Guía de DHCPlab
versión 0.6

Alberto Cortés Martín[1]
Traducción: Estrella García Lozano y Patricia Arias Cabarcos[2]

14 de octubre de 2008

[1]email: alcortes@it.uc3m.es
[2]email: emglozan@it.uc3m.es, parias@it.uc3m.es

Índice general

Capítulo 1

Introducción

1.1. Acerca de este documento

Este documento está dirigido a los alumnos de Redes de Ordenadores 2, quinto curso de Ingeniería Informática en la Universidad Carlos III de Madrid, durante el curso 2008-2009.

Aquí se explica como utilizar el entorno de pruebas montado en los laboratorios y que hemos llamado DHCPlab (o laboratorio de DHCP).

Dicho entorno de pruebas se basa en una simulación utilizando la máquina virtual `user-mode-linux`. Este documento también incluye una guía detallada sobre como preparar entornos `user-mode-linux` para simulación y pruebas.

Los nombres de comandos aparecen en un tipo de letra característico y a menudo van seguidos de la sección del manual, entre paréntesis, donde encontrar más información sobre ellos. Por ejemplo `ls(1)`, significa que se puede encontrar más información sobre el comando `ls` leyendo su página de manual en la sección 1, es decir, ejecutando:

```
hagrid; man 1 ls
```

Los ejemplos literales, trazas de ejecución y URLs utilizan el formato que precede a este párrafo.

En la portada, debajo del título, encontrarás la versión de este documento. Puedes bajarte la versión más reciente desde:

```
http://it.uc3m.es/alcortes/asignaturas/iro2/guia-dhcplab.↙
pdf
```

Si tienes comentarios, mejoras o has detectado erratas ponte en contacto con el autor, tienes su dirección de correo como nota al pié en la portada.

La distribución de este documento es ilimitada en cualquier formato, siempre que sea gratuita y el documento se mantenga íntegro. El uso de este documento no está restringido.

1.2. Organización de este documento

El capítulo 1 contiene una breve introducción de lo que es un cliente de DHCP, así como una descripción de cual debería ser el entorno ideal para desarrollar uno. Le sigue una discusión acerca de por que no podemos ofreceros dicho entorno ideal en aulas y la alternativa que hemos elegido, DHCPlab.

El capítulo 2 explica como utilizar DHCPlab.

El capítulo 3 es una descripción exhaustiva de como hemos montado DHCPlab, incluyendo una descripción completa de como montar un sistema `user-mode-linux`.

Si solo te interesa hacer la práctica, es suficiente con que leas el capítulo 1 y el capítulo 2. Si estás interesado en montar tus propios entornos `user-mode-linux`, lee el capítulo 3.

1.3. Un rápido vistazo a los clientes DHCP

La función principal de un cliente DHCP es configurar y levantar una interfaz de red de forma desatendida. Lo interesante es que para obtener los parámetros de dicha configuración, se dialoga con un servidor DHCP utilizando esa misma interfaz *desconfigurada*.

En general, la configuración de una interfaz de red se puede hacer manualmente o de forma automática. En la configuración manual, el administrador de la máquina conoce los parámetros necesarios para configurar la interfaz (IP, gateway, servidores DNS...) o se los pregunta al administrador de la red. Entonces configura la interfaz mediante el comando `ifconfig(8)`.

En la configuración automática, un cliente DHCP pregunta a un servidor DHCP por los parámetros necesarios y configura la interfaz de red sin intervención humana. Por supuesto, el administrador de la red ha tenido que configurar previamente el servidor DHCP para que responda a los clientes con la información adecuada.

El cliente DHCP tiene que correr con permisos de administrador para poder utilizar la interfaz no configurada y para configurarla posteriormente. El cliente tendrá que hacer cierta magia para poder utilizar la interfaz antes

