嵌入式简答题或填空

嵌入式系统定义

嵌入式系统是"控制、监视或者辅助设备、机器和车间运行的装置"。——IEEEE的定义

嵌入式系统是嵌入到计算机体系中的、用于执行独立功能的**专用计算机系统**

嵌入式系统三要素:

1、嵌入性： 满足环境要求

2、专用性： 满足配置要求

3、计算机系统： 满足控制要求

嵌入式系统与桌面通用系统的区别:

1、嵌入式系统中运行的任务是专用而确定的，桌面通用系统需要支持**大量的、需求多样**的应用程序

2、嵌入式系统往往对**实时性**提出较高要求

3、嵌入式系统对**可靠性**要求高

4、嵌入式系统有功耗约束

5、嵌入式系统内核小，可用资源少

6、嵌入式系统开发需要**专用工具**和**特殊方法**

> 以**应用**为中心，以**计算机、通信、控制**等技术为基础，采用可剪裁软硬件，适用于对功能、可靠性、成本、体积、功耗等有严格要求的**专用计算机系统**

嵌入式中的核心技术：3C

计算机体系结构:冯诺依曼结构、哈佛结构

类别	CISC	RISC
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拷贝；或将多个寄存器的内容拷贝到存贮器	没有较长执行时间的指令
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

ARM CPU + 外部设备 = ARM芯片

嵌入式系统结构

嵌入式系统一般由**嵌入式处理器**、**外围硬件设备**、**嵌入式操作系统**（可选），以及**用户的应用软件系统**组成

三层含义：

① 一个公司的名字:英国知识产权核（IP）设计公司

② 一类微处理器的通称

③ 一种技术的名字（ARM微处理器核）

R14两种功能:1、每种模式下可用于保存子程序的返回地址 2、发生异常时用于保存异常处理后的返回地址

虚拟存储管理方法:

1、分段式存储管理

2、分页式存储管理

3、段页式存储管理

存储器管理单元MMU:

存储器管理单元MMU

- 将虚拟地址转换成物理地址——将主存地址从虚拟存储空间映射到物理存储空间。
- 存储器访问权限控制。
- 设置虚拟存储空间的缓冲特性等。

注意：MMU无效时，虚拟地址将直接输出到物理地址总线。

虚拟存储空间到物理存储空间是以**存储块**为单位进行的

处理器模式	用　途
USR（用户模式）	ARM处理器正常的程序执行状态
FIQ（快速中断模式）	用于高速数据传输或通道处理
IRQ（外部中断模式）	用户通用的中断处理
SVC（管理模式）	操作系统使用的保护模式，处理软件中断
ABT（数据访问中止模式）	当数据或指令预取中止时进入该模式，可用于虚拟存储器和存储器保护
UND（未定义指令模式）	当未定义的指令执行时进入该模式，可用于支持硬件协处理器的软件仿真
SYS（系统模式）	运行具有特权的操作系统任务

ARM支持的存储块类型:

ARM920T支持的存储块类型:

- 段描述符
  - 对应段， 大小为【1MB存储块】
- 大页描述符
  - 对应大页， 大小为【64KB存储块】
- 小页描述符
  - 对应小页， 大小为【4KB存储块】
- 极小页描述符
  - 对应极小页， 大小为【1KB存储块】

CP15——系统控制协处理器

实现使能MMU的汇编指令

```
MRC   P15,0,R0,C1,C0,0      ;C1的内容赋给R0
ORR   R0,#0x1
MCR   P15,0,R0,C1,C0,0      ;R0的内容赋给C1
```

➢ 虚拟地址=物理地址——平板地址映射(Flat Address Mapping)模式。

MMU对应4种存储访问失效:

