

1 All In One

1.1 嵌入式系统概述

1.1.1 定义：专用计算机系统

1. 以应用为中心
2. 以计算机技术为基础
3. 软硬件可剪裁
4. 适应应用系统对功能、可靠性、成本、体积、功耗、严格要求

1.1.2 嵌入式系统特点

1. (同 3) 系统内核小
2. (定义) 专用性强
3. (发展趋势) 系统精简
4. 高实时性的操作系统软件是嵌入式软件的基本要求
5. (发展趋势) 嵌入式软件开发要想走向标准化，就必须使用多任务的操作系统
6. (发展趋势) 嵌入式系统开发需要专门的开发工具和环境

1.1.3 发展趋势

1. 可移植
2. SDK
3. 通用计算机技术
4. embed linux
5. net
6. simplify
7. ui

1.1.4 关键词

片上系统 (System On Chip, SOC) 在单芯片上集成数字信号处理器、微控制器、存储器、数据转换器、接口电路等电路模块，可以直接实现信号的采集、转换、存储、处理等功能，其中知识产权核 (IP Core) 设计是 SOC 设计的基础

IP Core IP 核是指具有知识产权的、功能具体的、接口规范的、可在多个集成电路设计中重复使用的功能模块，是实现系统芯片（SOC）的基本构件

- IP 软核: 用硬件描述语言文本形式提交给用户，其中不包含任何具体的物理信息。软核是以源代码的形式提供，IP 知识产权不易维护
- IP 硬核: 基于半导体工艺的物理设计，提供给用户的形式是电路物理结构掩模版图和全套工艺文件。IP 易于保护，但灵活性与可移植性差
- IP 固核: 设计介于软核和硬核之间

1.1.5 嵌入式系统的组成

1.1.5.1 简化

1. 处理器
2. 存储器
3. I/O
4. 软件

1.1.5.2 具体

1. 硬件层, 分类 SoC

1. MPU 嵌入式微处理器: ARM、MIPS、PowerPC、x86、SH

1. 优化 RTOS 支持
2. 存储区保护功能
3. 处理器结构可扩展
4. 低功耗

2. MCU

1. 存储器: Cache、主存、辅助存储器
2. 通用设备接口和 I/O 接口

3. DSP: Digital Signal Processor

2. 中间层, 硬件抽象层 (HAL), 板级支持包 (BSP), 使得底层驱动程序与硬件无关

1. 硬件相关性, (硬件接口)
2. 操作系统相关性, (软件接口)

3. 嵌入式系统初始化

1. 片级初始化
2. 板级初始化 (对应 BIOS)
3. 系统初始化 (对应 kernal)
4. 软件层 (内核)
 1. OS, E(embedded)OS
 1. (硬, 软)RTOS
 1. 进程管理
 2. 进程间通信与同步
 3. 内存管理
 4. I/O 资源管理
 5. 评价指标: 中断响应事件, 临界情况 (Worst-case)(系统调用) 执行时间
 2. 分时操作系统
 2. fs
 3. GUI
 4. 网络系统、通用组建模块
5. 应用

1.2 嵌入式系统的基本知识

1.2.1 MPU

1. 控制单元: 负责取指、译码和取操作数等基本动作, 控制单元中两个重要寄存器: 程序计数器 (PC) 和指令寄存器 (IR)
2. 算术逻辑单元: 分为算术运算单元和逻辑运算单元两部分
3. 寄存器: 存储暂时性的数据

1.2.1.1 结构

冯·诺依曼, 哈佛

1.2.1.2 CISC 和 RISC

- 复杂指令集计算机 (CISC)

- 精简指令集计算机 (RISC)

