实用操作系统课程实验报告

实验名称：	实验三 鸿蒙 LiteOS-a 内核移植——增加一个单板
实验日期：	2023-10-15
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

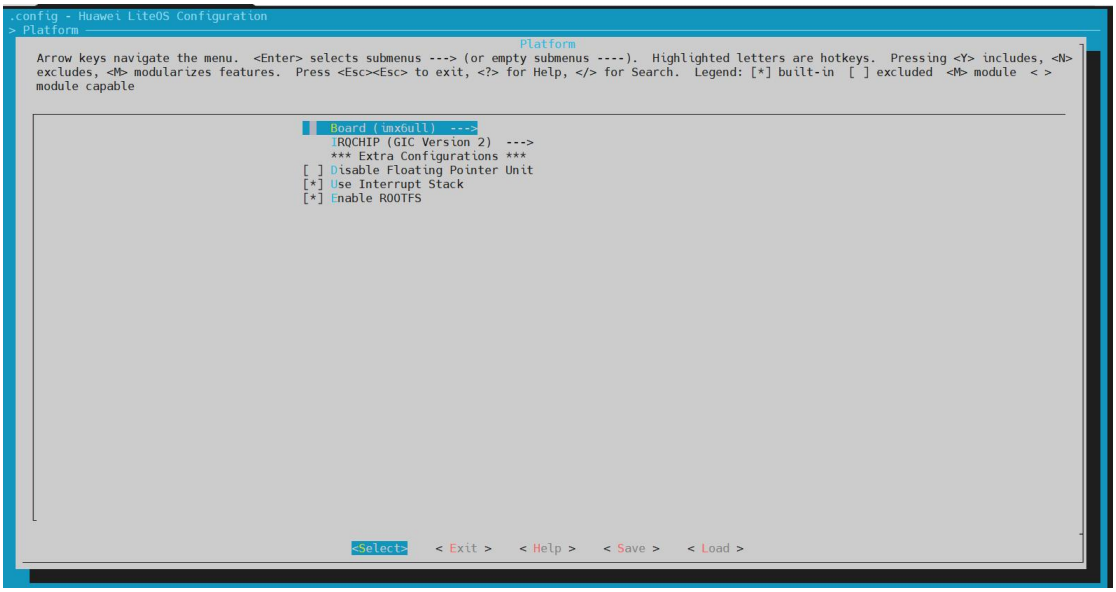
1.实验目的

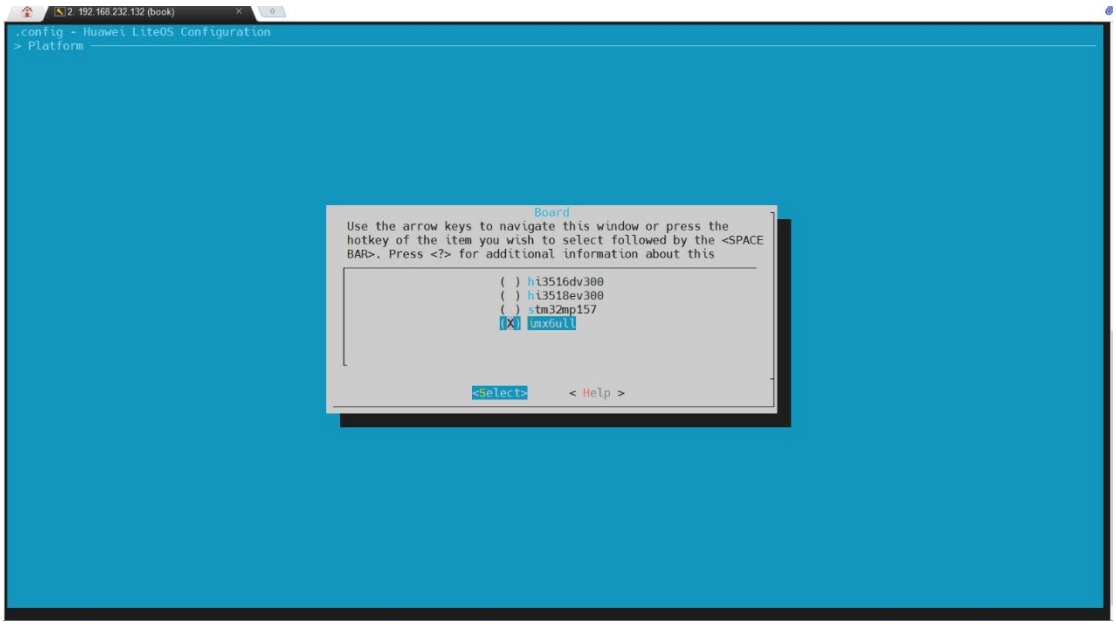
- 移植鸿蒙 LiteOS-a 内核，为其增加一个单板

2.实验内容和步骤

（1） 查看当前配置菜单：

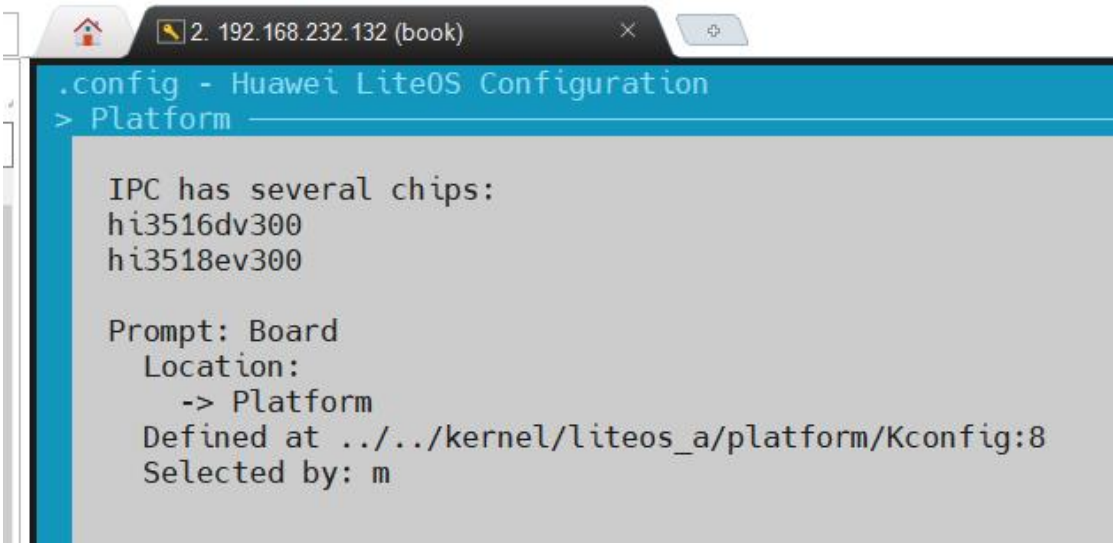
```
book@ry-virtual-machine:~$ cd openharmony
book@ry-virtual-machine:~/openharmony$ cd kernel/liteos_a
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ make menuconfig
```





可见当前配置中有四个选项，实验目的是为其再增加一个单板选项

接着查看平台，可见配置文件路径：



（2）修改配置文件，增加 demochip：

```
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ vi platform/Kconfig
```

