

# 嵌入式系统重点

## 一、选择、填空、判断

---

### 1. 嵌入式系统的定义和基本要素

---

定义:

嵌入式系统是以应用为中心，以计算机技术为基础，软硬件高度定制，适用于对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。

特点:

- 专用性
- 可剪裁
- 可靠性
- 实时性
- 低功耗

### 2. 嵌入式系统体系结构、存储设备的分类

---

结构:

- 硬件平台：处理器、外围设备
- 操作系统
- 应用软件

处理器:

- 微处理器
- 微控制器
- 数字信号处理器（DSP）
- SOC

外围设备包括:

- 显示
- 存储

- 通信

## 存储设备

存储设备的分类：

- RAM
- ROM

RAM:

- SRAM (static: 静态不刷新)
- DRAM (dynamic: 动态刷新，成本低)

SDRAM (Synchronous):

SDR -- DDR -- DDR2 -- DDR3 -- DDR4

频率计算：

DDR200 MHz (100 \* 2) -- PC1600 (100 \* r \* 2 \* 8)

ROM:

- NOR ROM
- NAND ROM

NOR:

指标	NOR	NAND
软件执行	直接执行	无法直接执行
速度	读快写慢（写操作需要擦除和写入两个过程，每次查出只能擦除一个扇区，不能逐个擦除）	读慢写快
芯片容量	小	大
单元尺寸	大（扇区为单位）	小（块为单位）
寿命	10万次	100万次
平均价格	贵	便宜

### 3. 典型的嵌入式系统操作系统的特点

---

??

### 4. 嵌入式软件的测试阶段和测试策略

---

??

### 5. 实验用开发板的 BootLoader 是什么?

---

VIVI

### 6. Flash的分类

---

两种，见2.

### 7. 嵌入式处理器的分类，博创公司的S3C2410s 使用的处理器是什么?属于ARM的哪个系列

---

分类

四种，见2.

使用的是三星的 S3C2410 ，属于 ARM9 架构

### 8.什么是 ARM ?

---

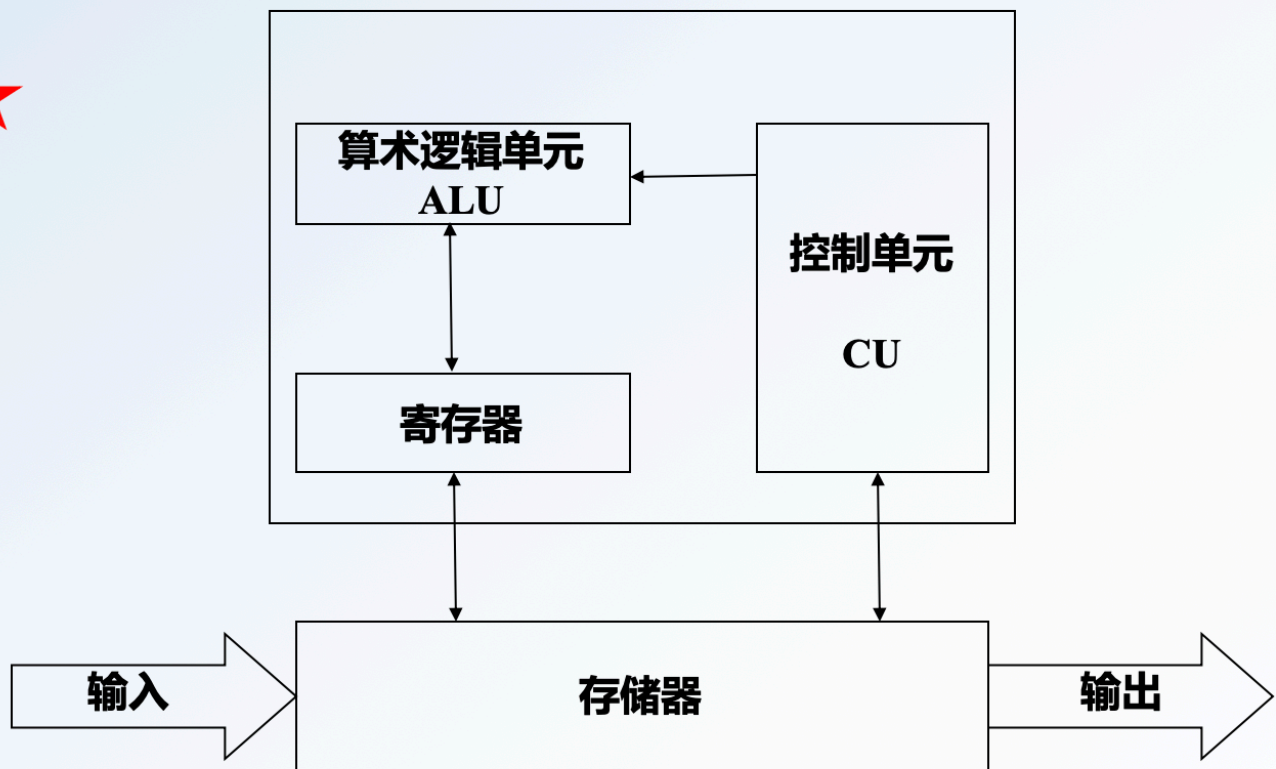
ARM (Advanced RISC machines) 先进精简指令及机器。是英国的一家微处理器行业的知名企业，该企业设计RISC (精简指令集) 处理器，只设计芯片，而不生产芯片。

