1. 概念题

大家随缘

1. 如何构建交叉编译环境，为什么要构建检查编译环境

构建交叉编译环境有两种方式，从现成的二进制代码构建和从源代码安装。

从现成的二进制代码构建：

1. 下载源文件，补丁和建立编译的目录
2. 建立内核头文件
3. 建立二进制工具（binutils）
4. 建立初始编译器（bootstrap gcc）
5. 建立c库（glibc）
6. 建立全套编译器（full gcc）

从源代码安装：

下载arm-elf-tools的最新版本或合适的版本，在root权限下运行sh./arm-elf-tools-20030314.sh.

为什么要构建交叉环境？因为嵌入式系统开发属于交叉平台开发，开发软件的系统和运行软件的系统不同，目的平台可能不允许或者不能够安装运行所需要的编译器，也可能目的平台都没有建立，连操作系统都没有，更谈不上运行编译器。

1. helloworld的开发在pc上和在嵌入式linux上有何不同

在pc上开发的开发环境大多是windows，linux，macos，直接在编译器上书写程序，写完后编译器编译链接，生成的可执行文件直接在本机上运行。

嵌入式linux下的应用开发首先应该是在pc机上的linux开发环境下开发，编译时使用交叉编译器编译，然后将编译生成的可执行文件，传到嵌入式设备上，方法有几种，通过串口，U盘，网络(ftp)，网络文件系统(nfs)，修改一下执行权限就可以执行了。 以上这些操作的前提是嵌入式设备中已经装好了linux系统。

1. 嵌入式操作系统的组成

由上而下分为：硬件无关的软件层，硬件抽象层：板级支持包（BSP）与设备驱动，硬件层。

（以下排序均自上而下）

硬件无关的软件层：应用层，{I/O系统，RTOS库，文件系统，RTOS核心，tcp/ip协议栈}。

硬件抽象层：{scsi驱动，flash驱动，mmu驱动，cache驱动，serial（串口）驱动，ethernet（以太网）驱动，设备驱动}。

1. 如何保证嵌入式linux库函数的完整性
2. 如何实现程序自启动

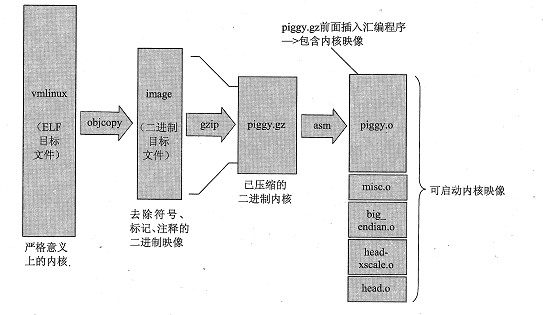
使用/etc/init.d/(或/etc/rc.d/init.d)下的脚本来配置应用程序的自动启动。

进入根文件系统的/etc/init.d目录下，修改rcS文件，加入需要启动的程序脚本。（参考实验3）

1. 压缩内核和未压缩内核在使用上的不同

make指令形成未压缩内核vmlinux： 这是一个没有压缩ELF映像文件，含有符号表，注释，调试信息。是严格意义上的linux内核，当通过引导装入程序启动linux内核时， vmlinux内核二进制映像将启动硬件，并且执行一个完整功能的内核。

Make zImage形成内核压缩镜像zimage：该文件就是压缩后vmlinux 加上第二阶段相关boot引导程序组成。是二进制内核模块，去掉了符号，标记，注释。



1. linux上有那些时钟，他们是怎么实现的

Linux上有两种时钟，一个是由主板电池驱动的RTC（real time clock）时钟，另一个是内核时钟，有软件来根据时间中断进行计数。内核时钟在系统关机时不存在，操作系统启动时读取RTC时间进行同步，并在系统关机时将时间写回RTC。

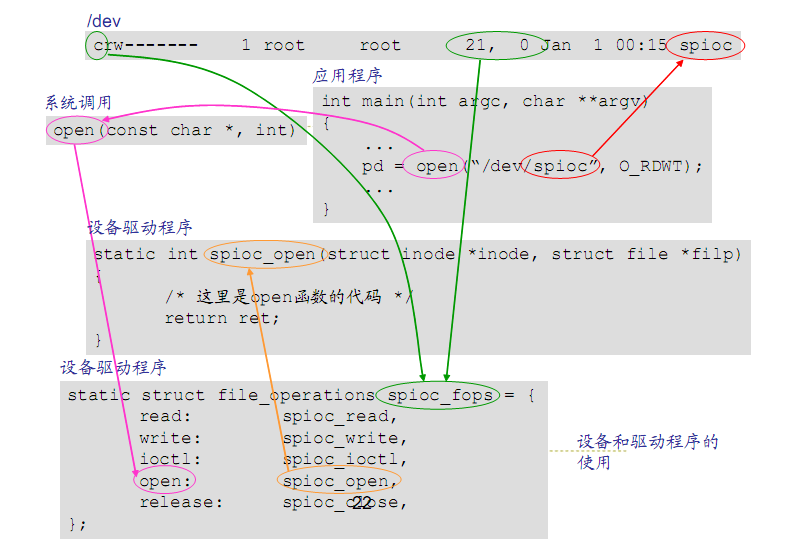
xtime是从系统中取得的时间，一般是从某一历史时刻开始到现在的时间，即操作系统上显示的时期，它的精度是微秒。

jiffies是记录着从电脑开机到现在总共的时钟中断次数。jiffies取决于系统的频率，单位是 Hz，是周期的倒数，周期一般是一秒钟中断产生的次数。LINUX系统时钟频率是一个常数HZ来决定的， 通常HZ＝100（Linux内核从2.5版内核开始把频率从100调高到1000），那么他的精度就是10ms。也就是说每10ms一次中断。所以一般来说Linux的精确度是10毫秒。

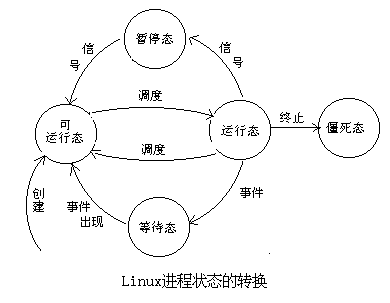
内核一般通过jiffies值来获取当前时间。尽管该数值表示的是自上次系统启动到当前的时间间隔，但因为驱动程序的生命期只限于系统的运行期 (uptime)，所以也是可行的。驱动程序利用jiffies的当前值来计算不同事件间的时间间隔。硬件给内核提供一个系统定时器用以计算和管理时间， 内核通过编程预设系统定时器的频率(即上面所说的HZ=100）。节拍率（tick rate)，每一个周期称作一个tick(节拍）。jiffies是内核中的一个全巨变量。系统启动一来产生的节拍数。譬如，如果计算系统运行了多长时间，可以用jiffies/tick rate 来计算。

1. 子函数调用的实现流程

一般简单的驱动，被调用是由应用层触发的，然后通过系统调用陷入到内核，内核vfs层会根据你的驱动设备号匹配对其驱动，然后函数就被调用了。



1. 进程调度时机
2. 进程状态发生变化时



1. 当前进程时间片用完时
2. 进程从系统调用返回到用户态时
3. 中断处理后，进程返回到用户态时

*个人整理，感谢张备同学提供题目，对错概不负责，如有雷同，关我啥事*