北京电子科技学院（BESTI）

**实 验 报 告**

课程：信息安全系统设计 班级：2214

姓名：徐鹿鸣 学号：20221414

成绩： 指导教师：娄嘉鹏 实验日期：2024/11/24

实验密级：无 预习程度：全预习

实验时间：3，4节 仪器组次：无 必修/选修：必修

实验序号：实验三

实验名称： 密码模块实现

实验目的与要求：基于商用密码标准的密码模块的实现。实现简单的密码引擎，能够提供对称密码算法、非对称密码算法、Hash算法等的密码服务。

实验仪器：

|  |  |  |
| --- | --- | --- |
| 名称 | 型号 | 数量 |
| 计算机 |  | 1 |
|  |  |  |

实验内容、步骤与体会（附纸）：

**1. 实验内容**

实现简单的密码引擎，能够提供对称密码算法、非对称密码算法、Hash算法等的密码服务。

**2. 实验步骤**

**2.1步骤1 裸卡 0018 代码测试**

**2.1.1 操作内容**

首先运行git pull命令，获得最新的代码。

在bestidiocs2024/ch06/rochs0018sdf中，使用tree查看文件结构，发现readMe.txt文件。运行cat readMe.txt命令得到之后的操作步骤。

我的裸卡是HSCEU芯片，所以要用老师在群中新发的代码。将老师在群中发的代码复制到桌面解压，再将其中唯一的文件添加zip后缀解压，得到HS\_besti\_linux\_SDK\_20200924文件夹。将其拷贝到Ubuntu系统下继续操作。

先运行config中的config.sh，再根据自己的平台情况，将x64文件夹中中.so库文件拷贝到example文件夹下。修改makefile文件。执行make命令。

此时直接运行可执行文件会有“open devces fail rv = 0x0100000”报错。这是因为没有连接到裸卡芯片。

参考[微软官方的教程](https://learn.microsoft.com/zh-cn/windows/wsl/connect-usb)或[我的过程文档](https://github.com/youer0219/Information-Security-System-Design-Assignment/blob/main/%E5%8A%A0%E5%88%86%E9%A1%B9/WSL%E8%AE%BF%E9%97%AEUSB%E5%8F%82%E8%80%83/WSL%E8%BF%9E%E6%8E%A5USB%E5%8F%82%E8%80%83.md)使用 [usbipd-win](https://github.com/dorssel/usbipd-win) 将 USB 设备连接到在 WSL 2 上运行。可以使用lsusb命令查看是否有设备信息，如有则是连接成功，或者观察Windows是否有U盘信息显示，如果没有，间接说明连接成功。

最后运行可执行文件，输出结果见操作结果中的图片。

如果出现了下面两行输出结果，但最终连接失败，说明USB与WSL2的连接是没有问题的，最大的可能是使用了错误的代码，请检查芯片型号：

open device success!

open session success!

