



下载APP



## 开篇词 | 为什么你要系统学习计算机的内存知识？

2021-10-25 海纳

《编程高手必学的内存知识》

课程介绍 &gt;



讲述：海纳

时长 10:40 大小 9.79M



你好，我是海纳，欢迎来到《编程高手必学的内存知识》的课堂。

我曾经是 Huawei JDK 团队的负责人，现在是华为鲲鹏生态的布道师，同时还在负责华为编译器领域的相关创新工作。2019 年我出版了《自己动手写 Python 虚拟机》一书。这两年，我利用业余时间又筹备了一本《从零开始写 Linux 内核》，预计明年上市。

计算机内存可以说是我的老朋友了。我和它结缘于我对基础软件开发的热爱。2007 年，我本科毕业，在人大金仓从事国产数据库开发。



记得有一次，我在公司看 B+ 树代码。我之前一直都搞不懂 B+ 树为什么要这么设计，直到那段时间，我研究 Linux 内核代码，看到磁盘 IO 和页缓存算法时才恍然大悟。我发现，原来我对很多技术的不理解，是来自于我计算机内存管理这些知识的匮乏。

后来，我又去读了研。2012 年，研究生毕业后，我去了网易做游戏开发，偏离了我曾经热爱的基础软件开发领域，但我居然又遇见了我的“老朋友”，内存管理。网易大量的游戏逻辑都是使用 Python 虚拟机构建的，但是由于 Python 采用了引用计数法进行自动内存管理，使用不当的时候，很容易发生大量的模型文件因循环引用不能自动释放的问题。

为了解决这个问题，我就希望能在 Python 虚拟机中引入基于复制的内存管理算法。所以，我就打开 Python 虚拟机的源代码，研究它的内存管理部分，做了一些尝试，虽然最终未能商业落地，但却收获了宝贵的经验。这些经验都总结在《自己动手写 Python 虚拟机》这本书里了。

两年后，我加入了华为，从事 Huawei JDK 的研发工作。再一次回到基础软件开发领域的我，又跟内存管理“杠”上了。那时，我遇到过一个非常棘手的问题。

这个问题的表象是，在鲲鹏 CPU 上，SynchronousQueue (SQ) 会卡死线程。因为 SQ 这个结构使用了多个 volatile 变量，volatile 的读写规则是在 Java 内存模型 (Java Memory Model, JMM) 中定义的，而华为鲲鹏芯片采用的是弱内存模型。所以，我们要解决这个问题，就不光要对 JMM 非常熟悉，还要对 CPU 的缓存管理有深入的理解。再加上，SQ 的具体实现会经过 HotSpot C2 编译器的优化，因此，我们同时还要搞清楚编译器是如何对内存操作进行翻译和优化的。

为了解决这个问题，我查阅了 JSR133 文档、CPU 的设计手册、JVM 的编译器设计和内存管理设计，还有多核 CPU 的操作系统设计等资料，深刻感受到想要分析这种涉及多个领域的问题，必须全面掌握内存相关的软硬件知识。

如果说，我与计算机内存的缘分，来自于我对基础软件开发和计算机底层原理的偏爱。那么今天，我想从更理性的、自身发展规划的角度看，和你谈谈为什么要系统地学习内存知识？

## 为什么我建议你系统学习内存知识？

现在，聊到程序员这个行业，逃不开的话题就是“职业生涯的 35 岁危机”。而且，你也能感受到，每年都有大量的人转行到计算机，初级的、写 CRUD 逻辑的人数已经逐渐见顶，出现供大于求的情况。

但是另一方面，结构性缺人又是非常常见的现象，很多公司还是感觉很难招到合适的开发人员。就拿我们编译器领域来说，招聘合适的人才，周期往往都要拉长到一年。

这是因为，在互联网的早期发展阶段，国内的 ICT 公司更多还是直接使用欧美的基础设施和技术，例如大型服务器、操作系统、数据库、编程语言、编译器、网络协议等等。随着国内企业的规模越来越大，各种定制化的需求也层出不穷，很多企业越来越希望自己能维护和修改基础设施。所以，当前市场对能进行底层开发的系统级程序员的需求也就越来越大。这对我们个人来说，是一个突破职业瓶颈的好机会。

那么，如何从一个应用开发程序员成长为资深的系统级程序员呢？

绕不开的一点是，我们需要熟练掌握 CPU 的工作原理、操作系统原理、编译器原理、分布式软件、图形渲染和数据库原理。**如果从这些庞大的知识体系中选择一条脉络的话，我还是会推荐以内存管理为线索去进行学习。**

举个例子，在 C 语言编写的程序中，一个变量要经过编译器、链接器、加载器和操作系统的进程管理，然后再经过 CPU 的 MMU 模块，才能最终出现在真正的物理内存里。如果你能把这个过程讲清楚了，那就说明你对编译、链接、加载、操作系统和 CPU 的工作原理有了相当的理解。所以说，内存管理的知识就相当于纲领，纲举则目张。

