

Rodando Mac OS X no Linux

Uma maçã por dia...



Os emuladores agora oferecem aos usuários a capacidade de rodar o amado Mac OS X no Linux, matando vários coelhos com uma só cajadada (nenhum animal foi ferido durante a produção deste artigo).

POR OLIVER FROMMEL

O modo mais fácil de rodar programas do Mac no Linux é o software livre *Mac-on-Linux* [1]. Ele não oferece emulação de hardware e, por isso, exige um computador Apple. O pacote *Mac-on-Linux* teoricamente permite o uso de outras arquiteturas PowerPC [2], como [3], mas a instalação do sistema operacional da Apple, o Mac OS, nessas arquiteturas iria transgredir o contrato de licença.

Linux para Apple

É possível usar o *Mac-on-Linux* em computadores Apple como os iMac, iBook, Power Mac ou PowerBook. Claro, é preciso que sua distribuição Linux tenha uma versão para PowerPC [4]. As seguintes distribuições têm essa capacidade: Fedora Core 3 (ainda em fase de testes [5]), Gentoo [6], Debian [7], Ubuntu [8] e Yellow Dog [9].

O *Mac-on-Linux* não emula hardware; ele passa comandos diretamente à CPU. Também traz alguns drivers de dispositivos que foram otimizados para essa aplicação de modo semelhante ao VMware, o que explica porque o *Mac-on-Linux* é muito mais rápido que um emulador.

Mac OS como um programa do Linux

O *Mac-on-Linux* roda um sistema operacional Apple padrão como um programa do Linux. Em nosso exemplo, usaremos o Mac OS X, mas o *Mac-on-Linux* também permite o uso de variantes mais antigas, como o Mac OS 9, que ainda é bastante comum.

Usuários do Gentoo podem baixar e compilar o *Mac-on-Linux* do típico modo Gentoo usando o comando `emerge mol`. O Yellow Dog Linux, na verdade, já inclui o programa em sua seleção padrão.

Usamos a distribuição Ubuntu para o processador PowerPC, já que vimos ser impossível

fazer com que o Gentoo ou a distribuição comercial Yellow Dog iniciassem o servidor gráfico em nosso iBook G4/800, cuja placa de vídeo usa um chipset Radeon Mobility 9200.

O instalador do Ubuntu funciona com o gerenciador de pacotes *apt* do Debian, que baixa arquivos da Internet. O *Mac-on-Linux* precisa de alguns módulos do kernel para o PPC Linux; você mesmo terá de compilar esses módulos. Isso significa instalar algumas ferramentas, se é que elas já não estão lá. O seguinte comando baixa os pacotes e os instala:

```
# sudo apt-get install build-essential \
linux-headers-2.6-powerpc
```

Se lhe pedirem uma senha, digite seu nome de usuário e senha normais, já que tipicamente o Ubuntu não tem uma conta de administrador (root, ver [8]). Ao invés disso, os usuários podem rodar comandos que tipicamente necessitam de privilégios de root usando `sudo nome_do_comando`. Obviamente você também pode instalar os

Quadro 1: Adições a `/etc/apt/sources.list`

```
deb http://archive.ubuntu.com/ubuntu warty multiverse
deb-src http://archive.ubuntu.com/ubuntu warty multiverse
```

pacotes usando o gerenciador gráfico de pacotes *Synaptic* e fazer tudo com o mouse, sem precisar ficar digitando comandinhos.

Infelizmente, o repositório padrão do Ubuntu não inclui os arquivos de que precisamos, de forma que será preciso antes de tudo adicionar à lista de fontes do APT o repositório *multiverse*, que inclui pacotes que não fazem parte da distribuição padrão do Ubuntu. Para isso, acrescente as linhas do **quadro 1** a seu arquivo `/etc/apt/sources.list`.

Agora atualize o repositório local para que aponte para o novo software, digitando `sudo apt-get update`. Em seguida, instale o código fonte para os módulos do kernel do Mac-on-Linux:

```
# sudo apt-get install mol-modules-source
```

Abra um editor de textos para editar o cabeçalho desse pacote e remover um erro que de outra forma impediria você de compilar os módulos. Se você preferir usar outro editor que não o *vi*, substitua *vi* pelo nome de seu editor favorito.

```
# sudo vi /usr/src/linux-headers-2.6.8.1-2
4-powerpc/include/asm/setup.h
```

Apague a linha com `#include <asm-m68k/setup.h>` e salve o arquivo novamente. Mude então para o diretório com o código fonte do Linux e descompacte o arquivo que o *apt-get* armazenou ali.

```
# cd /usr/src
# sudo tar xzvf mol-modules.tar.gz
```

