



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Màquines virtuals amb user-mode-linux (6607 lectures)

Per **Carles Bayés**, phalangana (<http://>)

Creado el 11/09/2004 18:18 modificado el 12/09/2004 21:25

Des de fa ja un temps sembla que les màquines virtuals s'han posat de moda, no només per a entorns de desenvolupament o proves, sinó també per a entorns de producció (hosting, "consolidació de servidors" i altres). Us mostro amb un petit exemple com muntar màquines virtuals amb user-mode-linux.

Quan comences a pensar en muntar màquines virtuals, trobes que tens al teu abast moltes solucions, algunes comercials com [VMWare](#)⁽¹⁾ o [VirtualPC](#)⁽²⁾ (recentment adquirida per Microsoft) i d'altres lliures, com [Bochs](#)⁽³⁾, [Plex86](#)⁽⁴⁾, [coLinux](#)⁽⁵⁾ o [user-mode-linux](#)⁽⁶⁾. Cada una d'aquestes opcions té els seus avantatges i els seus inconvenients. Jo em vaig decantar per user-mode-linux, perquè és lliure i perquè el que segons el meu parer és el seu inconvenient més important (no permet corre altre sistema operatiu que Linux) no m'importava en absolut pel que tenia pensat. I d'entre les opcions lliures era la que en principi oferia millor rendiment i més simplicitat.

Com deia al resum inicial, els usos d'aquestes tecnologies poden ser molts. Alguns dels que més se senten últimament són els següents, però no són els únics ni de bon tros:

- **Hosting:** Cada vegada són més els ISP que ofereixen servidors virtuals utilitzant aquestes tecnologies.
- **Consolidació de servidors:** Es tracta d'agrupar tots els servidors d'una empresa en una sola màquina (que ha de tenir relativa solvència de recursos, evidentment). La idea es basa en aprofitar millor els recursos del servidor, ja que és habitual el desaprofitament de recursos de hardware en aquests temps en els que el hardware avança tant de pressa. En aquests casos, com sempre que es fan servir màquines virtuals, la realització de còpies de seguretat de cada una de les màquines resulta molt fàcil, ja que en general suposarà la còpia d'un sol fitxer.
- **Honeypots:** Màquines posades a internet perquè hi "juguin" els hackers. Es fan servir en general per aprendre els comportaments i les noves tècniques que fan servir els intrusos informàtics.
- **Màquines de desenvolupament o proves:** Sempre és millor provar les coses en una màquina que no és crítica pel negoci i que, com en el cas de les màquines virtuals, es pot recuperar en molt poc temps.

Crec que no hi ha res com un cas pràctic, per tant, ara que ja he fet una mica d'introducció potser que anéssim al gra: El que vull muntar en primera instància és un servidor virtual per a desenvolupament d'aplicacions web amb Apache, Mysql i Php (i potser Perl), però l'objectiu final és implementar-ho en un servidor web on ostararé les aplicacions que desenvolupi, de manera que aquest primer pas (a més de per passar-m'ho bé una estona, perquè ens hem d'enganyar...) em servirà com a test per a muntar-ho a producció. El que vull en producció és crear dues màquines virtuals, de manera que en una sola màquina "física" vull que hi hagi el firewall (que situaré al host), un IDS en una màquina virtual (snort) i el servidor web en una altra (el que no sé és si l'arcaica màquina que tinc pensat utilitzar ho suportarà tot o hauré de muntar l'IDS al host...).

Els passos que seguiré són per a Debian (SID, per ser exactes), però com que compilaré a partir de les fonts, hauria de ser molt similar a les altres distribucions.

La raó per compilar des de les fonts és que en el moment de fer aquestes proves la versió per a SID tenia un bug (concretament el [260111](#)⁽⁷⁾) que feia que la màquina virtual es pengés en arrencar. Sense aquest petit inconvenient, el procés d'instal·lació hauria estat molt més senzill, com és habitual en Debian. N'hi hauria hagut prou amb

```
# apt-get install user-mode-linux uml-utilities
```



i passar als següents passos.

L'*apt-get uml-utilities*, però, sí que cal executar-lo si no voleu haver de compilar les utilitats manualment, perquè el kernel UML requereix algunes d'aquestes utilitats externes per a certes coses, com ara la configuració de la xarxa amb dispositius TAP.

