

Oliver Graf
Kristen Brandt

Lab 4

Ejercicio 1:

```
osreader@OSC: ~
File Edit View Search Terminal Help
osreader@OSC:~$ sudo apt-get install systemtap
[sudo] password for osreader:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
libdw libsystemtap-common libsystemtap-runtime
Suggested packages:
systemtap-doc vim-addon-manager
The following NEW packages will be installed:
libdw libsystemtap-common libsystemtap-runtime
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,715 kB of archives.
After this operation, 6,907 kB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

```
File Edit View Search Terminal Help
GNU nano 2.2.6           File: profiler.stp          Modified
probe timer.profile{
    printf("Proceso: %s\n", execname())
    printf("ID del proceso: %d\n",pid())
}
```

sudo stap profiler.stp:

```
osreader@OSC: ~/Documents/lab4
File Edit View Search Terminal Help
ID del proceso: 8
Proceso: swapper/1
ID del proceso: 8
Proceso: swapper/3
ID del proceso: 8
Proceso: swapper/2
ID del proceso: 8
Proceso: swapper/0
ID del proceso: 8
Proceso: swapper/1
ID del proceso: 8
Proceso: swapper/3
ID del proceso: 8
Proceso: swapper/2
ID del proceso: 8
Proceso: swapper/0
ID del proceso: 8
Proceso: stапlo
ID del proceso: 1367
Proceso: swapper/3
ID del proceso: 8
Proceso: swapper/2
ID del proceso: 8
```



A screenshot of a terminal window titled "osreader@OSC: ~/Documents/lab4". The window has a standard OS X-style title bar with icons for close, minimize, and zoom. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane displays the following text:

```
ID del proceso: 0
Proceso: swapper/1
ID del proceso: 0
Proceso: swapper/0
ID del proceso: 0
Proceso: swapper/2
ID del proceso: 0
Proceso: swapper/3
ID del proceso: 0
Proceso: swapper/1
ID del proceso: 0
Proceso: swapper/0
ID del proceso: 0
Proceso: swapper/2
ID del proceso: 0
Proceso: swapper/3
ID del proceso: 0
Proceso: swapper/1
ID del proceso: 0
Proceso: swapper/0
ID del proceso: 0
Proceso: swapper/2
ID del proceso: 0
```

- ¿Qué puede ver en el *output* cuando realiza estas acciones?

Lo que se puede ver en el *output* es que cuando uno realiza algún tipo de movimiento o click el ID del proceso es mucho mas alto.

- ¿Para qué sirve SystemTap?

SystemTap proporciona una infraestructura de software libre (GPL) para simplificar la recopilación de información sobre el sistema Linux en ejecución, lo que ayuda a diagnosticar un problema funcional o de rendimiento. SystemTap elimina la necesidad de que el desarrollador tenga que pasar por el tedioso y perturbador instrumento, recompilar, instalar y reiniciar la secuencia que de otro modo podría ser necesaria para recopilar datos.

- ¿Qué es una *probe*?

Un probe del sistema operativo es para supervisar los recursos del sistema, el uso de la CPU, la actividad del disco, el uso de la memoria física y el tráfico de la red. Supervisa la cantidad de tráfico de red en el servidor. El tipo de probe más simple es rastrear un evento.

- ¿Cómo funciona SystemTap?

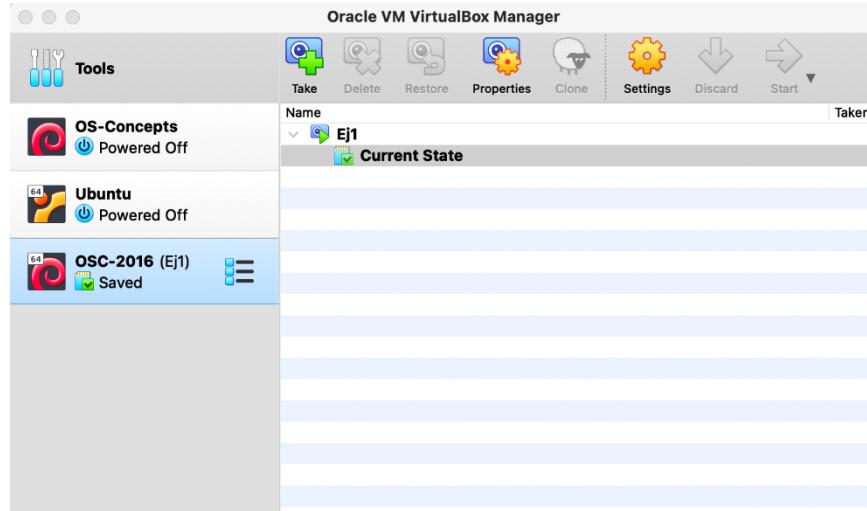
SystemTap proporciona una interfaz de línea de comandos simple y un lenguaje de scripting para escribir instrumentación para un kernel en ejecución más aplicaciones de espacio de usuario. Estamos publicando muestras, así como ampliando la biblioteca de scripts "tapset" interna para ayudar a la reutilización y abstracción.

- ¿Qué es hacer *profiling* y qué tipo de *profiling* se hace en este ejercicio?

Hacer profiling es una forma de análisis dinámico de programas que mide el espacio (memoria) o la complejidad temporal de un programa, el uso de instrucciones particulares o la frecuencia y duración de las llamadas a funciones. El tipo de profiling que se hace en este ejercicio es

timer.profile es un temporizador que se activa periódicamente en cada CPU.

Ejercicio 2:



```
*simple.c
#ifndef _SIMPLE_H_
#define _SIMPLE_H_
#include <linux/init.h>
#include <linux/module.h>
#include <linux/list.h>
int simple_init(void)
{
    printk(KERN_INFO "Loading Module\nwwwlwa");
    return 0;
}
void simple_exit(void)
{
    printk(KERN_INFO "Removing Module\nwwwlwa");
}
module_init(simple_init);
module_exit(simple_exit);
MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Ejercicio2");
MODULE_AUTHOR("Oliver y Kristen");

```

- ¿Cuál es la diferencia en C entre un método que no recibe parámetros y uno que recibe void?

La diferencia entre un método que no recibe parámetros y uno que recibe void es que el método que no toma ningún argumento. El método que no recibe parámetros es un método que puede tomar cualquier cosa por sus argumentos.

- ¿Qué diferencia hay entre `printf` y `printk`?

`Printf`: Escribe la cadena C apuntada por formato en la salida estándar (stdout). Si el formato incluye especificadores de formato (subsecuencias que comienzan con%), los argumentos

adicionales que siguen al formato se formatean y se insertan en la cadena resultante reemplazando sus respectivos especificadores.

Printk: Función C de la interfaz del kernel de Linux que imprime mensajes en el registro del kernel. Printk acepta un parámetro de cadena llamado cadena de formato, que especifica un método para convertir un número arbitrario de parámetros de tipo de datos variados en una cadena, que luego se imprime en el registro del kernel.

- ¿Qué es y para qué sirve KERN_INFO?

KERN_INFO es el nivel de registro utilizado para mensajes informativos sobre la acción realizada por el kernel. Este sirve al ser utilizado dentro de un printk(KERN_INFO "mensaje").

