

Linux 内核剖析

概览

历史和体系结构分析

Linux 的简短历史



M. Jones

2010 年 9 月 20 日发布

Linux 内核的属性

Linux 内核的主要子系统

由于本文的目标是对 Linux 内核进行介绍并探索其体系结构和主要组件，因此首先回顾一下 Linux 内核的一些有用特性。次审视 Linux 内核的体系结构，最后介绍它的主要子系统。Linux 内核具有超过 600 万行的代码，请[使用指向其他内容的链接](#)进一步学习。

相关主题

Linux 的简短历史

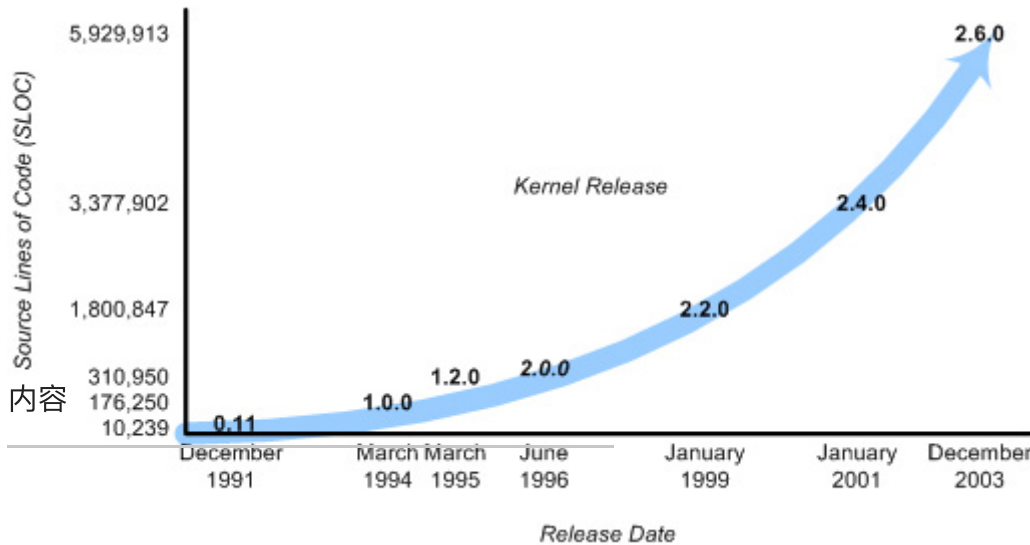
评论

尽管 Linux 绝对是最流行的开源操作系统，但是相对于其他操作系统的漫长历史来说，Linux 的历程，程序员是使用硬件语言在裸硬件上进行开发的。缺少操作系统就意味着在某个时间只有一个应用这些庞大而又昂贵的设备。早期的操作系统是在 20 世纪 50 年代开发的，用来提供简单的开发。General Motors Operating System (GMOS) 和 North American Aviation 为 IBM 709 开发的 FOR System (FMS)。

在 20 世纪 60 年代，MIT (Massachusetts Institute of Technology) 和一些公司为 GE-645 开发了 Multics (Multiplexed Information and Computing Service) 的实验性的操作系统。这个操作系统了 Multics，并在 1970 年开发了自己的名为 Unics 的操作系统。与这个操作系统一同诞生的是 C 的，然后它们使用 C 语言对操作系统进行了重写，使操作系统开发具有可移植性。

二十年后，Andrew Tanenbaum 创建了一个微内核版本的 UNIX®，名为 MINIX (代表 minimal UI 计算机上运行。这个开源操作系统在 20 世纪 90 年代激发了 Linus Torvalds 开发 Linux 的灵感 (请参

