轻量级鸿蒙结合RISC-V的快速应用方法

苏州大学计算机科学与技术学院 王宜怀
南京沁恒微电子股份有限公司 杨 勇
2024年8日23日· 杭州

内容简介

实时操作系统（RTOS）是嵌入式人工智能与物联网终端的重要工具和运行载体，如何从应用与原理两个层面把RTOS的脉络梳理清楚是实时操作系统课程的核心。本次交流阐述国产轻量级鸿蒙（LiteOS）结合RISC-V架构微控制器的快速应用方法，试图把复杂问题简单化；从应用视角，阐述轻量级鸿蒙LiteOS的线程、调度、延时函数、事件、消息队列、信号量、互斥量等基本知识要素。分析RTOS的应用与原理剖析既有区别又有联系的两个层面的基本着眼点。

目录

一、学习实时操作系统的目标

1.1 几个层面：应用开发、了解原理、理解原理、撰写RTOS

学习RTOS的几个可能出发点：

（1）应用开发：学会在RTOS场景下进行基本应用程序开发；（绝大多数目标）

（2）了解原理：在掌握应用编程的前提下，了解其运行原理，进行深度应用程序开发。（服务于应用开发）

（3）理解原理：知其然，知其所以然；（人数很少：移植与驻留）

（4）撰写一个RTOS。（极少数人做的事情）

进一步说，应用RTOS与理解RTOS可以简单地表述为：

应用就是在基本了解线程、调度等基本概念基础上，正确运用RTOS下的延时函数、事件、消息队列、信号量、互斥量等基本要素，进行应用程序的开发，让RTOS成为嵌入式软件开发的辅助工具。

理解就是要理解RTOS工作的基本原理，也就就是理解调度机制，理解延时函数、事件、消息队列、信号量、互斥量等基本要素工作机制，为应用编程提供更加坚实的基础。

以消息队列为例，应用编程就是这样一个场景：有个“装消息”的篮子，有个发送消息处，当发送消息处发出消息，消息就会进入“装消息”的篮子，而只要消息“篮子”中有消息，等待消息的程序就会运行。理解运行机制就是为什么发送消息处发出消息，消息就会自动进入消息“篮子”，为什么消息“篮子”中有消息，等待消息的程序就会运行。

1.2 实时操作系统的选型问题

RTOS种类繁多，有国外的，也有国产的；有收费的，也有免费的；有带有持续维护升级的，也有依赖爱好者更新升级的。初学者有时不知所措。但是无论如何，学习RTOS，必须以一个具体的RTOS为蓝本。不同RTOS，其应用方法及原理大同小异，掌握其共性是学习的关键，这样才能达到举一反三的效果。

这里推荐的是国产轻量级鸿蒙LiteOS实时操作系统，它是华为公司于2015年开始推出的一款开源实时操作系统，具有高效、灵活、安全等特点，面向各种嵌入式设备。自推出以来，经过多次版本迭代和优化，已成为面向嵌入式人工智能与物联网领域的重要操作系统之一。被广泛应用于工业控制、智能家居、智能穿戴等众多行业领域。将其内核用到实际产品中，没有收费陷阱问题。本次交流以轻量级鸿蒙LiteOS为蓝本，以RISC-V架构的CH32V303微控制器构建的通用嵌入式计算机（GEC）为硬件载体，阐述RTOS中的线程、调度、延时函数、事件、消息队列、信号量、互斥量等基本知识要素。

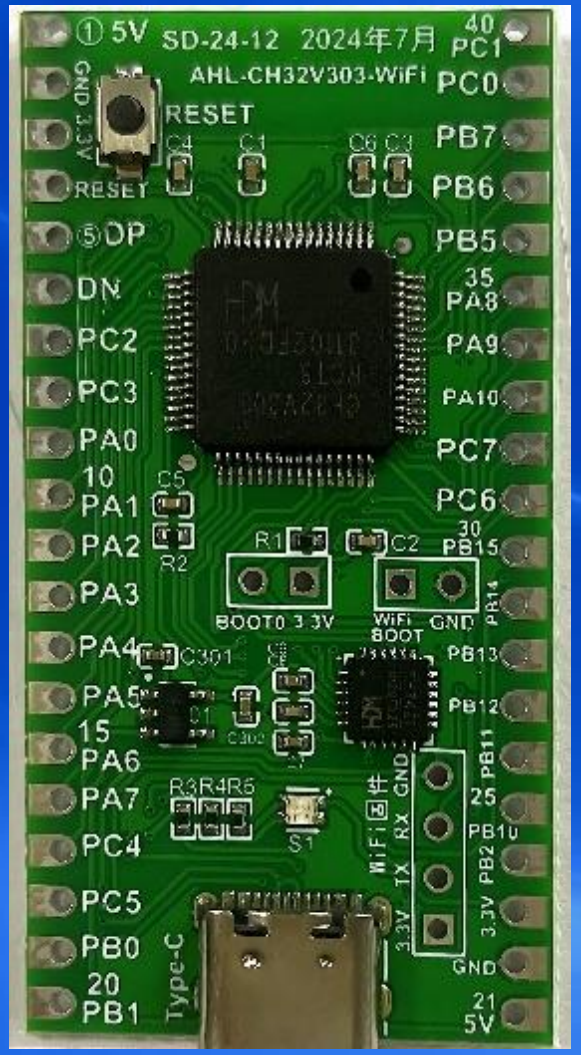
1.3 实践问题

RTOS是理论联系实践的典型体系，学习RTOS，不仅要以一个具体的RTOS为蓝本，还要一个具体的MCU为蓝本进行实践，否则就可能成为纸上谈兵。

对实践器材的要求是：满足RTOS的基本知识要素实践训练，价格低廉，使用方便，稳定可靠。

开发环境：苏州大学嵌入式人工智能与物联网实验经过十多年努力研发了AHL-GEC-IDE （http://sumcu.suda.edu.cn/AHLwGECwIDE/main.htm），可以方便地用于RTOS实践。

