04 | 震撼的Linux全景图：业界成熟的内核架构长什么样?

2021-05-17 LMOS

进入课程 >

讲述：陈晨

时长 17:08　大小 15.69M

你好，我是 LMOS。

什么？你想成为计算机黑客?

梦想坐在计算机前敲敲键盘，银行账号里的数字就会自己往上涨。拜托，估计明天你就该被警察逮捕了。真正的黑客是对计算机技术有近乎极致的追求，而不是干坏事。

下面我就带你认识这样一个计算机黑客，看看他是怎样创造出影响世界的 Linux，然后进一步了解一下 Linux 的内部结构。

同时，我也会带你看看 Windows NT 和 Darwin 的内部结构，三者形成对比，你能更好地了解它们之间的差异和共同点，这对我们后面写操作系统会很有帮助。

关于 Linus

Linus Benedict Torvalds，这个名字很长，下面简称 Linus，他 1969 年 12 月 28 日出生在芬兰的赫尔辛基市，并不是美国人。Linus 在赫尔辛基大学学的就是计算机，妻子还是空手道高手，一个“码林高手”和一个“武林高手”真的是绝配啊。

Linus 在小时候就对各种事情充满好奇，这点非常具有黑客精神，后来有了自己的计算机更是痴迷其中，开始自己控制计算机做一些事情，并深挖其背后的原理。就是这种黑客精神促使他后来写出了颠覆世界的软件——Linux，也因此登上了美国《时代》周刊。

你是否对很多垃圾软件感到愤慨，但自己又无法改变。Linus 就不一样，他为了方便访问大学服务器中的资源 ，而在自己的机器上写了一个文件系统和硬盘驱动，这样就可以把自己需要的资源下载到自己的机器中。

再后来，这成为了 Linux 的第一个版本。看看牛人之所以为牛人就是敢于对现有的规则说不，并勇于改变。

如果仅仅如此，那么也不会有后来的 Linux 内核。Linus 随后做了一个重要决定，他把这款操作系统雏形开源，并加入到自由软件运动，以 GPL 协议授权，允许用户自由复制或者改动程序代码，但用户必须公开自己的修改并传播。

无疑，正是 Linus 的这一重要决定使得 Linux 和他自己名声大振。短短几年时间，就已经聚集了成千上万的狂热分子，大家不计得失的为 Linux 添砖加瓦，很多程序员更是对 Linus 像神明一样顶礼膜拜。

Linux 内核

好了回到正题，回到 Linux。Linus 也不是什么神明，现有的 Linux，99.9% 的代码都不是 Linus 所写，而且他的代码，也不一定比你我的代码写得更好。

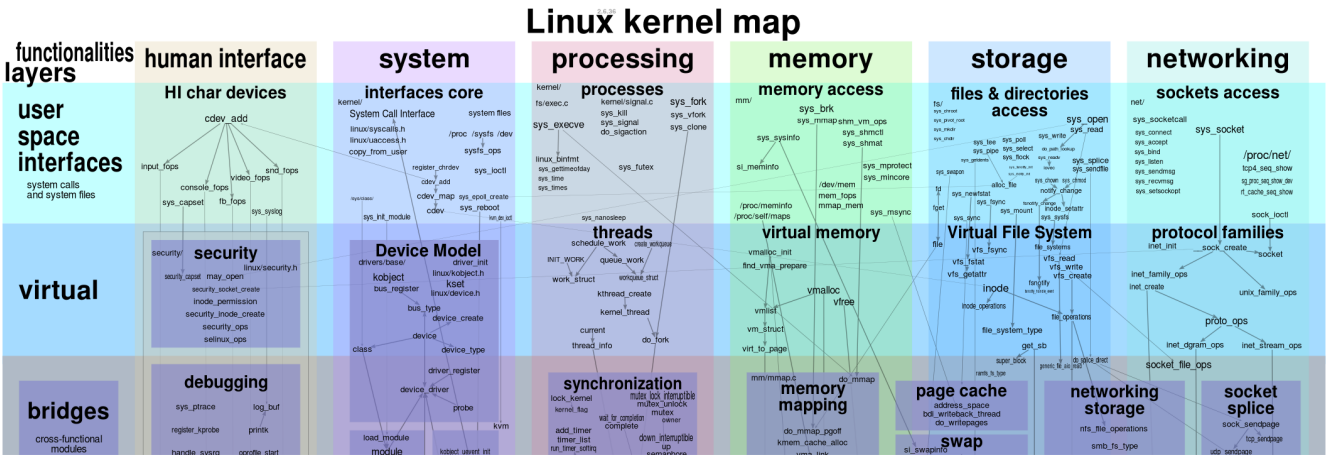
Linux，全称 GNU/Linux，是一套免费使用和自由传播的操作系统，支持类 UNIX、POSIX 标准接口，也支持多用户、多进程、多线程，可以在多 CPU 的机器上运行。由于互联网的发展，Linux 吸引了来自全世界各地软件爱好者、科技公司的支持，它已经从大型机到服务器蔓延至个人电脑、嵌入式系统等领域。

