄存器2, 偏移地址0x98*/
57     volatile unsigned int UESC; /*<UART Escape Character Register, offset: 0x9C> 串口转义字符寄存器, 偏移地址0x9C*/

```

```
Makefile ~/stm32mp157/...  C bsd_board.c 3  C uarth 3  C uart_hardware.h X  C platform.h ~/demochip/... 2  C hal_platform_ints.h 1  C platform.h

openharmory > vendor > democom > demochip > driver > uart > C uart_hardware.h >...

24  * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
25  * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
26  * OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
27  * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
28  * OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
29  * ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
30  */
31
32  #ifndef _AMBA_PL011_UART_H
33  #define _AMBA_PL011_UART_H
34
35  #include "los_typedef.h"
36
37  #ifdef __cplusplus
38  #if __cplusplus
39  extern "C" {
40  #endif /* __cplusplus */
41  #endif /* __cplusplus */
42
43
44  /*根据IMX6ULL芯片手册<<55.15 UART Memory Map/Register Definition>>的3608页, 定义UART的结构体,*/
45  typedef struct {
46      volatile unsigned int URXD;          /*< UART Receiver Register, offset: 0x0      串口接收寄存器, 偏移地
47      unsigned char RESERVED_0[60];
48      volatile unsigned int UTXD;          /*< UART Transmitter Register, offset: 0x40    串口发送寄存器, 偏移地
```

使用 imx6ull_uart.c 文件替换整个 uart_hardware.c 文件

```
Makefile ~/stm32mp157/...  C bsd_board.c 3  C uarth 3  C uart_hardware.h  C uart_imx6ull.c 9+ X  C platform.h ~/demochip/... 2  C hal_platform_ints.h 1

openharmory > vendor > nxp > imx6ull > driver > imx6ull-uart > C uart_imx6ull.c

12  #include "linux/spinlock.h"
13  #include "imx6ull_uart.h"
14  #include "asm/platform.h"
15  #include <sys/bus.h>
16
17  #include "uart.h"
18  #include "uart_dev.h"
19  #include "string.h"
20
21  #include "los_magickey.h"
22  #include "imx6ull_uart.h"
23
24  struct imx6ull_port {
25      int enable;
26      unsigned long phys_base;
27      unsigned int irq_num;
28      struct uart_driver_data *udd;
29  };
30
31  __attribute__((section(".data"))) UINT32 g_uart_fputc_en = 1;
32
33
34  LITE_OS_SEC_BSS STATIC SPIN_LOCK_INIT(g_uartOutputSpin);
```

```
Makefile ~/stm32mp157/...  C bsd_board.c 3  C uarth 3  C uart_hardware.h  C uart_hardware.c 9+ X  C platform.h ~/demochip/... 2  C hal_platform_ints.h 1

openharmory > vendor > democom > demochip > driver > uart > C uart_hardware.c

364  (void)snprintf_s(dev_name, MAX_DEV_NAME_SIZE, MAX_DEV_NAME_SIZE - 1, "/dev/uartdev-%d", udd->num);
365  if (unregister_driver(dev_name)) {
366      uart_error("imx6ull_detach unregister /dev/uartdev-%d fail!\n", udd->num);
367  }
368  port = udd->private;
369  if (port == NULL) {
370      return -1;
371  }
372  if (port->phys_base) {
373      iounmap((void *) (uintptr_t) port->phys_base);
374      port->phys_base = 0;
375  }
376  (VOID)LOS_MemFree(m_aucSysMem0, port);
377  udd->private = NULL;
378  return 0;
379  }
380
381  static device_method_t uart_methods[] =
382  {
383      /* Device interface */
384      DEVMETHOD(device_probe, imx6ull_probe),
385      DEVMETHOD(device_attach, imx6ull_attach),
386      DEVMETHOD(device_detach, imx6ull_detach),
387      DEVMETHOD(device_shutdown, bus_generic_shutdown),
388      DEVMETHOD_END
389  }
```

修改第 22 行的包含文件，将其修改成 uart_hardware.h

