**福建信息职业技术学院教案**

第 10 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  [事件](https://gitee.com/Lockzhiner-Electronics/lockzhiner-rk2206-openharmony3.0lts/blob/master/vendor/lockzhiner/rk2206/samples/a6_kernel_event/README_zh.md) | | | |
| 目的要求 | - 理解OpenHarmony LiteOS-M事件的核心概念（任务间同步机制，无数据传输，32位无符号整型变量每一位代表一个独立事件）； - 掌握事件核心API（LOS\_EventInit()初始化、LOS\_EventRead()等待事件、LOS\_EventWrite()写入事件、LOS\_EventClear()清除事件）的参数、返回值及使用场景（支持一对多/多对多同步）； - 明确事件与信号量的区别（事件无“计数器”，仅同步；信号量有计数器，支持同步+互斥）。 | | | |
| 学情分析 | 1. 基础：已掌握任务、信号量、定时器的开发流程，具备C语言指针/结构体基础，但对“事件掩码”“读取模式（如LOS\_WAITMODE\_AND）”认知空白，易混淆事件与信号量的功能边界； 2. 难点：易误解LOS\_EventRead()的mode参数（如“AND模式需指定事件全发生”）、eventMask（事件掩码，如0x1代表第1位事件）的含义，以及事件控制块EVENT\_CB\_S的初始化逻辑； 3. 需求：需通过“信号量vs事件对比实验”“错误演示（如设错掩码）”降低抽象难度，强化API参数与配置文件的关联性认知。 | | | |
| 重 难 点  分 析 | 1. 重点： - API应用： - LOS\_EventInit(PEVENT\_CB\_S eventCB)：初始化事件控制块（如EVENT\_CB\_S m\_event; LOS\_EventInit(&m\_event)）； - LOS\_EventRead(&m\_event, EVENT\_WAIT, LOS\_WAITMODE\_AND, LOS\_WAIT\_FOREVER)：eventMask=EVENT\_WAIT（如0x1）、mode=LOS\_WAITMODE\_AND（需指定事件全发生）； - LOS\_EventWrite(&m\_event, EVENT\_WAIT)：写入指定事件掩码； - 编译配置：修改vendor/lockzhiner/rk2206/sample/BUILD.gn添加./a6\_kernel\_event:event\_example，修改device/lockzhiner/rk2206/sdk\_liteos/Makefile添加-levent\_example； - 结果验证：通过串口观察“event\_master\_thread每2秒写事件（0x1）并清除，event\_slave\_thread阻塞读事件后每1秒打印”的效果。 2. 难点： - 事件机制理解：eventMask（如0x3代表第1、2位事件）与mode（AND/OR）的匹配逻辑； - 事件与信号量的区别：事件无计数器、仅同步，支持多事件组合；信号量有计数器、支持互斥； - API错误处理：识别LOS\_EventRead()的失败返回值（如LOS\_ERRNO\_EVENT\_EVENTMASK\_INVALID代表掩码无效）。 | | | |
| 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. 讲解鸿蒙事件在国产工业监控设备中的应用：如某国产生产线通过事件实现“温度传感器任务+压力传感器任务”同步触发“设备停机任务”，确保生产安全； 2. 对比国外嵌入式系统（如FreeRTOS）事件功能的适配复杂度，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 | 1. 实操任务： - 修改事件模式：将LOS\_EventRead()的mode改为LOS\_WAITMODE\_OR，添加第2个事件（EVENT\_WAIT2=0x2），让event\_master\_thread交替写入0x1和0x2，观察event\_slave\_thread的打印效果； - 测试超时逻辑：在LOS\_EventRead()中设置超时时间（如3000毫秒），观察“超时后任务打印错误信息”的效果，提交代码及日志截图。 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\_QueueCreate()的基本功能，思考“事件与消息队列的适用场景区别”（如事件无数据，消息队列可传数据）。 | | | |
| 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： - 事件与信号量的对比表格（含功能、同步模型、数据传输）； - 鸿蒙事件API参数图解（标注eventMask“位含义”）、LOS\_EventRead()错误码查询表； - 事件同步动画视频（如“多传感器任务触发同一处理任务”）； 2. 发布预习任务：标注LOS\_EventRead()的4个参数含义，记录“事件掩码为什么用十六进制”的疑问。 | 1. 观看动画视频，绘制“事件vs信号量”对比表； 2. 记录预习疑问（如“LOS\_WAITMODE\_AND和OR的区别”），在平台留言。 | 用可视化资源降低抽象难度，衔接旧知（信号量），提前铺垫事件基础概念，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：掌握事件API与开发流程，理解多事件同步机制，认同国产系统优势； 2. 串联课程逻辑：从“多任务同步需求”到“事件解决方案”，再到“国产设备应用”，融入思政目标。 | 1. 记录学习目标，标注“事件掩码”“读取模式”两个重点； 2. 提出对“事件实际应用”的好奇（如“智能家居如何用事件协调多设备”）。 | 清晰学习方向，激发对事件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（无事件）：event\_master\_thread（写事件）与event\_slave\_thread（读事件）执行混乱（slave未等待直接打印）； - 实验2（有事件）：添加事件后，slave仅在master写事件后打印，且每2秒触发一次连续打印； 2. 提问引导：“为什么实验1会混乱？实验2的事件如何实现‘写→读’同步？” | 1. 观察两组实验的串口日志，记录“有无事件”的执行差异； 2. 分组讨论问题，初步关联“事件”与“多任务同步”的关系。 | 用直观对比实验呈现事件的核心价值，通过问题引导聚焦“多事件同步”这一核心知识点。 |
