**福建信息职业技术学院教案**

第 17 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  互斥 | | | |
| 目的要求 | - 理解互斥锁本质（特殊二值信号量，保护共享资源）； - 掌握核心API（LOS\_MuxCreate()创建、LOS\_MuxPend()申请、LOS\_MuxPost()释放）功能； - 明确互斥锁解决的问题（资源竞争、优先级翻转）。 | | | |
| 学情分析 | 1. 基础：已掌握任务、信号量开发，具备C语言共享变量认知，但对互斥锁与信号量区别、优先级翻转陌生； 2. 难点：易混淆LOS\_MuxPend()与LOS\_SemPend()，忽略互斥锁不能在中断中使用； 3. 需求：需通过对比实验、错误演示降低理解难度。 | | | |
| 重 难 点  分 析 | 1. 重点： - API应用：LOS\_MuxCreate(&m\_mutex\_id)创建、LOS\_MuxPend(m\_mutex\_id, LOS\_WAIT\_FOREVER)申请； - 编译配置：修改vendor/lockzhiner/rk2206/sample/BUILD.gn添加./a4\_kernel\_mutex:mutex\_example，Makefile添加-lmutex\_example； - 结果验证：串口观察“write\_thread写数据后3秒，read\_thread才读数据”。 2. 难点： - 互斥锁逻辑：独占共享资源（m\_data），避免竞争； - 与信号量区别：互斥锁解决优先级翻转，信号量不能。 | | | |
| 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 | 通过国产软件的崛起及应用事例的讲解，树立学生的民族自豪感和科技报国情怀 | | |
| 作业布置 | 1. 实操任务： - 修改代码：在read\_thread中添加超时时间（3000ms），观察超时后日志，提交代码与截图； - 验证区别：用信号量实现相同功能，对比互斥锁与信号量的运行差异。 2. 预习任务： - 阅读参考资料中“鸿蒙内核消息队列”章节，了解LOS\_QueueSend()基本功能。 | | | |
| 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. 上传预习资料： - 互斥锁与信号量对比表、优先级翻转动画视频； - 互斥锁API参数图解； 2. 发布任务：标注LOS\_MuxPend()超时参数含义，记录“互斥锁为何能解决优先级翻转”的疑问。 | 1. 观看视频，记录对比表； 2. 提交预习疑问，平台互动。 | 课前让学生复习相关教学内容，提高课堂教学效率，让学生更好的理解授课内容。 |
| 课中 | 课程介绍（5min） | 1. 明确目标：掌握互斥锁API与资源保护，理解国产内核优势； 2. 串联逻辑：从“资源竞争问题”到“互斥锁解决”，融入思政目标。 | 1. 记录核心目标； 2. 提问“共享资源竞争会导致什么问题”。 | 让学生对本课程有初步了解 |
| 任务导入  （10min） | 1. 演示两组实验： - 实验1（无互斥锁）：write\_thread与read\_thread竞争m\_data，数据读取混乱； - 实验2（有互斥锁）：数据“写→3秒后读”，顺序稳定； 2. 提问：“实验1为何混乱？互斥锁如何实现独占访问？” | 1. 观察日志差异； 2. 分组讨论，梳理“创建→申请→释放”流程。 | 用对比实验凸显互斥锁价值，聚焦核心逻辑。 |
| 知识储备  （15min） | 1. 讲解核心知识： - 互斥锁本质：类比“单人卫生间钥匙”（独占使用）； - API解析：LOS\_MuxCreate()（创建）、LOS\_MuxPend()（申请）、LOS\_MuxPost()（释放）； - 解决问题：资源竞争、优先级翻转； 2. 思政融入：穿插国产设备应用案例，对比国外系统。 | 1. 绘制“互斥锁工作流程图”； 2. 标注API关键参数。 | 夯实理论，结合类比降低抽象难度。 |
| 任务导入  （5min） | 1. 明确实操任务：创建a4\_kernel\_mutex文件夹，编写互斥锁保护共享数据代码； 2. 强调易错点：互斥锁需成对使用（申请后必须释放）。 | 1. 记录实操步骤； 2. 标注“避免忘记LOS\_MuxPost()”。 | 明确任务边界，减少失误。 |
| 知识储备  （10min） | 1. 补充细节： - 代码结构：mutex\_example()创建互斥锁与任务，write\_thread/read\_thread操作共享数据； - 配置语法：BUILD.gn与Makefile的库名、路径匹配； 2. 错误演示：忘记释放互斥锁，展示任务阻塞日志。 | 1. 记录代码模板； 2. 练习排查阻塞问题。 | 补充实操细节，提升问题解决能力。 |
|  | 任务实施  （40min） | ① 创建a4\_kernel\_mutex文件夹； ② 编写代码（含互斥锁创建、任务操作共享数据）； ③ 修改BUILD.gn和Makefile； ④ 烧写验证； 2. 重点帮扶：解决API调用顺序、配置路径错误。 | 1. 跟随操作，每步自查； 2. 遇错先排查，再求助； 3. 成功后拍照记录日志。 | 通过实操突破重点，解决难点 |
|  | 任务总结  （5min） | 1. 梳理流程：代码→配置→烧写→验证； 2. 强化重点：互斥锁成对使用、与信号量区别； 3. 思政升华：国产内核互斥锁的自主价值。 | 1. 补充流程笔记； 2. 分享实操心得。 | 巩固知识，深化思政认知。 |
| 作业 |  | 1. 学习通发布作业：明确代码命名、截图要求； 2. 提示下次课重点：消息队列。 | 1. 记录作业要求； 2. 规划完成时间。 | 验收成果，铺垫后续课程。 |
| 课后 |  | 1. 上传PPT、实操视频、错误排查手册； 2. 24小时内回复问题，汇总高频错误。 | 1. 下载资源复盘； 2. 提交疑问，查看解答。 | 提供复习支持，查漏补缺。 |