```
Makefile ~/stm32mp157/...  C bsd_board.c 3  C uarth 3  C uart_hardware.h  C uart_hardware.c 4 X  C platform.h ~/demochip/... 2
openharmony > vendor > democom > demochip > driver > uart > C uart_hardware.c > ...
10 #include "linux/interrupt.h"
11 #include "linux/kernel.h"
12 #include "linux/spinlock.h"
13 #include "imx6ull_uart.h"
14 #include "asm/platform.h"
15 #include <sys/bus.h>
16
17 #include "uart.h"
18 #include "uart_dev.h"
19 #include "string.h"
20
21 #include "los_magickey.h"
22 #include "uart_hardware.h"
23
24 struct imx6ull_port {
25     int enable;
26     unsigned long phys_base;
27     unsigned int irq_num;
28     struct uart_driver_data *udd;
29 };
30
31 __attribute__((section(".data"))) UINT32 g_uart_fputc_en = 1;
32
33
34 LITE OS SEC BSS STATIC SPIN LOCK INIT(g_uartOutputSpin);
```

使用全局替换的方式将 imx6ull 替换成 demochip

```
openharmony > vendor > democom > demochip > driver > uart > C uart_hardware.c > C imx6ull_uart_irq(int, void)
106 }
107 VOID uart_early_init(VOID)
108 {
109     /* enable uart transmit */
110 }
111
112 VOID uart_init(VOID)
113 {
114 }
115
116 #define FIFO_SIZE 128
117
118 static irqreturn_t imx6ull_uart_irq(int irq, void *data)
119 {
120     char buf[FIFO_SIZE];
121     unsigned int count = 0;
122     struct imx6ull_port *port = NULL;
123     struct uart_driver_data *udd = (struct uart_driver_data *)data;
124     UART_Type *uartRegs;
125     uint32_t status;
126
127     if (udd == NULL) {
128         uart_error("udd is null!\n");
129         return IRQ_HANDLED;
130     }
131 }
```

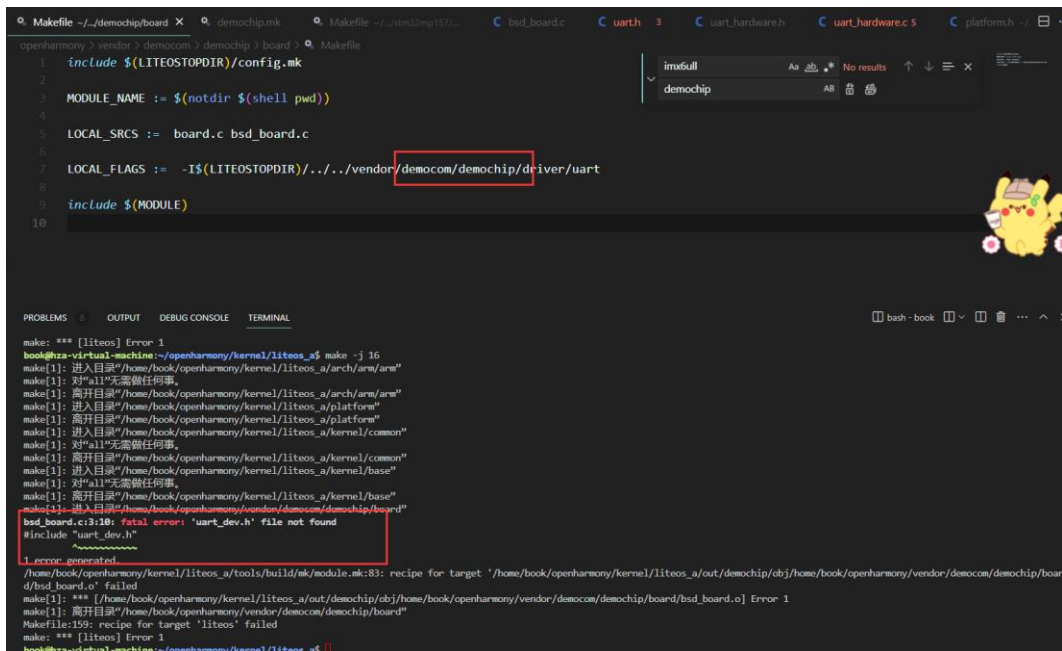
删掉这个头文件包含