Instal·lació

Per instal·lar user-mode-linux el primer que cal fer és baixar-se les fonts del kernel (en el meu cas 2.4.26, l'últim de la branca 2.4 amb patch UML disponible el dia d'escriure aquest mini-howto) i el corresponent patch. Ho podem fer des de kernel.org⁽⁸⁾ i des de la pàgina del projecte [UML](http://uml.org)⁽⁶⁾, respectivament:

```
~/software/uml$ wget http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.26.tar.bz2(9)
~/software/uml$ wget http://belnet.dl.sourceforge.net/sourceforge/user-mode-linux/uml-patch-2.4.26-3.bz2(10)
```

I Ara només cal descomprimir el kernel, aplicar-hi el patch i compilar-lo. Jo ho faig amb les opcions per defecte per simplificar i perquè l'entorn que vull no és res de l'altre món:

```
~/software/uml$ tar xvfj linux-2.4.26.tar.bz2
~/software/uml$ cd linux-2.4.26
~/software/uml/linux-2.4.26$ bzip2 ../uml-patch-2.4.26-3.bz2 | patch -p1
~/software/uml/linux-2.4.26$ make xconfig ARCH=um
~/software/uml/linux-2.4.26$ make linux ARCH=um
```

Ara mateix ja estem en disposició d'executar linux en espai d'usuari, però ens falta el sistema de fitxers arrel (si executem *./linux* ens ho advertirà). Podem descarregar-ne alguns des de la [pàgina de descàrregues d'uml](http://uml.org)⁽¹¹⁾. En el meu cas, faré servir el sistema de fitxers de Debian, però n'hi ha de moltes distribucions (UML pot ser una bona manera de provar altres distribucions sense haver de redimensionar particions ni posar en risc el sistema que ja teniu corrent a la màquina):

```
~/software/uml$ wget http://belnet.dl.sourceforge.net/sourceforge/user-mode-linux/Debian-3.0r0.ext2.bz2(12)
~/software/uml$ cd ~/uml
~/uml$ bunzip2 ../software/uml/Debian-3.0r0.ext2.bz2
~/uml$ mv ../software/uml/Debian-3.0r0.ext2 ./
~/uml$ cp ../software/uml/linux-2.4.26/linux ./
```

I ara sí que definitivament ja podem (o hauríem de poder) executar linux dins de linux:

```
~/uml$ ./linux ubd0=Debian-3.0r0.ext2 mem=32M
```

Si tot ha anat bé, s'han obert 3 consoles esperant que ens hi identifiquem. Ens hi podem autenticar com a root (sense contrasenya):

...

```
Debian GNU/Linux 3.0 (none) ttys/0
```

```
(none) login: root
Last login: Sun Aug 22 20:28:35 2004 on ttys/1
Linux (none) 2.4.26 #1 ds ago 21 20:23:10 CEST 2004 i686 unknown
```

```
Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program
are described in the individual files in /usr/share/doc/*/copyright
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
(none):~# uname -a
Linux (none) 2.4.26 #1 ds ago 21 20:23:10 CEST 2004 i686 unknown
(none):~#
```

...



De tota manera, el sistema de fitxers original ens serveix de poc, perquè no es pot dir que dugui massa aplicacions (per comprovar-ho en Debian, *dpkg -l*) i li queda poc espai lliure.

```
(none):~# df
Filesystem          1k-blocks      Used Available Use% Mounted on
/dev/ubd/0           59386          48365      7952   86% /
```

En aquest punt tenim diverses opcions. Podríem crear més particions per incrementar l'espai disponible o podem redimensionar la partició que tenim. Jo opto per aquesta darrera opció. Podeu fer-ho amb *resize2fs* o creant una partició més gran i copiar-hi tot el contingut de la partició original.

Amb *resize2fs*:

```
~/uml$ cp Debian-3.0r0.ext2 root_fs
```

(*root_fs* és el nom de fitxer que busca uml per defecte. D'aquesta manera ens podríem estalviar l'argument *ubd0=...*).

```
~/uml$ /sbin/e2fsck -f root_fs
~/uml$ /sbin/resize2fs root_fs 250M
~/uml$ /sbin/tune2fs -j -i 30 root_fs
```

Creant una partició des de zero:

```
~/uml$ dd if=/dev/zero of=root_fs.ext3 bs=1M count=250
~/uml$ /sbin/mkfs.ext3 root_fs.ext3
~/uml$ mkdir mnt
~/uml$ mkdir mnt/debian
~/uml$ mkdir mnt/root_fs
~/uml$ su
Password:
# mount -o loop Debian-3.0r0.ext2 mnt/debian
# mount -o loop root_fs.ext3 mnt/root_fs
# cp -fa mnt/debian/* mnt/root_fs
# umount mnt/debian
# umount mnt/root_fs
# exit
```

Podem crear també una partició de swap (recomanable en el meu cas, ja que em sobra disc i em falta ram).

```
~/uml$ dd if=/dev/zero of=swap bs=1M count=64
~/uml$ /sbin/mkswap swap
~/uml$ ./linux ubd0=root_fs ubd1=swap mem=32M
```

