**KVM简明手册**

KVM是一个基于linux内核的虚拟机解决方案。本文介绍KVM的使用方法，以及注意要点。

参考资料： [KVM虚拟机和QEMU](http://jianlee.ylinux.org/Computer/Software/kvm-qemu.html) [linux brctl](http://blog.csdn.net/HoO_flying/archive/2007/04/05/1553175.aspx)

[先决条件](http://blog.csdn.net/wuyao721/article/details/5790961#sec1)

[硬件环境](http://blog.csdn.net/wuyao721/article/details/5790961#sec2)

[CPU硬件虚拟化](http://blog.csdn.net/wuyao721/article/details/5790961#sec3)

[32位与64位](http://blog.csdn.net/wuyao721/article/details/5790961#sec4)

[安装KVM](http://blog.csdn.net/wuyao721/article/details/5790961#sec5)

[虚拟机连网](http://blog.csdn.net/wuyao721/article/details/5790961#sec6)

[brctl](http://blog.csdn.net/wuyao721/article/details/5790961#sec7)

[桥接](http://blog.csdn.net/wuyao721/article/details/5790961#sec8)

[安装以及运行虚拟机](http://blog.csdn.net/wuyao721/article/details/5790961#sec9)

[kvm常用命令](http://blog.csdn.net/wuyao721/article/details/5790961#sec10)

[安装虚拟操作系统](http://blog.csdn.net/wuyao721/article/details/5790961#sec11)

[远程登录](http://blog.csdn.net/wuyao721/article/details/5790961#sec12)

**先决条件**

**硬件环境**

x86(包括32位和64位)，就是我们常用的系统。

**CPU硬件虚拟化**

kvm需要系统CPU支持，下面命令如果有输出，说明CPU支持硬件虚拟化：

egrep "flags.\*:.\*(svm|vmx)" /proc/cpuinfo

**32位与64位**

下面是32bit与64bit和主机系统与虚拟操作系统的关系：

| **主机操作系统** | **虚拟机操作系统** | **是否支持** |
| --- | --- | --- |
| 32位 | 32位 | 是 |
| 32位 | 46位 | 否 |
| 64位 | 32位 | 是 |
| 64位 | 64位 | 是 |

总结为一句话：不能在32位主机上装64位的操作系统。

**安装KVM**

yao@debian:~$ apt-get install kvm qemu uml-utilities

一般情况下，安装好kvm后能就自动加载内核模块，你也可以手动加载：

yao@debian:~$ modprobe kvm\_intel #or kvm\_amd

**虚拟机连网**

虚拟机连网是头等大事，所以我们先解决这个问题。通过桥接可以实现联网，主要用到的命令有：brctl,ifconfig,tunctl,route

**brctl**

brctl是一个以太网桥接工具，常见的用法有：

| **命令原型** | **例子** | **说明** |
| --- | --- | --- |
| brctl show | brctl show | 显示已有网桥 |
| brctl addbr <bridge> | brctl addbr br0 | 增加网桥br0 |
| brctl delbr <bridge> | brctl delbr br0 | 删除网桥br0 |
| brctl addif <bridge> <device> | brctl addif br0 eth0 | 将接口eth0接到网桥br0 |
| brctl delif <bridge> <device> | brctl delif br0 eth0 | 从网桥上删除一个接口 |

