



---

# P03- GESTIÓN DEL KERNEL DE LINUX MINT

---

ANA OROZCO ASENSIO



2ºASIR  
I.E.S. ANTONIO MACHADO  
ASO

## Contenido

Contenido.....	2
Introducción.....	3
Gestión de los módulos de Kernel.....	3
Indica la versión del Kernel, tanto desde el entorno gráfico, como el de texto....	3
Lista los módulos que se están ejecutando actualmente. ¿Qué dos comandos tenemos para ello? .....	4
Indica, ejecutando una sola sentencia, por un lado, el número de módulos que está ejecutando el Kernel actualmente y por otro, el que ocupa mayor espacio.	5
Indica las dependencias y el chero del módulo snd. ¿Cómo lo haríamos, por un lado, mostrando toda la información del módulo y, por otro, mostrando únicamente lo que me piden? .....	6
Descargar y recargar el módulo de impresión. Comprobar en todo momento su estado (es decir, verificar que está presente, que se ha descargado y que se ha vuelto a cargar).....	6
Obtener toda la información del módulo correspondiente al driver de la tarjeta de red e interpretarla.....	7
Obtener e interpretar el listado de dispositivos pci del sistema. ....	9
Obtener e interpretar el listado de dispositivos usb del sistema (hacerlo teniendo, al menos, un pendrive conectado al mismo).....	10
Instalación de nuevo Kernel y añadirlo al Grub.....	11
Incidencias .....	15
Parte 1 (USB) .....	15
Parte 2 (comando make) .....	15
Valoración.....	16

# Introducción.

En esta práctica aprenderemos a gestionar el Kernel de Linux y a realizar instalaciones mediante compilación de ficheros fuente.

Para ello vamos a necesitar una máquina con Linux Mint 20.2-Mate, la iso se nos proporciona por lo que solo será necesario la creación de la máquina, a la cual he puesto 7GB de RAM, y 130 GB de disco duro.

## Gestión de los módulos de Kernel.

Indica la versión del Kernel, tanto desde el entorno gráfico, como el de texto.

Para ello usaremos este comando:

```
ana@ana-VirtualBox:~$ uname -r  
5.4.0-200-generic
```

Para verlo en modo gráfico debemos ir al menú y buscaremos el Monitor del sistema:



Abiremos el programa y la información del Kernel la tendremos en Sistema:

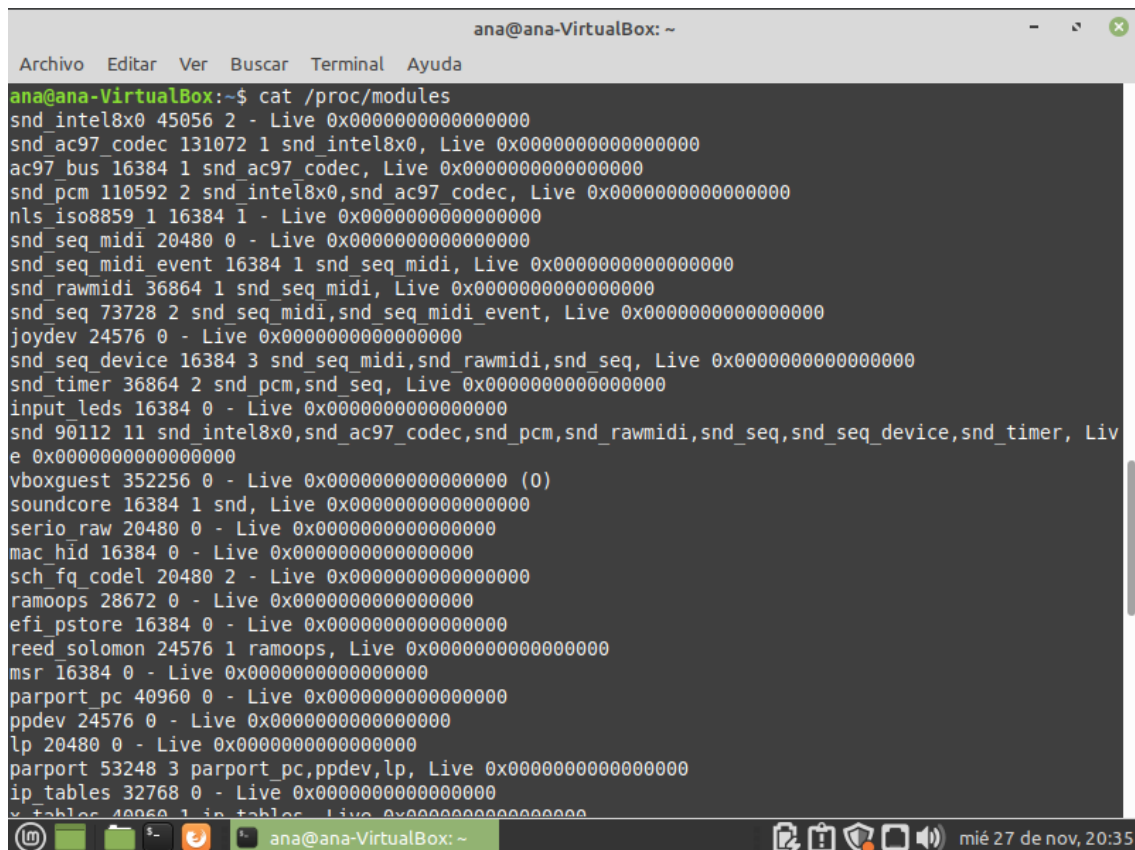


Lista los módulos que se están ejecutando actualmente. ¿Qué dos comandos tenemos para ello?