Després d'arrencar la màquina virtual, cal informar al nou sistema que *ubd1* és la partició de swap. Ho podem fer amb *swapon* o introduint l'entrada corresponent a */etc/fstab* perquè les successives sessions ja vegin la partició des de l'inici:

```
(none):~# echo /dev/ubd/1 none swap sw 0 0 >> /etc/fstab
```

Configuració de la xarxa

Ja tenim una màquina virtual corrent Linux 2.4.26 al nostre sistema, però no hi ha xarxa, és a dir, estem aïllats del món. I un Linux aïllat... No m'ho puc imaginar. Em fa mal només de pensar-hi!

Hi ha diverses maneres de proporcionar connectivitat a l'instància UML (podeu consultar-les a la secció de [xarxa](#)⁽¹³⁾ de la web del projecte). Una de les més senzilles és a través de la utilitat *uml_net* (UML s'encarrega de crear-te un dispositiu TAP i assignar-li una IP simplement passant-li un argument a la línia de comandes), però jo m'inclino per crear un dispositiu TAP al host i crear un bridge on hi afegiré aquest dispositiu (que es correspondrà amb *eth0* de la instància UML) i l'interfície *eth0* del host. D'aquesta manera la màquina virtual té accés directe a la xarxa local. No cal dir que necessiteu suport al host tant per *bridging* com per dispositius *TUN/TAP*.



Abans de tot això, però, cal donar permisos d'escriptura sobre el dispositiu `/dev/net/tun` a l'usuari que executarà UML (d'altra manera no tindria accés a la xarxa). Per fer-ho, crec que el millor és assignar al dispositiu un nou grup, donar-li permisos d'escriptura i anar afegint-hi tots els usuaris que hagin d'executar instàncies UML:

```
# addgroup umlnetwork
S'està afegint el grup umlnetwork (1001)...
Fet.
# chgrp umlnetwork /dev/net/tun
# adduser umluser umlnetwork
S'està afegint l'usuari umluser al grup umlnetwork...
Fet.
# chmod g+w /dev/net/tun
```

Ara hem d'executar els passos necessaris per fer tot el muntatge. Ho poso en forma d'script per simplicitat (l'script s'ha d'executar com a `root` i recordeu que heu de tenir instal·lades les `uml-utilities` i `bridge-utils`):

```
#!/bin/bash
# Inici de l'uml en mode bridge. Basat en els exemples de
#http://user-mode-linux.sourceforge.net/networking.html.
# S'ha d'executar com a root.

#carreguem el mòdul tun
modprobe tun

#creem un dispositiu per a l'usuari
tunctl -d umltap0 2>/dev/null
tunctl -u umluser -t umltap0

#afegim el bridge
ifconfig br0 down 2>/dev/null
brctl delbr br0 2>/dev/null
brctl addbr br0
ifconfig eth0 0.0.0.0 promisc up
ifconfig umltap0 0.0.0.0 promisc up
ifconfig br0 192.168.0.5 netmask 255.255.255.0 up
brctl stp br0 off
brctl setfd br0 1
brctl sethello br0 1
brctl addif br0 eth0
brctl addif br0 umltap0
route add default gw 192.168.0.1

#arranquem la instància uml
su umluser -c './linux ubd0=root_fs ubd1=swap mem=32M eth0=tuntap,umltap0'
```

Un cop a dins de la màquina virtual cal configurar la xarxa:

```
(none):~# ifconfig eth0 192.168.0.100 up
(none):~# ifconfig
eth0      Link encap:Ethernet  HWaddr FE:FD:C0:A8:02:64
          inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:5
```

I amb això ja tenim la màquina connectada amb l'exterior:

```
(none):~# ping -c 1 192.168.0.5
PING 192.168.0.5 (192.168.0.5): 56 data bytes
64 bytes from 192.168.0.5: icmp_seq=0 ttl=64 time=1.0 ms

--- 192.168.0.5 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
```



```
round-trip min/avg/max = 1.0/1.0/1.0 ms
```

De totes maneres, per anar bé caldria introduir les entrades pertinents a `/etc/network/interfaces` de manera que quan arranquem la màquina virtual ja tinguem xarxa:

```
auto lo
iface lo inet loopback

iface eth0 inet static
    address 192.168.0.100
    netmask 255.255.255.0
    gateway 192.168.0.1
```

(Per anar encara més bé caldria posar les entrades adequades a `/etc/resolv.conf`, per a resolució de noms).

