

闪客新系列！你管这破玩意叫操作系统源码

Original 闪客sun 低并发编程 2021-11-08 08:30

收录于话题

#你管这破玩意叫操作系统源码

20个 >

核心信息提炼到开头，以节约大家的时间。

要干嘛：写一个系列

啥目的：带大家把 Linux 0.11 核心代码与操作系统的设计思想啃下来

叫啥名：你管这破玩意叫操作系统源码 — 像小说一样品读 Linux 0.11 核心代码

发文时间：每周一和每周四

预计章节：60 回

互动方式：微信群（文末有加入方式）

系列整体布局：

第一部分：进入内核前的苦力活

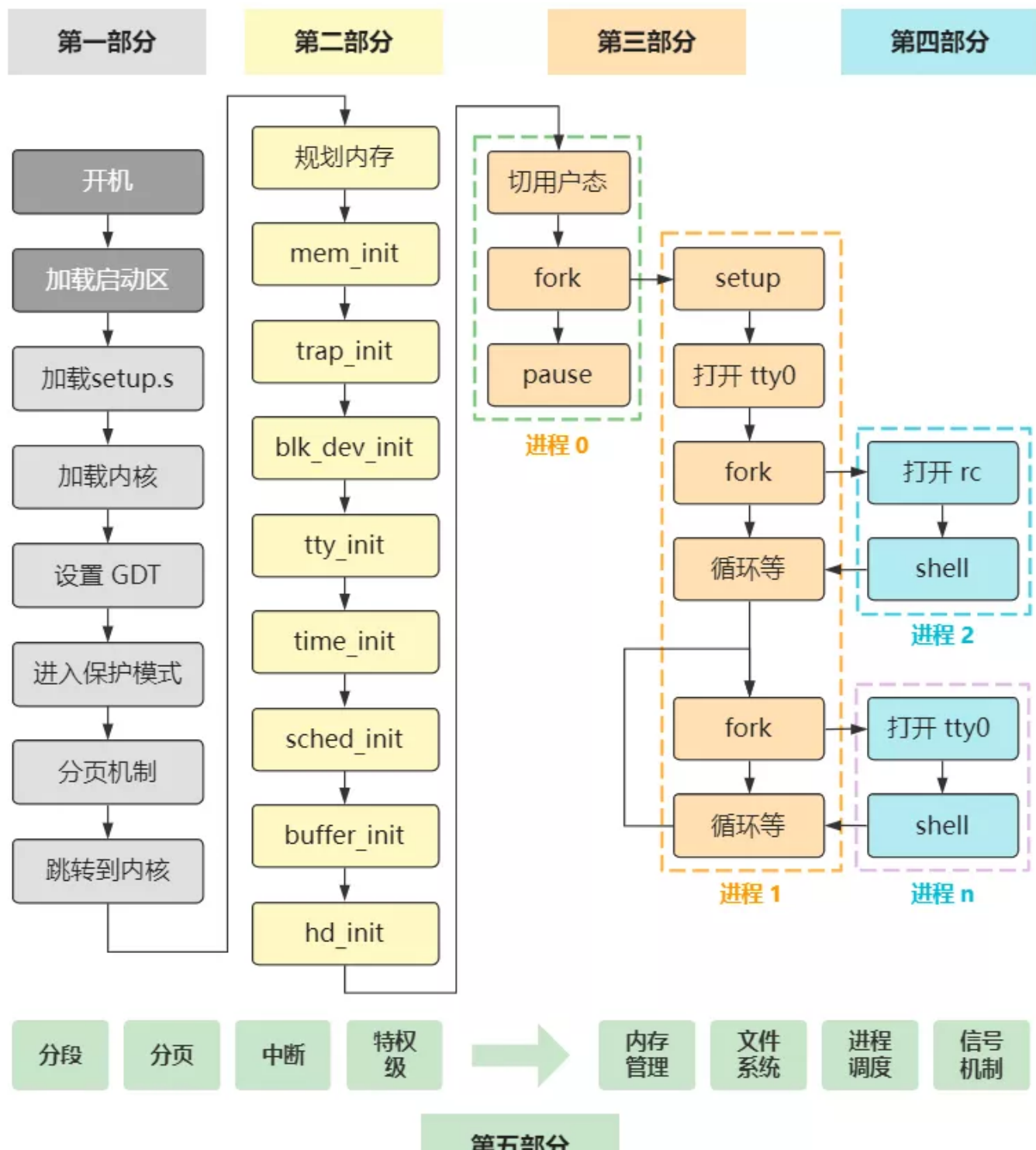
第二部分：大战前期的初始化工作

第三部分：一个新进程的诞生

第四部分：shell 程序的到来

第五部分：从一个命令的执行看操作系统各模块的运作

第六部分：操作系统哲学与思想



OK，以上就是要说的重要的事，以下是啰嗦的部分。

----- 华丽的分割线 -----

每个程序员都有一个**操作系统梦**，而操作系统也是每个程序员的心结。

很粗糙地了解一点操作系统知识，一知半解的，已经无法满足当下程序员的口味了。但要说深入剖析操作系统，又是大部分程序员都很惶恐的一件事。那如果是要读一遍操作系统源码，那简直跟要了命一样。

其实，操作系统的源码并没有那么可怕，可为什么即便是 Linux 0.11 这种代码量最少的版本，仍然令很多人望而却步呢？

为什么望而却步

其实，很多优秀的操作系统书籍都是以 Linux 0.11 这个经典版本为研究对象进行讲解的，比如《**Linux 内核设计的艺术**》、《**Linux 内核完全注释**》等。但我们想一下，当我们读一本小说时，为什么即便是非常大部头的小说，也能酣畅淋漓从头读到尾？

先不直接回答这个问题，我们看一下《**天龙八部**》的最开头。