Hay dos comandos que dan esa información aunque de manera distinta, el primer comando es `lsmod` que nos muestra la información de manera legible y clara:

The image shows a terminal window titled 'ana@ana-VirtualBox: ~' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal displays the output of the `lsmod` command, showing a list of loaded kernel modules and their dependencies. The output is as follows:  
ana@ana-VirtualBox:~\$ uname -r  
5.4.0-200-generic  
ana@ana-VirtualBox:~\$ lsmod  
Module Size Used by  
snd\_intel8x0 45056 2  
snd\_ac97\_codec 131072 1 snd\_intel8x0  
ac97\_bus 16384 1 snd\_ac97\_codec  
snd\_pcm 110592 2 snd\_intel8x0,snd\_ac97\_codec  
snd\_isa8859\_1 16384 1  
snd\_seq\_midi 20480 0  
snd\_seq\_midi\_event 16384 1 snd\_seq\_midi  
snd\_rawmidi 36864 1 snd\_seq\_midi  
snd\_seq 73728 2 snd\_seq\_midi,snd\_seq\_midi\_event  
joydev 24576 0  
snd\_seq\_device 16384 3 snd\_seq,snd\_seq\_midi,snd\_rawmidi  
snd\_timer 36864 2 snd\_seq,snd\_pcm  
input\_leds 16384 0  
snd 90112 11 snd\_seq,snd\_seq\_device,snd\_intel8x0,snd\_timer,snd\_ac97\_codec,snd\_pcm,snd\_rawmidi  
vboxquest 352256 0  
soundcore 16384 1 snd  
serio\_raw 20480 0  
mac\_hid 16384 0  
sch\_fq\_codel 20480 2  
ramoops 28672 0  
efi\_pstore 16384 0  
reed\_solomon 24576 1 ramoops  
nsr 16384 0  
parport\_pc 40960 0  
ppdev 24576 0  
n 20480 0

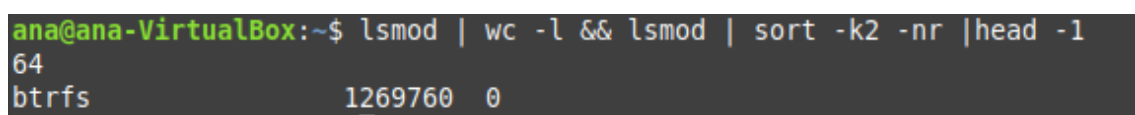
Pero también tenemos esta forma de mirarlo que es usando el comando cat que nos muestra el fichero del que pongamos la ruta, esta forma nos muestra la información más detallada por lo que puede ser mas interesante para análisis mas profundos y más avanzados.



```
ana@ana-VirtualBox:~$ cat /proc/modules
snd_intel8x0 45056 2 - Live 0x0000000000000000
snd_ac97_codec 131072 1 snd_intel8x0, Live 0x0000000000000000
ac97_bus 16384 1 snd_ac97_codec, Live 0x0000000000000000
snd_pcm 110592 2 snd_intel8x0,snd_ac97_codec, Live 0x0000000000000000
nls_iso8859_1 16384 1 - Live 0x0000000000000000
snd_seq_midi 20480 0 - Live 0x0000000000000000
snd_seq_midi_event 16384 1 snd_seq_midi, Live 0x0000000000000000
snd_rawmidi 36864 1 snd_seq_midi, Live 0x0000000000000000
snd_seq 73728 2 snd_seq_midi,snd_seq_midi_event, Live 0x0000000000000000
joydev 24576 0 - Live 0x0000000000000000
snd_seq_device 16384 3 snd_seq_midi,snd_rawmidi,snd_seq, Live 0x0000000000000000
snd_timer 36864 2 snd_pcm,snd_seq, Live 0x0000000000000000
input_leds 16384 0 - Live 0x0000000000000000
snd 90112 11 snd_intel8x0,snd_ac97_codec,snd_pcm,snd_rawmidi,snd_seq,snd_seq_device,snd_timer, Live 0x0000000000000000
vboxguest 352256 0 - Live 0x0000000000000000 (0)
soundcore 16384 1 snd, Live 0x0000000000000000
serio_raw 20480 0 - Live 0x0000000000000000
mac_hid 16384 0 - Live 0x0000000000000000
sch_fq_codel 20480 2 - Live 0x0000000000000000
ramoops 28672 0 - Live 0x0000000000000000
efi_pstore 16384 0 - Live 0x0000000000000000
reed_solomon 24576 1 ramoops, Live 0x0000000000000000
msr 16384 0 - Live 0x0000000000000000
parport_pc 40960 0 - Live 0x0000000000000000
ppdev 24576 0 - Live 0x0000000000000000
lp 20480 0 - Live 0x0000000000000000
parport 53248 3 parport_pc,ppdev,lp, Live 0x0000000000000000
ip_tables 32768 0 - Live 0x0000000000000000
x_tables 40960 1 ip_tables, Live 0x0000000000000000
```

Indica, ejecutando una sola sentencia, por un lado, el número de módulos que está ejecutando el Kernel actualmente y por otro, el que ocupa mayor espacio.

