Emulação de sistemas com o QEMU

Máquinas virtuais

Sempre quis rodar o Linux dentro do Linux? Ou que tal o DOS dentro do pingüim? O QEMU é um programa de código aberto que permite a emulação completa de hardware dentro de seu PC.

por Fabrizio Ciacchi

Peter Zelei – www.sxc.hu

Existem para Linux diversos aplicativos que permitem a emulação das condições de uma dada arquitetura de hardware – ou seja, a criação um "PC virtual" dentro do programa. Com eles, podemos instalar outros sistemas operacionais que rodam, aparentemente, como se fossem um programa qualquer dentro do Linux. Também é possível testar programas que têm acesso direto ao hardware – coisa que o Linux não permite no hardware "de verdade". Um programa que emule um ambiente de hardware é conhecido como *emulador de sistema*.

Há dois emuladores de sistema bastante populares para Linux. Um deles é o Bochs [1], um programa com inúmeros e poderosos recursos, mas que é lento e dá uma baita dor de cabeça para configurar. O outro é o famosíssimo VMWare [2], o excelente e veloz emulador comercial que, por isso mesmo, é caríssimo – leia análise do VMWare na edição 3 da Linux Magazine, na página 50. Entretanto, outro competidor subiu ao ringue para desafiar os campeões. Neste artigo veremos como funciona o QEMU, um emulador de sistema bastante poderoso, gratuito e livre.

Usar o QEMU é de uma facilidade extrema. O programa dispõe de comandos simples para tarefas que podem ser complicadas em outros emuladores. Mostraremos como usar o QEMU na prática, mas tenha em mente que este artigo cobre apenas uma pequena fração dos recursos e comandos disponíveis. Para ver por si mesmo, baixe o QEMU hoje mesmo e ponha-o para trabalhar!

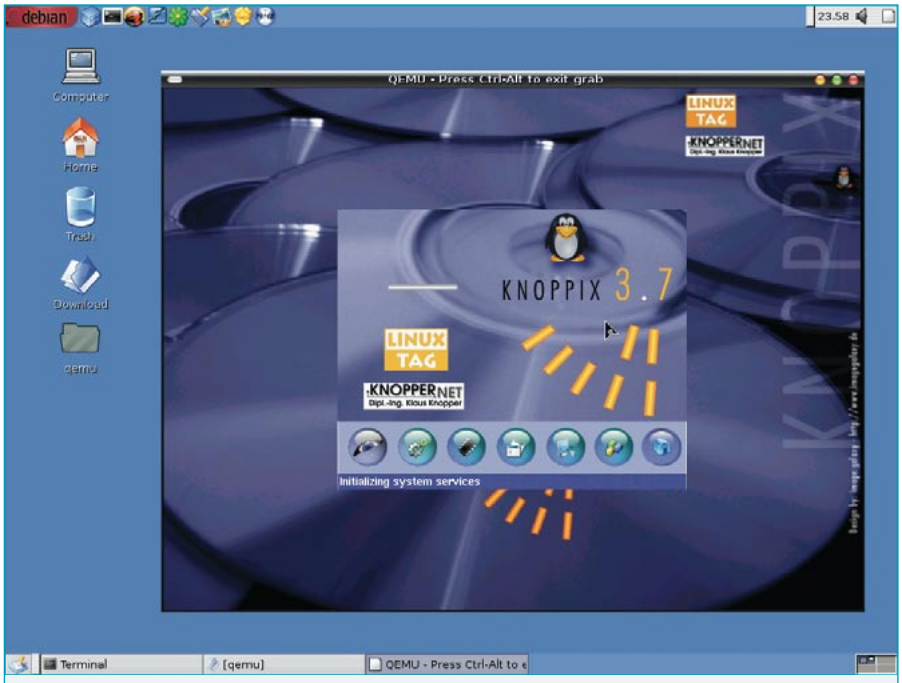

Figura 1: Uma imagem ISO do Knoppix 3.7 iniciado dentro da emulação.

