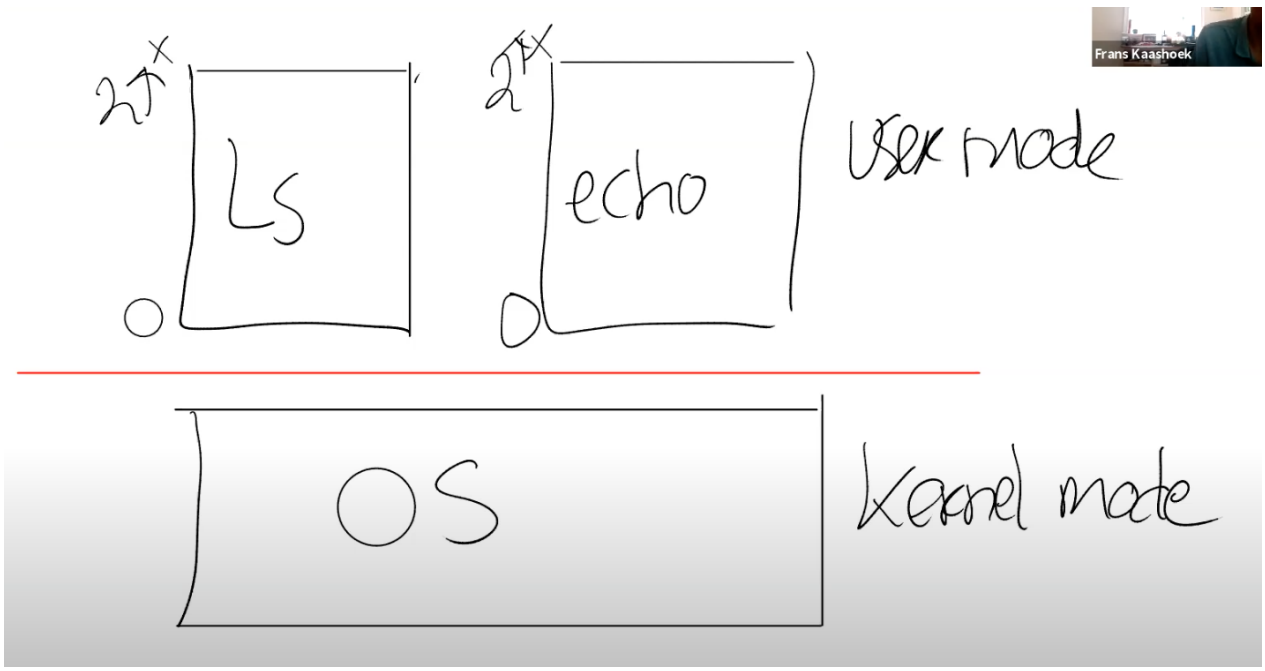
huihongxiao GitBook: [master] 13 p... ✓ Latest commit 7540b83 on Apr 24 History