### 9. 哈佛总线结构和冯·诺依曼结构各自的特点， 记住图

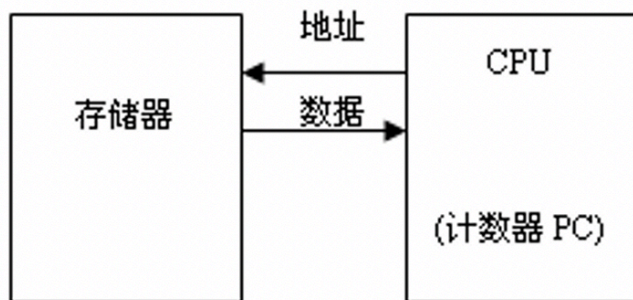
---

CPU

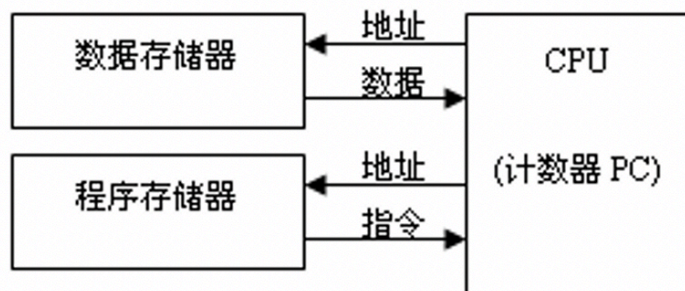
## 微处理器



体系结构



冯·诺依曼结构



哈佛结构



## 10. ARM处理器工作模式，特权模式、异常模式

## 11. ARM中寄存器的个数，哪些模式下的寄存器完全一样

## 12. 从应用上讲，Linux的主要组成部分(即Linux的基本操作系统结构)

- 内核
- shell
- 文件系统
- 实用工具

## 13. “[root@BC root]#”中各符号的含义

- root：用户名
- BC：主机名
- root：路径
- #：提示符

## 14. Vi的模式

---

- 命令行模式
- 插入模式
- 底行模式

## 15. GCC的基本用法

---

四个阶段：

- 预处理：-E .i
- 编译：-S .s
- 汇编：-c .o
- 连接：a.out

## 16. GDB本地调试的命令

---

- 首先 `gcc -g main.c` 编译
- `gdb a.out`
- `file a.out` #加载文件
- `list` #显示代码
- `break`
- `delete`
- `run (r)`
- `continue (c)`
- `step (s)`
- `next (n)` #nextline
- `finish (f)` # 跳出函数
- `print` #打印
- `quit (q)`