Instalando a fera

O QEMU é distribuído nas duas formas usuais dos programas livres: como código fonte ou como um binário pré-compilado para Linux. Ambos estão disponíveis no site oficial do QEMU em [3]. Baixe a versão binária para o diretório raiz de seu sistema. Abra um console e, como usuário root, emita os comandos:

```
$ cd /
$ su digite a senha do root e pressione [ENTER]
# tar zxvf qemu-0.6.1-i386.tar.gz
# qemu
```

O programa será descompactado e todos os arquivos serão colocados nos lugares certos em seu sistema. Se sua distribuição fugir muito da organização padrão de diretórios é possível que encontre problemas. Nesse caso, compile o QEMU a partir do código fonte. Para isso, digite:

```
$ su digite a senha do root e pressione [ENTER]
# tar zxvf qemu-0.6.1.tar.gz
# cd qemu-0.6.1
# ./configure
# make
# make install
# qemu
```

O QEMU deve ser iniciado dentro do ambiente gráfico – ou seja, precisa que o X Window esteja rodando. Ao iniciar o QEMU, o programa emula o ambiente de hardware no qual ele próprio está rodando. Se você estiver num Athlon, o

QEMU irá emular um Athlon. Se estiver num Macintosh, o QEMU emulará uma máquina PowerPC. Para emular uma arquitetura diferente da do seu computador, especifique o nome da arquitetura como um parâmetro do comando `qemu`. Para uma lista das arquiteturas reconhecidas pelo QEMU, digite `qemu-` e pressione a tecla ***[TAB]*** duas vezes.

Iniciando um LiveCD

Um dos usos mais bacanas do QEMU é testar imagens ISO fresquinhas. Por exemplo, acabamos de baixar uma imagem novíssima do Gobo Linux [4] porque queríamos estudar seu inovador sistema de arquivos, radicalmente diferente (mas compatível) com os Unix tradicionais. Outra coisa que nos chamou a atenção foi o fato de essa distribuição não usar gerenciadores de pacotes, pois o próprio sistema de arquivos administra o problema. Mas não queremos queimar um CD inteirinho com a distribuição para descobrir, depois, que não gostamos dela. Nesses casos, o QEMU vem ao nosso socorro.

A imagem que baixamos é chamada de `GoboLinux-011-i686.iso`. Para testá-la, precisamos enganar o QEMU, fazendo-o acreditar que a imagem é na verdade uma unidade de CDROM com o CD lá dentro. Abra um terminal, torne-se root – dessa forma garantindo que o programa conseguirá acessar todos os periféricos sem problemas – e digite:

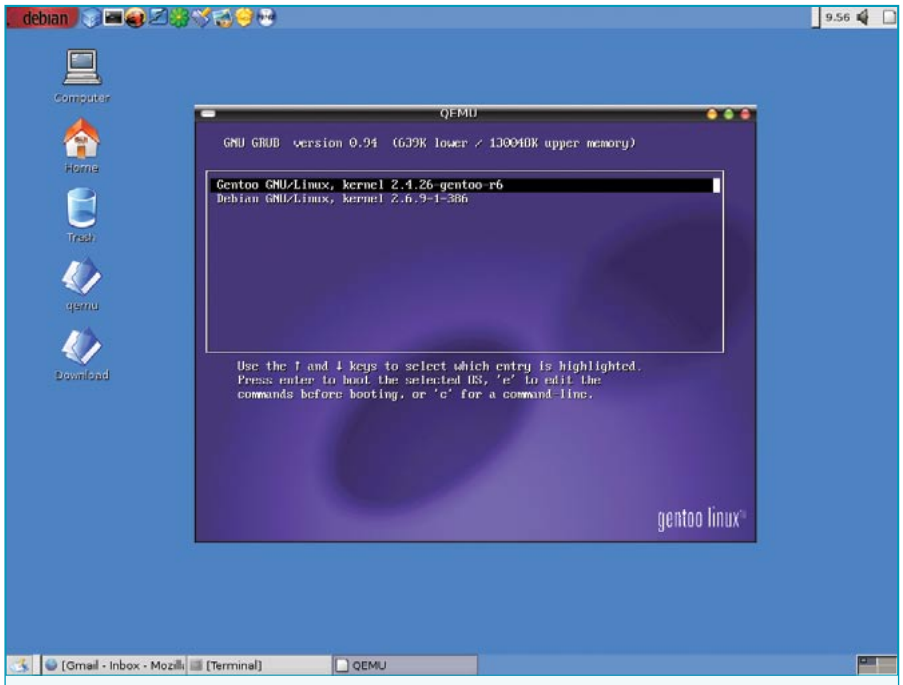

Figura 2: Ao tentar iniciar do disco rígido, o gerenciador de boot aparece.

```
$ su digite a senha do root e pressione [ENTER]
# qemu -cdrom GoboLinux-011-i686.iso
```

Outra janela se abrirá e a emulação começará como se o programa estivesse sendo lido de um drive real. O Gobo Linux apresenta sua tela usual de boot e, depois de o usuário escolher entre os métodos de inicialização, entra em modo gráfico. Você pode, a partir daí, usar o Gobo Linux normalmente. A única limitação é óbvia: sendo um programa que compete com os outros no sistema hospedeiro pela atenção da CPU, o QEMU (e o sistema operacional "convidado") será sensivelmente mais lento do que se estivesse rodando diretamente em hardware real.