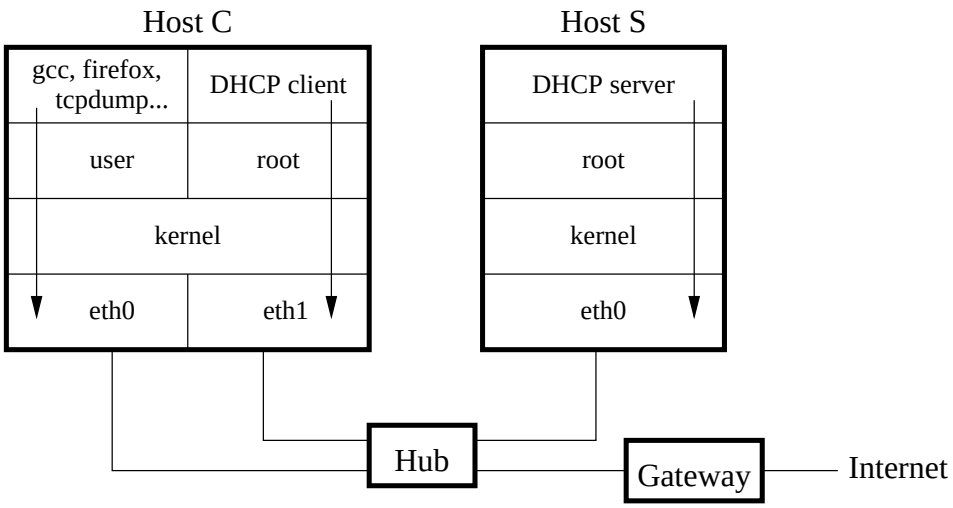

Figura 1.1: Laboratorio ideal para el desarrollo de un cliente DHCP

de su configuración, accediendo a funciones de nivel de enlace de la tarjeta de red que normalmente quedan ocultos tras los APIs de niveles superiores.

1.4. Un laboratorio ideal de DHCP

Según lo dicho en la sección anterior, un laboratorio ideal para el desarrollo de un cliente DHCP podría ser el que muestra en la figura 1.1. En ella se muestran dos máquinas, C y S, conectadas por un hub Ethernet a un gateway que les da salida a Internet.

La máquina S (servidor), tiene una única interfaz de red `eth0`. Un usuario con permisos de administrador (`root`) está ejecutando un servidor de DHCP en esa misma interfaz.

La máquina C (cliente), tiene dos interfaces de red: `eth0` y `eth1`. La interfaz `eth0` está siempre configurada y levantada, y es la que utiliza el usuario (`user`) para sus comunicaciones durante el desarrollo (consultar documentación en linea, por ejemplo). La interfaz `eth1` está normalmente sin configurar y es sobre la que se va a probar el cliente DHCP en desarrollo.

Cuando el usuario quiere probar su cliente, lanza un analizador de redes (por ejemplo `tcpdump(8)`) en la interfaz `eth0` y lanza su cliente desde un usuario con privilegios de administrador (`root`). Después de las pruebas, desconfigura manualmente la interfaz `eth1` para dejarlo preparado para la siguiente prueba.

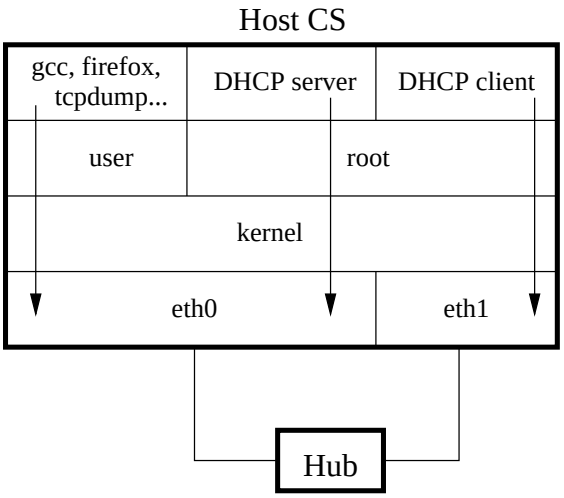

Figura 1.2: Laboratorio ideal simplificado, servidor y cliente corren en la misma máquina y hemos eliminado el acceso a Internet

Una versión simplificada de este mismo escenario es el que se muestra en la figura 1.2, donde servidor y cliente corren en la misma máquina, a la que hemos llamado CS (cliente-servidor). Preferimos esta aproximación a la anterior por que nos ahorramos una máquina.

Al mismo tiempo hemos eliminado el gateway y la conexión a internet aislando la máquina y el hub del mundo exterior. Esta simplificación es razonable si entendemos que el usuario dispone de otras máquinas conectadas a Internet mientas desarrolla el cliente DHCP.

De ahora en adelante cuando nos referimos al *escenario ideal*, nos estaremos refiriendo a la figura 1.2.