1 contributor

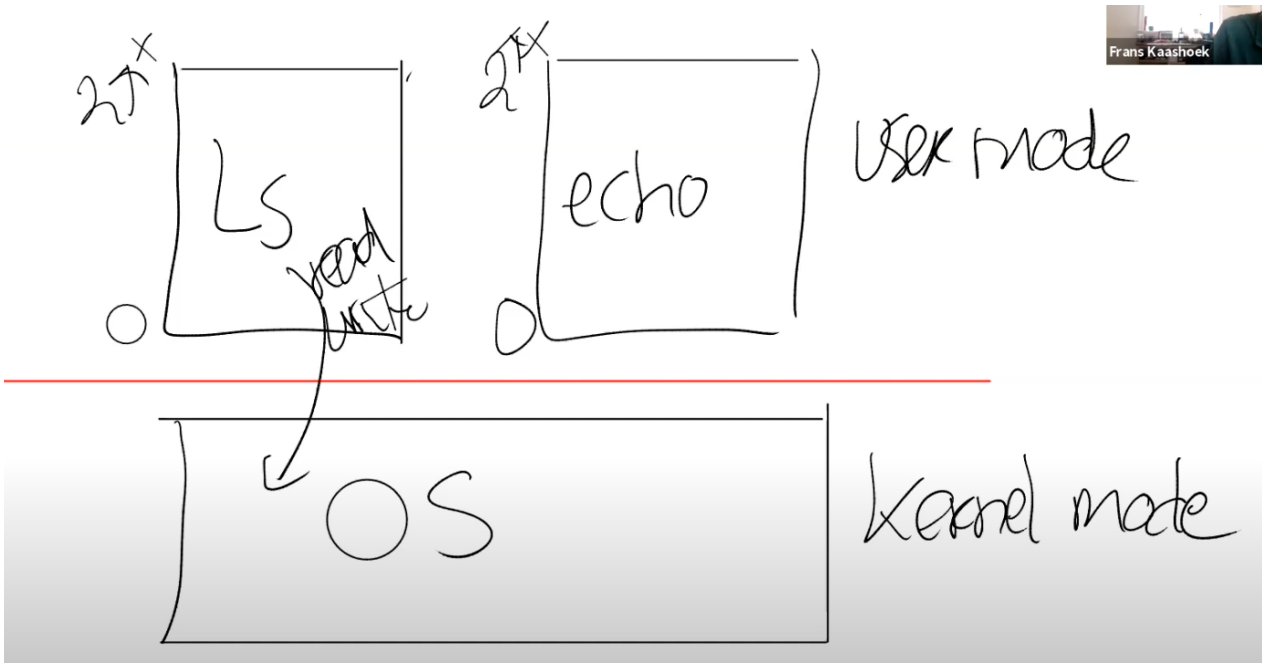
46 lines (29 sloc) 5.08 KB Raw Blame

3.5 User/Kernel mode切换

我们可以认为user/kernel mode是分隔用户空间和内核空间的边界，用户空间运行的程序运行在user mode，内核空间的程序运行在kernel mode。操作系统位于内核空间。

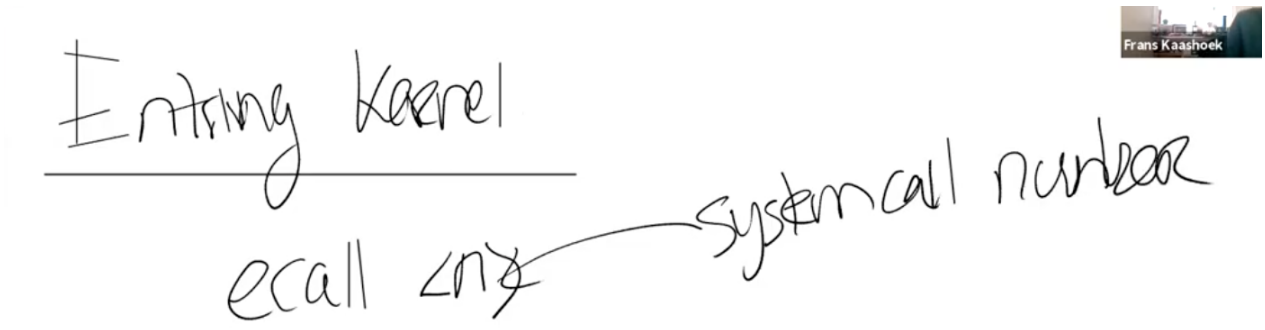


你们应该将这张图记在你们的脑子中。但是基于我们已经介绍的内容，这张图有点太过严格了。因为我们用矩形包括了一个程序的所有部分，但是这里没有描述如何从一个矩形将控制权转移到另一个矩形的，而很明显这种转换是需要的，例如当ls程序运行的时候，会调用read/write系统调用；Shell程序会调用fork或者exec系统调用，所以必须要有一种方式可以使得用户的应用程序能够将控制权以一种协同工作的方式转移到内核，这样内核才能提供相应的服务。



