

补充题目集

1. 请简述“看门狗”的工作原理、应用和设计思路。

原理：看门狗是一种监控系统的运行状况的手段，通过软硬件结合的方式实现对系统运行状况的监控。稳定运行的软件会在执行完特定指令后进行喂狗，若在一定周期内看门狗没有收到来自软件的喂狗信号，则认为系统故障，会进入中断处理程序或强制系统复位。

应用：看门狗广泛应用于嵌入式系统、服务器、网络设备等领域。在嵌入式系统中，看门狗可以监测系统的运行状态并自动重启系统，以确保系统的可靠性和稳定性。在服务器和网络设备中，看门狗可以监测系统的网络连接状态和服务运行状态，并在出现故障或异常时及时采取措施，以保证系统的高可用性和稳定性。

设计思路：看门狗通常采用硬件和软件相结合的方式。看门狗实际为一个计时器，当程序停止喂狗重置计时器导致计时器溢出时触发系统复位信号或者执行相应措施。

硬件看门狗一般由一个独立的计时器、一组寄存器和一个中断处理器组成，可以在系统出现故障或异常时自动重启系统。

软件看门狗则是通过在系统中运行一个后台程序来实现，可以定期检测系统状态并在出现故障或异常时采取相应的措施。

2. Linux 中驱动的加载使用主要有哪些方法？它们的差别是什么？

- 静态加载：将驱动程序的代码编译到内核中，使得驱动程序随内核启动而加载。这种方法可以提高系统的安全性和性能，但是需要重新编译内核，不太方便。
- 动态加载：将设备驱动单独编译成模块，以动态加载方式加入内核。在系统启动时，使用 `insmod` 命令将驱动程序加载到内核中。这种方法不需要重新编译内核，但是需要手动加载驱动程序，比较繁琐。

最主要的差别就在于编译方法和加载时机上，静态需要和内核一起编译，在内核启动时一起加载

而动态加载单独编译，只在用户需要时自行使用 `insmod` 加载，也可以通过 `rmmmod` 卸载。

3. 字符设备驱动的初始化和退出需要完成哪些工作？

- 初始化，完成字符设备注册：
 - 申请设备号：获取字符设备驱动的设备号，包括主设备号和次设备号。主设备号用于标识设备类型，次设备号用于标识具体的设备。
 - 创建设备文件 `device_create`：在系统中注册设备号对应的设备文件，使得用户空间可以通过文件系统接口访问设备。
 - 定义字符设备结构体 `cdev`：为字符设备驱动分配设备结构体，用于管理设备的状态和操作。
 - 初始化字符设备结构体 `cdev_init`：初始化设备结构体的各个字段，包括设备状态、设备操作函数(编写处理文件各操作的函数，构造 `file_operations` 结构，注册到对应的设备文件)等。
 - 添加字符设备 `cdev_add`：将字符设备添加到内核的字符设备管理列表中，使得内核可以调用设备驱动中的函数。

- 申请和初始化其它资源
- 退出，完成字符设备注销：
 - 删除设备文件 device_destroy
 - 移除字符设备 cdev_del
 - 注销设备号 unregister_chrdev_region
 - 释放使用的其它资源。

4. 请简述 RTOS 和 OS 两者之间的区别。

OS 专注于计算吞吐量，而 RTOS 专注于快速的响应时间。

RTOS 具有高度的可预测性和可靠性，可以确保应用程序在严格的时间限制下运行，而 OS 更注重通用性和功能性。

RTOS 适合用于实时应用程序的开发，例如工控、嵌入式、自动化等领域；而 OS 适合用于个人电脑、服务器、移动设备等领域。

5. 请简述 Bootloader 两阶段的启动过程。

阶段1：实现依赖于CPU体系结构的代码

- 硬件设备初始化
 - 屏蔽所有的中断
 - 设置CPU的速度和时钟频率
 - RAM初始化
 - 初始化LED
 - 关闭CPU内部指令 / 数据Cache
- 为加载阶段2准备RAM空间
 - 除了阶段2可执行映像的大小外，还必须把堆栈空间也考虑进来
 - 必须确保所安排的地址范围的确是可读写的RAM空间
 - 内存区域有效性检测方法
 - 保存指定内存区域
 - 写入预定数据
 - 读入数据并比较
 - 恢复内存数据
- 拷贝阶段2代码到RAM中
- 设置堆栈指针sp
- 跳转到阶段2的C语言入口点