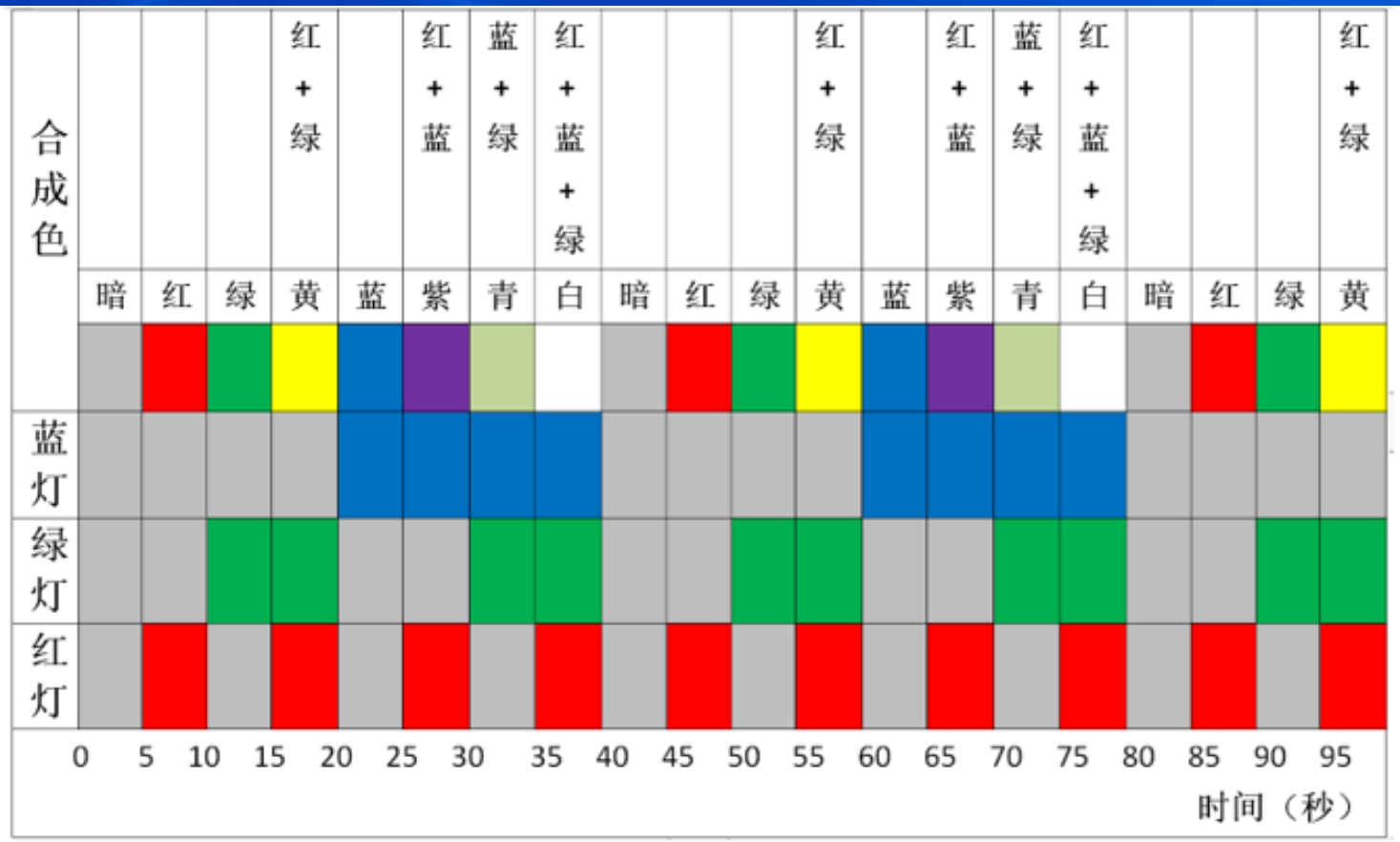
硬件系统：AHL-CH32V303-WiFi开发套件
内含CH32V303VCT6芯片及其硬件最小系统、三色灯、复位按钮、内部温度传感器、WiFi通信、两路TTL-USB串口。



1.4 实践样例

“照葫芦画瓢”教学实践基本理念：照葫芦画瓢是比喻照着样子模仿，表示简单易行之含义。古希腊哲学家亚里士多德说过：“人从儿童时期起就有模仿本能，他们用模仿而获得了最初的知识，模仿就是学习”。孟子则曰：“大匠诲人必以规矩，学者亦必以规矩”。借此，期望通过建立符合软件工程基本原理的“葫芦”，为“依葫芦画瓢”提供坚实基础，达到降低学习实践难度之目标。

第一个样例功能



第一个样例程序功能

建立标准工程框架：做到“分门别类，各有归处”

01_Doc	文档文件夹：文档作为工程密切相关部分，是软件工程的基本要求
02_CPU	CPU 文件夹：存放 CPU 相关文件
03_MCU	MCU 文件夹：含有 linker_file、startup、MCU_drivers 下级文件夹
04_GEC	GEC 文件夹：引入通用嵌入式计算机（GEC）概念，预留该文件夹
05_UserBoard	用户板文件夹：含有硬件接线信息的 User.h 文件及应用驱动
06_AlgorithmComponent	软件构件文件夹：含有与硬件无关的软件构件
07_AppPrg	应用程序文件夹：应用程序主要在此处编程

工程框架

第一个样例工程中的线程

归属	线程名	执行函数	优先级	线程功能	中文含义
内核	main_thread	app_init	10	创建其他线程	主线程
	idle	idle	31	空闲线程	空闲线程
用户	thd_redlight	thread_redlight	10	红灯以 5s 为周期闪烁	红灯线程
	thd_greenlight	thread_greenlight	10	绿灯以 10s 为周期闪烁	绿灯线程
	thd_bluelight	hread_bluelight	10	蓝灯以 20s 为周期闪烁	蓝灯线程

1.5 为鸿蒙轻量级LiteOS教学所做的准备工作总结

（1）实践硬件：AHL-CH32V303-WiFi
（2）开发环境：AHL-GEC-IDE
（3）样例程序编制：针对各基本要素，按照统一模板编制了样例程序
（4）撰写书籍：实时操作系统应用技术—基于轻量级鸿蒙LiteOS与RISC-V的编程实践，机械工业出版社，（待出版）
（5）课件：需要时可以获取，正在进行其他教学资源建设
（6）科研拓展：AHL-EORS应用、NB-IoT应用开发