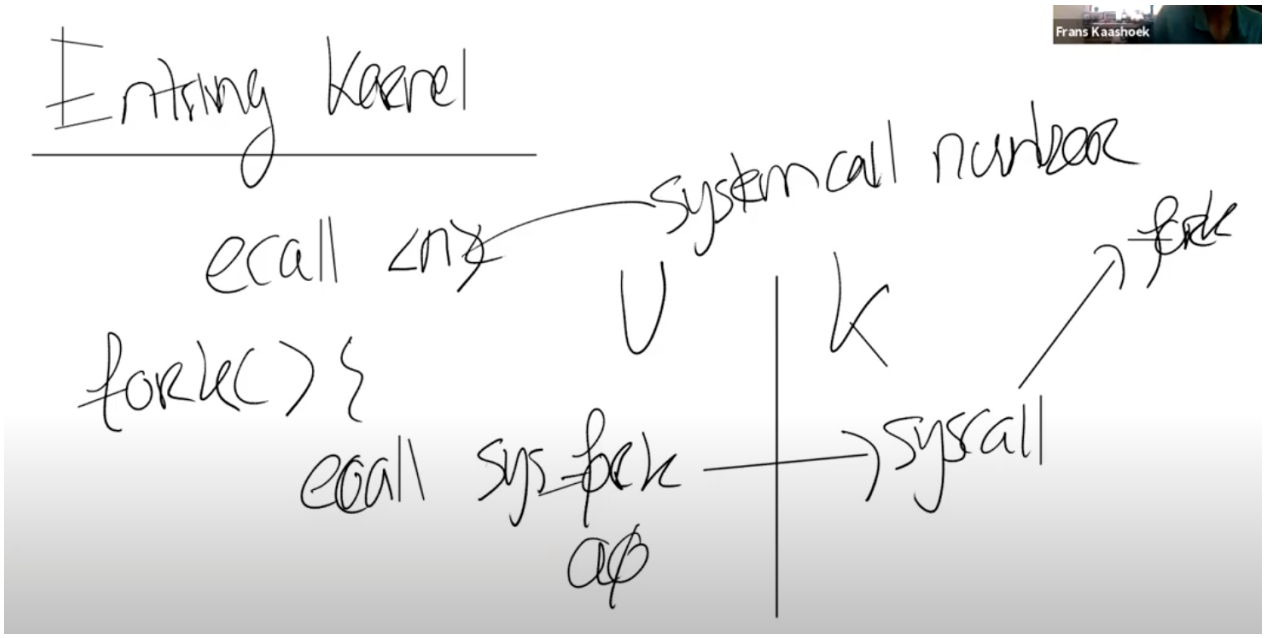
所以，需要有一种方式能够让应用程序可以将控制权转移给内核（Entering Kernel）。

在RISC-V中，有一个专门的指令用来实现这个功能，叫做ECALL。ECALL接收一个数字参数，当一个用户程序想要将程序执行的控制权转移到内核，它只需要执行ECALL指令，并传入一个数字。这里的数字参数代表了应用程序想要调用的System Call。



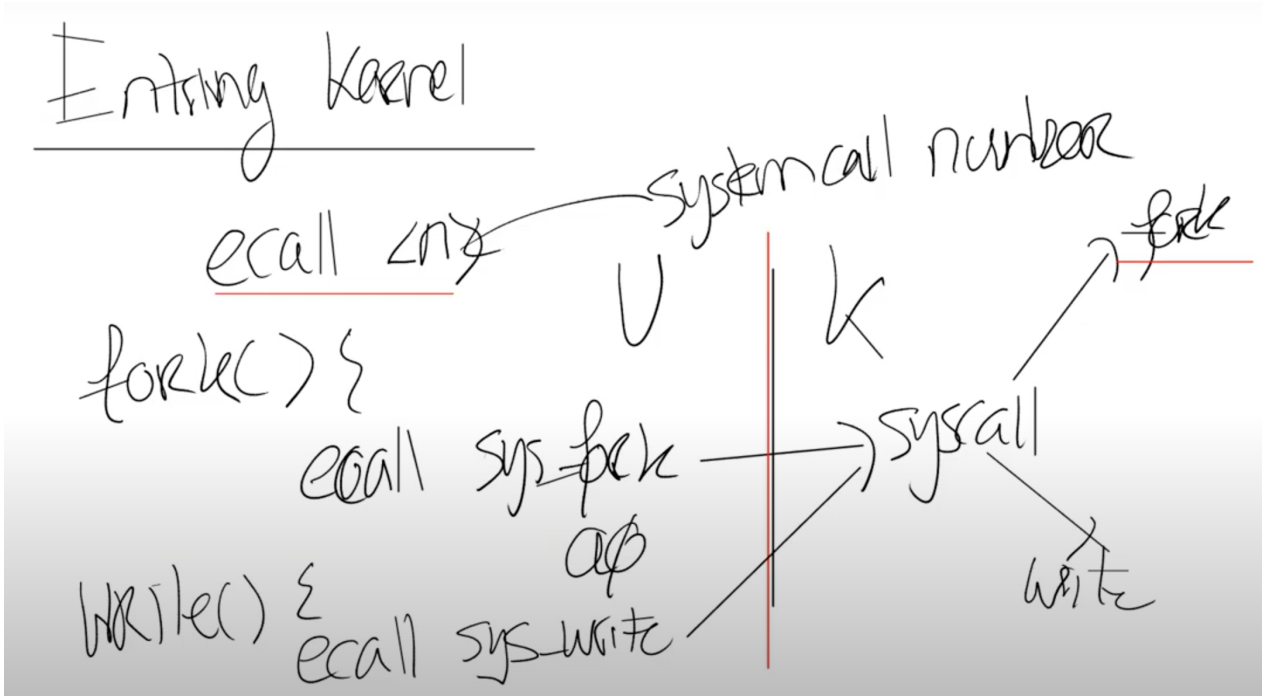
ECALL会跳转到内核中一个特定，由内核控制的位置。我们在这节课的最后可以看到在XV6中存在一个唯一的系统调用接入点，每一次应用程序执行ECALL指令，应用程序都会通过这个接入点进入到内核中。举个例子，不论是Shell还是其他的应用程序，当它在用户空间执行fork时，它并不是直接调用操作系统中对应的函数，而是调用ECALL指令，并将fork对应的数字作为参数传给ECALL。之后再通过ECALL跳转到内核。

