

介绍Linux内核函数Start_kernel()的功能 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/240/2021_2022__E4_BB_8B_E7_BB_8DLinu_c103_240663.htm Start_kernel()中调用了一系列初始化函数，以完成kernel本身的设置。这些动作有的是公共的，有的则是需要配置的才会执行的。在start_kernel()函数中，输出Linux版本信息（printk(linux_banner)）设置与体系结构相关的环境（setup_arch()）页表结构初始化（paging_init()）使用"arch/alpha/kernel/entry.S"中的入口点设置系统自陷入口（trap_init()）使用alpha_mv结构和entry.S入口初始化系统IRQ（init_IRQ()）核心进程调度器初始化（包括初始化几个缺省的Bottom-half，sched_init()）时间、定时器初始化（包括读取CMOS时钟、估测主频、初始化定时器中断等，time_init()）提取并分析核心启动参数（从环境变量中读取参数，设置相应标志位等待处理，（parse_options()）控制台初始化（为输出信息而先于PCI初始化，console_init()）剖析器数据结构初始化（prof_buffer和prof_len变量）核心Cache初始化（描述Cache信息的Cache，kmem_cache_init()）延迟校准（获得时钟jiffies与CPU主频ticks的延迟，calibrate_delay()）内存初始化（设置内存上下界和页表项初始值，mem_init()）创建和设置内部及通用cache（"slab_cache"，kmem_cache_sizes_init()）创建uid taskcount SLAB cache（"uid_cache"，uidcache_init()）创建文件cache（"files_cache"，filescache_init()）创建目录cache（"dentry_cache"，dcache_init()）创建与虚存相关的cache（"vm_area_struct"，"mm_struct"，vma_init()）块设备读写缓冲区初始化（同时创建"buffer_head"cache用户加速访问

, `buffer_init()`) 创建页cache (内存页hash表初始化
, `page_cache_init()`) 创建信号队列cache ("signal_queue"
, `signals_init()`) 初始化内存inode表 (`inode_init()`) 创建内存
文件描述符表 ("filp_cache" , `file_table_init()`) 检查体系结构
漏洞 (对于alpha , 此函数为空 , `check_bugs()`) SMP机器其
余CPU (除当前引导CPU) 初始化 (对于没有配置SMP的内
核 , 此函数为空 , `smp_init()`) 启动init过程 (创建第一个核心
线程 , 调用`init()`函数 , 原执行序列调用`cpu_idle()` 等待调度
, `init()`) 至此`start_kernel()`结束 , 基本的核心环境已经建立起
来了。 100Test 下载频道开通 , 各类考试题目直接下载。详细
请访问 www.100test.com