

Lab#4 – Módulos de Kernel, SystemTap, Grub

```
oscreader@OSC: ~  
File Edit View Search Terminal Help  
oscreader@OSC:~$ sudo apt-get install systemtap  
[sudo] password for oscreader:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  libdw1 systemtap-common systemtap-runtime  
Suggested packages:  
  systemtap-doc vim-addon-manager  
The following NEW packages will be installed:  
  libdw1 systemtap systemtap-common systemtap-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]
```

Ponemos “Y”

```
oscreader@OSC: ~  
File Edit View Search Terminal Help  
Preparing to unpack .../libdw1_0.159-4.2_i386.deb ...  
Unpacking libdw1:i386 (0.159-4.2) ...  
Selecting previously unselected package systemtap-runtime.  
Preparing to unpack .../systemtap-runtime_2.6-0.2_i386.deb ...  
Unpacking systemtap-runtime (2.6-0.2) ...  
Selecting previously unselected package systemtap-common.  
Preparing to unpack .../systemtap-common_2.6-0.2_all.deb ...  
Unpacking systemtap-common (2.6-0.2) ...  
Selecting previously unselected package systemtap.  
Preparing to unpack .../systemtap_2.6-0.2_i386.deb ...  
Unpacking systemtap (2.6-0.2) ...  
Processing triggers for man-db (2.7.0.2-5) ...  
Setting up libdw1:i386 (0.159-4.2) ...  
Setting up systemtap-runtime (2.6-0.2) ...  
Adding stapdev group...  
Adding stapusr group...  
Adding stapsys group...  
Setting up systemtap-common (2.6-0.2) ...  
ERROR: systemtap-common is broken - called emacs-package-install as a new-style  
add-on, but has no compat file.  
Install systemtap-common for emacs  
Setting up systemtap (2.6-0.2) ...  
Processing triggers for libc-bin (2.19-18+deb8u1) ...  
oscreader@OSC:~$
```

```
oscreader@OSC: ~  
File Edit View Search Terminal Help  
GNU nano 2.2.6 New Buffer Modified  
probe timer.profile{  
    printf("Proceso: %s\n", execname())  
    printf("ID del proceso: %d\n", pid())  
}  
  
^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
```

```
oscreader@OSC: ~  
File Edit View Search Terminal Help  
ID del proceso: 0  
Proceso: swapper/3  
ID del proceso: 0  
Proceso: swapper/0  
ID del proceso: 0  
Proceso: swapper/1  
ID del proceso: 0  
Proceso: swapper/2  
ID del proceso: 0  
Proceso: swapper/1  
ID del proceso: 0  
Proceso: swapper/0  
ID del proceso: 0  
Proceso: swapper/3  
ID del proceso: 0  
Proceso: swapper/2  
ID del proceso: 0  
Proceso: stapio  
ID del proceso: 2867  
Proceso: swapper/0  
ID del proceso: 0  
Proceso: swapper/1  
ID del proceso: 0
```

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

Se mantiene en el proceso con ID 0 y cambia cuando se mueve el mouse o se hace alguna acción. Como se puede ver en la imagen anterior, hay un proceso con el ID de 2867.

¿Para qué sirve SystemTap?

Es un software libre (GPL) para facilitar junta de información sobre el sistema Linux en ejecución. Puede ser útil para encontrar problemas de funcionamiento. Nos brinda una interfaz de línea de comandos simple y un lenguaje de secuencias de comandos 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 una probe?

