

## 02 | RISC特性与发展：RISC-V凭什么成为“半导体行业的Linux”？

2022-07-20 LMOS 来自北京



天下无鱼

<https://shikey.com/>

《计算机基础实战课》

[课程介绍 >](#)



讲述：陈晨

时长 14:23 大小 13.19M



你好，我是 LMOS。

上节课，我带你见证了两种计算机指令集的设计结构——CISC 与 RISC。而今天我们的“主角”就是 RISC 中的一个代表性特例，它就是 RISC-V。

作为未来芯片指令集的主流，RISC-V 是今后芯片设计的最佳方案，甚至可以说它就是硬件行业里的 Linux。

为什么这么说呢？这节课，我会从 RISC-V 发展历史、原理与技术特性几个方面入手，带你弄明白为什么 RISC-V 在半导体行业中发展得如此迅猛。

### RISC-V 从何而来

让我们“穿越时空”，把时间线拉回到 2010 年。在加州伯克利分校的校园中，Krste Asanovic 教授正为了学生们学习计算机架构而发愁。由于现存芯片指令集冗余且专利许可费用昂贵，还有很多 IP 法律问题，没有一款合适的 CPU 用于学习。

于是他想要带领一个研究团队，来设计一款用于学生学习的 CPU。研究团队在选择架构的时候，对比了传统已经存在的 ARM、MIPS、SPARC 以及 x86 架构等，发现这些架构指令集要么极其复杂，要么极其昂贵。所以，他们的研究团队最终决定设计一套全新的指令集。

这个研究团队最开始只有 4 个人，却在三个月之内完成了指令集原型开发，其能力可见一斑。后来，这个项目被计算机体系结构方面的泰斗 David Patterson 发现，并且得到了他的大力支持。

早在 1981 年，伯克利分校已经设计出了第一代 RISC 指令集，之后又陆续设计了四代 RISC 指令集的架构。有了这些设计经验，在设计 RISC-V 指令集架构时，研究团队就变得驾轻就熟。

用 RISC-V 来命名该指令集架构，有两层意思：RISC-V 中的“V”，一方面代表第 5 代 RISC；另一方面，“V”取“Variation”之意，代表变化。

虽然 RISC-V 在 2010 年才开始研发，但在第二年（2011 年）就进行了首次流片，流片就是按照芯片设计生产出可以工作的芯片成品。从这以后，基于 RISC-V 的 CPU 设计，或者在 SOC 中集成 RISC-V 架构，各种软件工具链的开发和各种基于 RISC-V 架构的操作系统移植，在不断涌现。这给 CPU 整个行业带来了不小轰动，当然，这也离不开泰斗 David Patterson 的号召作用。

一个产品的成功，技术当然是非常重要的原因，但也需要非常给力的运营。在 2015 年，由最初的 25 个成员一起成立了非盈利性组织 RISC-V 基金会（RISC-V Foundation）。

之后，有多达 300 个单位加入 RISC-V 基金会，其中包括阿里、谷歌、华为、英伟达、高通、麻省理工学院、普林斯顿大学、印度理工学院、中科院计算所、联发科等。这些学术机构、芯片开发公司、设计服务与系统厂商的加入，为 RISC-V 的发展建立了良好的生态环境。

## RISC-V 是什么

通过上节课，我们知道了 RISC 就是指精简指令集计算机体系结构。而前面也说了，RISC-V 是伯克利分校设计出的第五代 RISC 指令集架构。但既然迅速走红的是 RISC-V，相比其他的 RISC，它必然有过人之处，这样才能立足于市场，不然很可能只沦为学生的学习工具。

如果只是对 RISC-V 下一个定义，会相当简单：**RISC-V 是一套开放许可证书、免费的、由基金会维护的、一个整数运算指令集外加多个扩展指令集的 CPU 架构规范。**

任何硬件开发商或者相关组织都可以免费使用这套规范，构建 CPU 芯片产品。如果我们的目的只是想对 RISC-V 有个概念了解，前面这些信息就足够了。不过想知道 RISC-V 为什么流行这个秘密，显然我们还需要更详细的信息，才能深入了解。