Linux 系统性能稳定且开源。在很多公司企业网络中被当作服务器来使用，这是 Linux 的一大亮点，也是它得以壮大的关键。

Linux 的基本思想是一切都是文件：每个文件都有确定的用途，包括用户数据、命令、配置参数、硬件设备等对于操作系统内核而言，都被视为各种类型的文件。Linux 支持多用户，各个用户对于自己的文件有自己特殊的权利，保证了各用户之间互不影响。多任务则是现代操作系统最重要的一个特点，Linux 可以使多个程序同时并独立地运行。

Linux 发展到今天，不是哪一个人能做到的，更不是一群计算机黑客能做到的，而是由很多世界级的顶尖科技公司联合开发，如 IBM、甲骨文、红帽、英特尔、微软，它们开发 Linux 并向 Linux 社区提供补丁，使 Linux 工作在它们的服务器上，向客户出售业务服务。

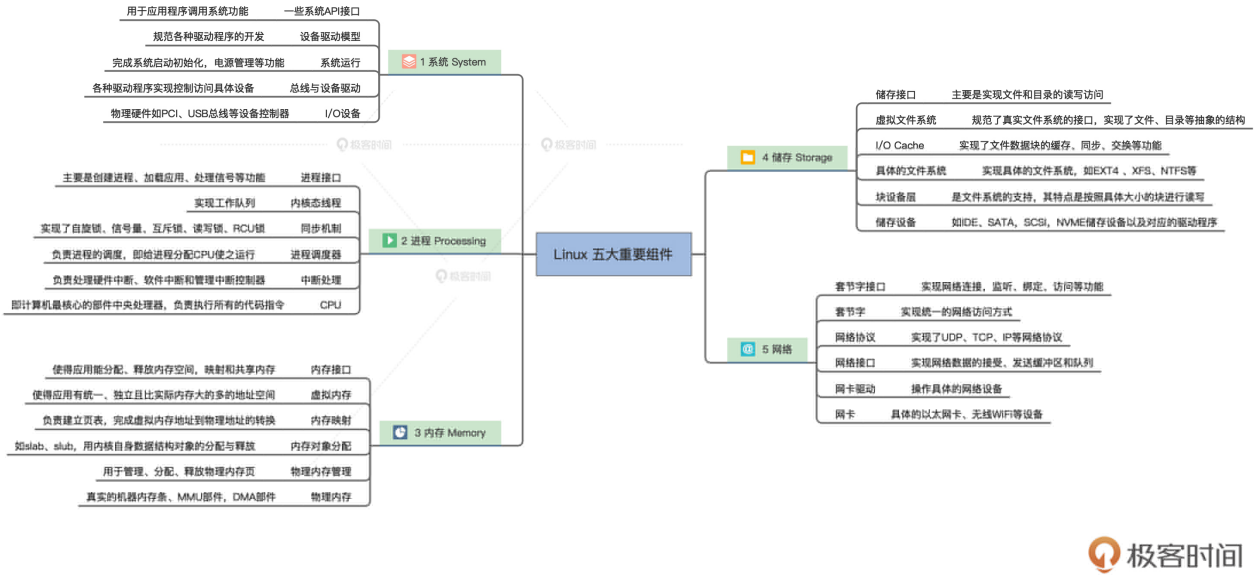
Linux 发展到今天其代码量近 2000 万行，可以用浩如烟海来形容，没人能在短时间内弄清楚。但是你也不用害怕，我们可以先看看 Linux 内部的全景图，从全局了解一下 Linux 的内部结构，如下图。