三、RTOS中核心概念导引

2.1 线程

线程是RTOS中最重要的概念之一。在RTOS下，把一个复杂的嵌入式应用工程按一定规则分解成一个个功能清晰的小工程，然后设定各个小工程的运行规则，交给RTOS管理，这就是基于RTOS编程的基本思想。这一个个小工程被称为线程（Thread），RTOS管理这些线程，被称为调度（Scheduling）。

补充说明：多进程与多线程

RTOS一般是单进程的，所以不再出现进程概念，这是RTOS由RTOS应用场合决定的，用于连续运行的实时性要求较高的工业场景，而台式计算机、笔记本电脑、平板电脑、手机等是多进程的，如看似同时打开WORD、播放音乐、发邮件等，每个正在运行的程序均是一个进程，每个进程一般会有若干线程

要给RTOS中的线程下一个准确而完整的定义并不十分容易，可以从不同视角理解线程。从线程调度视角理解，可以认为，RTOS中的线程是一个功能清晰的小程序，是RTOS调度的基本单元；从RTOS的软件设计视角来理解，就是在软件设计时，需要根据具体应用，划分出独立的、相互作用的程序集合，这样的程序集合就被称为线程，每个线程都被赋予一定的优先级；从CPU视角理解，在单CPU下，某一时刻CPU只会处理（执行）一个线程，或者说只有一个线程占用CPU。RTOS内核的关键功能就是以合理的方式为系统中的每个线程分配时间（即调度），使之得以运行。

实际上，根据特定的RTOS，线程可能被称为任务（Task），也可能使用其他名词，含义或许稍有差异，但本质不变，也不必花过多精力追究其精确语义，因为学习RTOS的关键在于掌握线程设计方法、理解调度过程、提高编程鲁棒性、理解底层驱动原理、提高程序规范性、可移植性与可复用性、提高嵌入式系统的实际开发能力等。

2.2 调度

调度（Scheduling）就是决定该轮到哪个线程运行了，它是内核最重要的职责。每个线程根据其重要程度不同，被赋予一定的优先级。不同的调度算法对RTOS的性能有较大影响，基于优先级的调度算法是RTOS常用的调度算法，其核心思想是，总是让处于就绪态的、优先级最高的线程先运行。然而何时高优先级线程掌握CPU的使用权，由使用的内核类型确定，基于优先级的内核有不可抢占型和可抢占型两种类型。

2.3 线程的三要素：线程函数、线程堆栈、线程描述符

线程函数。一个线程，对应一段函数代码，完成一定功能。从代码上看，线程函数与一般函数并无区别，但是从线程自身视角来看，它认为CPU就是属于它自己的，并不知道还有其他线程存在。线程函数也不是用来被其他函数直接调用的，而是由RTOS内核调度运行。要使线程函数能够被RTOS内核调度运行，必须将线程函数进行“登记”，要给线程设定优先级、设置线程堆栈大小、给线程编号等，不然有几个线程都要运行起来，RTOS内核如何知道哪个该先运行呢？由于任何时刻只能有一个线程在运行（处于激活态），当RTOS内核使一个线程运行时，之前的运行线程就会退出激活态。CPU被处于激活态的线程所独占，从这个角度看，线程函数与无操作系统（NOS）中的“main”函数性质相近，一般被设计为“永久循环”，认为线程一直在执行，永远独占处理器。

线程堆栈。它是独立于线程函数之外的RAM，按照“先进后出”策略组织的一段连续存储空间，是RTOS中线程概念的重要组成部分。在 RTOS中被创建的每个线程都有自己私有的堆栈空间，在线程的运行过程中，堆栈用于保存线程程序运行过程中的局部变量、线程调用普通函数时会为线程保存返回地址等参数变量、保存线程的上下文等。

线程描述符。线程被创建时，系统会为每个线程创建一个唯一的线程描述符（Task Descriptor，TD），它相当于线程在RTOS中的一个“身份证”，RTOS就是通过这些“身份证”来管理线程和查询线程信息的。

2.4 线程的四种状态：终止态、阻塞态、就绪态和激活态

RTOS中的线程一般有四种状态，分别为终止态、阻塞态、就绪态和激活态。在任一时刻，线程被创建后所处的状态一定是四种状态之一。

终止态：线程已经完成或被删除，不再需要使用CPU。

阻塞态：又可称为“挂起态”。线程未准备好，不能被激活，因为该线程需要等待一段时间或某些情况发生；当等待时间到或等待的情况发生时，该线程才变为就绪态，处于阻塞态的线程描述符存放于等待列表或延时列表中。

就绪态：线程已经准备好可以被激活，但未进入激活态，因为其优先级等于或低于当前的激活线程，一旦获取CPU的使用权就可以进入激活态，处于就绪态的线程描述符存放于就绪列表中。

激活态：又称“运行态”，该线程在运行中，线程拥有CPU使用权。

2.5 RTOS下的延时函数

ROTS中，线程一般不采用原地空跑（空循环）的方式进行延时（该方式线程仍然占用CPU的使用权），而往往会使用到延时函数（该方式线程会让出CPU使用权），通过使用延时列表管理延时线程，从而实现对线程的延时。在RTOS下使用延时函数，内核把暂时不需执行的线程，插入到延时列表中，让出CPU的使用权，并对线程进行调度。

