



# 岁寒

(<https://lvwenhan.com/>) 任何事情，从现在开始做，都不晚！

## 性能之殇（三）-- 通用电子计算机的胎记：事件驱动

2018-11-13 / 阅读数：15947 / 分类：操作系统 (<https://lvwenhan.com/sort/36>)

Event-Driven（事件驱动）这个词这几年随着 Node.js® 的大热也成了热词，似乎已经成了“高性能”的代名词，殊不知事件驱动其实是通用计算机的胎记，是一种与生俱来的能力。本文我们就要一起了解一下事件驱动的价值和本质。

### 通用电子计算机中的事件驱动

首先我们定义当下最火的 x86 PC 机为典型的通用电子计算机：可以写文章，可以打游戏，可以上网聊天，可以读U盘，可以打印，可以设计三维模型，可以编辑渲染视频，可以作路由器，还可以控制巨大的工业机器。那么，这种计算机的事件驱动能力就很容易理解了：

1. 假设 Chrome 正在播放 Youtube 视频，你按下了键盘上的空格键，视频暂停了。这个操作就是事件驱动：计算机获得了你单击空格的事件，于是把视频暂停了。
2. 假设你正在跟人聊 QQ，别人发了一段话给你，计算机获得了网络传输的事件，于是将信息提取出来显示到了屏幕上，这也是事件驱动。

### 事件驱动的实现方式

事件驱动本质是由 CPU 提供的，因为 CPU 作为 控制器 + 运算器，他需要随时响应意外事件，例如上面例子中的键盘和网络。

CPU 对于意外事件的响应是依靠 Exception Control Flow（异常控制流）来实现的。

## 强大的异常控制流

异常控制流是 CPU 的核心功能，它是以下听起来就很牛批的功能的基础：

### 时间片

CPU 时间片的分配也是利用异常控制流来实现的，它让多个进程在宏观上在同一个 CPU 核心上同时运行，而我们都知道在微观上在任一个时刻，每一个 CPU 核心都只能运行一条指令。

### 虚拟内存

这里的虚拟内存不是 Windows 虚拟内存，是 Linux 虚拟内存，即逻辑内存。

逻辑内存是用一段内存和一段磁盘上的存储空间放在一起组成一个逻辑内存空间，对外依然表现为“线性数组内存空间”。逻辑内存引出了现代计算机的一个重要的性能观念：

**内存局部性天然的让相邻指令需要读写的内存空间也相邻，于是可以把一个进程的内存放到磁盘上，再把一小部分的“热数据”放到内存中，让其作为磁盘的缓存，这样可以在降低很少性能的情况下，大幅提升计算机能同时运行的进程的数量，大幅提升性能。**

虚拟内存的本质其实是使用 缓存 + 乐观 的手段提升计算机的性能。

### 系统调用

系统调用是进程向操作系统索取资源的通道，这也是利用异常控制流实现的。

### 硬件中断

键盘点击、鼠标移动、网络接收到数据、麦克风有声音输入、插入 U 盘这些操作全部需要 CPU 暂时停下手头的工作，来做出响应。

### 进程、线程

进程的创建、管理和销毁全部都是基于异常控制流实现的，其生命周期的钩子函数也是操作系统依赖异常控制流实现的。线程在 Linux 上和进程几乎没有功能上的区别。

## 编程语言中的 try catch

C++ 编译成的二进制程序，其异常控制语句是直接基于异常控制流的。Java 这种硬虚拟机语言，PHP 这种软虚拟机语言，其异常控制流的一部分也是有最底层的异常控制流提供的，另一部分可以由逻辑判断来实现。

## 基于异常控制流的事件驱动

其实现在人们在谈论的事件驱动，是 Linux kernel 提供的 epoll，是 2002 年 10 月 18 号伴随着 kernel 2.5.44 发布的，是 Linux 首次将操作系统中的 I/O 事件的异常控制流暴露给了进程，实现了本文开头提到的 Event-Driven（事件驱动）。

### Kqueue

FreeBSD 4.1 版本于 2000 年发布，起携带的 Kqueue 是 BSD 系统中事件驱动的 API 提供者。BSD 系统如今已经遍地开花，从 macOS 到 iOS，从 watchOS 到 PS4 游戏机，都受到了 Kqueue 的蒙荫。

### epoll 是什么

操作系统本身就是事件驱动的，所以 epoll 并不是什么新发明，而只是把本来不给用户空间用的 api 暴露在了用户空间而已。

### epoll 做了什么

网络 IO 是一种纯异步的 IO 模型，所以 Nginx 和 Node.js® 都基于 epoll 实现了完全的事件驱动，获得了相比于 select/poll 巨量的性能提升。而磁盘 IO 就没有这么幸运了，因为磁盘本身也是单体阻塞资源：即有进程在写磁盘的时候，其他写入请求只能等待，就是天王老子来了也不行，磁盘做不到呀。所以磁盘 IO 是基于 epoll 实现的非阻塞 IO，但是其底层依旧是异步阻塞，即便这样，性能也已经爆棚了。Node.js 的磁盘 IO 性能远超其他解释型语言，过去几年在 web 后端霸占了一些对磁盘 IO 要求高的领域。