```
Makefile ~/stm32mp157/... X  C bsd_board.c 3  C uarth 3  C uart_hardware.h  C uart_hardware.c 4 X  C platform.h ~/demochip/...  C hal_platfo
openharmony > vendor > democom > demochip > driver > uart > C uart_hardware.c > ...
3
4 #include "los_typedef.h"
5 #include "los_task.h"
6 #include "los_base.h"
7 #include "los_event.h"
8 #include "errno.h"
9
10 #include "linux/interrupt.h"
11 #include "linux/kernel.h"
12 #include "linux/spinlock.h"
13 #include "demochip_uart.h"
14 #include "asm/platform.h"
15 #include <sys/bus.h>
16
17 #include "uart.h"
18 #include "uart_dev.h"
19 #include "string.h"
20
21 #include "los_magickey.h"
22 #include "uart_hardware.h"
23
24 struct demochip_port {
25     int enable;
26     unsigned long phys_base;
27     unsigned int irq_num;
```


之后将对应的配置文件复制到指定目录，之后执行 make clean 和 make -j 16 命令编译内核

```
book@hza-virtual-machine:~/openharmy$ cd kernel/liteos_a
book@hza-virtual-machine:~/openharmy/kernel/liteos_a$ cp tools/build/config/debug/demochip_clang.config .config
book@hza-virtual-machine:~/openharmy/kernel/liteos_a$
```

这里会出现第一次编译出错，原因就是前面提到的 makefile 的文件没有修改全



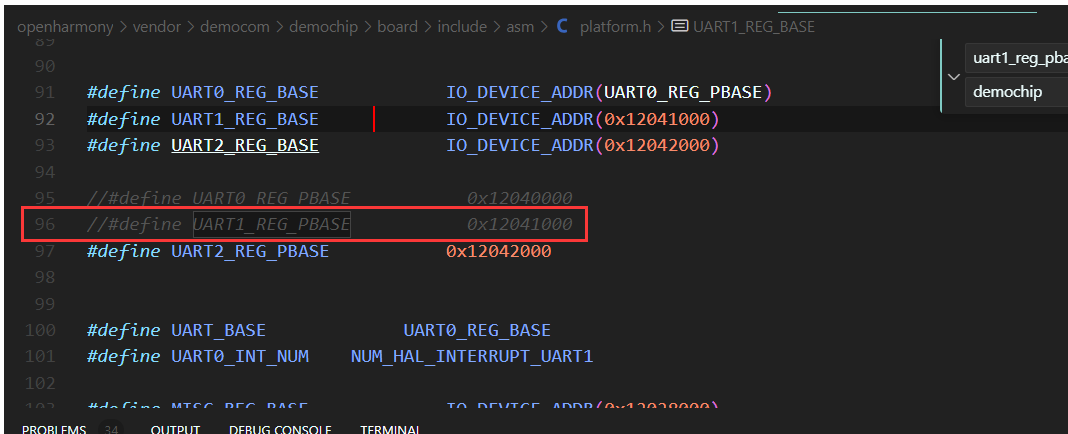
```
Makefile - ./demochip/board X demochip.mk Makefile - ./demochip/board X demochip.mk
openharmy > vendor > democom > demochip > board > Makefile
1 include $(LITEOSTOPDIR)/config.mk
2
3 MODULE_NAME := $(notdir $(shell pwd))
4
5 LOCAL_SRCS := board.c bsd_board.c
6
7 LOCAL_FLAGS := -I$(LITEOSTOPDIR)/../../vendor/democom/demochip/driver/uart
8
9 include $(MODULE)
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
make: *** [liteos] Error 1
book@hza-virtual-machine:~/openharmy/kernel/liteos_a$ make -j 16
make[1]: 进入目录"/home/book/openharmy/kernel/liteos_a/arch/arm/arm"
make[1]: 对"all"无需做任何事。
make[1]: 离开目录"/home/book/openharmy/kernel/liteos_a/arch/arm/arm"
make[1]: 进入目录"/home/book/openharmy/kernel/liteos_a/platform"
make[1]: 对"all"无需做任何事。
make[1]: 离开目录"/home/book/openharmy/kernel/liteos_a/platform"
make[1]: 进入目录"/home/book/openharmy/kernel/liteos_a/kernel/common"
make[1]: 对"all"无需做任何事。
make[1]: 离开目录"/home/book/openharmy/kernel/liteos_a/kernel/common"
make[1]: 进入目录"/home/book/openharmy/kernel/liteos_a/kernel/base"
make[1]: 对"all"无需做任何事。
make[1]: 离开目录"/home/book/openharmy/kernel/liteos_a/kernel/base"
make[1]: 进入目录"/home/book/openharmy/vendor/democom/demochip/board"
make[1]: bsd_board.c:3:10: fatal error: 'uart_dev.h' file not found
#include "uart_dev.h"
^~~~~~
1 error generated.
/home/book/openharmy/kernel/liteos_a/tools/build/mk/module.mk:83: recipe for target '/home/book/openharmy/kernel/liteos_a/out/demochip/obj/home/book/openharmy/vendor/democom/demochip/board/bsd_board.o' failed
make[1]: *** [/home/book/openharmy/kernel/liteos_a/out/demochip/obj/home/book/openharmy/vendor/democom/demochip/board/bsd_board.o] Error 1
make[1]: 离开目录"/home/book/openharmy/vendor/democom/demochip/board"
Makefile:159: recipe for target 'liteos' failed
make: *** [liteos] Error 1
book@hza-virtual-machine:~/openharmy/kernel/liteos_a$
```

再次编译内核，出现另一个报错，发现是宏重复定义了，对其进行修改