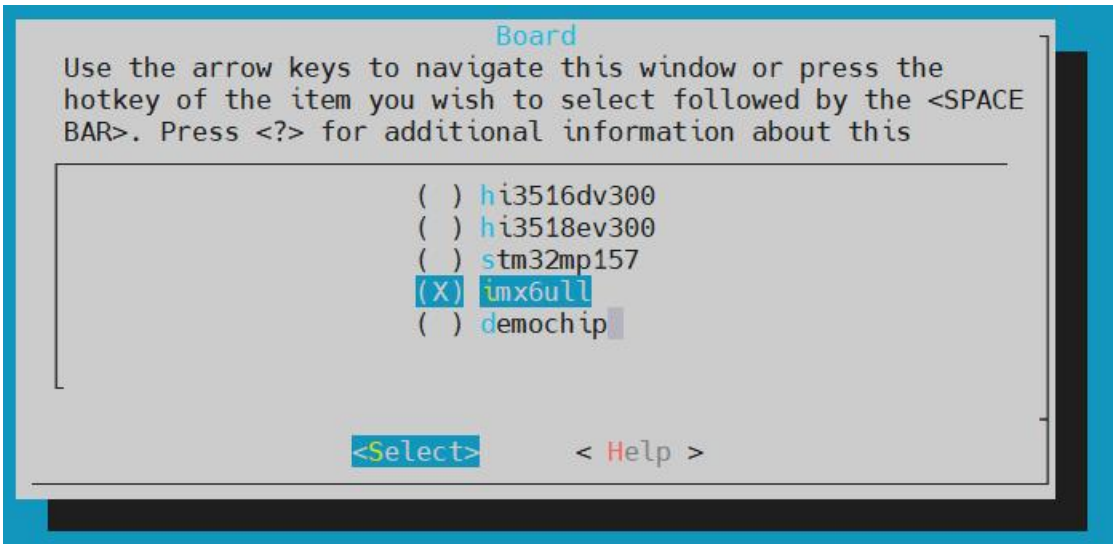
```
config PLATFORM
    string
    default "hi3516dv300"     if PLATFORM_HI3516DV300
    default "hi3518ev300"     if PLATFORM_HI3518EV300
    default "imx6ull"         if PLATFORM_IMX6ULL
    default "stm32mp157"      if PLATFORM_STM32MP157
    default "demochip"        if PLATFORM_DEMOCHIP

choice
    prompt "Board"
    default PLATFORM_HI3516DV300
    help
      IPC has several chips:
      hi3516dv300
      hi3518ev300

config PLATFORM_HI3516DV300
    bool "hi3516dv300"
    select ARCH_CORTEX_A7

config PLATFORM_HI3518EV300
    bool "hi3518ev300"
    select ARCH_CORTEX_A7

config PLATFORM_STM32MP157
    bool "stm32mp157"
    select ARCH_CORTEX_A7

config PLATFORM_IMX6ULL
    bool "imx6ull"
    select ARCH_CORTEX_A7

config PLATFORM_DEMOCHIP
    bool "demochip"
    select ARCH_CORTEX_A7

endchoice
```

红框为修改部分。

```
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ vi .config
```

```
#
# Platform
#
LOSCFG_PLATFORM="demochip"
# LOSCFG_PLATFORM_HI3516DV300 is not set
# LOSCFG_PLATFORM_HI3518EV300 is not set
# LOSCFG_PLATFORM_STM32MP157 is not set
# LOSCFG_PLATFORM_IMX6ULL is not set
LOSCFG_PLATFORM_DEMOCHIP=y
LOSCFG_PLATFORM_BSP_GIC_V2=y
```

此时再次编译运行配置菜单，即可看见新增的选项：



（2） 根据配置项修改 Makefile：

```
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ grep "LOSCFG_PLATFORM_STM32MP157" * -nr
arch/arm/arm/src/startup/reset_vector_up.S.rej:7:+#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S.rej:50:+#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S.rej:61:+#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S.rej:71:+#if defined(LOSCFG_PLATFORM_IMX6ULL) || defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S.rej:79:+#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S.rej:90:+#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S.rej:103:+#ifndef  LOSCFG_PLATFORM_STM32MP157
arch/arm/arm/src/startup/reset_vector_up.S.rej:119:+#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S:116:#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S:158:#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S:178:#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S:187:#if defined(LOSCFG_PLATFORM_IMX6ULL) || defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S:195:#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S:217:#if defined(LOSCFG_PLATFORM_STM32MP157)
arch/arm/arm/src/startup/reset_vector_up.S:273:#ifndef  LOSCFG_PLATFORM_STM32MP157
arch/arm/arm/src/startup/reset_vector_up.S:314:#if defined(LOSCFG_PLATFORM_STM32MP157)
kernel/base/include/los_vm_zone.h:57:#elif defined LOSCFG_PLATFORM_STM32MP157
kernel/base/include/los_vm_zone.h.rej:12:+#elif defined LOSCFG_PLATFORM_STM32MP157
Makefile:65:ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
Makefile:113:else ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
Makefile:162:ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
Makefile.rej:10:+ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
Makefile.rej:23:+else ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
Makefile.rej:40:+ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
platform/Makefile:43:ifeq ($(findstring y, $(LOSCFG_PLATFORM_HI3518EV300)$(LOSCFG_PLATFORM_HI3516DV300)$(LOSCFG_PLATFORM_IMX6ULL)$(LOSCFG_PLATFORM_STM32MP157)
), y)
platform/bsp.mk.rej:12:+else ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
platform/bsp.mk.rej:25:+else ifeq ($(LOSCFG_PLATFORM_STM32MP157),y)
platform/bsp.mk:62:else ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
platform/bsp.mk:92:else ifeq ($(LOSCFG_PLATFORM_STM32MP157),y)
platform/Makefile.rej:8:+ifeq ($(findstring y, $(LOSCFG_PLATFORM_HI3518EV300)$(LOSCFG_PLATFORM_HI3516DV300)$(LOSCFG_PLATFORM_IMX6ULL)$(LOSCFG_PLATFORM_STM32MP
157)), y)
```