## 为什么内存相关知识这么难学？

从我个人的学习经历中也可以看出来，想学好内存管理，必须要掌握 CPU 核设计知识、CPU 的段页式管理等知识，还要深入了解操作系统的内存管理模块、编译器的内存分配、并发锁、基础库的设计原理等等。整体的知识结构，你可以参考下面这张图：



这里面涉及的知识覆盖了《数字电路设计》、《计算机原理》、《微处理器架构》、《体系结构》、《操作系统》、《编译原理》，还有各种编译语言库和虚拟机等等。每一点拎出来，都是一门独立的课程。即使你是在国内最好的计算机院校，想在这些课程里都取得高分也是非常困难的。更不要说，这些课程还没有涉及最新、最前沿的体系结构。

所以，我们不难看出，内存相关的知识体系的特点是：

1. **内容非常多，涉及的学科特别多。**软硬件的工作原理都得掌握才能学得好。
2. **内容非常散，不成体系。**高校里传统的课程都是分层水平设置的。想学好内存管理的知识，需要将它从不同的课程拉取出来，自己进行合并总结才能最终形成知识体系。

针对这些难点，我设计了这套课程，希望能避免水平式讲解方式难以形成体系的缺点，通过垂直化地学习软硬件相关知识，能让你快速地形成知识体系。

## 这门课是怎么设计的？

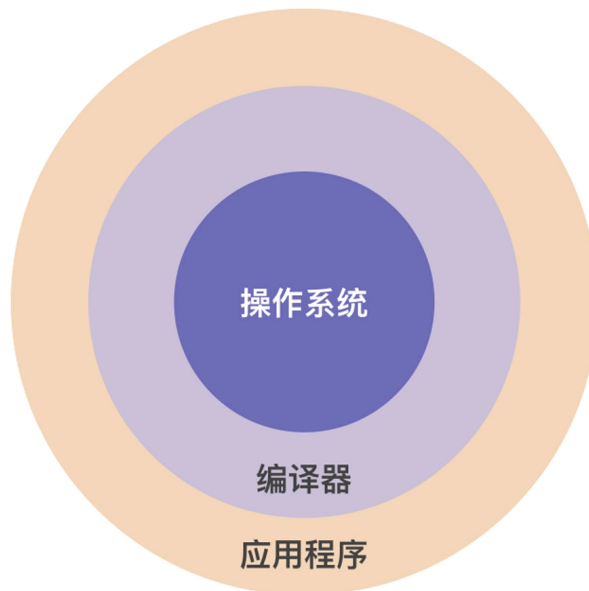
我们的课程分为三个部分：软件篇，硬件篇和自动内存管理篇。

### 软件篇

这一部分，我们会以操作系统为核心，将进程和内存的关系彻底讲清楚，让你对操作系统、编译器，以及应用程序的运行原理有深入的理解。这样，当你遇到进程 crash 时，分析 coredump、查看内存映射等都能游刃有余。

具体来讲，操作系统是软硬件设计的核心，它管理着所有的硬件资源，同时又为各种运行在它上面的应用程序提供服务。而内存是计算机的核心资源之一，如何高效地管理内存是操作系统的基础任务，甚至可以说是最重要的任务，它深刻影响着应用程序使用内存的方式。

而且，编译器和应用程序都是围绕这个核心来构建的。所以，在具体的讲述上，我们会采取由核心向外延的方式，把软件篇的知识都给你串联起来，如下图所示。



另外，我们这个专栏所举的例子都是运行在 Linux 系统上的，采用的编译器是 gcc，Java 语言使用的 JVM 是 Huawei JDK。我希望你在学习时候也准备好相关的运行环境，只要 Linux 内核不低于 4.1，gcc 版本不低于 4.8，JDK 版本不低于 8 就可以了。

## 硬件篇

这一部分，我们将会学习与内存相关的计算机体系架构，包括存储器电路、存储体系结构和多核 CPU 内存模型。

通过这部分的学习，你可以快速掌握缓存原理、多核 CPU 通信的核心知识，让你最大限度地使用好缓存，写出正确而高效的代码。同时，我们课程里对硬件知识的讲解，还可以让你掌握 CPU 设计一些简单原理，并且在此基础上，可以深入地学习 RISC-V 等开源 CPU 的实现。

我们都知道，随着 CPU 主频的提升越来越困难，厂商的主流做法是使用更多的 CPU 和更多的核来计算进行并行加速。这个时候，如何保证 CPU 核间的数据同步就成了硬件工程师的一大挑战，系统程序员必须得对这些原理有足够了解，才能写出正确高效的代码。

在讲存储器电路的时候，你会发现，根据不同的物理特性原理，人们制作了不同的存储器件，有的成本高、容量小，但速度快；有的则是成本低、容量大，但速度慢。而如何合理使用不同的器件构建一套高效的系统，让各种存储器件“扬长避短”，就是我们在计算机体系结构方面要重点研究的内容。