## 17. 基本Makefile结构

---

```
target : dependency_files      #依赖关系
<TAB>  command                 #命令
```

## 18. Make工具的使用

---

直接使用 `make` 命令，会默认指向第一个目标文件。

按照 `GNUmakefile` , `makefile` , `Makefile` 顺序搜索 `makefile`

也可以指定`Makefile`

```
make -f filename
```

## 19. Makefile变量

---

### 递归展开方式（引用赋值）

```
a = original_value
b = $(a)
a = later_value
# 现在 a=later_value
```

### 简单扩展方式（值传递）

```
a = original_value
b := $(a)
a = later_value
# 现在 a=original_value
```

## 20. Makefile规则

---

- 隐式规则
- 模式规则

## 21. Makefile应用和注意事项

---

### 小结

- 如果`make`命令中没有指定目标，则系统默认`target`指向描述文件中第一个目标文件（即第一个用：标志的对象）
- 按照依赖文件依次执行相应的命令
- 没有依赖关系的目标文件不被编译
- 变量的定义最好放在开始处
- `.h`文件可以不出现在依赖文件列表中
- `make clean`命令中如果不加文件名，默认的`Makefile`文件是`GNUmakefile`、`makefile`、`Makefile`。
- 当程序的规模较大时，文件依赖列表就会变得很长，或者命令的参数很多时，命令行很长，这时可采用

分行的做法。

## 22. 实验用ARM目标板采用的编译器(armv4l- unknown-linux-gcc)

---

## 23. Shell脚本文件结构

---

## 24. 位运算符

---

## 25. 交叉编译

---

- 所谓交叉编译就是在一个平台上生成可以在另一个平台上执行的代码。
- 编译最主要的工作是将程序转化成运行该程序的CPU所能识别的机器代码
- 交叉编译就是在特殊的编译环境下，如同翻译一样，把程序代码编译成不同的CPU所对应的机器代码。
- 交叉编译环境建立在宿主机（一台普通计算机）上，对应的嵌入式开发板叫做目标板（或开发板）。

## 26. minicom和Windows XP超级终端的特点和使用注意事项

---

- 若系统时 *Linux*，则只能使用minicom
- 若使用Windows和虚拟机的情况，minicom和windowsXP超级终端都可以使用。但是启动了minicom之后超级终端不可用，需要重新启动系统。

## 27. NFS服务、Samba服务的基本概念和配置方法

---

### NFS（network file system）

指网络文件系统。可以使网络上同为Linux或UNIX的操作系统通信。

```
mount -t nfs -o nolock 192.168.123.118:/arm2410s /mnt
```

### Samba

可以实现 Windows 和 Linux 不同系统之间的文件共享。

## 28. BootLoader的基本概念

---

启动加载程序



是系统加电后运行的第一段代码，是在操作系统内核运行前运行的一段小程序。

功能：

- 初始化硬件设备（CPU的主频、SDRAM、中断、串口等）
- 内核启动参数
- 启动内核
- 与知己交互，从串口，USB或者网口下载映像文件，并可以对Flash等存储设备进行管理