图 1. 主要 Linux 内核发行版简史

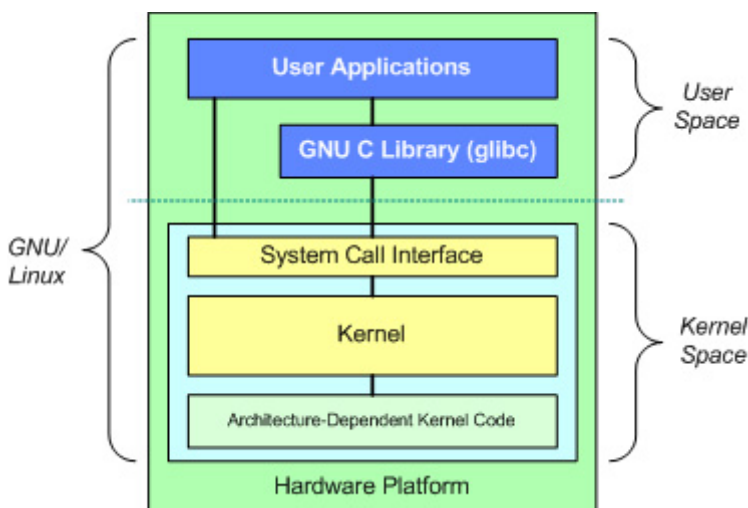


Linux 快速从一个个人项目进化成为一个全球数千人参与的开发项目。对于 Linux 来说，最为重要（General Public License）。在 GPL 保护之下，Linux 内核可以防止商业使用，并且它还从 GNU 项目（其源代码要比 Linux 内核大得多）的用户空间开发受益。这允许使用一些非常有用的应用程序，例如 GNU C Library）和各种 shell 支持。

Linux 内核简介

现在让我们从一个比较高的高度来审视一下 GNU/Linux 操作系统的体系结构。您可以从两个层次来展示。

图 2. GNU/Linux 操作系统的基本体系结构



最上面是用户（或应用程序）空间。这是用户应用程序执行的地方。用户空间之下是内核空间，Linux

GNU C Library（glibc）也在这里。它提供了连接内核的系统调用接口，还提供了在用户空间应用程序运行。这点非常重要，因为内核和用户空间的应用程序使用的是不同的保护地址空间。每个用户空间

空间，而内核则占用单独的地址空间。更多信息，请参看 [参考资料](#) 一节中的链接。

Linux 内核可以进一步划分成 3 层。最上面是系统调用接口，它实现了一些基本的功能，例如 `read` 是内核代码，可以更精确地定义为独立于体系结构的内核代码。这些代码是 Linux 所支持的所有其他代码之下是依赖于体系结构的代码，构成了通常称为 BSP（Board Support Package）的部分。处理器和特定于平台的代码。

Linux 内核的属性

在讨论大型而复杂的系统的体系结构时，可以从很多角度来审视系统。体系结构分析的一个目标是代码，这正是本文的目的。

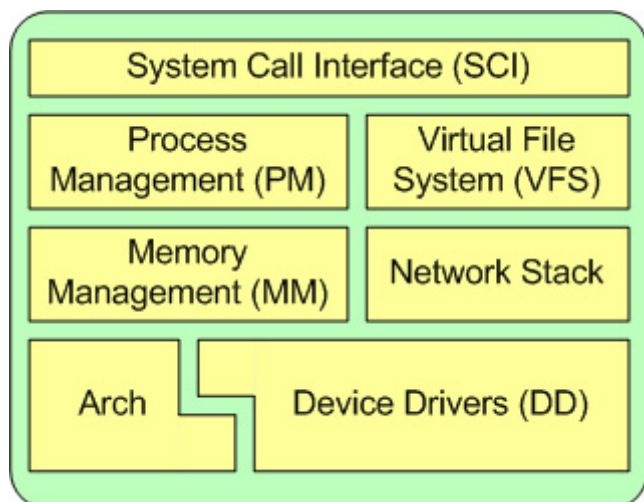
Linux 内核实现了很多重要的体系结构属性。在或高或低的层次上，内核被划分为多个子系统。Linux 为它会将所有这些基本服务都集成到内核中。这与微内核的体系结构不同，后者会提供一些基本的服务和进程管理，更具体的服务都是插入到微内核层中的。每种内核都有自己的优点，不过这里并不对