**2.1.2操作结果**

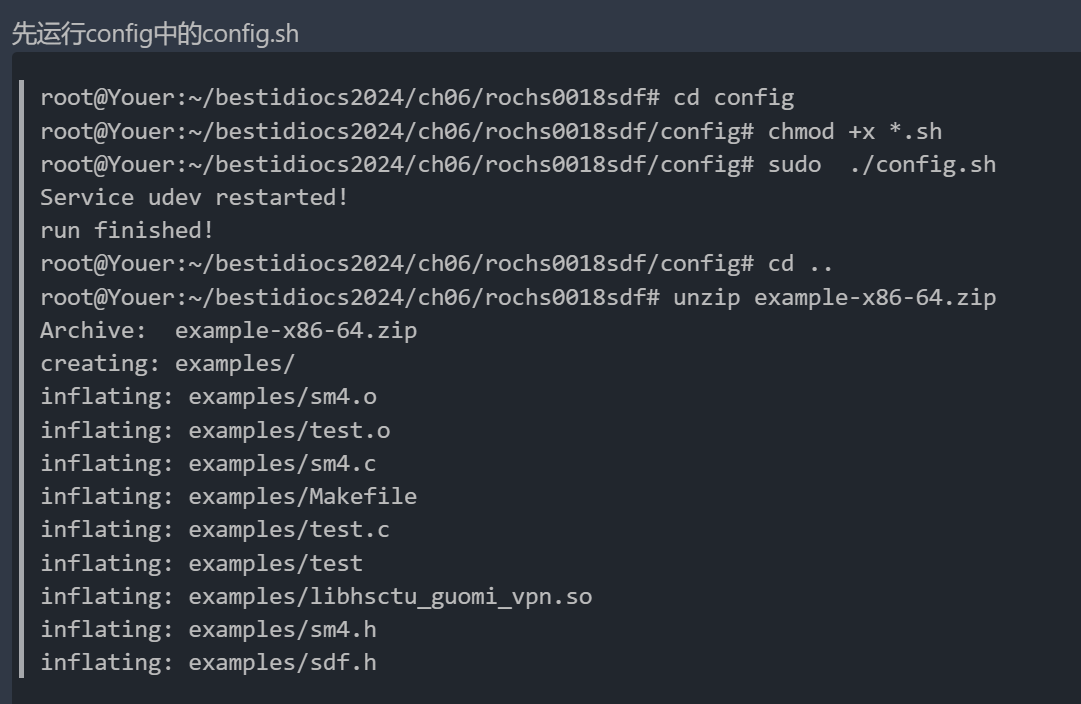


图 1 先运行config中的config.sh

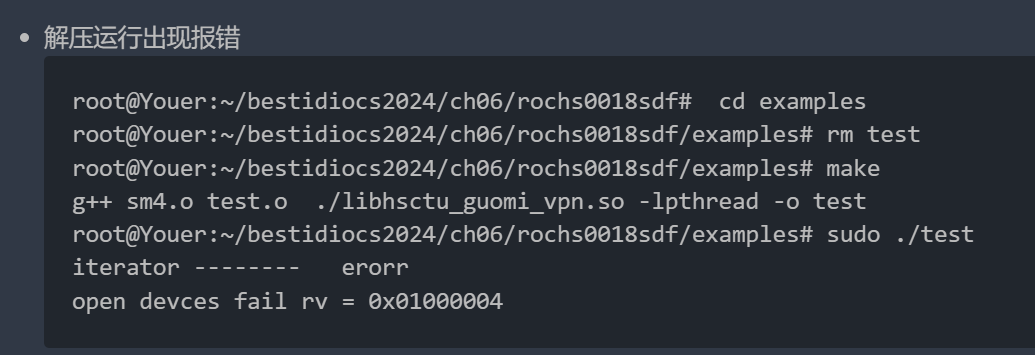


图 2 USB未连接到WSL时的报错

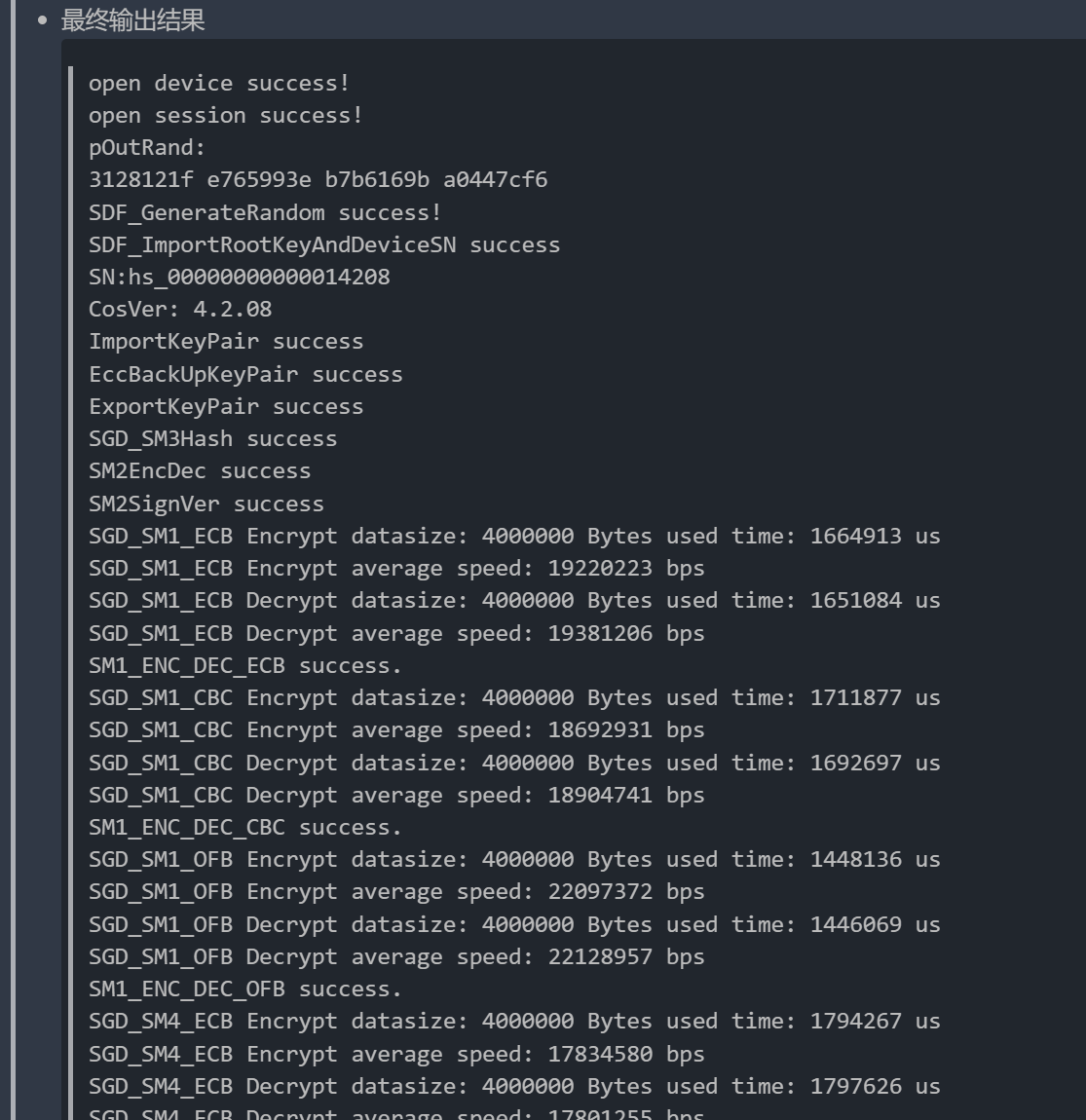


图 3 验证成功的输出结果

**2.2步骤2 龙脉 0016 代码测试**

**2.2.1 操作内容**

这个步骤并未成功。下面将讲述我的过程与解决的尝试，供参考。

插入龙脉芯片，参考之前的步骤将USB连接到WSL中。解压老师仓库中的longmaiskf0016-stu.zip压缩包。在samples/skf/linux\_mac文件夹中新建lib/linux文件夹，在linux文件夹下把longmaiskf0016-stu\skf\linux\x64文件夹拷贝过来。之后每次make一个示例代码，只需要将makefile中的x86改为x64即可。同时参考其readme.txt内容把90-mtoken.rules文件放置到 /etc/udev/rules.d 目录下。一些文件会显示中文乱码，把编码环境改为GB 2312即可。（vscode有通过编码重新打开功能）

我之前是先编译运行了encrypt文件夹内容，发现并无输出，添加了很多调试信息后发现设备枚举成功，但设备信息获取失败——返回的设备名称为空。事后来看，先编译运行enumdevinfo文件中内容会更好，因为其本身就是连接的测试函数，自带合适的报错信息。

参考[在WSL中进行USB设备测试](https://github.com/dorssel/usbipd-win/wiki/Tested-Devices)的内容，我对该USB设备进行了测试。其在Windows下运行正常，在WSL中也可也通过lsusb命令看见设备信息；通过dmesg | tail命令可以看见有一条最近的消息，指示已发现该设备；通过udevadm info -a -n /dev/bus/usb/001/002命令查看USB信息，无明显错误；判断这是复合型USB设备，通过lsusb –tree命令查看其是否显示了所有预期的 USB 接口。在过程中使用AI辅助理解其中的输出，但都没有找到明显的问题。

