

linux 3.6 启动源码分析(五) kernel_init进程

原创

2013年12月16日 14:58:49

3058

在start_kernel最后的rest_init函数中内核创建了两个内核线程，一个是内核线程的管理者，另一个是内核初始化线程kernel_init。

kernel_init它将完成设备驱动程序的初始化，并调用init_post函数启动用户空间的init进程。

```
[cpp]
1. static int __init kernel_init(void * unused)
2. {
3.     /*Wait until kthreadd is all set-up.*/
4.     wait_for_completion(&kthreadd_done);
5.     /* Now the scheduler is fully set up and can do blocking allocations */
6.     gfp_allowed_mask = __GFP_BITS_MASK;
7.
8.     /* init can allocate pages on any node */
9.     set_mems_allowed(node_states[N_HIGH_MEMORY]);
10.    /* init can run on any cpu. */
11.    set_cpus_allowed_ptr(current, cpu_all_mask);
12.    cad_pid = task_pid(current);
13.    smp_prepare_cpus(setup_max_cpus);
14.    do_pre_smp_initcalls();
15.    lockup_detector_init();
16.    smp_init();
17.    sched_init_smp();
18.    //以上代码是在SMP系统做准备，激活所有CPU，并开始SMP系统的调度
19.
20.    /*do_basic_setup函数主要是初始化设备驱动，完成其他驱动程序（直接编译进内核的模块）的初始化。内核中大部分的
    启动数据输出（都是各设备的驱动模块输出）都是这里产生的*/
21.    do_basic_setup();
22.    /* Open the /dev/console on the rootfs, this should never fail */
23.    if (sys_open((const char __user *) "/dev/console", O_RDWR, 0) < 0)
24.        printk(KERN_WARNING "Warning: unable to open an initial console.\n");
25.    (void) sys_dup(0);
26.    (void) sys_dup(0);
27.    /*
28.     复制两次标准输入（0）的文件描述符（它是上面打开的/dev/console，也就是系统控制台）：
29.     一个作为标准输出（1）
30.     一个作为标准出错（2）
31.     * check if there is an early userspace init. If yes, let it do all
32.     * the work
33.     */
34.
35.    /*检查是否有早期用户空间的init程序。如果有，让其执行*/
36.    if (!ramdisk_execute_command)
37.        ramdisk_execute_command = "/init";
38.
39.    if (sys_access((const char __user *) ramdisk_execute_command, 0) != 0) {
40.        ramdisk_execute_command = NULL;
41.        prepare_namespace();
42.    }
43.
44.    /*
45.     * Ok, we have completed the initial bootup, and
46.     * we're essentially up and running. Get rid of the
47.     * initmem segments and start the user-mode stuff..
48.     */
49.    printk(KERN_NOTICE "47---");
50.
51.    init_post();//在内核init线程的最后执行了init_post函数，在这个函数中真正启动了用户空间进程init
52.    return 0;
53. }
```

```
[cpp]
1. static noinline int init_post(void)
2. {
3.     /* need to finish all async __init code before freeing the memory */
4.     /* 在释放内存前，必须完成所有的异步 __init 代码 */
5.     async_synchronize_full();
6.     free_initmem();//释放所有init.* 段中的内存。
7.
8.     mark_rodata_ro();//通过修改页表，保证只读数据段为只读属性。大部分构架为空函数
9.     system_state = SYSTEM_RUNNING;//设置系统状态为运行状态
10.
11.     numa_default_policy();//设定NUMA系统的内存访问策略为默认
12.
13.     current->signal->flags |= SIGNAL_UNKILLABLE;//设置当前进程（init）为不可以杀进程（忽略致命的信号）
14.     flush_delayed_fput();
15.
16.     if (ramdisk_execute_command) { //如果ramdisk_execute_command有指定的init程序，就执行它
17.         run_init_process(ramdisk_execute_command);
18.         printk(KERN_WARNING "Failed to execute %s\n",
19.                ramdisk_execute_command);
20.     }
21.
22.     /*
23.     * We try each of these until one succeeds.
24.     * The Bourne shell can be used instead of init if we are
25.     * trying to recover a really broken machine.
26.     我们尝试以下的每个函数，直到函数成功执行。
27.     如果我们试图修复一个真正有问题的设备，
28.     Bourne shell 可以替代init进程。
29.     */
30.
31.     if (execute_command) {
32.         run_init_process(execute_command);
33.         printk(KERN_WARNING "Failed to execute %s. Attempting "
34.                "defaults...\n", execute_command);
35.     }
36.     run_init_process("/sbin/init");
37.     run_init_process("/etc/init");
38.     run_init_process("/bin/init");
39.     run_init_process("/bin/sh");
40.
41.     panic("No init found. Try passing init= option to kernel. "
42.           "See Linux Documentation/init.txt for guidance.");
43. }
```