2.6 事件、消息队列、信号量与互斥量

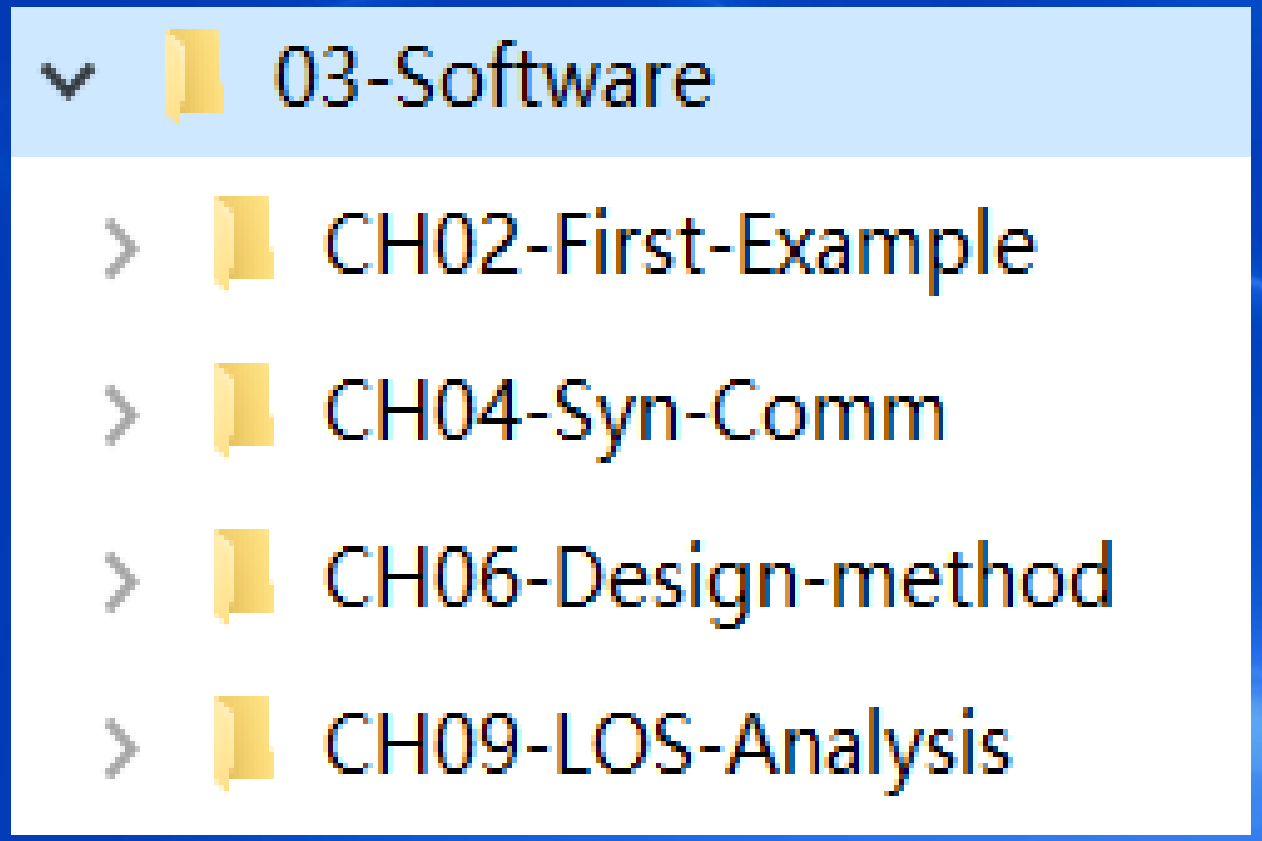
在RTOS中，当为了协调中断与线程之间或者线程与线程之间同步，但不需要传送数据时，常采用事件作为手段。

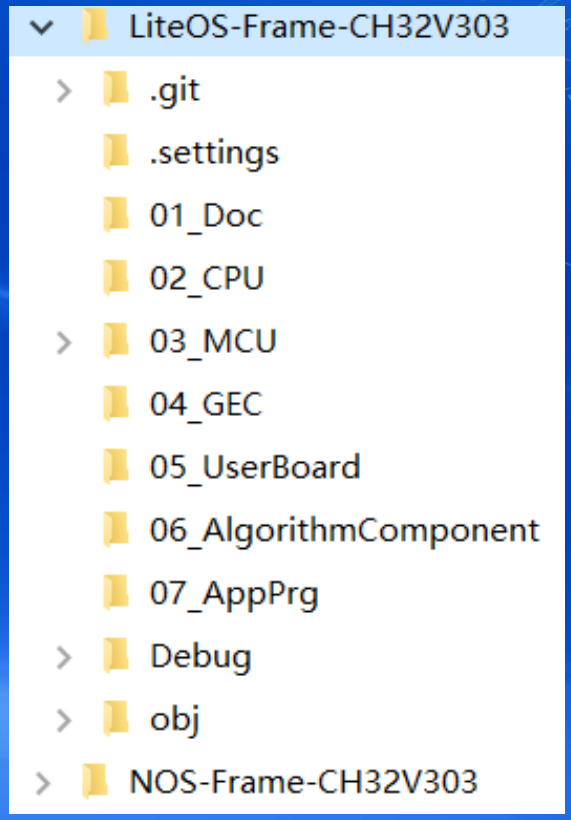
在RTOS中，如果需要在线程间或线程与中断间传送数据，那就需要采用消息队列作为同步与通信手段。