青光闪动，一柄青钢剑倏地刺出，指向中年汉子左肩，使剑少年不待剑招用老，腕抖剑斜，剑锋已削向那汉子右颈。那中年汉子...

记住这个感觉没？我们再看一下《**Linux 内核设计的艺术**》的最开头。

对于操作系统而言，稳定且可靠地运行是最重要的。现行技术方案是将用户进程与用户进程之间、用户进程与操作系统之间进行分离，操作系统可以管理用户进程，但是用户进程之间不能相互干预 ...

好了，不用读下去了，这句话看似高瞻远瞩地从宏观上帮我们梳理操作系统体系结构，但对于尚不了解操作系统的人来说，完全不知道它在说什么，只有劝退的作用。

虽然说思想很重要，但你在没有任何细节做积累时去强行进行思想的拔高，是拔不上去的，还不如一直保持一张白纸的状态。

反观《天龙八部》的开头，连人物的名字都没有，更别说梳理整个体系结构了，直接上来一个精彩镜头，让你迅速进入故事情节。

可是读完整部小说的读者，无一不对里面的人物如数家珍，对大理的风光仿佛亲眼看见了一般，对宋辽矛盾的激烈感同身受。

为什么会这样呢？因为一切的爱恨情仇和民族矛盾，都是我们通过一个个人物和事件的刻画，感悟出来的。**只有自己感悟出来的知识和靠自己总结出来的结论，才真正属于自己。**

而那些一上来就试图把整个脉络给你梳理清楚的尝试，对于新手来说无一不是徒劳，即便是死记硬背记住了，也终究不是属于自己的知识，无法感同身受。

我学习操作系统的过程中，也有这样的感觉。

我曾一次次试图从一个上帝视角来看操作系统的知识体系，从宏观层面跟着大部头书籍梳理操作系统的整体逻辑，我发现无一不是以失败告终。

而当我放下包袱，**用读一本小说的心态来去阅读 Linux 源代码时**，我发现，我从来没有去想着梳理出什么体系，但不知道从哪一行代码开始，整个操作系统的体系结构已经较为清晰地出现在我面前了，竟是那么的**不知不觉**。

