

master ▾

MIT6.S081 / lec04-page-tables-frans / 4.1-ke-  
cheng-nei-rong-jian-jie.md

Go to file

...



huihongxiao GitBook: [master...



Latest commit 7bbde8e on Oct 31, 2020



History

1 contributor

30 lines (20 sloc) 2.85 KB

Raw

Blame



## 4.1 课程内容简介

今天的主题是虚拟内存（Virtual Memory）。具体来说，我们会介绍页表（page tables）。在后面的课程中，我们还会介绍虚拟内存相关的其他内容。

首先，我们会从一些问答开始今天的课程。我想问你们还记得课程6.004和课程6.033中虚拟内存的内容吗？我先来说一下我自己对于虚拟内存或者页表的认知吧。当我还是个学生并第一次听到学到这个词时，我认为它还是很直观简单的。这能有多难呢？无非就是个表单，将虚拟地址和物理地址映射起来，实际可能稍微复杂一点，但是应该不会太难。可是当我开始通过代码管理虚拟内存，我才知道虚拟内存比较棘手，比较有趣，功能也很强大。所以，希望在接下来的几节课和几个实验中，你们也能同样的理解虚拟内存。接下来我会问几个同学你们对于虚拟内存的理解是什么。

学生1：这就是用来存放虚拟内存到物理内存映射关系的。

学生2：这是用来保护硬件设备的。在6.004中介绍的，虚拟地址是12bit，最终会映射到某些16bit的物理地址。

学生3：通过虚拟内存，每个进程都可以有独立的地址空间。通过地址管理单元（Memory Management Unit）或者其他的技术，可以将每个进程的虚拟地址空间映射到物理内存地址。虚拟地址的低bit基本一样，所以映射是以块为单位进行，同时性能也很好。

学生4：虚拟地址可以让我们对进程隐藏物理地址。通过一些聪明的操控，我们可以读写虚拟地址，最后实际读写物理地址。

学生5：虚拟内存对于隔离性来说是非常基础的。每个进程都可以认为自己有独立的内存可以使用。

刚刚的回答中，很明显有两件事情是对的。这里存在某种形式的映射关系；并且映射关系对于实现隔离性来说有帮助。

隔离性是我们讨论虚拟内存的主要原因。在接下来的两节课，尤其当我们开始通过代码管理虚拟内存之后，我们可以真正理解它的作用。这节课我们会主要关注虚拟内存的工作机制，之后我们会看到如何使用这里的机制来实现非常酷的功能。

今天的内容主要是3个部分：

1. 首先我会讨论一下地址空间（Address Spaces）。这个在刚刚的问答中有的同学也提到了。
2. 接下来，我会谈一下支持虚拟内存的硬件。当然，我介绍的是RISC-V相关的硬件。但是从根本上来说，所有的现代处理器都有某种形式的硬件，来作为实现虚拟内存的默认机制。
3. 最后，我们会过一下XV6中的虚拟内存代码，并看一下内核地址空间 and 用户地址空间的结构。

6.S081: page tables (vm)

plan: 1) Address spaces  
2) paging hw (RISC-V)  
3) XV6 vm code + layout



以上就是这节课的主要内容。