### 指令集命名方式

现在假定我们是一家芯片公司的芯片工程师，公司有了新的业务，对 CPU 提出了更高的要求，让我们基于 RISC-V 指令集架构实现一款全新的 CPU。根据我们公司的业务场景，我们只需要选择 RISC-V 架构中的一部分指令，CPU 的位宽也有特定的要求。

因此，现在需要一些命名方式，来对我们选择的 RISC-V 架构进行命名。这样，用户在阅读该 CPU 文档时，马上就能了解这款 CPU 是多少位的、有哪些指令集，它们具体有什么功能。

其实这个命名方式在 RISC-V 规范中有相关定义说明，以 RV 为前缀，然后是位宽，最后是代表指令集的字母集合，具体形式如下：

1 RV[###][abc.....xyz]

复制代码

我用表格为你说明一下这个格式，如下所示：

符号	说明
RV	用于标识 RISC-V 体系架构的前缀，即 RISC-V 的缩写
[###]	用于标识处理器的位宽，取值{32, 64, 128}，也就是处理器的寄存器的宽度
[abc.....xyz]	标识该处理器支持的指令集模块集合



举个例子，比如：RV64IMAC，就表示 64 位的 RISC-V，支持整数指令、乘除法指令、原子指令和压缩指令。

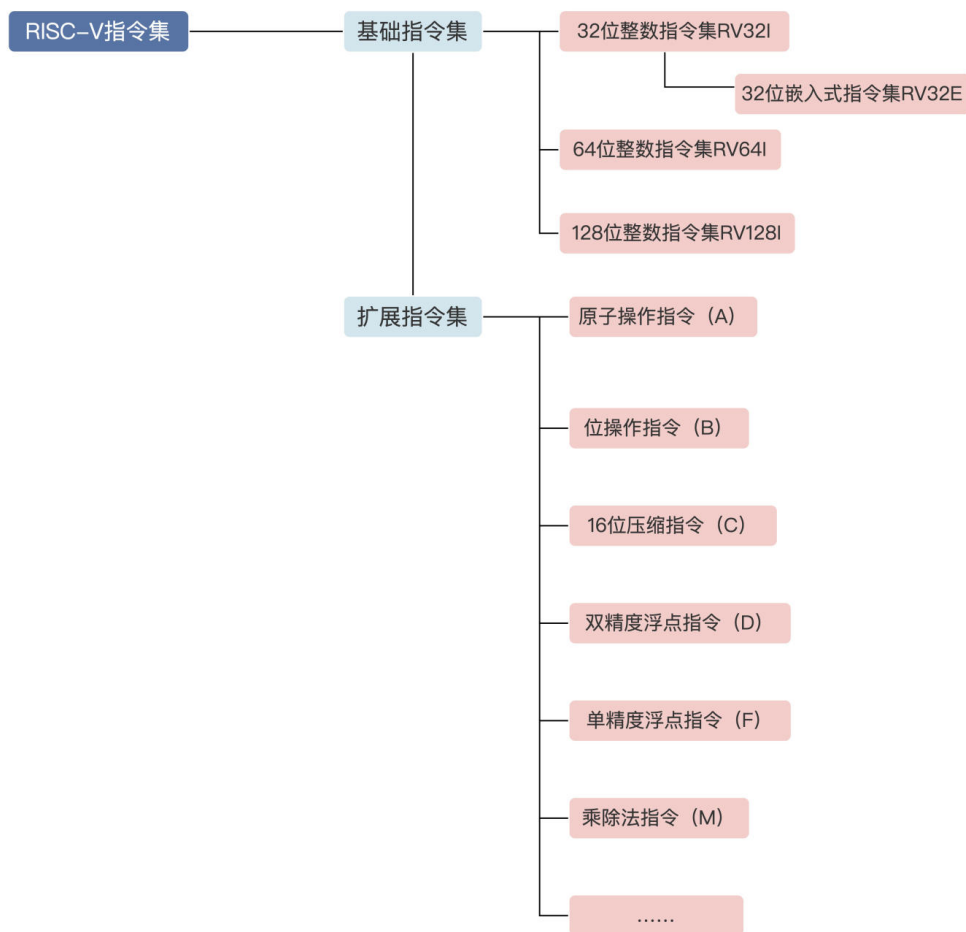
## 指令集模块

接着我们一起来看看指令集模块。指令集是一款 CPU 架构的主要组成部分，是 CPU 和上层软件交互的核心，也是 CPU 主要功能的体现。

但 RISC-V 规范只定义了 CPU 需要包含基础整形操作指令，如整型的储存、加载、加减、逻辑、移位、分支等。其它的指令称为可选指令或者用户扩展指令，比如乘、除、取模、单精度浮点、双精度浮点、压缩、原子指令等，这些都是扩展指令。扩展指令需要芯片工程师结合功能需求自定义。

所以 RISC-V 采用的是模块化的指令集，易于扩展、组装。它适应于不同的应用场景，可以降低 CPU 的实现成本。

我给你列了一张图，图里展示的是 RISC-V 指令集的各个组成部分。



RISC-V 的基本指令集和扩展指令集你有个大致印象就可以，更详细的技术讲解，后面第五节课我再展开。下面我们去看看 RISC-V 的寄存器。

