目录

[1 总论 2](#_Toc309404217)

[1.1 软件 2](#_Toc309404218)

[1.2 硬件 2](#_Toc309404219)

[2 嵌入式系统 2](#_Toc309404220)

[2.1 典型结构 2](#_Toc309404221)

[2.2 软硬件框架 2](#_Toc309404222)

[2.3 系统初始化 3](#_Toc309404223)

[3 嵌入式开发 3](#_Toc309404224)

[3.1 基本流程 3](#_Toc309404225)

[3.2 编译和加载过程 4](#_Toc309404226)

[3.3 软件开发流程 4](#_Toc309404227)

[4 考试形式 5](#_Toc309404228)

[4.1 闭卷，笔试 5](#_Toc309404229)

[4.2 范围 5](#_Toc309404230)

[4.3 题型： 5](#_Toc309404231)

[5 去年的参考题 6](#_Toc309404232)

[5.1 嵌入式硬件系统基础。 6](#_Toc309404233)

[5.2 ARM 7](#_Toc309404234)

[5.3 嵌入式软件系统 8](#_Toc309404235)

[5.4 实时系统： 9](#_Toc309404236)

[5.5 嵌入式操作系统的体系结构， 9](#_Toc309404237)

[5.6 板级支持包BSP与bootloader。 9](#_Toc309404238)

[5.7 嵌入式开发。 10](#_Toc309404239)

[5.8 ucOS 10](#_Toc309404240)

[6 复习 10](#_Toc309404241)

[6.1 嵌入式系统的定义，特点。嵌入式系统发展与现状。分类。应用。 11](#_Toc309404242)

[6.2 嵌入式系统的基本组成。 11](#_Toc309404243)

[6.3 嵌入式硬件系统基础。 11](#_Toc309404244)

[6.4 嵌入式系统软件知识 13](#_Toc309404245)

[6.5 嵌入式系统开发过程 16](#_Toc309404246)

[6.6 ucOS-II 17](#_Toc309404247)