```
In file included from /home/book/openharmy/kernel/liteos_a/../../vendor/democom/demochip/board/include/clock.h:19:
/home/book/openharmy/kernel/liteos_a/../../vendor/democom/demochip/board/include/asm/platform.h:96:9: src/los_hw.c:32:
In file included from /home/book/openharmy/kernel/liteos_a/arch/arm/arm/src/include/los_hw_pri.h:35:
In file included from /home/book/openharmy/kernel/liteos_a/kernel/include/los_base.h:43:
In file included from /home/book/openharmy/kernel/liteos_a/platform/../../kernel/common/los_config.h:39:
/home/book/openharmy/kernel/liteos_a/../../vendor/democom/demochip/board/include/platform_config.h:19:
/home/book/openharmy/kernel/liteos_a/../../vendor/democom/demochip/board/include/clock.h:19:
error: 'UART1_REG_PBASE' macro redefined [-Werror, -Wmacro-redefined]
#define UART1_REG_PBASE 0x12041000
^
/home/book/openharmy/kernel/liteos_a/../../vendor/democom/demochip/board/include/asm/platform.h:86:9: note: previous definition is here
#define UART1_REG_PBASE 0x02020000 /* imxull uart1 */
^
error: 'UART1_REG_PBASE' macro redefined [-Werror, -Wmacro-redefined]
#define UART1_REG_PBASE 0x12041000
^
/home/book/openharmy/kernel/liteos_a/../../vendor/democom/demochip/board/include/asm/platform.h:86:9: note: previous definition is here
#define UART1_REG_PBASE 0x02020000 /* imxull uart1 */
^
In file included from src/los_hw_tick.c:32:
In file included from /home/book/openharmy/kernel/liteos_a/kernel/base/include/los_sys_pri.h:35:
In file included from /home/book/openharmy/kernel/liteos_a/kernel/include/los_sys.h:40:
In file included from /home/book/openharmy/kernel/liteos_a/kernel/include/los_base.h:43:
In file included from /home/book/openharmy/kernel/liteos_a/platform/../../kernel/common/los_config.h:39:
/home/book/openharmy/kernel/liteos_a/../../vendor/democom/demochip/board/include/platform_config.h:19:
error: 'UART1_REG_PBASE' macro redefined [-Werror, -Wmacro-redefined]
#define UART1_REG_PBASE 0x12041000
^
/home/book/openharmy/kernel/liteos_a/../../vendor/democom/demochip/board/include/asm/platform.h:86:9: note: previous definition is here
#define UART1_REG_PBASE 0x02020000 /* imxull uart1 */
^
```