类别	CISC	RISC
指令系统	指令系统指令数量很多	较少，通常少于 100
执行时间	有些指令执行时间很长，如整块的存储器内容复制；或将多个寄存器的内容复制到存储器	没有较长执行时间的指令
编码长度	编码长度编码长度可变，1 15 字节	编码长度固定，通常为 4 个字节
寻址方式	寻址方式多样	简单寻址
操作	可以对存储器和寄存器进行算术和逻辑操作	只能对寄存器进行算术和逻辑操作，Load Store 体系结构
编译	难以用优化编译器生成高效的目标代码程序	采用优化编译技术，生成高效的目标代码程序

1.2.1.3 信息存储的字节顺序

0x4321

- 小端：低字节, 低内存 (1234)
- 大端：高字节, 低内存 (4321)

选择大端还是小端存储法并不存在技术原因，只涉及处理器设计厂商的习惯

1.2.1.4 嵌入式微处理器的分类

- ARM
- MIPS
- Power PC
- x86
- 68K/Coldfire

1.2.2 嵌入式软件

- 规模较小
- 开发难度大
- 实时性和可靠性要求高
- 要求固化存储

1.2.2.1 嵌入式软件体系结构

1. 无操作系统

1. 循环轮转方式: `for: a(); b(); c();`
2. 前后台系统: 循环轮转 + 中断处理, `mainloop(); mainloop.emit(...);`

2. 有操作系统

1. 提高了系统的可靠性
2. 提高了系统的开发效率, 降低了开发成本, 缩短了开发周期
3. 有利于系统的扩展和移植

1.2.2.2 嵌入式操作系统的分类

1. 按系统的类型分类

1. 商用系统: VxWorkd、Windows CE、PalmOS
2. 专用系统
3. 开源系统: μ C/OS、嵌入式 Linux 系统

2. 按响应时间分类

1. 硬实时系统: 系统对响应时间有严格的要求, 如果响应时间不能满足, 这是绝对不允许的, 可能会造成系统崩溃或其他错误
2. 软实时系统: 系统对响应时间有要求, 如果响应时间不能满足, 将带来额外的代价, 但是这种代价通常是能够接受的

3. 按软件结构分类

1. 单体结构: 嵌入式 Linux 和 PDOS
2. 分层结构: MS-DOS 结构
3. 微内核结构: OS-9、VxWorks、CMX-RTX、Nucleus Plus

1.2.2.3 嵌入式操作系统的重要概念

1. 占先式内核
2. 调度策略分析
3. 任务优先级分配
4. 时间的可确定性
5. 任务切换时间
6. 中断响应时间 (可屏蔽中断)

7. 优先级反转
8. 任务执行时间的抖动
9. 任务划分

1.2.2.4 常见实时嵌入式操作系统

1. 商用型实时嵌入式操作系统

- VxWorks
- Windows Embedded
- pSOS
- PalmOS
- OS-9
- LynxOS
- QNX

2. 免费型实时嵌入式操作系统

- 嵌入式 Linux
- μ C/OS

1.2.2.5 嵌入式操作系统的选型原则

1. 操作系统的硬件支持. bus, IO, A/D, uart, i2c, spi, usb, 外设接口, eth, audio (iis?).
2. 开发工具支持程度
3. 能否满足应用需求

1.2.2.6 设计方法

1. 规格说明书
2. 体系结构设计
3. 执行装置, 硬件, 软件设计
4. 系统集成
5. 测试
6. 发布 (产品)

1.3 ARM 嵌入式微处理器

消费市场主流 32 位嵌入式微处理器

评价指标 功耗, 代码存储密度, 集成度, 多媒体加速

32 位, RISC. ARM7 冯·诺依曼, ARM9 以上哈佛

ARM 体数据类: 字 (4), 半字 (2), 字节

1.3.1 工作状态

- ARM 状态: 此时处理器执行 32 位的字对齐的 ARM 指令
- Thumb 状态: 此时处理器执行 16 位的、半字对齐的 Thumb 指令, (提高代码密度) (ARM 指令压缩形式的子集)