### WRITTEN BY



## JohnLui (<https://github.com/johnlui>)

程序员, Swift Contributor

### 相关日志:

性能之殇（六）-- 现代计算机最亲密的伙伴：局部性与乐观  
(<https://lvwenhan.com/操作系统/497.html>)

性能之殇（四）-- Unix 进程模型的局限 (<https://lvwenhan.com/操作系统/495.html>)

性能之殇（七）-- 分布式计算、超级计算机与神经网络共同的瓶颈  
(<https://lvwenhan.com/操作系统/498.html>)

软件工程师需要了解的网络知识：从铜线到HTTP（五）—— HTTP 和 HTTPS (<https://lvwenhan.com/操作系统/489.html>)

软件工程师需要了解的网络知识：从铜线到HTTP（四）—— TCP 和路由器 (<https://lvwenhan.com/操作系统/488.html>)

标签: MAC (<https://lvwenhan.com/tag/MAC>) 性能之殇  
(<https://lvwenhan.com/tag/%E6%80%A7%E8%83%BD%E4%B9%8B%E6%AE%87>)  
性能 (<https://lvwenhan.com/tag/%E6%80%A7%E8%83%BD>) CPU  
(<https://lvwenhan.com/tag/CPU>) epoll (<https://lvwenhan.com/tag/epoll>) kqueue  
(<https://lvwenhan.com/tag/kqueue>)

← 性能之殇（四）-- Unix 进程模型的局限 (<https://lvwenhan.com/操作系统/495.html>)  
性能之殇（二）-- 分支预测、流水线与多核 CPU →  
(<https://lvwenhan.com/操作系统/493.html>)

### 评论:

**幻の上帝**

2019-02-07 18:24

用异常控制流实现所谓牛逼的分时抢占式多任务也只是一种相当原始的实现风格。像样点的东西么.....

[https://en.wikipedia.org/wiki/Engine\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Engine_(computer_science))

至于下面什么事件驱动，说穿了核心不就是I/O多路复用么.....一点都不提？

回复

**幻の上帝**

2019-02-07 18:19

底层本来根本没什么异常控制流的说法，只不过为了搞所谓的控制流分析简单才有那么个提法。

至于PL里的异常，跟这个根本两回事。基本上现实的任何 userspace 的所谓异常这个特性的实现，都不会直接用处理器提供的东西，因为这些异常（明确包括C++的这种）原则上是同步的，用处理器提供的机制只会浪费。像Windows SEH或者UNIX信号之类这种超出常规语言“正常实现”的异常才会用到这种不管是运行时性能还是抽象上都低效的东西。

当然，现在的这些现实的实现，大部分仍然是开洞。理论上正经一点的实现，比如<https://gist.github.com/sebfisch/2235780>，一般所谓主流语言的还没发展到能用上的程度。

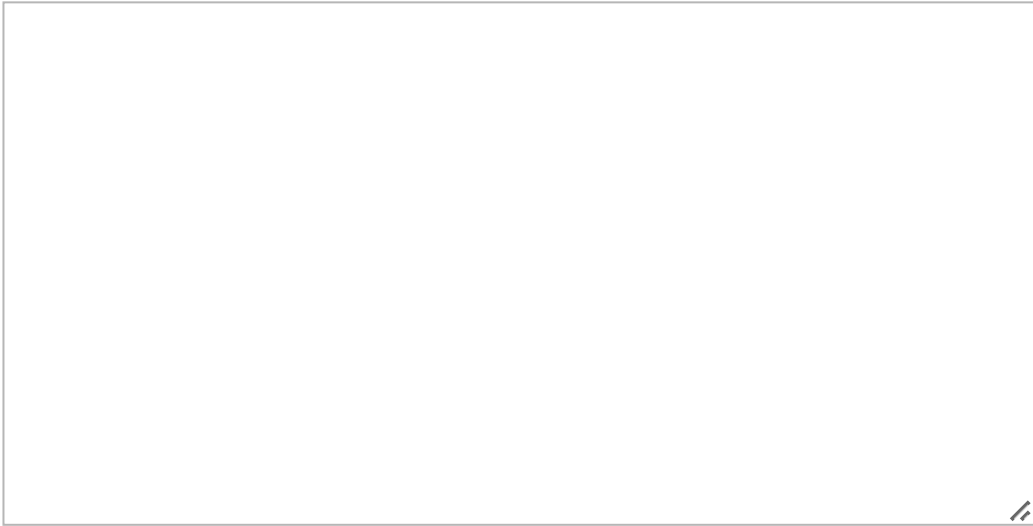
回复

**发表评论：**

昵称

邮件地址 (选填)

个人主页 (选填)



发表评论

友情链接： #Mukti's Blog (<http://www.feizhaojun.com/>) #住范儿 (<http://www.zhufaner.com/>) #Arron.y  
(<http://blog.helloarron.com/>) 京ICP备13030650号-2 (<https://beian.miit.gov.cn>)

© 2011-2022 岁寒 (<https://lvwenhan.com/>) | Powered by Emlog (<http://www.emlog.net/>)