1.5. Problemas del escenario ideal

Lamentablemente, en los laboratorios de prácticas no se puede montar nada parecido a la figura 1.2. Las razones son diversas, pero probablemente la más importante es que los alumnos no tienen permisos de administrador sobre las máquinas por lo que no pueden lanzar el cliente ni configurar interfaces de red.

Entre las diversas soluciones que hay, hemos optado por ofreceros un entorno virtual, denominado DHCPlab, igual al de la figura 1.2. Por tanto para probar vuestro cliente en los laboratorios tendréis que ejecutar un hub

virtual y una máquina linux virtual con las características de la figura 1.2. Por supuesto, cuando trabajéis en casa o en otros entornos donde tengáis permisos de administrador, no os será necesario ejecutar las virtualizaciones para probar la práctica.

Capítulo 2

Instrucciones de uso de DHCPlab

2.1. Arrancar el entrono de pruebas virtual

Para arrancar el entorno de pruebas virtual es necesario arrancar el hub virtual y la máquina virtual (en este orden). Recuerda que este escenario de pruebas está pensado exclusivamente para que pruebes tu práctica no para que la desarrolles: codifica y compila normalmente en las máquina reales de los laboratorios. La máquina virtual y la máquina real que la alberga son compatibles a nivel binario y de arquitectura.

2.1.1. Arrancar el hub virtual

Para arrancar el hub virtual es necesario abrir un terminal y ejecutar el siguiente comando:

```
uml_switch -hub -unix ${HOME}/hub.ctl
```

Esta invocación lanza un proceso `uml_switch` que hace de hub virtual y crea un fichero control `$HOME/hub.ctl` que permite a otros procesos comunicarse con él.

El proceso que acabas de lanzar se queda corriendo bloqueando el terminal e irá mostrando por la salida estándar información útil sobre el estado de las conexiones que otros procesos establecen con el hub virtual.

Cuando quieras terminar el proceso que hace de hub virtual puedes hacerlo mandándole la señal de interrupción (SIGINT). Una forma fácil de hacerlo es presionar Control+c sobre el terminal que lo ejecuta.

2.1.2. Arrancar el linux virtual

Para arrancar el linux virtual es necesario abrir un terminal y ejecutar el siguiente comando:

```
linux mem=64M ubd0=${HOME}/dhcplab.cow,/var/dhcplab.rootfs↙
 eth0=daemon,fe:fd:00:00:01:00,unix,${HOME}/hub.ctl eth1=↙
daemon,fe:fd:00:00:01:01,unix,${HOME}/hub.ctl eth2=daemon,↙
fe:fd:00:00:01:02,unix,${HOME}/hub.ctl con=pty con0=fd:0,↙
fd:1 con1=xterm con2=xterm xterm=xterm,-T,-e umid=dhcplab
```

El arranque de la máquina virtual no suele tardar más de 1 minuto, pero en algunas de las máquinas de los laboratorios puede tardar hasta 20 minutos.

La invocación recomendada lanza un `user-mode-linux`[1] con la siguiente configuración:

- 64 Megas de memoria (robadas de la memoria de la máquina real).
- Su disco duro será el fichero `/var/dhcplab.rootfs`, que está protegido por un sistema de ficheros COW (*copy on write*) montado sobre un fichero en tu cuenta (`${HOME}/dhcplab.cow`).
- Tiene 3 interfaces de red, `eth0`, `eth1` y `eth2`, cada uno con las direcciones MAC especificadas y que están conectados al hub virtual que hemos lanzado antes.
- Tiene redirigidas los terminales serie a la entrada y salida estándar.
- Dos de los terminales virtuales aparecerán como `xterm`s en tu sesión de `X`.
- La máquina virtual tiene el identificador `dhcplab`, por si quieres acceder a su interfaz de control mediante el comando `uml_mconsole`

El fichero `/var/dhcplab.rootfs` también está disponible desde la URL:

```
http://it.uc3m.es/alcortes/asignaturas/iro2/dhcplab.rootfs
```

Esta configuración concreta es la que me parece más cómoda, pero puedes cambiarla y lanzar la máquina virtual como te apetezca. Consulta la documentación sobre `user-mode-linux` para cualquier detalle que quieras cambiar.