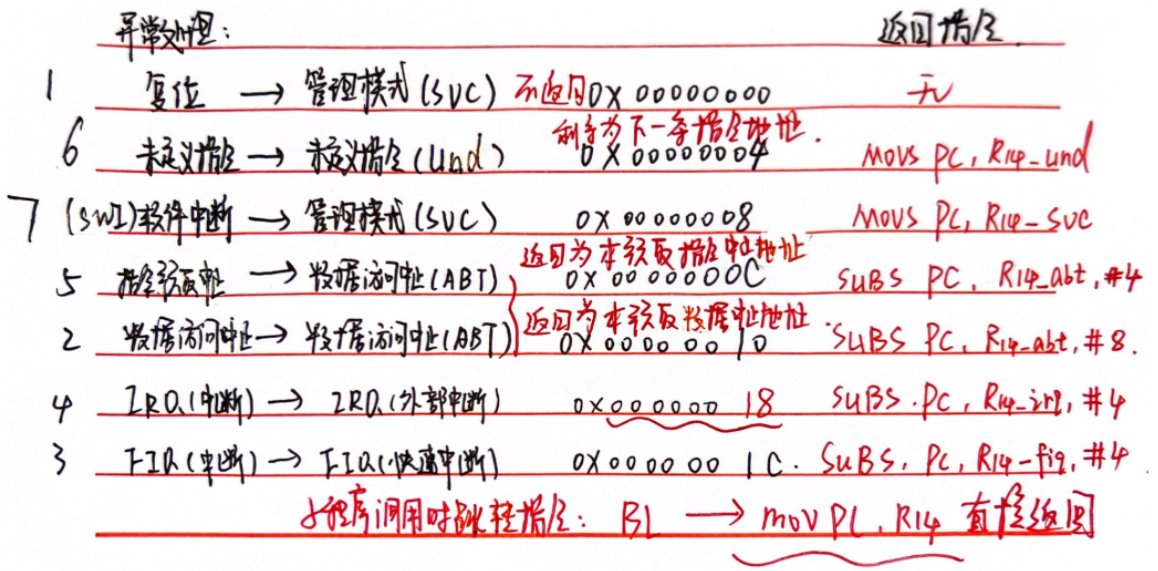
- 1、地址对齐失效
- 2、地址变换失效
- 3、域控制失效
- 4、访问权限控制失效

- **发生存储访问失效时，存储系统可以中止3种存储访问:**
  - Cache内容预取
  - 非缓冲的存储器访问
  - 页表访问

存储访问中止异常:

1、数据访问周期:数据访问中止异常

2、指令预取周期:指令预取中止异常



异常处理的过程

- 当异常发生时(例如处理一个外部的中断请求)，系统执行完当前指令后，需要保存处理器的当前状态(执行现场)，然后将跳转到相应的异常处理程序处执行。
- 当异常处理程序执行完成后，程序应恢复所保存的执行现场，从而返回到发生异常时的下一条指令继续执行。
- 处理器允许多个异常同时发生，它们将会按固定的优先级进行处理。