而且我也清晰地知道，这样的体系是怎么一步步从第一行代码开始，逐步建立起来的。

虽然我还没有触类旁通，真正理解操作系统的哲学与设计思想，也不能凭自己的想法写出一个新的操作系统来，但最初的这道坎我算是过去了。

所以我想把这些梳理一下，**分享给大家**。同时也能更好地让自己巩固细节，最终真正理解操作系统的思想，达到触类旁通。

如何分享呢？

打算写个新系列

我并不是说《Linux 内核设计的艺术》这本书不好，而是想着能不能也以**写小说一样的心态和方式**，来给大家从头到尾讲述这部 Linux 0.11 的源码。

我打算把这个系列叫做

你管这破玩意叫操作系统源码 —— 像小说一样品读 Linux 0.11 核心代码

在这里面，我不会经常把操作系统的体系架构挂在嘴边，时时刻刻强行塞到你的脑子里，而是**通过一行一行代码逐渐带入情节**，最终让你不知不觉地发现，原来整个操作系统的体系就这样一点一点建立起来了。

本系列的每一章内容都很短，千万不要有心里负担，而正是这些简单的事情联系到一起，就构成了整个操作系统的复杂的设计。

所以也导致了，单独看任何一章都不会有显著的收获，但如果整个系列都能跟下来，并且每一章的内容都能做出思考，把不懂的地方及时解决，我保证你会对操作系统有一个全新的，深入到细节的认识。

希望你跟我走完这个系列，也能发出一句感叹，**原来操作系统源码不过如此**，就是这么个**破玩意**而已！

但同时，你也要有耐心，你不要总想着，读到这了我怎么还是没觉得自己懂操作系统了呢？怎么还是没讲多进程如何调度呢？**记住，享受当下，当下你学的每一个看似没啥用的知识，都是后面豁然开朗这种感觉的基石。**

等等，我记得闪客之前也说要写个自制操作系统系列呢？不过好像...