Si tot ha anat bé, ja hauríem de tenir un Linux plenament funcional i connectat a la xarxa. Ara és quan (després d'assignar contrasenya a `root` i crear un compte d'usuari "normal") hauríem d'actualitzar el sistema i començar a instal·lar-hi aplicacions o, com un servidor, passar-nos primer de tot a SARGE o SID.

Notes finals

Mode skas

Per raons de seguretat i rendiment, és recomanable aplicar el patch `skas` al kernel del host. Jo no ho he fet perquè encara no ha sortit el patch per la versió del kernel que faig servir ara mateix i perquè de moment no ho poso a producció, on sí que ho considero imprescindible. Per més informació consulteu la pàgina sobre [skas](#)⁽¹⁴⁾.

Accés al host

Si voleu poder accedir al disc del host des de la màquina virtual, no necessiteu NFS ni FTP. Podeu fer-ho de manera molt còmoda per mitjà de [hostfs](#)⁽¹⁵⁾, simplement muntant el sistema de fitxers com qualsevol altre:

```
(none):~# mkdir /mnt/host
(none):~# mount -t hostfs none /mnt/host
(none):~# ls /mnt/host
bin  cdrom  etc      home      initrd  lost+found  mnt  proc  root-n  sys  usr  vmlinuz
boot dev  floppy  include  lib      man        opt  root  sbin   tmp  var
(none):~#
```

Evidentment, l'accés al sistema de fitxers del host es farà amb els privilegis que hi tingui l'usuari que executa UML. Malgrat tot, hi ha entorns en els que segurament no interressi que els usuaris de la màquina virtual accedeixin al host (per exemple, en entorns de hosting). En aquest cas, podem compilar el kernel UML sense suport per `hostfs`.

Com compartir sistemes de fitxers

Com a mesura per estalviar espai en disc, podeu compartir sistemes de fitxers entre diferents instàncies UML a través de [COW](#)⁽¹⁶⁾ (*copy-on-write*). El sistema de fitxers original serà de només lectura (si no, les instàncies UML entrarien en conflicte) i és la capa COW la que dóna accés de lectura-escritura a cada una de les instàncies. Això ho fa guardant i mantenint els canvis en un dispositiu privat de cadascuna. Per entendre'ns: quan des d'una instància UML s'escriu al sistema de fitxers compartit, es guarden les dades en un dispositiu de la pròpia instància, mentre que quan l'accés és de lectura la informació es va a buscar als dos dispositius (al privat i, si no hi és, al compartit).

Informació de depuració

Si seguïeu al peu de la lletra aquest article us trobareu amb un executable (`./linux`) extremadament voluminós ja que conté informació de debug. Per eliminar aquesta informació podeu executar la comanda:

```
~/uml$ strip linux
```

(Gràcies: *ivaniclixx*).



Enllaços

- <http://user-mode-linux.sourceforge.net>⁽¹⁷⁾. La pàgina del projecte. És realment completa (hi és TOT).
 - [User Mode Linux Howto](#)⁽¹⁸⁾. També és a la pàgina del projecte, però el poso perquè realment val la pena.
 - usermodelinux.org⁽¹⁹⁾. Grup d'usuaris de user mode linux
 - [Article interessant](#)⁽²⁰⁾. Del creador de la criatura.
-

Lista de enlaces de este artículo:

1. <http://www.vmware.com>
 2. <http://www.microsoft.com/windows/virtualpc>
 3. <http://bochs.sourceforge.net>
 4. <http://plex86.sourceforge.net>
 5. <http://www.colinux.org>
 6. <http://user-mode-linux.sourceforge.net>
 7. <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=260111>
 8. <http://www.kernel.org>
 9. <http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.26.tar.bz2>
 10. <http://belnet.dl.sourceforge.net/sourceforge/user-mode-linux/uml-patch-2.4.26-3>
 11. <http://user-mode-linux.sourceforge.net/dl-fs-sf.html>
 12. <http://belnet.dl.sourceforge.net/sourceforge/user-mode-linux/Debian-3.0r0.ext2.b>
 13. <http://user-mode-linux.sourceforge.net/networking.html>
 14. <http://user-mode-linux.sourceforge.net/skas.html>
 15. <http://user-mode-linux.sourceforge.net/hostfs.html>
 16. http://user-mode-linux.sourceforge.net/shared_fs.html
 17. <http://user-mode-linux.sourceforge.net/>
 18. <http://user-mode-linux.sourceforge.net/UserModeLinux-HOWTO.html>
 19. <http://usermodelinux.org>
 20. <http://www.linux-mag.com/cgi-bin/printer.pl?issue=2001-04&amp;amp;>
-

E-mail del autor: phalangan_a_ARROBA_yahoo.com

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2093>