即修改红框中的几个文件

```
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ vim platform/Makefile
```

```
                -I $(LITEOSTOPDIR)/bsd/dev/random

ifeq ($(findstring y, $(LOSCFG_PLATFORM_HI3518EV300)$(LOSCFG_PLATFORM_HI3516DV300)$(LOSCFG_PLATFORM_IMX6ULL)$(LOSCFG_PLATFORM_STM32MP157)$(LOSCFG_PLATFORM_DEMOCHIP)), y)
LOCAL_SRCS += $(wildcard ../kernel/common/*.c)
LOCAL_SRCS := $(filter-out ../kernel/common/los_rootfs.c, $(LOCAL_SRCS))
ifneq ($(LOSCFG_FS_VFS), y)
LOCAL_SRCS := $(filter-out ../kernel/common/console.c ../kernel/common/virtual_serial.c, $(LOCAL_SRCS))
endif
```

```
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ vim Makefile
```

```
ifeq ($(LOSCFG_PLATFORM_HI3518EV300), y)
FSTYPE = jffs2
endif
ifeq ($(LOSCFG_PLATFORM_HI3516DV300), y)
FSTYPE = vfat
endif
ifeq ($(LOSCFG_PLATFORM_IMX6ULL), y)
FSTYPE = jffs2
endif
ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
FSTYPE = jffs2
ROOTFS_SIZE = 0xA00000
ifeq ($(LOSCFG_PLATFORM_DEMOCHIP),y)
FSTYPE = vfat
ROOTFS_SIZE = 0xA00000
endif
ROOTFS_DIR = $(OUT)/rootfs
ROOTFS_ZIP = $(OUT)/rootfs.zip
VERSION =
```

```
ifeq ($(LOSCFG_PLATFORM_IMX6ULL),y)
BOARD_INCLUDE_DIR := $(LITEOSTOPDIR)/../../vendor/nxp/imx6ull/board
else ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
BOARD_INCLUDE_DIR := $(LITEOSTOPDIR)/../../vendor/st/stm32mp157/board
else ifeq ($(LOSCFG_PLATFORM_DEMOCHIP),y)
BOARD_INCLUDE_DIR := $(LITEOSTOPDOR)/../../vendor/democom/demochip/board
else
BOARD_INCLUDE_DIR := $(LITEOSTOPDIR)/../../vendor/hisi/hi35xx/$(LITEOS_PLATFORM)/config/board
endif

$(OUT): $(LITEOS_MENUCONFIG_H)
```

```
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ vim platform/bsp.mk
```

```
########################## HI3518EV300 Options#############################
else ifeq ($(LOSCFG_PLATFORM_HI3518EV300), y)
    HWI_TYPE     := arm/interrupt/gic
    TIMER_TYPE   := hisoc/timer
    HRTIMER_TYPE := hisoc/hrtimer
    NET_TYPE     := hieth
    UART_TYPE    := amba_pl011
    USB_TYPE     := usb3.0_hi3518ev300
    LITEOS_CMACRO_TEST += -DTEST3518EV300
