

www.dynamips.cn

整理：小漏

致力于 Dynamips 和 Pemu 的应用！

Pemu 是 PIX 模拟器 作者正在更新 即将加入 IDS 模块！

Qemu 中文使用手册

Qemu 使用手册

快速启动

在我们下载并解压 linux 镜像 (linux.img) 以后, 我们可以输入下面的命令来启动:

```
qemu linux.img
```

这样 Linux 就会启动并会展示给我们一个提示.

调用

用法: `qemu [option] [disk_image]`

`disk_image` 是代表 IDE 的硬盘的硬盘镜像.

一般选项:

`-M machine`

选择模拟的机器 (我们可以输入 `-M?` 得到一个模拟的机器列表)

`-fda file`

`-fdb file`

使用 `file` 作为软盘镜像. 我们也可以通过将 `/dev/fd0` 作为文件名来使用主机软盘.

`-hda file`

`-hdb file`

`-hdc file`

`-hdd file`

使用 `file` 作为硬盘 0, 1, 2, 3 的镜像.

`-cdrom file`

使用文件作为 CD-ROM 镜像 (但是我们不可以同时使用 `'-hdc'` 和 `'-cdrom'`). 我们可以通过使

用 `/dev/cdrom` 作为文件名来使用主机的 CD-ROM.

`-boot [a|c|d]`

由软盘(a), 硬盘(c)或是 CD-ROM(d). 在默认的情况下由硬盘启动.

`-snapshot`

写入临时文件而不是写入磁盘镜像文件. 在这样的情况下, 并没有写回我们所使用的磁盘镜像文件. 然而我们却可以通过按下 `C-a s` 来强制写回磁盘镜像文件.

`-m megs`

设置虚拟内存尺寸为 `megs M` 字节. 在默认的情况下为 128M.

`-smp n`

模拟一个有 `n` 个 CPU 的 SMP 系统. 为 PC 机为目标, 最多可以支持 255 个 CPU.

`-nographic`

在通常情况下, Qemu 使用 SDL 来显示 VGA 输出. 使用这个选项, 我们可以禁止所有的图形输出, 这样 Qemu 只是一个简单的命令行程序. 模拟的串口将会重定向到命令行. 所以, 我们仍然可以在 Qemu 平台上使用串口命令来调试 Linux 内核.

`-k language`

使用键盘布局语言(例如 `fr` 为法语). 这个选项只有在不易得到 PC 键盘的情况下使用. 我们在 PC/Linux 或是 PC/Windows 主机不需要使用这个选项. 可用的布局如下:

```
ar de-ch es fo fr-ca hu ja mk no pt-br sv
da en-gb et fr fr-ch is lt nl pl ru th
de en-us fi fr-be hr it lv nl-be pt sl tr
```

默认的为 `en-us`

`-audio-help`

这个选项将会显示声音子系统的帮助: 驱动列表以及可调用的参数.

`-soundhw card1,card2 or -soundhw all`

允许声音并选择声音硬件. 使用?可以列出所有可用的声音硬件

`qemu -soundhw sb16,adlib hda`

`qemu -soundhw es1370 hda`

`qemu -soundhw all hda`

`qemu -soundhw ?`

`-localtime`

设置时钟为本地时间(默认为 UTC 时间). 如果在 MS-DOS 或是 Windows 上这个选项则需要正确的日期.

`-full-screen`

以全屏方式启动.

`-pidfile file`

在 `file` 文件中存许 Qemu 的进程 PID. 如果我們是由脚本启动的, 这个选项是相当有用的.

`-win2k-hack`

当安装 Windows 2000 时可以使用这个选项来避免磁盘错误. 在安装上 Windows 2000 系统, 我们就不再需要这个选项(这个选项降低 IDE 的传输速度).

USB 选项:

`-usb`

允许 USB 驱动(很快就将成为默认的选项)

`-usbdevice devname`

添加 USB 设备名. 我们可以查看监视器命令 `usb_add` 来得到更为详细的信息.