La sentencia completa sería “lsmod | wc -l && lsmod | sort -k2 -nr | head -1”, vamos a usar el comando “lsmod” que ya hemos visto lo que hace, también usaremos el comando wc -l que cuenta las líneas q devuelve el comando lsmod, con srt -k2 -nr ordenamos los resultados por tamaño y en orden descendente, por ultimo, con head -1 devuelve solo el primer módulo.



```
ana@ana-VirtualBox:~$ lsmod | wc -l && lsmod | sort -k2 -nr | head -1
64
btrfs 1269760 0
```

Indica las dependencias y el chero del módulo snd. ¿Cómo lo haríamos, por un lado, mostrando toda la información del módulo y, por otro, mostrando únicamente lo que me piden?

Mostrando toda la información del módulo usamos este comando:

```
ana@ana-VirtualBox:~$ modinfo snd
filename:      /lib/modules/5.4.0-200-generic/kernel/sound/core/snd.ko
alias:         char-major-116-*
license:       GPL
description:    Advanced Linux Sound Architecture driver for soundcards.
author:         Jaroslav Kysela <perex@perex.cz>
srcversion:     A196C05FAA92EECB8652D81
depends:         soundcore
retpoline:     Y
intree:        Y
name:          snd
vermagic:      5.4.0-200-generic SMP mod_unload modversions
sig_id:         PKCS#7
signer:        Build time autogenerated kernel key
sig_key:        62:68:82:25:61:A3:0F:C7:F4:C0:34:A2:36:AC:FE:B9:51:E6:DF:D3
sig_hashalgo:  sha512
```

Si solo queremos ver el nombre y las dependencias lo haremos de esta forma:

```
ana@ana-VirtualBox:~$ modinfo snd -n && modinfo snd | grep depends
/lib/modules/5.4.0-200-generic/kernel/sound/core/snd.ko
depends:         soundcore
ana@ana-VirtualBox:~$
```

Descargar y recargar el módulo de impresión. Comprobar en todo momento su estado (es decir, verificar que está presente, que se ha descargado y que se ha vuelto a cargar).

Para esto vamos a volver a usar lsmod, y vamos a buscar el módulo en concreto que es “lp”.

```
ana@ana-VirtualBo
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
ramoops          28672  0
efi_pstore       16384  0
reed_solomon     24576  1 ramoops
msr               16384  0
parport_pc       40960  0
ppdev            24576  0
lp              20480  0
parport          53248  3 parport_pc,lp,ppdev
ip_tables        32768  0
x_tables         40960  1 ip_tables
autofs4          45056  2
btrfs            1269760 0
xor              24576  1 btrfs
zstd_compress    167936  1 btrfs
raid6_pq         114688  1 btrfs
```

Vamos descargarlo con el comando sudo rmmod lp:

```
ana@ana-VirtualBox:~$ sudo rmmod lp
[sudo] contraseña para ana:
ana@ana-VirtualBox:~$
```

Vamos a comprobar que no está:

```
ana@ana-VirtualBox:~$ lsmod | grep lp
glue_helper      16384  1 aesni_intel
drm_kms_helper   184320  1 vmwgfx
syscopyarea      16384  1 drm_kms_helper
sysfillrect      16384  1 drm_kms_helper
sysimgblt        16384  1 drm_kms_helper
fb_sys_fops      16384  1 drm_kms_helper
drm              495616  5 vmwgfx,drm_kms_helper,ttm
ana@ana-VirtualBox:~$
```

Y por último vamos a recargarlo y comprobaré que se ha recargado:

```
ana@ana-VirtualBox:~$ sudo modprobe lp
ana@ana-VirtualBox:~$ lsmod | grep lp
lp              20480  0
parport         53248  3 parport_pc,lp,ppdev
glue_helper     16384  1 aesni_intel
drm_kms_helper  184320  1 vmwgfx
syscopyarea     16384  1 drm_kms_helper
sysfillrect     16384  1 drm_kms_helper
sysimgblt       16384  1 drm_kms_helper
fb_sys_fops     16384  1 drm_kms_helper
drm             495616  5 vmwgfx,drm_kms_helper,ttm
ana@ana-VirtualBox:~$
```

Obtener toda la información del módulo correspondiente al driver de la tarjeta de red e interpretarla.