else ifeq ($(LOSCFG_PLATFORM_IMX6ULL), y)
    HWI_TYPE     := arm/interrupt/gic
    TIMER_TYPE   := arm/timer/arm_generic
    HRTIMER_TYPE := imx6ull/hrtimer
else ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
    HWI_TYPE     := arm/interrupt/gic
    TIMER_TYPE   := arm/timer/arm_generic
    HRTIMER_TYPE := stm32mp157/hrtimer
else ifeq ($(LOSCFG_PLATFORM_DEMOCHIP),y)
    HWI_TYPE     := arm/interrupt/gic
    TIMER_TYPE   := arm/timer/arm_generic
    HRTIMER_TYPE := demochip/hrtimer
endif

HWI_SRC     := hw/$(HWI_TYPE)
```

```
ifeq ($(findstring y, $(LOSCFG_PLATFORM_HI3518EV300)$(LOSCFG_PLATFORM_HI3516DV300)), y)
    PLATFORM_INCLUDE += -I $(LITEOSTOPDIR)/../../vendor/hisi/hi35xx/$(LITEOS_PLATFORM)/config/board/include/hisoc
else ifeq ($(LOSCFG_PLATFORM_IMX6ULL),y)
    PLATFORM_INCLUDE += -I $(LITEOSTOPDIR)/../../vendor/nxp/imx6ull/board/include
else ifeq ($(LOSCFG_PLATFORM_STM32MP157),y)
    PLATFORM_INCLUDE += -I $(LITEOSTOPDIR)/../../vendor/st/stm32mp157/board/include
else ifeq ($(LOSCFG_PLATFORM_DEMOCHIP),y)
    PLATFORM_INCLUDE += -I $(LITEOSTOPDIR)/../../vendor/democom/demochip/board/include
endif
```

```
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ vim Kconfig +39
```

```
config COMPILER_HIMIX_32
    bool "arm-linux-ohoseabi"
    depends on PLATFORM_HI3518EV300 || PLATFORM_HI3516DV300 || PLATFORM_IMX6ULL || PLATFORM_STM32MP157 || PLATFORM_DEMOCHIP

config COMPILER_CLANG_LLVM
    bool "clang-llvm"
        depends on PLATFORM_HI3518EV300 || PLATFORM_HI3516DV300 || PLATFORM_IMX6ULL || PLATFORM_STM32MP157 || PLATFORM_DEMOCHIP

endchoice
```

```
ifeq ($(LOSCFG_PLATFORM_HI3518EV300), y)
FSTYPE = jffs2
endif
ifeq ($(LOSCFG_PLATFORM_HI3516DV300), y)
FSTYPE = vfat
endif
ifeq ($(LOSCFG_PLATFORM_IMX6ULL), y)
FSTYPE = jffs2
endif
ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
FSTYPE = jffs2
ROOTFS_SIZE = 0xA00000
endif
ifeq ($(LOSCFG_PLATFORM_DEMOCHIP),y)
FSTYPE = vfat
ROOTFS_SIZE = 0xA00000
endif
ROOTFS_DIR = $(OUT)/rootfs
ROOTFS_ZIP = $(OUT)/rootfs.zip
VERSION =
```