网络选项:

```
-net nic[,vlan=n][,macaddr=addr]
```

创建一个新的网卡并与 VLAN n (在默认的情况下 n=0) 进行连接. 在 PC 机上, NIC 当前为 NE2000. 作为可选项的项目, MAC 地址可以进行改变. 如果没有指定 -net 选项, 则会创建一个单一的 NIC.

```
-net user[,vlan=n]
```

使用用户模式网络堆栈, 这样就不需要管理员权限来运行. 如果没有指定 -net 选项, 这将是默认的情况.

```
-net tap[,vlan=n][,fd=h][,ifname=name][,script=file]
```

将 TAP 网络接口 name 与 VLAN

n 进行连接, 并使用网络配置脚本 file 进行配置. 默认的网络配置脚本为 /etc/qemu-ifup. 如果没有指定 name, OS 将会自动指定一个. fd=h 可以用来指定一个已经打开的 TAP 主机接口的句柄. 例如:

```
qemu linux.img -net nic -net tap
```

下面的是一个更为复杂的例子 (两个 NIC, 每一个连接到一个 TAP 设备):

```
qemu linux.img -net nic,vlan=0 -net tap,vlan=0,ifname=tap0 \
                -net nic,vlan=1 -net tap,vlan=1,ifname=tap1
```

```
-net socket[,vlan=n][,fd=h][,listen=[host]:port][,connect=host:port]
```

使用 TCP socket 将 VLAN

n 与远程的另一个 Qemu 虚拟机的 VLAN 进行连接. 如果指定了 listen, Qemu 将在 port 端口监听连入请求 (host 是可选的),

connect 可以用来使用 listen 选项与另一个 Qemu 实例进行连接. fd=h 指定了一个已经打开的 TCP socket. 例如:

```
# launch a first QEMU instance
```

```
qemu linux.img -net nic,macaddr=52:54:00:12:34:56 -net socket,listen=:1234
```

```
# connect the VLAN 0 of this instance to the VLAN 0 of the first instance
```

```
qemu linux.img -net nic,macaddr=52:54:00:12:34:57 -net
socket,connect=127.0.0.1:1234
```

```
-net socket[,vlan=n][,fd=h][,mcast=maddr:port]
```

创建一个 VLAN n, 并使用 UDP 多址通信套接口与其他的 QEMU 虚拟机进行共享, 尤其是对于每一个使用多址通信地址和端口的 QEMU 使用同一个总线.

PS: 上面是关键命令

在这里我们要注意以下几点:

- 1 几个 QEMU 可以运行在不同的主机上但却使用同一个总线 (在这里假设为这些主机设置了正确的多址通信)
- 2 mcast 支持是与用户模式 Linux 相兼容的.
- 3 使用 fd=h 指定一个已经打开的 UDP 多址通信套接口.

例如:

```
# launch one QEMU instance
```

```
qemu linux.img -net nic,macaddr=52:54:00:12:34:56 -net
```

```
socket,mcast=230.0.0.1:1234
```

```
# launch another QEMU instance on same "bus"
```

```
qemu linux.img -net nic,macaddr=52:54:00:12:34:57 -net
```

```
socket,mcast=230.0.0.1:1234
```

```
# launch yet another QEMU instance on same "bus"
```

```
qemu linux.img -net nic,macaddr=52:54:00:12:34:58 -net
```

```
socket,mcast=230.0.0.1:1234
```

下面的为用户模式 Linux 的例子:

```
# launch QEMU instance (note mcast address selected is UML's default)
```

```
qemu linux.img -net nic,macaddr=52:54:00:12:34:56 -net
```

```
socket,mcast=239.192.168.1:1102
```

```
# launch UML
```

```
/path/to/linux ubd0=/path/to/root_fs eth0=mcast
```

```
-net none
```