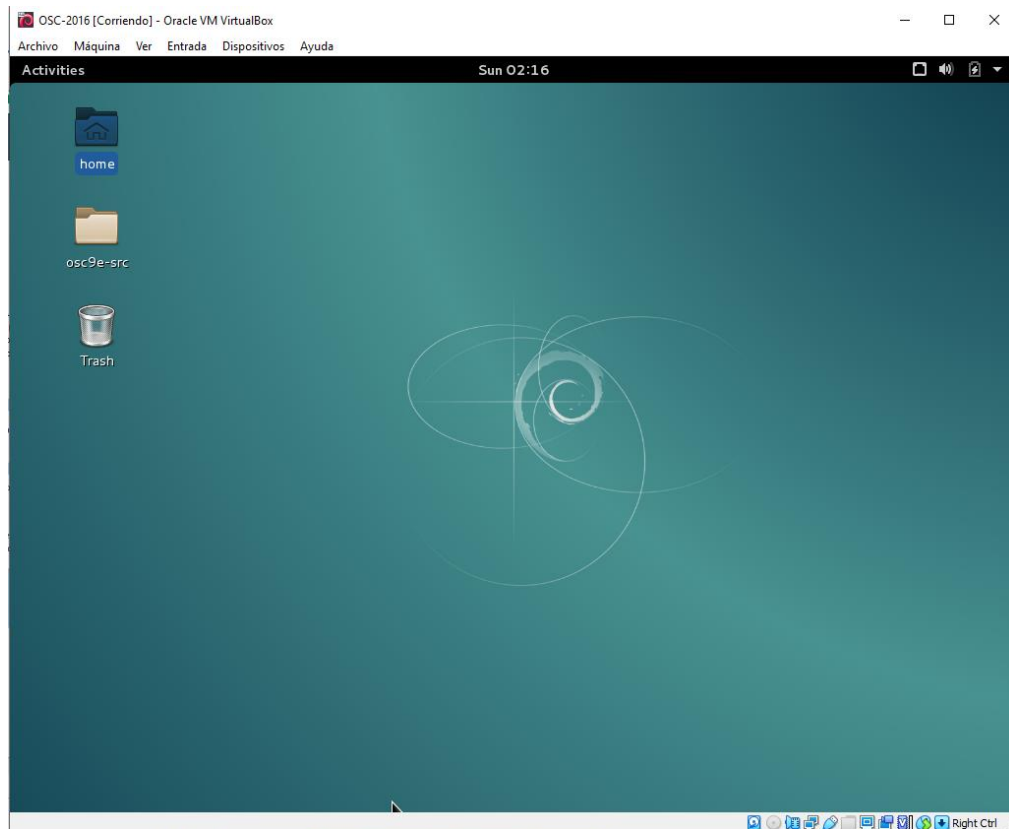
Es un programa u otro dispositivo que tiene una forma de resolver una colisión al hacer hash de números en una matriz.

¿Cómo funciona SystemTap?

Systemtap nombra eventos para luego ponerle handlers. Cada vez que pasa un evento específico, el kernel de Linux ejecuta el controlador como si fuera una subrutina rápida y luego se reanuda. SystemTap pasa el script a C, y se ejecuta para crear un módulo del kernel. Una vez cargado el módulo, activa todos los eventos probados montandolos al kernel. Este proceso se da por medio de "stap".

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

Es una forma de análisis dinámico de programas que mide, por ejemplo, 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. Por lo general, la información de creación de perfiles sirve para ayudar a la optimización del programa y, más específicamente, a la ingeniería del rendimiento. Este ejercicio hace uso del tipo de profiling llamado event-based porque se evalúa un proceso por medio de un cronometraje.



```
Open ▾ [icon] simple.c ~/
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/list.h>

int simple_init(void){
    printk(KERN_INFO "Loading Module\n Algo");
    return 0;
}

void simple_exit(void){
    printk(KERN_INFO "Removing Module\n Algo mas");
}

module_init(simple_init);
module_exit(simple_exit);
MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("ejercicioNo2")
MODULE_AUTHOR("Jose Block")
```

