

# 实验3 鸿蒙 LiteOS-a 内核移植——增加一个单板

22920212204396

黄子安

## 一、实验目的

1、向鸿蒙 LiteOS-a 中添加一个单板，对内核编译做进一步了解

## 二、实验环境

- 1、物理机 Windows11
- 2、虚拟机 Ubuntu18.04

## 三、实验内容

### 0.单板概念

单板是指集成了处理器、存储器、各种接口和外设的一个完整的电路板，可以独立运行，也就是可以把手里的开发板理解成一个单板。向源码中增加一个单板就是使得源码可以在这个单板硬件平台上运行。

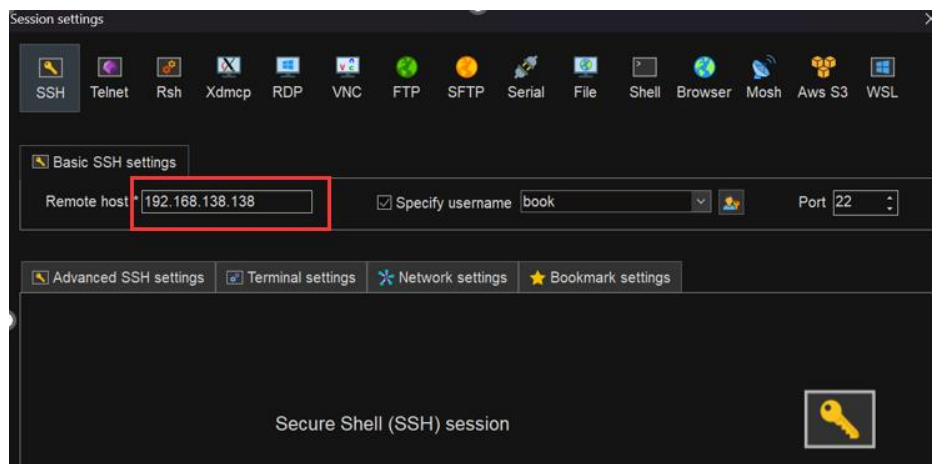
### 1.在配置界面里添加单板

先打开 Liteos 单板的配置界面，这里如果直接在虚拟机中打开会因为屏幕大小问题显示报错，尝试拉伸屏幕后发现无法直接解决问题，因此通过 Xterm 使用 SSH 来连接虚拟机进行操作

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

book@hza-virtual-machine:~/openharmony/kernel/liteos_a$ make menuconfig
/home/book/openharmony/kernel/liteos_a/tools/menuconfig/mconf /home/book/openharmony/kernel/liteos_a/Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
Makefile:140: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 1
```

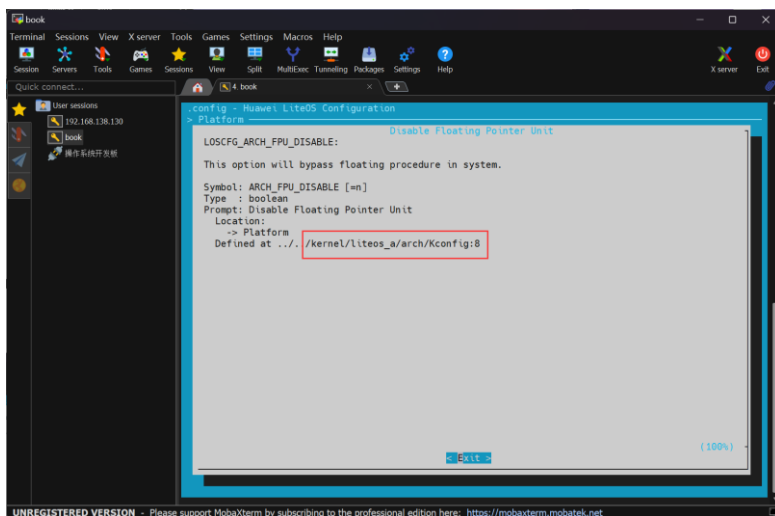
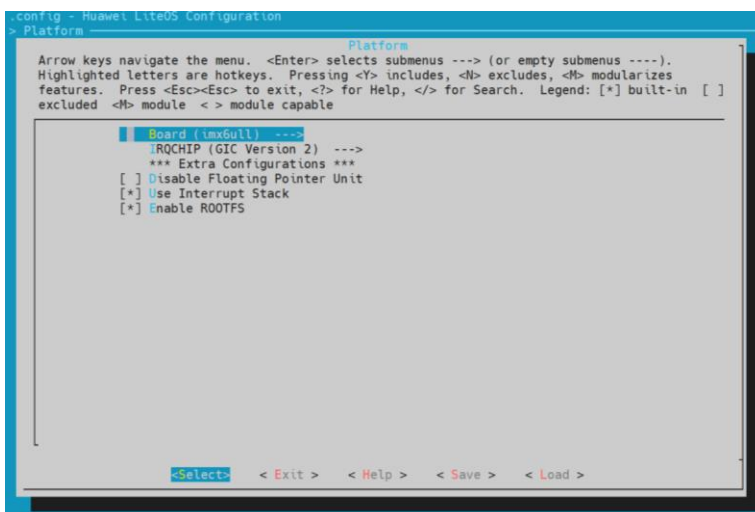
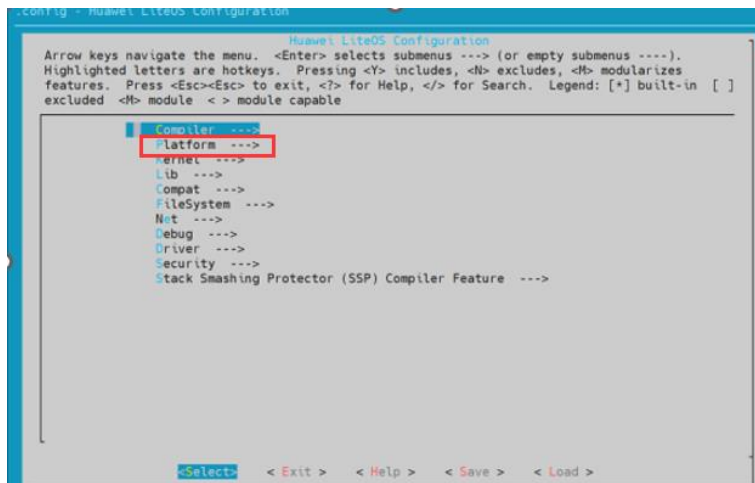
配置会话信息，填入虚拟机的 IP 地址和连接的用户，之后创建该连接会话



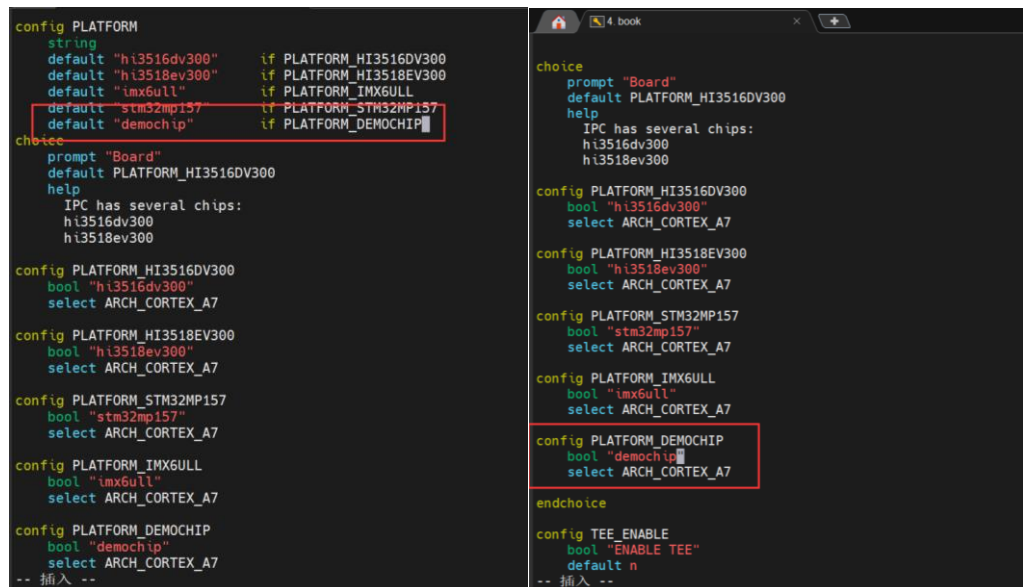
连接成功打开配置界面

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

在配置界面中查看单板信息，选择单板后按 **h** 可以在 **helper** 中看到配置界面中单板被定义在 **Kconfig** 文件当中，我们需要在该文件当中添加新的单板显示信息



在终端中输入 `vi platform/Kconfig` 使用 vim 编辑器打开该文件，按 `e` 进入编辑模式，之后仿照其他的几个单板信息添加新的单板信息，按 `Esc` 退出编辑模式，再输入 `：“wq”` 保存并退出