表明没有网络设备需要进行配置. 如果没有指定 `-net` 选项, 则会用来覆盖活跃的默认配置.

```
-tftp prefix
```

当 使用用户模式网络堆栈, 激活一个内置的 TFTP 服务器. 所有的以 `prefix` 开始的文件将会使用一个 TFTP 客户端从主机下载到本地. 在本地的 TFTP 客户端必须以二进制模式进行配置(使用 Unix 的 TFTP 客户端的 `bin` 命令). 在客户机上的主机 IP 地址如通常的 10.0.2.2.

```
-smb dir
```

当使用用户模式的网络堆栈, 激活一个内建的 SMB 服务器, 这样 Windows 系统就可以透明的访问主机的 `dir` 目录中的文件. 在客户机的 Windows 系统中, 下面的行:

[10.0.2.4](#) smbserver

必须添加在文件 `C:\WINDOWS\LMHOSTS'` (for windows

9x/Me) 或者是 `C:\WINNT\SYSTEM32\DRIVERS\ETC\LMHOSTS` (Windows

NT/2000). 然后可以用 `\\smbserver\qemu` 的方式访问 `dir`. 在这里我们要注就是在主机系统中必须安有 SAMBA 服务器.

```
-redir [tcp|udp]:host-port:[guest-host]:guest-port
```

当 使用用户模式网络栈, 将连接到主机端口 `host-port` 的 TCP 或是 UDP 连接重定向到客户机端口 `guest-port` 上. 如果没有指定客户机端口, 他的值为 10.0.2.15 (由内建的 DHCP 服务器指定默认地址). 例如: 要重定向从 `screen 1` 到客户机 `screen`

0 的 X11 连接, 我们可以使用下面的方法:

```
# on the host
```

```
qemu -redir tcp:6001::6000 [...]
```

```
# this host xterm should open in the guest X11 server
```

```
xterm -display :1
```

To redirect telnet connections from host port 5555 to telnet port on the guest, use the following:

on the host

```
qemu -redir tcp:5555::23 [...]
```

```
telnet localhost 5555
```

然后当我们在主机 `telnet localhost 5555` 上使用时，我们连接到了客户机的 `telnet` 服务器上。

Linux 启动相关：

当我们使用这些选项时，我们可以使用一个指定的内核，而没有将他安装在磁盘镜像中。这对于简单的测试各种内核是相当有用的。

```
`-kernel bzImage'
```

使用 `bzImage` 作为内核映像。

```
`-append cmdline'
```

使用 `cmdline` 作为内核的命令行。

```
-initrd file'
```

使用 `file` 作为初始的 `ram` 磁盘。

调试选项：

```
`-serial dev'
```

重定向虚拟串到主机的设备 `dev`。可用的设备如下：

`vc`

虚拟终端

`pty`

(Linux) 伪 TTY (自动分配一个新的 TTY)

`null`

空设备

`/dev/XXX`

(Linux) 使用主机的 `tty`。例如，`'/dev/ttyS0'`。主机的串口参数通过模拟进行设置。

`/dev/parportN`

(Linux) 使用主机的并口 `N`。当前只可以使用 `SPP` 的并口特征。

`file: filename`

将输出写入到文件 `filename` 中。没有字符可读。

`stdio`

(Unix) 标准输入/输出

`pipe: filename`

(Unix) 有名管道 `filename`

在图形模式下的默认设备为 `vc`，而在非图形模式下为 `stdio`。这个选项可以被多次使用，最多可以模拟 4 个串口。

```
'-parallel dev'
```

重定向虚拟并口到主机的设备 `dev` (与串口相同的设备)。在 Linux 主机上，`'/dev/parportN'` 可以被用来使用与相应的并口相连的硬件设备。这个选项可以使用多次，最多可以模拟 3 个并口。

```
`-monitor dev'
```

重定向监视器到主机的设备 `dev` (与串口相同的设备)。在图形模式下的默认设备为 `vc`，而在非图形模式下为 `stdio`。

```
'-s'
```

等待 `gdb` 连接到端口 1234.

