1. Linux的学习方法-框架学习法

学习框架：

---单片机+C语言

---Linux基础框架(Bootloader、Kernel、文件系统)

---Linux应用程序编程

---Linux字符类驱动程序

---深入学习，不断迭代

ARM9：外设不丰富，但有MMU，MMU是上大型操作系统必须的硬件支持，如Linux、WinCE

Cortex-M系列无MMU(内存管理单元)

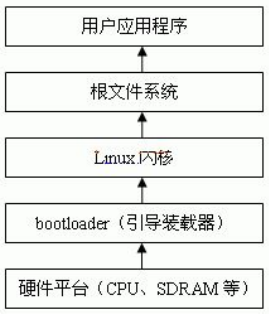
补：MPU(Cortex-M有)内存保护单元

Linux框架：

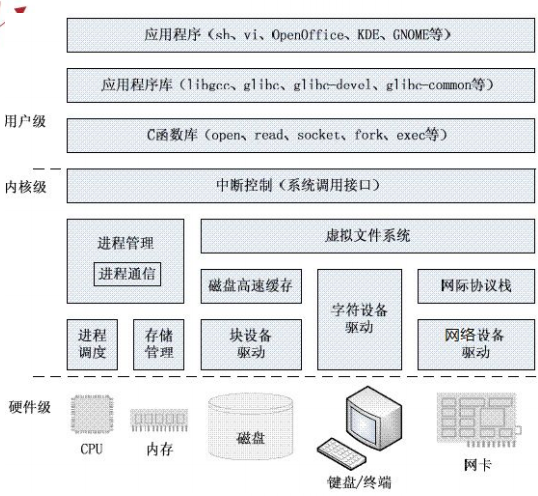
* 单片机和C语言知识
* 引导程序Bootloader：初始化外部设备
* 装载Linux内核(Kernel)：调整到Linux内核执行UBOOT
* 挂载文件系统：如Android 、QT、Ubuntu(三种差异在于文件系统不同)

跟4412开发板学习步骤：

* 学习编程和烧写系统
* 学习应用程序编程
* 学习驱动程序编程-驱动程序属于Linux内核的一部分



Linux框架



Linux 的驱动程序

1. **嵌入式研发岗位**

**--嵌入式驱动工程师**

* 单片机原理
* C语言编程经验
* Linux基础框架知识

**--高级嵌入式驱动工程师--8年起**

* 数据结构
* 算法原理

**--系统架构工程师**

* 计算机系统体系架构
* 操作系统(从抽象到具体)

1. **学习Linux时间**

* --学习编程和烧写系统
* 学习应用程序编程(即系统编程)-每天2小时，1年后可转做驱动(要求英文读写都很高)
* 学习驱动程序编程(BSP)-3年到熟练的程度

1. **4412开发板资料学习顺序**

* **主线使用《精英版用户手册》**
* **副线使用《用户入门手册》**
* **学习《Linux系统编程》**

**《精英版用户手册》-3个月**

* **平台软软件基础软件安装与学习：**
* **烧写开发板镜像**(uboot(和硬件相关，分POP/SCP1G/SCP2G)、内核镜像(zImage，分POP和SCP版本)、系统文件(即文件系统,各个核心板通用))
* **Android 系统编译**(Linux下搭建编译开发环境)
* **Linux-C程序测试：**测试硬件及驱动；C程序在Android上运行需要使用gcc4.4.1
* **定制Linux内核：**
* **Android应用开发-**环境搭建,可在windows平台搭建

**《用户入门手册》**

**对《精英版用户手册》种技术细节的展开讨论**

**《Linux系统编程》--六个月**

**60多个实验：用于学习Linux系统编程：**

* **最小系统(无图形)：搭建开发环境(busybox)、编译、烧写**
* **编译在开发板可以运行的程序，使用U盘/TF下载程序到开发板**
* **了解linux种的用户，用户组，权限**
* **文件IO操作**
* **字符设备控制**
* **串口编程：**和单片机出口开发完全不同
* **网络通信：**Linux支持所有的网络通信协议，如TFTP服务器搭建和测试、在开发板实现TFTP文件传输、实现NFS网络文件系统
* **man命令使用：**用于查看帮助文档
* **文件管理和目录**
* **进程及进程通信：**用于实现多任务，主要了解系统接口(API/函数)
* **网络通信：套接字TCP、套接字UDP、web服务器**

1. **U-boot--和硬件相关**

**u-boot：**

**<https://www.cnblogs.com/princepeng/p/11212854.html>**

**BIOS、u-boot(bootloader一部分)、bootloader(引导程序)**

**<https://blog.csdn.net/conowen/article/details/7253620>**

-