Linux设备驱动开发详解日志

引申概念

```
GNU计划
Posix标准
GNU C与ANSI C语法差异
```

内核组成

1. 进程调度SCHED：属于内核的中心位置，其他子系统都依赖它；当进程请求的资源不能得到满足的时候，驱动会调度其他进程执行，对于进程会进入睡眠状态，直到它请求的资源被释放，才会被唤醒而进入就绪。进程的状态有：运行态、就绪态、睡眠态、暂停态等。
2. 内存管理MMU：控制进程安全i共享主内存区域，一般而言，Linux的每个进程享有4GB内存，其中0~3GB属于用户空间，由用户进程占有，低4GB属于内核占有。
3. 虚拟文件系统VFS：隐藏不同的文件系统或设备文件的硬件初始化等不同的操作，抽象出统一的操作接口给调用设备。
4. 进程间通信IPC：提供如信号量、共享内存、邮箱、管道等多种通信机制协助多个进程、多资源的互斥访问、同步、资源共享和消息传递。
5. 网络接口NET：提供对各种网络标准的存取和各种网络硬件的支持。分为网络协议和网络设备程序。

内核引导

1. BIOS:一般是x86 pc具有，固化在ROM或flash中，系统上电后程序PC指针被赋值为BOIS特定的启动地址，然后BIOS会去搜索可以引导的设备，如硬盘、Flash等，如果从硬盘启动，就会加载MBR中的内容到RAM。
2. Bootloader：
    1. 可以在系统上电或复位后以某种方式执行的程序；包括被BIOS引导执行、直接在NOR Flash中执行、NAND Flash中的代码被MCU自动拷贝到RAM中执行等等；
    2. 能将U盘、Flash、ROM、SD卡等存储介质，甚至网口中的操作系统加载到RAM中、并把控制权交给操作系统源代码执行。
    3. 著名的Bootloader包括LILO\GRUB\RedBoot等。
3. MBR：磁盘主引导程序
4. 内核：当内核映像zImage被加载到RAM中，Bootloader的控制权就被释放，zImage是一个gzip压缩文件，且包含一个未被压缩的解压引导代码。
5. init进程：

```
//执行基本的硬件设置
start()//bool/head.S,
//设置堆栈，清除BSS段
startup_32()//bool/compressed/head.S,
//解压内核
decompress_kernel()//bool/compressed/msic.c/,
//进程0，初始化页表，启动内存分页机制
```

```
startup_32()//kernel/head.S
//初始化中断设置，进一步的内存配置
start_kernel()//init/main.c
//启动第一个核心线程进入用户模式，执行init(),原来的线程调用cpu_idle()等待调度
kernel_thread()//kernel/process.c
    //核心线程1,完成外设和驱动程序的加载和初始化，挂载根fs，重定向标准输出到控制台
 (dev/console) 等,
    //然后搜索init程序，也可以由init=指定init程序
    //使用execve系统调用执行init程序
    init()//init/main.c
    //原有执行序列调用cpu_idle等待调度
    cpu_idle()//init/main.c
```

6. 用户空间：