

1. 作为Linux内核的重要组成部分，设备驱动程序主要完成哪些功能？

答：(1) 对设备初始化和释放。

(2) 把数据从内核传送到硬件和从硬件读取数据。

(3) 读取应用程序传送给设备文件的数据和回送应用程序请求的数据。

(4) 检测错误和处理中断。

2. 内核空间物理页分配技术主要使用了哪些方法？

答：1 基于**slab**分配器的管理技术，**slab**分配器主要的功能就是对频繁分配和释放的小对象提供高效的内存管理。它的核心思想是实现一个缓存池，分配对象的时候从缓存池中取，释放对象的时候再放入缓存池。

2 内核非连续内存分配(**vmalloc**)。内核提供**vmalloc**函数分配内核虚拟内存，该函数不同于**kmalloc**，它可以分配较**kmalloc**大得多的内存空间（可远大于**128K**，但必须是页大小的倍数），但相比**Kmalloc**来说，**vmalloc**需要对内核虚拟地址进行重映射，必须更新内核页表，因此分配效率上相对较低。

3.简述交叉开发模式一般采用的三个步骤？

答：（1）在主机上编译 **BootLoader**(引导加载程序)，然后通过 **JTAG** 接口烧写到目标板。这种方式速度较慢，一般在目标板上还未运行起可用的 **BootLoader** 时采用。如果开发板上已经运行起可用 **BootLoader**，并且支持烧写 **Flash** 功能，则可利用 **BootLoader** 通过网络下载映像文件并烧写，速度较快。

（2）在主机上编译 **Linux** 内核，然后通过 **BootLoader** 下载到目标板以启动或烧写到 **Flash**。为了方便调试，内核应该支持网络文件系统(**Network File System, NFS**)，这样，目标板启动 **Linux** 内核后，可以通过 **NFS** 方式挂载根文件系统。

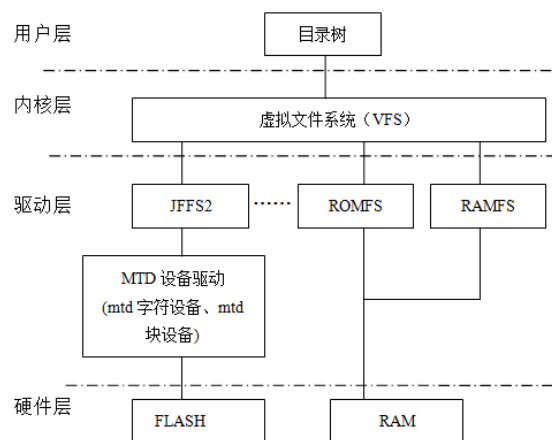
（3）在主机上编译各类应用程序，通过 **NFS** 运行、调试这些程序，验证无误后再将制作好的文件系统映像烧写到目标板。

4.什么是嵌入式文件系统？请简要说明嵌入式Linux文件系统的层次结构。

答：

嵌入式文件系统是指嵌入式系统中实现文件存取、管理等功能的模块，这些模块提供一系列文件输入输出等文件管理功能，为嵌入式系统和设备提供文件系统支持。

嵌入式Linux文件系统的层次结构自下而上主要由硬件层，驱动层，内核层和用户层组成，如下图所示。



5.什么是根文件系统？其主要配置目录有哪些？

答：根文件系统是一种特殊的文件系统，该文件系统不仅具有普通文件系统的存储数据文件的功能，它还是内核启动时所挂载（**mount**）的第一个文件系统，内核代码的映像文件保存在根文件系统中，系统引导启动程序会在根文件系统挂载之后从中把一些初始化脚本和服务加载到内存中去运行。

主要配置目录：**/bin**，**/dev**，**/etc**，**/lib**，**/proc**，**/var**，**/usr**等

6. BootLoader的功能是什么？请简要说明BootLoader的普遍工作流程。

答：**BootLoader**是嵌入式系统在加电后执行的第一段代码，在它完成CPU和相关硬件的初始化之后，再将操作系统映像或固化的嵌入式应用程序装载到内存中然后跳转到操作系统所在的空间，启动操作系统运行。

工作流程：

BootLoader启动大多数都分为两个阶段。

第一阶段主要包含依赖于CPU的体系结构硬件初始化的代码，通常都用汇编语言来实现。这个阶段的任务有：

- 基本的硬件设备初始化（屏蔽所有的中断、关闭处理器内部指令/数据Cache等）。
- 为第二阶段准备RAM空间。
- 如果是从某个固态存储媒质中，则复制**BootLoader**的第二阶段代码到RAM。
- 设置堆栈。
- 跳转到第二阶段的C程序入口点。