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

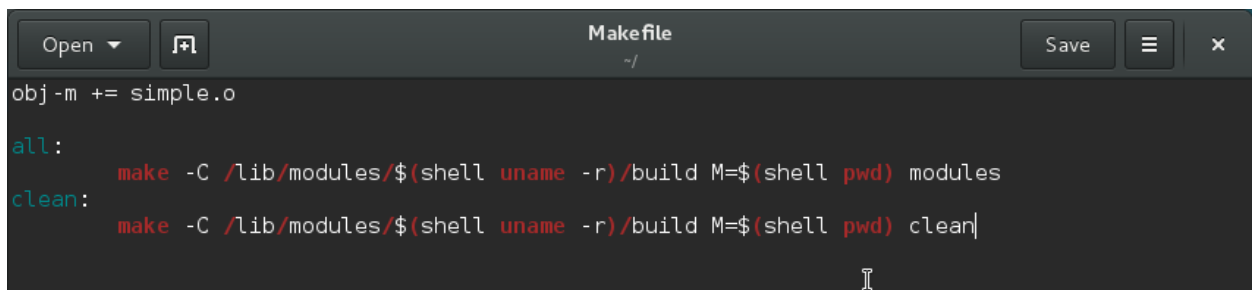
Cuando no hay parámetros, quiere decir que toma una cierta cantidad de parámetros sin especificar. Por otro lado, cuando se tiene void como parámetro, quiere decir que no recibe ningún parámetro.

¿Qué diferencia hay entre printk y printf?

La diferencia es el nivel de “log” al que puede acceder, printf es para un acceso general y printk para un acceso especial, a nivel kernel.

¿Qué es y para qué sirve KERN_INFO?

Es el nivel de log al que se hace del kernel. Usa pr_fmt() para generar un formato de “string”. Continúa un mensaje de registro anterior en la misma línea.

A screenshot of a text editor window titled "Makefile" with a dark theme. The window has a menu bar with "Open", "Save", and a close button. The content of the Makefile is as follows:

```
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
```

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

Es la parte principal del kbuild Makefile. Se encarga de definir a todos los archivos que se necesitan, ya sea una opción especial de compilación o cualquier subdirectorio al que se ingrese recursivamente. El segmento de obj-m especifica archivos que se compilan como módulos del kernel cargables.

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

La parte “all:” se refiere a todo lo que el make debe hacer para tener una compilación completa, y el “clean:” se encarga de lo opuesto.

¿Qué hace la opción -C en este Makefile?

Se traslada a una dirección específica antes de leer los archivos MAKE o incluso puede hacer otras operaciones.

¿Qué hace la opción M en este Makefile?

Permite asignar un directorio al que se regresa después de procesar un el Makefile.

```

oscreader@OSC:~$ make
make -C /lib/modules/3.16.0-4-686-pae/build M=/home/oscreader 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/simple.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/oscreader/simple.mod.o
  LD [M]  /home/oscreader/simple.ko
make[1]: Leaving directory '/usr/src/linux-headers-3.16.0-4-686-pae'
oscreader@OSC:~$ █

```

```

[ 41.883959] cfg80211: DFS Master region: unset
[ 41.883960] cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp), (dfs_cac_time)
[ 41.883963] cfg80211: (2402000 KHz - 2472000 KHz @ 40000 KHz), (N/A, 2000 mBm), (N/A)
[ 41.883965] cfg80211: (2457000 KHz - 2482000 KHz @ 40000 KHz), (N/A, 2000 mBm), (N/A)
[ 41.883967] cfg80211: (2474000 KHz - 2494000 KHz @ 20000 KHz), (N/A, 2000 mBm), (N/A)
[ 41.883970] cfg80211: (5170000 KHz - 5250000 KHz @ 80000 KHz, 160000 KHz AU TO), (N/A, 2000 mBm), (N/A)
[ 41.883972] cfg80211: (5250000 KHz - 5330000 KHz @ 80000 KHz, 160000 KHz AU TO), (N/A, 2000 mBm), (0 s)
[ 41.883975] cfg80211: (5490000 KHz - 5730000 KHz @ 160000 KHz), (N/A, 2000 mBm), (0 s)
[ 41.883977] cfg80211: (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A, 2000 mBm), (N/A)
[ 41.883979] cfg80211: (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 0 mBm), (N/A)
[ 42.441187] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 42.443601] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 42.443620] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
oscreader@OSC:~$ █

```

¿Para qué sirve dmesg?

Sirve para examinar o controlar el “kernel ring buffer”. La acción predeterminada es mostrar todos los mensajes del “kernel ring buffer”.

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

Se le pasa a module_init, y es la encargada de cargar el módulo.