## RISC-V 寄存器

指令的操作数是来源于寄存器，精简指令集架构的 CPU，都会提供大量的寄存器，RISC-V 当然也不例外。RISC-V 的规范定义了 32 个通用寄存器以及一个 PC 寄存器，这对于 RV32I、RV64I、RV128I 指令集都是一样的，只是寄存器位宽不一样。

如果实现支持 F/D 扩展指令集的 CPU，则需要额外支持 32 个浮点寄存器。而如果实现只支持 RV32E 指令集的嵌入式 CPU，则可以将 32 个通用寄存器缩减为 16 个通用寄存器。

寄存器	ABI名称	说明
x0	zero	0值寄存器，硬编码为0，写入数据忽略，读取永远为0
x1	ra	用于返回地址
x2	sp	用于栈指针
x3	gp	用于通用指针
x4	tp	用于线程指针
x5	t0	用于存放临时数据或者备用链接寄存器
x6~x7	t1~t2	用于存放临时数据寄存器
x8	s0/fp	需要保存的寄存器或者帧指针寄存器
x9	s1	需要保存的寄存器
x10~x11	a0~a1	函数传递参数寄存器或者函数返回值寄存器
x12~x17	a2~a7	函数传递参数寄存器
x18~x27	s2~s11	需要保存的寄存器
x28~x31	t3~t6	用于存放临时数据寄存器



为了帮助你聚焦要点，不常用的 32 个浮点寄存器并没有列在这张表里。表中的 **ABI** 名称，即应用程序二进制接口，你可以理解为寄存器别名，高级语言在生成汇编语言的时候会用到它们。

比如 C 语言处理函数调用时，用哪些寄存器传递参数、返回值，调用者应该保护哪些寄存器，用什么寄存器管理栈帧等等。

定义好 **ABI** 标准，我们就能在语言间互相调用函数了。比如 C 语言函数调用汇编语言函数，这里我先卖个关子，后面语言与指令的篇章再给你详细展开。

## RISC-V 特权级

研究完了 RISC-V 寄存器，我们再来看看 RISC-V 的特权级。不同的特权级能访问的系统资源不同，高特权级能访问低特权级的资源，反之则不行。RISC-V 的规范文档定义了四个特权级别（**privilege level**），特权等级由高到低排列，如下表所示。

名称	级别	缩写	编码
用户、应用程序特权级	0	U	00
管理员特权级	1	S	01
虚拟机监视特权级	2	H	10
机器特权级	3	M	11

一个 RISC-V 硬件线程（hart），相当于一个 CPU 内的独立的可执行核心，在任一时刻，只能运行在某一个特权级上，这个特权级由 CSR（控制和状态寄存器）指定和配置。