Linux内部的全景图

啊哈！是不是感觉壮观之后一阵头晕目眩，头晕目眩就对了，因为 Linux 太大了，别怕，下面我们来分解一下。但这里我要先解释一下，上图仍然不足于描述 Linux 的全部，只是展示了重要且显而易见的部分。

上图中大致分为**五大重要组件**，每个组件又分成许多模块从上到下贯穿各个层次，每个模块中有重要的函数和数据结构。具体每个模块的主要功能，我都给你列在了文稿里，你可以详细看看后面这张图。



不要着急，不要心慌，因为现在我们不需要搞清楚这些 Linux 模块的全部实现细节，只要在心里默念 Linux 的模块真多啊，大概有五大组件，有好几十个模块，每个模块主要完成什么功能就行了。

是不是松了口气，先定定神，然后我们就能发现 Linux 这么多模块挤在一起，之间的通信主要是函数调用，而且函数间的调用没有一定的层次关系，更加没有左右边界的限定。函数的调用路径是纵横交错的，从图中的线条可以得到印证。

继续深入思考你就会发现，这些纵横交错的路径上有一个函数出现了问题，就麻烦大了，它会波及到全部组件，导致整个系统崩溃。当然调试解决这个问题，也是相当困难的。同样，模块之间没有隔离，安全隐患也是巨大的。

当然，这种结构不是一无是处，它的性能极高，而性能是衡量操作系统的一个重要指标。这种结构就是传统的内核结构，也称为**宏内核架构**。

想要评判一个产品好不好，最直接的方法就是用相似的产品对比。你说 Linux 很好，但是什么为好呢？我说 Linux 很差，它又差在什么地方呢？

下面我们就拿出 Windows 和 macOS 进行对比，注意我们只是对比它们的内核架构。

Darwin-XNU 内核

我们先来看看 Darwin，Darwin 是由苹果公司在 2000 年开发的一个开放源代码的操作系统。

一个经久不衰的公司，必然有自己的核心竞争力，也许是商业策略，也许是技术产品，又或是这两者的结合。而作为苹果公司各种产品和强大的应用生态系统的支撑者——Darwin，更是公司核心竞争力中的核心。

苹果公司有台式计算机、笔记本、平板、手机，台式计算机、笔记本使用了 macOS 操作系统，平板和手机则使用了 iOS 操作系统。Darwin 作为 macOS 与 iOS 操作系统的核心，从技术实现角度说，它必然要支持 PowerPC、x86、ARM 架构的处理器。

Darwin 使用了一种微内核（Mach）和相应的固件来支持不同的处理器平台，并提供操作系统原始的基础服务，上层的功能性系统服务和工具则是整合了 BSD 系统所提供的。苹果公司还为其开发了大量的库、框架和服务，不过它们都工作在用户态且闭源。

下面我们先从整体看一下 Darwin 的架构。

Darwin架构图

什么？两套内核？惊不惊喜？由于我们是研究 Darwin 内核，所以上图中我们只需要关注内核 - 用户转换层以下的部分即可。显然它有两个内核层——**Mach 层与 BSD 层**。

