1. **基本概念：**
2. Norflash与Nandflash的区别  
    (1)、NAND闪存的容量比较大  
    (2)、由于NandFlash没有挂接在地址总线上，所以如果想用NandFlash作为系统的启动盘，就需要CPU具备特殊的功能，  
    如s3c2410在被选择为NandFlash启动方式时会在上电时自动读取NandFlash的4k数据到地址0的SRAM中。  
    (3)、NAND Flash一般地址线和数据线共用，对读写速度有一定影响。NOR Flash闪存数据线和地址线分开，所以相对而言读写速度快一些。
3. 反码、补码:

反码：对原码除符号位外的其余各位逐位取反就是反码

补码：负数的补码就是对反码加1  
正数的原码、反码、补码都一样

1. ROM、RAM、SRAM、DRAM、SDRAM

ROM：是只读存储器，掉电不丢失  
RAM：是读写存储器，掉电丢失  
SRAM：CPU的缓存就是SRAM，静态的随机存取存储器，加电情况下，不需要刷新，

数据不会丢失  
DRAM：动态随机存取存储器最为常见的系统内存，需要不断刷新，才能保存数据  
SDRAM：同步动态随机存储器，即数据的读取需要时钟来同步。

1. 编译和链接有什么不同？（如外部符号的处理）  
    （1）编译生成的是目标文件(object  \*.o)；编译过程中对于外部符号不做任何解释和处理。外部符号对应的就是“符号”

（2）链接生成的是可执行程序；链接将会解释和处理外部符号。外部符号对应的是地址。

1. 引用：

引用：是一个变量的另一个名字，又称别名。定义方式：   
int a=10；   
int &b=a；

在这里，意思就是给a变量起了一个新名字b，因此b不可再次被重新定义。   
引用必须初始化，无空引用，并且引用不分等级。

1. 引用（C++）和指针（C）的区别：

(1). 指针是一个实体，而引用仅是个别名；  
(2). 引用使用时无需解引用（\*），指针需要解引用；  
(3). 引用只能在定义时被初始化一次，之后不可变；指针可变；  
(4). 引用没有 const，指针有 const，const 的指针不可变；  
(5). 引用不能为空，指针可以为空；  
(6). “sizeof 引用”得到的是所指向的变量（对象）的大小，  
    而“sizeof指针”得到的是指针本身（所指向的变量或对象的地址）的大小；  
(7). 指针和引用的自增（++）运算意义不一样；

1. 全局变量和局部变量的区别。  
   **全局变量**：储存在静态区.进入main函数之前就被创建.生命周期为整个源程序;   
   **局部变量：**在栈中分配.在函数被调用时才被创建.生命周期为函数内。
2. [**Linux**](http://lib.csdn.net/base/linux)驱动程序流程及功能。  
   设备驱动程序的功能：  
   对设备初始化和释放；  
   把数据从内核传送到硬件和从硬件读取数据；  
   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；  
   检测和处理设备出现的错误；
3. 冒泡排序 --- 时间换空间  
   快速排序，堆排序 --- 空间换时间
4. 局部变量能否和全局变量重名？  　　  
    **能，局部会屏蔽全局**。要用全局变量，需要使用 ":: "  局部变量可以与全局变量同名，**在函数内**引用这个变量时，**会用**到同名的**局部变量**，而**不会**用到**全局变量。**
5. **如何引用一个已经定义过的全局变量？**  　　  
   **extern** 可以用引用头文件的方式，也可以用extern关键字，如果用引用头文件方式来引用某个在头文件中声明的全局变理。  
   假定你将那个变量写错了，那么在编译期间会报错。

如果你用extern方式引用时，假定你犯了同样的错误，那么在编译期间不会报错，而在连接期间报错。

1. 全局变量可不可以定义在可被多个.C文件包含的头文件中？为什么？

可以，**在不同的C文件**中以**static形式来声明同名全局变量**。

可以在不同的C文件中声明同名的全局变量，前提是其中**只能有一个C文件中对此变量赋初值**，此时连接不会出错

1. 局部变量：存在于（堆栈）中；

全局变量：存在于（静态区 ）中；

动态申请数据：存在于（堆）中。

1. IP地址的编码分为哪俩部分？  
    IP地址由两部分组成：网络号+主机号。

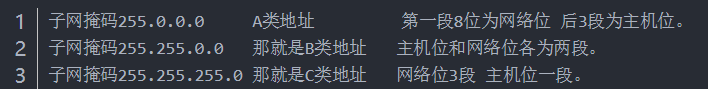
要和“子网掩码”按位与之后才能区分网络位、主机位

**网络位：**该地址所处的网络能划分几个网段；

**主机位：**每个网段容纳多少台电脑；

**IP地址：**网络部分对应的掩码部分全为“1”，主机部分对应的掩码全为“0”。

**eg:**



1. 共享内存：( shared memory)：

共享内存就是映射一段能被其他进程所访问的内存，  
这段共享内存由一个进程创建，但多个进程都可以访问。共享内存是最快的IPC方式。