# 总论

## 软件

操作系统

应用程序

## 硬件

处理器

IO接口

存储器

外设

# 嵌入式系统

## 典型结构

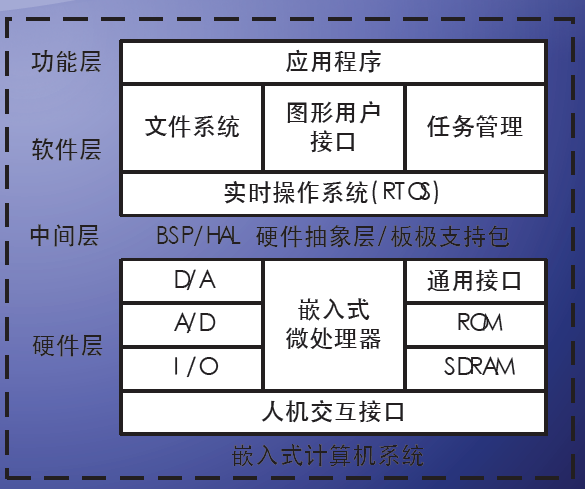
感知器

控制器

存储器

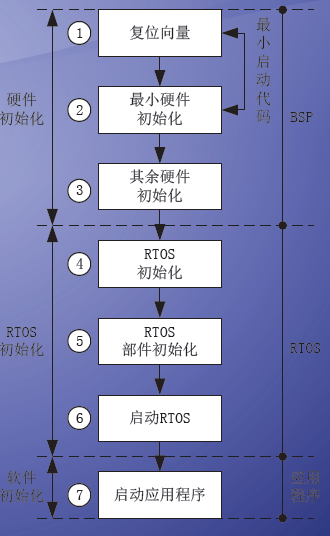
执行器

## 软硬件框架





## 系统初始化

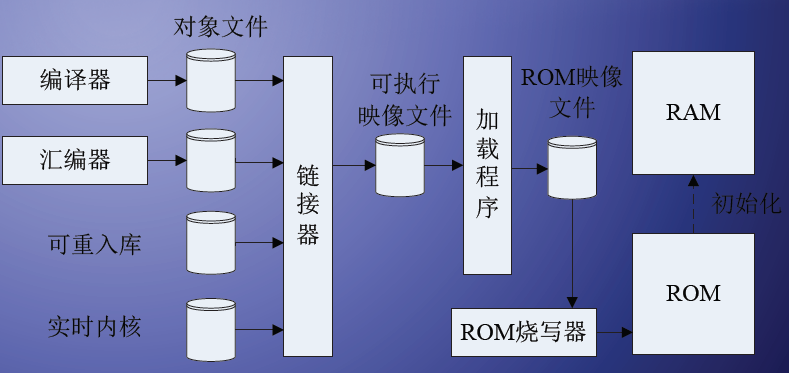


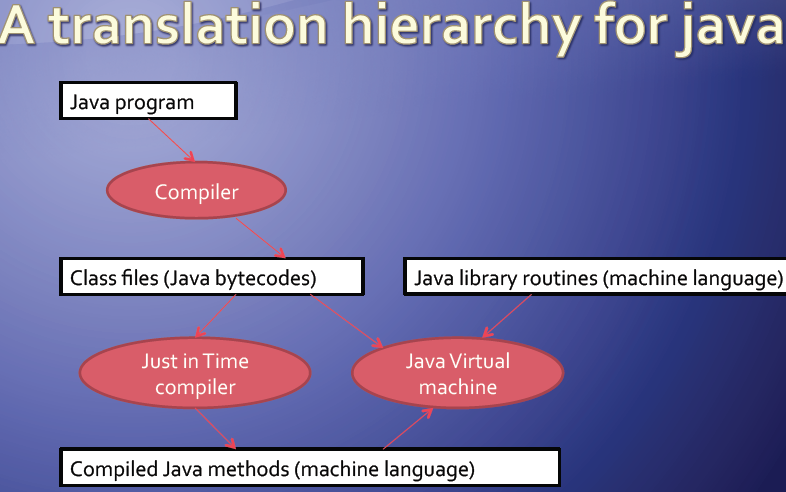
# 嵌入式开发

## 基本流程



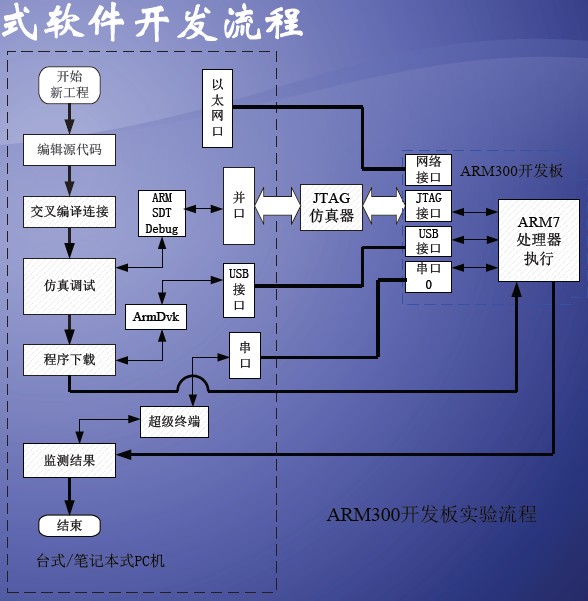
## 编译和加载过程





Java嵌入效能提升JAT；AOT；DAC；硬件JVM——了解

## 软件开发流程



实验。交叉开发。

# 考试形式

## 闭卷，笔试

基本概念、基本原理、设计应用技术

## 范围

以ppt涵盖内容为主。

## 题型：

选择题 10个

简答题

问答题（含有UCOSII的编程问题）。

# 去年的参考题

## 嵌入式硬件系统基础。

* 1. 嵌入式微处理器体系结构： 诺伊曼体系结构和哈佛体系结构。
     + 冯诺依曼体系结构数据和程序放在同一个存储单元，统一编址，指令和数据通过同一个总线访问。
     + 哈佛体系结构的程序和数据不是放在同一个存储空间中，因此有两条总线，也就是说数据吞吐率是冯诺依曼结构的两倍。
  2. RISC和CISC。基本架构。

