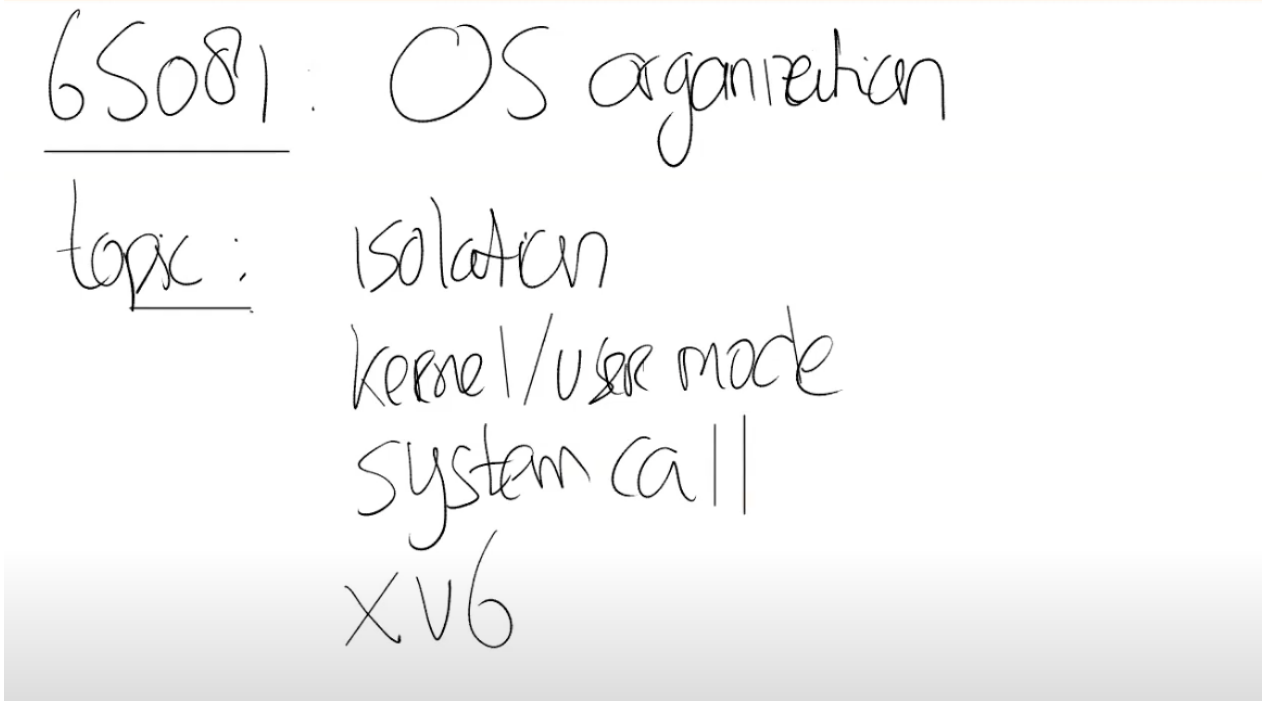
3.1 上一节课回顾

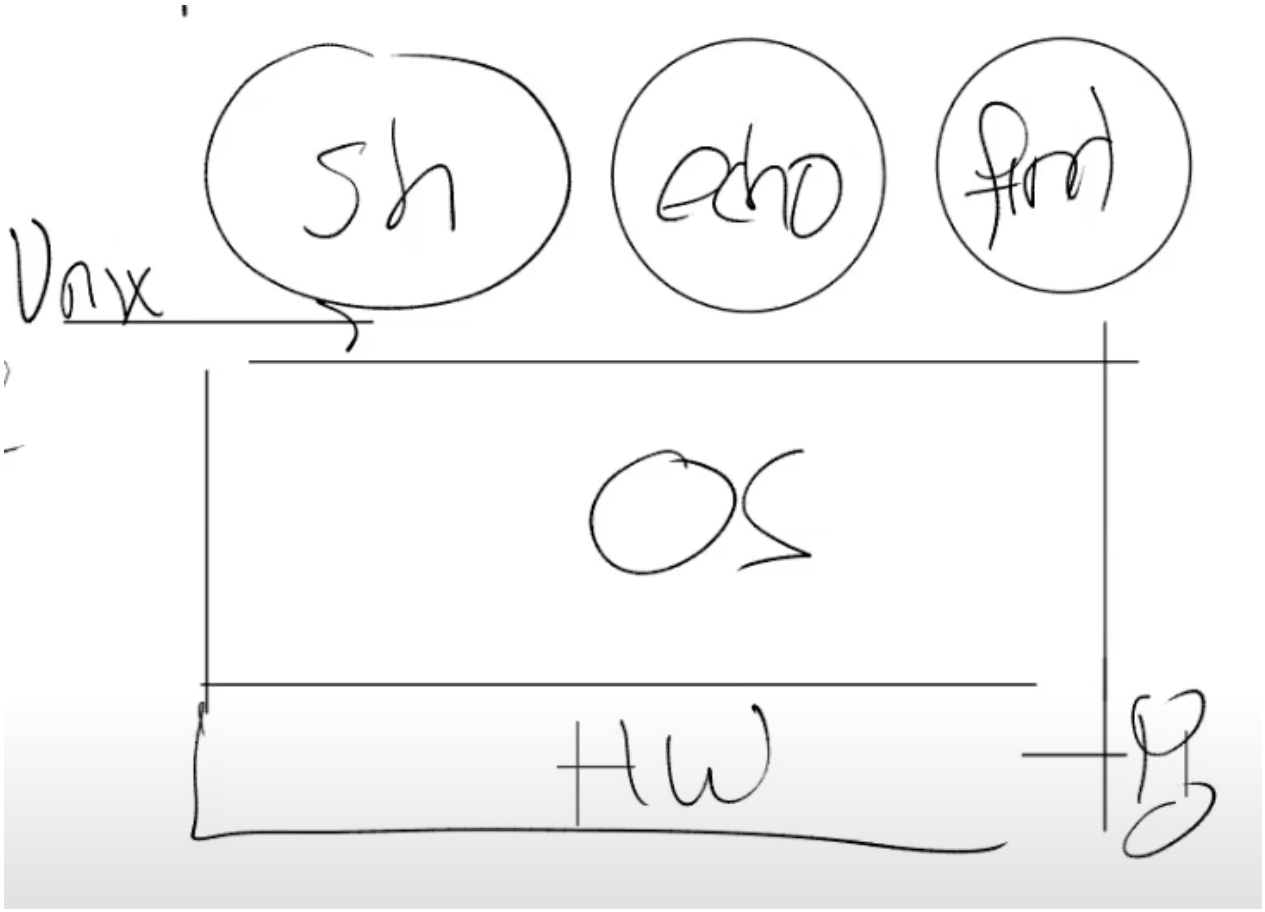
这节课的内容是操作系统的组织结构。今天我们主要讨论4个话题：

- Isolation。隔离性是设计操作系统组织结构的驱动力。
- Kernel和User mode。这两种模式用来隔离操作系统内核和用户应用程序。
- System calls。系统调用是你的应用程序能够转换到内核执行的基本方法，这样你的用户态应用程序才能使用内核服务。
- 最后我们会看到所有的这些是如何以一种简单的方式在XV6中实现。