从下面这张图你也可以看到，硬件的结构是从下到上层层搭建的。所以，我会采用自底向上的讲解方式，带你理解存储系统是怎样由简单的器件一步步变得越来越复杂的。



## 内存管理篇

在这一部分，我们的关注点会从计算机软硬件，转向对具体内存管理算法的学习。



学习内存管理算法，能够让我们正确地使用各种不同的语言，例如在 Python 和 Swift 中你要注意解循环引用等等。而且，你还可以通过学习原理处理各种现实的问题，例如我们前面提到的 Java 程序的 STW 问题，等等。

我们知道，Java、Go、Python、JavaScript 等语言的内存都是自动管理的，也就是由语言虚拟机托管的，所以，开发者不需要关心内存的申请和释放的问题，我们把这种类型的语言称为 Managed Language。

自动内存管理极大地减少了程序员的心智负担。但是，作为程序员，如果你不了解自动内存管理的原理，还是会遇到各种各样的问题，比如，垃圾回收时 stop the world 带来的服务器失去响应、out of memory error 等等。这也是为什么，JVM 的内存管理一直都是 Java 程序员的热门话题。在这个课程里，我们将在内存管理篇把这部分内容一次性讲透。

总的来说，内存管理算法一直在不断克服以前算法的缺点，持续向前演进，而我们这部分整体的讲述逻辑，就是按照算法的演进逻辑进行的。

以上就是软件篇，硬件篇和自动内存管理篇的主要内容，整个课程的详细目录，我也放在了这里，你可以看一下：

## 《编程高手必学的内存知识》课程目录

### ■ 开篇词 | 为什么你要系统学习计算机的内存知识?

## 软件篇

01 虚拟内存：为什么可用内存远超物理内存？

02 聊聊 X86 体系架构中的实模式和保护模式

03 内存布局：应用程序是如何安排数据的？

04 深入理解栈：从 CPU 和函数的视角看栈的管理

05 栈的魔法：从栈切换的角度理解进程和协程

06 静态链接：变量与内存地址是如何映射的？

07 动态链接（上）：地址无关代码是如何生成的？

08 动态链接（下）：延迟绑定与动态链接器是什么？

09 深入理解堆：malloc 和内存池是怎么回事？

10 页中断：fork、mmap 背后的守护神

11 即时编译：高性能 JVM 的核心秘密

12 内存虚拟化：云原生时代的奠基者

## 硬件篇

13 存储电路：计算机存储芯片的电路结构是怎样的？

14 CPU Cache: 访存速度是如何大幅提升的？

15 MESI 协议：多核 CPU 是如何同步高速缓存的？

16 内存模型：有了 MESI 为什么还需要内存屏障？

17 NUMA：非均匀访存带来了哪些提升与挑战？

## 自动内存管理篇

18 Java 内存模型：Java 中的 volatile 有什么用？

19 垃圾回收：如何避免内存泄漏？



20 Scavenge: 基于 copy 的垃圾回收算法

21 分代算法: 基于生命周期的内存管理

22 G1 GC: 分区回收算法说的是什么呢?


23 Pauseless GC: 挑战无暂停的垃圾回收

24 GC 实例: Python 和 Go 的内存管理机制是怎样的?

■ 结束语 | 自主基础软件开发的那片星辰大海

在最后,我想说成长为系统程序员,不仅仅能帮你突破职业瓶颈,提升个人的薪酬待遇,更重要的是,这是顺应国情和行业发展大趋势的选择。在当前复杂的国际环境下,我们要走出一条自主可控的芯片设计和基础软件设计之路,这是迫在眉睫的事情。我希望,未来能有越来越多的开发者从事 CPU 设计,操作系统和编译器开发。如果你也是这么想的,那么这个专栏可能就是你最好的启蒙,一起开始吧。

分享给需要的人, Ta 订阅后你可得 **20 元现金奖励**

 生成海报并分享

 赞 4  提建议

© 版权归极客邦科技所有,未经许可不得传播售卖。页面已增加防盗追踪,如有侵权极客邦将依法追究其法律责任。

下一篇 [导学 \(一\) | 拆解CPU的基本结构和运行原理](#)

1024 活动特惠

VIP 年卡直降 ¥2000

新课上线即解锁, 享 365 天畅看全场

超值拿下 ¥999



## 精选留言 (6)

写留言



2021-10-25

听过老师几次直播, 干货满满, 感谢。话说, 准备要用的Huawei JDK与我们平常用的JDK有什么区别么?

作者回复: 有一些arm分支上的特有问题, openjdk的修复合入了高版本, 低版本openjdk不管了。Huawei jdk低版本也会修复。

1

6



Noah

2021-10-25

今年疯狂学习操作系统知识, 不为别的, 就为更好的自己!

1

3



飞

2021-10-25

直播有回放吗

展开

编辑回复: 有的，第二天会更新到专栏的“课程导学”章节里~ 另外，B站上关注极客时间官方账号，也可查看直播回放哦~

1

...

2021-10-26

是一直要努力的方向，但是任重道远  
展开

kylin

2021-10-25

看了老师之前的直播，非常精彩，果断买了  
展开

Geek\_24df3c

2021-10-25

华为jdk是开源的吗  
展开

1