具体分级如下：

1. 机器特权级（M）：RISC-V 中 hart 可以执行的最高权限模式。在 M 模式下运行的 hart，对内存、I/O 和一些必要的底层功能（启动和系统配置）有着完全的控制权。因此，它是唯一一个所有标准 RISC-V CPU 都必须实现的权限级。实际上，普通的 RISC-V 微控制器仅支持机器特权级。
2. 虚拟机监视特权级（H）：为了支持虚拟机监视器而定义的特权级。
3. 管理员特权级（S）：主要用于支持现代操作系统，如 Linux、FreeBSD 和 Windows。
4. 用户应用特权级（U）：用于运行应用程序，同样也适用于嵌入式系统。

好了，关于 RISC-V 的特权级，你了解这些，在现阶段已经足够了。需要把握的重点是，特权级起到了怎样的作用。

有了特权级的存在，就给指令加上了权力，从而去控制用指令编写的程序。应用程序只能干应用程序该做的事情，不能越权操作。操作系统则拥有更高的权力，能对系统的资源进行管理。

## RISC-V 因何流行

RISC-V 指令集架构在 2010 年才开发出来，到今天不过 10 多年的时间。这个时间从 CPU 行业的发展看，是非常短的，也可以说是非常年轻的。相比 x86 的 40 多岁的年纪，还有 ARM、MIPS、SPARC 的 30 多的年纪，RISC-V 简直是个孩子。

要知道 ARM、MIPS、SPARC 都是 RISC 系的，MIPS 和 SPARC 甚至已经进入了死亡阶段。按道理讲，RISC-V 不应该在这么短的时间内流行起来，成为芯片行业一颗耀眼的新星。

那么 RISC-V 流行起来，肯定有其优势：**一是 RISC-V 完全开放，二是 RISC-V 指令简单，三是 RISC-V 实行模块化设计，易于扩展。**

我们先来看看为什么说 RISC-V 是开放的。之前硬件和软件一样，都是小心地保护自己的“源代码”，因为那是自己的命脉。

直到后来，软件界出现了开源的 Linux，一经开源就迅速席卷了全球。在今天的互联网、云计算、手机等领域 Linux 已经无处不在。但是硬件依然保护着自己的“源代码”，Intel 和 AMD 还是以售卖 x86 芯片为主，而 ARM 直接售卖 ARM CPU 的“源代码”，连生产芯片的步骤都省了。

这种模式下，无论厂商还是个人，要获得 CPU 都要付出昂贵的代价。这时 RISC-V 应运而生，它完全毫无保留地开放了 CPU 设计标准，任何人都可以使用该标准，自由地设计生产 CPU，不需要支付任何费用，也没有任何法律问题。这相当于硬件界的“Linux”，推动了开放硬件的运动和发展。

然后，我们来看看为什么说 RISC-V 很简单？RISC-V 提供了一个非常强大且开放的精简指令集架构，只有 32 个通用寄存器、40 多条常用指令、4 个特权级。如果需要其它功能，则要进行指令集的扩展，单核心的规范文档才不到 300 页，一个人在一周之内就能搞清楚。

相比 ARM、x86 动不动就有 8000 多页的规范文档，这实在是太简单了。其实，简单也意味着可靠和高效，同时可以让学生或者硬件开发者迅速入手，降低学习和开发成本。

最后，我们来说说 RISC-V 的模块化设计。RISC-V 虽然简单，但这并不意味着功能的缺失。通过模块化的设计，就能实现对各种功能组件的剪裁和扩展。

事实上，现代 IT 架构已经发生了巨大的改变。举几个我们身边的例子吧。你正在使用的网卡，上面越来越多的网络处理任务和功能，都从主处理器上移到了网卡中，由网卡芯片自己来