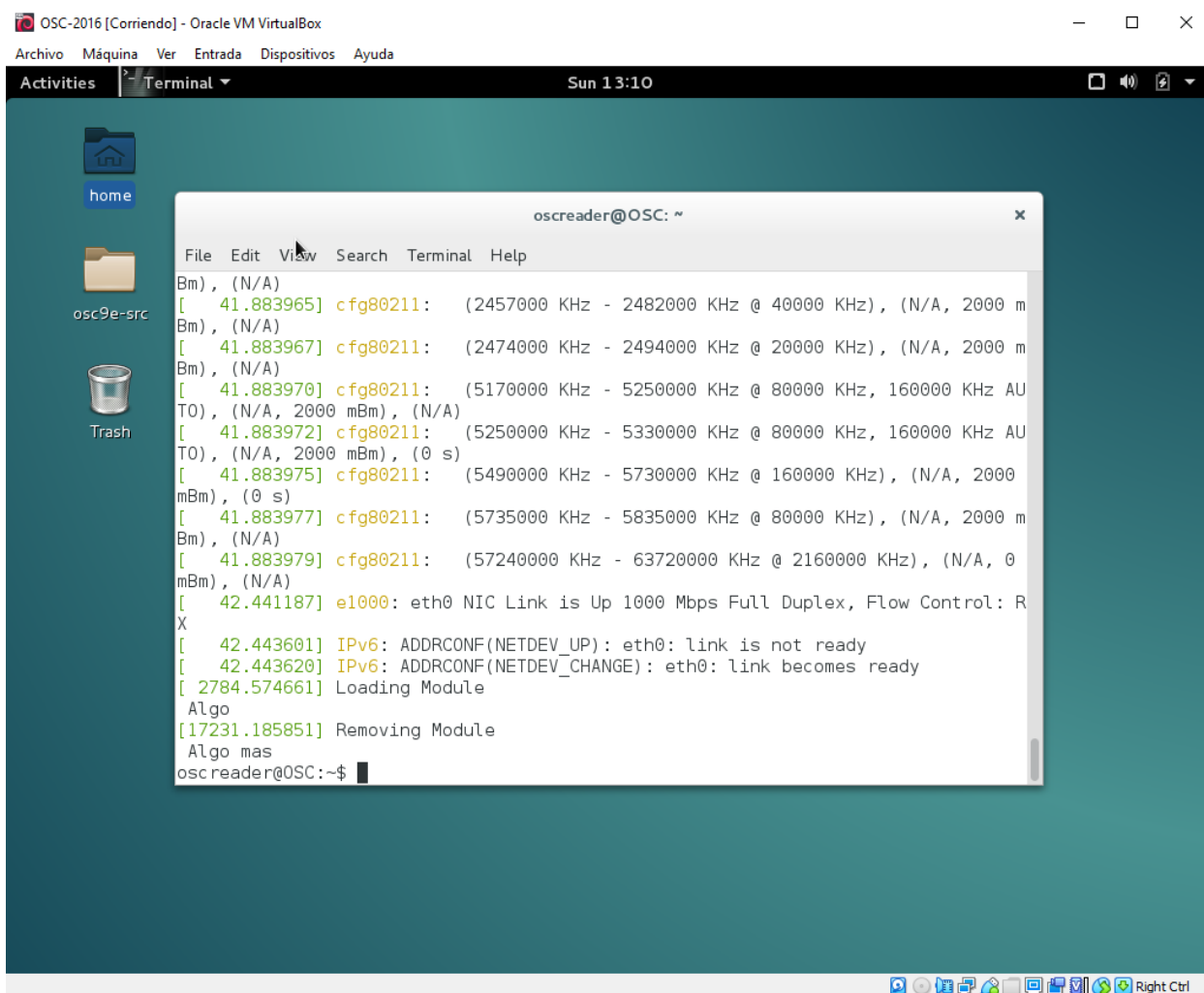
```
oscreader@OSC: ~  
File Edit View Search Terminal Help  
Bm), (N/A)  
[ 41.883965] cfg80211: (2457000 KHz - 2482000 KHz @ 40000 KHz), (N/A, 2000 m  
Bm), (N/A)  
[ 41.883967] cfg80211: (2474000 KHz - 2494000 KHz @ 20000 KHz), (N/A, 2000 m  
Bm), (N/A)  
[ 41.883970] cfg80211: (5170000 KHz - 5250000 KHz @ 80000 KHz, 160000 KHz AU  
TO), (N/A, 2000 mBm), (N/A)  
[ 41.883972] cfg80211: (5250000 KHz - 5330000 KHz @ 80000 KHz, 160000 KHz AU  
TO), (N/A, 2000 mBm), (0 s)  
[ 41.883975] cfg80211: (5490000 KHz - 5730000 KHz @ 160000 KHz), (N/A, 2000  
mBm), (0 s)  
[ 41.883977] cfg80211: (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A, 2000 m  
Bm), (N/A)  
[ 41.883979] cfg80211: (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 0  
mBm), (N/A)  
[ 42.441187] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: R  
X  
[ 42.443601] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready  
[ 42.443620] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready  
[ 2784.574661] Loading Module  
Algo  
[17231.185851] Removing Module  
Algo mas
```