qing_ping

关注

原创	粉丝	喜欢	评论
30	31	1	1

等级：[博客](#) 访问量：3万+
积分：650 排名：7万+



他的最新文章

更多文章

- Linux设备模型（四）class
- Linux设备模型（三）platform
- Linux设备模型（二）上层容器
- linux 设备模型(一)对象层
- Linux中断子系统-中断接口

文章分类

linux开发	2篇
linux 驱动学习	3篇
linux源码学习	14篇

文章存档

2014年1月	2篇
2013年12月	12篇
2013年11月	1篇
2013年10月	4篇
2013年9月	9篇
2012年2月	1篇

展开

他的热门文章

- linux 3.6 启动源码分析(五) kernel_init进程 3057
- linux 3.6 启动源码分析(二) start_kernel 2909
- linux 3.6 启动源码分析(七) do_initcalls 2454
- linux 3.6 启动源码分析(一) 2078
- linux 3.6 启动源码分析(三) setup_arch 1893
- 嵌入式linux 运行期间升级u-boot，kernel和文件系统 1792
- linux 3.6 启动源码分析(六) do_basic_set up 1681
- linux下读写u-boot环境变量 1354
- Linux中断子系统-中断初始化 1329
- linux 3.6 启动源码分析(四) rest_init 1307

至此，内核的初始化结束，正式进入了用户空间的初始化过程至此，内核的初始化结束，正式进入了用户空间的初始化过程至此，内核的初始化结束，正式进入了用户空间的初始化过程，在kernel_init线程中调用的do_basic_setup()函数会去初始化设备驱动，完成其他驱动程序（直接编译进内核的模块）的初始化。内核中大部分的启动数据输出（都是各设备的驱动模块输出）都是这里产生的。是我们驱动工程师需要重点关注的函数。



0



严禁讨论涉及中国之军/政相关话题，违者会被禁言、封号！

linux kernel_init



chenliang0224 2017年12月10日 23:45 46

前言：内核在启动用户空间程序时会创建两个线程，kthread() 和 kernel_init()线程，在前一篇介绍了kthread()线程 点击打开链接，本文不在赘述，这里主要是对kernel_in...

Linux下1号进程的前世(kernel_init)今生(init进程)----Linux进程的管理与调度（六）

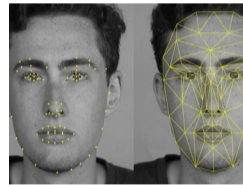
日期 内核版本 架构 作者 GitHub CSDN 2016-05-29 Linux-4.5 X86 & arm gatieme LinuxDeviceDrive...



gatieme 2016年05月29日 17:01 11061

从小白到人工智能工程师，我只花了4个月！

这一份学习路线希望能够帮到你们！



内核启动阶段kernel_init (init) 进程分析



qq_28992301 2016年08月02日 14:17 1768

在kernel进入c语言阶段后，会开始执行start_kernel函数，它负责进行kernel正式运行前各个功能的初始化：打印了一些信息、内核工作需要的模块的初始化被依次调用（譬如内存管理、调度系统、...

Kernel启动流程源码解析 7 rest_init()



xichangbao 2016年10月26日 21:03 1032

— rest_init 1.0 rest_init 定义在init/main.c中 static noinline void __init_refok rest_init(v...

start_kernel到init进程启动的过程



u010521171 2016年03月11日 15:06 1322

杨明辉+ 原创作品转载请注明出处 + 《Linux内核分析》MOOC课程http://mooc.study.163.com/course/USTC-1000029000 " 一、实验过程 1.进入实...