Mach 内核是卡耐基梅隆大学开发的经典微内核，意在提供最基本的操作系统服务，从而达到高性能、安全、可扩展的目的，而 BSD 则是伯克利大学开发的类 UNIX 操作系统，提供一整套操作系统服务。

那为什么两套内核会同时存在呢？

MAC OS X（2011 年之前的称呼）的发展经过了不同时期，随着时代的进步，产品功能需求增加，单纯的 Mach 之上实现出现了性能瓶颈，但是为了兼容之前为 Mach 开发的应用和设备驱动，就保留了 Mach 内核，同时加入了 BSD 内核。

Mach 内核仍然提供十分简单的进程、线程、IPC 通信、虚拟内存设备驱动相关的功能服务，BSD 则提供强大的安全特性，完善的网络服务，各种文件系统的支持，同时对 Mach 的进程、线程、IPC、虚拟内核组件进行细化、扩展延伸。

那么应用如何使用 Darwin 系统的服务呢？应用会通过用户层的框架和库来请求 Darwin 系统的服务，即调用 Darwin 系统 API。

在调用 Darwin 系统 API 时，会传入一个 API 号码，用这个号码去索引 Mach 陷入中断服务表中的函数。此时，API 号码如果小于 0，则表明请求的是 Mach 内核的服务，API 号码如果大于 0，则表明请求的是 BSD 内核的服务，它提供一整套标准的 POSIX 接口。

就这样，Mach 和 BSD 就同时存在了。

Mach 中还有一个重要的组件 Libkern，它是一个库，提供了很多底层的操作函数，同时支持 C++ 运行环境。

依赖这个库的还有 IOKit，IOKit 管理所有的设备驱动和内核功能扩展模块。驱动程序开发人员则可以使用 C++ 面向对象的方式开发驱动，这个方式很优雅，你完全可以找一个成熟的驱动程序作为父类继承它，要特别实现某个功能就重载其中的函数，也可以同时继承其它驱动程序，这大大节省了内存，也大大降低了出现 BUG 的可能。

如果你要详细了解 Darwin 内核的话，可以自行阅读相应的代码。而在这里，你只要从全局认识一下它的结构就行了。

Windows NT 内核

接下来我们再看下 NT 内核。现代 Windows 的内核就是 NT，我们不妨先看看 NT 的历史。

如果你是 90 后，大概没有接触过 MS-DOS，它的交互方式是你在键盘上输入相应的功能命令，它完成相应的功能后给用户返回相应的操作信息，没有图形界面。

在 MS-DOS 内核的实现上，也没有应用现代硬件的保护机制，这导致后来微软基于它开发的图形界面的操作系统，如 Windows 3.1、Windows95/98/ME，极其不稳定，且容易死机。

加上类 UNIX 操作系统在互联网领域大行其道，所以微软急需一款全新的操作系统来与之竞争。所以，Windows NT 诞生了。

Windows NT 是微软于 1993 年推出的面向工作站、网络服务器和大型计算机的网络操作系统，也可做 PC 操作系统。它是一款全新从零开始开发的新操作系统，并应用了现代硬件的所有特性，“NT”所指的便是“新技术”（New Technology）。

而普通用户第一次接触基于 NT 内核的 Windows 是 Windows 2000，一开始用户其实是不愿意接受的，因为 Windows 2000 对用户的硬件和应用存在兼容性问题。

随着硬件厂商和应用厂商对程序的升级，这个兼容性问题被缓解了，加之 Windows 2000 的高性能、高稳定性、高安全性，用户很快便接受了这个操作系统。这可以从 Windows 2000 的迭代者 Windows XP 的巨大成功，得到验证。

