

# 实 验 指 南

|  |  |
| --- | --- |
| 课 程： | 嵌入式系统协同设计 |
| 实验序号： | 第1次实验 |
| 实验名称： | LiteOS-M的开发环境配置实验 |
| 院 系： | 软件学院 |
| 专业班级： | 软件工程2102班 |
| 学 号： | U202117254 |
| 姓 名： | 陈德霆 |

2023年07月07日

**一、实验目的**

（1）熟悉Linux或Windows上LiteOS-M嵌入式开发环境的配置；

（2）熟悉LiteOS-M内核的目录结构和模块；

（3）熟悉LiteOS-M内核的编译、下载和测试流程。

**二、实验内容**

针对华为Hi3861嵌入式CPU开发板配置开发环境，并完成“HelloHuawei”，“HelloWorld”字符串输出；

**三、实验环境**

宿主机操作系统：Ubuntu 20.04LTS

编译工具：gcc/g++

编辑工具：vim/vscode

主要依赖包：

binutils binutils-dev git git-lfs gnupg flex bison gperf build-essential zip curl zlib1g-dev gcc-multilib g++-multilib gcc-arm-linux-gnueabi libc6-dev-i386 libc6-dev-amd64 lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev ccache libgl1-mesa-dev libxml2-utils xsltproc unzip m4 bc gnutls-bin python3.8 python3-pip ruby genext2fs device-tree-compiler make libffi-dev e2fsprogs pkg-config perl openssl libssl-dev libelf-dev libdwarf-dev u-boot-tools mtd-utils cpio doxygen liblz4-tool openjdk-8-jre gcc g++ texinfo dosfstools mtools default-jre default-jdk libncurses5 apt-utils wget scons python3.8-distutils tar rsync git-core libxml2-dev lib32z-dev grsync xxd libglib2.0-dev libpixman-1-dev kmod jfsutils reiserfsprogs xfsprogs squashfs-tools pcmciautils quota ppp libtinfo-dev libtinfo5 libncurses5-dev libncursesw5 libstdc++6 gcc-arm-none-eabi vim ssh locales libxinerama-dev libxcursor-dev libxrandr-dev libxi-dev

**四、实验设计思路**

**4.1Windows或Linux上开发环境的配置**

1. sudo apt-get install 以上依赖包，如果ubuntu的apt源有问题，或者有无法下载的依赖关系，请更换源。

实测清华大学的镜像源没有问题，方法如下：

1. #备份源配置文件
2. cd /etc/apt/
3. cp sources.list sources.list.old
4. #更换源
5. vim sources.list
6. deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted
7. deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted
8. deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal universe
9. deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates universe
10. deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal multiverse
11. deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates multiverse
12. deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
13. deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted
14. deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security universe
15. deb http://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security multiverse
16. #刷新软件源信息
17. sudo apt update
18. sudo apt upgrade
19. 安装python

我的是3.8.10，可以在终端输入which python3.8查看python位置

建议将Python和Python3切换为Python 3.8，防止之后pip和pip3出错

1. sudo update-alternatives --install /usr/bin/python python {Python 3.8 路径} 1
2. #{Python 3.8 路径}为上一步查看的Python 3.8的位置
3. sudo update-alternatives --install /usr/bin/python3 python3 {Python 3.8 路径} 1
4. #{Python 3.8 路径}为上一步查看的Python 3.8的位置
5. 获取源码

注意，请不要下载最新的master开发分支，会有问题，建议下载release>=3.2发行版,由于3.1版qemu配置有小问题，本人亲测避雷。

配置repo：

1. #更新软件源
2. sudo apt-get update
3. #通过以下命令安装：
4. sudo apt-get install git git-lfs
5. #第一次请配置信息
6. git config --global user.name "yourname"
7. git config --global user.email "your-email-address"
8. git config --global credential.helper store
9. #执行如下命令安装码云repo工具，安装路径以"~/bin"为例
10. mkdir ~/bin
11. curl https://gitee.com/oschina/repo/raw/fork\_flow/repo-py3 -o ~/bin/repo
12. chmod a+x ~/bin/repo
13. pip3 install -i https://repo.huaweicloud.com/repository/pypi/simple requests
14. #将repo添加到环境变量。
15. vim ~/.bashrc
16. # 在环境变量的最后添加一行repo路径信息
17. export PATH=~/bin:$PATH
18. source ~/.bashrc

获取源码：

1. #初始化仓库
2. repo init -u git@gitee.com:openharmony/manifest.git -b  OpenHarmony-3.2-Release  --no-repo-verify
3. #同步仓库代码
4. repo sync -c
5. #下载大文件
6. repo forall -c 'git lfs pull'