随着时间的流逝，Linux 内核在内存和 CPU 使用方面具有较高的效率，并且非常稳定。但是对于不同大小和复杂性的前提下，依然具有良好的可移植性。Linux 编译后可在大量处理器和具有不同体系结构上运行。一个例子是 Linux 可以在一个具有内存管理单元（MMU）的处理器上运行，也可以在那些不支持 MMU 的处理器上运行。Linux 内核的 uClinux 移植提供了对非 MMU 的支持。更详细信息请参看 [参考资料](#) 一节的内容。

Linux 内核的主要子系统

现在使用图 3 中的分类说明 Linux 内核的主要组件。

图 3. Linux 内核的一个体系结构透视图



系统调用接口

SCI 层提供了某些机制执行从用户空间到内核的函数调用。正如前面讨论的一样，这个接口依赖于处理器家族内也是如此。SCI 实际上是一个非常有用的函数调用多路复用和多路分解服务。在 `./linux/` 中实现，并在 `./linux/arch` 中找到依赖于体系结构的部分。有关这个组件的更详细信息可以在 [参考资料](#)

内容 进程管理

进程管理的重点是进程的执行。在内核中，这些进程称为线程，代表了单独的处理器虚拟化（线程和处理器）。在用户空间，通常使用进程这个术语，不过 Linux 实现并没有区分这两个概念（进程和线程）。应用程序编程接口（API）来创建一个新进程（`fork`、`exec` 或 `Portable Operating System Interface`（`kill`、`exit`），并在它们之间进行通信和同步（`signal` 或者 `POSIX` 机制）。

进程管理还包括处理活动进程之间共享 CPU 的需求。内核实现了一种新型的调度算法，不管有多少方法都可以在固定时间内进行操作。这种算法就称为 $O(1)$ 调度程序，这个名字就表示它调度多个线程所使用的时间是相同的。 $O(1)$ 调度程序也可以支持多处理器（称为对称多处理器或 `SMP`）。您可管理的源代码，在 `./linux/arch` 中可以找到依赖于体系结构的源代码。在 [参考资料](#) 一节中可以了解

内存管理

内核所管理的另外一个重要资源是内存。为了提高效率，如果由硬件管理虚拟内存，内存是按照页（对于大部分体系结构来说都是 4KB）。Linux 包括了管理可用内存的方式，以及物理和虚拟映射

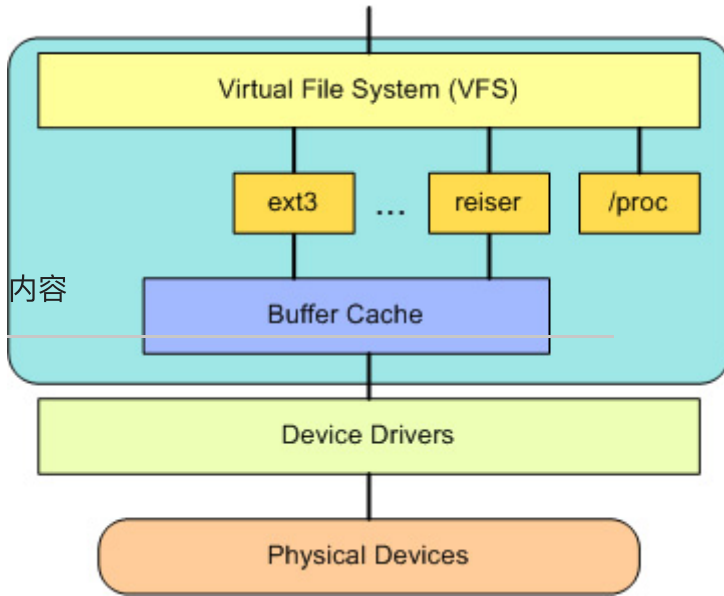
不过内存管理要管理的可不止 4KB 缓冲区。Linux 提供了对 4KB 缓冲区的抽象，例如 `slab` 分配器以缓冲区为基数，然后从中分配结构，并跟踪内存页使用情况，比如哪些内存页是满的，哪些页面没有。这样就允许该模式根据系统需要来动态调整内存使用。