## 29. 一个嵌入式Linux系统从软件的角度看，分为四个层次

---

- BootLoader、（驱动程序）
- Linux内核
- 根文件系统
- 用户应用程序

## 30. BootLoader特点、安装媒介

---

特点：

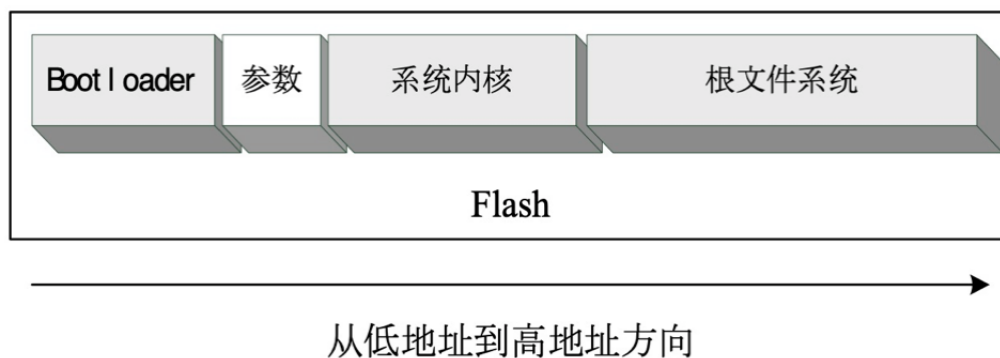
- 严重依赖于硬件，每种不同的CPU体系结构都有不同的 BootLoader
- BootLoader 还依赖具体的嵌入式版及配置

安装媒介：??

- 系统加电或复位后，所有的处理器通常都从某个预先安排的地址上取指令。比如，ARM在复位时从地址0x0 取指
- 嵌入式系统通常都有某种类的固态存储设备被映射到这个预先安排地址上。因此在系统加电后，处理器将首先执行BootLoader 程序
- BootLoader 是最先被系统执行的程序

## 31. 固态存储设备的典型空间分配结构

---



## 32. BootLoader的操作模式，哪种面向用户？哪种用于烧写 内核 和根文件系统？

---

用户：启动加载模式

烧写：下载

## 32. BootLoader与主机之间的通信设备及协议

---

串口：

- xmodem
- ymodem
- zmodem

以太网口：

TFTP

## 33. 几种流行的Linux BootLoader

---

- U-Boot
- BLOB
- RedBoot
- VIVI

## 34. vivi下载模式的进入方法

---

load 命令

## 35. 使用串口下载文件到Flash的一般命令

## 36. 操作系统内核的复杂性相当程度上来自于什么

内存管理

## 37. MMU的含义

内存管理单元

- 实现\*虚拟地址到物理地址映射
- n实现对地址访问的保护和限制

## 38. Linux系统启动时，使用什么将内核映像载入

BootLoader (stage2)

## 39. Linux的模块基本概念及相应的命令

- Linux 中的可加载模块（module）是 Linux 内核支持的动态加载模块，它们是内核的一部分，但是并没有编译到内核中。
- 模块可以单独编译成目标代码，以 .o 的形式存在。它可以再需要启动后动态加载到系统内核中。当模块不再被需要时，可以动态地卸载出系统内核。
- Linux 中大多数设备驱动程序或文件系统都是以模块形式存在
- 超级用户可以通过 `insmod` 和 `rmmod` 命令显示地将模块载入和卸载。内核也可以在需要时，请求内核守护进程装载和卸载模块。
- 由于模块技术使用内核更加模块化，因而成为一种增加内容到内核的较好方式，许多常用的设备驱动程序就做成模块。

### 命令

命令	功能
lsmod	列出已安装的 module
insmod	安装模块
rmmod	卸载模块
depmod	创建依赖文件

## 40. 系统调用是操作系统内核和应用程序之间的接口

---

## 41. 设备驱动程序是操作系统内核和机器硬件之间的接口

---

## 42. Linux系统的设备分类

---

- 字符设备
- 块设备
- 网络设备

## 43. 设备号及其分类

---

- 主设备号：对应一类设备，将设备与相应的驱动程序对应起来
- 次设备号：对应具体设备

## 44. 设备进入点的创建、查看和删除命令

---

命令	功能	举例
mknod	创建	<b>mknod /dev/demo_drv c 98 0</b>
ls		
rm		

## 45. 设备驱动程序加载、查看和卸载命令

---

命令	功能
insmod <设备驱动程序.o>	加载
lsmod	查看模块
rmmod <设备驱动程序>	删除驱动模块

## 46. 设备文件的存放位置

---

/dev/

## 47. file\_operations结构体

---

## 48. 基于Flash的文件系统和基于RAM的文件系统

---

### 基于Flash

- JFFS2
- YAFFS
- Cramfs

### 基于RAM

- RAMdisk
- Ramfs/tmpfs

## 49. 进程和线程的概念

---

- 包含关系
- 地址空间
- 独立性

任务 -- (至少有一个)-- 进程 --(至少有一个)-- 线程 进程：是一个程序的冬天执行，有独立的代码和数据空间，是资源分配的最小单位。线程之间具有独立性，崩溃后不会影响。线程：不具备独立的地址空间，共享地址空间，线程奔溃后进程也会崩溃。