Você precisa ajustar algumas variáveis de ambiente para se certificar de que os módulos combinam com sua versão do kernel atual:

```
# export KVERS="$(uname -r)"
# export KSRC="/usr/src/linux-2
headers-$(uname -r)"
# export KDREV="ubuntu0"
```

Agora vá ao subdiretório com o Mac-on-Linux e chame o script de compilação:

```
# cd modules/mol
# sudo debian/rules build
```

Depois de um curto intervalo, a compilação estará completa; agora você pode criar um pacote Debian com os arquivos criados por essa etapa:

```
# sudo debian/rules binary-mol-modules
```

O produto final deverá estar em `/usr/src`. Mude para esse diretório e instale o software em seu sistema Ubuntu:

```
# sudo dpkg -i mol-modules-2.6.8.1-3-2
powerpc_0.9.70+ubuntu0_powerpc.deb
```

Isso finalmente completa o trabalho de preparação e você poderá então instalar os pacotes restantes do Mac-on-Linux:

```
# sudo apt-get install mol mol-drivers-macosx
```

O gerenciador de pacotes resolve dependências por si só e adicionalmente instala o *mol*, o *mol-drivers-macosx* e o *mol-drivers-linux*, necessário apenas se você quiser rodar uma versão PPC do Linux.

Se você especificar o parâmetro `--loadonly` ao iniciar o programa, ele informa se o módulo do kernel foi carregado (**figura 1**). O engano mais comum é ter versões do kernel e do módulo que não combinam. Embora o parâmetro `-a` permita diferentes números de versão, você não deve esperar que uma configuração onde os números não combinam funcione.




Figura 1: Iniciando o Mac-on-Linux com a opção `--loadonly` inclui nos registros do sistema (*logs*) os módulos necessários para rodar o programa.

Depois de carregar o módulo, você pode passar à configuração do vídeo. O comando `sudo molvconfig` inicia o programa requerido, que testa sozinho os modos de operação ou permite escolher um modo manualmente. Em nosso laboratório, ambas as opções funcionaram bem com a configuração padrão.

`startmol` ajuda a organizar as atribuições do teclado. O parâmetro `--keyconfig` inicia um programa interativo, que exibe o nome da tecla (por exemplo, *Return* ou *Apple-Key*) e pede que você a pressione.

Depois de completar a configuração, você pode iniciar o Mac OS X numa janela digitando `sudo startmol -X`. O Mac OS X é imediatamente localizado no disco rígido e o sistema “inicia”. O Mac-on-Linux usa uma janela de terminal para exibir mensagens que surjam durante o boot. Essas mensagens são armazenadas permanentemente num arquivo sob `/var/log`, que é onde você deve olhar se algo der errado.

Túnel de rede

Configurar a rede é uma tarefa mais difícil e, infelizmente, não há um assistente para nos mostrar o caminho certo. Comece habilitando a interface de túnel no arquivo `/etc/mol/molrc.net`; essa interface carrega automaticamente o módulo do kernel e atribui um endereço IP. Para habilitar a interface de túnel, remova o `#` na linha

Quadro 2: `/etc/mol/tunconfig`

```
#!/bin/bash
/sbin/ifconfig tun0 192.168.40.1
/sbin/iptables -D POSTROUTING -t nat -s 192.168.40.0/24 -d ! 192.168.40.0/24 -j MASQUERADE
/sbin/iptables -t nat -s 192.168.40.0/24 -d ! 192.168.40.0/24 -A POSTROUTING -j MASQUERADE
/sbin/iptables -t filter -P FORWARD ACCEPT
echo 1 > /proc/sys/net/ipv4/ip_forward
```

com `netdev: tun0 -tun`. Agora suas mensagens de boot deverão incluir uma linha semelhante a `Ethernet Interface 'tun-<tun0>' @ 00:00:0D:EA:DB:EE`.

Um novo disco aparece no Mac OS X; abra-o com um duplo clique. Esse disco contém os drivers especiais de que o Mac OS X precisa para acessar a rede em modo de emulação. Para instalar os drivers, clique no pacote `Pkg` e siga as instruções.

Você provavelmente vai querer usar o Mac-on-Linux para acessar tanto a rede quanto a Internet. Para passar tráfego ao Mac-on-Linux, é preciso editar o script `/etc/mol/tunconfig`, que é iniciado por `startmol`. A versão padrão deste script inicia um servidor DHCP e usa o `iptables` para mapear endereços IP.

Em nosso laboratório, usamos um `tunconfig` alternativo com um endereço IP estático (ver **quadro 2**). No Mac OS X, atribua o endereço IP `192.168.40.2` à interface de rede `en3` e configure `192.168.40.1` como roteador.