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

Función que se le asigna a `module_exit`, es la encargada de eliminar el módulo.

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

Otorga un acceso privilegiado a las capacidades del hardware.



```
oscreader@OSC:~$ sudo apt-get --purge install lilo grub-legacy-
[sudo] password for oscreader:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package 'grub-legacy' is not installed, so not removed
The following NEW packages will be installed:
  lilo
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 275 kB of archives.
After this operation, 613 kB of additional disk space will be used.
Get:1 http://ftp.us.debian.org/debian/ jessie/main lilo i386 1:24.1-1 [275 kB]
Fetched 275 kB in 0s (335 kB/s)
Preconfiguring packages ...
Selecting previously unselected package lilo.
(Reading database ... 154246 files and directories currently installed.)
Preparing to unpack .../lilo_1%3a24.1-1_i386.deb ...
Unpacking lilo (1:24.1-1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up lilo (1:24.1-1) ...
oscreader@OSC:~$
```



```

oscreader@OSC:~$ cd /dev/disk/by-id
oscreader@OSC:/dev/disk/by-id$ ls -Al
total 0
lrwxrwxrwx 1 root root 9 Mar 28 04:48 ata-VBOX_CD-ROM_VB2-01700376 -> ../../sr0
lrwxrwxrwx 1 root root 9 Mar 28 04:48 ata-VBOX_HARDDISK_VB03cb385e-45c0d9b2 ->
../../sda
lrwxrwxrwx 1 root root 10 Mar 28 04:48 ata-VBOX_HARDDISK_VB03cb385e-45c0d9b2-par
t1 -> ../../sda1
lrwxrwxrwx 1 root root 10 Mar 28 04:48 ata-VBOX_HARDDISK_VB03cb385e-45c0d9b2-par
t2 -> ../../sda2
lrwxrwxrwx 1 root root 10 Mar 28 04:48 ata-VBOX_HARDDISK_VB03cb385e-45c0d9b2-par
t5 -> ../../sda5
oscreader@OSC:/dev/disk/by-id$

```

```

# /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/sda1 during installation
UUID=5f2e2232-4e47-4fe8-ae94-45ea749a5c92 /          ext4      errors=remount-ro 0      1
# swap was on /dev/sda5 during installation
UUID=ce96e707-6ab8-4556-aafc-5799e9a86292 none        swap      sw          0      0
/dev/sr0      /media/cdrom0  udf,iso9660 user,noauto 0        0

```

¿Qué es y para qué sirve el archivo fstab?

Es parte de la configuración de un sistema, cuenta con la lista de discos y particiones disponibles, indica como montar cada dispositivo y qué configuración usar.

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

Tiene todos los archivos de configuración de partes del SO y otros. En Windows es la carpeta “system32” en “drivers”.

¿Qué se almacena en /devy en /dev/disk?

En dev está todo tipo dispositivos de almacenamiento, hasta USB. En dev/disk hay información de particiones de almacenamiento del sistema. Contienen también la ubicación de los punteros de los drivers.

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

Lo que pasa es que se está dando el “carácter special file”. Esto es el archivo que representa un dispositivo que en este caso es el disco.

