

资料1：关于hypervisor

<https://www.ibm.com/developerworks/cn/linux/l-hypervisor/>

1. KVM+QEMU才是一套完整的虚拟化方案，libvirt则用于管理虚拟机的API接口，可以通过virish等工具调用。
2. hypervisor是HOST和Guest之间的软件隔离层，包括：全虚拟化、准虚拟化。
全虚拟化中Guest并不知道自己运行在虚拟环境上，HOST完全通过软件模拟出硬件设备。
准虚拟化中，Guest知道自己运行在虚拟环境中，并且充分利用设备提供的高级指令让交互过程更高效。
3. hypervisor可以是各种形式的，举个栗子：KVM是Linux的一个内核模块，可以将Linux内核虚拟成一个hypervisor层。同样这么干的还有Lguest。

资料2：虚拟化概念

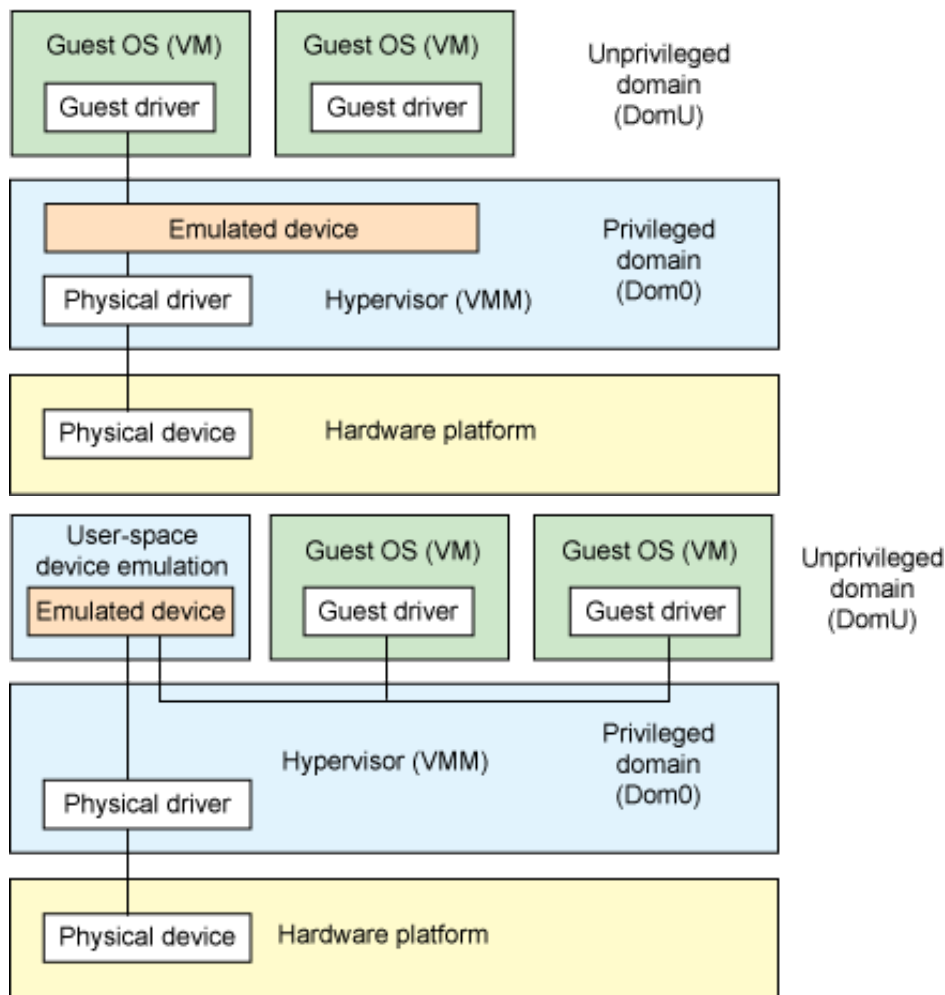
<https://www.ibm.com/developerworks/cn/linux/l-linuxvirt/>

1. full virtualization 和 paravirtualization 相比区别在于，是否修改客户机的操作系统。full virtualization 通过hypervisor捕获硬件实现进行模拟，而paravirtualization修改了客户机操作系统，使Guest和hypervisor的结合更加紧密。
2. 操作系统级的虚拟化 ---- Docker
3. QEMU 是 PC 仿真器，它们允许诸如 Windows 或 Linux 之类的操作系统在Linux 操作系统的用户空间中运行????

资料3：Linux 虚拟化和 PCI 透传技术

<https://www.ibm.com/developerworks/cn/linux/l-pci-passthrough/>

1. 模拟I/O设备有两种方式：一是在用户空间模拟（如KVM和Qemu的关系，两者是分离的），二是在hypervisor进行模拟（如WMWare）



2. 准虚拟化的优势在于，当前的物理设备大量支持透传技术，如Intel 将这种支持称为 Virtualization Technology for Directed I/O (VT-d)，而 AMD 称之为 I/O Memory Management Unit (IOMMU)。

FAQ

1.物理CPU的硬件支持

```
egrep -o '(vmx|svm)' /proc/cpuinfo
```

2.一个KVM的虚机在宿主机中对应一个qemu-kvm进程，而虚拟机的一个vCPU对应一个qemu-kvm进程的线程

在宿主机线可以查看qemu-kvm进程

```
ps -ef|grep qemu-kvm
```