码农不会英语怎么行？英语文档都看不懂！

软件工程出身的英语老师，教你用数学公式读懂天下英文→



linux kernel_init



jake9602 2017年07月13日 13:46 223

Kernel:asm linkage __visible void __init start_kernel(void) { trap_init(); sched_in...

linux启动流程（从start_kernel中的rest_init函数到init进程（1））

linux启动流程（从start_kernel中的rest_init函数到init进程（1）） 在init/main.c文件中有个函数叫start_kernel，它是用来启动内核的主函数，我...



zhongyhc 2013年04月23日 23:35 1892

Linux init程序分析



zhoudaxia 2011年08月07日 14:44 16535

1、init程序剖析 init进程是内核引导过程完成时创建的第一个进程。Linux使用了init进程来对组成Linux的服务和应用程序进行初始化。 当 init 进程启动时（使用传...

Linux下init进程源码分析

2008年01月02日 08:25 547KB

下载



linux 3.6 启动源码分析(二) start_kernel



qing_ping 2013年12月16日 13:38 2910

在构架相关的汇编代码运行完之后，程序跳入了构架无关的内核C语言代码：init/main.c中的start_kernel函数，在这个函数中Linux内核开始真正进入初始化阶段，进行一系列与内核相关的初...

linux 3.6 启动源码分析(四) rest_init



kunkliu 2017年09月18日 09:32 267

转载地址：http://blog.csdn.net/qing_ping/article/details/17351933在内核初始化函数start_kernel执行到最后，就是调用rest_init函...

开发一个app多少钱



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

网上商城系统

电子商务网站建设

百度广告



0



linux 3.6 启动源码分析(一)



qing_ping

2013年12月16日 13:02

2080

作为需要和硬件打交道的工程师来说，比较关注的是驱动和CPU初始化这一块。所以我沿着启动的路线，重点学习一下和硬件相关的代码。就从linux解压的入口说起。学习阶段，基本是参考大神文章<http://bl...>

从start_kernel到init进程启动 《Linux内核分析》笔记

从start_kernel到init进程启动 《Linux内核分析》笔记GDB的使用



iSerendipity

2017年03月12日 23:20

915

在进入GDB调试前，首先掌握GDB最常用的命令，以方便完成跟踪。 b[reak] lineNumber : 加断点 s[...]

linux下一些特殊进程



hkh1_235

2017年11月28日 11:45

108

linux下 init kthreadd migration ksofirqd watchdogd events等进程介绍！

跟踪分析Linux内核的启动过程（start_kernel到init进程启动）

SA16225055 冯金明 原创作品转载请注明出处《Linux内核分析》MOOC课程<http://mooc.study.163.com/course/USTC-1000029000> 实验内容 1...



Kimi_ming

2017年03月07日 16:51

220

分析Linux内核启动过程：从start_kernel到init

郑德伦 原创作品转载请注明出处《Linux内核分析》MOOC课程 <http://mooc.study.163.com/course/USTC-1000029000> STEP



a363344923

2015年03月17日 20:05

7257

1：在自己的linux...

还为你的代码被反编译而头疼？

无需手动加密Assembly.DLL代码，自动编译mono，防止反编译



内核初始化 - 从start_kernel到init



wh8_2011

2016年01月30日 21:33

782

如图所示，内核的初始化过程由start_kernel函数开始，至第一个用户进程init结束，调用了一系列的初始化函数对所有的内核组件进行初始化。其中，start_kernel、rest_init、ke...

Linux内核很吊之 module_init解析（下）



Richard_LiuJH

2015年07月04日 22:00

3189

分析module_init 的作用。和module_init注册函数过程，以及注册函数被执行的过程。...

linux 3.6 启动源码分析(四) rest_init



qing_ping

2013年12月16日 14:25

1310

在内核初始化函数start_kernel执行到最后，就是调用rest_init函数，这个函数的主要使命就是创建并启动内核线程init。这个函数虽然意思为剩下的初始化，但是这个“剩下”的可是内容颇多，下...

linux 3.6 启动源码分析(七) do_initcalls



qing_ping

2013年12月18日 14:16

2457

do_initcalls()将按顺序从由__initcall_start开始，到__initcall_end结束的section中以函数指针的形式取出这些编译到内核的驱动模块中初始化函数起始地址，来依...