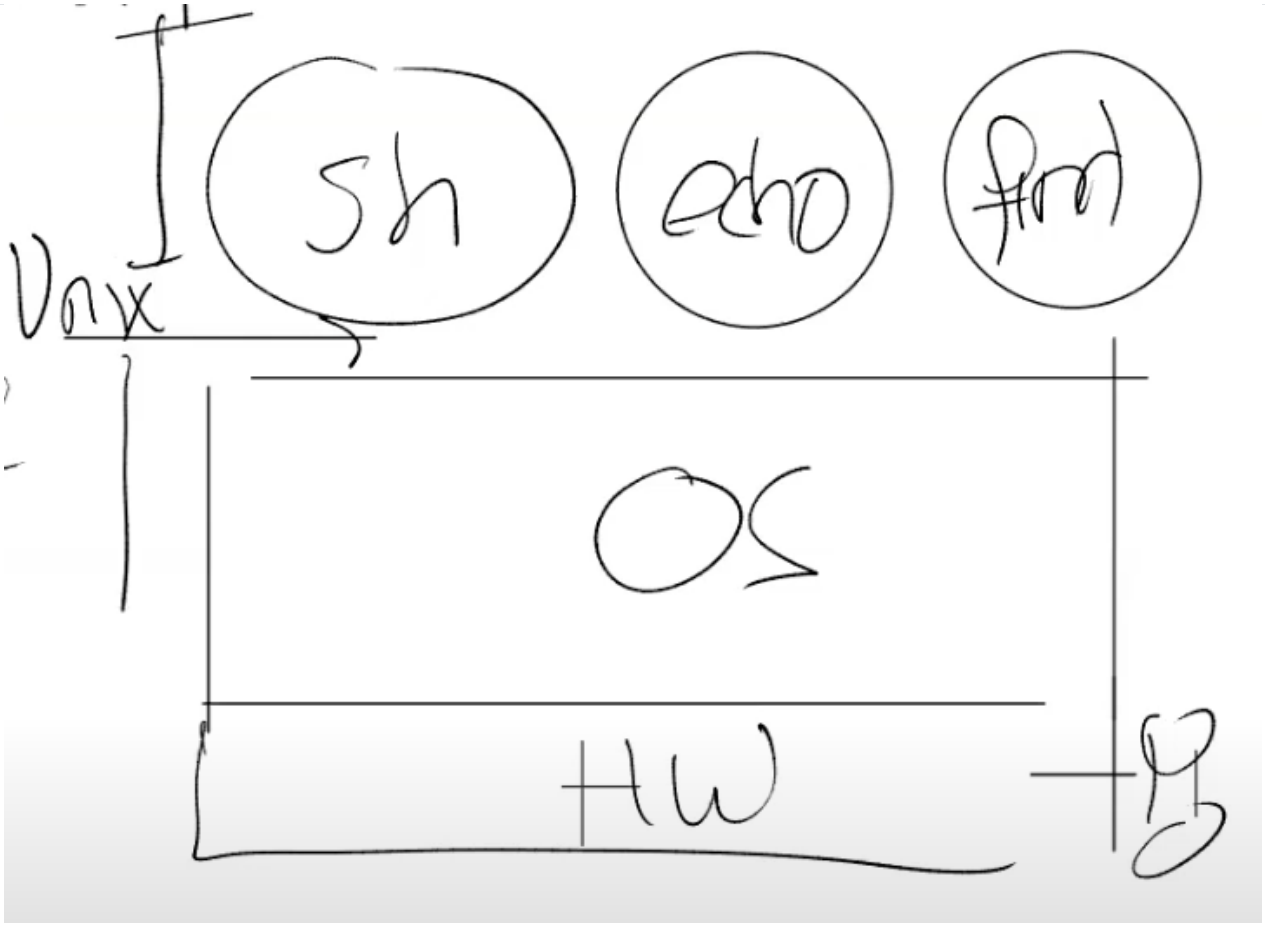


首先我们来复习一下上一节课。学习完第一课之后，你应该对操作系统的结构有个大致的认知。

- 首先，会有类似于Shell，echo，find或者任何你实现的工具程序，这些程序运行在操作系统之上。
- 而操作系统又抽象了一些硬件资源，例如磁盘，CPU。
- 通常来说操作系统和应用程序之前的接口被称为系统调用接口（System call interface），我们这门课程看到的接口都是Unix风格的接口。基于这些Unix接口，你们在lab1中，完成了不同的应用程序。



lab1主要集中在理解上图中的应用程序到操作系统内核之间的接口。而我们这节课以及后面的许多节课，都会实际查看，在操作系统内核中，这些接口是如何实现的。



实际上，在这个学期，我们都会花时间来理解如何实现操作系统接口（也就是系统调用接口），所以这是第一节有关这些内容的课。你们通过邮件和网站提出了很多非常棒的问题，我们现在还不能立即开始很多细节的讨论，因为讨论这些细节需要仔细检查操作系统的内部实现，而毕竟这是第一节讨论实现的课程。我们这节课会讨论很多东西，但是很多东西会在后面的课程变得更加清晰，因为我们会在后面的课程进行更加深入的讨论。不管怎样，如果有哪个地方不太清楚，请随时打断我并把问题提出来。

在我们进一步讨论之前，让我向你们提一个问题来开始今天的课程。你们在之前的utils lab（也就是lab1）中学到的最有意思的东西是什么？

我会先来回答这个问题。在实现lab1的代码之后，让我感到吃惊的一件事情是：我对于exec的使用频率比之前更多了。我之前主要用另一种与exec相同功能的方法，但是做完了exec实验之后，我发现使用exec更加的方便。所以，从那以后，我变成了一个更加激进的多的exec的使用者。同时我也好奇，你们的体验是怎么样？我会点几个名字，你们可以来说说你们对于lab1的体验。

> 这部分属于检查学生的实验完成情况，内容离散且对整体课程内容无影响，故跳过。

我希望你们能喜欢前一个实验，当然我也希望你们能喜欢后面的实验。如果你们还没有完成前一个实验的话，今天的课程的内容从某种程度上来说，也可以帮助你们继续完成System Call的实验。再次，你们可以随时打断我并提出问题。