```
openharmy > vendor > democom > demochip > board > include > asm > C platform.h > UART1_REG_PBASE
90
91 #define UART0_REG_BASE IO_DEVICE_ADDR(UART0_REG_PBASE)
92 #define UART1_REG_BASE IO_DEVICE_ADDR(0x12041000)
93 #define UART2_REG_BASE IO_DEVICE_ADDR(0x12042000)
94
95 // #define UART0_REG_PBASE 0x12040000
96 // #define UART1_REG_PBASE 0x12041000
97 #define UART2_REG_PBASE 0x12042000
98
99
100 #define UART_BASE UART0_REG_BASE
101 #define UART0_INT_NUM NUM_HAL_INTERRUPT_UART1
102
103 #define UART1_REG_PBASE IO_DEVICE_ADDR(0x12041000)
```


最后再次编译内核发现没有报错，链接成功生成 liteos.bin 文件，将其复制到烧写工具的 files 文件中之后烧写到开发板中

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
/home/hza/llvm/bin/./bin/ld -static --gc-sections -L/home/book/openharmony/kernel/liteos_a/../../vendor/demochip/hdf/libs/demochip -L/home/book/openharmony/kernel/liteos_a/tools/scr
ipts/ld -L/home/book/openharmony/kernel/liteos_a/platform -L/home/book/openharmony/kernel/liteos_a/out/demochip/lib -L/home/book/openharmony/kernel/liteos_a/out/demochip/lib/obj -L/home/book/op
enharmony/kernel/liteos_a/tools/build -L/home/hza/llvm/bin/./lib/clang/9.0.0/lib/arm-liteos/a7_softfp_neon-vfpv4 -L/home/hza/llvm/bin/./lib/arm-liteos/c++/a7_softfp_neon-vfpv4 -T/home/book/op
enharmony/kernel/liteos_a/tools/build/liteos_llvm.ld -utask_shellcmd -ug_usrVdoDataPage -uvm_shellcmd -ucupup_shellcmd -uhelp_shellcmd -ufree_shellcmd -usem_shellcmd -umutex_shellcmd -uqueue
_shellcmd -usysteminfo_shellcmd -useter_shellcmd -uhdi_shellcmd -ofindsys_shellcmd -ubar_scan_shellcmd -uadmsc_shellcmd -update_shellcmd -ureadXcinfo_shellcmd -uwatch_shellcmd -uadlock_shellc
md -ukill_shellcmd -upgm_shellcmd -uarp_shellcmd -utelnet_shellcmd -uping_shellcmd -unetstat_shellcmd -uifconfig_shellcmd -ucd_shellcmd -ufonmt_shellcmd -upartition_shellcmd -uwriteproc_shell
cmd -upartinfo_shellcmd -uamout_shellcmd -umount_shellcmd -uvirstatfs_shellcmd -ulsfd_shellcmd -usu_shellcmd -uchmod_shellcmd -uchown_shellcmd -uchgrp_shellcmd -uiperf_shellcmd -ulvip_dump_she
llcmd -ureset_shellcmd -ustartap_shellcmd -uhiddrs_shellcmd -unandbad_shellcmd -ui2c_read_shellcmd -ussp_read_shellcmd -uuart_config_shellcmd -uusb_debug_shellcmd -uramfs_fsmap -unfs_fsmap -uf
at_fsmap -uiffs_fsmap -uprocs_fsmap -ug_fsmap -ui2c_init -ugpio_init -uregulator_init -uMtdInitList -uhsipi_init -uhifmc100_init -uhifsc350_init -unand_hifmc100_init -uhifmc100_parallel_init -
usd_mci_init -uhl_mci_init -upl011_init -uhinfc620_init -uhisnfc100_init -uregulator_machine_init -uhisimedia_regulator_init -ucupfreq_init -uhisilicon_cpufreq_init -ucupfreq_machine_init -ude
vfreq_init -umedia_devfreq_init -udevfreq_machine_init -uhieth_machine_init -uhimac_machine_init -umachine_init -Map/home/book/openharmony/kernel/liteos_a/out/demochip/liteos.map -o /home/bo
ok/openharmony/kernel/liteos_a/out/demochip/liteos --start-group -lclang_rt.builtins -lundin -no-dependent-libraries -lcortex-a7 -lbsp -lroots -lbase -lboard -lmtcommon -lspinor_flash -ls
tm32mp157-fb -luart -lcup -ldynload -ldvso -ltickless -lliteipc -lpcres -lscrcw -lcm+ -lcmabi -lcpusupport -lz -lposix -lbsd -llinuxkpi -lvfs -lmulti-partition -lbch -lifat -lvirpart
-ldisk -lbcache -lramfs -lnfs -lproc -ljffs2 -llwip --whole-archive -lhdf -lhdf_config -lhello --no-whole-archive -lhivert -lmem -lmtcommon -lvideo -lbilog -lshell -ltelnet -lsyscall -lsecur
ity --end-group
/home/hza/llvm/bin/./bin/llvm-objcopy -R .bss -O binary /home/book/openharmony/kernel/liteos_a/out/demochip/liteos /home/book/openharmony/kernel/liteos_a/out/demochip/liteos.bin
/home/hza/llvm/bin/./bin/llvm-objdump -t /home/book/openharmony/kernel/liteos_a/out/demochip/liteos | sort >/home/book/openharmony/kernel/liteos_a/out/demochip/liteos.sym.sorted
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/apps"
make[2]: 进入目录"/home/book/openharmony/kernel/liteos_a/apps/shell"
make[2]: 离开目录"/home/book/openharmony/kernel/liteos_a/apps/shell"
make[2]: 进入目录"/home/book/openharmony/kernel/liteos_a/apps/init"
make[2]: 离开目录"/home/book/openharmony/kernel/liteos_a/apps/init"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a/apps"
book@hza-virtual-machine:~/openharmony/kernel/liteos_a$
```