如果发现无法删除网桥，那么可能是还没有将其关闭：

yao@twomoon:~$ sudo brctl delbr br0

bridge br0 is still up; can't delete it

yao@twomoon:~$ sudo ifconfig br0 down

yao@twomoon:~$ sudo brctl delbr br0

**桥接**

说了这么多，到这才是重点。首先，将物理网卡桥接：

yao@twomoon:~$ ifconfig -a |grep eth

eth0 Link encap:Ethernet HWaddr 90:fb:a6:14:cd:42

yao@twomoon:~$ brctl addbr br0

yao@twomoon:~$ brctl addif br0 eth0

yao@twomoon:~$ ifconfig eth0 0.0.0.0

yao@twomoon:~$ ifconfig br0 192.168.1.51 up

这样又多了一个网络设备br0：

yao@twomoon:~$ LANG=C ifconfig -a |grep Ethernet

br0 Link encap:Ethernet HWaddr 2a:24:d3:aa:99:e7

eth0 Link encap:Ethernet HWaddr 90:fb:a6:14:cd:42

如果没有弄明白为什么，可以上网找找桥接的资料和brctl的用法。

当然不能每次开机后都要手工输入这么多命令。你可以将这些命令写成一个脚本，每次开机就执行这个脚本：

#!/bin/bash

# networking.sh

# change eth0 to your network interface(eg. eth1)

ifconfig lo 127.0.0.1 up

if [ ! -z "`ifconfig -a | grep br0`" ]; then

ifconfig br0 down

brctl delif br0 eth0

brctl delbr br0

fi

brctl addbr br0

brctl addif br0 eth0

ifconfig eth0 0.0.0.0 up

ifconfig br0 192.168.1.51 up

route add default gw 192.168.1.1

exit 0

也可以写到网卡的配置文件配置文件(/etc/network/interfaces)里，效果是一样的：

auto lo

iface lo inet loopback

auto br0

iface br0 inet static

address 192.168.1.51

netmask 255.255.255.0

gateway 192.168.1.1

bridge\_ports eth0

**安装以及运行虚拟机**

**kvm常用命令**

| **参数** | **示例** | **说明** |
| --- | --- | --- |
| -hda | -hda /data/windows.img | 指定windows.img作为硬盘镜像 |
| -cdrom | -hda /data/windows.iso | 指定windows.iso作为光盘镜像 |
| -boot | -boot c | 从硬件启动 |
|  | -boot d | 从光盘启动 |
| -m | -m 512 | 分配512M内存给虚拟系统 |
| -vnc | -vnc :0 | 作为vnc服务器 |
| -cpu | -cpu ? | 列出支持的CPU |
|  | -cpu core2duo | 指定CPU为core2duo |
| -smp | -smp 2 | 指定虚拟机有2个CPU |
| -net | -net nic | 为虚拟机网卡（默认为tap0） |
|  | -net tap | 系统分配tap设备（默认为tap0）[1](http://blog.csdn.net/wuyao721/article/details/5790961" \l "fn.1) |
|  | -net nic -net tap | 将虚拟机的网卡eth0连接真机里的tap0 |

**安装虚拟操作系统**

生成镜像文件：

yao@twomoon:~$ sudo kvm-img create windows2003-x86.img 25G -f "vmdk"

安装系统：

yao@twomoon:~$ sudo kvm -boot d -m 512 -hda ~/kvm/windows2003-x86.img -cdrom ~/software/wzmb2003.iso

运行系统：

yao@twomoon:~$ sudo kvm -boot c -m 512 -hda ~/kvm/windows2003-x86.img -net nic -net tap

可以发现多了一个网络设备tap0，这是系统自动创建的。

yao@twomoon:~$ LANG=C ifconfig -a |grep Ethernet

br0 Link encap:Ethernet HWaddr 2a:24:d3:aa:99:e7

eth0 Link encap:Ethernet HWaddr 90:fb:a6:14:cd:42

tap0 Link encap:Ethernet HWaddr 2a:24:d3:aa:99:e7

**远程登录**

如果虚拟机是linux，那么就用ssh；如果是windows，那么就用rdesktop。

yao@debian:~$ rdesktop -u username -p passwd 192.168.1.52 -f &

记得用Ctrl-Alt-Enter从全屏中切回来。

在计算机网络中，TUN与TAP是操作系统内核中的虚拟网络设备。不同于普通靠硬件网路板卡实现的设备，这些虚拟的网络设备全部用软件实现，并向运行与操作系统上的软件提供与硬件的网络设备完全相同的功能。 TAP 等同于一个以太网设备，它操作第二层数据包如以太网数据帧。TUN模拟了网络层设备，操作第三层数据包比如IP数据封包。操作系统通过TUN/TAP设备向绑定该设备的用户空间的程序发送数据，反之，用户空间的程序也可以像操作硬件网络设备那样，通过TNU/TAP设备发送数据。在后种情况下，TUN/TAP设备向操作系统的网络栈投递（或“注入”）数据包，从而模拟从外部接受数据的过程。