```
oscreader@OSC: ~/Documents/lab4
File Edit View Search Terminal Help
GNU nano 2.2.6          File: Makefile           Modified
obj-m := simple.o
all:
    make -C /lib/modules/$(shell uname -r)/build M=$(shell pwd) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(shell pwd) clean
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Insertfu Where Te Navt Dancs Imprt Text Tn Snail
```

- ¿Qué es una **goal definition** o definición de meta en un *Makefile*, y qué se está haciendo con la definición de meta obj-m?

Las goal definition de meta en un makefile hacen que, en última instancia, conviene esforzarse por actualizar. De forma predeterminada, el objetivo es el primer objetivo en el archivo MAKE (sin contar los objetivos que comienzan con un punto). Por lo tanto, los archivos MAKE generalmente se escriben de manera que el primer objetivo sea compilar todo el programa o los programas que describen. Con la definición de meta obj-m se especifican archivos objeto que se crean como cargables módulos del kernel. Un módulo puede construirse a partir de un archivo fuente o de varias fuentes y/o archivos. En el caso de un archivo fuente, kbuild makefile simplemente agrega el archivo a obj-m.

- ¿Qué función tienen las líneas all: y clean:?

Cuando make se ejecuta sin argumentos, el primer objetivo encontrado será construido. En el nivel superior Makefile el primer objetivo presente es all:

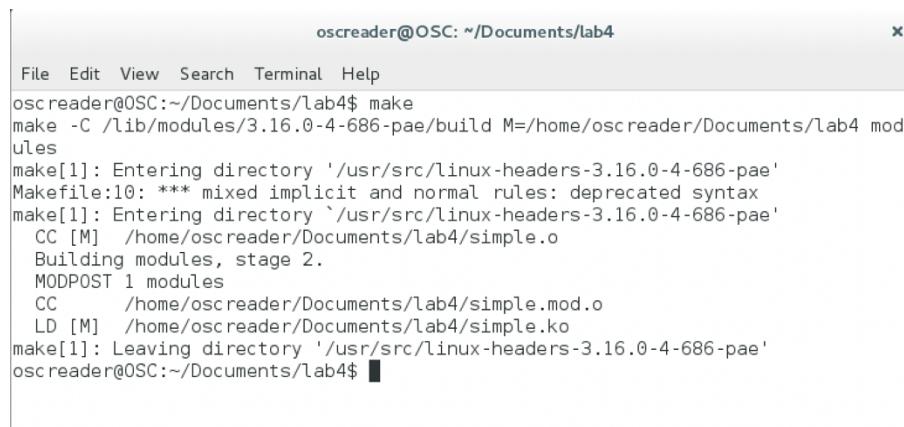
Clean: es una regla opcional. Esta permite escribir en la línea de comando para deshacerse de su objeto y archivos ejecutables. A veces, el compilador vinculará o compilará archivos incorrectamente y la única forma de comenzar de nuevo es eliminar todos los archivos de objetos y ejecutables.

- ¿Qué hace la opción -C en este *Makefile*?

La opción -c en Makefile es cambiar al directorio dir antes de leer los archivos MAKE. Si se especifican varias opciones '-C', cada una se interpreta en relación con la anterior '-C'.

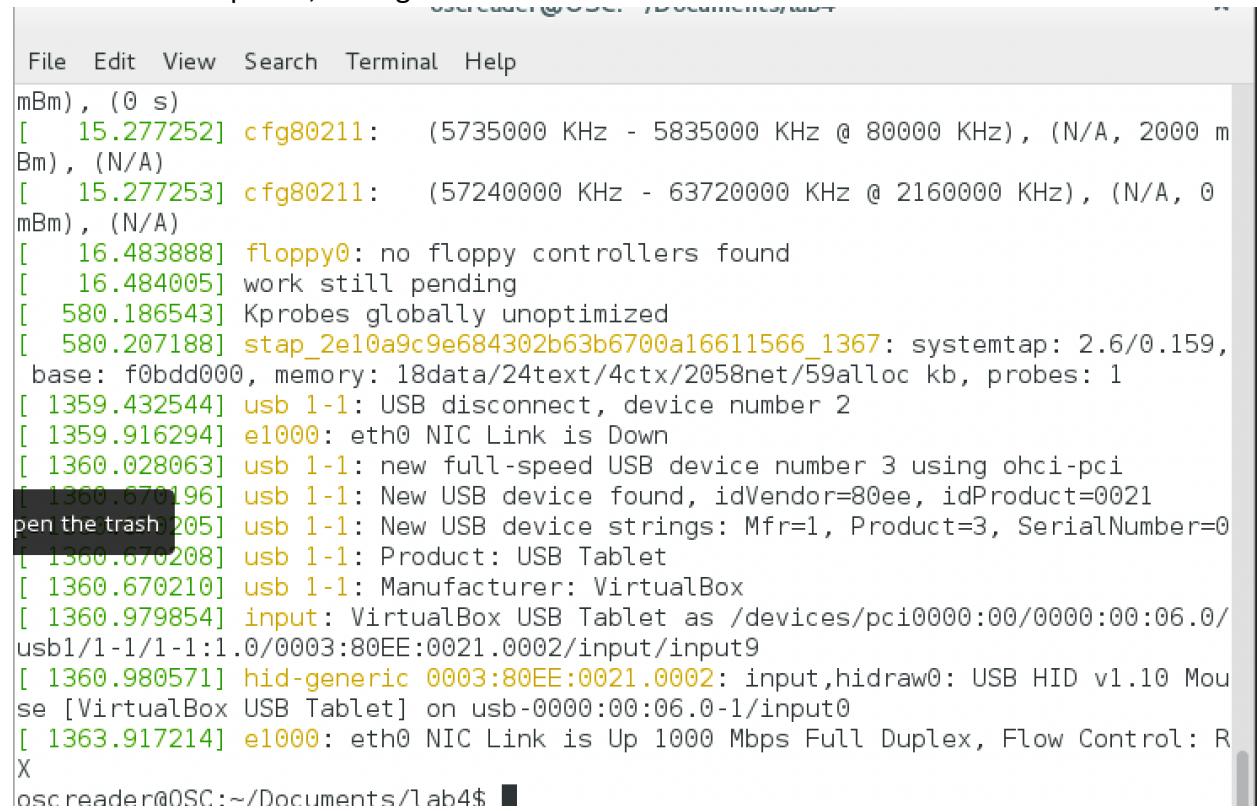
- ¿Qué hace la opción M en este *Makefile*?

La opción M en el Makefile es para especificar en que dirección está presente el módulo.



