# Qemu-USB存儲設備仿真

[kingtj](https://www.twblogs.net/u/5bb270272b71770e645de389) 2018-10-10 03:09

<https://www.twblogs.net/a/5bbcfcdb2b71776bd30bc2d2>

qemu既支持仿真虛擬的usb存儲設備，也支持連接實際的設備（如U盤）。相關的命令參數為：

-usb

-device usb-storage,drive=drive\_id

-device usb-uas

-device usb-bot

-device usb-host,hostbus=bus,hostaddr=addr

-device usb-host,vendorid=vendor,productid=product

* “**-usb**” 使能usb總線，如果系統已添加usb總線，則不需要這個參數
* “**usb-storage**” 仿真使用bulk-only傳輸協議的虛擬設備（絕大部分u盤採用的傳輸方式），只支持單一邏輯單元
* “**usb-uas**” 仿真使用UAS（USB attached SCSI）傳輸協議的虛擬設備，支持多邏輯單元（MLUN）
* “**usb-bot**” 也是仿真使用bulk-only傳輸協議的虛擬設備，支持多邏輯單元
* “**usb-host**” 連接實際的usb存儲設備

下面介紹常用的 usb-storage 和 usb-host 方式。

## 使用虛擬設備

首先創建設備文件，可以選擇固定大小的raw格式或者大小動態增長的qcow2等格式。  
使用raw格式的好處是可以直接在PC上對設備文件進行分區及格式化等操作，只是佔用較多的存儲空間。  
而使用qcow2格式則只需佔用很小的存儲空間，適合仿真大容量設備，只是得掛載後才能分區或格式化。

## raw格式

創建設備：

|  |
| --- |
| $ qemu-img create -f raw disk\_01.img 32M  Formatting 'disk\_01.img', fmt=raw size=33554432  $  $ ls -l disk\_01.img  -rw-r--r-- 1 cjz cjz 33554432 Oct 6 21:18 disk\_01.img |

格式化設備：

|  |
| --- |
| $ mkfs.vfat disk\_01.img  mkfs.fat 3.0.26 (2014-03-07) |

qemu掛載設備：

|  |
| --- |
| qemu <other options ...> \  -usb \  -drive if=none,format=raw,id=disk1,file=disk\_01.img \  -device usb-storage,drive=disk1 |

還可以添加usb主控制器設備（root hub）來增加usb總線，並掛載設備到新總線上，比如：

|  |
| --- |
| qemu <other options ...> \  -drive **if**=none,format=raw,**id**=disk1,file=disk\_01.img \  -device ich9-usb-ehci1,**id**=usb \  -device usb-storage,bus=usb.0,drive=disk1 |

使用命令 “**qemu -device ?**” 可以查看支持的虛擬設備信息

## qcow2格式

創建設備：

|  |
| --- |
| $ qemu-img create -f qcow2 disk\_02.img 4G  Formatting 'disk\_02.img', fmt=qcow2 size=4294967296 encryption=off cluster\_size=65536 lazy\_refcounts=off  $  $ ls -l disk\_02.img  -rw-r--r-- 1 cjz cjz 197120 Oct 6 21:26 disk\_02.img |

可以看到4GB大小的設備實際只佔用很小存儲空間（將隨著設備的使用情況而動態增加）。  
qemu掛載方法與上面相同，只是把格式改成 “**format=qcow2**” ：

|  |
| --- |
| qemu <other options ...> \  -usb \  -drive if=none,format=qcow2,id=disk2,file=disk\_02.img \  -device usb-storage,drive=disk2 |

## 使用實際設備

首先使用 **lsusb** 命令查看連接到PC的usb設備信息，比如：

|  |
| --- |
| $ lsusb  ...  Bus 002 Device 005: ID 05e3:0736 Genesys Logic, Inc. microSD Reader/Writer  ... |

1. qemu掛載方式1： 使用廠家和產品的唯一標識碼

|  |
| --- |
| qemu <other options ...> \  -usb \  -device usb-host,vendorid=0x05e3,productid=0x0736 |

即對應lsusb顯示的 “**ID 05e3:0736**”。

1. qemu掛載方式2： 使用總線地址信息

|  |
| --- |
| qemu <other options ...> \  -usb \  -device usb-host,hostbus=2,hostaddr=5 |