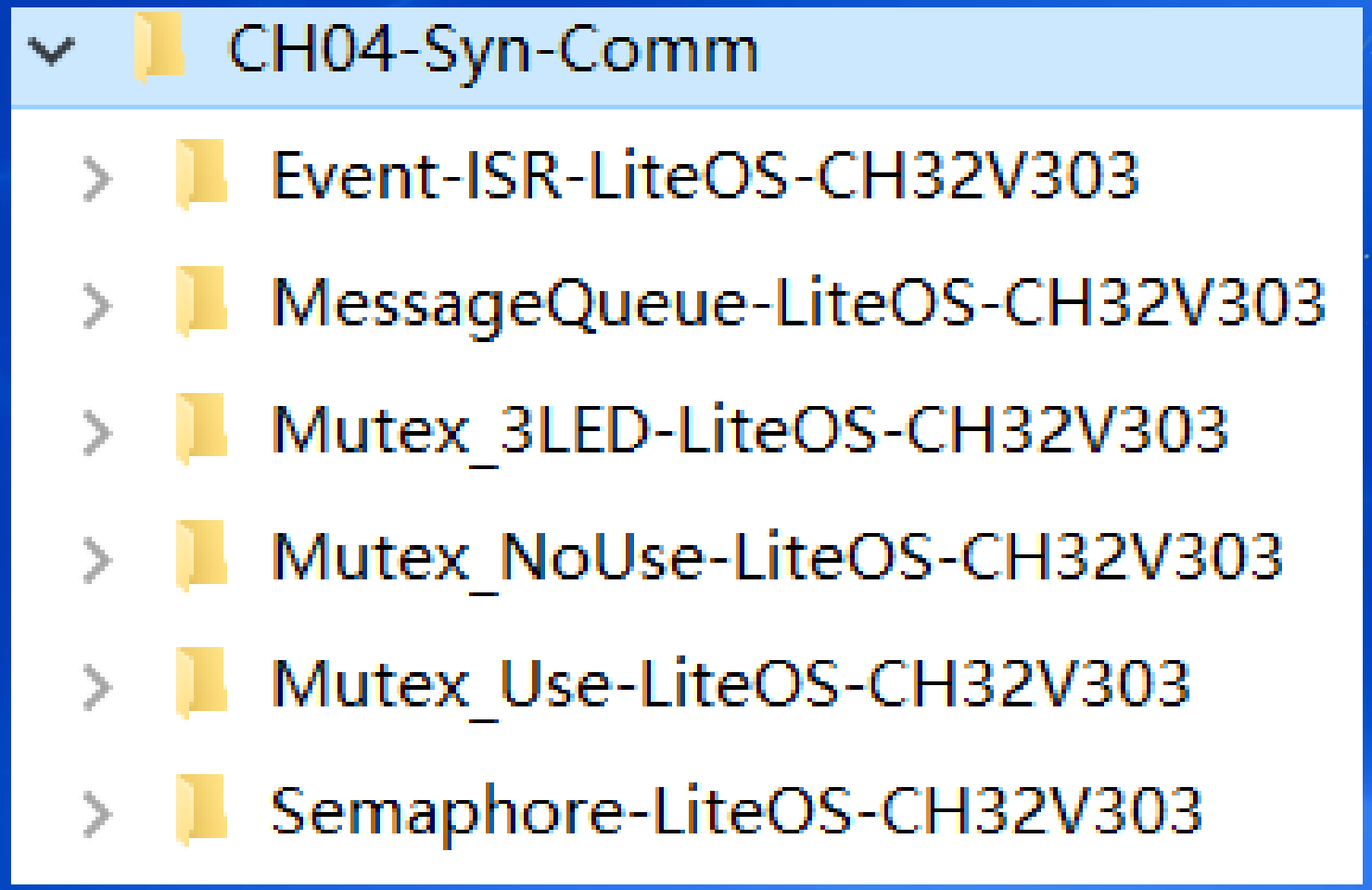
当共享资源有限时，可以采用信号量来表达资源可使用的次数，当线程获得信号量时就可以访问该共享资源了。

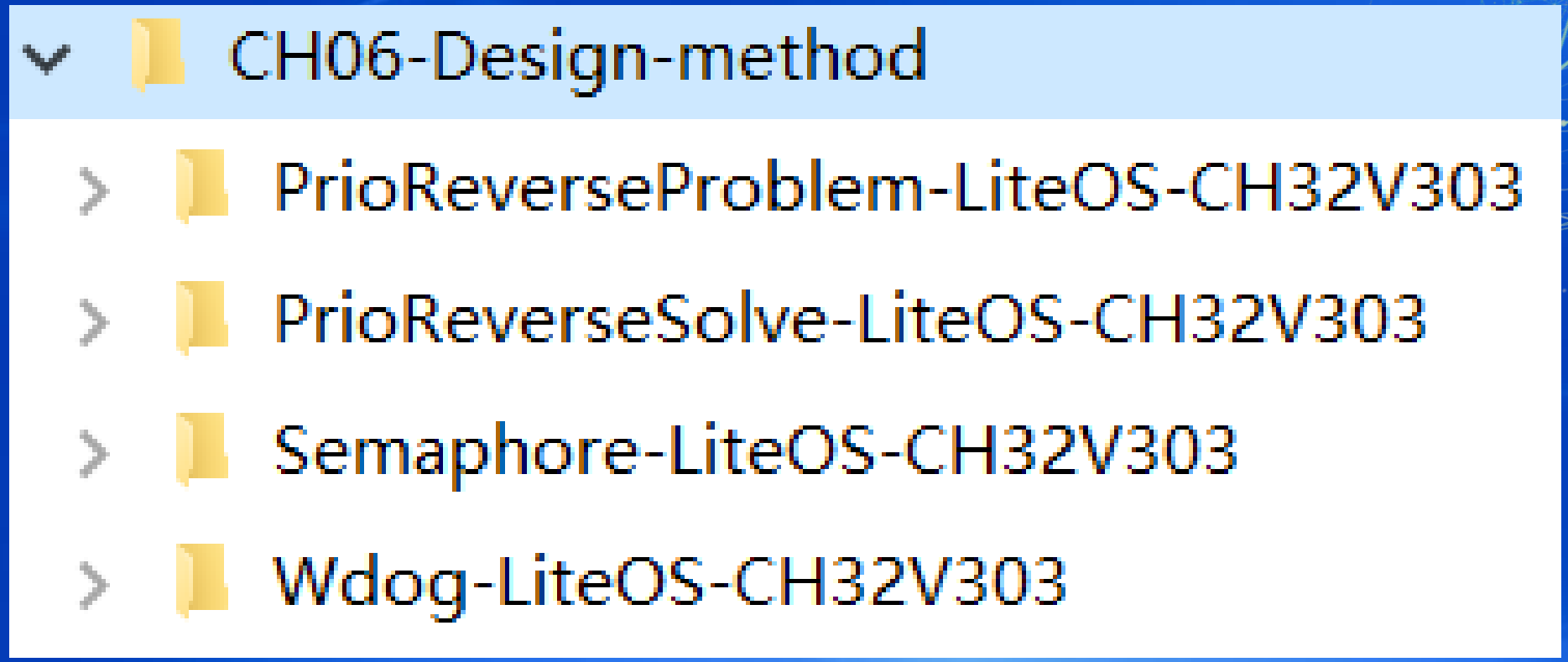
当共享资源只有一个时，为了确保在某个时刻只有一个线程能够访问该共享资源，可以考虑采用互斥量来实现。