[1] la página de manual de `user-mode-linux` es `linux.uml(1)`

El fichero que hace de disco duro virtual y del que arranca la máquina virtual (`/var/dhcplab.rootfs`) ha sido poblado con una instalación básica de Debian Etch de mediados de septiembre de 2008. Aparte de los paquetes básicos, tiene instalado el siguiente software:

- `gdb(1)` por si quieres debugear tu cliente en la propia máquina virtual.
- `tcpdump(8)` para capturar paquetes de red.
- `valgrind(1)` para detectar fugas de memoria.
- `udhcpd`, un pequeño servidor de DHCP configurado para escuchar en la interfaz `eth0`. El servidor se arranca como demonio al iniciarse la máquina virtual. El fichero de configuración esta en `/etc/udhcp.conf`. Una vez modificado tendrás que reiniciar el demonio con el comando `/etc/init/udhcpd restart` desde la cuenta de root.
- `udhcpc`, un pequeño cliente DHCP que puedes usar para hacer pruebas y para compararlo con el que tu tienes que hacer en la práctica.

Creemos que este sistema básico tiene todo lo necesario para que puedas trabajar cómodamente, pero una vez más, si quieres hacer cualquier cambio o instalar más software eres libre de hacerlo, para eso estas lanzando la máquina virtual sobre un COW en tu cuenta.

La contraseña de la cuenta de root es `root`.

Para cerrar la máquina virtual invoca el comando `halt(8)` desde la cuenta de root. También puedes cerrar la máquina virtual accediendo a su interfaz de control con el comando `uml_mconsole(1)` desde la máquina que la alberga e invocando el comando de control `halt`.

2.2. Compartir ficheros con la máquina virtual

Aunque la máquina virtual no tiene ninguna conexión de red compartida con la máquina que la alberga se pueden compartir directorios entre ambas usando `hostfs` desde la máquina virtual. Por ejemplo, para montar el directorio `/etc` de la máquina real en el directorio `/mnt/host-etc` de la máquina virtual usaremos el siguiente comando:

```
dhcplab:~# mkdir -p /mnt/host-etc
dhcplab:~# mount none /mnt/host-etc -t hostfs -o /etc
dhcplab:~# cat /etc/hostname
dhcplab
dhcplab:~# cat /mnt/host-etc/hostname
```

```
hagrid
```

Es necesario compartir ficheros entre ambas máquinas. Por ejemplo, cuando quieras probar tu cliente tendrás que copiar el binario desde la máquina donde lo desarrollas a la máquina virtual.

Consulta las páginas de manual de `mount(8)` y `umount(8)` para más información sobre estos comandos.

DHCPlab ya está configurado para montar automáticamente el directorio `/tmp` de la máquina que lo aloja bajo el directorio `/mnt/host-tmp` de la máquina virtual.

2.3. Capturando tráfico en la máquina virtual

Utiliza el comando `tcpdump(8)` para espiar el tráfico que pasa por la red virtual. Por ejemplo, en uno de los terminales virtuales de DHCPlab lanza el siguiente comando:

```
dhcplab:~# tcpdump -s0 -i eth0 -A -x
```

Desde otro terminal puedes probar a lanzar un cliente DHCP sobre la interfaz `eth1` con el siguiente comando:

```
dhcplab:~# udhcpc -i eth1
```

Observarás que en el primer terminal se han ido mostrando el contenido de los paquetes intercambiados entre el cliente y el servidor DHCP. Termina la ejecución de `tcpdump` mandándole la señal de interrupción con un Control+c.

La opción `-w` de `tcpdump` te permite guardar el tráfico capturado en un fichero para interpretarlo posteriormente con el propio tcpdump o con otras aplicaciones mas visuales como `wireshark(1)`.

`Wireshark` no está instalado en DHCPlab, pero si en las máquinas del laboratorio, si quieres utilizarlo para visualizar tus tramas, deberías capturar el tráfico con `tcpdump` en la máquina virtual, guardándolo en un fichero y luego pasar ese fichero a la máquina real para poder leerlo desde `wireshark`.