Vamos a buscar el nombre del módulo con lspci -k, de esta manera nos saldrá toda la información de los dispositivos que tenemos conectados en el sistema:

```
ana@ana-VirtualBox:~$ lspci -k
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
        Kernel driver in use: ata_piix
        Kernel modules: pata_acpi
00:02.0 VGA compatible controller: VMware SVGA II Adapter
        Subsystem: VMware SVGA II Adapter
        Kernel driver in use: vmwgfx
        Kernel modules: vmwgfx
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller
        Subsystem: Intel Corporation PRO/1000 MT Desktop Adapter
        Kernel driver in use: e1000
        Kernel modules: e1000
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Services
        Kernel driver in use: vboxguest
```

El módulo de la tarjeta de red es el e1000, por lo que vamos a usar el comando modinfo que nos enseñará información detallada sobre el módulo del kernel de la tarjeta de red:

```
ana@ana-VirtualBox:~$ modinfo e1000
filename:       /lib/modules/5.4.0-200-generic/kernel/drivers/net/ethernet/intel/e1000/e1000.ko
version:       7.3.21-k8-NAPI
license:       GPL v2
description:    Intel(R) PRO/1000 Network Driver
author:        Intel Corporation, <linux.nics@intel.com>
srcversion:    FCB88217EDA1AA26ACC1A02
alias:         pci:v00008086d00002E6Esv*sd*bc*sc*i*
alias:         pci:v00008086d000010B5sv*sd*bc*sc*i*
alias:         pci:v00008086d00001099sv*sd*bc*sc*i*
alias:         pci:v00008086d0000108Asv*sd*bc*sc*i*
alias:         pci:v00008086d0000107Csv*sd*bc*sc*i*
alias:         pci:v00008086d0000107Bsv*sd*bc*sc*i*
alias:         pci:v00008086d0000107Asv*sd*bc*sc*i*
alias:         pci:v00008086d00001079sv*sd*bc*sc*i*
alias:         pci:v00008086d00001078sv*sd*bc*sc*i*
alias:         pci:v00008086d00001077sv*sd*bc*sc*i*
alias:         pci:v00008086d00001076sv*sd*bc*sc*i*
alias:         pci:v00008086d00001075sv*sd*bc*sc*i*
alias:         pci:v00008086d00001028sv*sd*bc*sc*i*
alias:         pci:v00008086d00001027sv*sd*bc*sc*i*
alias:         pci:v00008086d00001026sv*sd*bc*sc*i*
alias:         pci:v00008086d0000101Esv*sd*bc*sc*i*
alias:         pci:v00008086d0000101Dsv*sd*bc*sc*i*
alias:         pci:v00008086d0000101Asv*sd*bc*sc*i*
alias:         pci:v00008086d00001019sv*sd*bc*sc*i*
alias:         pci:v00008086d00001018sv*sd*bc*sc*i*
alias:         pci:v00008086d00001017sv*sd*bc*sc*i*
alias:         pci:v00008086d00001016sv*sd*bc*sc*i*
```

La información que nos encontramos significa:

**Filename:** aquí nos aparece la ruta junto con el nombre del archivo y su extensión.

**Version:** versión del módulo.

**License:** la licencia bajo la cual se distribuye, esta es GPL v2 que es General Public License.

**Description:** Una descripción, en este caso nos dice que es el driver de red de Intel.

**Author:** El autor, en este caso Intel.

**Srcversion:** Es el identificador único de la versión del código fuente con el que se compiló el módulo del kernel.

**Alias:** en este caso aparece en varias ocasiones y son los alias del módulo.

En algunos casos pueden aparecer más tipo de información, pero en esta ocasión esta es la información que ha proporcionado.

## Obtener e interpretar el listado de dispositivos pci del sistema.

Volvemos a este comando, esta vez vamos a interpretar los datos que nos proporcionan:

```
ana@ana-VirtualBox:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: VMware SVGA II Adapter
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0b.0 USB controller: Intel Corporation 82801FB/FBM/FW/FW/FW (ICH6 Family) USB2 EHCI Controller
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)
```

**Puente de Host (Host Bridge):** Es como el "intermediario" entre el procesador (la parte que hace los cálculos) y el resto del sistema. Sin él, el procesador no podría comunicarse con el resto de las partes.

**Puente ISA (ISA Bridge):** Conecta componentes más viejos (como tarjetas de sonido antiguas o módems) con el sistema moderno. Es como un adaptador.

**Controlador IDE (IDE Controller):** Se encarga de que el sistema pueda comunicarse con los discos duros antiguos y las unidades ópticas (como CD o DVD) a través de un tipo de conexión llamada IDE. Es como el "gestor de archivos" para estos dispositivos más viejos.

**Controlador VGA (VGA Controller):** Se ocupa de enviar la señal gráfica del ordenador al monitor, es decir, se encarga de todo lo que vemos en la pantalla.

**Controlador Ethernet (Ethernet Controller):** Es el responsable de conectar tu ordenador a la red, sin este driver por mucho que tuviéramos conector, cable y red, sería imposible conectarnos.

**Periférico del Sistema (System Peripheral):** Se encarga de tareas internas dentro del sistema que no son tan visibles para el usuario, como la gestión de algunos controladores de bajo nivel. Son como "asistentes" que ayudan a que todo funcione sin que los notemos.

**Controlador de Audio Multimedia (Multimedia Audio Controller):** Se encarga de gestionar el sonido del sistema, tanto la salida como la entrada.