ARM处理器处理异常的具体步骤

NO.1
- 将下一条指令的地址存入相应链接寄存器LR，以便程序在处理异常返回时能从正确的位置重新开始执行。

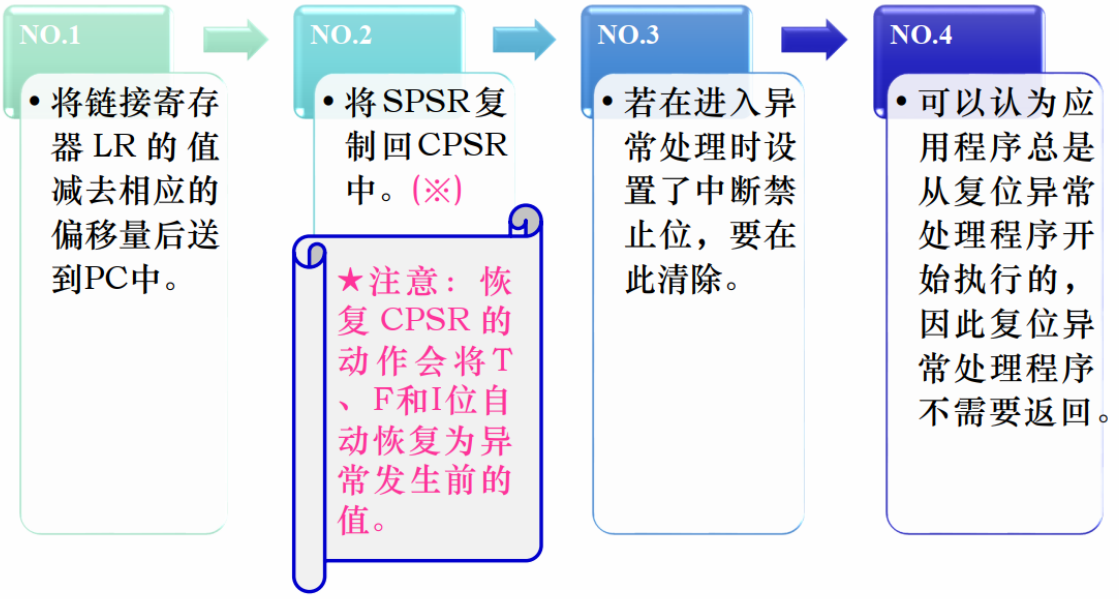
NO.2
- 将CPSR复制到相应的SPSR中。

NO.3
- 根据异常类型，强制设置CPSR的运行模式位。

NO.4
- 强制PC从相关的异常向量地址取下一条指令执行，从而跳转到相应的异常处理程序处。同时设置中断禁止位。

★注意：若当前处理器处于Thumb状态，则当异常向量地址加载入PC时，处理器会自动切换到ARM状态。

ARM处理器异常返回的具体步骤



指令:计算机控制各功能部件协调动作的命令

指令系统:微处理器所能执行的全部指令的集合

机器语言指令:能被微处理器直接识别和执行的二进制编码，是指令在计算机内部的表示形式

汇编语言指令:机器指令的符号化表示形式

使用专门的跳转指令。——可以实现向前或向后32MB的地址空间的跳转

直接向程序计数器PC写入跳转地址值。——可以实现在4GB的地址空间的跳转

IA 传送后地址加 IB传送前地址增 DA传送后地址减少 DB 传送前地址减少

-

➢在LDM指令且寄存器列表中包含有PC时，除了正常的多寄存器传送外，还将SPSR拷贝到CPSR中，用于异常处理返回。

➢若寄存器列表不包含PC，则加载/存储的是用户模式的寄存器，而不是当前(异常)模式的寄存器。

微处理器芯片与外部存储器芯片的连接:**本质上是三种总线的连接**

Nor Flash 的特点:

Nor Flash存储器性能特点

- 读取速度快，具有芯片内执行XIP特性。
- 写入速度慢，单位体积下容量小，价格高。
- 擦写次数约10万次。
- 带有SRAM接口，与微处理器连接方便，便于数据存取。
- 适用于存储固化的系统启动引导代码、操作系统代码、应用程序代码。
- 通常配置到Bank0。当系统上电或复位后从其内获取指令并开始执行。

NAND Flash： **以页为最小单位进行读写，以块为最小单位进行擦除**

支持自启动引导

存储大量用户数据和程序代码

- 擦除和写入速度很快。单位体积下数据存储密度大，价格相对便宜。
- 使用时需要复杂的I/O接口电路（专用控制器）和存储管理操作。
  - 以页(Page)为最小单位进行读写；以块(Block）为最小单位进行擦除。
- 擦写次数约100万次。
- 适用于存储大量的用户数据、程序代码。
- 支持自动启动引导。

NAND Flash 相关引脚:

1、NCON：选择位(普通或高级)

2、GPG13：页容量选择

3、GPG14：地址周期选择

4、GPG15：总线宽度选择

NAND Flash 工作模式:

1、自动引导模式

2、普通闪存模式

存储控制寄存器:

1、总线宽度和等待控制寄存器BWSCON

2、存储块控制寄存器BANKCONn

3、刷新控制寄存器REFRESH

4、存储块大小控制寄存器BANKSIZE

NAND Flash 相关寄存器

1、**配置寄存器NFCONF**

2、**控制寄存器NFCONT**

3、**命令寄存器NFCMMD**

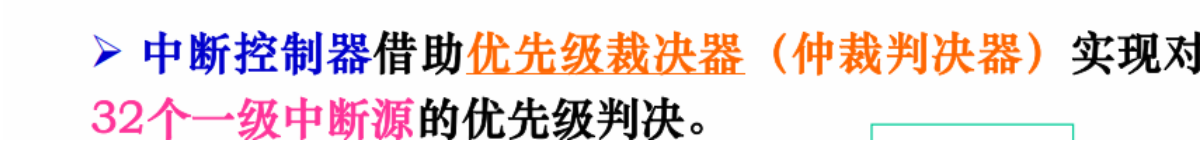
4、**地址寄存器NFADDR**

5、**数据寄存器NFDATA**

6、**状态寄存器NFSTAT**

通用输入输出端口(GPIO)

用于连接各类输入输出设备，实现其与微处理之间的数据传输。



中断作用:

1、并行处理 2、实时处理 3、故障处理

中断控制器的功能:

1、外部中断请求信号管理

2、中断模式设定

3、中断请求信号标记

4、中断屏蔽设定

5、中断优先级管理

6、中断服务标记

中断优先级的类型:

1、静态优先级 - 优先级固定

2、动态优先级 - 优先级循环

定时器0包含用于大电流驱动的死区发生器

看门狗定时器:用于当噪声或系统错误引起故障时，恢复微处理器操作的定时器

时钟部件:时钟控制模块

定时部件:

1、定时器

2、实时时钟

3、看门狗定时器

定时器功能:

1、定时

2、计数

3、脉宽调制(PWM)

看门狗定时器

以S3C2440为例

- WDT(WatchDog Timer)是一种用于当噪音或系统错误引起故障时，恢复微处理器操作的定时器。
- 本质上是一个16位内部定时器。当计数器值为0(即超时)时，通过触发中断服务，激活128个PCLK时钟周期的内部复位信号。
- 可用作一个普通的带中断请求(INT_WDT)的16位定时器。
- 只能使用PCLK时钟信号作为源输入时钟信号，且没有对外的输出信号引脚。
- 系统处于嵌入式ICE调试模式时，WDT必须无操作(自动关闭)。