为了支持多个用户使用内存，有时会出现可用内存被消耗光的情况。由于这个原因，页面可以移出内存，称为交换，因为页面会被从内存交换到硬盘上。内存管理的源代码可以在 `./linux/mm` 中找到。

虚拟文件系统

虚拟文件系统（VFS）是 Linux 内核中非常有用的一个方面，因为它为文件系统提供了一个通用的接口。所支持的文件系统之间提供了一个交换层（请参看图 4）。

图 4. VFS 在用户和文件系统之间提供了一个交换层



在 VFS 上面，是对诸如 open、close、read 和 write 之类的函数的一个通用 API 抽象。在 VFS 下层函数的实现方式。它们是给定文件系统（超过 50 个）的插件。文件系统的源代码可以在 `./linux`

文件系统层之下是缓冲区缓存，它为文件系统层提供了一个通用函数集（与具体文件系统无关）。段时间（或者随即预先读取数据以便在需要是就可用）优化了对物理设备的访问。缓冲区缓存之特定物理设备的接口。

网络堆栈

网络堆栈在设计上遵循模拟协议本身的分层体系结构。回想一下，Internet Protocol (IP) 是传输层 (TCP) 下面的核心网络层协议。TCP 上面是 socket 层，它是通过 SCI 进行调用的。

socket 层是网络子系统的标准 API，它为各种网络协议提供了一个用户接口。从原始帧访问到 IP 和 TCP 和 User Datagram Protocol (UDP)，socket 层提供了一种标准化的方法来管理连接，并在各网络源代码可以在 `./linux/net` 中找到。

设备驱动程序

Linux 内核中有大量代码都在设备驱动程序中，它们能够运转特定的硬件设备。Linux 源码树提供录又进一步划分为各种支持设备，例如 Bluetooth、I2C、serial 等。设备驱动程序的代码可以在。

依赖体系结构的代码

尽管 Linux 很大程度上独立于所运行的体系结构，但是有些元素则必须考虑体系结构才能正常操作。子目录定义了内核源代码中依赖于体系结构的部分，其中包含了各种特定于体系结构的子目录（对于典型的桌面系统来说，使用的是 i386 目录。每个体系结构子目录都包含了很多其他子目录，每个子目录方面，例如引导、内核、内存管理等。这些依赖体系结构的代码可以在 `./linux/arch` 中找到。

内容

Linux 内核的一些有用特性

如果 Linux 内核的可移植性和效率还不够好，Linux 还提供了其他一些特性，它们无法划分到上面

作为一个生产操作系统和开源软件，Linux 是测试新协议及其增强的良好平台。Linux 支持大量网络以及高速网络的扩展（大于 1 Gigabit Ethernet [GbE] 和 10 GbE）。Linux 也可以支持诸如流控制协议，它提供了很多比 TCP 更高级的特性（是传输层协议的接替者）。

Linux 还是一个动态内核，支持动态添加或删除软件组件。被称为动态可加载内核模块，它们可以在设备需要这个模块）或在任何时候由用户插入。

Linux 最新的一个增强是可以用作其他操作系统的操作系统（称为系统管理程序）。最近，对内核虚拟机（KVM）。这个修改为用户空间启用了一个新的接口，它可以允许其他操作系统在启用了 Linux 的其他实例之外，Microsoft® Windows® 也可以进行虚拟化。惟一的限制是底层处理器必须信息请参看 [参考资料](#) 一节的内容。

结束语

本文对 Linux 内核体系结构及其特性和功能进行了简要介绍。有关内核的详细内容，可以参考每个 Documentation 目录。请一定查看本文末尾的 [参考资料](#) 一节，了解有关本文中所讨论主题的更详细

相关主题

- 您可以参阅本文在 developerWorks 全球站点上的 [英文原文](#)。
- [GNU 站点](#) 介绍了 GNU GPL，它涵盖了 Linux 内核及其附带的大量有用的应用程序。另外还介绍了 Lesser GPL（LGPL）。