## 50. 嵌入式GUI的特点

---

- 轻型
- 占用资源少
- 性能高
- 高可靠
- 可配置

## 51. Qt的两个支撑工具

---

- qmake
- Qt designer

## 52. Qt的基本概念及主要的类(3个)

---

Qt 是挪威Trolltech公司出品的一个跨平台的C++图形用户界面库。

- QObject
- QApplication
- QWidget

## 53. 信号和插槽的基本概念、机制和连接

---

信号和插槽是Qt自行定义的一种通信机制，它独立于标准的C/C++语言，因此要正确地处理信号和插槽，必须借助于一个称为moc（Meta Object Compiler）的Qt工具。

信号和插槽机制是一种强有力的对象间通信机制。

当对象改变状态时，信号就由该对象发射（emit）出去了，这就是对象所要做的全部工作，它不知道另一端是谁在接收这个信号。这就是真正的信息封装。

插槽用于接收信号，但它们是普通的对象成员函数。一个插槽并不知道是否有任何信号与自己相连接。而且，对象并不了解具体的通信机制

所有从QObject或其子类（例如QWidget）派生的类都能包含信号和插槽。

## 54. 声明信号的关键字，信号的特点

---

## 55. 包含signal和slot的类需要的关键字

---

在类内加入关键字： `Q_OBJECT`

声明关键字： `signals slots`

## 56. Qt程序的框架

---

### 设计流程

- 创建和初始化子部件
- 设置子部件布局
- 设置Tab键次序（不是必须）
- 建立信号与插槽的连接
- 编写事件处理函数

## 57. qvfb的使用

---

Qvfb工具是一个非常有用的工具，用它来生成Virtual FrameBuffer，qvfb可以模拟在开发板上的显示情况，如果在Virtual FrameBuffer中运行没有问题的话，可以直接通过交叉编译在开发板上运行

## 58. 嵌入式数据库的特点

- 体积小，嵌入式数据库编译后的产品不过几十千字节。
- 可靠性，应该可以预知嵌入式数据库的操作、系统的大小、性能
- 提供功能定制，可以根据其具体应用的环境来定制数据库的系统功能
- 支持SQL查询语言，提供数据库及数据表的管理功能
- 提供接口函数，以供在高级语言中调用
- 实时性
- 有一定的底层控制能力

## 59. SQLite最大特点

无数据类型

## 60. SQLite3的命令

Linux安装软件的步骤： \* 解压sqlite-3.5.9.tar.gz \* 配置 \* 编译并安装 \* 验证

命令（第二行开始需要在前面加 '.'）	功能
sqlite3 <数据库名>	创建数据库
tables	查看当前数据库中有所少个数据表
schema	查看表结构
database	查看当前数据库
output <文件名>	把查询结果输出到文件
dump <表明>	输出表结构，同时输出操作记录
help	查看帮助
quet/exit	退出

## 61. 含有SQLite3的API函数的应用程序，在编译时需要加上的参数

## 62. SQLite的驱动程序是什么?Qt4要实现与数据库的连接需用到的类

---

QSQLITE

QSqlDatabase 类

## 63. 套接字、套接字类型

---

IP:端口，即 IP 与端口的组合

- 字节流套接字
- 数据报套接字
- 原始套接字

## 64. Socket网络函数所需的头文件

---

```
#include <sys/types.h>
#include <socket.h>
```

## 65. Web服务器的主要任务

---

通常是在远程机器上，负责对浏览器页面请求作出响应，返回HTML编码或类似的数据流。

## 66. Qt提供的用于TCP协议编程的两个类

---

- QTcpServer
- QTcpSocket