summer2022(2)

记录相关基本信息

- 课程主题 RT-Thread线程开发
- 指导单位 RT-Thread
- 指导老师 刘贤良
- 记录时间 2022.7.19
- 課程大纲

忠于本意,按秩序列出所有与本次课程主题相关的内容知识点,并将其作一个简单标签式的归类,便于快速精准地查找

- 参知识点
 - Part1线程基本概念与调度管理
 - 线程的组成:
 - 线程代码 (入口函数)
 - 线程控制块
 - 线程栈
 - 每一个线程就是一个任务,宏观上类比逻辑上的多个 while(1)死循环。微观上是分时复用CPU执行权限,切换时 候为系统配置的tick
 - 多线程开发模式与传统开发模式
 - 轮询开发模式
 - 前后台开发模式
 - 中断保证了与外部交互的实时性(中断里的代码尽量简短)(短而急)
 - 抢占式多线程开发模式 (大部分RTOS)
 - 优先级最低任务是什么? (比低优先级任务还要再 低)
 - 时间片多线程开发模式

- 在同优先级里面适用,相同优先级之间任务可以相 互交替来。
- 线程和中断服务例程是RTT中最基本的调度单位
- 线程中至执行任务的环境叫做线程上下文
- 中断服务例程中执行的环境叫做中断上下文
- 线程具有优先级改建
- 线程具有状态 (是不是自己需要应该运行)
- 调度器保证当前运行的线程一定是系统的最高就绪态的线程
- 时间片调度,他们各自占用CPU的时间可以配置
- RTT调度机制
 - 时间片调度机制
 - 中断具有最高执行权限
 - 讲抢占
- * 线程的状态
 - 初始状态
 - 就绪状态:线程已经满足运行的条件但是不是系统中最高优先级的任务或者是最高优先级但是时间片这个时刻不属于这个线程的
 - 运行状态: 当前系统中最高优先级的且处于就绪态的任务
 - 挂起状态:
 - 关闭状态:退出或者关闭线程
- 线程状态的相关切换
 - 挂起函数难
- 特殊线程
 - 空闲线程
 - idle线程
 - 系统创建最低优先级的线程,线程状态永远为就绪态(要么执行,要么被抢占,不会待在挂起态,是

最后托底的)。回收被删除线程的资源。

- 主线程main 10
 - 入口函数为main-thread_entry(),
- tinish线程
 - 优先级20
- RTOS调度器对比
- RTT调度算法
 - 抢占式调度bitmap算法
 - 时间片调度round-robin算法
- 编程
 - 如何创建线程
 - 静态
 - · 需要外部制定堆栈地址与大小,且需要人为8字节 对齐(浮点相关)ALIGH(RT_ALIGH_SIZE)
 - 需要外部指定线程句柄
 - 动态
 - 由系统自动分配堆栈地址,只需要指定大小
 - 由系统自动分配线程句柄相关内存
 - 栈起始空间, 句柄都没有。
 - 区别
 - 静态
 - 图 结构体指针 句柄
 - RO段(只读数据,CODE段,(不能随便改)RW段 (可读可写)ZI(为0)不需要FLASH
 - 假设所有代码4节变量数组,复制1000个,
 - 启动线程
 - t截止,基本都是指针
 - 线程睡眠
 - 线程睡眠就进入挂起状态

- 线程退出
 - 线程入口函数执行完毕return后,线程将被系统(空闲 线程)自动删除
- * 错误码分析
 - 符号有故障,0才是对的约定俗成

以上内容整理于 幕布文档