1.3.2 流水线

1. 取指: 将指令从内存中取出来
2. 译码: 操作码和操作数被译码以决定执行何种操作
3. 执行: 执行已译码的指令
4. 数据缓存
5. 写回

减少周期工作量, 使用更高时钟频率

1.3.3 寄存器

1. (31) 通用寄存器
 1. (8) 未分组寄存器 R0-R7
 2. (22) 分组寄存器
 1. (10) R8-R12 * {fiq, usr}
 2. (6) R13_mode 堆栈指针
 3. (6) R14_mode 子程序链接寄存器
 3. (1) 程序计数器 PC (R15)
2. (6) 状态寄存器

1. (1) 程序状态寄存器 CPSR
 1. 保存 ALU 中的当前操作信息
 2. 控制允许和禁止中断
 3. 设置处理器的运行模式
2. (5) 备份程序状态寄存器 SPSR

1.3.3.1 程序状态寄存器

- 高 5 位
 - Negative: 补码运算时, N=1 表示负数, N=0 表示正数
 - Zero: Z=1 表示运算结果为 0, Z=0 表示运算结果非 0
 - Carry: 加减法或移位运算的进位或溢出位
 - Overflow: 补码运算时的符号位溢出
 - Q: (ARMv5+) 增强 DFP 运算是否发生溢出
- 低 8 位
 - Irq disable
 - Fiq disable
 - State
 - Mode4-M0

1.3.4 7 modes

1. USR & SYS
 1. 用户模式 (USR): 正常程序执行状态
 2. 系统模式 (SYS): 运行具有特权的操作系统任务
2. 快速中断模式 (FIQ): 用于高速数据传输或通道管理
3. 外部中断模式 (IRQ): 用于通用的中断处理
4. 管理模式 (SVC): 操作系统使用的保护模式
5. 数据访问终止模式 (ABT): 当数据或指令预取终止时进入该模式, 保护存储
6. 未定义指令终止模式 (UND): 当未定义指令执行时进入该模式

1.3.5 异常

- 复位

- 未定义指令
- 非法访问
 - 指令预取中止 (指令不存在, 执行时发生)
 - 数据中止 (地址不存在, 非法)
- 中断
 - 软件中断
 - IRQ
 - FIQ

1.3.6 嵌入式系统初始化过程

Boot Loader 一般存放在 ROM、EEPROM、Flash 中

- 启动模式
 1. 初始化硬件配置
 1. 异常中断初始化
 2. 相关硬件设备初始化
 3. 拷贝第二阶段代码拷贝到内存中
 4. 跳到第二阶段的内存处
 2. 加载操作系统
 1. BSS 段清零
 2. 设置各种模式下的堆栈
 3. 开 IRQ 中断
 4. 跳转到 C 代码的 MAIN 入口
- 下载模式