¿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 Rercord(VBR), y su relación con UEFI.

- Un block device es un archivo que hace referencia a un dispositivo con datos que se pueden “read” o “write” en el dispositivo en bloques. Generalmente representan almacenamiento masivo como una partición de disco.
- sdxN forma parte de los “block special files” en donde x es un dispositivo físico y N una partición en ese dispositivo.
- EL MBR es un tipo de tabla es la que se comenzó a emplear hacia el año 1983. Es un tipo especial de sector de arranque que se encuentra al comienzo de los dispositivos de almacenamiento masivo de computadoras particionadas, como discos fijos o unidades extraíbles, destinados a su uso con sistemas compatibles con IBM PC y más.
- El VBR es un tipo de sector de arranque introducido por IBM Personal Computer. Puede encontrarse en un dispositivo de almacenamiento de datos particionado, como un disco duro, o un dispositivo no particionado, como un disquete, y contiene código de máquina para programas de arranque almacenados en otras partes del dispositivo. En los dispositivos de almacenamiento sin particiones, es el primer sector del dispositivo.

¿Qué es hacer chain loading?

Remplaza el programa de ejecución que estaba predeterminado con un nuevo programa que usa áreas de información comunes para pasar la información desde el programa que se tenía al nuevo.

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

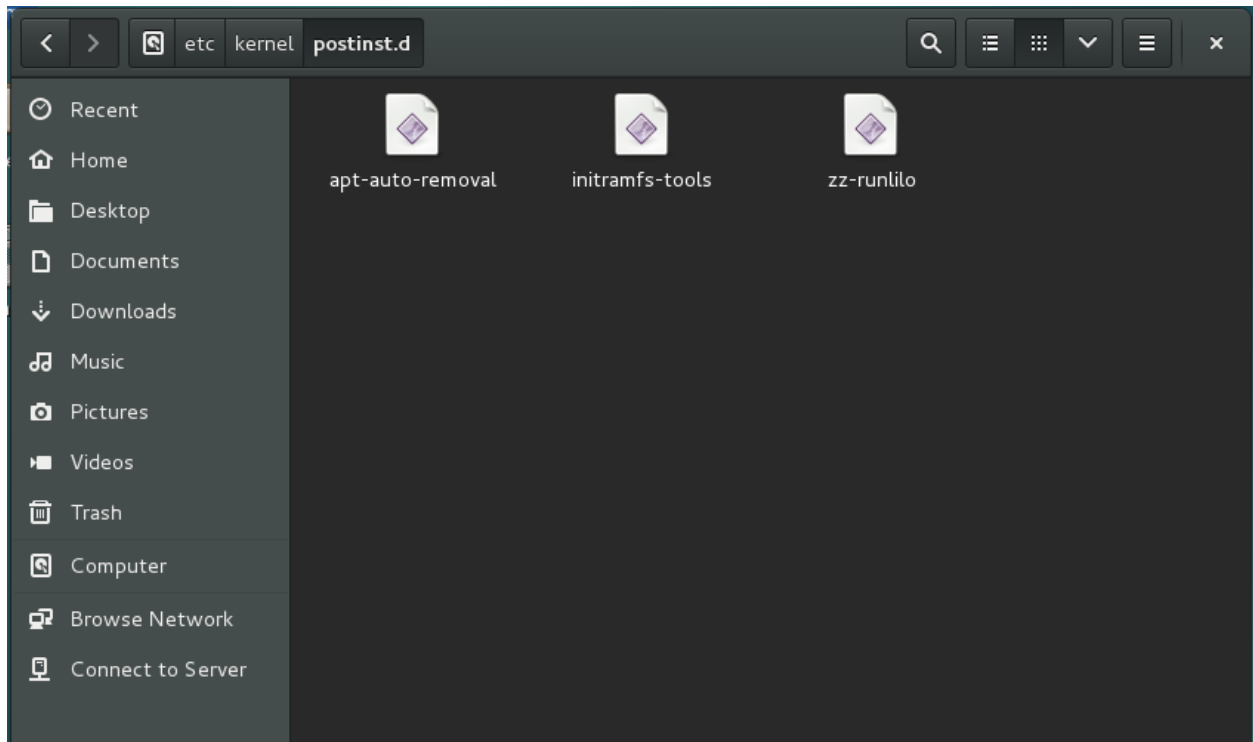
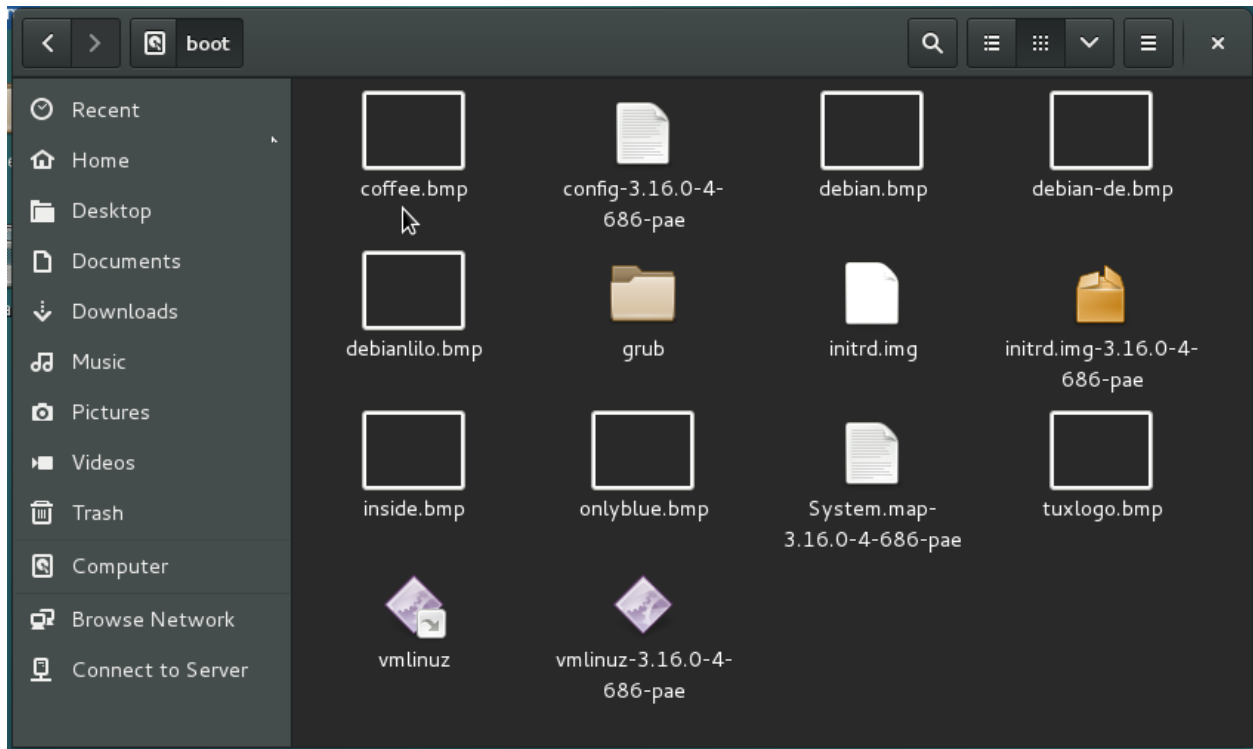
Indica las particiones de memoria del disco en donde se monta el sistema operativo. En otros casos no se hacen partición, ya que tienen dispositivos separados para el SO.