```
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

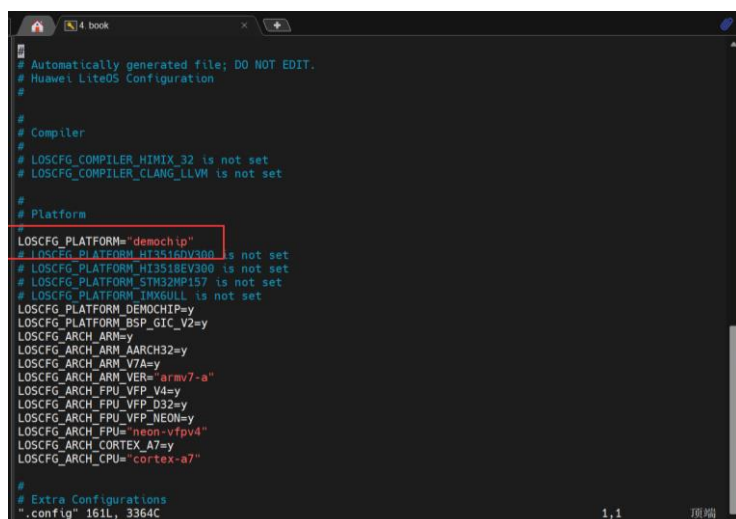
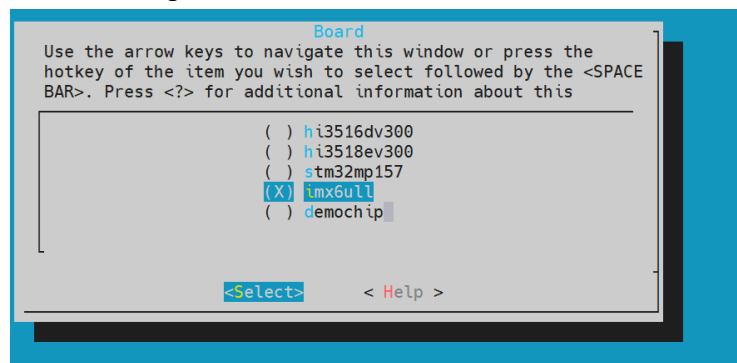
config PLATFORM_IMX6ULL
bool "imx6ull"
select ARCH_CORTEX_A7

config PLATFORM_DEMOCHIP
bool "demochip"
select ARCH_CORTEX_A7

endchoice

config TEE_ENABLE
bool "ENABLE TEE"
default n
```

之后就能在配置界面看到我们新增加的单板信息，但是这个时候单板没有源码，只是在显示配置界面上了这一项显示，如果我们选择了 `demochip` 可以在 `.config` 中看到 `demochip` 被选中而别的单板显示没有被设置



```
# Automatically generated file; DO NOT EDIT.
# Huawei LiteOS Configuration
#
# Compiler
#
# LOSCFG_COMPILER_HIMIX_32 is not set
# LOSCFG_COMPILER_CLANG_LLVM is not set
#
# 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
LOSCFG_ARCH_ARM=y
LOSCFG_ARCH_ARM_AARCH32=y
LOSCFG_ARCH_ARM_V7A=y
LOSCFG_ARCH_ARM_VER="armv7-a"
LOSCFG_ARCH_FPU_VFP_V4=y
LOSCFG_ARCH_FPU_VFP_D32=y
LOSCFG_ARCH_FPU_VFP_NEON=y
LOSCFG_ARCH_FPU="neon-vfpv4"
LOSCFG_ARCH_CORTEX_A7=y
LOSCFG_ARCH_CPU="cortex-a7"
#
# Extra Configurations
.config" 161L, 3364C
```

## 2.修改 makefile 文件

Makefile 文件提供了对应源程序的编译信息，增加单板时为了正确编译源文件需要对相关 makefile 文件进行修改，这里为了使编译正确需要修改三个 makefile 文件，分别是顶层的 makefile 文件、platform 中的 makefile 文件和 platform 中的 bsp.mk 文件，可以类比与其他的单板参照上下文添加对应的编译信息

顶层 makefile 文件需要修改两个地方，第一处指定了后续要包含的头文件的文件路径信息，第二处与文件信息有关

```
#-----need move when make version-----#
##### make lib #####
$(LIBS): $(OUT) $(CXX_INCLUDE)

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 := $(LITEOSTOPDIR)/../vendor/denocn/denochip/board
else
BOARD_INCLUDE_DIR := $(LITEOSTOPDIR)/../vendor/hist/hl35xx/$(LITEOS_PLATFORM)/config/board
endif

$(OUT): $(LITEOS_MENUCONFIG_H)
$(HIDE)mkdir -p $(OUT)/lib
$(HIDE)$(CC) -IS(LITEOS_PLATFORM_BASE)/include -IS(BOARD_INCLUDE_DIR) \
-E $(LITEOS_PLATFORM_BASE)/board.ld.S \
-o $(LITEOS_PLATFORM_BASE)/board.ld -P

$(BUILD):
$(HIDE)mkdir -p $(BUILD)

$(LITEOS_LIBS_TARGET): $(LIBS)
$(HIDE)for dir in $(LIB_SUBDIRS); \
do $(MAKE) -C $dir all || exit 1; \

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 =

all: $(OUT) $(BUILD) $(LITEOS_TARGET) $(APPS)
lib: $(OUT) $(BUILD) $(LITEOS_LIBS_TARGET)

help:
$(HIDE)echo "-----"
$(HIDE)echo "I.==make help: get help information of make"
```

platform 中的 makefile 文件有一处需要修改