运行后发现内核没有启动，根据视频里的方法进行调试

```
COM9 (USB-Enhanced-SERIAL CH9102 (COM9))
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
sessions
192.168.138.130
47.96.157.89 (root)
book
COM9 (USB-Enhanced-SERIAL CH9102 (CO
  刷机
  操作系统开发板
  测试机
  数据库机
Starting download of 920412 bytes
.....
downloading of 920412 bytes finished
UCmd: UCmd: mw.L 98000000 ffffffff 1000000
UCmd: UCmd: cp.b ${fastboot_buffer} 98000000 ${fastboot_bytes}
Starting download of 891820 bytes
.....
downloading of 891820 bytes finished
UCmd: UCmd: cp.b ${fastboot_buffer} 81000000 ${fastboot_bytes}
UCmd: UCmd: go 81000000
## Starting application at 0x81000000 ...
```

在内核启动前的汇编程序中增加一段调试代码，如果编译平台是 DEMOCHIP 的话就执行下 uart_putc_phy 函数，其作用是输出一个字符

```
los_process.c 4. M reset_vector_up.S M Makefile ~/.../uart Makefile ~/.../demochip/board demochip.m
openharmony > kernel > liteos_a > arch > arm > src > startup > reset_vector_up.S
111
112 /* Startup code which get the machine into supervisor mode */
113 .global reset_vector
114 .type reset_vector,function
115 reset_vector:
116 #if defined(LOSCFG_PLATFORM_DEMOCHIP)
117 ldr sp, =0x80000000 + 0x1000000
118 mov r0, #'S'
119 bl uart_putc_phy
120 #endif
121 #if defined(LOSCFG_PLATFORM_STM32MP157)
122 ldr sp, =0xc0000000 + 0x1000000
123 mov r0, #'S'
124 bl uart_putc_phy
125 #endif
126 #if 1
127 /*
128 * disable interrupts (FIQ and IRQ), also set the cpu to SVC32 mode,
```

这里调试的时候需要使用物理地址，因为这个时候内核刚启动，MMU 还没有启动，所以还没有虚拟地址的概念，因此需要使用物理地址

```
openharmy > vendor > democom > demochip > driver > uart > C uart_hardware.c > ...
27 | struct uart_driver_data *udd;
28 | };
29 |
30 | __attribute__((section(".data"))) UINT32 g_uart_fputc_en = 1;
31 |
32 |
33 | LITE_OS_SEC_BSS STATIC SPIN_LOCK_INIT(g_uartOutputSpin);
34 |
35 | void uart_putc_phy(char c)
36 | {
37 |     UART_Type *uartRegs = (UART_Type *)0x0202000;
38 |
39 |     while (!((uartRegs->USR2) & (1<<3))); /*等待上个字节发送完毕*/
40 |     uartRegs->UTXD = (unsigned char)c;
41 | }
42 |
43 | STATIC INLINE UINTPTR uart_to_ptr(UINTPTR n)
```