A maneira mais fácil de usar a Internet e comunicar-se com o ambiente hospedeiro é usar a opção `-user-net`:

```
# qemu -user-net -cdrom ➥
GoboLinux-011-i686.iso
```

Se um servidor Samba estiver instalado no hospedeiro, o emulador pode acessá-lo com a opção `-smb <diretório>`. Esta opção pode ser usada apenas em conjunto com o parâmetro `-user-net`.

Se, em vez de um arquivo ISO, você tiver uma distribuição em CD – como as da Linux Magazine – ou mesmo um LiveCD, é possível iniciá-lo com o comando:

```
# qemu -user-net -cdrom /dev/cdrom
```

Antes disso, insira o CD no drive sem montá-lo. O QEMU usa o arquivo de dispositivo como "drive", e o CD ou disquete fica disponível tanto para o hospedeiro como para a emulação.

Figura 3: O Gentoo, instalado na partição *hda2*, rodando sob o Debian instalado na partição *hda1*.

Usando o mesmo disco rígido

Outra situação na qual podemos usar o `/dev` em vez de uma imagem de CD é quando queremos fazer o QEMU iniciar um sistema operacional residente no disco

rígido. Um exemplo típico é um computador com dois sistemas Linux instalados, Gentoo e Debian por exemplo. O que acontece se estivermos no Debian mas quisermos usar o Gentoo? Em uma situação normal, teríamos que fechar todos os programas e reiniciar o computador. Com o QEMU, é possível iniciar o segundo sistema sem sair do primeiro e, muito menos, desligar a máquina.

```
# qemu -snapshot -hda /dev/hda
```

A opção `-snapshot` especifica que todas as modificações feitas no disco serão guardadas em um arquivo temporário ao invés do próximo disco. Com isso previne-se a perda de dados que todos tememos em situações assim. Se o sistema possuir um gerenciador de boot como o GRUB (**figura 2**) instalado na MBR, veremos o sistema emulado iniciar. Uma vez iniciado, é possível usá-lo normalmente (**figura 3**).

A opção `-m` do QEMU permite que especifiquemos a quantidade de RAM virtual (em Megabytes) a ser reservada para a emulação. O padrão é 128 MB. Se você possuir RAM sobrando, especificar mais RAM virtual melhora bastante o desempenho da emulação. Por exemplo, se você possuir 512 MB de memória física e quiser garantir um desempenho satisfatório de seu sistema emulado, use o comando:

```
# qemu -snapshot -m 256 -hda /dev/hda
```

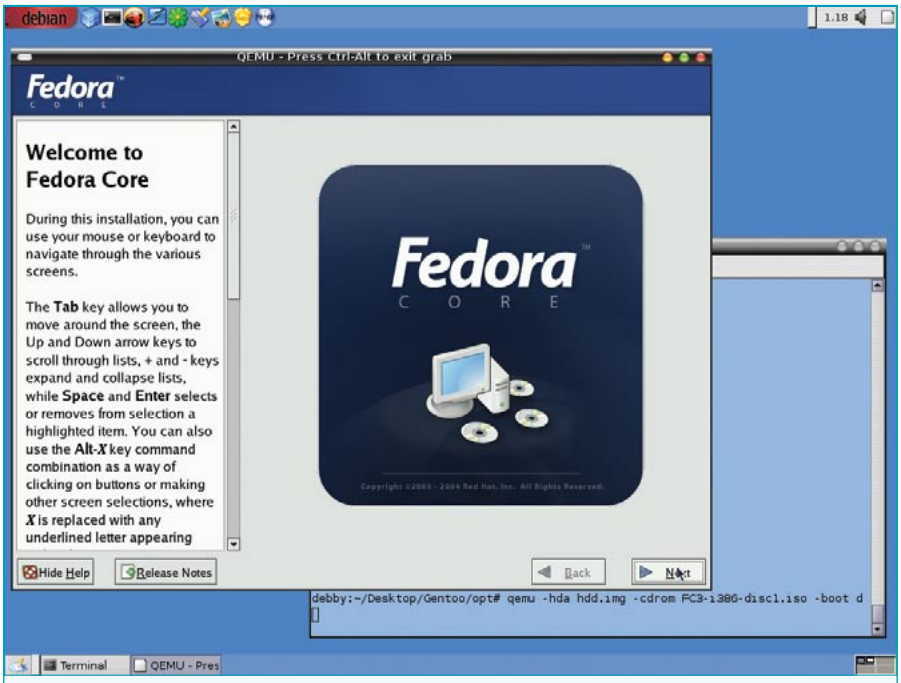

Figura 4: Com uma imagem para o disco rígido, podemos instalar outra distribuição sem medo.

