《openEuler内核编程》

课程讲稿

第十一章 第1讲

虚拟化基础

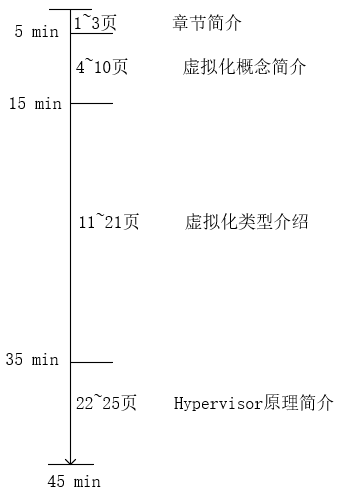
软件所制

第十一章 第1讲 虚拟化基础

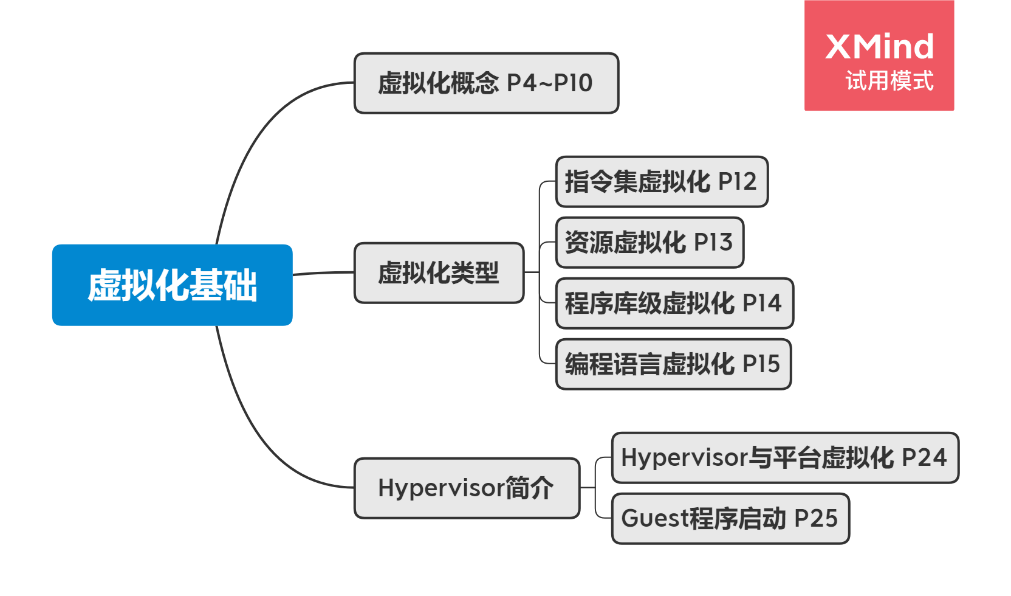
**学时：**1学时

**教学目的：** 了解虚拟化的概念及其在真实产品中的体现，了解虚拟机相关的基础概念。

**课程时间线：**

****

**知识框图：**





**课外参考读物：**

1. <https://zhuanlan.zhihu.com/p/19796979>
2. https://www.oschina.net/news/111410/linux-market-growth
3. https://lkml.org/lkml/2020/5/17/314
4. https://www.oschina.net/news/100042/linux-september-2018-stats

**PPT讲稿：**



虚拟化技术实际上是对计算机资源的一种抽象，在实际运行环境来模拟虚拟环境，对虚拟环境中的软件隐藏细节、提供逻辑上的计算机资源。虚拟化技术是云计算技术的重要基础，对硬件资源的虚拟化带来了更高的资源利用率，部署云服务更加灵活、安全。后面我们将会看到，虚拟化的思想在各处都有所体现，技术上的多样性并没有带来设计上的矛盾，虚拟化的实质都是将底层资源进行抽象，为上层提供一个虚拟的执行环境，与上层应用真实运行时所见到的完全（或部分）一致。

广义上来看，虚拟化指的是一种工程设计的概念，通过底层的模拟，为上层提供了虚拟的接口或资源，从而提供虚拟环境用于上层软件的执行。具体的虚拟化手段可以包括指令集层次的虚拟化、对计算资源的虚拟化、对程序库的虚拟化等各个方面。

狭义上说，虚拟化一般指的是虚拟机的相关技术，即在一台计算机上模拟运行多个操作系统，对这些操作系统隐藏底层物理硬件的真实细节，从而让各个操作系统透明地使用硬件，分享硬件并彼此之间相互隔离。

每一项技术的发展都最终受力于现实的需要，虚拟化技术能够蓬勃发展的原因主要是云服务在大数据时代的飞速发展，对服务器的资源利用率、服务隔离性、安全性等提出了更高的要求，传统的服务器架构灵活性较差、部署成本高、资源利用率不尽如人意，而虚拟化技术对客户系统的隔离性和客户系统之间的硬件共享，使这种需求得到一定程度的满足。