|  |  |  |
| --- | --- | --- |
|  | CISC | RISC |
| 价格 | 由硬件完成部分软件功能，硬件复杂性增加，芯片成本高 | 有软件完成部分硬件功能，软件复杂性增加，芯片成本低 |
| 性能 | 减少代码尺寸，增加指令的执行周期数 | 使用流水线降低指令的执行周期数，增加代码尺寸 |
| 指令集 | 大量的混杂指令集，有简单快速的指令，也有复杂多周期指令，符合HLL | 简单的但周期指令，在汇编指令方面有相应的CISC微代码指令 |
| 高级语言支持 | 硬件完成 | 软件完成 |
| 寻址模式 | 复杂的寻址模式，支持内存到内存寻址 | 简单的寻址模式，仅允许LOAD和STORE指令存取内存，其它所有的操作都基于寄存器到寄存器 |
| 寄存器数目 | 寄存器较少 | 寄存器较多 |

* 1. 流水线技术。
     + RISC机器用来减少指令周期的一种技术，提高处理器和总线的使用率；
  2. 嵌入式系统中的微处理器、总线、存储设备（种类，优缺点）。
     + 微处理器有不同操作位数的，也有不同架构的；
     + 总线有片内总线和片外总线（ＰＣＩ啊什么的）；
     + 存储设备有Ｆｌａｓｈ，ＤＲＡＭ等等

## ARM

体系结构；ARM汇编语言。重点、难点：CPSR、寻址模式、处理器模式（7种）、处理器状态（2种）； CPSR- current program status register

* + - 很多的指令后缀S就会改变这个寄存器的值

1. 寻址模式：
   * + 立即寻址
     + 寄存器寻址
     + 寄存器间接寻址
     + 基址寻址
     + 堆栈寻址
     + 快拷贝寻址
     + 相对寻址
2. 处理器模式
   * + 用户模式（User）
     + 快速中断FIQ
     + 外部中断IRQ
     + 管理模式（Super Visor）
     + 系统模式（System）
     + 未定义（Undifined）
3. 处理器状态
   * + ARM
     + Thumb

## 嵌入式软件系统

1. 体系结构。
   * + 驱动层
     + 操作系统层
     + 中间层
     + 应用层
2. 嵌入式软件运行流程。
   * + 上电复位
     + 板级初始化
     + 引导操作系统
     + 操作系统初始化
     + 应用程序初始化
     + 多任务应用

## 实时系统：

* 1. 定义：能够在限定响应时间内提供所需水平服务的系统
  2. 分类：强实时性，毫秒微秒级别；一般实时性，秒级别；弱实时性，十秒或者更多
  3. 实时系统的调度
     + 抢占式调度
     + 非抢占式调度

## 嵌入式操作系统的体系结构，

* 1. 嵌入式内核的功能：
     + 任务调度；
     + 内存管理；
     + 通信，互斥和同步机制
     + 中断管理：包括发出中断，保存现场，跳转至中断处理，恢复现场；
  2. 常用的嵌入式操作系统。
     + uCOS-2
     + 嵌入式Linux
     + VxWorks
     + Symbian
     + Palm OS

## 板级支持包BSP与bootloader。

* 1. BSP的概念：板级支持包，类似于BIOS一样的东西
  2. 特点与功能：
     + 硬件相连性
       1. BSP必须为操作系统提供操作和控制具体硬件的方法；
     + 操作系统相连性
       1. BSP必须为不同的操作系统提供接口；
  3. bootloader的概念：bootloader是一段小程序，将操作系统从外部存储拷贝到内存中，并且跳转到内核首地址
  4. 嵌入式系统引导加载过程。
     + 如果使用boolloader会消耗时间，但是可以将操作系统保存在外存中，节省空间，适合于硬件条件比较低的机器
     + 如果使用bootloader时间效率高，系统迅速启动，直接在非易失性介质中运行，速度比较慢