```
LOCAL_INCLUDE += -I $(LITEOSTOPDIR)/compat/posix/src \
-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
else
LOCAL_SRCS += $(wildcard ../kernel/common/los_config.c)
LOCAL_SRCS += $(wildcard ../kernel/common/los_printf.c)
endif

ifeq ($(LOSCFG_DRIVERS_USB), y)
LOCAL_SRCS += $(wildcard $(USB_SRC)/*.c)
endif
```

最后 platform 中的 bsp.mk 文件需要修改两处

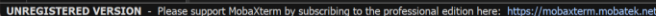
```
##### HI3518EV300
Options#####
else ifeq ($(LOSCFG_PLATFORM_HI3518EV300),y)
HWI_TYPE := arm/interrupt/gic
TIMER_TYPE := hisoc/timer
HRTIMER_TYPE := hisoc/hrtimer
NET_TYPE := hieth
UART_TYPE := amba_pl011
USB_TYPE := usb3.0_hi3518ev300
LITEOS_CMACHRO_TEST += -DTEST3518EV300
else ifeq ($(LOSCFG_PLATFORM_IMX6ULL),y)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

-I $(PLATFORM_BSP_HISI_BASE)/include \
-I $(PLATFORM_BSP_HISI_BASE)/$(UART_SRC)

ifeq ($(findstring y, $(LOSCFG_PLATFORM_HI3518EV300)$
(LOSCFG_PLATFORM_HI3516DV300),y)
PLATFORM_INCLUDE += -I $(LITEOSTOPDIR)/../vendor/hist/hl35xx/$(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/denocn/demochip/board/include
endif

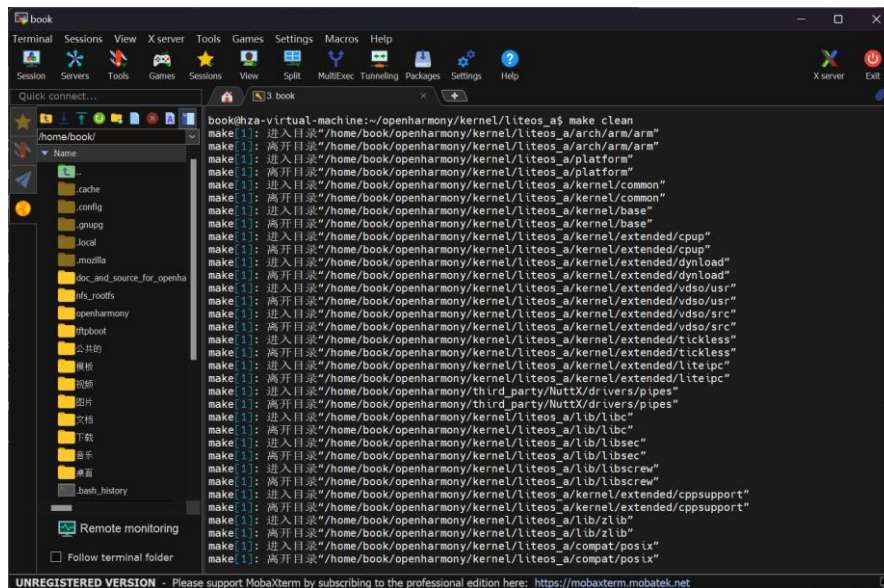
#
#-include $(LITEOSTOPDIR)/platform/bsp/board/$(LITEOS_PLATFORM)/board.mk
#
```

器





之后使用 `make clean` 清除之前编译生成的文件



再使用 `make -j 8` 命令进行编译，此时会出现一个视频中没有出现的报错情况，提示没有找到头文件

```
/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:120: recipe for target '/home/book/openharmony/kernel/liteos_a/out/demochip' failed
make: *** [/home/book/openharmony/kernel/liteos_a/out/demochip] Error 1
book@hza-virtual-machine:~/openharmony/kernel/liteos_a$
```

根据报错信息的内容以及前面 `makefile` 文件的修改信息可以推测出原因是在 `/vendor` 目录缺少了对应的源代码，因此导致找不到头文件故没法完成编译，该部分视频教程跳了十分钟，增加源码的部分缺失了，网上也没有找到相关的教程，只能猜测该过程，好在通过后续教程的验证这里的源码就是复制了别的单板文件

```
#----need move when make version----#
#### make lib ####
$(__LIBS): $(OUT) $(CXX_INCLUDE)

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 := $(LITEOSTOPDIR)/../../vendor/democom/demochip/board
else
BOARD_INCLUDE_DIR := $(LITEOSTOPDIR)/../../vendor/hisi/hi35xx/$(LITEOS_PLATFORM)/config/board
endif

$(OUT): $(LITEOS_MENUCONFIG_H)
$(HIDE)mkdir -p $(OUT)/lib
$(HIDE)$@ $(CC) -I$(LITEOS_PLATFORM_BASE)/include -I$(BOARD_INCLUDE_DIR) \
-E $(LITEOS_PLATFORM_BASE)/board.ld.S \
-o $(LITEOS_PLATFORM_BASE)/board.ld -P

$(BUILD):
$(HIDE)mkdir -p $(BUILD)

$(LITEOS_LIBS_TARGET): $(__LIBS)
$(HIDE)for dir in $(LIB_SUBDIRS); \
do $(MAKE) -C $$dir all || exit 1; \
done
```

### 3.增加源代码

这里先找到了实验文档一开始说的 `patch` 文件，通过虚拟机保存屏幕快照的方式保存了当前状态，再使用 `patch` 文件打补丁看看完成后的源代码情况，之后进行回退操作。

使用提供的文档打完补丁后可以发现在 `demochip` 文件里的 `driver` 是 `stm32mp157` 单板的内容，因此我们也直接将该文件内容复制到 `demochip` 里面



具体操作如下，复制 stm32mp157 单板的内容到新建的 democom 文件夹里，然后将文件名修改为 demochip

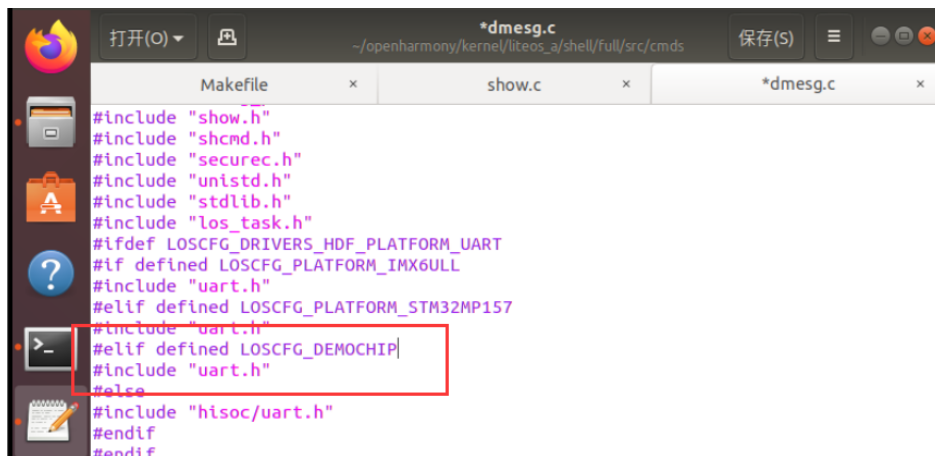


再执行编译可以寻找到 **board** 头文件并完成部分编译，之后出现的编译错误与视频中的相同