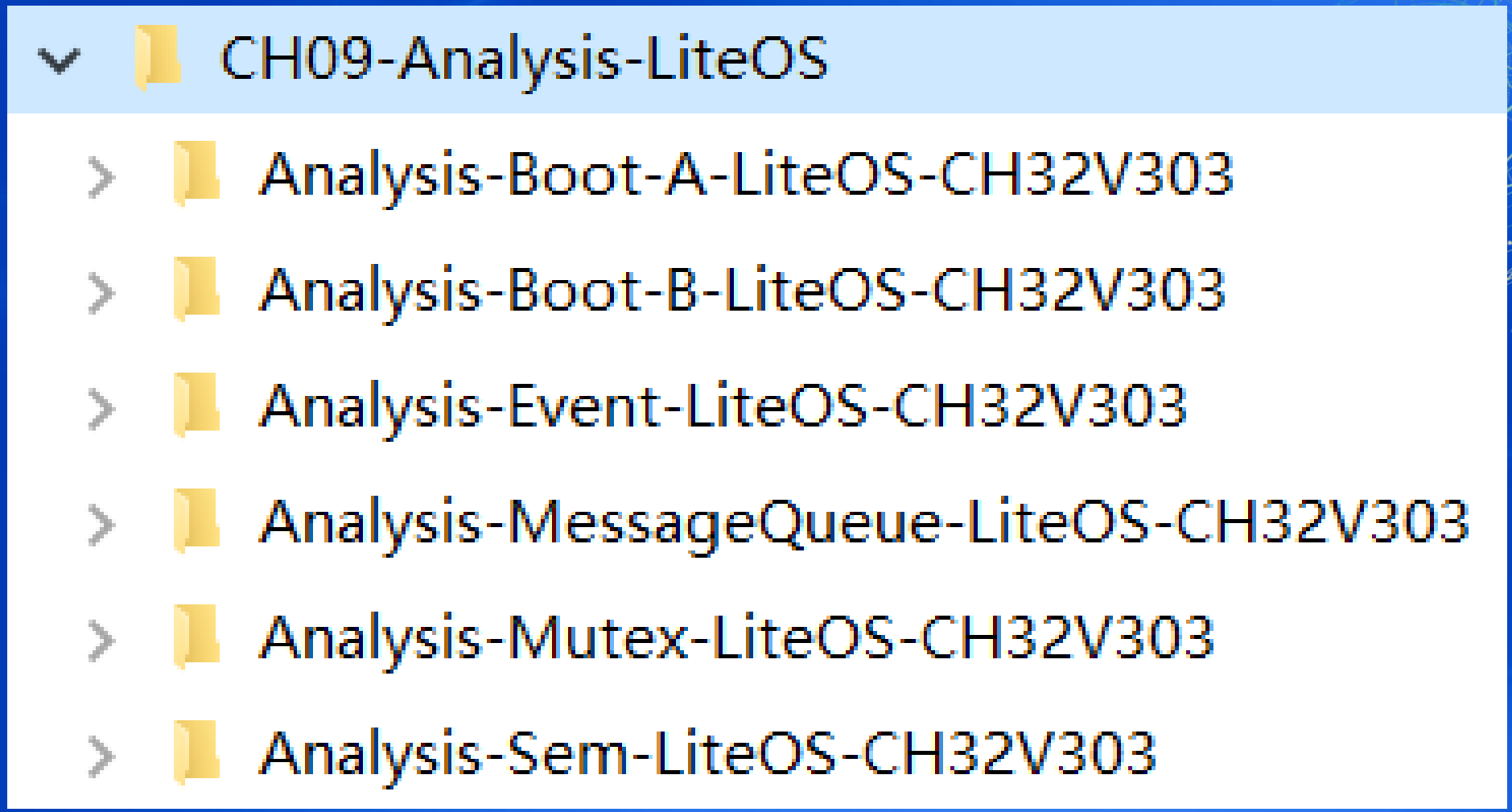
三、实践源代码导引


03-Software
CH02-First-Example
CH04-Syn-Comm
CH06-Design-method
CH09-LOS-Analysis


LiteOS-Frame-CH32V303
.git
.settings
01_Doc
02_CPU
03_MCU
04_GEC
05_UserBoard
06_AlgorithmComponent
07_AppPrg
Debug
obj
NOS-Frame-CH32V303