## 嵌入式开发。

嵌入式系统的调试技术。理解实验流程。理解交叉编译；理解下载并运行程序的不同方法。JTAG接口。

## ucOS

原理：任务调度、中断与时钟、同步与通信、存储管理

# 复习

## 嵌入式系统的定义，特点。嵌入式系统发展与现状。分类。应用。

定义：具有一定功能或者用途应藏在某种设备中的软硬件的集合，没有固定的特征；

组成：一般是由嵌入式软件和硬件组成的，硬件包括以微处理器为核心集成存储器和系统专用的输入输出设备，软件包括初始化代码及驱动，嵌入式操作系统和应用程序等；

特点：

形式多样，面向特定用途；

得到多种处理器和处理器架构支持；

嵌入式系统通常非常注重成本问题；

有实时性和可靠性的要求；

需要专门的方法和开发工具。

……

四个阶段：

第一阶段：以四位到八位单片机为核心的可编程控制器系统，同时具有检测伺服指示设备相配合的功能。

第二阶段：以八位和十六位嵌入式中央处理器为基础，以简单操作系统为核心的嵌入式系统。

第三阶段：以三十二位RISC嵌入式中央处理器加嵌入式操作系统为标志的嵌入式系统。

第四阶段：以基于INTERNET接入为表示的嵌入式系统。

分类：

按照处理器：4,8,16,32,64位嵌入式系统

按照用途：信息家电，移动终端，工业控制，汽车电子，通信类；

按照速度：强实时操作系统，毫秒或者微秒；一般，几秒；弱，十秒或者更长；

## 嵌入式系统的基本组成。

一般是有软件和硬件构成，硬件是以微处理器为核心集成存储器和系统专用IO设备；软件包括初始化代码及驱动，嵌入式操作系统和应用程序；

## 嵌入式硬件系统基础。

3.1 嵌入式微处理器基础

3.1.1嵌入式微处理器体系结构——参见[5.1](#_嵌入式硬件系统基础。)

冯诺伊曼结构与哈佛结构

CISC与RISC

流水线技术

信息存储的字节顺序（大端存储法和小端存储法）

大端以最高位作为字地址。

小端以最低位作为字地址

3.1.2嵌入式系统处理器的结构和类型

分类：

R 嵌入式微处理器种类繁多，按位数可分为4位、

8位、16位、32位和64位。

R 根据功能不同，嵌入式微处理器分为四种：

嵌入式微处理单元（MPU）

嵌入式微控制器（MCU）

嵌入式DSP处理器——新号处理

嵌入式SoC

体系结构

R 算术格式（Arithmetic Format）

由于低成本和低功耗的限制，大多数的嵌入式微处理器使用定点运算（fixed--‐pointarithmetic），即数值被表示为整数或在－1.0和＋1.0之间的分数，比数值表示为尾数和指数的浮点版本的芯片便宜。