Bootloader：(引导装载程序) 是操作系统内核运行之前运行的第一段程序

用途:为最终调用操作系统内核准备好正确的软硬件环境

Bootloader运行模式;

1、启动加载模式

2、下载模式

---

stage1通常包括以下步骤(按执行的先后顺序)：

- 硬件设备初始化。
- 为加载Bootloader的stage2准备RAM空间。
- 拷贝Bootloader的stage2到RAM空间中。
- 设置堆栈。
- 跳转到stage2的C程序入口点。

stage2通常包括以下步骤(按执行的先后顺序)：

- 初始化本阶段要使用到的硬件设备。
- 检测系统内存映射(memory map)。
- 将内核(kernel)映像和根文件系统映像从Flash存储器读到RAM空间中。
- 为内核设置启动参数。
- 调用内核。

常用的Bootloader：

内核:运行程序和管理硬件设备的核心，用于管理内存、CPU和其它相关组件

内核的核心行为:系统调用控制

典型Linux发行版包括:Linux内核、GNU程序库和工具 、命令行shell、图形界面和办公软件、应用软件

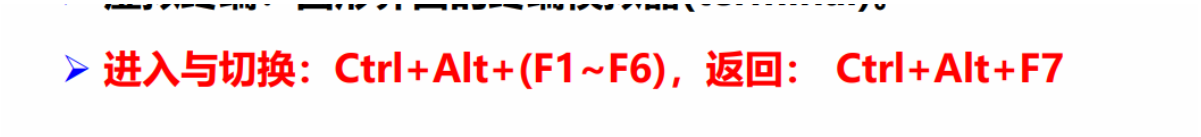
五个主要子系统:进程调度、进程间通信、内存管理、虚拟文件系统、网络接口

内核烧写方法:

文件管理系统:Linux操作系统中负责管理和存储文件信息的组件

Windows文件系统只负责文件存储，Linux文件系统管理所有软硬件资源

Linux中三种基本文件类型:**普通文件、目录文件、设备文件**



- 

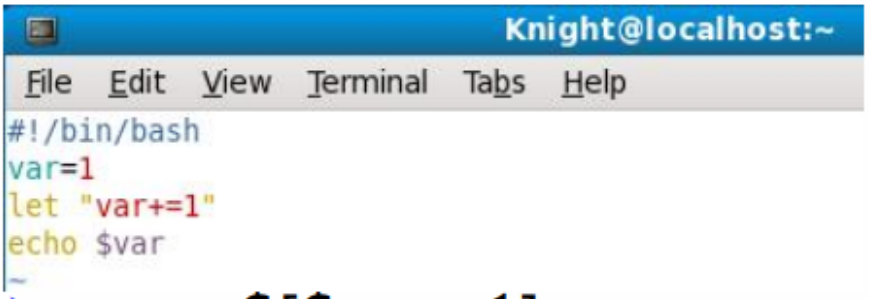
对文件或目录进行访问的三种用户类型:

文件所有者，与所有者同组的用户，其它用户

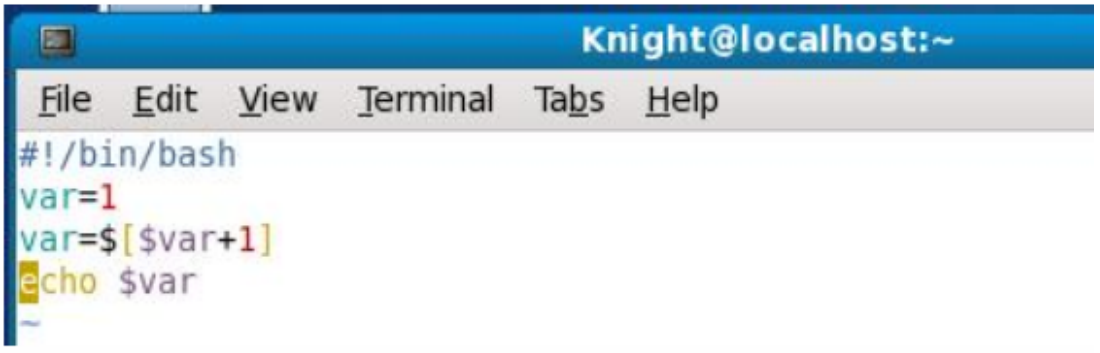
Shell程序的一般结构:Shell类型、函数、主过程

- 一般步骤
  - ✓编辑文件
  - ✓保存文件
  - ✓将文件赋予可以执行的权限
  - ✓运行及排错

- let "var+=1 "



- var=$[$var+1]



- var=`expr $var + 1` #注意加号

变量引用时:等号两边不能有空格 字符串中有空格必须加引号

Linux文件的开发流程:使用gcc编译代码、生成预处理文件、生成汇编文件、生成目标文件、生成可执行文件