**福建信息职业技术学院教案**

第 7 号

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | 嵌入式操作系统及应用 | | 授课日期 |  |
| 班 级 | 物联网2411|物联网2412 | | 课堂类型 | 一体化 |
| 教 材 | OpenHarmony嵌入式系统原理与应用——基于RK2206芯片 | | | |
| 章节名称 | 基础开发  信号量 | | | |
| 目的要求 | - 理解OpenHarmony LiteOS-M信号量的核心概念（同步与互斥的工具，基于“计数器+等待队列”实现）； - 掌握信号量核心API（LOS\_SemCreate()创建、LOS\_SemPend()等待、LOS\_SemPost()释放）的参数、返回值及应用场景（同步用初始值0，互斥用初始值1）； - 明确信号量与任务、定时器的区别（任务是运行单元，信号量是任务间协调工具）。 | | | |
| 学情分析 | 1. 基础：已掌握任务创建、定时器开发流程，具备C语言函数调用基础，但对“任务间协调机制”（同步/互斥）认知空白，易混淆信号量与任务、定时器的功能边界； 2. 难点：易误解信号量初始值的含义（如初始值0用于“等待-释放”同步，初始值1用于“独占资源”互斥）、LOS\_SemPend()的“等待超时”逻辑，以及配置文件中编译目标与库名的匹配； 3. 需求：需通过“问题案例（如任务执行顺序混乱）+信号量解决演示”降低抽象难度，结合错误演示（如设错初始值）强化API参数理解。 | | | |
| 重 难 点  分 析 | 1. 重点： - API应用：LOS\_SemCreate()参数配置（初始值count=0/1、信号量ID、信号量名），LOS\_SemPost()释放信号量（计数器+1），LOS\_SemPend()等待信号量（计数器-1，无信号则阻塞）； - 编译配置：修改vendor/lockzhiner/rk2206/sample/BUILD.gn添加./a2\_kernel\_semaphore:sem\_example，修改device/lockzhiner/rk2206/sdk\_liteos/Makefile添加-lsem\_example； - 结果验证：通过串口观察“任务A释放信号量→任务B等待信号量后打印”的同步效果（如任务A每2秒释放1次，任务B仅在收到信号后打印）。 2. 难点： - 信号量机制理解：同步（初始值0，“先等待后释放”，解决任务执行顺序问题）与互斥（初始值1，“独占资源”，解决资源竞争问题）的区别； - API错误处理：识别LOS\_SemCreate()的失败返回值（如LOS\_ERRNO\_SEM\_COUNT\_INVALID代表初始值错误）； - 配置关联：确保BUILD.gn的编译目标路径与Makefile的库名（sem\_example）完全一致 | | | |
| 信息化应用方法 | 网络教学平台、视频、开发板，开展现场教学，通过项目任务驱动进行混合式教学； | | | |
| **思政元素**  **融合设计** | 思政元素 | 融入方式 | | |
| 民族自豪感、  科技报国情怀 | |  |  | | --- | --- | | 1. 讲解鸿蒙信号量在国产工业控制器中的应用：如某国产智能机床通过信号量实现“主轴任务”与“进给任务”的同步，确保加工精度； 2. 对比国外嵌入式系统（如FreeRTOS）信号量的适配复杂度，强调鸿蒙系统“API本土化设计、源码自主可控”的优势，引导学生认同国产嵌入式生态。 | 1. 讲解鸿蒙信号量在国产工业控制器中的应用：如某国产智能机床通过信号量实现“主轴任务”与“进给任务”的同步，确保加工精度； 2. 对比国外嵌入式系统（如FreeRTOS）信号量的适配复杂度，强调鸿蒙系统“API本土化设计、源码自主可控”的优势，引导学生认同国产嵌入式生态。 | | | |
| 作业布置 |  | | | |
| 参考资料 | 1.OpenHarmony嵌入式系统原理与应用——基于RK2206芯片  2.https://gitee.com/Lockzhiner-Electronics/lockzhiner-rk2206-openharmony3.0lts.git | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 教学  环节 | 教学  内容 | 教师活动 | 学生活动 | 设计意图 |