```
book@hza-virtual-machine:~$ cd ./openharmony/kernel/liteos_a
book@hza-virtual-machine:~/openharmony/kernel/liteos_a$ make -j 8
make[1]: 进入目录`/home/book/openharmony/kernel/liteos_a/arch/arm/arm`
make[1]: 对`all`无需做任何事。
make[1]: 离开目录`/home/book/openharmony/kernel/liteos_a/arch/arm/arm`
make[1]: 进入目录`/home/book/openharmony/kernel/liteos_a/platform`
make[1]: 离开目录`/home/book/openharmony/kernel/liteos_a/platform`
make[1]: 进入目录`/home/book/openharmony/kernel/liteos_a/kernel/common`
make[1]: 对`all`无需做任何事。
make[1]: 离开目录`/home/book/openharmony/kernel/liteos_a/kernel/common`
make[1]: 进入目录`/home/book/openharmony/kernel/liteos_a/kernel/base`
make[1]: 对`all`无需做任何事。
make[1]: 离开目录`/home/book/openharmony/kernel/liteos_a/kernel/base`
make[1]: 进入目录`/home/book/openharmony/kernel/liteos_a/kernel/extended/cpup`
make[1]: 对`all`无需做任何事。
make[1]: 离开目录`/home/book/openharmony/kernel/liteos_a/kernel/extended/cpup`
make[1]: 进入目录`/home/book/openharmony/kernel/liteos_a/kernel/extended/dynload`
make[1]: 对`all`无需做任何事。
make[1]: 离开目录`/home/book/openharmony/kernel/liteos_a/kernel/extended/dynload`
make[1]: 进入目录`/home/book/openharmony/kernel/liteos_a/kernel/extended/vdso/usr`
make[1]: 离开目录`/home/book/openharmony/kernel/liteos_a/kernel/extended/vdso/usr`
make[1]: 进入目录`/home/book/openharmony/kernel/liteos_a/kernel/extended/vdso/src`
make[1]: 对`all`无需做任何事。
make[1]: 离开目录`/home/book/openharmony/kernel/liteos_a/kernel/extended/vdso/src`
```

提示某些文件里出现了找不到包含头文件的情况，查看对应的.c 文件可以发现这些文件中如果没有定义的单板则会包含“hisoc/uart.h”文件，因此我们需要在这五个文件中增加一个 elif 判断语句将我们定义的单板加进去

```
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/shell"
full/src/base/shmsg.c:50:10: fatal error: 'hisoc/uart.h' file not found
#include "hisoc/uart.h"
^~~~~~
1 error generated.
/home/book/openharmony/kernel/liteos_a/tools/build/mk/module.mk:83: recipe for target '/home/book/openharmony/kernel/liteos_a/out/demochip/obj/shell/full/src/base/shmsg.o' failed
make[1]: *** [/home/book/openharmony/kernel/liteos_a/out/demochip/obj/shell/full/src/base/shmsg.o] Error 1
make[1]: *** 正在等待未完成的任务....
full/src/base/shcmd.c:47:10: fatal error: 'hisoc/uart.h' file not found
#include "hisoc/uart.h"
^~~~~~
1 error generated.
/home/book/openharmony/kernel/liteos_a/tools/build/mk/module.mk:83: recipe for target '/home/book/openharmony/kernel/liteos_a/out/demochip/obj/shell/full/src/base/shcmd.o' failed
make[1]: *** [/home/book/openharmony/kernel/liteos_a/out/demochip/obj/shell/full/src/base/shcmd.o] Error 1
full/src/cmds/dmesg.c:68:10: fatal error: 'hisoc/uart.h' file not found
#include "hisoc/uart.h"
^~~~~~
1 error generated.
full/src/base/show.c:43:10: fatal error: 'hisoc/uart.h' file not found
#include "hisoc/uart.h"
^~~~~~
1 error generated.
/home/book/openharmony/kernel/liteos_a/tools/build/mk/module.mk:83: recipe for target '/home/book/openharmony/kernel/liteos_a/out/demochip/obj/shell/full/src/cmds/dmesg.o' failed
make[1]: *** [/home/book/openharmony/kernel/liteos_a/out/demochip/obj/shell/full/src/cmds/dmesg.o] Error 1
```





活动 文本编辑器 星期三 21:08 zh 保存(S)

Makefile x show.c x dmesg.c x \*shcmd.c x

```
#include "stdlib.h"
#include "unistd.h"
#include "dirent.h"
#include "securec.h"
#include "los_mux.h"
#include "los_memory.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
```

活动 文本编辑器 星期三 21:10 zh 保存(S)

\*dmesg.c

```
#####|#####
+-----+-----+
| Tail | Head |
*/
#include "sys_config.h"
#ifdef LOSCFG_SHELL_DMESG
#include "dmesg_pri.h"
#include "show.h"
#include "shcmd.h"
#include "securec.h"
#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
```

活动 文本编辑器 星期三 21:08 zh 保存(S)

Makefile x show.c x dmesg.c x shcmd.c x \*shmsg.c x

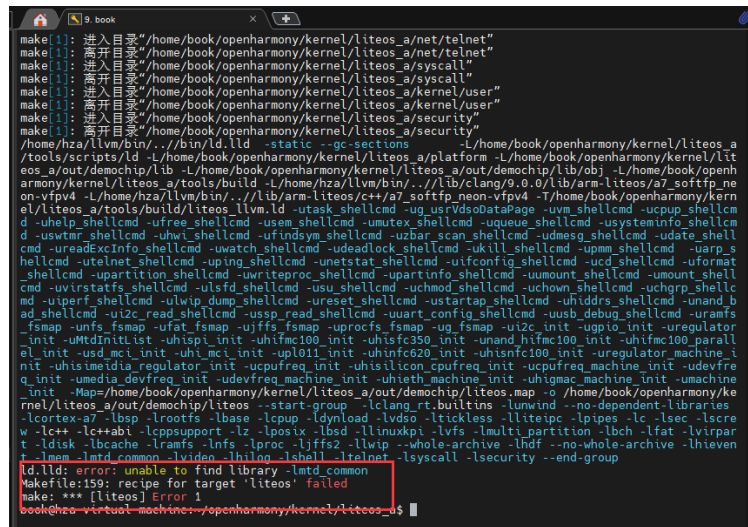
```
#include "shcmd.h"
#include "stdlib.h"
#include "stdio.h"
#include "unistd.h"
#include "securec.h"
#include "los_base.h"
#include "los_task.h"
#include "los_event.h"
#include "los_list.h"
#include "los_printf.h"
#ifdef LOSCFG_DRIVERS_HDF_PLATFORM_UART
#if defined LOSCFG_PLATFORM_IMX6ULL
#include "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

#ifdef LOSCFG_FS_VFS
#include "console.h"
#endif
```


C 制表符宽度: 8 第 50 行, 第 18 列 插入

## 4. 解决链接错误

之后继续编译发现编译成功，但是出现链接错误，根据视频教程定位到对应的文件中进行修改