```
oscreader@OSC: ~/Documents/lab4
File Edit View Search Terminal Help
oscreader@OSC:~/Documents/lab4$ make
make -C /lib/modules/3.16.0-4-686-pae/build M=/home/oscreader/Documents/lab4 modules
make[1]: Entering directory '/usr/src/linux-headers-3.16.0-4-686-pae'
Makefile:10: *** mixed implicit and normal rules: deprecated syntax
make[1]: Entering directory `/usr/src/linux-headers-3.16.0-4-686-pae'
  CC [M]  /home/oscreader/Documents/lab4/simple.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/oscreader/Documents/lab4/simple.mod.o
  LD [M]  /home/oscreader/Documents/lab4/simple.ko
make[1]: Leaving directory '/usr/src/linux-headers-3.16.0-4-686-pae'
oscreader@OSC:~/Documents/lab4$
```

sudo insmod simple.ko; dmesg:



```
File Edit View Search Terminal Help
mBm), (0 s)
[ 15.277252] cfg80211: (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A, 2000 m
Bm), (N/A)
[ 15.277253] cfg80211: (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 0
mBm), (N/A)
[ 16.483888] floppy0: no floppy controllers found
[ 16.484005] work still pending
[ 580.186543] Kprobes globally unoptimized
[ 580.207188] stap_2e10a9c9e684302b63b6700a16611566_1367: systemtap: 2.6/0.159,
base: f0bdd000, memory: 18data/24text/4ctx/2058net/59alloc kb, probes: 1
[ 1359.432544] usb 1-1: USB disconnect, device number 2
[ 1359.916294] e1000: eth0 NIC Link is Down
[ 1360.028063] usb 1-1: new full-speed USB device number 3 using ohci-pci
[ 1360.670196] usb 1-1: New USB device found, idVendor=80ee, idProduct=0021
pen the trash [205] usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
[ 1360.670208] usb 1-1: Product: USB Tablet
[ 1360.670210] usb 1-1: Manufacturer: VirtualBox
[ 1360.979854] input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/
usb1/1-1/1-1:1.0/0003:80EE:0021.0002/input/input9
[ 1360.980571] hid-generic 0003:80EE:0021.0002: input,hidraw0: USB HID v1.10 Mou
se [VirtualBox USB Tablet] on usb-0000:00:06.0-1/input0
[ 1363.917214] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: R
X
oscreader@OSC:~/Documents/lab4$
```

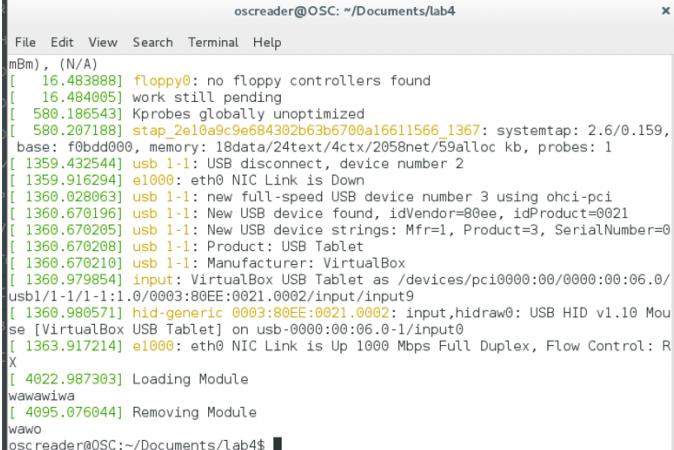
- ¿Para qué sirve dmesg?

La función dmesg sirve para examinar o controlar el kernel ring buffer. Esta función ayuda a imprimir los mensajes de bootup de la máquina/módulo.

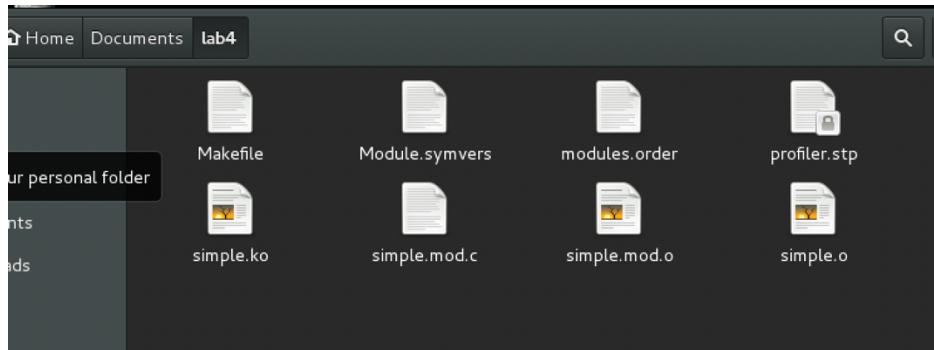
- ¿Qué hace la función simple_init en su programa simple.c?

Esta función inicializa nuestro módulo e imprime el proceso.

`sudo rmmod simple, demsg:`