在源码根目录下执行prebuilts脚本，安装编译器及二进制工具。

bash build/prebuilts\_download.sh

**4.2 内核的编译、下载和测试**

（1）安装hb工具

1. #安装hb并更新至最新版本
2. pip3 install --user build/lite
3. #设置环境变量
4. vim ~/.bashrc
5. export PATH=~/.local/bin:$PATH
6. source ~/.bashrc
7. #在源码目录执行"hb -h"，界面打印版本信息即表示安装成功
8. hb set选择开发板

选择 qemu\_riscv开发板，若出错，请参考实验错误排查和解决方法。

1. hb build -f全量编译

显示build success，生成qemu可以运行的映像文件，若出错，请参考实验错误排查和解决方法。

（4）在根目录运行./qemu-run脚本

**4.3 程序结构**

**4.3.1 程序的基本结构**

建议在./applications/sample/新建目录，用于存放业务源码文件。

.

└── applications

└── sample

└── my\_first\_app

│── hello\_world.c

└── BUILD.gn

hello\_world.c负责程序逻辑

BUILD.gn负责构建静态库

1. 方法一：构建静态库链接

在demo文件夹中为con.json里的force\_link\_libs[]添加新组件名，比如myapp

在BUILD.gn中添加组件路径

（2）方法二：修改文件build/lite/components/communication.json，添加组件

参考quick-start开发文档

https://gitee.com/openharmony/docs/tree/master/zh-cn/device-dev/quick-start

**4.3.2 关键函数或参数或某种机制**

1. SYS\_RUN( func)

是一个宏定义，用于在系统启动过程的系统启动阶段，。这个宏帮助开发者按特定的顺序来初始化和启动系统的各个部分。

func：这个参数表示要在系统运行阶段初始化和启动的函数的入口。这个函数的类型必须是void (\*)(void)，也就是说，这个函数不能有任何参数，也没有返回值。

**五、关键代码分析**

**5.1 程序一：hello\_world.c**

位于./applications/sample/hello目录下

#include <stdio.h>

#include "ohos\_init.h"

#include "ohos\_types.h"

void HelloWorld(void)

{

printf("[DEMO] Hello world.\n");

}

SYS\_RUN(HelloWorld);

使用OpenHarmony启动恢复模块接口SYS\_RUN()启动业务

（SYS\_RUN定义在ohos\_init.h文件中）

**5.2 程序二：BUILD.gn**

位于./applications/sample/hello目录下

static\_library("myapp") {

sources = [

"hello\_world.c"

]

include\_dirs = [

"//utils/native/lite/include"

]

}

1.static\_library中指定业务模块的编译结果，为静态库文件libmyapp.a

2.sources中指定静态库.a所依赖的.c文件及其路径

(若路径中包含"//"则表示绝对路径若不包含"//"则表示相对路径)

3.include\_dirs中指定source所需要依赖的.h文件路径

**5.3 程序三：config.json**

在./vender/ohemu/qemu\_riscv32\_mini\_system\_demo目录下

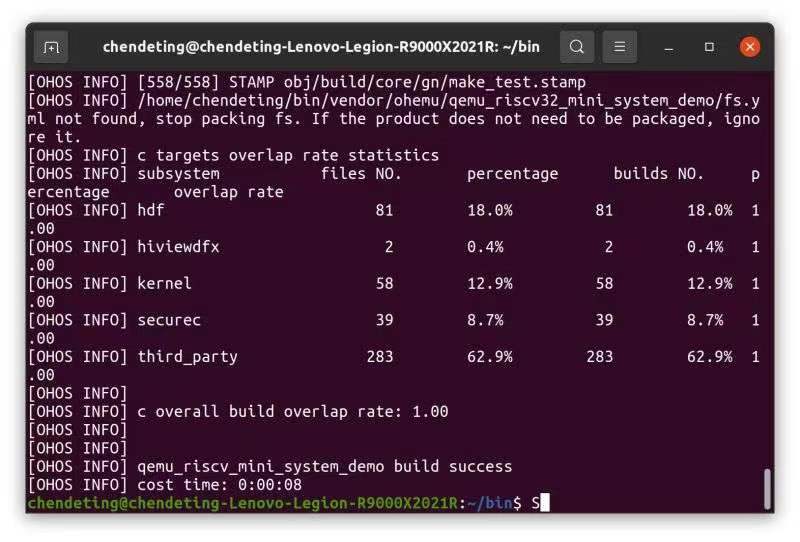
force\_link\_libs[myapp]