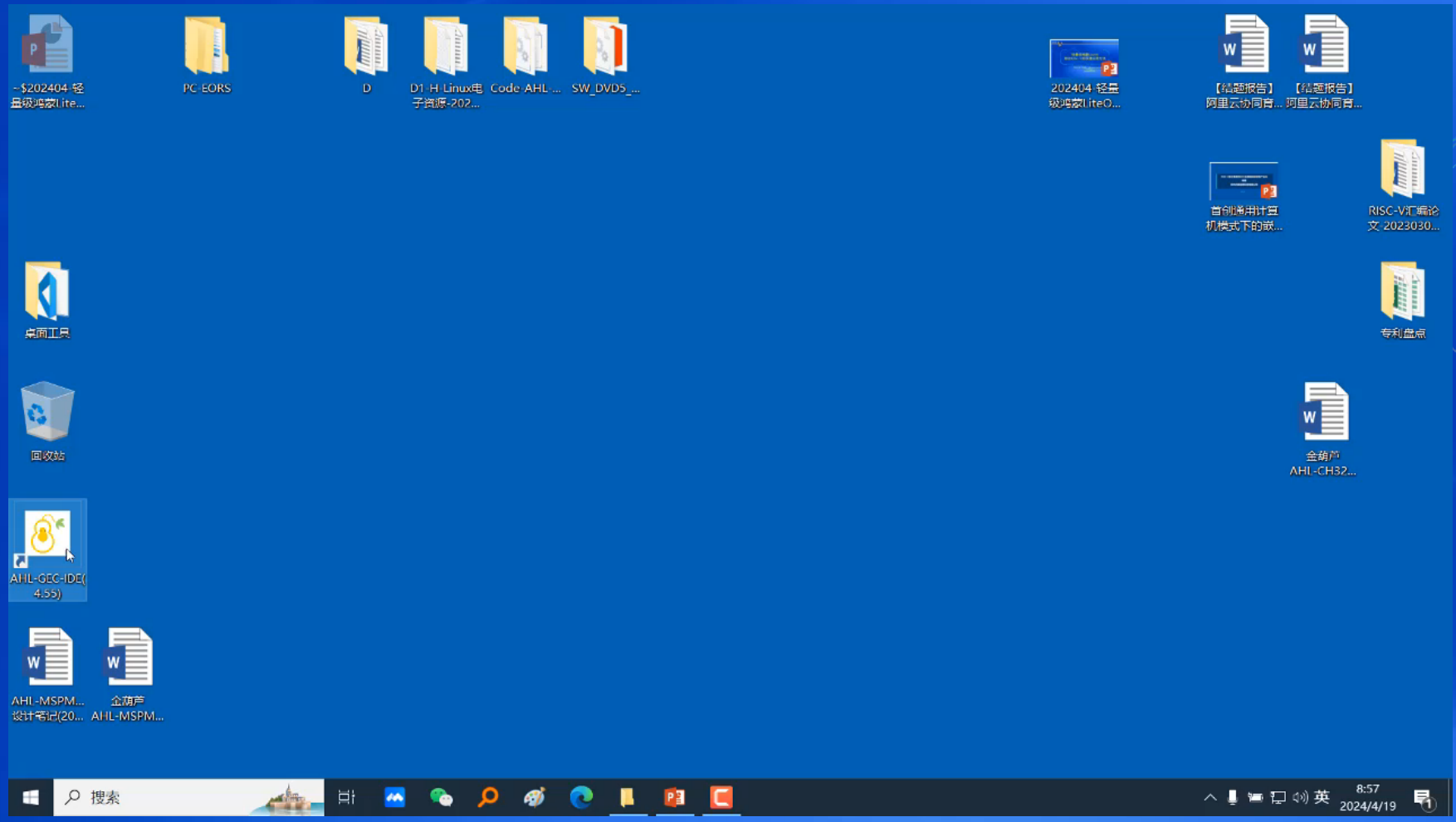

CH04-Syn-Comm
Event-ISR-LiteOS-CH32V303
MessageQueue-LiteOS-CH32V303
Mutex_3LED-LiteOS-CH32V303
Mutex_NoUse-LiteOS-CH32V303
Mutex_Use-LiteOS-CH32V303
Semaphore-LiteOS-CH32V303


CH06-Design-method
PrioReverseProblem-LiteOS-CH32V303
PrioReverseSolve-LiteOS-CH32V303
Semaphore-LiteOS-CH32V303
Wdog-LiteOS-CH32V303


CH09-Analysis-LiteOS
Analysis-Boot-A-LiteOS-CH32V303
Analysis-Boot-B-LiteOS-CH32V303
Analysis-Event-LiteOS-CH32V303
Analysis-MessageQueue-LiteOS-CH32V303
Analysis-Mutex-LiteOS-CH32V303
Analysis-Sem-LiteOS-CH32V303