```
oscreader@OSC: ~/Documents/lab4
File Edit View Search Terminal Help
mBm), (N/A)
[ 16.483888] floppy0: no floppy controllers found
[ 16.484005] work still pending
[ 580.186543] Kprobes globally unoptimized
[ 580.207188] stап_2e10a9c9e684302b63b6700a16611566 1367: systemtap: 2.6/0.159,
base: f0bdd000, memory: 18data/24text/4ctx/2058net/59alloc kb, probes: 1
[ 1359.432544] usb 1-1: USB disconnect, device number 2
[ 1359.916294] e1000: eth0 NIC Link is Down
[ 1360.028063] usb 1-1: new full-speed USB device number 3 using ohci-pci
[ 1360.670196] usb 1-1: New USB device found, idVendor=00ee, idProduct=0021
[ 1360.670205] usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
[ 1360.670208] usb 1-1: Product: USB Tablet
[ 1360.670210] usb 1-1: Manufacturer: VirtualBox
[ 1360.979854] input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/
usb1/1-1:1.0/0003:00EE:0021.0002/input/input9
[ 1360.980571] hid-generic 0003:00EE:0021.0002: input,hidraw0: USB HID v1.10 Mouse [VirtualBox USB Tablet] on usb-0000:00:06.0-1/input0
[ 1363.917214] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 4022.987303] Loading Module
wawawiwa
[ 4095.076044] Removing Module
wawo
osc reader@OSC:~/Documents/lab4$
```



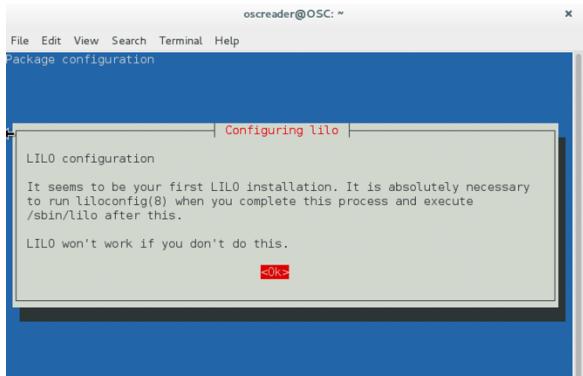
- ¿Qué hace la función simple_exit en su programa simple.c?

La función finaliza nuestro módulo y lo desinstala.

- Usted ha logrado crear, cargar y descargar un módulo de Linux. ¿Qué poder otorga el ejecutar código de esta forma?

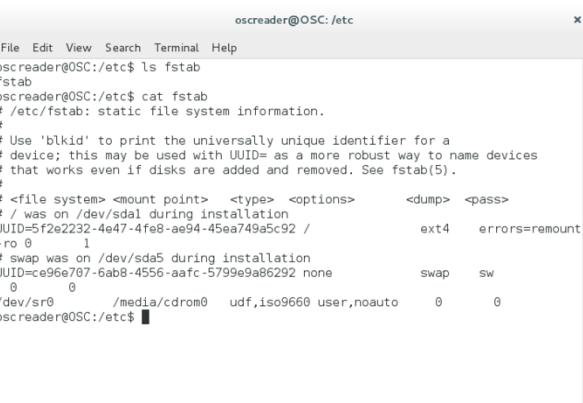
Trabajar con módulos permite reutilizar código para evitar la redundancia en nuestros programas. Además, programar de manera modular permite tener código mucho más limpio y elegante, porque abstrae funciones que tenga que cumplir el programa.

Ejercicio 3:



The terminal window shows the configuration of the LILO boot loader. It displays a message about the first LILO installation and the necessity to run liloconfig(8) after completion. A red "OK" button is visible at the bottom right of the message box.