万一中途又鸽了怎么办

话说回来，我的老读者有点惨，之前总是看我信誓旦旦说要写操作系统的系列，但无一例外都是中途失败了。

博客园上写过，公众号上也写过，而公众号里的好多读者就是被我的自制操作系统系列骗进来的，然后写到第五章我就鸽了，实在太过分了！

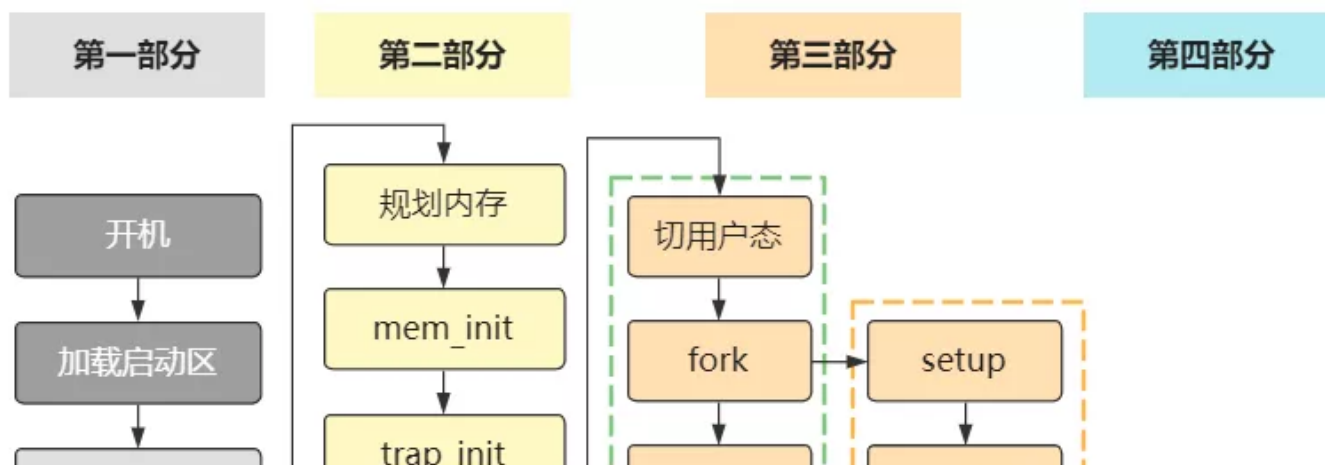
所以接下来自然就有一个大家都关心的问题，就是这次会不会又鸽呢？

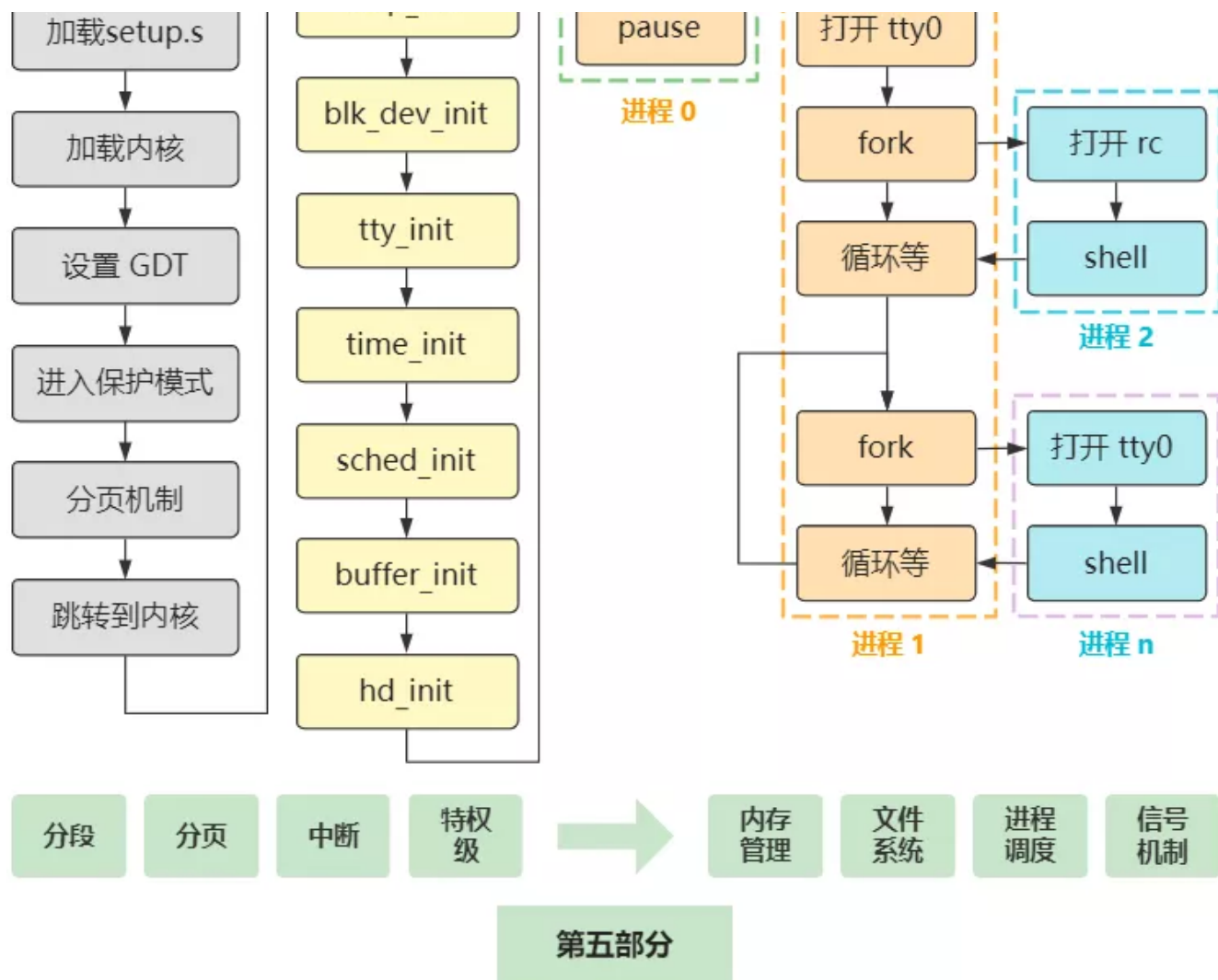
我可以肯定的是，这次绝对绝对绝对绝对绝对绝对不会中途鸽了！**因为我此时此刻已经把全部的大纲以及文章的前 33 章的内容写好了，就藏在我的石墨文档里。后面的内容我打算根据前面的反馈和问答，不断做出调整。**