处理了。

数据处理器 (DPU) 也体现了这一点。由于通用处理器对大规模数据处理能力的限制，所以我们需要专用的数据处理器。而人工智能领域，现在也已经开始通过 GPU 运行相关算法。

这些例子都在告诉我们，专用处理器芯片的需求在大量激增，而这正是 RISC-V 的用武之地。RISC-V 的标准开放，指令功能模块可以自由组合，所以用 RISC-V 就能定制一款满足特殊用途的处理器。芯片工程师会自由组合 RISC-V 现有的指令功能模块，按需对齐进行修改优化，或者实现一个新的指令功能模块，就像你根据需要修改和使用 Linux 内核一样。

正是因为 RISC-V 开放、简单和模块化这三大特点，硬件工程师和软件工程师才能站在巨人的肩膀上开发，自由地调用和组装功能模块，快速去实现特定业务场景下的芯片需求。因此，才有了 RISC-V 引爆芯片行业，迅速火热起来的现象，这是推动开放硬件的革命性壮举。

## 重点回顾

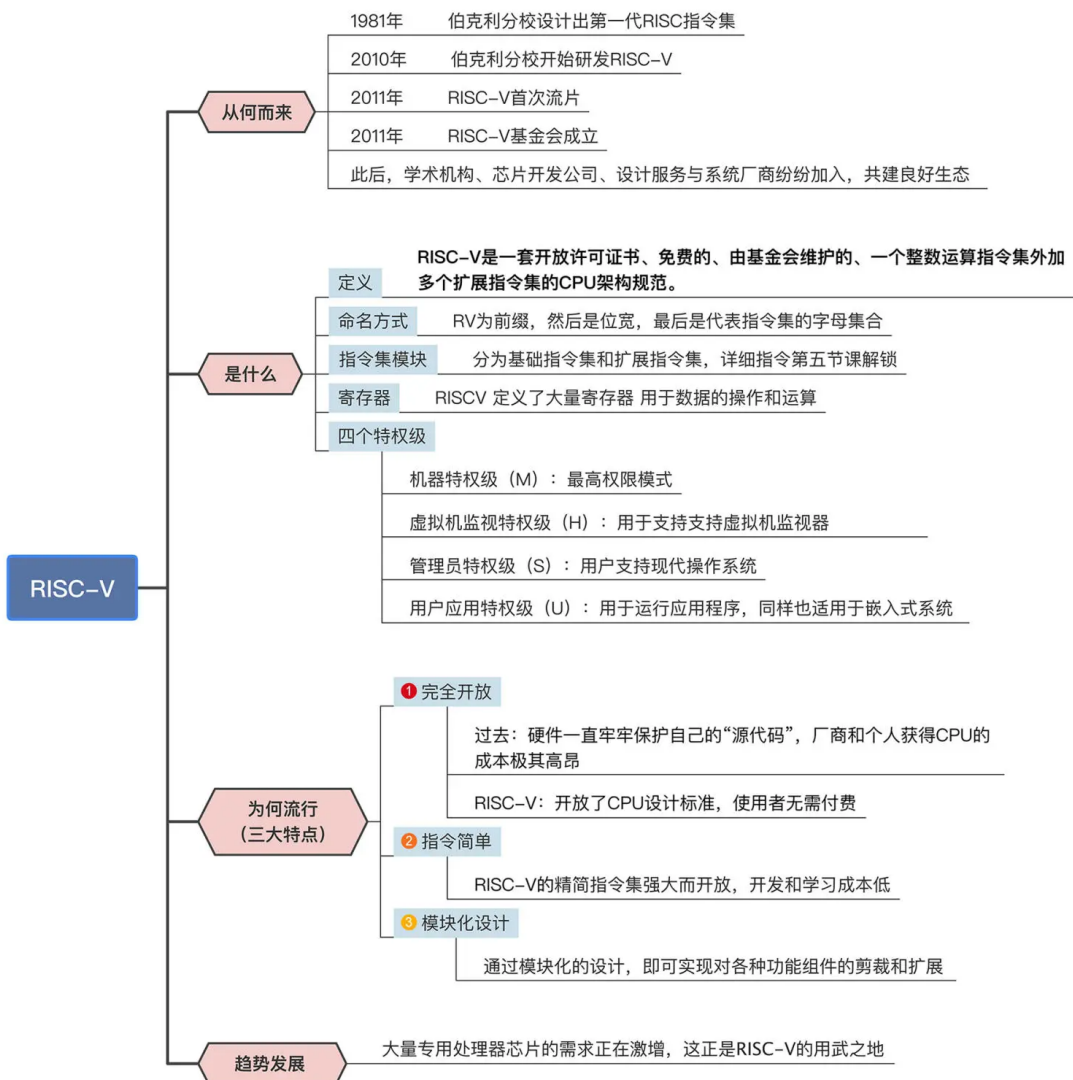
今天的课程又到了尾声，我们还是来看一下，在这节课中，我们都学习了什么。

首先，我们了解了 RISC-V 从何而来，明白了 RISC-V 发源于加州伯克利分校，是该校第五代 RISC 指令集，即第五代精简指令集。起初是为了学生有一套用来学习研究的指令集。后来因为有技术泰斗 David Patterson 的加入，又成立 RISC-V 基金会，RISC-V 慢慢流行了起来。

之后我们研究了 RISC-V 是什么，我带你了解了 RISC-V 指令集的命名方式、组成模块、寄存器与特权级。这些部分共同组成了 RISC-V 指令集架构规范。任何硬件厂商都可以按照这个规范，实现自己的 RISC-V 处理器。