```
oscreader@OSC:~$ cd /dev/disk/by-id
oscreader@OSC:/dev/disk/by-id$ ls -Al
total 0
lrwxrwxrwx 1 root root 9 Mar 26 01:16 ata-VBOX_CD-ROM_VB2-01700376 -> ../../sr0
lrwxrwxrwx 1 root root 9 Mar 26 01:16 ata-VBOX_HARDDISK_VB48e5ec32-d284bfd1 ->
[REDACTED]sda
lrwxrwxrwx 1 root root 10 Mar 26 01:16 ata-VBOX_HARDDISK_VB48e5ec32-d284bfd1-par
t1 -> ../../sda1
lrwxrwxrwx 1 root root 10 Mar 26 01:16 ata-VBOX_HARDDISK_VB48e5ec32-d284bfd1-par
t2 -> ../../sda2
lrwxrwxrwx 1 root root 10 Mar 26 01:16 ata-VBOX_HARDDISK_VB48e5ec32-d284bfd1-par
t5 -> ../../sda5
oscreader@OSC:/dev/disk/by-id$
```

The terminal window shows the contents of the /etc/fstab file. It lists the filesystems mounted on the system, including the root partition (ext4), swap space (swap), and other mounted volumes.

```
oscreader@OSC:/etc$ ls fstab
fstab
oscreader@OSC:/etc$ cat fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sdal during installation
UUID=5f2e2232-4e47-4fe8-ae94-45ea749a5c92 / ext4 errors=remount
-ro 0 1
# swap was on /dev/sda5 during installation
UUID=c96e707-6ab8-4556-aaf8-5799e9a86292 swap swap
0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
oscreader@OSC:/etc$
```

- ¿Qué es y para qué sirve el archivo `fstab`?

El archivo `fstab` contiene descripciones de los filesystems que uno (la computadora) monta frecuentemente. Este archivo sirve para montar filesystems automáticamente cuando se hace un `system boot`.

- ¿Qué almacena el directorio `/etc`? ¿En Windows, quién (hasta cierto punto) fungie como `/etc`?

Es una carpeta que contiene todos los archivos de configuración del sistema. En Windows hay un archivo llamado hosts que se encuentra dentro de la carpeta etc. Este archivo funge como /etc.

- ¿Qué se almacena en /dev y en /dev/disk?

En el directorio /dev están todos los archivos de dispositivo especiales para todos los dispositivos. En /dev/disk se almacena información adicional sobre las particiones del sistema.

The screenshot shows a terminal window with the title 'oscreader@OSC: /etc'. The window contains the configuration file 'lilo.conf' from the 'nano' editor. The file content includes boot parameters like 'boot=' and 'root=' pointing to specific disk partitions, and kernel configurations for 'vmlinuz' and 'vmlinuz.old'. At the bottom of the screen, there is a status bar with various keyboard shortcuts for navigating the file.

```
oscreader@OSC: /etc
File Edit View Search Terminal Help
GNU nano 2.2.6           File: lilo.conf          Modif.

boot= ata-VBOX_HARDDISK_VB48e5ec32-d284bfd1
compact
default= Linux
delay=40
install=menu
large-memory
tba32
map=/boot/map
root="UUID=5f2e2232-4e47-4fe8-ae94-45ea749a5c92"
read-only
vga=normal
image=/boot/vmlinuz
    label=Linux
    initrd=/boot/initrd.img
image=/boot/vmlinuz.old
    label=LinuxOld
    initrd=/boot/initrd.img.old
optional

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit      ^J Justifyv ^W Where Is ^V Next Page ^U UnCut Text ^T To Spcl
```

- ¿Por qué se usa <la dirección completa del link hacia sda> en lugar de sólo /dev/sda, y cuál es el papel que el programa udev cumple en todo esto?

Se utiliza el link completo porque debe poder ser accedido desde cualquier directorio y es una dirección específica. Udev se encarga de proporcionar eventos de dispositivo, detectando los dispositivos que están conectados a la computadora.

- ¿Qué es un *block device* y qué significado tiene `sdxN`, donde `x` es una letra y `N` es un número, en direcciones como /dev/sdb? Investigue y explique los conceptos de *Master Boot Record* (MBR) y *Volume Boot Record* (VBR), y su relación con UEFI.

Un block device es un archivo que hace referencia a un dispositivo. Estos archivos se distinguen de los archivos de caracteres porque proporcionan acceso al dispositivo sin mostrar las características del hardware del dispositivo. El MBR es la primera partición de la memoria secundaria que le dice a la computadora cómo cargar el sistema operativo. Mientras que el VBR

está ubicado al principio de una partición de la memoria secundaria y ayuda en el boot similarmente a cómo funciona el MBR.

- ¿Qué es hacer *chain loading*?

Es reemplazar un programa que se está ejecutando con otro nuevo, utilizando un espacio en la memoria compartida para trasladar datos del programa actual al programa nuevo.

- ¿Qué se está indicando con la configuración `root="<el file system anotado>"`?

Es para especificar cual sección del disco utilizar para cargar el kernel/SO.