运行后发现输出一个 S，说明该部分没有问题

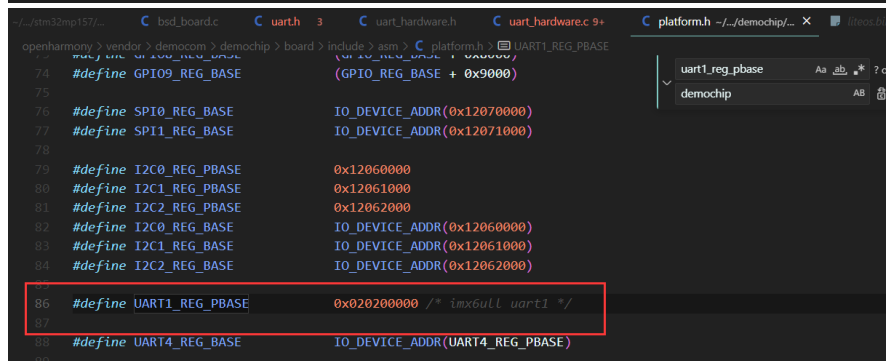
```
Video: MXSFB: 'videomode' variable not set!
In: serial
Out: serial
Err: serial
switch to partitions #0, OK
mmc1(part 0) is current device
flash target is MMC:1
Net: No ethernet found.
Fastboot: Normal
Hit any key to stop autoboot: 0
Starting download of 9011940 bytes
.....
downloading of 9011940 bytes finished
UCmd : UCmd:mw.l 98000000 ffffffff 1000000
UCmd : UCmd:cp.b ${fastboot_buffer} 98000000 ${fastboot_bytes}
Starting download of 891744 bytes
.....
downloading of 891744 bytes finished
UCmd : UCmd: cp.b ${fastboot_buffer} 81000000 ${fastboot_bytes}
UCmd : UCmd: go 81000000
## Starting application at 0x81000000 ...
S
```

之后再根据视频继续调试

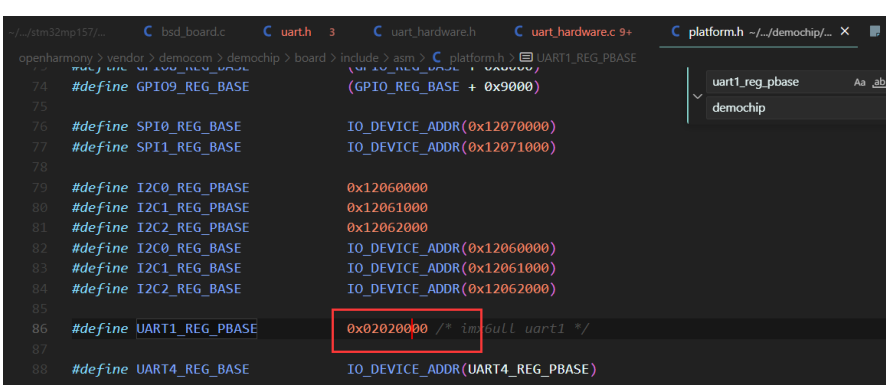
```
C los_process.c 4.M reset_vector_up.S M Makefile ~/./uart Makefile ~/./demochip/board demochip
openharmy > kernel > liteos_a > arch > arm > arm > src > startup > reset_vector_up.S
312 | #endif
313 |
314 | #ifdef LOSCFG_GDB_DEBUG
315 |     /* GDB_START - generate a compiled binary, This function will get GDB stubs start
316 |     bl GDB_START
317 |     .word 0xe7ffdeff
318 | #endif
319 | #if defined(LOSCFG_PLATFORM_STM32MP157)
320 |     mov r0, 'm'
321 |     bl uart_putc_virt
322 | #endif
323 |
324 | #if defined(LOSCFG_PLATFORM_DEMOCHIP)
325 |     mov r0, 'm'
326 |     bl uart_putc_virt
327 | #endif
328 |
329 |     bl main
```

再实现一下这个 `uart_putc_virt`,这个时候 MMU 已经启动了因此可以使用虚拟地址, 这里韦东山老师发现多输了一个 0, 同步进行修改即可

```
43 void uart_putc_virt(char c)
44 {
45     UART_Type *uartRegs = (UART_Type *)UART_REG_BASE;
46
47     while (!((uartRegs->USR2) & (1<<3))); /*等待上个字节发送完毕*/
48     uartRegs->UTXD = (unsigned char)c;
49 }
50
```



```
74 #define GPIO0_REG_BASE      (GPIO_REG_BASE + 0x9000)
75
76 #define SPI0_REG_BASE      IO_DEVICE_ADDR(0x12070000)
77 #define SPI1_REG_BASE      IO_DEVICE_ADDR(0x12071000)
78
79 #define I2C0_REG_PBASE      0x12060000
80 #define I2C1_REG_PBASE      0x12061000
81 #define I2C2_REG_PBASE      0x12062000
82 #define I2C0_REG_BASE      IO_DEVICE_ADDR(0x12060000)
83 #define I2C1_REG_BASE      IO_DEVICE_ADDR(0x12061000)
84 #define I2C2_REG_BASE      IO_DEVICE_ADDR(0x12062000)
85
86 #define UART1_REG_PBASE      0x02020000 /* imx6ull uart1 */
87
88 #define UART4_REG_BASE      IO_DEVICE_ADDR(UART4_REG_PBASE)
89
```