最后，我们讨论了 RISC-V 因何流行。RISC-V 是开放的，没有任何法律和许可证问题，又极其简单，指令集是模块化的，易于剪裁和扩展。这种开放、简单、易于扩展的特点，使得硬件工程师非常容易上手和定制满足特定功能需要的处理器，这直接推动了开放硬件的革命。

课程里的重点内容，我整理成了导图，供你参考。



## 思考题

为什么 RISC-V 要定义特权级？

欢迎你在留言区跟我交流互动，如果觉得内容还不错，也推荐你把这节课分享给更多朋友。下节课我们就进入手写 miniCPU 的部分，敬请期待！

分享给需要的人，Ta订阅超级会员，你最高得 50 元

Ta单独购买本课程，你将得 24 元

生成海报并分享

上一篇 01 | CISC & RISC：从何而来，何至于此

下一篇 03 | 硬件语言筑基（一）：从硬件语言开启手写CPU之旅

## 精选留言 (25)

💬 写留言



听水的湖 置顶

2022-07-25

又见面啦，我是这门课的编辑小新。

以下是下节课的预告和“预习作业”：

下个模块开始手写CPU啦（8月1日更新），我们这次用Verilog语言写。老师正奋战中，把配套代码等等都为大家准备上，大伙儿可以趁等待更新这周吸收消化下前两节课。

还可以看看下面的“预习”作业：

1. 第一季也有一篇内容稍微涉及了一点Verilog语言，代码也不多，会带你实现一个可以计算任意 N 位二进制数的 ALU。迫不及待的同学可以先行去瞅瞅，找个感觉：<https://time.geekbang.org/column/article/409790>

2. 也许你之前并没接触过硬件语言，但对比学习大法还是很不错的。可以先自行思考一下Verilog语言和我们熟悉的软件编程所用语言有哪些不同，带着疑问学习，效果加倍哦。答案我们将在后面揭晓！