（4）此时如果编译内核，将会遇到连接错误，因此应修改以下文件：

shell/full/src/base/show.c

shell/full/src/cmds/dmesg.c

shell/full/src/base/shcmd.c

shell/full/src/base/shmsg.c

按文件顺序分别做以下修改：

```
#include "show.h"
#include "shmsg.h"
#include "shcmd.h"
#include "console.h"
#include "asm/hal_platform_ints.h"
#ifdef LOSCFG_DRIVERS_HDF_PLATFORM_UART
#if defined LOSCFG_PLATFORM_IMX6ULL
#include "uart.h"
#elif defined LOSCFG_PLATFORM_STM32MP157
#include "uart.h"
#elif defined LOSCFG_PLATFORM_DEMOCHIP
#include "uart.h"
#else
#include "hisoc/uart.h"
#endif
#endif

-- 插入 --
```

```
#include "unistd.h"
#include "stdlib.h"
#include "los_task.h"
#ifdef LOSCFG_DRIVERS_HDF_PLATFORM_UART
#if defined LOSCFG_PLATFORM_IMX6ULL
#include "uart.h"
#elif defined LOSCFG_PLATFORM_STM32MP157
#include "uart.h"
#elif defined LOSCFG_PLATFORM_DEMOCHIP
#include "uart.h"
#else
#include "hisoc/uart.h"
#endif
#endif
#include "inode/inode.h"

#ifdef __cplusplus
#if __cplusplus
extern "C" {
```

```
#ifdef LOSCFG_DRIVERS_HDF_PLATFORM_UART
#if defined LOSCFG_PLATFORM_IMX6ULL
#include "uart.h"
#elif defined LOSCFG_PLATFORM_STM32MP157
#include "uart.h"
#elif defined LOSCFG_PLATFORM_DEMOCHIP
#include "uart.h"
#else
```

```
#if defined LOSCFG_PLATFORM_IMX6ULL
#include "uart.h"
#elif defined LOSCFG_PLATFORM_STM32MP157
#include "uart.h"
#elif defined LOSCFG_PLATFORM_DEMOCHIP
#include "uart.h"
#else
#include "hisoc/uart.h"
#endif
#endif
```

（5）编译内核：

```
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ make -j 16
make[1]: 进入目录“/home/book/openharmony/kernel/liteos_a”
/home/book/openharmony/kernel/liteos_a/tools/menuconfig/conf --silentoldconfig /home/book/openharmony/kernel/liteos_a/Kconfig
*
* Restart config...
*
*
* Compiler
*
LiteOS_Compiler_Type
  1. arm-linux-ohoseabi (COMPILER_HIMIX_32) (NEW)
> 2. clang-llvm (COMPILER_CLANG_LLVM)
choice[1-2?]:
#
# configuration written to .config
#
```

```
-lsecurity --end-group
/home/book/llvm/bin/..//bin/llvm-objcopy -R .bss -O binary /home/book/openharmony/kernel/liteos_a/out/demochip/liteos /home/book/openharmony/kernel/liteos_a/o
ut/demochip/liteos.bin
/home/book/llvm/bin/..//bin/llvm-objdump -t /home/book/openharmony/kernel/liteos_a/out/demochip/liteos |sort >/home/book/openharmony/kernel/liteos_a/out/democ
hip/liteos.sym.sorted
/home/book/llvm/bin/..//bin/llvm-objdump -d /home/book/openharmony/kernel/liteos_a/out/demochip/liteos >/home/book/openharmony/kernel/liteos_a/out/demochip/li
teos.asm
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@ry-virtual-machine:~/openharmony/kernel/liteos_a$
```

3.实验总结

通过这次实验，我在 Ubuntu 平台中首先接触了鸿蒙 liteos 的内核移植，并为其增加了单板，这让我对于鸿蒙系统的内核有了更深的理解。虽然这次实验并没有接触到更底层的代码修改，只是在配置文件中进行修改，但也让我从整体上了解了如何为将鸿蒙 liteos 移植到其他开发板上做最基础的增加选项，为后续的移植奠定坚实的基础。

4.遇到的困难及解决方法

在编译内核时报错，提示缺少头文件：

```
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ vim shell/full/src/base/shcmd.c
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ vim shell/full/src/base/shmsg.c
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$ make -j 16
/home/book/openharmony/kernel/liteos_a/platform/board.ld.S:32:10: fatal error: 'include/board.h' file not found
#include "include/board.h"
         ^~~~~~~~~~~~~~~~~
1 error generated.
Makefile:126: recipe for target '/home/book/openharmony/kernel/liteos_a/out/demochip' failed
make: *** [/home/book/openharmony/kernel/liteos_a/out/demochip] Error 1
book@ry-virtual-machine:~/openharmony/kernel/liteos_a$
```

解决方法：添加课程资料中的补丁，之后再编译即可通过。