``-p port``

改变 gdb 连接端口。

``-S``

在启动时并不启动 CPU（我们必须在监视器中输入 'c'）

``-d``

输出日志到/tmp/qemu.log

``-hdachs c,h,s,[,t]``

强 制硬盘 0 的物理参数 ($1 \leq c \leq 16383$, $1 \leq h \leq 16$, $1 \leq s \leq$

63), 并且可以选择强制 BIOS 的转换模式 ($t=none$, lba or

auto). 通常 QEMU 可以检测这些参数. 这个选项对于老的 MS-DOS 磁盘映像是相当有用的.

``-std-vga``

模拟一个 Bochs VBE 扩展的标准 VGA 显卡(默认情况下为 Cirrus Logic GD5446 PCI VGA)

``-loadvm file``

从一个保存状态启动.

组合键

在图形模拟时, 我们可以使用下面的这些组合键:

Ctrl-Alt-f

全屏

Ctrl-Alt-n

切换虚拟终端 'n'. 标准的终端映射如下:

1 目标系统显示

2 监视器

3 串口

Ctrl-Alt

抓取鼠标和键盘

在虚拟控制台中, 我们可以使用 Ctrl-Up, Ctrl-Down, Ctrl-PageUp 和 Ctrl-PageDown 在屏幕中进行移动.

在模拟时, 如果我们使用 ``-nographic`` 选项, 我们可以使用 Ctrl-a h 来得到终端命令:

Ctrl-a h

打印帮助信息

Ctrl-a x

退出模拟

Ctrl-a s

将磁盘信息保存入文件(如果为 `-snapshot`)

Ctrl-a b

发出中断

Ctrl-a c

在控制台与监视器进行切换

Ctrl-a Ctrl-a

发送 Ctrl-a

磁盘映像

从 0.6.1 起, QEMU 支持多种磁盘映像格式, 包括增长的磁盘映像, 压缩与加密的磁盘映像.

我们可以用下面的命令来创建一个磁盘映像:

`qemu-img create myimage.img mysize`

这里 myimage.img 是磁盘映像的文件名, 而 mysize 是以 K 表示的尺寸. 我们可以使用 M 前缀来使用 M 表示尺寸或是 G 作为前缀使用 G 表示尺寸.

qemu-img 的调用方法:

方法: `qemu-img command [command options]`

可以支持下面的一些命令:

``create [-e] [-b base_image] [-f fmt] filename [size]``

``commit [-f fmt] filename``

``convert [-c] [-e] [-f fmt] filename [-O output_fmt] output_filename``

``info [-f fmt] filename``

命令参数

filename

磁盘映像文件名.

base_image

只读的磁盘映像, 可以作为拷贝到写映像的基础. 写映像上的拷贝只存储修改的数据.

fmt

磁盘映像格式. 在大多数情况下可以自动检测. 可以支持下面的格式:

raw

raw 磁盘格式(默认). 这种格式有简单并且易于移植到其他模拟器的优点. 如果我们的文件系统支持 holes(例如在 Linux 上的 ext2 或是 ext3), 然后只有写入的部分保持空白. 使用 `qemu-img info` 来得到映像使用的实际的大小或是在 Unix/Linux 上使用 `ls -ls`.

qcow

QEMU 映像格式. 最通用的格式. 使用他可以获得较小的映像(如果我们的文件系统不支持 holes, 例如在 Windows 上, 这是相当有用的), 可以选用 AES 加密或是基于 zlib 的压缩.

cow

在写映像格式上的用户模式的 Linux 拷贝. 在 QEMU 中作为增长的映像格式使用. 这个选项只是为了与以前版本的兼容, 并不能在 Win32 上使用.

vmdk

VMware 3 或是 4 兼容的映像格式.

cloop

Linux 压缩的循环映像, 重用直接压缩的 CD-ROM 映像.

size

以 K 表示的磁盘映像的尺寸. 同时可以支持 M 或是 G 作为前缀.

output_filename

目的磁盘映像文件名

output_fmt

目标格式

-c

表明目标映像必须是压缩的(只是 qcow 格式)

-e

表明目标映像必须是加密的(只是 qcow 格式)