这里运行后出现了 Sm 和 main 的提示,说明串口可以使用,根据视频进行研究可以发现是中断控制器的地址没有进行修改

```
Out: serial
Err: serial
switch to partitions #0, OK
mmc1(part 0) is current device
flash target is MMC:1
Net: No ethernet found.
Fastboot: Normal
Hit any key to stop autoboot: 0
Starting download of 9011940 bytes
.....
downloading of 9011940 bytes finished
UCmd : UCmd: mw.l 98000000 ffffffff 1000000
UCmd : UCmd: cp.b ${fastboot_buffer} 98000000 ${fastboot_bytes}
Starting download of 891744 bytes
.....
downloading of 891744 bytes finished
UCmd : UCmd: cp.b ${fastboot_buffer} 81000000 ${fastboot_bytes}
UCmd : UCmd: go 81000000
## Starting application at 0x81000000 ...
Sm
*****Main*****
```

对中断控制器相关的内容进行修改，删除掉这里的加 0x20000 偏移

```
openharmy > vendor > democom > demochip > board > include > asm > platform.h > GIC_BASE_ADDR
28 #endif /* __cplusplus */
29 #endif /* __cplusplus */
30
31 /*-----
32  * GIC reg base address
33  *-----*/
34 /* GIC_PHY_BASE(0xA0000000) ==> GIC_VIRT_BASE
35  * GICD(0xA0021000)
36  * GICC(0xA0022000)
37  * so: GIC_BASE_ADDR = vir addr of 0xA0020000
38  */
39 //define GIC_BASE_ADDR TO_DEVICE_ADDR(0x00a00000)
40 #define GIC_BASE_ADDR (GIC_VIRT_BASE)
41 #define GICD_OFFSET 0x1000 /* interrupt distributor offset */
42 #define GICC_OFFSET 0x2000 /* CPU interface register offset */
43
```

删除掉该宏后面加上的东西

```
openharmy > kernel > liteos_a > kernel > base > include > los_vm_zone.h > GIC_VIRT_BASE
73 #define PERIPH_CACHED_BASE (PERIPH_CACHED_SIZE)
74 #define PERIPH_CACHED_SIZE PERIPH_CACHED_SIZE
75 #define PERIPH_UNCACHED_BASE (PERIPH_CACHED_BASE + PERIPH_CACHED_SIZE)
76 #define PERIPH_UNCACHED_SIZE PERIPH_PMM_SIZE
77
78 //define GIC_VIRT_BASE (PERIPH_UNCACHED_BASE + PERIPH_UNCACHED_SIZE)
79 #define GIC_VIRT_BASE (PERIPH_UNCACHED_BASE)
80 #define GIC_VIRT_SIZE GIC_PHY_SIZE
```

之后再重新编译与烧写内核，运行后可以看到成功输出了信息，后面文件系统没有移植，因此相关部分没有运行成功，会在后续实验中完成

```
## Starting application at 0x81000000 ...
*****Main*****
*****Welcome*****
Processor : Cortex-A7
Run Mode : UP
GIC Rev : GICv2
Build time : Nov 16 2023 02:04:31
Kernel : Huawei LiteOS 2.0.0.35/debug
*****
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
DeviceManagerStart start ...
[ERR][HDF:E/hcs_blob_if]CheckHcsBlobLength: the blobLength: 76, byteAlign: 1, totalSize: -56
[ERR][HDF:E/HDF_LOG_TAG]HdfAttributeManagerGetHostList get hdf manager node is null
[ERR]No drivers need load by hdf manager!DeviceManagerStart end ...
[ERR]No console dev used.
[ERR]No console dev used.
[ERR]OsGetFileLength[64], The file: /bin/init is invalid!
```

四、实验心得

本次实验和之前代码阅读实验共享中断有些类似，但是更加全面了解了串口有关的驱动程序，实验如果直接在虚拟机中操作文件互相跳转过于麻烦，因此学会了配置 vscode 的 ssh 连接，方便了开发虚拟机中的文件

此外还学到了一些调试的方法，之前阅读代码的时候最不敢碰的就是汇编代码，这里韦东山老师使用了汇编调 C 函数的方法展示了一种调试的手段

另外本次实验也进一步解了操作系统对于 IO 的管理和 IO 的内存映射关系，通过代码的角度对这些之前在计组里学到的东西有些新的认知。