| 知识储备  （15min） | 1. 分层讲解核心知识： - 事件本质：类比“快递柜取件”（eventMask=取件码，mode=“需所有码”/“任一码”），一对多（一个快递员投件，多个用户取件）、多对多（多个快递员投件，多个用户取件）同步模型； - API解析： - LOS\_EventInit(PEVENT\_CB\_S eventCB)：初始化事件控制块（需定义EVENT\_CB\_S类型变量，如m\_event）； - LOS\_EventRead(PEVENT\_CB\_S eventCB, UINT32 eventMask, UINT32 mode, UINT32 timeout)：eventMask为事件掩码（如0x1），mode为读取模式（AND/OR），timeout为超时时间； - LOS\_EventWrite(PEVENT\_CB\_S eventCB, UINT32 events)：写入事件掩码（如0x1，对应bit0置1）； - LOS\_EventClear(PEVENT\_CB\_S eventCB, UINT32 eventMask)：清除指定事件（如~m\_event.uwEventID清除所有事件）； - 错误码解读：如LOS\_ERRNO\_EVENT\_EVENTMASK\_INVALID代表事件掩码超出32位范围； 2. 思政融入：穿插讲解某国产智能家居网关（搭载鸿蒙系统）通过事件实现“灯光任务+窗帘任务”同步响应“离家模式”，对比国外系统的适配成本。 | 1. 绘制“事件同步流程图”，记录API参数表格； 2. 举手提问“多对多同步中，多个任务写同一事件会怎样”，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） | 1. 承接知识储备，明确本次实操任务： - 在vendor/lockzhiner/rk2206/sample下创建a6\_kernel\_event文件夹； - 编写事件同步代码（master每2秒写事件并清除，slave阻塞读事件后每秒打印）； - 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：event\_example()初始化事件与任务，event\_master\_thread写事件，event\_slave\_thread读事件； - 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a6\_kernel\_event”“库名是event\_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： - 定义事件控制块：EVENT\_CB\_S m\_event;（全局变量，确保任务间可访问）； - 写事件逻辑：LOS\_EventWrite(&m\_event, 0x1); LOS\_Msleep(2000); LOS\_EventClear(&m\_event, ~m\_event.uwEventID);； - 配置文件修改语法： - BUILD.gn添加编译目标：./a6\_kernel\_event:event\_example（路径+目标名）； - Makefile添加库名：hardware\_LIBS += -levent\_example（-l+库名）； - 错误预判：演示“事件控制块未初始化导致LOS\_EventWrite()失败”的日志，教学生根据错误码（如LOS\_ERRNO\_EVENT\_PTR\_NULL）定位问题。 | 1. 记录代码模板片段（如事件初始化与读写的完整代码）； 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a6\_kernel\_event文件夹（mkdir -p a6\_kernel\_event）； ② 编写event\_example.c： - 包含头文件#include "los\_event.h"； - 定义全局事件控制块：EVENT\_CB\_S m\_event;、事件掩码：#define EVENT\_WAIT 0x1； - 实现event\_master\_thread()（每2秒写事件0x1，后清除事件）； - 实现event\_slave\_thread()（阻塞读事件0x1，读成功后每秒打印）； - 在event\_example()中初始化事件、创建两个优先级为5的任务； ③ 修改BUILD.gn：在vendor/lockzhiner/rk2206/sample/BUILD.gn中添加./a6\_kernel\_event:event\_example； ④ 修改Makefile：在device/lockzhiner/rk2206/sdk\_liteos/Makefile的hardware\_LIBS后添加-levent\_example； ⑤ 烧写程序，打开串口软件（波特率115200），观察日志（master每2秒写事件，slave连续打印2次读事件）； 2. 重点帮扶：针对“事件控制块未全局定义导致任务访问失败”“配置文件路径错误”等问题，一对一指导排查。 | 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mode为OR验证效果。 | 通过“分步演示+针对性指导”突破重点，强化动手能力；鼓励学生自主探索读取模式，深化对事件机制的理解。 |
|  | 任务总结  （5min） | 1. 流程梳理：用思维导图回顾“事件开发全流程”（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：eventMask（位操作）、mode（AND/OR）的匹配逻辑； - 配置关键：BUILD.gn与Makefile的“路径-库名”匹配； - 事件vs信号量：事件无计数器、支持多事件组合，信号量有计数器、支持互斥； 3. 思政升华：展示国产工业监控设备（搭载鸿蒙+RK2206芯片）通过事件实现“多传感器同步触发报警”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“事件掩码”“配置关联”两个核心点； 2. 分享实操心得（如“原来事件控制块必须全局定义才能跨任务访问”）。 | 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“event\_example\_姓名.c”，日志截图需标注“读取模式”“事件掩码”； - 预习作业要求：简述LOS\_QueueCreate()的功能（不少于200字），对比事件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，为修改读取模式做准备。 | 通过作业验收实操成果，强化事件机制理解；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（含事件机制图解、API参数表）； - 实操步骤视频（含配置文件修改、串口日志解读）； - 事件错误排查手册（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“事件控制块未全局定义”“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。 |