Um pingüim dentro do outro

Se você quiser instalar uma distribuição Linux no ambiente emulado, é preciso criar um arquivo no qual o QEMU vai se travestir de disco rígido. Para isso use o programa `qemu-img`, um dos utilitários incluídos com o QEMU. A sintaxe é muito simples: basta informar o nome da imagem a ser criada e seu tamanho em megabytes. Em nosso caso, criamos um arquivo chamado `hdd.img` com um tamanho de 200 Mbytes com o comando:

```
# qemu-img create hdd.img 2000M
```

Podemos agora instalar o Linux diretamente a partir de uma imagem ISO. Por exemplo, poderíamos baixar a distribuição Ubuntu [5] e instalá-la no ambiente emulado – se não quiser baixar a imagem, o Ubuntu foi incluído no CD da edição 7 da Linux Magazine. O que queremos é instalar direto da imagem ISO, sem precisar gravá-la num CD. Quando tiver terminado de baixar a imagem, informe ao `qemu` o arquivo a ser usado como disco rígido (`-hda hdd.img`), o caminho até a imagem ISO a ser usada como CD-ROM e a opção de boot pelo disco virtual informado (`-boot d`). Por padrão, o QEMU assume que o ambiente emulado deve iniciar pelo disco rígido real, se estiver presente.

```
# qemu -hda hdd.img -cdrom ubuntu.iso ➥
-boot d
```

E o que acontece se quisermos instalar uma distribuição ou sistema operacional que possua mais de um CD – como, por exemplo, o Debian com seus sete CDs ou o Solaris 10? Nesse caso, é preciso usar a opção `-monitor stdio`. Com ela, quando o QEMU inicia a emulação, um shell interativo se abre no terminal.

```
# qemu -monitor stdio -hda hdd.img ➥
-cdrom fedora_cd1.iso -boot d
```

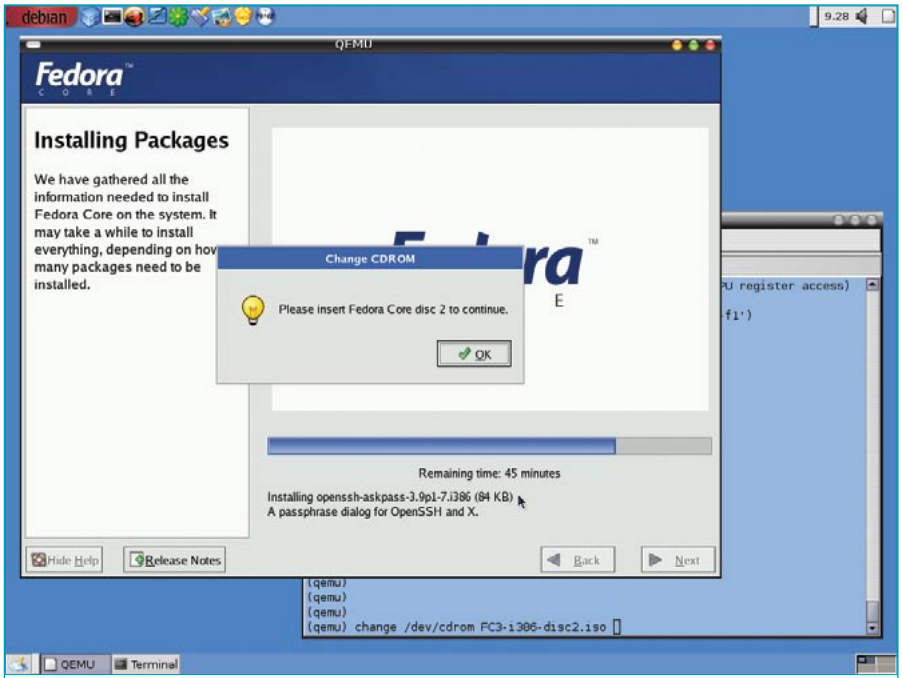

Figura 5: Com a monitoração ligada, podemos alterar a imagem ISO usada como um dispositivo.