Capítulo 3

Genera tu propio DHCPlab

3.1. Pasos previos

Utilizamos bash para el login en en las cuentas de usuario y de root.

```
hagrid; cat /etc/debian_version
4.0
hagrid; date
Tue Sep 16 20:25:17 CEST 2008
```

3.2. Instalación de user-mode-linux

```
hagrid; sudo aptitude install user-mode-linux uml-↙
utilities debootstrap fakeroot user-mode-linux-doc
...
```

La página del manual asociada a `User-mode-linux` es `linux.uml(1)`. El ejecutable que instalamos se denomina `linux`. La documentación es instalada en:

```
file:///usr/share/doc/user-mode-linux-doc/html/index.html
```

3.3. Creación del fichero rootfs

En primer lugar, se crea el fichero que rootfs de la máquina virtual. En este ejemplo se usa un disco duro virtual de 1GB. El nombre elegido para este proyecto es dhcplab:

```
hagrid; dd if=/dev/zero of=dhcplab.rootfs bs=1M seek=1024 ↙
count=0
0+0 records in
0+0 records out
0 bytes (0 B) copied, 8.0005e-05 seconds, 0.0 kB/s
```

En DHCPLab no queremos pocos ficheros, así que utilizamos lo siguiente en lugar de la instrucción anterior:

```
hagrid; dd if=/dev/zero of=dhcplab.rootfs bs=1M count=320
320+0 records in
320+0 records out
335544320 bytes (336 MB) copied, 6.20514 seconds, 54.1 MB/↙
s
```

A continuación, le damos formato. Por ejemplo `ext3`:

```
hagrid; /sbin/mkfs.ext3 -F dhcplab.rootfs
mke2fs 1.40-WIP (14-Nov-2006)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
81920 inodes, 327680 blocks
16384 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67633152
40 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information:↙
 done

This filesystem will be automatically checked every 26 ↙
mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to ↙
override.
```

3.4. Creación del fichero swapfs

Asignamos 64M para swapfs:

```
hagrid; dd if=/dev/zero of=dhcplab.swapfs bs=1M count=64
64+0 records in
64+0 records out
67108864 bytes (67 MB) copied, 1.31707 seconds, 51.0 MB/s
```

Damos formato swap:

```
hagrid; /sbin/mkswap dhcplab.swapfs
Setting up swapspace version 1, size = 67104 kB
no label, UUID=aa2dba9c-4e83-4c17-9e75-991b9de1eb57
```

3.5. Instalación de un sistema base en rootfs

En primer lugar, es necesario montar rootfs. Después, utilizaremos debootstrap para poblarlo:

```
hagrid; export MOUNT=/mnt/rootfs
hagrid; sudo mkdir -p $MOUNT
hagrid; sudo mount -o loop dhcplab.rootfs $MOUNT
```

Instalación de un sistema Debian básico:

```
hagrid; sudo debootstrap --include=ssh,udev,gcc,gdb,↙
tcpdump,less,make,valgrind,udhcpd,udhcpc etch $MOUNT http↙
://ftp.debian.org/debian
...
I: Base system installed successfully.
```

3.6. Establecimiento del sistema operativo virtual

Habrá 3 dispositivos ethernet:

- `eth0 = fe:fd:00:00:01:00`
- `eth0 = fe:fd:00:00:01:01`
- `eth0 = fe:fd:00:00:01:02`

Los dispositivos pueden ser configurados automáticamente por el sistema `udev` de la máquina virtual la primera vez que se arranca, pero es preferible que lo hagamos nosotros mismos:

```
hagrid; sudo vim $MOUNT/etc/udev/rules.d/z25_persistent-net.rules
hagrid; cat $MOUNT/etc/udev/rules.d/z25_persistent-net.rules
# This file was automatically generated by the /lib/udev/write_net_rules
# program, probably run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single line.
# MAC addresses must be written in lowercase.

SUBSYSTEM=="net", DRIVERS=="?*", ATTRS{address}=="fe:fd:00:00:01:00", NAME="eth0"

SUBSYSTEM=="net", DRIVERS=="?*", ATTRS{address}=="fe:fd:00:00:01:01", NAME="eth1"

SUBSYSTEM=="net", DRIVERS=="?*", ATTRS{address}=="fe:fd:00:00:01:02", NAME="eth2"
```

Para el establecimiento de la red, elegimos la dirección `192.168.100.0` para la interfaz `eth0` de la máquina virtual:

```
hagrid; sudo vim $MOUNT/etc/network/interfaces
hagrid; cat $MOUNT/etc/network/interfaces
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.100.0
    netmask 255.255.255.0
    broadcast 192.168.100.255
```

Establecimiento del nombre de host:

```
hagrid; export HOST_NAME="dhcplab"
hagrid; sudo sh -c "echo $HOST_NAME > $MOUNT/etc/hostname"
```