当嵌入式系统中需要使用浮点运算时，可采用软件模拟的方式实现浮点运算，只不过这样要占用更多的处理器时间。

R 功能单元（FunctionalUnits）

通常包括不止一个的功能单元，典型的是包含一个ALU、移位器和MAC，处理器通常用一条指令完成乘法操作。

R 流水线（Pipeline）

通通常采用单周期执行指令，可能导致比较长的流水线

常采用单周期执行指令，可能导致比较长的流水线

3.1.3异常

x86读 程序地址

arm读 指令

可屏蔽中断、不可屏蔽中断

中断优先级、中断嵌套

3.2 嵌入式系统的存储体系

3.2.1存储器系统

存储器系统的层次结构

寄存器—Cache（SRAM）——Mem（DRAM）——ROM；Flash；HD

高速缓存（Cache）

内存管理单元（MMU）

3.2.2 ROM的种类与选型

ROM ；eROM；EEROM

3.2.3 Flash Memory的种类与选型

Nor Nand

3.2.4 RAM的种类与选型

DRAM SRAM  
3.2.5 外存

3.3 嵌入式系统I/O接口

3.3.1 GPIO接口基本原理与结构

3.3.2 A/D、D/A接口基本原理与结构

3.4嵌入系统通信及网络接口

## 嵌入式系统软件知识

4.1嵌入式软件基础知识

4.1.1嵌入式软件的分类

系统软件——控制，管理软件——OS，中间件

支撑软件——辅助开发软件和工具

应用软件

开发平台软件和嵌入式系统软件

4.1.2无操作系统支持的嵌入式软件体系结构

轮询——慢速简单系统

中断

前后台——后台轮询系统，前台中断处理过程模块。

4.1.3有操作系统支持的嵌入式软件体系结构

设备驱动层——操作系统层——中间件层——应用层（任务）

4.1.4板极支持包基础知识（系统初始化、设备驱动程序）

软硬件驱动。

上电复位

板级初始化

引导/升级系统 系统升级

系统初始化

应用初始化

多任务应用

4.1.5嵌入式中间件（GUI、数据库）（不考）

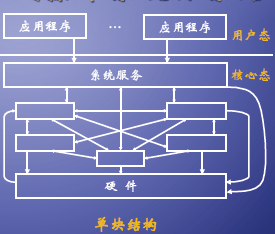
java dcom cobar 。。。

4.2 嵌入式操作系统基础知识

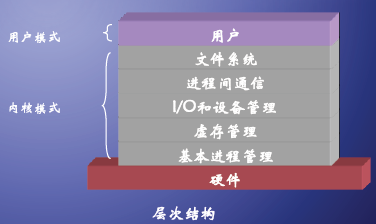
无操作系统——简单操作系统——实时操作系统——面向Internet

4.2.1嵌入式操作系统体系结构

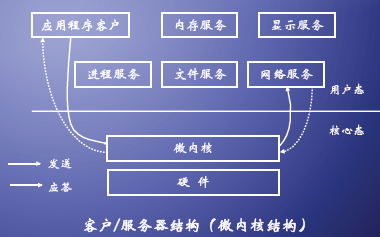
单体结构



分层结构



（CS）微内核结构



4.2.2任务管理

进程、线程、任务的概念

任务的实现（任务的层次结构、任务控制块、任务的状态及状态转换、任务队列）

任务调度（可抢占调度、不可抢占调度、先来先服务、时间片轮转算法、优先级算法）

实时系统及任务调度

任务间通信（共享内存、消息、管道、信号）

同步与互斥（竞争条件、临界区、互斥、信号量、死锁）

4.2.3存储管理

4.2.4设备管理

设备无关性、I/O地址、I/O控制、中断处理

4.2.5文件系统基础知识

4.2.6操作系统移植基础知识

4.3 嵌入式系统程序设计

4.3.1嵌入式软件开发基础知识

4.3.2嵌入式程序设计语言

汇编、编译、解释系统的基础知识和基本工作原理

(以下不考)

汇编语言

各类程序设计语言的主要特点和适用情况

4.3.3嵌入式软件开发环境

宿主机、目标机

编辑器、编译器、链接器、调试器、模拟器

常用嵌入式开发工具

集成开发环境

开发辅助工具

4.3.4嵌入式软件开发

## 嵌入式系统开发过程

嵌入式系统设计方法学

需求分析

规格说明

体系结构——软硬件划分

构件设计

系统集成

软硬件系统设计

HW-­‐SW

system involves

Specification（设计描述）

modeling（设计建模）

design space exploration and partitioning

synthesis and optimization（综合与优化）

Validation（设计验证）

implementation（设计实现）

Specification（设计描述）

List the functions of a system that describe the

behavior of an abstraction clearly with out ambiguity.

Modeling（设计建模）

Process of conceptualizing and refining the specifications,and producing a hardware and software model.

Validation:

Process of achieving a reasonable level of confidence（置信度） that the system will work as designed.

Takes different flavors per application domain（根据应用领域而不同）:

cosimulation for performance and correctness（性能与功能的协同仿真）

Implementation:

Physical realization of the hardware(through synthesis)

and of executable software (through compilation).