**5.4 程序四：BUILD.gn**

在./vender/ohemu/qemu\_riscv32\_mini\_system\_demo目录下

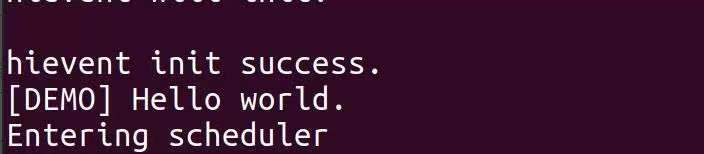
deps=[“hello：myapp”]

1. **程序运行结果和分析**
2. **hb build**



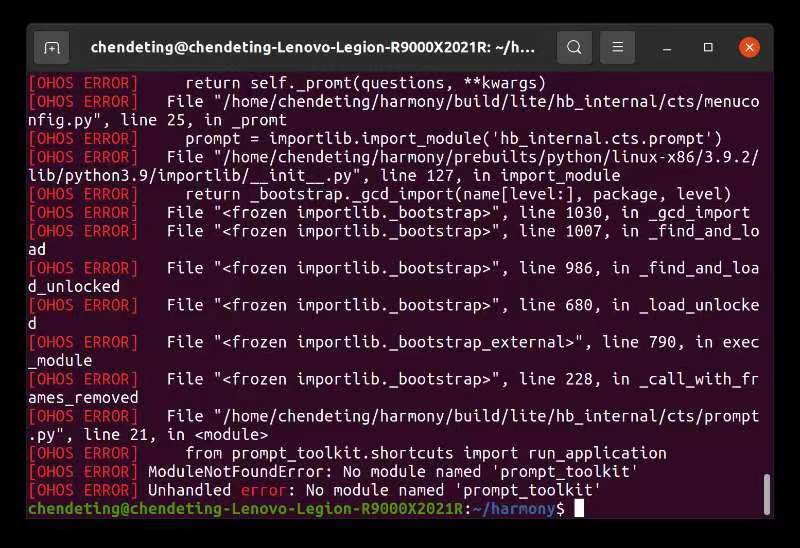
请确保以上环境构建已经完成，若出错请参考实验错误排查和解决方法

1. **./qemu-run**



**七、实验错误排查和解决方法**

**7.1** Q:hb set报错

****

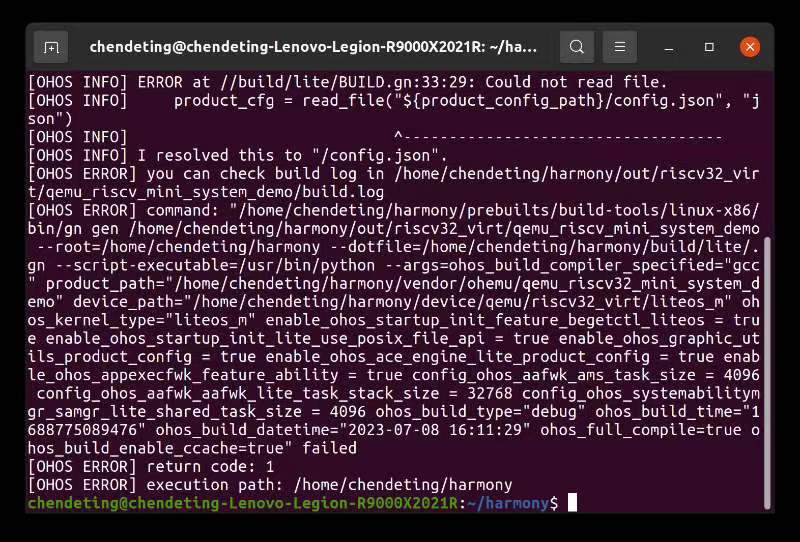
A:首先请确保安装了合适的prompt\_toolkit版本

pip3 list查看prompt\_toolkit安装版本

若仍然报错，请按照以下方法：

1. python3 -m pip uninstall ohos-build
2. python3 -m pip uninstall ohos-build==0.4.3
3. hb set 发现成功，但这个版本的hb build会有问题
4. python3 -m pip uninstall ohos-build
5. python3 -m pip install --user build/lite（在根目录下运行）

7.2：Q:hb build出错



A：参考问题1，hb0.4.3不支持这个版本的hb build

1. **实验参考资料和网址**

（1）教学课件

（2）网址链接1：https://gitee.com/openharmony/docs/tree/master/zh-cn/device-dev/quick-start

（3）网址链接2：

https://gitee.com/openharmony/device\_qemu