Depois de terminar essa configuração, o Mac-on-Linux terá acesso à Internet.

Acelerador

Para iniciar o Mac-on-Linux em modo tela cheia no X, é preciso modificar um arquivo chamado `/etc/mol/molrc.video`. Na linha `enable_xvideo: ...`, mude o valor de `yes` para `no`. Esse valor faz com que o Mac OS X rode muito mais rápido, já que elimina a necessidade de uma porção de operações gráficas comedoras de tempo. De acordo com a documentação do Yellow Dog, a velocidade é de “quase 100%” em comparação com um sistema que roda diretamente no hardware. É possível conseguir o mesmo efeito desprezando o X e iniciando o Mac-on-Linux diretamente num console texto (ou seja, não numa janela de terminal do ambiente gráfico). Para maiores informações consulte **[10]**.

Uma desvantagem do Mac-on-Linux é ser impossível trocar os CD ROMs após iniciar o emulador. Resta a esperança de que os desenvolvedores do Mac-on-

Linux removam esse problema numa versão futura. Quem sabe você mesmo não tenha as habilidades necessárias e queira dar uma mão?

Mac OS sem Hardware Apple

Se você não possui um computador da Apple, mas ainda estiver a fim de dar uma olhada nesse colorido sistema operacional, precisa sair à cata de uma alternativa. Ou seja, você precisa de um emulador de verdade que tapeie o Mac OS X para que ele acredite estar rodando numa plataforma Apple. Também ajuda se você gostar de experimentar – e, é claro, você vai precisar de um conjunto de CDs de instalação do Mac OS X. Lembre-se de que o Mac OS X é um software comercial, portanto nada de pirataria, hein?

O *PearPC* **[11]** implementa uma CPU PowerPC e um conjunto de periféricos. A instalação é “sopa no mel”:

```
# tar xjf pearpc-0.3.1.tar.bz2
# cd pearpc-0.3.1
# ./configure --enable-ui=SDL --enable-cpu=
jitc_x86
# make
# su -c 'make install'
```

Neste exemplo, o PearPC está usando a biblioteca SDL para desenhar a área de trabalho, embora seja possível usar `--enable-ui=x11` como alternativa.

Ao término da compilação, crie um arquivo vazio, que o PearPC usará como disco virtual. O seguinte comando cria um arquivo de 3 GBytes chamado `macosx.img`:

```
# dd if=/dev/zero of=macosx.img bs=516096
seek=6241 count=0
```

Ao contrário do que dizem as instruções na documentação do PearPC, é possível particionar o disco rígido virtual sem ter de recorrer ao *Darwin OS*, que é a versão livre do sistema operacional da Apple. Porém não se anime: o Darwin inclui



Figura 2: A tela de início do PearPC permite escolher a partição de boot.

apenas os componentes em modo texto. A interface gráfica, a área de trabalho e outros componentes principais (como o Quicktime) estão faltando.

É uma boa idéia criar um novo diretório para abrigar o emulador e chamá-lo `ppc`, por exemplo. Então copie os arquivos-modelo de configuração para esse diretório com o comando:

```
# cp pearpc-0.3.1/ppccfg.example ppc/ppcfg
```

Para imitar o adaptador gráfico, o PearPC também necessita de um arquivo `video.x`, que você precisa copiar novamente para seu diretório de trabalho. Mova também a imagem do disco rígido, em nosso exemplo chamada `macosx.img`, para o mesmo diretório; assim, todos os arquivos de que você necessita estarão juntos no lugar certo.

É possível editar o arquivo de configuração `ppccfg` com um editor de texto. Os ajustes padrão funcionarão bem, mas você precisa modificar o nome da imagem de seu disco rígido. Para isso, mude uma linha no arquivo, como segue:

```
pci_ide0_master_image = "macosx.img"
```

A linha `prom_bootmethod` especifica como o PearPC inicia. O valor pré-configurado, `auto`, usa a primeira partição de boot. Se você mudar esse valor para `select`, o emulador esperará que você selecione uma partição (**figura 2**).

Insira o primeiro CD de instalação do Mac OS X para começar a instalação. Se você já não estiver no diretório `ppc`, vá até ele e inicie o emulador, especificando seu arquivo de configuração como parâmetro:

```
# ppc ppcfg
```

Se você escolher iniciar a partir do CD, a tela de início do Mac OS X aparecerá, como mostrado na **figura 3**. Você pode pressionar **[Alt]+[Enter]** para alternar entre os modos de exibição em tela cheia e janela.