| 课前 | 预习（30分钟） | 1. 在学习平台上传预习资料： - 信号量与任务、定时器的对比表格（含功能、适用场景）； - 鸿蒙信号量API参数图解、LOS\_SemCreate()错误码查询表； - 信号量同步/互斥场景动画视频（如“任务A通知任务B执行”的同步案例）； 2. 发布预习任务：标注LOS\_SemCreate()的3个核心参数（初始值、信号量ID、信号量名），记录“为什么同步场景初始值设为0”的疑问。 | 1. 观看动画视频，绘制“信号量vs任务vs定时器”对比表； 2. 记录预习疑问（如“信号量计数器的作用”），在平台留言互动。 | 用可视化资源降低抽象难度，衔接旧知（任务、定时器），提前铺垫信号量基础概念，培养自主学习能力。 |
| 课中 | 课程介绍（5min） | 1. 明确本节课核心目标：掌握信号量API与开发流程，理解同步/互斥机制，认同国产系统的技术优势； 2. 串联课程逻辑：从“任务执行顺序混乱问题”到“信号量解决方案”，再到“国产设备应用”，融入思政目标。 | 1. 记录学习目标，标注“同步/互斥区别”“API参数”两个重点； 2. 提出对“信号量实际应用”的好奇（如“智能家居里如何用信号量协调任务”）。 | 清晰学习方向，激发对信号量解决实际问题的兴趣，关联思政目标。 |
| 任务导入  （10min） | 1. 演示两组对比实验： - 实验1（无信号量）：任务A（打印“释放信号”）与任务B（打印“收到信号”）执行顺序混乱（有时B先打印）； - 实验2（有信号量）：添加信号量后，任务B仅在任务A释放信号后打印，顺序稳定； 2. 提问引导：“为什么实验1会混乱？实验2的信号量如何解决顺序问题？” | 1. 观察两组实验的串口日志，记录“有无信号量”的执行差异； 2. 分组讨论问题，初步关联“信号量”与“任务顺序控制”的关系。 | 用直观对比实验呈现信号量的核心价值，通过问题引导聚焦“同步机制”这一核心知识点。 |
| 知识储备  （15min） | 1. 分层讲解核心知识： - 信号量本质：类比“电影院检票”（计数器=余票，等待队列=排队观众），同步（按顺序检票）与互斥（一人一座）的区别； - API解析： - LOS\_SemCreate(UINT32 count, UINT32 \*semID, const CHAR \*name)：count为初始值（同步0/互斥1），semID为信号量ID，name为信号量名； - LOS\_SemPost(UINT32 semID)：释放信号量（计数器+1）； - LOS\_SemPend(UINT32 semID, UINT32 timeout)：等待信号量（计数器-1，超时则返回错误）； - 错误码解读：如LOS\_ERRNO\_SEM\_COUNT\_INVALID代表初始值超出范围（LiteOS-M信号量初始值通常0-32767）； 2. 思政融入：穿插讲解某国产工业设备（搭载鸿蒙系统）通过信号量实现“传感器数据采集任务”与“数据上传任务”的同步，对比国外系统的适配成本。 | 1. 绘制“信号量同步流程图”，记录API参数表格； 2. 举手提问“互斥场景中，两个任务同时等待信号量会怎样”，标记“错误码查询方法”。 | 用生活化类比降低抽象难度，结合国产设备案例强化思政认知，夯实理论基础，化解API理解难点。 |
| 任务导入  （5min） | . 承接知识储备，明确本次实操任务： - 在vendor/lockzhiner/rk2206/sample下创建a2\_kernel\_semaphore文件夹； - 编写同步场景代码（任务A每2秒释放信号量，任务B等待信号量后打印）； - 修改配置文件，烧写程序并验证串口日志； 2. 强调实操关键点： - 代码结构：sem\_example()创建信号量与任务，task\_release()释放信号量，task\_wait()等待信号量； - 配置文件：BUILD.gn的编译目标路径、Makefile的库名必须匹配。 | 1. 记录实操步骤框架（创建文件夹→写代码→改配置→烧写验证）； 2. 标注“文件夹名是a2\_kernel\_semaphore”“库名是sem\_example”两个易错点。 | 将理论落地到具体任务，明确实操重点与易错点，降低操作失误率，为后续实操铺垫。 |
| 知识储备  （10min） | 1. 补充实操细节知识： - 代码模板：UINT32 sem\_id = 0;初始化信号量ID，LOS\_SemCreate(0, &sem\_id, "sync\_sem");创建同步信号量； - 配置文件修改语法： - BUILD.gn添加编译目标：./a2\_kernel\_semaphore:sem\_example（路径+目标名）； - Makefile添加库名：hardware\_LIBS += -lsem\_example（-l+库名）； - 错误预判：演示“信号量ID未初始化导致LOS\_SemPost()失败”的日志，教学生根据错误码（如LOS\_ERRNO\_SEM\_ID\_INVALID）定位问题。 | . 记录代码模板片段（如信号量创建与释放的完整代码）； 2. 模拟“库名少写字母”的错误，练习通过日志排查配置问题。 | 补充实操中的细节知识，提前预判常见错误，提升学生问题解决能力，确保实操顺利推进。 |