```
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/net/telnet"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a/net/telnet"
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/syscall"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a/syscall"
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/kernel/user"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a/kernel/user"
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/security"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a/security"
/home/hza/llvm/bin/../bin/ld.lld -static --gc-sections -L/home/book/openharmony/kernel/liteos_a
/tools/scripts/ld -L/home/book/openharmony/kernel/liteos_a/platform -L/home/book/openharmony/kernel/liteos_a/out/demochip/lib/obj -L/home/book/openh
army/kernel/liteos_a/tools/build -L/home/hza/llvm/bin/../lib/clang/9.0.0/lib/arm-liteos/a7_softfp.ne
on-vfpv4 -L/home/hza/llvm/bin/../lib/arm-liteos/c++/a7_softfp_neon-vfpv4 -T/home/book/openharmony/kern
el/liteos_a/tools/build/liteos.llvm.ld -utask_shellcmd -ug_usrVdsaDataPage -uvm_shellcmd -ucupp_shellc
md -uhelp_shellcmd -ufree_shellcmd -usem_shellcmd -umutex_shellcmd -uqueue_shellcmd -usysteminfo_shellc
md -uvmemr_shellcmd -uhwrt_shellcmd -ufindsym_shellcmd -uzbar_scan_shellcmd -udmesg_shellcmd -udate_shel
lcmd -ureadExInfo_shellcmd -uwatch_shellcmd -udeadlock_shellcmd -ukill_shellcmd -upmm_shellcmd -uarp_s
hellcmd -utelnet_shellcmd -uping_shellcmd -unetstat_shellcmd -uifconfig_shellcmd -ucd_shellcmd -uformat
_shellcmd -upartition_shellcmd -uwriteproc_shellcmd -upartinfo_shellcmd -uumount_shellcmd -umount_shel
lcmd -uvirstatfs_shellcmd -ulsfd_shellcmd -usu_shellcmd -uchmod_shellcmd -uchown_shellcmd -uchgrp_shel
lcmd -uperf_shellcmd -ulwp_dump_shellcmd -ureset_shellcmd -ustartap_shellcmd -uhidrs_shellcmd -unand_b
ad_shellcmd -uizc_read_shellcmd -ussp_read_shellcmd -uuart_config_shellcmd -uusd_debug_shellcmd -urams
fsmmap -unfs_fsmmap -ufat_fsmmap -ujifs_fsmmap -uprocs_fsmmap -ug_fsmmap -ui2c_init -ugpio_init -uregulator
_init -umtdInitList -uhispi_init -uhifmc100_init -uhisfc350_init -unand_hifmc100_init -uhifmc100_parall
el_init -usd_mci_init -uhi_mci_init -upl011_init -uhifmc200_init -uhisfc100_init -uregulator_machine_i
nit -uhismedia_regulator_init -ucupfreq_init -uhisilicon_cpufreq_init -ucupfreq_machine_init -udevfre
q_init -umedia_devfreq_init -udevfreq_machine_init -uhiseth_machine_init -uhigmac_machine_init -umachin
e_init -Map=/home/book/openharmony/kernel/liteos_a/out/demochip/liteos.map -o /home/book/openharmony/ke
rnel/liteos_a/out/demochip/liteos --start-group -lclang_rt.builtins -lunwind --no-dependent-libraries
-lcortex-a7 -lbasp -lroots -lbase -lcpup -ldynload -lvdso -ltickless -lliteipc -lpipes -lc -lsec -lscre
w -lc++ -lc++abi -lcppsupport -lz -lposix -lbsd -llinuxkpi -lvfs -lmulti_partition -lbch -lfat -lvirpar
t -ldisk -lbcache -lramfs -lnfs -lproc -ljffs2 -llwp --whole-archive -lhd -no-whole-archive -lhiven
t -lmon -lmtc-common -lvideo -lhiblog -lshell -lsyscall -lsecurity --end-group
ld.lld: error: unable to find library -lmtc-common
Makefile:159: recipe for target 'liteos' failed
make: *** [liteos] Error 1
```



```
#####
LITEOS_BASELIB += -lbase
LIB_SUBDIRS += kernel/base
LITEOS_KERNEL_INCLUDE := -I $(LITEOSTOPDIR)/kernel/include

ifeq ($(LITEOS_PLATFORM), imx6ull)
include $(LITEOSTOPDIR)/../../vendor/nxp/imx6ull/imx6ull.mk
endif

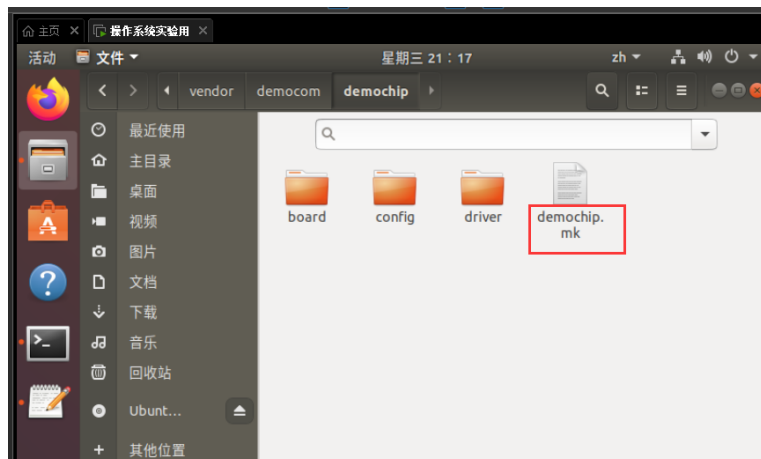
ifeq ($(LITEOS_PLATFORM), stm32mp157)
include $(LITEOSTOPDIR)/../../vendor/st/stm32mp157/stm32mp157.mk
endif

ifeq ($(LITEOS_PLATFORM), demochip)
include $(LITEOSTOPDIR)/../../vendor/democom/demochip/demochip.mk
endif

ifeq $(findstring y, $(LOSCFG_PLATFORM_HI3518EV300))$
(LOSCFG_PLATFORM_HI3516DV300)), y)
LITEOS_BASELIB += -lhi35xx_bsp
LIB_SUBDIRS += $(LITEOSTOPDIR)/../../vendor/hi35xx/$(LITEOS_PLATFORM)/
config/board/
endif

ifeq ($(LOSCFG_KERNEL_CPUP), y)
LITEOS_BASELIB += -lcoun
```

修改 vendor/democom/demochip 的 mk 文件名



修改该文件中包含的路径信息，使得编译的时候进入正确的文件夹当中，注意这里要把文件中的所有路径信息（后续共有 8 处，可以使用 **ctrl+F** 查找）都进行修改



```
DEMOCHIP_BASE_DIR := $(LITEOSTOPDIR)/../vendor/democom/demochip

LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/board
LITEOS_BASELIB += -lboard

LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/mtd/common
LITEOS_BASELIB += -lmtd_common

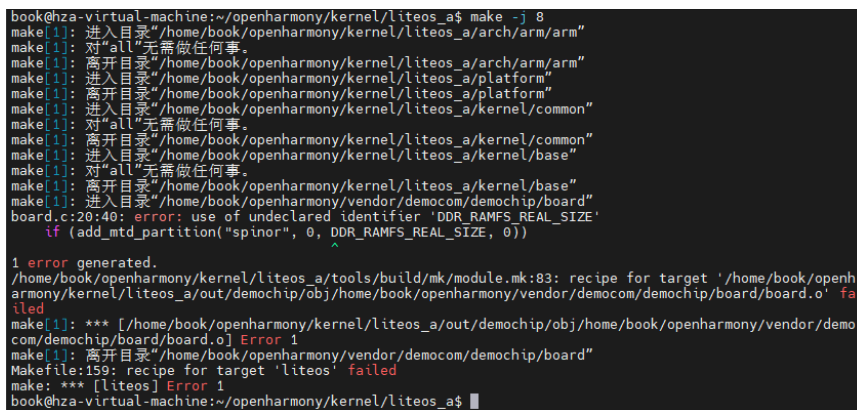
LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/mtd/spi_nor
LITEOS_BASELIB += -lspinor_flash

ifeq ($(LSCFG_DRIVERS_VIDEO), y)
LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/stm32mp157-fb
LITEOS_BASELIB += -lstm32mp157-fb
endif