Nesse shell, é possível controlar a emulação com comandos. Os vários comandos disponíveis permitem reiniciar a emulação, gravar o estado atual para continuá-la posteriormente ou trocar o arquivo de um dado dispositivo emulado. Se a distribuição escolhida possuir mais de um CD para instalação, é possível trocar as "mídias virtuais" (ou seja, o arquivo ISO) com um comando como este (**figura 5**):

```
# qemu change cdrom fedora_cd2.iso
```

No final do processo de instalação, você terá uma imagem de disco rígido pela qual o QEMU pode iniciar um sistema operacional. Para iniciar a emulação com esse disco virtual, digite:

```
# qemu hdd.img
```

Neste caso, não é preciso informar nenhuma opção, pois os parâmetros padrão do disco rígido verdadeiro, `hda`, são válidos também para a imagem.

E por que não o DOS?

Quem não se lembra do DOS? Ainda hoje, muitas empresas dependem daquele programinha em Clipper do qual não podem prescindir nem por poucos minutos – infelizmente, ele roda apenas em DOS. Em vez de criar uma partição de 50 MB para um único programa e reiniciar o Linux toda vez que precisar usá-lo, use o QEMU com uma imagem do MS-DOS. Se não possuir uma licença dele ou quiser uma alternativa livre, sempre há o FreeDOS [6] (clone do DOS distribuído sob a licença GPL).

Como exemplo, vamos usar o FreeDOS. Baixe a imagem já pronta (`fdos-100meg.tar.gz`, disponível em [7]) e extraia o arquivo `fdos_8h1.img` em um diretório – você pode usar o comando `tar` na linha de comando ou ferramentas gráficas como o *file-roller* no Gnome e o *ark* no KDE. Como root, digite:

```
# qemu -hda fdos_8h1.img -fda /dev/fd0 ➥
-boot c
```

Observe que passamos a opção `-fda` para o QEMU. De forma similar às opções `-hda` e `-cdrom`, já vistas anteriormente, a opção `-fda` é usada para ler o conteúdo do disquete no ambiente emulado. O disquete será visto como o drive *A:*, exatamente como numa sessão "real" do DOS.

Com isso, o FreeDOS é iniciado e fica de prontidão. Com o DOS funcionando, você pode fazer muitas outras coisas, como instalar o SEAL [8], um ambiente gráfico para o DOS (como o Windows 3.1) mas com funcionalidade bastante parecida com o Windows 98 – e muito mais bonito! Os arquivos de instalação do SEAL estão no diretório `C:\fdos\seal2`. Às vezes o programa não inicia devido a problemas de gerenciamento de memória, portanto é preciso usar os utilitários que acompanham o SEAL para criar um arquivo de troca (*swap*) e transformá-lo em memória virtual:

```
C:\> cd c:\fdos\seal2
C:\> cwsparam
C:\> cwsdpmi
C:\> cwsdpr0
C:\> install
```

Agora basta reiniciar os programas de memória virtual (*swap*) e chamar o programa do mouse. Depois disso, o SEAL pode ser chamado (**figura 6**).

```
C:\> cwsdpmi
C:\> cwsdpr0
C:\> cd c:\seal2
C:\> ctmouse
C:\> seal
```

O QEMU é um software bastante poderoso. Como outros emuladores, sofre dos problemas de velocidade que a pouca memória impõe – já que há dois sistemas operacionais rodando ao mesmo tempo.

Quem quiser usar o QEMU para testar outros sistemas operacionais encontrará um número impressionante de imagens de disco no site FreeOSZoo [9]. Mesmo as imagens criadas para o Bochs [10] podem ser usadas. ■

Figura 6: Uma sessão do SEAL sob o QEMU.

Informações

[1] Bochs: http://bochs.sourceforge.net

[2] VMWare: http://www.vmware.com

[3] QEMU: http://fabrice.bellard.free.fr/qemu

[4] Gobo Linux: http://www.gobolinux.org

[5] Ubuntu: http://www.ubuntulinux.org

[6] FreeDOS: http://www.freedos.org

[7] Imagem de 100 MB do FreeDOS para o Bochs: http://prdownloads.sourceforge.net/bochs/fdos-100meg.tar.gz?download

[8] SEAL: http://sealsystem.sourceforge.net

[9] FreeOSZoo: http://www.freeoszoo.org

[10] Imagens para o Bochs: http://sourceforge.net/project/showfiles.php?group_id=12580&package_id=27799

Sobre o autor

Fabrizio Ciacchi (http://fabrizio.ciacchi.it) é um estudante italiano de Ciência da Computação na Universidade de Pisa. Trabalha como consultor e escreve artigos sobre Linux.