**Controlador USB - Apple (USB Controller - Apple):** Este es el que maneja todos los puertos y dispositivos USB en ordenadores Apple.

**Puente PCI (PCI Bridge):** Es el "puente" que conecta diferentes partes del sistema que usan el bus PCI (una vía de comunicación para otros dispositivos). Permite que esas partes se comuniquen entre sí.

**Controlador USB - Intel (USB Controller - Intel):** Similar al de Apple, pero este es específico para sistemas con procesadores Intel. Gestiona las conexiones USB, como los puertos donde puedes conectar dispositivos como teclados, ratones, o discos duros externos.

**Controlador SATA (SATA Controller):** Se encarga de la conexión de discos duros y unidades SSD a través del bus SATA. Es el que permite que tu computadora lea y escriba datos en estos dispositivos de almacenamiento más recientes.

Obtener e interpretar el listado de dispositivos usb del sistema (hacerlo teniendo, al menos, un pendrive conectado al mismo).

```
ana@ana-VirtualBox:~$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
ana@ana-VirtualBox:~$
```

Bus: El número del bus en el que está el dispositivo.

Device: El número identificativo del dispositivo.

ID: Los identificadores del fabricante y del producto.

\*Si tuviéramos un USB debería aparecer así:

```
Bus 001 Device 003: ID 0951:1666 Kingston Technology DataTraveler 100 G3/G4/SE9 G2
```

# Instalación de nuevo Kernel y añadirlo al Grub.

Lo primero que haré será hacer una actualización con update.

```
ana@ana-VirtualBox:~$ sudo apt update
Obj:1 http://archive.canonical.com/ubuntu focal InRelease
Des:2 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Ign:3 http://packages.linuxmint.com uma InRelease
Obj:4 http://archive.ubuntu.com/ubuntu focal InRelease
Obj:5 http://packages.linuxmint.com uma Release
Des:6 http://archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Des:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3.304 kB]
Des:9 http://archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Des:10 http://archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [1.056 kB]
Des:11 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3.681 kB]
Des:12 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [835 kB]
Des:13 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [65,2 kB]
Des:14 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 DEP-11 Metadata [212 B]
Des:15 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [1.016 kB]
Des:16 http://archive.ubuntu.com/ubuntu focal-updates/main Translation-en [564 kB]
Des:17 http://security.ubuntu.com/ubuntu focal-security/universe i386 Packages [683 kB]
Des:18 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [160 kB]
Des:19 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [940 B]
Des:20 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [276 kB]
Des:21 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 DEP-11 Metadata [212 B]
Des:22 http://archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [810 kB]
Des:23 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1.238 kB]
Des:24 http://archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [297 kB]
Des:25 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [446 kB]
```

Vamos a descargar la versión requerida:

```
ana@ana-VirtualBox:~$ wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.11.7.tar.xz
--2024-11-27 20:56:03-- https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.11.7.tar.xz
Resolviendo cdn.kernel.org (cdn.kernel.org)... 151.101.133.176, 2a04:4e42:1f::432
Conectando con cdn.kernel.org (cdn.kernel.org)[151.101.133.176]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 146976356 (140M) [application/x-xz]
Guardando como: "linux-6.11.7.tar.xz"

linux-6.11.7.tar.xz 100%[=====] 140,17M 52,3MB/s en 2,7s
2024-11-27 20:56:07 (52,3 MB/s) - "linux-6.11.7.tar.xz" guardado [146976356/146976356]
```

Y vamos a descomprimirlo

```
ana@ana-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda

ana@ana-VirtualBox:~$ tar -xvf linux-6.11.7.tar.xz
linux-6.11.7/
linux-6.11.7/.clang-format
linux-6.11.7/.coocciconfig
linux-6.11.7/.editorconfig
linux-6.11.7/.get_maintainer.ignore
linux-6.11.7/.gitattributes
linux-6.11.7/.gitignore
linux-6.11.7/.mailmap
linux-6.11.7/.rustfmt.toml
linux-6.11.7/COPYING
linux-6.11.7/CREDITS
linux-6.11.7/Documentation/
linux-6.11.7/Documentation/.gitignore
linux-6.11.7/Documentation/ABI/
linux-6.11.7/Documentation/ABI/README
linux-6.11.7/Documentation/ABI/obsolete/
linux-6.11.7/Documentation/ABI/obsolete/o2cb
linux-6.11.7/Documentation/ABI/obsolete/sysfs-i8k
linux-6.11.7/Documentation/ABI/obsolete/sysfs-bus-iiio
linux-6.11.7/Documentation/ABI/obsolete/sysfs-bus-usb
linux-6.11.7/Documentation/ABI/obsolete/sysfs-class-typec
linux-6.11.7/Documentation/ABI/obsolete/sysfs-cpuidle
linux-6.11.7/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-arvo
linux-6.11.7/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-isku
linux-6.11.7/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-koneplus
linux-6.11.7/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-konepure
linux-6.11.7/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-kovapplus
linux-6.11.7/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-lua
linux-6.11.7/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-pyra
```