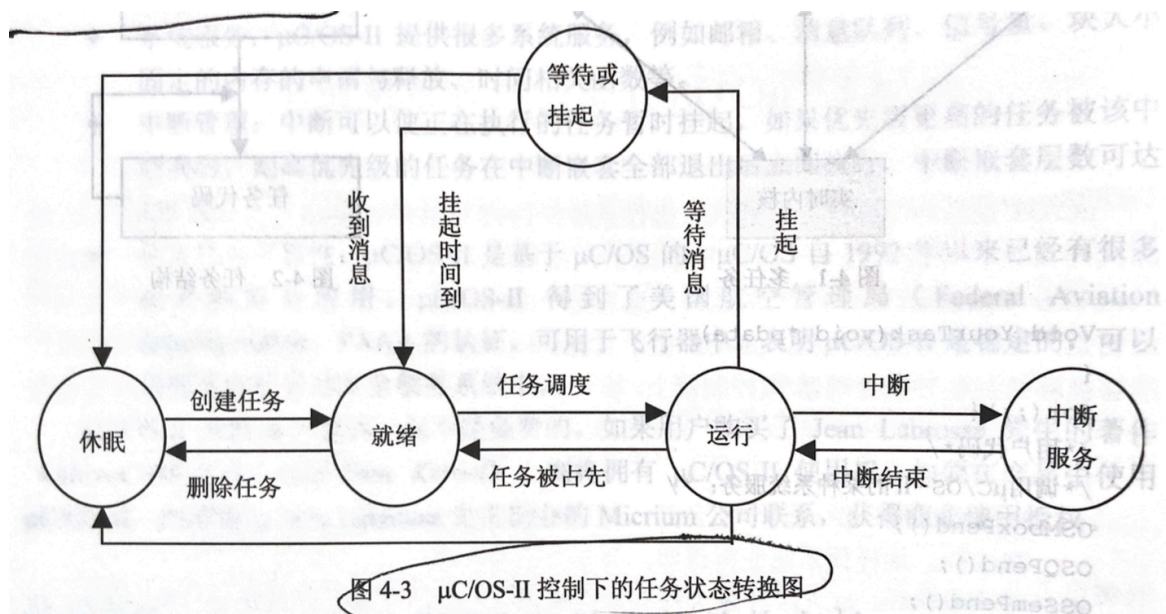
1.4 第四章 $\mu\text{C}/\text{OS-II}$ 嵌入式实时操作系统内核分析

1.4.1 $\mu\text{C}/\text{OS-II}$ 主要特点

- 发行方式
 - 开源
- 项目设计

- 可移植
- 可固化 (ROMable)
- 可裁剪
- Feature
 - 多任务
 - * 抢占式内核
 - * 多任务
 - * 任务栈
 - 中断管理
 - Real-time
 - * (系统调用时间) 可确定性
 - 系统服务
- 稳定性和可靠性

1.4.2 $\mu\text{C}/\text{OS-II}$ 多任务



1.4.2.1 中断处理

1. 有中断源发生中断
2. 系统允许中断

1.4.2.2 任务关系

- 相互独立
- 任务互斥
- 任务同步
- 任务通信

1.4.2.3 任务互斥

1. 关闭中断
2. 忙等待
3. 信号量

死锁

1. 条件
 1. 互斥
 2. 请求和保持, 可同时占用和请求
 3. 资源不可抢占
 4. 环路
2. 避免
 1. ...
 2. 先全部请求
 3. ...
 4. 同样顺序请求, 释放时使用相反顺序

1.4.2.4 任务间的通信

1. 共享内存
2. 消息传递

1.4.3 任务通信机制

1. 时间控制块
2. 信号量
3. 邮箱

4. 消息队列

1.4.4 移植 $\mu\text{C}/\text{OS-II}$ 基本要求

1. 处理器的 C 编译器能产生可重入代码, (没有副作用)
2. 开闭中断 Programable
3. 支持 (定时) 中断
4. 支持能够容纳一定量数据的硬件堆栈
5. 支持将堆栈指针和其他 CPU 寄存器存储、读出到堆栈的指令

1.5 第五章嵌入式系统硬件平台与接口设计

1.5.1 存储器

1. 寄存器
2. Cache
3. {S, D, DD}RAM
4. Flash, {P, EP, E2P}ROM
5. 外部存储器
6. 远程二级存储

1.5.1.1 内存管理单元

在 CPU 和物理内存之间进行地址转换

- 虚拟存储空间到物理存储空间的映射
- 存储器的访问权限的控制
- 设置虚拟存储空间的缓冲特性

1.5.2 串口

1. 单工
2. 半双工
3. 全双工

1.5.3 I/O 接口

- 解决主机 CPU 和外围设备之间的时序配合和通信联络问题
- 解决 CPU 和外围设备之间的数据格式转换和匹配问题
- 解决 CPU 的负载能力和外围设备端口选择问题

1.5.3.1 I/O 接口编址方式

- I/O 接口独立编址
- I/O 接口与存储器统一编址方式