下图中通过一根竖线来区分用户空间和内核空间，左边是用户空间，右边是内核空间。在内核侧，有一个位于syscall.c的函数syscall，每一个从应用程序发起的系统调用都会调用到这个syscall函数，syscall函数会检查ECALL的参数，通过这个参数内核可以知道需要调用的是fork（3.9会有相应的代码跟踪介绍）。



这里需要澄清的是，用户空间和内核空间的界限是一个硬性的界限，用户不能直接调用fork，用户的应用程序执行系统调用的唯一方法就是通过这里的ECALL指令。

假设我现在要执行另一个系统调用write，相应的流程是类似的，write系统调用不能直接调用内核中的write代码，而是由封装好的系统调用函数执行ECALL指令。所以write函数实际上调用的是ECALL指令，指令的参数是代表了write系统调用的数字。之后控制权到了syscall函数，syscall会实际调用write系统调用。



> 学生提问：操作系统在什么时候检查是否允许执行fork或者write？现在看起来应用程序只需要执行ECALL再加上系统调用对应的数字就能完成调用，但是内核在什么时候决定这个应用程序是否有权限执行特定的系统调用？

Frans教授：是个好问题。原则上来说，在内核侧实现fork的位置可以实现任何的检查，例如检查系统调用的参数，并决定应用程序是否被允许执行fork系统调用。在Unix中，任何应用程序都能调用fork，我们以write为例吧，write的实现需要检查传递给write的地址（需要写入数据的指针）属于用户应用程序，这样内核才不会被欺骗从别的不属于应用程序的位置写入数据。

学生提问：当应用程序表现的恶意或者就是在一个死循环中，内核是如何夺回控制权限的?

Frans教授：内核会通过硬件设置一个定时器，定时器到期之后会将控制权限从用户空间转移到内核空间，之后内核就有了控制能力并可以重新调度CPU到另一个进程中。我们接下来会看一些更加详细的细节。

学生提问：这其实是一个顶层设计的问题，是什么驱动了操作系统的设计人员使用编程语言C?

Frans教授：啊，这是个好问题。C提供了很多对于硬件的控制能力，比如说当你需要去编程一个定时器芯片时，这更容易通过C来完成，因为你可以得到更多对于硬件资源的底层控制能力。所以，如果你要做大量的底层开发，C会是一个非常方便的编程语言，尤其是需要与硬件交互的时候。当然，不是说你不能用其他的编程语言，但是这是C成功的一个历史原因。

学生提问：为什么C比C++流行的多？仅仅是因为历史原因吗？有没有其他的原因导致大部分的操作系统并没有采用C++?

Frans教授：我认为有一些操作系统是用C++写的，这完全是可能的。但是大部分你知道的操作系统并不是用C++写的，这里的主要原因是Linus不喜欢C++，所以Linux主要是C语言实现。