```
GNU nano 2.2.6           File: kernel-img.conf           Modified

# Kernel image management overrides
# See kernel-img.conf(5) for details
do_symlinks = yes
relative_links = yes
do_bootloader = no
do_initrd = yes
link_in_boot = yes

I

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U Uncut Text^T To Spell
```

```
File Edit View Search Terminal Help
oscreader@OSC:/$ ls
bin dev home lib media opt root sbin sys usr vmlinuz
boot etc initrd.img lost+found mnt proc run srv tmp var
oscreader@OSC:/$ sudo rm vmlinuz
oscreader@OSC:/$ sudo rm initrd.img
oscreader@OSC:/$ █
```

```
oscreader@OSC:/boot$ sudo ln -s initrd.img-3.16.0-4-686-pae initrd.img
oscreader@OSC:/boot$ sudo ln -s vmlinuz-3.16.0-4-686-pae vmlinuz
oscreader@OSC:/boot$ ls
coffee.bmp          initrd.img          tuxlogo.bmp
config-3.16.0-4-686-pae  initrd.img-3.16.0-4-686-pae  vmlinuz
debian.bmp          inside.bmp          vmlinuz-3.16.0-4-686-pae
debian-de.bmp       onlyblue.bmp
debianlilo.bmp      System.map-3.16.0-4-686-pae
oscreader@OSC:/boot$
```

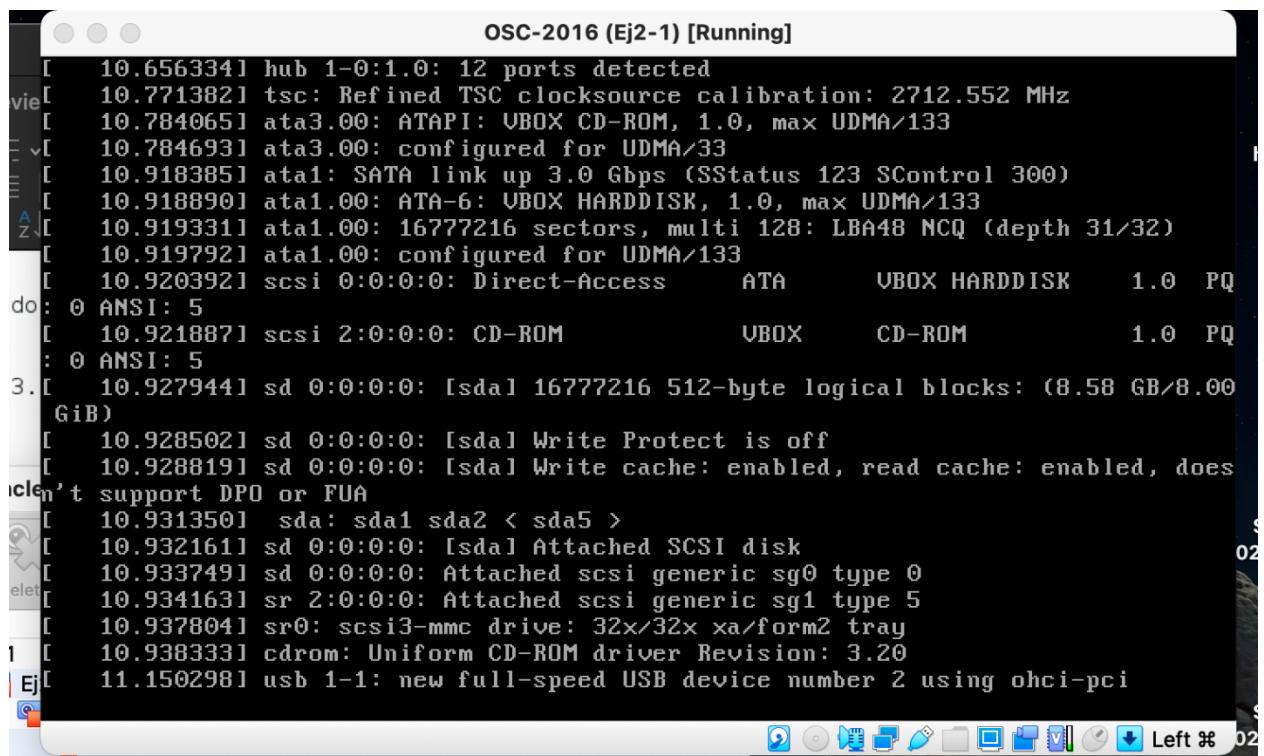
- ¿Qué es vmlinuz?

Es el ejecutable del kernel de Linux. Éste carga el sistema operativo en la memoria principal. Vmlinuz realiza una rutina que configura el hardware al hacer el boot del sistema.