|  | 任务实施  （40min） | 1. 分步演示+巡视指导： ① 终端创建a2\_kernel\_semaphore文件夹（mkdir -p a2\_kernel\_semaphore）； ② 编写sem\_example.c： - 包含头文件#include "los\_sem.h"； - 定义task\_release()（LOS\_Msleep(2000); LOS\_SemPost(sem\_id);）和task\_wait()（LOS\_SemPend(sem\_id, LOS\_WAIT\_FOREVER); printf("Task wait get semaphore\n");）； - 在sem\_example()中创建信号量（初始值0）和两个任务； ③ 修改BUILD.gn：在vendor/lockzhiner/rk2206/sample/BUILD.gn的features或编译目标中添加./a2\_kernel\_semaphore:sem\_example； ④ 修改Makefile：在device/lockzhiner/rk2206/sdk\_liteos/Makefile的hardware\_LIBS后添加-lsem\_example； ⑤ 烧写程序，打开串口软件（如SecureCRT），设置波特率（通常115200），观察日志（任务B每2秒打印1次，与任务A释放信号同步）； 2. 重点帮扶：针对“信号量ID未传递到任务函数”“配置文件路径错误”等问题，一对一指导排查（如通过printf打印信号量ID验证是否创建成功）。 | 1. 跟随演示逐步操作，每完成一步自我检查（如核对BUILD.gn路径是否正确）； 2. 遇到错误先对照预习资料中的错误码表排查，无法解决则举手求助； 3. 成功看到同步日志后，拍照记录，并尝试修改初始值为1，观察互斥场景效果。 | 通过“分步演示+针对性指导”突破重点，强化动手能力；鼓励学生自主探索互斥场景，深化对信号量机制的理解。 |
|  | 任务总结  （5min） | 1. 流程梳理：用思维导图回顾“信号量开发全流程”（需求分析→API选择→代码编写→配置修改→编译烧写→结果验证）； 2. 重难点强化： - 核心机制：同步（初始值0，解决顺序）、互斥（初始值1，解决竞争）； - 配置关键：BUILD.gn与Makefile的“路径-库名”匹配； - 错误排查：先看API参数（如初始值、ID），再查配置文件； 3. 思政升华：展示国产智能家居网关（搭载鸿蒙+RK2206芯片）通过信号量协调“Wi-Fi连接任务”与“设备控制任务”的案例，强调自主可控技术对嵌入式产业的支撑作用。 | 1. 补充思维导图细节，标记“同步/互斥区别”“配置关联”两个核心点； 2. 分享实操心得（如“原来信号量ID必须在任务间传递才能用”）。 | 巩固知识点与实操流程，结合实际应用案例深化思政认知，形成“理论-实操-应用”的完整学习闭环。 |
| 作业 |  | 1. 在学习通发布作业： - 实操作业要求：代码命名格式为“sem\_example\_姓名.c”，日志截图需标注“同步/互斥场景”“信号量初始值”； - 预习作业要求：简述LOS\_QueueCreate()的功能（不少于200字），对比信号量与消息队列的适用场景； 2. 提示下次课重点：鸿蒙内核消息队列（任务间传递数据的工具）。 | 1. 记录作业要求，规划完成时间（实操2小时+预习1小时）； 2. 课后立即回顾本节课代码，为修改互斥场景做准备。 | 通过作业验收实操成果，强化信号量机制理解；预习消息队列内容，为后续课程衔接铺垫，保持知识连贯性。 |
| 课后 |  | 1. 上传教学资源： - 本节课PPT（含信号量机制图解、API参数表）； - 实操步骤视频（含配置文件修改、串口日志解读）； - 信号量错误排查手册（常见错误+解决方法）； 2. 开放答疑通道： - 24小时内回复学习平台的代码问题； - 汇总“信号量ID未初始化”“配置文件路径错误”等高频问题，下次课开篇讲解。 | 1. 登录学习平台下载资源，对照视频复盘实操过程； 2. 完成作业时遇到问题及时在平台提问，查看同学的疑问及解答，补充自身知识漏洞。 | 提供课后复习支持，及时解决学习问题；通过高频错误汇总优化后续教学，帮助学生查漏补缺，巩固学习效果。 |