- [UNIX](#)、[MINIX](#) 和 [Linux](#) 在 Wikipedia 上都有介绍，另外还详细介绍了操作系统系列。
- [GNU C Library](#)，即 glibc，是标准 C 库的实现。它用于 GNU/Linux 操作系统，也可用于 [GNU/](#)
- [uClinux](#) 是 Linux 内核的一个移植，它可以在一些缺少 MMU 的系统上执行。这允许 Linux 内核例如 PalmPilot Personal Digital Assistants (PDAs) 上使用的 Motorola DragonBall 4
- “[使用 Linux 系统调用的内核命令](#)” (developerWorks, 2007 年 3 月) 对 SCI 进行了介绍，这层，具有 glibc 的用户空间支持，可以在用户空间和内核之间启用函数调用。

内容“[Linux 调度器内幕](#)” (developerWorks, 2006 年 6 月) 介绍了 Linux 2.6 中引入的新 O(1) 调度方法，可以扩展到很大数量的进程（线程），并且可以利用 SMP 系统。

- “[使用 /proc 文件系统来访问 Linux 内核的内容](#)” (developerWorks, 2006 年 3 月) 介绍了 /proc 文件系统，为用户空间的应用程序提供了一种全新的方法与内核进行通信。这篇文章展示了 /proc
- “[服务器诊所: 使虚拟文件系统工作](#)” (developerWorks, 2003 年 4 月) 深入介绍了 VFS 层，支持各种不同的文件系统。这个相同的接口也可以用于其他类型的设备，例如 socket。
- “[Linux 引导过程内幕](#)” (developerWorks, 2006 年 5 月) 介绍了 Linux 的引导过程，该过程不管是从硬盘、软盘、USB 内存条还是通过网络引导系统，该过程是相同的。
- “[Linux 初始 RAM 磁盘 \(initrd\) 概述](#)” (developerWorks, 2006 年 7 月) 介绍了初始 RAM 磁盘物理介质隔离开来。
- “[使用 SCTP 优化网络](#)” (developerWorks, 2006 年 2 月) 介绍了一种非常有用的网络协议：TCP 类似，但是增加了很多有用的特性，例如消息传递、多宿主和多流。如果您对网络协议是怎么样，都是一种非常好的操作系统。
- “[Linux slab 分配器详解](#)” (developerWorks, 2007 年 5 月) 介绍了 Linux 内存管理中最有用种机制源自于 SunOS，不过它在 Linux 内核中找到了合适的位置。
- “[虚拟 Linux](#)” (developerWorks, 2006 年 12 月) 介绍了 Linux 是如何充分利用具有虚拟化能力
- “[Linux 和对称多处理](#)” (developerWorks, 2007 年 3 月) 讨论了 Linux 怎样利用提供芯片级多
- “[探索 Linux 内核虚拟机](#)” (developerWorks, 2007 年 4 月) 介绍了最近引入内核的虚拟化技术其他虚拟化的操作系统的系统管理程序。
- 有关在用户空间对 Linux 进行编程的更多信息，请参考 Tim 撰写的 [GNU/Linux Application Pr](#)
- 在 [developerWorks Linux 专区](#) 中可找到适合 Linux 开发人员的更多资源，包括 [Linux 教程](#) 以及
- 利用可直接从 developerWorks 下载的 [IBM 试用软件](#) 在 Linux 上构建您的下一个开发项目。

评论

添加或订阅评论，请先[登录](#)或[注册](#)。

☐ 有新评论时提醒我

developerWorks

站点反馈

我要投稿

投稿指南

报告滥用

第三方提示

关注微博

加入

ISV 资源 (英语)

选择语言

English

中文

日本語

Русский

Português (Brasil)

Español

한글

技术文档库

订阅源

社区

dW 中国时事通讯

软件下载

[联系 IBM](#) [隐私条约](#) [使用条款](#) [信息无障碍选项](#) [反馈](#) [Cookie 首选项](#)