A continuación vamos a instalar todas las herramientas para compilar e instalar el núcleo nuevo:

```
ana@ana-VirtualBox:~$ sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc flex libelf-dev bison
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota, seleccionando «libncurses-dev» en lugar de «ncurses-dev»
bc ya está en su versión más reciente (1.07.1-2build1).
bison ya está en su versión más reciente (2:3.5.1+dfsg-1).
fakeroot ya está en su versión más reciente (1.24-1).
flex ya está en su versión más reciente (2.6.4-6.2).
xz-utils ya está en su versión más reciente (5.2.4-1ubuntu1.1).
Se instalarán los siguientes paquetes adicionales:
  g++ g++-9 git-man liberror-perl libstdc++-9-dev zlib1g-dev
Paquetes sugeridos:
  g++-multilib g++-9-multilib gcc-9-doc git-daemon-run | git-daemon-sysvinit git-doc git-el
  git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn ncurses-doc libssl-doc
  libstdc++-9-doc
Se instalarán los siguientes paquetes NUEVOS:
  build-essential g++ g++-9 git git-man libelf-dev liberror-perl libncurses-dev libssl-dev
  libstdc++-9-dev zlib1g-dev
0 actualizados, 11 nuevos se instalarán, 0 para eliminar y 4 no actualizados.
Se necesita descargar 17,8 MB de archivos.
Se utilizarán 97,0 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Vamos ahora a ir al directorio en el que hemos descomprimido y vamos a copiar el archivo de configuración en otro directorio:

```
ana@ana-VirtualBox:~$ cd linux-6.11.7/
ana@ana-VirtualBox:~/linux-6.11.7$ ls
arch      CREDITS    fs          ipc          lib          mm          samples    tools
block     crypto     include     Kbuild      LICENSES     net         scripts    usr
certs     Documenta init        Kconfig     MAINTAINERS  README     security   virt
COPYING   drivers    io_uring    kernel       Makefile     rust       sound
ana@ana-VirtualBox:~/linux-6.11.7$
cp -v /boot/config-$(uname -r) .config
cp -v '/boot/config-5.4.0-200-generic' -> '.config'
ana@ana-VirtualBox:~/linux-6.11.7$
```

Vamos a poner estos dos comandos para evitar que de problemas (a algunos compañeros les ha funcionado solo hacer esto):

```
ana@ana-VirtualBox:~/linux-6.11.7$ scripts/config --disable SYSTEM_TRUSTED_KEYS
ana@ana-VirtualBox:~/linux-6.11.7$ scripts/config --disable SYSTEM_REVOCATION_KEYS
```

\*En mi caso tuve que hacer algo más, como muestro en incidencias.

Por último, vamos a compilarlo, usando make, este paso tarda mucho tiempo:

```
ana@ana-VirtualBox:~/linux-6.11.7$ make
SYNC      include/config/auto.conf.cmd
HOSTCC    scripts/selinux/genheaders/genheaders
HOSTCC    scripts/selinux/mdp/mdp
CC         scripts/mod/empty.o
MKELF     scripts/mod/elfconfig.h
HOSTCC    scripts/mod/modpost.o
CC         scripts/mod/devicetable-offsets.s
HOSTCC    scripts/mod/file2alias.o
HOSTCC    scripts/mod/sumversion.o
HOSTCC    scripts/mod/symsearch.o
HOSTLD    scripts/mod/modpost
CC         kernel/bounds.s
CC         arch/x86/kernel/asm-offsets.s
CALL      scripts/checksyscalls.sh
DESCEND   objtool
INSTALL   libsubcmd_headers
CC         init/main.o
CC         init/version.o
CC         init/do_mounts.o
CC         init/do_mounts_initrd.o
CC         init/initramfs.o
CC         init/calibrate.o
CC         init/init_task.o
AR         init/built-in.a
CC         arch/x86/coco/core.o
CC         arch/x86/coco/sev/core.o
AR         arch/x86/coco/sev/built-in.a
AR         arch/x86/coco/built-in.a
AS         arch/x86/entry/entry.o
```

Una vez termine correctamente vamos a instalar los módulos necesarios:

```
ana@ana-VirtualBox:~/linux-6.11.7$ sudo make modules_install
[sudo] contraseña para ana:
SYMLINK /lib/modules/6.11.7/build
INSTALL /lib/modules/6.11.7/modules.order
INSTALL /lib/modules/6.11.7/modules.builtin
INSTALL /lib/modules/6.11.7/modules.builtin.modinfo
INSTALL /lib/modules/6.11.7/kernel/arch/x86/events/intel/intel-cstate.ko
SIGN     /lib/modules/6.11.7/kernel/arch/x86/events/intel/intel-cstate.ko
INSTALL /lib/modules/6.11.7/kernel/arch/x86/events/rapl.ko
SIGN     /lib/modules/6.11.7/kernel/arch/x86/events/rapl.ko
INSTALL /lib/modules/6.11.7/kernel/arch/x86/kernel/cpu/mce/mce-inject.ko
SIGN     /lib/modules/6.11.7/kernel/arch/x86/kernel/cpu/mce/mce-inject.ko
INSTALL /lib/modules/6.11.7/kernel/arch/x86/kernel/msr.ko
SIGN     /lib/modules/6.11.7/kernel/arch/x86/kernel/msr.ko
INSTALL /lib/modules/6.11.7/kernel/arch/x86/kernel/cpuid.ko
SIGN     /lib/modules/6.11.7/kernel/arch/x86/kernel/cpuid.ko
INSTALL /lib/modules/6.11.7/kernel/arch/x86/crypto/twofish-x86_64.ko
SIGN     /lib/modules/6.11.7/kernel/arch/x86/crypto/twofish-x86_64.ko
INSTALL /lib/modules/6.11.7/kernel/arch/x86/crypto/twofish-x86_64-3way.ko
SIGN     /lib/modules/6.11.7/kernel/arch/x86/crypto/twofish-x86_64-3way.ko
INSTALL /lib/modules/6.11.7/kernel/arch/x86/crypto/twofish-avx-x86_64.ko
SIGN     /lib/modules/6.11.7/kernel/arch/x86/crypto/twofish-avx-x86_64.ko
INSTALL /lib/modules/6.11.7/kernel/arch/x86/crypto/serpent-sse2-x86_64.ko
SIGN     /lib/modules/6.11.7/kernel/arch/x86/crypto/serpent-sse2-x86_64.ko
```

Por último, vamos a realizar la instalación del Kernel que hemos compilado:

```
ana@ana-VirtualBox:~/linux-6.11.7$ sudo make install
INSTALL /boot
run-parts: executing /etc/kernel/postinst.d/dkms 6.11.7 /boot/vmlinuz-6.11.7
* dkms: running auto installation service for kernel 6.11.7 [ OK ]
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 6.11.7 /boot/vmlinuz-6.11.7
update-initramfs: Generating /boot/initrd.img-6.11.7
run-parts: executing /etc/kernel/postinst.d/pm-utils 6.11.7 /boot/vmlinuz-6.11.7
run-parts: executing /etc/kernel/postinst.d/vboxadd 6.11.7 /boot/vmlinuz-6.11.7
VirtualBox Guest Additions: Building the modules for kernel 6.11.7.

VirtualBox Guest Additions: Look at /var/log/vboxadd-setup.log to find out what
went wrong
run-parts: executing /etc/kernel/postinst.d/xx-update-initrd-links 6.11.7 /boot/vmlinuz-6.11.7
I: /boot/initrd.img.old is now a symlink to initrd.img-5.4.0-200-generic
I: /boot/initrd.img is now a symlink to initrd.img-6.11.7
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 6.11.7 /boot/vmlinuz-6.11.7
Obteniendo el archivo «/etc/default/grub»
Obteniendo el archivo «/etc/default/grub.d/50_linuxmint.cfg»
Obteniendo el archivo «/etc/default/grub.d/init-select.cfg»
Generando un fichero de configuración de grub...
Encontrada imagen de linux: /boot/vmlinuz-6.11.7
Encontrada imagen de memoria inicial: /boot/initrd.img-6.11.7
Encontrada imagen de linux: /boot/vmlinuz-5.4.0-200-generic
Encontrada imagen de memoria inicial: /boot/initrd.img-5.4.0-200-generic
Encontrada imagen de linux: /boot/vmlinuz-5.4.0-74-generic
Encontrada imagen de memoria inicial: /boot/initrd.img-5.4.0-74-generic
hecho
ana@ana-VirtualBox:~/linux-6.11.7$
```

Una vez terminado correctamente vamos a actualizar y cargar el grub:

```
ana@ana-VirtualBox:~/linux-6.11.7$ sudo update-initramfs -c -k 6.11.7
update-initramfs: Generating /boot/initrd.img-6.11.7
```

```
ana@ana-VirtualBox:~/linux-6.11.7$ sudo update-grub
Obteniendo el archivo «/etc/default/grub»
Obteniendo el archivo «/etc/default/grub.d/50_linuxmint.cfg»
Obteniendo el archivo «/etc/default/grub.d/init-select.cfg»
Generando un fichero de configuración de grub...
Encontrada imagen de linux: /boot/vmlinuz-6.11.7
Encontrada imagen de memoria inicial: /boot/initrd.img-6.11.7
Encontrada imagen de linux: /boot/vmlinuz-5.4.0-200-generic
Encontrada imagen de memoria inicial: /boot/initrd.img-5.4.0-200-generic
Encontrada imagen de linux: /boot/vmlinuz-5.4.0-74-generic
Encontrada imagen de memoria inicial: /boot/initrd.img-5.4.0-74-generic
```

Reiniciaremos la máquina y vamos a comprobar que lo hemos hecho bien:

```
ana@ana-VirtualBox:~$ uname -mrs
Linux 6.11.7 x86_64
ana@ana-VirtualBox:~$
```

# Incidencias

## Parte 1 (USB)

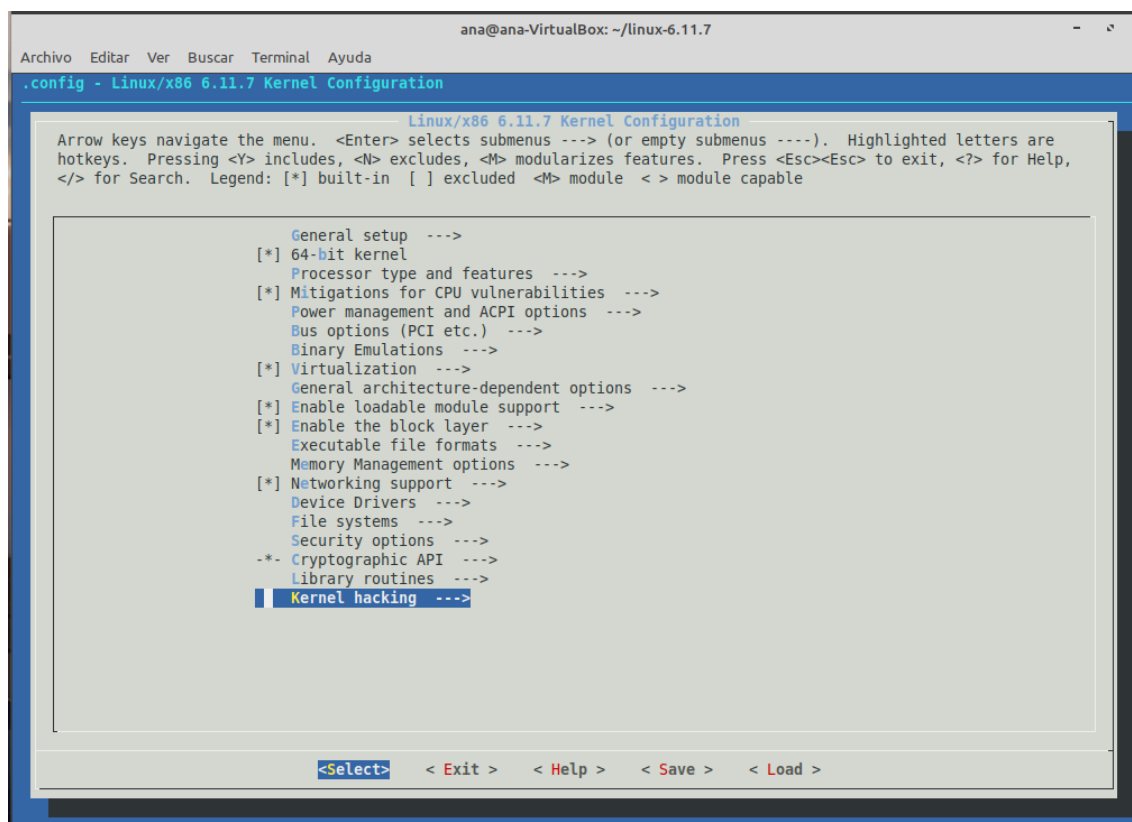
Por un momento pensé que no me aparecía el usb que tenía conectado por un error del virtualbox pero más tarde intentando buscar una solución me di cuenta de que se me había roto el usb. Por lo que pedí una captura a un compañero para poder poner como se vería.

## Parte 2 (comando make)

Aunque a mis compañeros con poner unos comandos le solucionó est problema, a mi no, por lo que tuve que hacer lo que está en los apuntes:

```
LD .tmp_vmlinux1
BTF: .tmp_vmlinux1: pahole (pahole) is not available
Failed to generate BTF for vmlinux
Try to disable CONFIG_DEBUG_INFO_BTF
make[2]: *** [scripts/Makefile.vmlinux:34: vmlinux] Error 1
make[1]: *** [/home/ana/linux-6.11.7/Makefile:1157: vmlinux] Error 2
make: *** [Makefile:224: __sub-make] Error 2
ana@ana-VirtualBox:~/linux-6.11.7$
```

Haremos make menuconfig:



Elimino la selección y guardo los cambios:

```
Legend: [?] built-in [ ] excluded <M> module < > module capable

    Debug information (Generate DWARF Version 4 debuginfo) --->
[ ] Reduce debugging information
    Compressed Debug information (Don't compress debug information) --->
[ ] Produce split debuginfo in .dwo files
[*] Generate BTF type information
[*] Provide GDB scripts for kernel debugging
(1024) Warn for stack frames larger than
[ ] Strip assembler-generated symbols during link
[ ] Generate readable assembler code
[ ] Install uapi headers to usr/include
[ ] Enable full Section mismatch analysis
[*] Make section mismatch errors non-fatal
[ ] Force all function address 64B aligned
-*- Compile the kernel with frame pointers
[*] Compile-time stack metadata validation
[ ] Generate vmlinux.map file when linking
[ ] Force weak per-cpu definitions
```

Y de esta manera al compilar ya no me dará ningún fallo cuando termine:

```
CC [M] net/nsh/nsh.mod.o
LD [M] net/nsh/nsh.ko
CC [M] net/hsr/hsr.mod.o
LD [M] net/hsr/hsr.ko
GEN scripts/gdb/linux/constants.py
na@ana-VirtualBox:~/linux-6.11.7$
```

## Valoración

Una práctica fácil e interesante, aunque larga debido a la tardanza de la compilación.