

Linux内核代码：简介

李中国

苏州大学计算机科学与技术学院

May 27, 2012

Linux内核源代码目录树

目录	描述
arch	硬件体系结构相关的代码
Documentation	内核源代码文档
drivers	设备驱动程序
fs	文件系统
include	内核头文件
ipc	进程间通讯代码
init	系统启动及初始化代码
kernel	核心代码（如进程调度器等）
mm	内存管理系统、虚拟内存系统
net	网络系统代码（如TCP/IP协议栈）
lib	辅助函数库

Linux内核与普通应用程序的区别

- ▶ 内核不能使用C语言标注库函数（及C标准头文件）
- ▶ Linux内核编程使用的是GNU C而不是标准C语言
- ▶ 内核代码缺乏内存保护机制（应用程序则有）
- ▶ 与应用程序比，内核运行时的栈非常小
- ▶ 内核代码必须正确处理同步与并发问题
- ▶ 可移植性对内核而言非常重要

为什么内核不能使用C语言库函数及头文件?

原因

- ▶ chicken-and-egg问题
- ▶ C标准库对内核而言太庞大、低效

Linux内核的应对策略

- ▶ 自己实现部分库函数功能: `lib/string.c`, 对应头文件`<linux/string.h>`
- ▶ 标准库中**printf**函数在内核中对应的是**printk**
 - ▶ 例如`printk(KERN_ERR "this is an error!\n")`

Linux内核使用GNU C编程

- ▶ GNU C支持内联函数(与宏相比有什么有点?)
 - ▶ 内联函数在使用前需先定义；一般置于头文件中
 - ▶ 例: `static inline void wolf(unsigned long t)`
- ▶ 内联汇编语言: 可以在C程序中直接使用汇编语句
 - ▶ 例: `asm volatile("rdtsc":"=a"(l), "=d"(h));`
- ▶ GNU C支持条件分支优化, 例如:

```
/* 'error' is nearly always zero */
if (unlikely(error)) {
    ...
}
```

```
/* 'success' is nearly always nonzero */
if (likely(success)) {
    ...
}
```

内核中没有内存保护机制

- ▶ 应用程序访问非法内存地址？
 - ▶ 操作系统会给该进程发送SIGSEGV信号并将其杀死
- ▶ 谁来阻止操作系统访问非法地址？
- ▶ 内核所占的物理内存无法进行分页
 - ▶ 所以内核在使用物理内存时必须特别节俭

核心态下的栈空间

- ▶ 应用程序可以使用的栈是动态增长的、具体大小仅由地址空间限制
- ▶ 内核可以利用的栈大小是固定的(2 pages):
 - ▶ 32位x86平台上，固定为8K
 - ▶ 64位平台上固定为16K
- ▶ 内核在编译前可以配置其所利用的栈为4K