文件名 ^
▸ 第一部分 进入内核前的苦力活
▸ 第二部分 大战前期的初始化工作
▸ 第三部分 一个新进程的诞生
▸ 第四部分 shell 程序的到来
▸ 第五部分 从一个命令的执行看操作系统各模块的运作
▸ 第六部分 操作系统哲学与思想

没错，总共我计划分**六大部分**，**前四部分**你会看到从开机一直到操作系统的最后一行代码的全部主流程，**第五部分**将通过一个命令的执行将操作系统各个模块的运作方式串一遍，而**第六部分**会做一个思想的拔高，这也是我自己给自己挖的一个坑和挑战。

而六大部分对应到 Linux 0.11 源码从开机到 shell 的流程图，是这样的。





其实本可以早早就开始边写边发布，但想着要是这次中途又鸽了读者，着实有点不好意思。所以就冒着中途写不下去的风险，先把大部分文章提前写好。

但我也给自己留了个后路，就是只提前写了 33 章，也够大家看好几个月了，要是大家不捧场，我后面就不更了，哼（傲娇表情.gif）！

提前写好有个好处，就是写到后面的时候，发现前面有些地方可以提前做个引子，这样整个系列就更完整了，可以**前后呼应**，可读性和收获也会大大增加。

如何互动

虽然已经提前写好，但中途还是要不断接受读者反馈和答疑，以便更好修正文章的内容，并且做一些知识点的补充，这也是我对自己要求和考验。

所以建立**微信群**增加互动性。

在这里你可以不断对文章内容提出**反馈意见**，以及和不同读者进行**心得交流**，以及进行**催更呐喊**。不要觉得你的意见无法左右这个系列，要知道，这个系列的名称就是低并发编程读者群的读者们群体的智慧想出来的。

我觉得这也是公众号里更新系列的一个优势，就是与读者的距离更近，更方便随时讨论和交流，互相促进成长。

加入方式是加我微信好友，备注

os-昵称-其他信息

比如

os-闪客-Java

我会将所有备注为本格式的好友邀请至微信群。



一定要按备注要求来哦~

----- 华丽的分割线 -----

好了！本篇文章就当做开篇词，今后不出意外至少**每周一周四**更新，每一个大部分结束后会看情况留一段时间集中消化与总结。

不过会以文章质量和准确性为主，不会带着问题强行按时更新的。

本系列**完全免费**，直到所有章节全部结束，所以你们的喜欢和传播就是对我最大的支持，可以**星标**我的公众号防止错过更新提醒。

公众号虽然与读者距离更近，但它却是个封闭的空间，平台和搜索引擎不会主动向外扩散，所以如果没有人主动帮忙传播，会一直限制在公众号粉丝的圈子里，越到后面看的人也会越少。

这也是很多公众号系列中途腰斩的因素之一，所以还是希望大家喜欢这个系列的话，可以多多帮忙传播，比如**朋友圈打打卡**，或者如果你也写博客的话，**文章里提提我**，都是可以帮到这个系列活下去的重要途径，在此多谢各位捧场啦！

同时我也会在 **GitHub** 上进行同步，因为公众号文章发了之后就无法修改，也没法进行整体调整。感兴趣的也可以点击[阅读原文](#)进入 GitHub。但我同步应该不会很及时，还是以公众号为主战场，毕竟时间和精力有限。

那就让我们一起期待吧！



低并发编程

战略上藐视技术，战术上重视技术

127篇原创内容



Official Account

收录于话题 [#你管这破玩意叫操作系统源码](#) 20

[下一篇 · 你管这破玩意叫操作系统源码 | 第一回 最开始的两行代码](#) >

[Read more](#)

People who liked this content also liked

Thanos 源码解析（五） - Receive Shipper

栋总侃技术



Spring Cloud Gateway源码解析之基本特性及核心概念

壹沅



智能指针最佳实践&源码分析

编码安全