阶段2：实现一些复杂的功能

- 初始化本阶段要使用到的硬件设备
 - 初始化至少一个串口，以便和终端用户进行I/O输出信息
 - 初始化计时器等
- 检测系统的内存映射
- 加载内核映像和根文件系统映像
- 设置内核的启动参数

- 调用内核

6. ARM 处理器的工作模式有几种？请简述每个工作模式的特点，并指出哪个（些）模式不属于 ARM 特权模式。

1. 用户模式（USR）：正常程序执行模式，不能直接切换到其他模式
2. 系统模式（SYS）：运行操作系统的特权任务，与用户模式类似，但具有可以直接切换到其他模式等特权
3. 快中断模式（FIQ）：支持高速数据传输及通道处理，FIQ异常响应时进入此模式
4. 中断模式（IRQ）：用于通用中断处理，IRQ异常响应时进入此模式
5. 管理模式（SVC）：操作系统保护模式，系统复位和软件中断响应时进入此模式（由系统调用执行软中断SWI命令触发）
6. 中止模式（ABT）：用于支持虚拟内存和/或存储器保护，在ARM7TDMI没有大用处
7. 未定义模式（UND）：支持硬件协处理器的软件仿真，未定义指令异常响应时进入此模式

只有用户模式不属于ARM特权模式,其余六种都是特权模式.

7. 请简述 RISC 指令集和 CISC 指令集的区别和他们各自在不同方面的特点。

RISC全称Reduced Instruction Set Compute，精简指令集。

- 选取使用频率较高的一些简单指令以及一些很有用但不复杂的指令，让复杂指令的功能由使用频率高的简单指令的组合来实现。
- 指令长度固定,指令格式种类少,寻址方式少
- 有较多的通用寄存器,大量的计算在寄存器中完成,不需要访问存储器
- 有专门访问存储器的指令
- 高度使用流水线技术,大部分指令可在一个时钟周期内完成
- 采用优化的编译程序

CISC全称Complex Instruction Set Computers，复杂指令集

- 复杂的指令系统，各种常用或少见的功能都有对应的处理指令
- 指令的长度不固定，指令格式和寻址方式多种多样
- 可以访存的指令不受限制
- 各种指令的执行时间差异较大，复杂指令需要多个时钟周期完成
- 难以优化编译生成高效目标代码

RISC	CISC
简单指令系统	复杂指令系统
指令数目，格式少，指令长度统一	指令数目，格式多，指令长度不统一
寻址方式少	寻址方式多
只有LOAD/STORE 指令可以访存	指令访存不受限制

RISC	CISC
指令执行周期相近，短	指令执行时间差异大，长
容易实现优化编译	难以实现优化编译
通用寄存器数量多，计算速度快	通用寄存器少，需要多次访存，计算速度慢

8. 名词解释：GPIO、UART、USART、HAL、DMA、ISR、PWM、MCU、JTAG、SWI、ISP、IAP、SPI。

- GPIO：General Purpose Input Output，通用数字输入输出端口，可以由CPU直接操纵的引脚，可以写，可以读，可以接通信单元。
- UART：Universal Asynchronous Receiver and Transmitter，通用异步收发器，是一种串行通信协议，用于在设备之间传输数据。
- USART：Universal Synchronous Asynchronous Receiver and Transmitter，通用同步/异步收发器，串行通信协议，可以完成同步通信或异步通信。
- HAL：Hardware Abstraction Layer，硬件抽象层，一种软件，对于底层硬件的抽象，对上层提供操作硬件的接口。
- DMA：Direct Memory Access，直接内存访问，提供在外设和存储器之间或者存储器和存储器之间的高速数据传输，无需CPU直接控制传输。
- ISR：Interrupt Service Routine，中断服务程序。用于处理系统触发的中断。
- PWM：Pulse Width Modulation，脉冲宽度调制，利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术。
- MCU：Microcontroller Unit，微控制器，将CPU以及片上外设（内存，串口等）集合在一块芯片上。
- JTAG：Joint Test Action Group，联合测试工作组，一种用于测试和调试嵌入式设备的接口标准
- SWI：Software Interrupt，软件中断。在软件中模拟中断的方法，例如ARM的软中断指令，从用户模式变换到系统模式。
- ISP：In-System Programming，在系统编程，将程序下载到MCU的技术。
- IAP：In-Application Programming，在应用编程，在运行时对MCU进行编程的技术。
- SPI：Serial Peripheral Interface，串行外设接口，一种同步串行通信协议，用于在设备之间传输数据。