Establecimiento de la zona horaria:

```
hagrid; export TZ=Europe/Madrid
hagrid; sudo sh -c "echo $TZ > $MOUNT/etc/timezone"
```

```
hagrid; sudo rm -f $MOUNT/etc/localtime
hagrid; sudo ln -sf /usr/share/zoneinfo/$TZ $MOUNT/etc/↙
localtime
```

Establecimiento de información sobre cómo debe montarse el sistema de ficheros estático:

```
hagrid; sudo vim $MOUNT/etc/fstab
hagrid; cat $MOUNT/etc/fstab
/dev/ubda    /            ext3     defaults     0    1
/dev/ubdb    none         swap     sw           0    0
none         /proc        proc     defaults     0    0
none         /dev/shm     tmpfs    defaults     0    0
none         /mnt/host-tmp hostfs  defaults,/tmp  0    0
hagrid; sudo mkdir $MOUNT/mnt/host-tmp
```

Establecimiento de la contraseña de root para la máquina virtual:

```
hagrid; sudo /usr/sbin/chroot $MOUNT passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Copia de los módulos del kernel UML en rootfs:

```
hagrid; sudo cp -Rp /usr/lib/uml/modules/* $MOUNT/lib/↙
modules/
```

Configuración del servidor dhcp:

```
hagrid; sudo cp $MOUNT/usr/share/doc/udhcpd/examples/↙
udhcpd.conf $MOUNT/etc/udhcpd.conf
hagrid; sudo vim $MOUNT/etc/udhcpd.conf
hagrid; cat $MOUNT/etc/udhcpd.conf
start 192.168.100.20
end   192.168.100.50
interface       eth0
max_leases      30
remaining       yes
auto_time       7200 #(2 hours)
decline_time    3600 #(1 hour)
conflict_time   3600 #(1 hour)
offer_time      60 #(1 minute)
min_lease       60
lease_file      /var/lib/misc/udhcpd.leases
pidfile /var/run/udhcpd.pid
notify_file     dumpleases
#siaddr         192.168.0.22
#sname          zorak
#boot_file      /var/nfs_root
```

```
opt     dns     192.168.10.2 192.168.10.10
option  subnet  255.255.255.0
opt     router  192.168.10.2
option  dns     129.219.13.81
option  domain  local
option  lease   864000          # 10 days of seconds
```

Retirar rootfs del sistema operativo anfitrión:

```
hagrid; sudo umount $MOUNT
```

3.7. COW para rootfs

Debemos indicar a la máquina virtual que se desea utilizar un COW (Copy On Write) para rootfs de sólo lectura, esto mantendrá el sistema rootfs intacto, y todos los cambios serán efectuados sobre el COW.

```
hagrid; chmod a-w dhcplab.rootfs
```

3.8. Prueba de la máquina virtual

Ahora que tenemos un sistema rootfs básico con debian, podemos probarlo:

```
hagrid; uml_switch -hub -unix switch.ctl
hagrid; linux mem=64M ubd0=dhcplab.cow,dhcplab.rootfs ubd1=/
=dhcplab.swapfs eth0=daemon,fe:fd:00:00:01:00,unix,switch./
ctl eth1=daemon,fe:fd:00:00:01:01,unix,switch.ctl eth2=/
daemon,fe:fd:00:00:01:02,unix,switch.ctl con=pty con0=fd/
:0,fd:1 con1=xterm con2=xterm xterm=xterm,-T,-e umid=/
dhcplab
...
```

En las consolas virtuales 1 y 2 corre un terminal xterm. Puedes conectarte a otras consolas con minicom o con cualquier otro programa terminal. El número de dispositivo asociado se anuncia en el prompt de arranque de la máquina virtual. Por ejemplo:

Asignamos la consola virtual 3 a pty /dev/ptyp1

Luego nos conectamos usando:

```
hagrid; minicom -o -p /dev/ttyp1
...
```

También se puede ejecutar la consola uml en el kernel actual:

```
hagrid; uml_mconsole dhcplab
(mconsole)  halt
OK
```

3.9. Compartición de directorios

Es posible montar directorios de la máquina anfitrión en la máquina virtual. Para ello, se usa hostfs:

```
dhcplab:~# mkdir /mnt/host-etc
dhcplab:~# mount none /mnt/host-etc -t hostfs -o /etc
dhcplab:~# cat /etc/hostname
dhcplab
dhcplab:~# cat /mnt/host-etc/hostname
hagrid
dhcplab:~# umount /mnt/host-etc
```