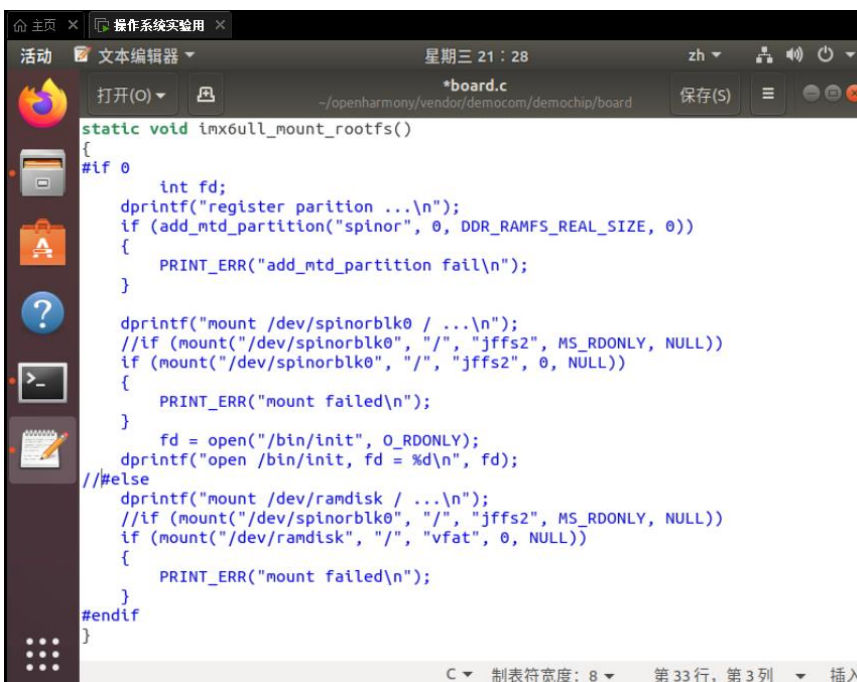
LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/stm32mp157-uart
LITEOS_BASELIB += -lstm32mp157-uart

LITEOS_MTD_SPI_NOR_INCLUDE += -I$(DEMOCHIP_BASE_DIR)/driver/mtd/common/
include \
$(DEMOCHIP_BASE_DIR)/driver/mtd/spi_nor/include
-I$
```

再次编译后出现宏定义缺失的错误，参考视频教程中的方法直接将出现该宏的地方设置为 0 或进行注释

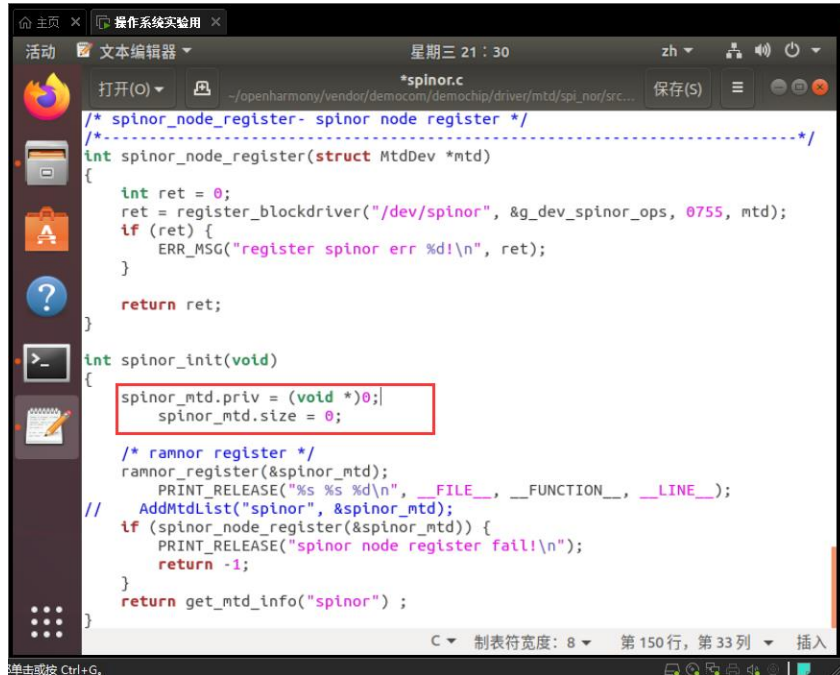


```
book@hza-virtual-machine:~/openharmy/kernel/liteos_a$ make -j 8
make[1]: 进入目录 "/home/book/openharmy/kernel/liteos_a/arch/arm/arm"
make[1]: 对 "all" 无需做任何事。
make[1]: 离开目录 "/home/book/openharmy/kernel/liteos_a/arch/arm/arm"
make[1]: 进入目录 "/home/book/openharmy/kernel/liteos_a/platform"
make[1]: 离开目录 "/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"
board.c:20:40: error: use of undeclared identifier 'DDR_RAMFS_REAL_SIZE'
    if (add_mtd_partition("spinor", 0, DDR_RAMFS_REAL_SIZE, 0))
                                   ^
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/board.o' failed
make[1]: *** [/home/book/openharmy/kernel/liteos_a/out/demochip/obj/home/book/openharmy/vendor/democom/demochip/board/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$
```



```
static void imx6ull_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
    {
        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");
        }
    }
}
#endif
}
```



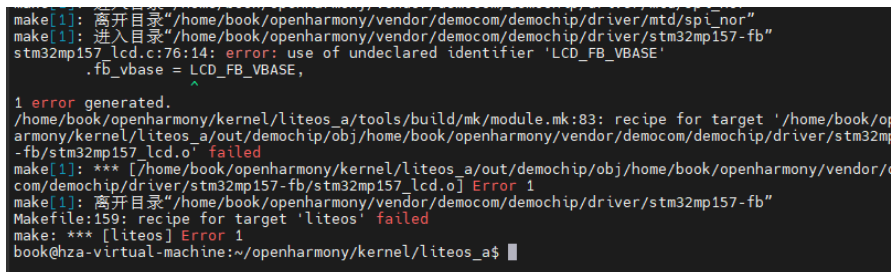
```
/* spinor_node_register- spinor node register */
/*-----*/
int spinor_node_register(struct MtdDev *mtd)
{
    int ret = 0;
    ret = register_blockdriver("/dev/spinor", &g_dev_spinor_ops, 0755, mtd);
    if (ret) {
        ERR_MSG("register spinor err %d!\n", ret);
    }

    return ret;
}

int spinor_init(void)
{
    spinor_mtd.priv = (void *)0;
    spinor_mtd.size = 0;

    /* ramnor register */
    ramnor_register(&spinor_mtd);
    PRINT_RELEASE("%s %s %d\n", __FILE__, __FUNCTION__, __LINE__);
    // AddMtdList("spinor", &spinor_mtd);
    if (spinor_node_register(&spinor_mtd)) {
        PRINT_RELEASE("spinor node register fail!\n");
        return -1;
    }
    return get_mtd_info("spinor");
}
```

编译后出现了一个视频中没有出现过的错误，参照视频里的做法将出现该宏的地方设为 0 或者进行注释，编译后该部分报错信息消失。推测该宏可能和开发板的显示屏有关，暂时不确定是否会对后续实验产生影响