虚拟化技术的发展自上个世纪60年代就已经开始，在IBM时代的大型机上已经开始使用虚拟化技术，虚拟化技术在硬件昂贵的时期显著提升了计算资源的使用效率。到上世纪90年代，符合摩尔定律的硬件发展规律以及Wintel联盟对个人计算机普及的大力助推，使得硬件的性能提升、价格低廉、系统更加易用。虚拟化技术进入发展的低潮时期。

到21世纪，由于服务器计算资源的利用率低，同时在一个服务器系统上部署多个应用存在安全风险，大量部署物理基础架构又导致部署成本高昂，需要一个更好的方案。

此时，以Vmware公司为代表的虚拟软件公司通过提供提供虚拟化和云计算基础架构改的支持，开始兴起。虚拟化解决方案为提升服务器、桌面虚拟化提供了良好的方案。

在不同的层次，通过虚拟化进行设计，可以实现不同的效果。虚拟化从实现角度和实现技术上都可进行细致的分类，这里将虚拟化技术从实现角度上分为指令集虚拟化等五类。

指令集虚拟化通过代码翻译、资源映射等手段，能够实现：在A指令集的物理机器和系统上，运行基于B指令集的程序。例如QEMU模拟器可以用于在x86机器上调试基于RISC-V指令集的软件。而真正在运行时，实际上B指令集的程序是被逐指令进行翻译、模拟的，这一点对正在运行的程序本身来说完全透明。

对于操作系统而言，能够对特定的系统资源进行虚拟化，实现更高的资源利用率。操作系统介于用户程序与真实硬件之间，用户程序所见的“硬件环境”实际是由操作系统营造的，因此，操作系统的设计本身就带有虚拟化的思想，真实的物理设备与操作系统呈现给用户程序的逻辑设备，往往在资源数量上有一定差别，但用户程序不可感知，无法判断。

针对程序库进行虚拟化能够对应用程序隐藏操作系统相关的细节，Wine通过程序库级别的虚拟化，在类UNIX系统上提供了部分Windows的系统API支持，从而在类UNIX系统上能够运行win32程序。与此相近但更贴近内核层的是WSL，在Windows系统上为Linux应用提供虚拟的系统调用服务，以此来在Windows系统上运行Linux的某些发行版如Ubuntu。



编程语言虚拟化的典型例子是Java虚拟机，通过在编程环境中提供一套各个软硬件平台统一的API，JVM构建了一套该编程语言特有的指令集。能够对开发者隐藏底层细节，达到一次编译、处处运行的效果。

系统虚拟化也称为平台虚拟化，是广义的虚拟化概念中与狭义虚拟化概念最接近的部分，为云平台提供了基础支持。平台级虚拟化真正实现了客户操作系统与物理硬件的分离，也是虚拟化技术在服务器基础架构中的主要应用形式。



从虚拟平台的角度来看，虚拟化技术可以分为全虚拟化和半虚拟化，全虚拟化虚拟出的平台是现实中存在的，对于运行在该平台上的客户机来说，运行于虚拟平台或真实平台是不可感知的。半虚拟化则通过对客户机的源码进行修改，来让客户机使用虚拟化的资源



从虚拟化技术的支持层次来看，实现虚拟化技术既可以采用纯软件的方法，也可以借助硬件辅助来完成虚拟化。Intel等厂商在芯片上直接提供了对虚拟化的支持，硬件在执行虚拟机优先级的时候可以对敏感指令进行虚拟化执行。



实现结构上，虚拟化的实现既可以使用Hypervisor虚拟结构，也可以采用宿主模型。二者的主要区别在于Hypervisor虚拟在硬件资源之上不运行操作系统，而是由Hypervisor管理硬件资源；而宿主模型中的VMM则依托于一个在真实物理硬件上运行的宿主OS。两种模型的效率和功能具有一定差别，对硬件驱动的支持来说，Hypervisor对设备驱动的支持不如采用宿主模型来的多。



混合模型则综合了两种结构的特点，首先由Hypervisor管理硬件，然后Hypervisor让出部分对设备的控制权，由运行在特权虚拟机中额特权操作系统Domain 0来管理。由Domain 0和VMM共同为客户虚拟机提供虚拟环境。



平台虚拟化的好处主要体现在：提高服务器硬件资源利用率，降低服务器部署及管理成本。



Hypervisor的主要职能是管理底层的物理资源，为上层的Guest OS提供虚拟硬件环境，并调度各Guest OS所依赖的资源，如内存、CPU核心、I/O设备等。在某些Hypervisor（如KVM）的实现中，对于CPU的调度，宿主机进程与Guest OS是统一起来的，交由宿主OS的任务调度器来进行调度。



要启动一个Guest程序，Hypervisor需要提供一定的资源，包括足够的内存空间、网络设备（可能是虚拟的）、块设备等。