现在，NT 内核在设计上层次非常清晰明了，各组件之间界限耦合程度很低。下面我们就来看看 NT 内核架构图，了解一下 NT 内核是如何“庄严宏伟”。如下图：

NT内核架构图

这样看 NT 内核架构，是不是就清晰了很多？但这并不是我画图画得清晰，事实上的 NT 确实如此。

这里我要提示一下，上图中我们只关注内核模式下的东西，也就是传统意义上的内核。

当然微软自己在 HAL 层上是定义了一个小内核，小内核之下是硬件抽象层 HAL，这个 HAL 存在的好处是：不同的硬件平台只要提供对应的 HAL 就可以移植系统了。小内核之上是各种内核组件，微软称之为内核执行体，它们完成进程、内存、配置、I/O 文件缓存、电源与即插即用、安全等相关的服务。

每个执行体互相独立，只对外提供相应的接口，其它执行体要通过内核模式可调用接口和其它执行体通信或者请求其完成相应的功能服务。所有的设备驱动和文件系统都由 I/O 管

理器统一管理，驱动程序可以堆叠形成 I/O 驱动栈，功能请求被封装成 I/O 包，在栈中一层层流动处理。Windows 引以为傲的图形子系统也在内核中。

显而易见，NT 内核中各层次分明，各个执行体互相独立，这种“高内聚、低偶合”的特性，正是检验一个软件工程是否优秀的重要标准。而这些你都可以通过微软公开的 WRK 代码得到佐证，如果你觉得 WRK 代码量太少，也可以看一看REACT OS这个号称“开源版”的 NT。

重点回顾

到这里，我们了解了 Linux、Darwin-XNU 和 Windows 的发展历史，也清楚了它们内部的组件和结构，并对它们的架构进行了对比，对比后我们发现：**Linux 性能良好，结构异常复杂，不利于问题的排查和功能的扩展，而 Darwin-XNU 和 Windows 结构良好，层面分明，利于功能扩展，不容易产生问题且性能稳定。**

下面我们来回顾下这节课的重点。

首先，我们从一名计算机黑客切入，简单介绍了一下 Linus，他由于沉迷于技术，对不好的规则敢于挑战而写出了 Linux 雏形，并且利用了 GNU 开源软件的精神推动了 Linux 后来的发展，这样的精神很值得我们学习。

然后我们探讨了 Linux 内核架构，大致搞清楚了 Linux 内核中的各种组件，它们是系统、进程、内存、储存、网络。其中，每个组件都是从接口到硬件经过了几个层次，组件与组件之间的层次互联调用。这些组件组合在一起，其调用关系形成了一个巨大的网状结构。因此，Linux 也成了宏内核的代表。

为了有所对比，我们研究了苹果的 Darwin-XNU 内核结构，发现其分层更细，固件层、Mach 层屏蔽了硬件平台的细节，向上层提供了最基础的服务。在 Mach 层之上的 BSD 层提供了更完善的服务，它们是进程与线程、IPC 通信、虚拟内存、安全、网络协议栈以及文件系统。通过 Mach 中断嵌入表，可以让应用自己决定使用 Mach 层服务还是使用 BSD 层的服务，因此 Darwin-XNU 拥有了两套内核，Darwin-XNU 内核层也成为了多内核架构的代表。

最后，我们研究了迄今为止，最成功的商业操作系统——Windows，它的内核是 NT，其结构清晰明了，各组件完全遵循了软件工程**高内聚、低偶合**的设计标准。最下层是

HAL（硬件抽象），HAL 层是为了适配各种不同的硬件平台；在 HAL 层之上就是微软定义的小内核，你可以理解成是 NT 内核的内核；在这个小内核之上就是各种执行体了，这些执行体提供了操作系统的进程、虚拟内存、文件数据缓存、安全、对象管理、配置等服务，还有 Windows 的技术核心图形系统。

思考题

Windows NT 内核属于哪种架构类型?

很期待在留言区看到你的分享，也欢迎你把这节课分享给身边的同事、朋友。

我是 LMOS，让我们下节课见。

上一篇 03 | 黑盒之中有什么：内核结构与设计

下一篇 05 | CPU工作模式：执行程序的三种模式

精选留言

写留言

> 由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。