```
oscreader@OSC:$ cd etc/kernel
oscreader@OSC:/etc/kernel$ ls
postinst.d  postrm.d
oscreader@OSC:/etc/kernel$ cd postinst.d
oscreader@OSC:/etc/kernel/postinst.d$ ls
apt-auto-removal  initramfs-tools  zz-runlilo  zz-update-grub
oscreader@OSC:/etc/kernel/postinst.d$ sudo rm zz-update-grub
oscreader@OSC:/etc/kernel/postinst.d$ cd ..
oscreader@OSC:/etc/kernel$ ls
postinst.d  postrm.d
oscreader@OSC:/etc/kernel$ cd postrm.d
oscreader@OSC:/etc/kernel/postrm.d$ ls
initramfs-tools  zz-runlilo  zz-update-grub
oscreader@OSC:/etc/kernel/postrm.d$ sudo rm zz-update-grub
oscreader@OSC:/etc/kernel/postrm.d$ ls
initramfs-tools  zz-runlilo
```

```
oscreader@OSC:/etc/initramfs/post-update.d$ sudo dpkg-reconfigure linux-image-3.16.0-4-686-pae
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-3.16.0-4-686-pae
/etc/kernel/postinst.d/zz-runlilo:
Added Linux *
Skipping /boot/vmlinuz.old
oscreader@OSC:/etc/initramfs/post-update.d$
```

Con LILO:



```
OSC-2016 (Ej2-1) [Running]
[ 10.656334] hub 1-0:1.0: 12 ports detected
[ 10.771382] tsc: Refined TSC clocksource calibration: 2712.552 MHz
[ 10.784065] ata3.00: ATAPI: VBOX CD-ROM, 1.0, max UDMA/133
[ 10.784693] ata3.00: configured for UDMA/33
[ 10.918385] ata1: SATA link up 3.0 Gbps (SStatus 123 SControl 300)
[ 10.918890] ata1.00: ATA-6: VBOX HARDDISK, 1.0, max UDMA/133
[ 10.919331] ata1.00: 16777216 sectors, multi 128: LBA48 NCQ (depth 31/32)
[ 10.919792] ata1.00: configured for UDMA/133
[ 10.920392] scsi 0:0:0:0: Direct-Access      ATA      VBOX HARDDISK 1.0 PQ
do: 0 ANSI: 5
[ 10.921887] scsi 2:0:0:0: CD-ROM          VBOX    CD-ROM      1.0 PQ
: 0 ANSI: 5
3.[ 10.927944] sd 0:0:0:0: [sda] 16777216 512-byte logical blocks: (8.58 GB/8.00
GiB)
[ 10.928502] sd 0:0:0:0: [sda] Write Protect is off
[ 10.928819] sd 0:0:0:0: [sda] Write cache: enabled, read cache: enabled, does
n't support DPO or FUA
[ 10.931350] sda: sda1 sda2 < sda5 >
[ 10.932161] sd 0:0:0:0: [sda] Attached SCSI disk
[ 10.933749] sd 0:0:0:0: Attached scsi generic sg0 type 0
[ 10.934163] sr 2:0:0:0: Attached scsi generic sg1 type 5
[ 10.937804] sr0: scsi3-mmc drive: 32x/32x xa/form2 tray
[ 10.938333] cdrom: Uniform CD-ROM driver Revision: 3.20
[ 11.150298] usb 1-1: new full-speed USB device number 2 using ohci-pci
```

- Mencione tres diferencias funcionales entre GRUB y LILO.
 - GRUB puede utilizarse con varios sistemas operativos, mientras que LILO solo soporta Linux.
 - LILO es más fácil y sencillo de utilizar que GRUB.
 - GRUB ofrece una interfaz gráfica mientras que LILO no.