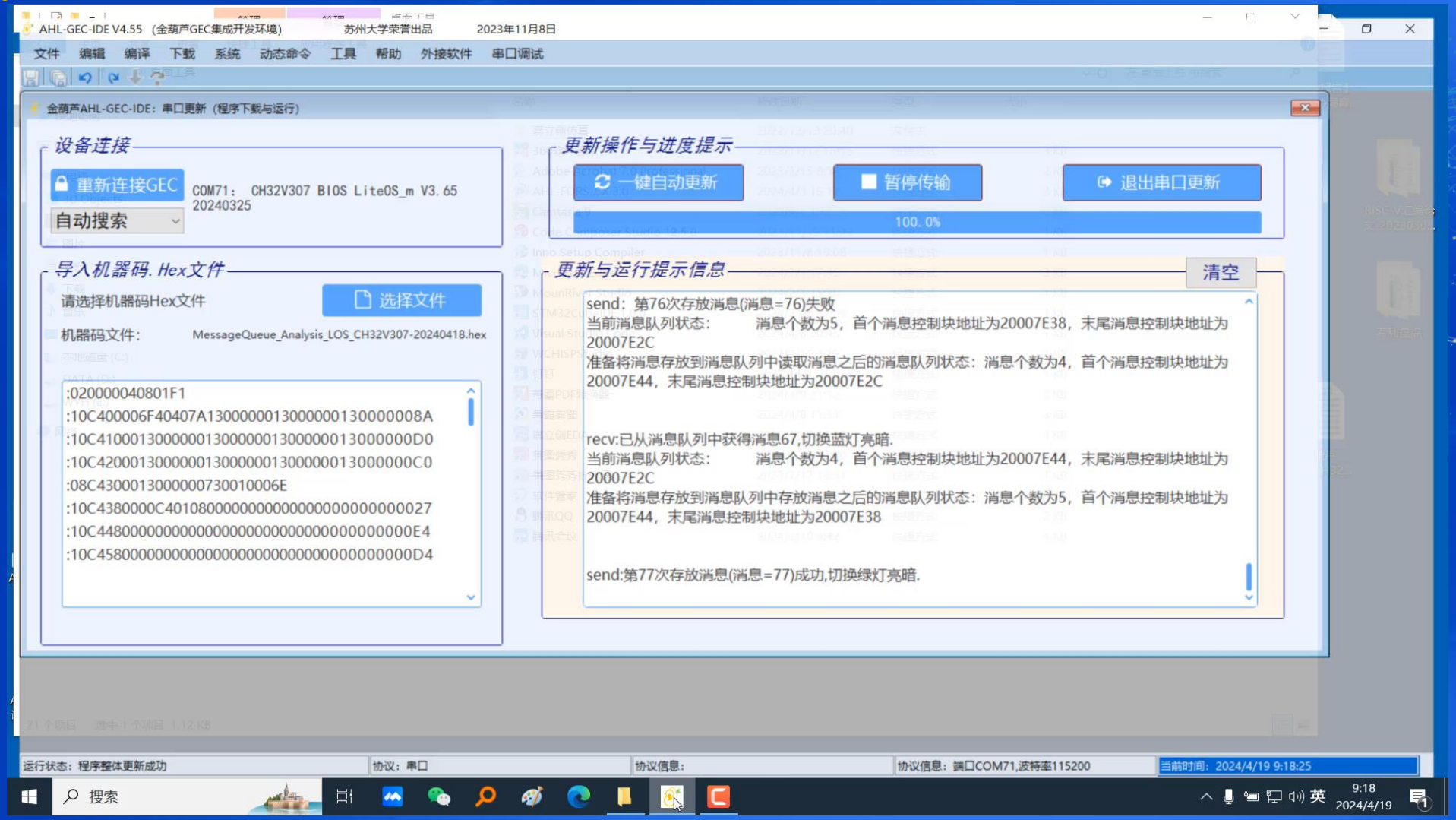
四、实践操作演示

演示延时函数、事件、消息队列、互斥量


~$202404 轻盈级鸿蒙Lite...
PC EORS
D
D1 H Linux电子资源-202...
Code AHL ...
SW_DVDS_...
202404 轻盈级鸿蒙LiteO...
【结题报告】阿里云协同育...
【结题报告】阿里云协同育...
首创浦用计算机模式下的就...
RISC-V汇编论文 2023030...
桌面工具
专利盘点
回收站
金葫芦 AHL-CH32...
AHL-GEC-IDE(4.55)
AHL-MSPM... 设计笔记(20...
金葫芦 AHL-MSPM...
搜索
英
8:57
2024/4/19

五、源码分析方法


AHL-GEC-IDE V4.54 （金葫芦GEC集成开发环境） 苏州大学荣誉出品 2023年9月25日
文件 编辑 编译 下载 系统 动态命令 工具 帮助 外接软件 串口调试
金葫芦AHL-GEC-IDE：串口更新（程序下载与运行）
设备连接
重新连接GEC
COM26： CH32V307 BIOS LiteOS_m V3.65 20240325
自动搜索
更新操作与进度提示
一键自动更新
暂停传输
退出串口更新
100.0%
导入机器码.Hex文件
请选择机器码Hex文件
选择文件
机器码文件： Mutex_Analysis_LOS_CH32V307-20240410.hex
:020000040801F1
:10C400006F40500E13000000130000001300000E6
:10C41000130000001300000013000000130000D0
:10C42000130000001300000013000000130000C0
:08C43000130000073001006E
:10C4380000C401080000000000000000000000027
:10C448000000000000000000000000000000000E4
:10C458000000000000000000000000000000000D4
:10C468006004020800000000700D020800000000CF
:10C47800000000000000000000000000000000B4
更新与运行提示信息
清空
互斥锁=1,表示已锁定(其所有者线程=2000664C),互斥量申请失败.
调用OsSchedTaskWait前互斥量阻塞列表（20006E8C）中的线程：20006E44->20006E44->20006E44
将当前运行线程(200066C4)放入互斥量阻塞列表(20008598)
调用OsSchedTaskWait后互斥量阻塞列表（20006E44）中的线程：200066C4->20006E44->200066C4
开始调度，从就绪列表获取优先级最高的线程(2000673C)
第一次进入红灯线程!
互斥锁=1,表示已锁定(其所有者线程=2000664C),互斥量申请失败.
调用OsSchedTaskWait前互斥量阻塞列表（20006E8C）中的线程：20006E44->200066C4->20006E44
将当前运行线程(2000673C)放入互斥量阻塞列表(200089A0)
调用OsSchedTaskWait后互斥量阻塞列表（20006E44）中的线程：200066C4->2000673C->20006E44
开始调度，从就绪列表获取优先级最高的线程(2000655C)


AHL-GEC-IDE V4.55 (金葫芦GEC集成开发环境)
苏州大学荣誉出品
2023年11月8日
文件 编辑 编译 下载 系统 动态命令 工具 帮助 外接软件 串口调试
金葫芦AHL-GEC-IDE：串口更新（程序下载与运行）
设备连接
重新连接GEC
COM71： CH32V307 BIOS LiteOS_m V3.65
20240325
自动搜索
导入机器码.Hex文件
请选择机器码Hex文件
选择文件
机器码文件：
MessageQueue_Analysis_LOS_CH32V307-20240418.hex
:020000040801F1
:10C400006F40407A13000000130000001300000008A
:10C41000130000001300000013000000130000D0
:10C42000130000001300000013000000130000C0
:08C43000130000007300100 6E
:10C4380000C401080000000000000000000000027
:10C448000000000000000000000000000000000E4
:10C458000000000000000000000000000000000D4
更新操作与进度提示
一键自动更新
暂停传输
退出串口更新
100.0%
更新与运行提示信息
清空
send：第76次存放消息(消息=76)失败
当前消息队列状态： 消息个数为5，首个消息控制块地址为20007E38，末尾消息控制块地址为20007E2C
准备将消息存放到消息队列中读取消息之后的消息队列状态：消息个数为4，首个消息控制块地址为20007E44，末尾消息控制块地址为20007E2C
recv:已从消息队列中获得消息67,切换蓝灯亮暗.
当前消息队列状态： 消息个数为4，首个消息控制块地址为20007E44，末尾消息控制块地址为20007E2C
准备将消息存放到消息队列中存放消息之后的消息队列状态：消息个数为5，首个消息控制块地址为20007E44，末尾消息控制块地址为20007E38
send:第77次存放消息(消息=77)成功,切换绿灯亮暗.
运行状态：程序整体更新成功
协议：串口
协议信息：
协议信息：端口COM71,波特率115200
当前时间：2024/4/19 9:18:25
搜索
英
9:18
2024/4/19

六、小结

（1）复杂问题简单化
（2）梳理知识体系，分清知识层次，确定目标定位；
（3）给出模板，“照葫芦画瓢”；
（4）规范编程实践；
（5）降低学习硬件成本。


Thank You