我尝试向官方反馈了这个问题，但目前无回复。

综上，我认为这不是我单人可以解决的问题。

**2.2.2操作结果**

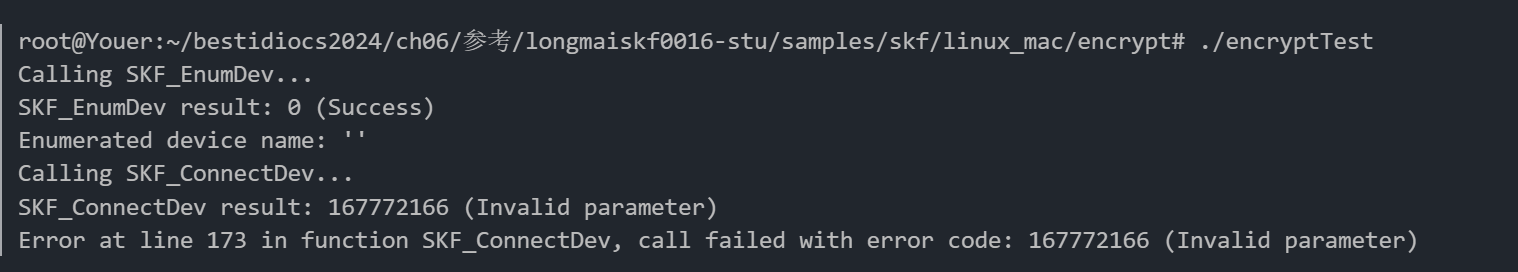


图 4 明确问题出在返回的设备名称为空上

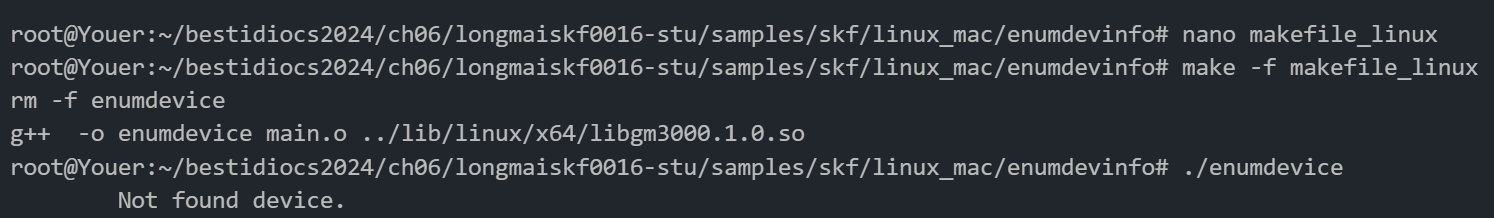


图 5 寻找设备失败



图 6 向官方反馈记录

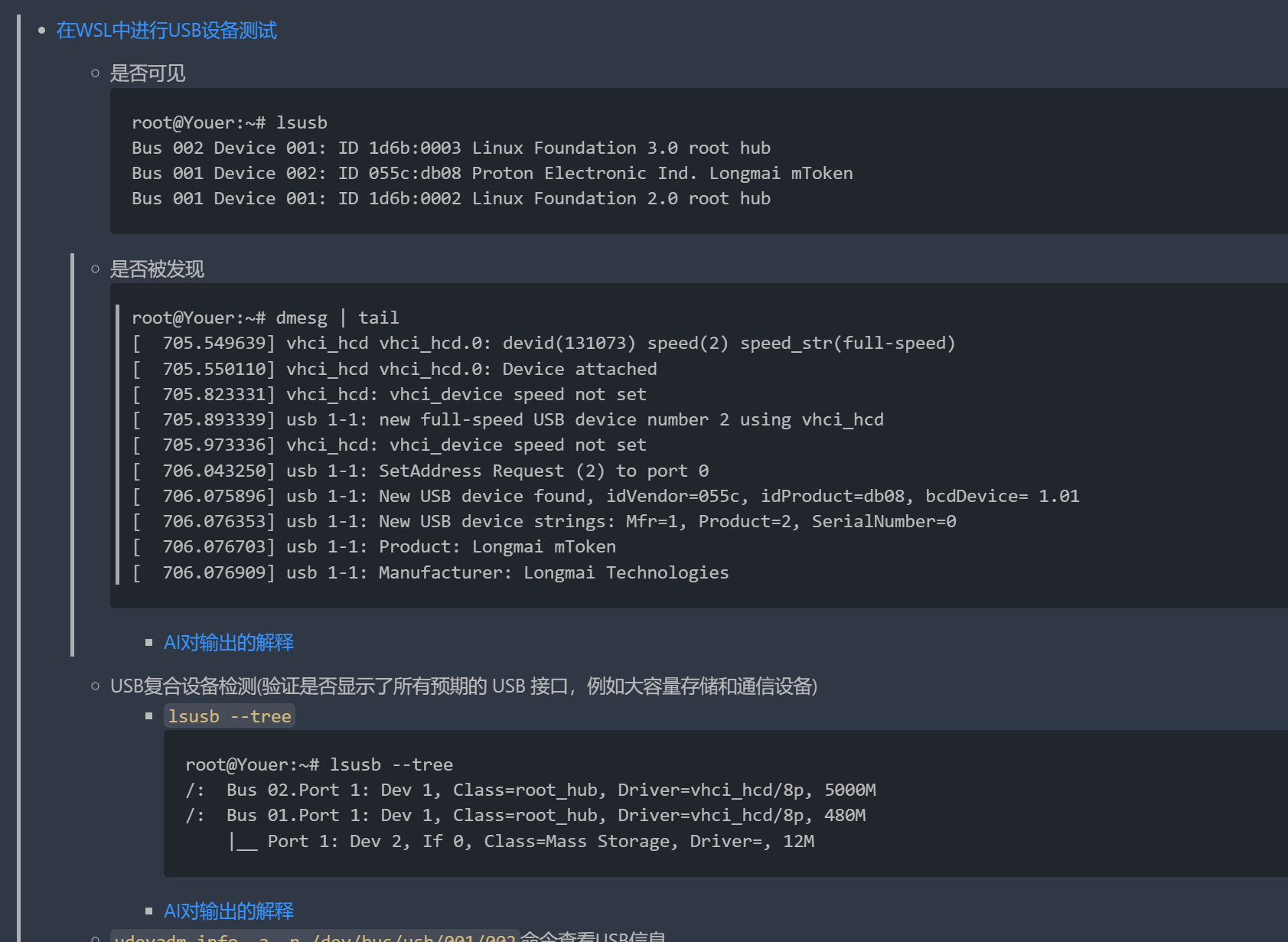


图 7 部分在WSL中进行USB设备测试的结果

**2.3步骤3 完成gmt0018中，对称加密解密，非对称加密解密，签名验签，hash运算，MAC运算接口中的至少一类**

**2.1.1 操作内容**

在查询密钥管理含义时偶然发现了[GmSSL/SoftSDF库](https://github.com/GmSSL/SoftSDF" \o "https://github.com/GmSSL/SoftSDF)。这是GmSSL官方根据标准0018-2012实现SDF接口的软件项目，与我们的实验要求一致。

实验要求本质就是在参考的sdf.h中找到或自己根据标准实现相关的函数声明，在sdf.c中实现该函数，在test.c中调用该函数来测试。

在该项目中sdf.h和sgd.h文件定义了各类常量结构和函数声明，softsdf.c中实现了所有声明的函数，在softsdftest.c中测试了Hash相关的功能函数是否与GmSSL库的效果一致。

参考[老师发的链接](https://www.cnblogs.com/zyj2000/p/14702490.html)中展示的项目结构组织自己的文件，使用AI辅助生成一个makefile文件。编译运行后输出结果“test ok”，说明函数测试成功。