即對應lsusb顯示的 “**Bus 002 Device 005**”。

使用實際設備需要root權限，否則會出現下面的錯誤：

|  |
| --- |
| libusbx: error [\_get\_usbfs\_fd] libusbx couldn't open USB device /dev/bus/usb/002/005: Permission denied libusbx: error [\_get\_usbfs\_fd] libusbx requires write access to USB device nodes. |

可以 “sudo” 運行qemu命令，或者創建udev規則來改變設備的訪問權限，比如：  
創建udev規則文件 **/etc/udev/rules.d/qemu-usb.rules**，並添加以下內容：

|  |
| --- |
| SUBSYSTEM=="usb", ATTRS{idVendor}=="05e3", ATTRS{idProduct}=="0736", MODE="0666" |

## QEMU測試

以實際設備為例，運行命令：

|  |
| --- |
| qemu <other options ...> \  -device ich9-usb-ehci1,id=usb \  -device usb-host,vendorid=0x05e3,productid=0x0736,bus=usb.0 |

這是qemu內核運行後的打印信息：

|  |
| --- |
| usb 1-1: **new** high-speed USB device number 2 using ehci-pci  usb 1-1: **New** USB device found, idVendor=05e3, idProduct=0736  usb 1-1: **New** USB device strings: Mfr=3, Product=4, SerialNumber=2  usb 1-1: Product: USB Storage  usb 1-1: Manufacturer: Generic  usb 1-1: SerialNumber: 000000000272  ...  / #  / # lsusb  Bus 001 Device 002: ID 05e3:0736  Bus 001 Device 001: ID 1d6b:0002  / #  / # cat /sys/bus/usb/devices/1-1/1-1:1.0/bInterfaceClass  08  / # cat /sys/bus/usb/devices/1-1/1-1:1.0/bInterfaceSubClass  06  / # cat /sys/bus/usb/devices/1-1/1-1:1.0/bInterfaceProtocol  50  / # |

如果內核已包含 CONFIG\_USB\_STORAGE，則會自動檢測usb存儲設備，否則加載該模塊：

該模塊還依賴於SCSI模塊： **CONFIG\_SCSI** 及 **CONFIG\_BLK\_DEV\_SD**  
另外要正確識別FAT格式化的設備，還需支持FAT文件系統 **CONFIG\_FAT\_FS** 及 **CONFIG\_VFAT\_FS**

**cat /proc/filesystems** 可以查看當前內核支持的文件系統列表

|  |
| --- |
| / # insmod /lib/modules/3.12.74/usb-storage.ko  usb\_storage: **module** verification failed: signature **and**/**or** required key missing - tainting kernel  usb-storage 1-1:1.0: USB Mass Storage device detected  scsi2 : usb-storage 1-1:1.0  usbcore: registered **new** interface driver usb-storage  / # scsi 2:0:0:0: Direct-Access Generic STORAGE DEVICE 0272 PQ: 0 ANSI: 0  sd 2:0:0:0: Attached scsi generic sg2 type 0  sd 2:0:0:0: [sdb] 7744512 512-byte logical blocks: (3.96 GB/3.69 GiB)  sd 2:0:0:0: [sdb] Write Protect **is** off  sd 2:0:0:0: [sdb] No Caching mode page found  sd 2:0:0:0: [sdb] Assuming drive cache: write through  sd 2:0:0:0: [sdb] No Caching mode page found  sd 2:0:0:0: [sdb] Assuming drive cache: write through  sdb: sdb1  sd 2:0:0:0: [sdb] No Caching mode page found  sd 2:0:0:0: [sdb] Assuming drive cache: write through  sd 2:0:0:0: [sdb] Attached SCSI removable disk |

**sdb** 為整個usb存儲設備，sdb1 為設備上的的第一個分區。

|  |
| --- |
| / # ls -l /dev/sdb\*  brw------- 1 0 0 8, 16 Oct 7 07:59 /dev/sdb  brw------- 1 0 0 8, 17 Oct 7 07:59 /dev/sdb1  / #  / # cat /sys/block/sdb/size  7744512  / # cat /sys/block/sdb/sdb1/size  7742431  / #  / # mount -t vfat /dev/sdb1 /tmp  / #  / # mount | grep sdb  /dev/sdb1 on /tmp **type** vfat (rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)  / #  / # df -h | grep sdb  /dev/sdb1 3.7G 100.0K 3.7G 0% /tmp  / # |