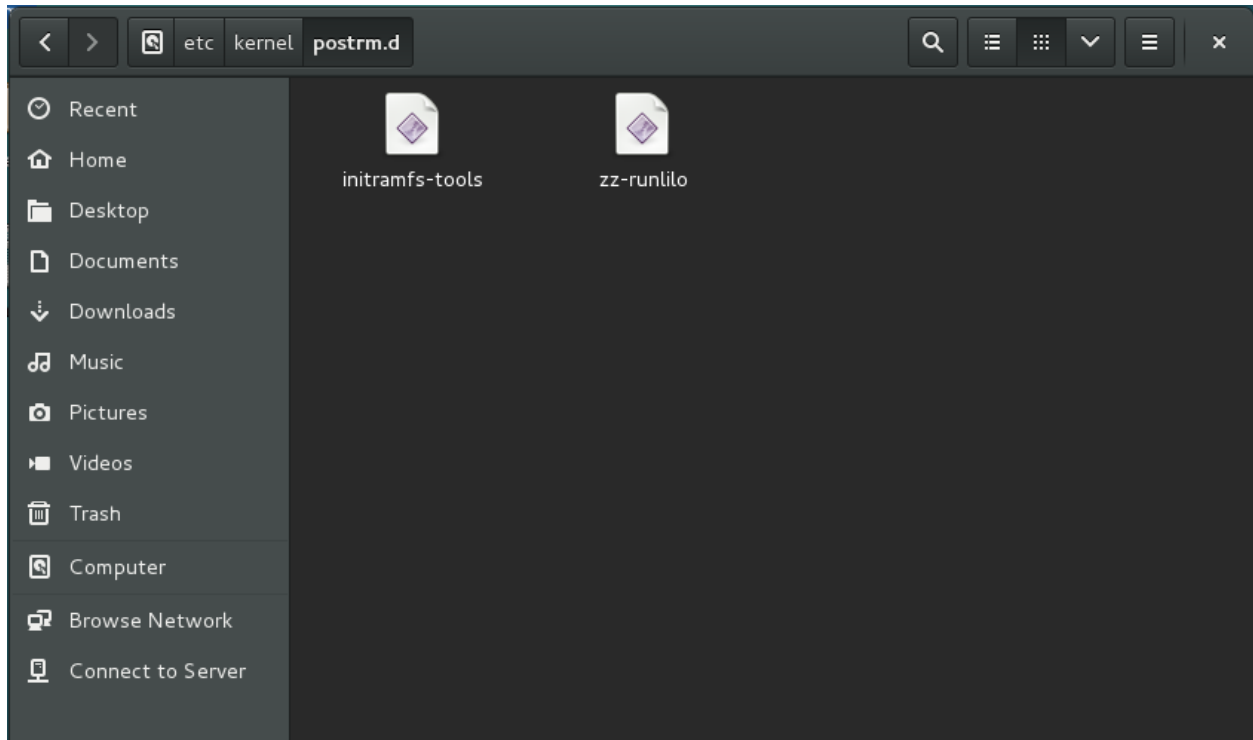
¿Qué es vmlinuz?

Es el propio kernel, o sea que este es el archivo que se carga al arrancar el servidor, que lleve la z en vez de vmlinux, quiere decir que es de forma comprimida.

```
oscreader@OSC:/etc/kernel$ ls
postinst.d  postrm.d
oscreader@OSC:/etc/kernel$
```

```
oscreader@OSC:~$ 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:
Unrecognized token "/dev/disk/by-id/../../sda" at or above line 2 in file '/etc
/lilo.conf'
run-parts: /etc/kernel/postinst.d/zz-runlilo exited with return code 1
Failed to process /etc/kernel/postinst.d at /var/lib/dpkg/info/linux-image-3.16.
0-4-686-pae.postinst line 634.
oscreader@OSC:~$
```





Mencione tres diferencias funcionales entre GRUB y LILO

En LILO no hay una interfaz de comando interactiva, no lee particiones ext2 y almacena la localización del kernel y el caso en el que otro sistema operativo se deba cargar en el MBR, o sea que LILO se debe de reinstalar.

Referencias:

<https://www.kernel.org/doc/html/latest/core-api/printk-basics.html>

<https://man7.org/linux/man-pages/man1/dmesg.1.html>

<https://www.quora.com/What-is-probing-in-data-structure>

<https://searchsecurity.techtarget.com/definition/probe>

<https://sourceware.org/systemtap/>

<https://es.wikipedia.org/wiki/Fstab>

<https://www.enmimaquinafunciona.com/pregunta/2866/que-es-el-vmlinuz-y-por-que-no-me-importa>

<https://hardzone.es/2018/09/29/mbr-vs-gpt-mejor-disco-duro-ssd/>

https://en.wikipedia.org/wiki/Master_boot_record