```
make[1]: 离开目录"/home/book/openharmony/vendor/democom/demochip/driver/mtd/spi_nor"
make[1]: 进入目录"/home/book/openharmony/vendor/democom/demochip/driver/stm32mp157-fb"
stm32mp157_lcd.c:76:14: error: use of undeclared identifier 'LCD_FB_VBASE'
    .fb_vbase = LCD_FB_VBASE,
                  ^
1 error generated.
/home/book/openharmony/kernel/liteos_a/tools/build/mk/module.mk:83: recipe for target '/home/book/openharmony/kernel/liteos_a/out/demochip/obj/home/book/openharmony/vendor/democom/demochip/driver/stm32mp157-fb/stm32mp157_lcd.o' failed
make[1]: *** [/home/book/openharmony/kernel/liteos_a/out/demochip/obj/home/book/openharmony/vendor/democom/demochip/driver/stm32mp157-fb/stm32mp157_lcd.o] Error 1
make[1]: 离开目录"/home/book/openharmony/vendor/democom/demochip/driver/stm32mp157-fb"
Makefile:159: recipe for target 'liteos' failed
make: *** [liteos] Error 1
book@hza-virtual-machine:~/openharmony/kernel/liteos_a$
```



```
spinor.c x stm32mp157_lcd.c x
.tvb= 20, /* 上边黑框, Vertical Back porch */
.tvf= 12, /* 下边黑框, Vertical Front porch */

/* 水平方向 */
.thp= 20, /* hsync脉冲宽度 */
.thb= 140, /* 左边黑框, Horizontal Back porch */
.thf= 160, /* 右边黑框, Horizontal Front porch */

.vclk= 51, /* MHz */
},
.xres= 1024,
.yres= 600,
.bpp= 16,
.fb_base = LCD_FB_BASE,
.fb_vbase = 0, //LCD_FB_VBASE,
};

int up_fbinitialize(int display)
{
    PRINT_RELEASE("%s %s %d\n", __FILE__, __FUNCTION__, __LINE__);
    Imx6ull_lcd_controller_init(&lcd_7_0_params);
    PRINT_RELEASE("%s %s %d\n", __FILE__, __FUNCTION__, __LINE__);
    return 0;
}
```



之后再次编译依旧出现报错，根据视频中的方法定位到对应出错的文件和包含该文件的 `makefile` 文件中进行修改

```

make[1]: 离开目录 "/home/book/openharmony/kernel/ltteos_a/security"
/home/hzsa/vllvm/bin/.../bin/ld.lld -static -gc-sections -L/home/book/openharmony/kernel/ltteos_a/
/home/hzsa/vllvm/bin/.../bin/ld.lld -L/home/book/openharmony/kernel/ltteos_a/platform -L/home/book/openharmony/kernel/ltteos_a/out/demochip/lib -L/home/book/openharmony/kernel/ltteos_a/out/demochip/lib/obj -L/home/book/openharmony/kernel/ltteos_a/out/demochip/lib/obj/arm -L/home/book/openharmony/kernel/ltteos_a/out/vfpv4 -L/home/hzsa/vllvm/bin/.../lib/arm-ltteos/cv4/v7 target-neon-vfpv4 -T/home/book/openharmony/kernel/ltteos_a/tools/build/ltteos.vllvm.ld -utask shellcmd -ug usVdsODataPage -uvm shellcmd -ucupnp shellcmd -d -uhelp shellcmd -ufree shellcmd -usm shellcmd -umutex shellcmd -uqueue shellcmd -usysteminfo shellcmd -d -usmrtm shellcmd -uhwi shellcmd -ufindsym shellcmd -uzbar_scan shellcmd -uadmesg shellcmd -update shellcmd -cmd -ureadExInfo shellcmd -uwatch shellcmd -udeadlock shellcmd -ukill shellcmd -upmm shellcmd -uarp shellcmd -uinetnet shellcmd -upring shellcmd -unetstat shellcmd -uifconfig shellcmd -ucd shellcmd -uformat shellcmd -upartition shellcmd -uwirteproc shellcmd -upartinfo shellcmd -uumount shellcmd -umount shellcmd -cmd -uivirstats shellcmd -ulufs shellcmd -usu shellcmd -uchmod shellcmd -uchown shellcmd -uchgrp shellcmd -md -uiperf shellcmd -ulwip_dum shellcmd -ureset shellcmd -ustartup shellcmd -uhidrs shellcmd -unand -b ad shellcmd -u12c read shellcmd -ussp read shellcmd -uuart_config shellcmd -usb debug shellcmd -uramfs fsmap -unfs fsmap -ufat fsmap -ujffs fsmap -uproofs fsmap -ug fsmap -u12c init -uugpio init -uregulator init -umtdinit init -uhci init -uhfmc100 init -uhisfc320 init -uand.hifmc100 init -uhifmc100 parallel init -uact init -uhpi init -u100 init -uhifmc20 init -uhisfc100 init -uregulator machine init -u100 init -u100 init -uupgrate init -uupgrate init -uhislon cmufrag init -ucpufrag machine init -udevfrag q init -umedia devfreq init -udevfreq machine init -uhieth machine init -uhigmac machine init -umachine init -Map=/home/book/openharmony/kernel/ltteos_a/out/demochip/ltteos.map.o /home/book/openharmony/kernel/ltteos_a/out/demochip/ltteos -start-group -lclang_rt.builtins -lunwind -no-dependent-libraries -lcortex-a7-lbnp -lroofs -lbase -lboard -lmtm -lcm -lspinor -lflash -lstm32mp157-fb -lstm32mp157-uart -lcpu -ldynload -lvdsio -ltimeless -llitepic -lipes -lc -lsec -lscrew -lcc -lct+abi -lcppsupp -lzl -lposix -lbstd -llinuxkpi -lvfs -lmulti-partition -lbch -lflash -lvpipart -ldisk -lbbacache -lramfs -lnfs -lproc -liffs2 -llwip -lwhole-archive -lhdf -no-whole-archive -lhiviewt -lmem -lmtm -lcm -lvideo -lhislog -lshell -ltsnet -lhwcall -lsecurity -lgroup
ld.lld: error: undefined symbol: HdfGetBuildInConfigData
referenced by ld-temp.o
>>>
ld.lld: error: undefined symbol: HcsGetRootNode
makefile:159: recipe for target 'ltteos_a' failed
make: *** [ltteos_a] Error 1
book@hza-virtual-machine:~/openharmony/kernel/ltteos_a$

```

```

# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

LITEOS_BASELIB += -lhdf_config
LIB_SUBDIRS += $(LITEOS_SOURCE_ROOT)/vendor/democom/$(LITEOS_PLATFORM)/
config

VENDOR_HDF_DRIVERS_ROOT := $(LITEOSTOPDIR)/../vendor/democom/hdf

ifeq ($(LSCFG_PLATFORM_DEMOCHIP), y)
#LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/stm32mp157-i2c
#LITEOS_BASELIB += -lstm32mp157-i2c
endif

#LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/touch
#LITEOS_BASELIB += -ltouch

LIB_SUBDIRS += $(DEMOCHIP_BASE_DIR)/driver/hello
LITEOS_LD_PATH += -L$(VENDOR_HDF_DRIVERS_ROOT)/libs/$(LITEOS_PLATFORM)

# lib path
LITEOS_LD_PATH += -L$(VENDOR_HDF_DRIVERS_ROOT)/libs/$(LITEOS_PLATFORM)

```

```
活动 文本编辑器 星期三 21:45 zh 保存(S)
*hdf_lite.mk
~/openharmony/drivers/hdf/lite

spinor.c x stm32mp157_lcd.c x *hdf_vendor.mk x *hdf_lite.mk
LITEOS_DRIVERS_HDF_INCLUDE += -I $(LITEOS_DRIVERS_HDF)/model/bus/usb/include
LITEOS_BASELIB += -lhdf_usb
LIB_SUBDIRS += $(LITEOS_DRIVERS_HDF)/model/bus/usb
endif