Em alguns instantes, aparece o programa de instalação. Se você atravessar as etapas necessárias, vai precisar definir um “volume alvo” em determinado momento, mas infelizmente você ainda não tem um. É preciso antes particionar o disco rígido virtual. Para iniciar a ferramenta de particionamento, selecione o *Installer | Hard disk tool*, no menu no topo da tela.

Na ferramenta de disco rígido clique na aba *Partitioning* e escolha a opção *1 Partition* no item *Volume Schema*. Clique em *Partition* e confirme a operação na tela seguinte. O programa particiona o disco virtual e formata a nova partição. Após sair da ferramenta, a partição estará visível no instalador.

Se o instalador se queixar de pouco espaço em disco, é possível desselecionar alguns pacotes, como os *Language packages* e *Additional voices*. Ao fim da seleção de pacotes, o instalador os copiará para



Figura 3: O instalador do Mac OS X inicia no PearPC como se você estivesse num Mac de verdade.

seu disco virtual. No final dessa etapa, saia do PearPC e insira o segundo CD de instalação. Ao reiniciar, selecione a partição criada anteriormente como partição de boot. O instalador copiará mais alguns arquivos e completará a instalação. Você pode pular as questões sobre o registro de sua conta de usuário.

Redes para especialistas

Para permitir que o Mac OS X use a rede num ambiente PearPC, é preciso modificar a linha referente à placa de rede no arquivo de configuração:

```
pci_rt18139_installed = 1
```

Ao reiniciar, o emulador deve reclamar de que está faltando uma interface de túnel de rede (*network tunnel interface*) `/dev/net/tun`. Se isso acontecer, rode `modprobe tun` como *root* para carregar o módulo do kernel do Linux.

A placa Ethernet virtual é automaticamente detectada pelo Mac OS X; verifique a configuração de rede nas Preferências do Sistema (*Maçã | Preferências do Sistema*). Atribua o endereço IP estático **192.168.1.1**, que foi configurado em `scripts/ifppc_up`. O endereço do roteador é **192.168.1.80**. Especifique um servidor DNS funcional no arquivo `/etc/resolv.conf`.

Assim como no Mac-on-Linux, o PearPC configurará a rede virtual de maneira bastante autônoma em segundo plano e até mesmo ajustará a tradução de endereços por NAT (*Network Address Translation*). Para isso, basta copiar todo o subdiretório `scripts` para o lugar certo:

```
# cp -R pearpc-0.3.1/scripts ppc
```

Como os programas mudam as configurações de rede do Linux, você precisa de privilégios de administrador para essa etapa. Em outras palavras, você precisa iniciar o PearPC como administrador ou atribuir privilégios SUID de root para os

programas `ifppc_up.setuid` e `ifppc_down.setuid` (use o comando `chmod +s nome_do_programa`).

Por padrão, o PearPC cria uma interface de rede chamada `ppc` e um endereço **192.168.1.80** no sistema Linux; chame `/sbin/ifconfig ppc` para verificar esse ponto.

Uma união feliz

O Mac-on-Linux e o PearPC impressionam por causa de sua riqueza de recursos e facilidade de uso. O fato de o Mac-on-Linux superar de longe seu competidor não é nem um pouco surpreendente. Afinal, estritamente falando, o Mac-on-Linux não é um emulador, mas um programa que roda o Mac OS X no Linux. De fato, ele é rápido o bastante para ser usado em trabalhos sérios num computador moderno. O modo como o programa lida com CDs é um problema. Nem o Mac-on-Linux nem o PearPC permitem que os usuários troquem os CDs depois de iniciados.

Ambos os programas têm uma configuração de rede bastante complicada, mas isso se relaciona mais à complexidade das funções de que o Linux precisa para permitir a conexão de rede virtual. Entretanto, isso deve ser corrigido no futuro. ■

INFORMAÇÕES

- | | |
|------|---|
| [1] | Mac-on-Linux: www.maconlinux.org |
| [2] | Processador PowerPC: www.ibm.com/chips/products/powerpc |
| [3] | Pegasos Computer www.pegasosppc.com |
| [4] | Site oficial do Linux no PowerPC: www.penguinppc.org |
| [5] | Fedora Core 3 PPC: fedoraproject.org/fedorappc/FC-3 |
| [6] | Gentoo PPC: www.gentoo.org/doc/en/handbook/handbook-ppc.xml |
| [7] | Debian PPC: www.debian.org/ports/powerpc |
| [8] | Ubuntu: www.ubuntulinux.org |
| [9] | Yellow Dog: www.yellowdoglinux.com |
| [10] | Modo tela cheia no Mac-on-Linux: www.maconlinux.org/userguide/cvideo.html |