作者回复: 比心



👍 13



pedro

2022-07-20

为什么riscv需要定义不同的特权级？

这并不是riscv独有的，让我们把注意力回到老朋友intel上。

事实上，cpu刚出现的那段时间里，压根就没有特权级这个概念，任何人都是实打实土皇帝，但是这样怎么管理？应用程序之间经常打架斗殴，搞不好还把机器搞垮了。

然后inter的那帮人就搞出了一个特权级的概念，你虽然想打架，但我不准你打，你出手，我就红牌，然后大家都老实了，系统运行就很稳定了。

说了这么多，特权级无非是让大家守规矩，讲安全，和谐共生。

作者回复: 我p哥 66666

共 5 条评论 >

👍 29



仰望星空

2022-07-21

为什么RISC-V要定义特权级？主要是为了系统能稳定和安全的运行，设想一下如果没有特权级，任何应用都能任意访问所有硬件，这必将会是一场灾难

作者回复: 是的 是的 学的明白



👍 2



可爱因子1&#47;n

2022-07-21

就像老师上课讲到的，设置特权级就是给指令加上了权力，这样的话应用程序所执行的指令就只能做自己权利范围以内的事情，避免对其他的应用程序产生不必要的干扰和破坏，维持这个生态的稳定和平衡

作者回复: 正确 理解到位



👍 2



杨景胜

2022-07-28 来自北京

催更催更催更

编辑回复: 更新啦，催学～

共 2 条评论 >

👍 1



苏流郁宓

2022-07-21

俺觉得特权级区分用户软件与操作系统功能，在操作系统那边讲过，亮点应该是硬件线程，任意时刻只能运行在一个特权级上。这算不算是在安全与好用上做平衡？算不算RISCv指令集对cpu发展的贡献？

作者回复: 是的 其它CPU的核心 也是一样的 同一时间只用运行在一个特权级上



J霸

2022-07-20

RISC-V定义特权级，有助于保护系统资源不被滥用和破坏，不会因为低特权程序的bug或恶意代码破坏环境。

作者回复: 对 正确的



小杰

2022-08-10 来自浙江

每个模块，每个指令，做自己应该要做的事，不然就乱套了。



Lpc-Win32

2022-08-01 来自湖北

实模式、保护模式

作者回复: 哈哈 类比正确



Over10@d

2022-07-29 来自陕西

排除虚拟化的应用场景,对于多核处理器理论上有没有可能一个核运行在高特权级,一个核运行在低特权级?

作者回复: 这当然可能



jk001

2022-07-26

这几天怎么没更新？

编辑回复: 看下置顶留言哦，可以预习下～



熊出没

2022-07-25

权级设置是为了从杂乱中设置秩序 否则硬件资源就会被无序使用 导致机器崩溃

作者回复: 嗯嗯 对的



砥砺前行

2022-07-25

不得不说 彭老师这思维导图真心到位

作者回复: 哈哈 为提供优质教程而努力



落叶 建良

2022-07-25

为什么设置特权级？

从业务角度来理解，应该就是权限管理，特权级就是管理员，其他的属于不同的角色，拥有不同的权限，目的是为了系统的安全和稳定

编辑回复: 哈哈这个联想很不错嘛



Geek\_53914b

2022-07-24

同一时间只能运行在一个特权级，那不就不会冲突了吗？不同时间特权级来回切换不更乱？

作者回复: 为什么会乱





hmg

2022-07-22

今天没有更新吗，今天是周五啊

编辑回复: 收到催更。上线一次性发了三篇哦，8月1日起每周一三五各更新一篇～

今天已经更新03啦，记得跟上～



Ball

2022-07-22

设定特权级别主要为了适应分时操作系统的全局调度操作，部分系统资源的访问权限收敛到一个地方，方便解决冲突和提高资源利用率

作者回复: 对的



东风第一枝

2022-07-22

老师怎么看基于RISC-V的XiangShan (香山)开源处理器

作者回复: 支持 香山



和某欢

2022-07-22

设置特权级是为了保护操作系统资源，不被上层应用程序滥用资源导致系统崩溃

作者回复: 对的



jeigiye

2022-07-22

”特权”就相对“一般”，他们都为程序服务，应用、服务、驱动、操作系统都程序，但程序也有分层，地层服务上层，比如操作系统为应用服务，这样操作系统就要稳定，一些危险操作或是控制就只能操作系统运行，即“特权”

作者回复: 是的 哈哈