# vendor lib
ifeq ($(findstring y, $(LOSCFG_PLATFORM_HI3518EV300)) $(LOSCFG_PLATFORM_HI3516DV300)), y)
include $(LITEOSTOPDIR)/../../vendor/huawei/hdf/hdf_vendor.mk
endif

ifeq ($(LOSCFG_PLATFORM_IMX6ULL), y)
include $(LITEOSTOPDIR)/../../vendor/nxp/hdf/hdf_vendor.mk
endif

ifeq ($(LOSCFG_PLATFORM_STM32MP157), y)
include $(LITEOSTOPDIR)/../../vendor/st/hdf/hdf_vendor.mk
endif

ifeq ($(LOSCFG_PLATFORM_DEMOCHIP), y)
include $(LITEOSTOPDIR)/../../vendor/democom/hdf/hdf_vendor.mk
endif

LITEOS_BASELIB += --no-whole-archive
endif
```



最后进行编译链接没有出现报错，说明添加单板成功，复制一份配置文件保存实验的结果为后续的实验做一些基础

```
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a"
clean demochip finish
book@hza-virtual-machine:~/openharmony/kernel/liteos_a$ make -j 16
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a"
mv -f /home/book/openharmony/kernel/liteos_a/tools/menueconfig/conf --silentoldconfig /home/book/openharmony/k
ernel/liteos_a/kconfig
mv -f /home/book/openharmony/kernel/liteos_a/include/generated/autoconf.h /home/book/openharmony/kernel
/liteos_a/platform/include/menueconfig.h
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a"
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/arch/arm/arm"
src/startup/reset_vector_up.S:138:2: warning: deprecated since v7, use 'dsb'
mcr p15, 0, r0, c7, c10, 4 @ DSB
^
src/startup/reset_vector_up.S:139:2: warning: deprecated since v7, use 'isb'
mcr p15, 0, r0, c7, c5, 4 @ ISB
^
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a/arch/arm/arm"
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/platform"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a/platform"
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/kernel/common"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a/kernel/common"
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/kernel/base"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a/kernel/base"
make[1]: 进入目录"/home/book/openharmony/vendor/democom/demochip/board"
make[1]: 离开目录"/home/book/openharmony/vendor/democom/demochip/board"
make[1]: 进入目录"/home/book/openharmony/vendor/democom/demochip/driver/mtcd/common"
make[1]: 离开目录"/home/book/openharmony/vendor/democom/demochip/driver/mtcd/common"
make[1]: 进入目录"/home/book/openharmony/vendor/democom/demochip/driver/mtcd/spi_nor"
make[1]: 离开目录"/home/book/openharmony/vendor/democom/demochip/driver/mtcd/spi_nor"
make[1]: 进入目录"/home/book/openharmony/vendor/democom/demochip/driver/stm32mp157-fb"
make[1]: 离开目录"/home/book/openharmony/vendor/democom/demochip/driver/stm32mp157-fb"
make[1]: 进入目录"/home/book/openharmony/vendor/democom/demochip/driver/stm32mp157-uart"
make[1]: 离开目录"/home/book/openharmony/vendor/democom/demochip/driver/stm32mp157-uart"
```

```
e_shellcmd -usem_shellcmd -umutex_shellcmd -uqueue_shellcmd -usysteminfo_shellcmd -uswtmr_shellcmd -uhw
i_shellcmd -ufindsym_shellcmd -uzbar_scan_shellcmd -udmesg_shellcmd -udate_shellcmd -ureadexcinfo_shell
cmd -uwatch_shellcmd -udeslock_shellcmd -ukill_shellcmd -upmm_shellcmd -uarp_shellcmd -utelnetshellc
md -upping_shellcmd -unetstat_shellcmd -uifconfig_shellcmd -ucd_shellcmd -uformat_shellcmd -upartition_s
hellcmd -uwritetproc_shellcmd -upartinfo_shellcmd -uumount_shellcmd -uvmount_shellcmd -uvirtstats_shellc
md -ulsfd_shellcmd -usu_shellcmd -uchmod_shellcmd -uchown_shellcmd -uchgrp_shellcmd -uiperf_shellcmd -ul
wip_dump_shellcmd -ureset_shellcmd -ustartup_shellcmd -uhidrs_shellcmd -unand_bad_shellcmd -ui2c_read_
shellcmd -ussp_read_shellcmd -uuart_config_shellcmd -uusb_debug_shellcmd -uramfs_fsmmap -unfs_fsmmap -ufa
t_fsmmap -uoffs_fsmmap -uprocs_fsmmap -ug_fsmmap -ui2c_init -ugpio_init -uregulator_init -uMtdInitList -uh
isp1_init -uhifmc100_init -uhisfc350_init -unand_hifmc100_init -uhifmc100_parallel_init -usd_mci_init -u
hi_mci_init -upl011_init -uhifmc620_init -uhisfc100_init -uregulator_machine_init -uhis_uemedia_regula
tor_init -ucpufreq_init -uhisilicon2cpuirq_init -ucpuirq_machine_init -udevfreq_init -umedia_devfreq_
init -udevfreq_machine_init -uhieth_machine_init -uhigmac_machine_init -umachine_init -Map=/home/book/
openharmony/kernel/liteos_a/out/demochip/liteos.map -o /home/book/openharmony/kernel/liteos_a/out/democ
hip/liteos -start-group -lclang_rt.builtins -lunwind --no-dependent-libraries -lcortex-a7 -lbps -lroo
tfs -lbase -lboard -lmtcd.common -lspnord flash -lstm32mp157-fb -lstm32mp157-uart -lcpup -ldynload -lvd
o -ltickless -lliteipc -lpipes -lc -lsec -lscrew -lcpp -lctabi -lcpusupport -lz -lposix -lbsd -llinux
pi -lvfs -lmulti-partition -lbch -lfat -lvirpart -ldisk -lbcache -lramfs -lnfs -lproc -ljffs2 -llwip --
whole-archive -lhdf -lhdf_config -lhello --no-whole-archive -lhievent -lmem -lmtcd.common -lvideo -lhilo
g -lshell -ltnet -lsyscall -lsecurity -end-group
/home/hza/llvm/bin/./bin/llvm-objcopy -R .bss -O binary /home/book/openharmony/kernel/liteos_a/out/de
mochip/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
/home/hza/llvm/bin/./bin/llvm-objdump -d /home/book/openharmony/kernel/liteos_a/out/demochip/liteos >
/home/book/openharmony/kernel/liteos_a/out/demochip/liteos.asm
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/apps"
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/apps/shell"
make[1]: 离开目录"/home/book/openharmony/kernel/liteos_a/apps/shell"
make[1]: 进入目录"/home/book/openharmony/kernel/liteos_a/apps/init"
make[1]: 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$
```

```
book@hza-virtual-machine:~/openharmony/kernel/liteos_a$ cp .config tools/build/config/debug/demochip_cl
ang.config
book@hza-virtual-machine:~/openharmony/kernel/liteos_a$
```

## 四、实验心得

本次实验主要参照视频为主，令我印象比较深的就是老师解决报错的方法，操作系统源码中文件错综复杂，学会看报错信息并有效定位到出错文件进行排错非常重要，这也直接要求了对编译器以及 makefile 文件至少要有的一些初步的了解，比如能看懂 makefile 文件和进行 makefile 中编译选项的解读；另外视频中也展现了 vim 编辑器的简单高效，虽然 vim 编辑器学习难度曲线在前期较大，但是能学会该编辑器将会很方便进行键盘文本编辑操作。