**2.1.2操作结果**

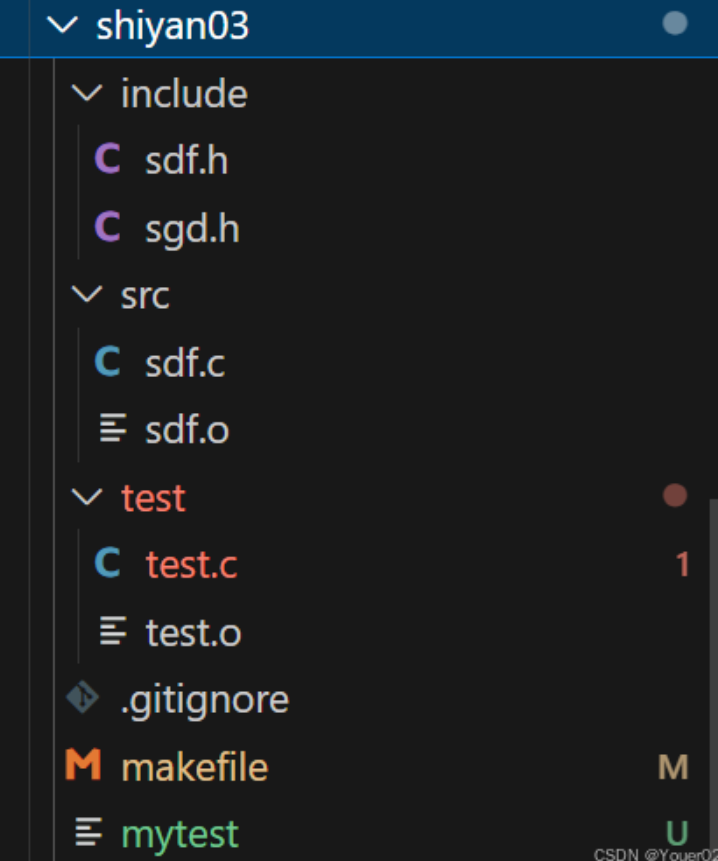


图 8 项目结构

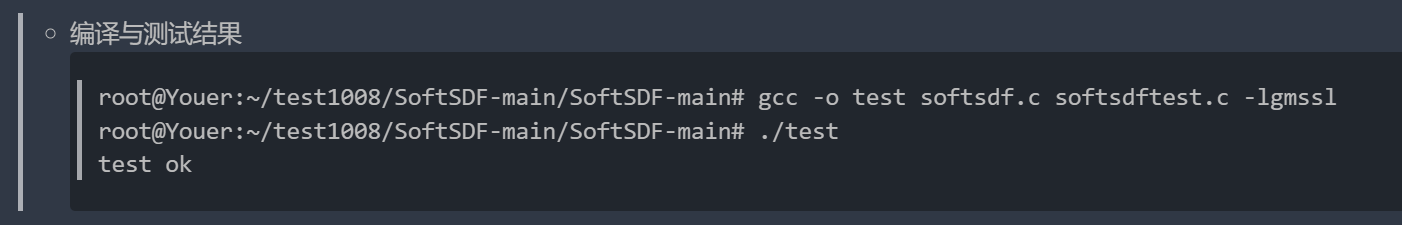


图 9 编译与测试结果

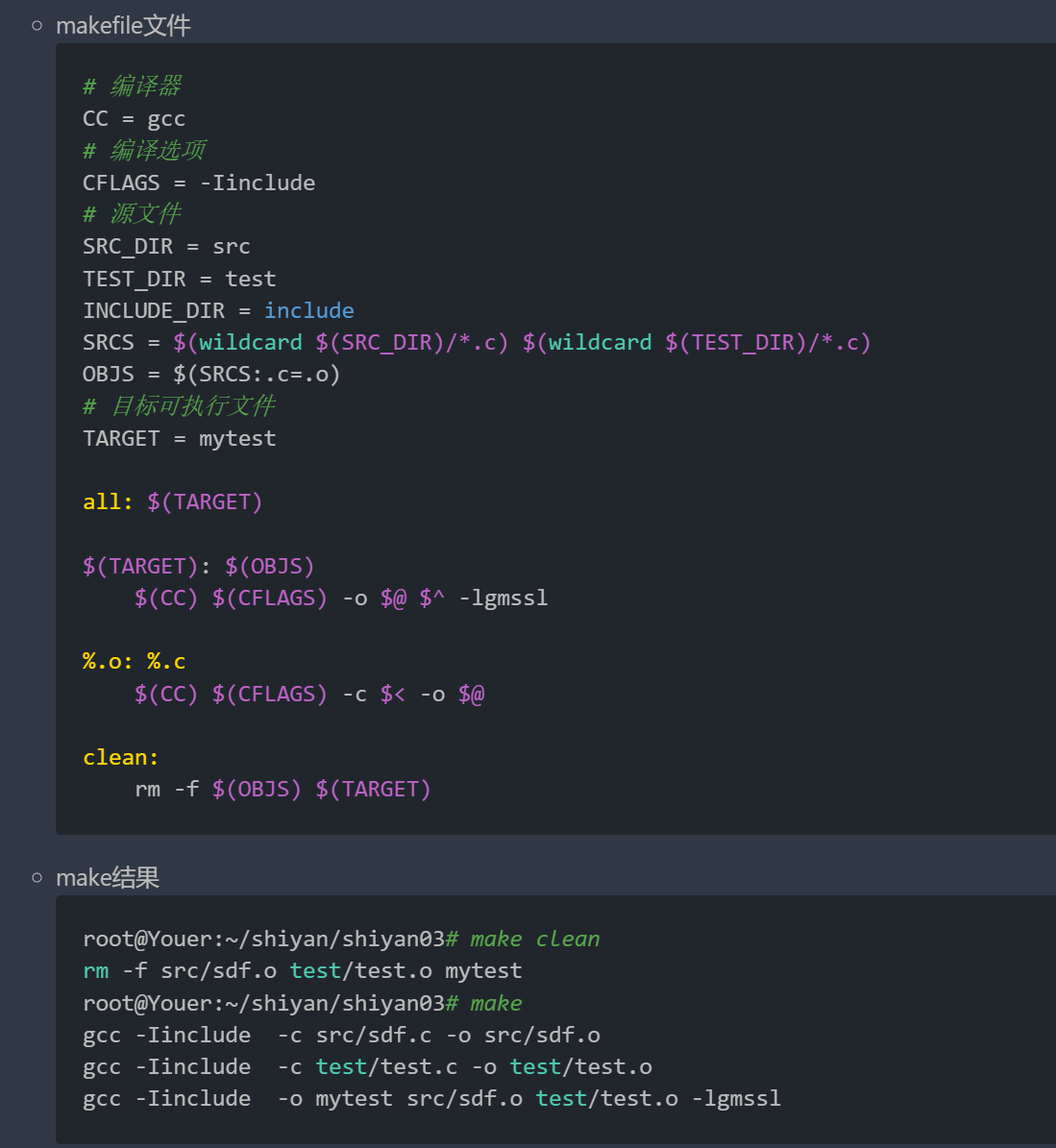


图 10 makefile文件内容与make结果

**3．实验体会**

**3.1** 调试中出现的问题及解决过程

**3.1.1** 步骤一中解决WSL无法连接USB问题

裸卡代码运行出现以下报错：

iterator -------- erorr

open devces fail rv = 0x01000004

这个问题在网上的解决方法的文章非常的多，只要搜索的关键词不要太偏应该都能很快找到。当时通过知乎文章的引用找到了微软官方的[解决方法文档](https://learn.microsoft.com/zh-cn/windows/wsl/connect-usb)，这无疑是最权威的解决方法。下面根据文档内容与自己实践来讲述解决方法，也可以参考[我的解决文档](https://github.com/youer0219/Information-Security-System-Design-Assignment/blob/main/%E5%8A%A0%E5%88%86%E9%A1%B9/WSL%E8%AE%BF%E9%97%AEUSB%E5%8F%82%E8%80%83/WSL%E8%BF%9E%E6%8E%A5USB%E5%8F%82%E8%80%83.md)来进行。

先决条件是：一、系统是Windows 11 x64 ； 二、 WSL版本为WSL2

第一步：打开`终端管理员`，运行“winget install --interactive --exact dorssel.usbipd-win”命令安装 USBIPD 。