第二阶段通常用C语言完成，以便实现更复杂的功能，也使程序有更好的可读性和可移植性。这个阶段的任务有：

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

7.Linux设备驱动程序分类有哪些？

答：(1). 字符设备

字符设备是传输数据以字符为单位进行的设备，字符设备驱动程序通常实现 **open**、**close**、**read**和**write**等系统调用函数，常见的字符设备有键盘、串口、控制台等。

(2). 块设备

所谓块设备是指对其信息的存取以“块”为单位。如常见的光盘、硬磁盘、软磁盘、磁带等，块长大小通常取**512B**、**1024B**或**4096B**等。块设备和字符设备一样可以通过文件系统节点来访问。在大多数linux系统中，只能将块设备看作多个块进行访问，一个块设备通常是**1024B**数据。

(3). 网络设备

网络设备驱动通常是通过套接字（**Socket**）等接口来实现操作。任何网络事务处理都可以通过接口来完成和其他宿主机数据的交换。内核和网络设备驱动程序之间的通信与字符设备驱动程序和块设备驱动程序与内核的通信是完全不同的。

8.Linux设备驱动程序有哪些组成部分？

答：（1）对子程序进行自动配置和初始化，检测驱动的硬件设备是否正常，能否正常工作。

（2）设备服务子程序和中断服务子程序，这两者分别是驱动程序的上下两部分。驱动上部分即设备服务子程序的执行是系统调用的结果，并且伴随着用户态向核心态的演变，在此过程中还可以调用与进程运行环境有关的函数，比如sleep()函数。驱动程序的下半部分即中断服务子程序。

9. Linux设备驱动程序是如何加载进内核的？

答：通常 linux 驱动程序可通过两种方式进行加载：

一种是将驱动程序编译成模块形式进行动态加载，常用命令有insmod（加载）、rmmod（卸载）等；

另一种是静态编译，即将驱动程序直接编辑放进内核。动态加载模块设计使 Linux 内核功能更容易扩展。而静态编译方法对于在要求硬件只是完成比较特定、专一的功能的一些嵌入式系统中，具有更高的效率。

10. 字符设备驱动程序编写通常都要涉及三个重要的内核数据结构，请简要叙述。

答：字符设备驱动程序编写通常都要涉及到三个重要的内核数据结构，分别是**file_operations**结构体、**file**结构体和**inode**结构体。

- **File_operations**为用户态应用程序提供接口，是系统调用和驱动程序关联的重要数据结构。
- **File 结构体**在内核代码 **include/linux/fs.h** 中定义，表示一个抽象的打开的文件，**file_operations** 结构体就是 **file** 结构的一个成员。
- **Inode** 结构表示一个文件，而 **file** 结构表示一个打开的文件。这正是二者间最重要的关系。

11.请简要叙述字符设备驱动程序의初始化流程。

答：字符设备驱动程序의初始化流程一般可以用如下的过程来表示：

- (1) 定义相关的设备文件结构体（如file_operation()中的相关成员函数的定义）。**
- (2) 向内核申请主设备号（建议采用动态方式）。**
- (3) 申请成功后，通过调用MAJOR()函数获取主设备号。**
- (4) 初始化cdev的结构体，可以通过调用cdev_init()函数实现。**
- (5) 通过调用cdev_add()函数注册cdev到内核。**
- (6) 注册设备模块，主要使用module_init()函数和module_exit()函数。**

12. Linux设备模型的主要功能有哪些？

答：其主要功能包括：

- (1) 电源管理和系统关机
- (2) 与用户空间通信
- (3) 热插拔(hotplug)设备管理
- (4) 设备类型管理
- (5) 对象生命周期处理

13. 简述QT5的信号和插槽机制。

答：信号和插槽用于两个对象之间的通信，信号和插槽（**signal/slot**）机制是Qt的核心特征，这也是Qt不同于其他开发框架的最突出的特点。信号和插槽是Qt自定义的一种通信机制，它独立于标准的C/C++语言，所有从QObject或其子类派生的类都能够包含信号和插槽。在Qt的部件类中已经定义了一些信号和插槽，但是更多的做法是子类化这个部件，然后添加自己的信号和插槽来实现想要的功能。

- 事件处理的方式也是回调
- 当对象状态发生改变的时候，发出**signal**通知所有的**slot**接收**signal**，尽管它并不知道哪些函数定义了**slot**，而**slot**也同样不知道要接收怎样的**signal**
signal和**slot**机制真正实现了封装的概念，**slot**除了接收**signal**之外和其它的成员函数没有什么不同，而且**signal**和**slot**之间也不是一一对应的。



14. 什么是嵌入式数据库系统？其主要特点是什么？

答：嵌入式数据库系统是指支持移动计算或某种特定计算模式的数据库管理系统，它通常与操作系统和具体应用集成在一起，运行在智能型嵌入式设备或移动设备上。

其主要特点是：

- 嵌入性；
- 移植性；
- 安全性；
- 实时性；
- 可靠性；
- 主动性；