1. 嵌入式系统的调试技术。
   * + 交叉调试方式

|  |  |
| --- | --- |
| 交叉调度 | 非交叉调度 |
| 调试器和被调试程序运行在不同的计算机上 | 调试器和被调试程序运行在同一台计算机上 |
| 可独立运行，无需操作系统支持 | 需要操作系统的支持 |
| 被调试程序的装载有调试器完成 | 被调试程序的装载有专门的Loader程序完成 |
| 需要通过外部通信的方式来控制被调试程序 | 不需要通过外部通信的方式来控制被调试程序 |
| 可以直接调试不同指令集 | 只能直接调试相同指令集的程序 |

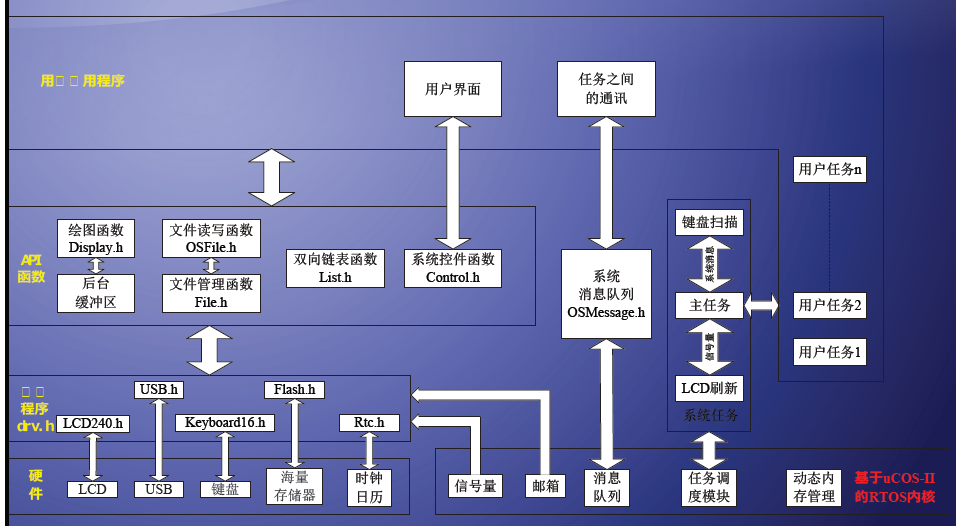
1. 理解实验流程。理解交叉编译：把在宿主机上编写的高级语言程序编译成可以运行在目标机上的代码，即在宿主机上能够编译生成另一种CPU（嵌入式微处理器）上的二进制程序。
2. 理解下载并运行程序的不同方法。

## ucOS-II

μC/OS--‐II是一个抢占式实时多任务内核。它是用ANSI的C语言编写的，包含一小部分汇编语言代码，使之可以提供给不同架构的微处理器使用。

μC/OS--‐II不支持时间片轮转调度法，所以赋予每个任务的优先级必须是不同。优先级号越低，任务的优先级越高。

任务调度、中断与时钟、同步与通信、存储管理。



1. 任务调度：

μC/OS\_II进行任务调度的思想是 “近似地每时每刻总是让优先级最高的就绪任务处于运行状态” 。为了保证这一点，它在系统或用户任务调用系统函数及执行中断服务程序结束时总是调用调度器，来确定应该运行的任务并运行它 。

μC/OS\_II进行任务调度的依据就是任务就绪表

1. 中断与时钟：
   * + μC/OS-II系统响应中断的过程为：
     + 系统接收到中断请求后，这时如果CPU处于中断允许状态（即中断是开放的），系统就会中止正在运行的当前任务，而按照中断向量的指向转而去运行中断服务子程序；当中断服务子程序的运行结束后，系统将会根据情况返回到被中止的任务继续运行或者转向运行另一个具有更高优先级别的就绪任务。
2. 同步与通信：
   * + 系统中的多个任务在运行时，经常需要互相无冲突地访问同一个共享资源，或者需要互相支持和依赖，甚至有时还要互相加以必要的限制和制约，才保证任务的顺利运行。因此，操作系统必须具有对任务的运行进行协调的能力，从而使任务之间可以无冲突、流畅地同步运行，而不致导致灾难性的后果。
3. 存储管理