第二步：附加USB设备

首先通过以管理员模式打开 PowerShell 并输入usbipd list命令，列出所有连接到 Windows 的 USB 设备

然后使用命令 usbipd bind 来共享你需要连接的设备，从而允许它附加到 WSL。这需要管理员权限。选择要在 WSL 中使用的设备总线 ID，然后运行“sbipd bind --busid 2-1”命令。运行命令后，可以再次使用命令 usbipd list 验证设备是否已共享。

继续执行下面的命令“usbipd attach --wsl --busid 2-1”。这里的“2-1”要换成自己的设备总线ID。

然后在Linux系统中运行lsusb命令。

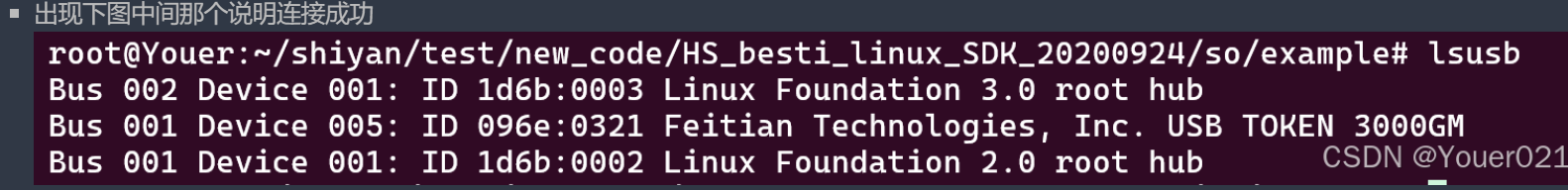


图 11 成功截图

根据你的应用程序，你可能需要配置 udev 规则以允许非根用户访问设备。(我直接是root用户，没有经过这一步骤，且最新版可能不需要这个操作。)

同时，请最好不要关闭WSL命令提示符。以及，只要 USB 设备连接到 WSL，Windows 将无法使用它。所以如果你在Windows下找不到这个U盘的提示，也可以间接说明你连接成功

以上便是WSL无法连接USB的常规解决方法。在这个过程中，你可能会遇到以下问题：

一：出现“usbipd: warning: A firewall appears to be blocking the connection; ensure TCP port 3240 is allowed.”提示

这说明你的防火墙阻止了WSL与USB的连接。因为这个连接本质上就是通过USB/IP协议在网络层进行的操作。只需要在防火墙入站规则中新建一个允许TCP连接端口3240的规则即可。

二：“WSL usbip: error: Attach Request for 2-1 failed - Device busy (exported)”报错

这说明你的USB设备正在Windows中使用。这可能发生在你刚插入设备就进行连接时，也可能真的在使用该设备。你可以再次尝试连接，或者找到使用该USB的进程并关闭，亦或者强制连接，添加一个“--force”参数。

**3.1.2** 步骤二中龙脉USB无法连接问题（未解决）

这个问题在之前已有比较充分的论述。这里只是做一个记录。

**3.2 心得体会**

这次实验都基本没有提前完成，主要是没有理解题目要求。事实证明，确实不需要提前开始。这也算是幸运了。

在这个过程中，我对AI的使用更加熟悉，对解决问题的方法有了更深的感悟——在这次实验中，我们使用了权威的官方的文档来帮助解决WSL连接USB问题，使用了官方的GmSSL/SoftSDF库来完成步骤三。这告诉找到官方的权威的帮助文档是极其重要与关键的，这比起用AI炼丹会舒服很多，且结果更加稳定。

那么，AI在这个过程中有什么帮助吗？AI首先可以帮助我们理解问题的本质，提供多种的猜测（尽管大部分都是套话），其次可以帮助查询合适的参考文章，这次的SoftSDF库就是在AI提供的参考链接中找到的，最后也可以在我们不懂的领域帮助我们有一个基本的认知，比如在进行WSL上的USB测试时帮助理解各种命令的输出的含义。

所以，比起实验二中的“赛博炼丹”，我想我有了更多的进步。

**3.3 实验改进建议**

建议明确实验要求，比如实验3-1，提前说好要用裸卡与龙脉设备进行实验，并给出一定的实验参考，便于提前做准备。以及提供更准确的代码，比如裸卡测试中不同芯片的代码。同时希望用正常的方法进行压缩，不需要自己去补一个后缀在解压。