



Universidad Nacional Autónoma de México
Facultad de Ciencias

Administración de Sistemas UNIX/Linux
Semestre: 2025-2

Práctica 4

Compilando el Kernel vol.1

Profesor: Luis Enrique Serrano Gutiérrez
Ayudante: Erick Iram García Velasco

Equipo:
Bonilla Reyes Dafne 319089660
Cabrera Ramirez Carlos 316699910
García Ponce José Camilo 319210536
García Sánchez Abigail 423026522

Marzo, 2024

Objetivos

- Comprender el proceso de compilación e instalación del kernel de Linux.
- Actualizar manualmente el kernel, asegurando que la nueva versión sea funcional y que los módulos requeridos sean correctamente instalados.
- Familiarizarnos con una porción de las opciones de configuración disponibles.
- Comparar diferentes versiones del kernel.

Desarrollo

Conexión a la computadora

1. Para hacer la conexión a la computadora del laboratorio primero nos conectamos usando **SSH** al sistema Ada, esto lo hacemos usando nuestro usuario y contraseña. Por ejemplo:

`ssh jcgarcia@132.248.181.180`

```
camilo@wowi:~$ ssh jcgarcia@132.248.181.180
jcgarcia@132.248.181.180's password:
Last login: Mon Mar 24 19:37:35 2025 from 189.217.203.50
jcgarcia@ada:~$
```

2. Luego dentro de Ada nos vamos a conectar a la computadora mediante:

`ssh yankees@192.168.23.46`

```
jcgarcia@ada:~$ ssh yankees@192.168.23.46
yankees@192.168.23.46's password:
Linux yankees 6.1.0-29-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.123-1 (2025-01-02)
) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 24 19:01:55 2025 from 192.168.13.180
yankees@yankees:~$
```

Instalación de bibliotecas

1. Primero, instalamos lo necesario para compilar el kernel con los comandos:

`sudo apt update`

```
sudo apt install build-essential libncurses-dev bison flex  
libssl-dev libelf-dev
```

```
yankees@yankees:~$ sudo apt update  
[sudo] contraseña para yankees:  
Obj:1 http://deb.debian.org/debian bookworm InRelease  
Obj:2 http://security.debian.org/debian-security bookworm-security InRelease  
Obj:3 http://deb.debian.org/debian bookworm-updates InRelease  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
Se pueden actualizar 30 paquetes. Ejecute «apt list --upgradable» para verlos.
```

```
yankees@yankees:~$ sudo apt install build-essential libncurses-dev bison flex libssl-dev libelf-dev  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-12 dirmngr dpkg-dev fakeroot  
  fontconfig-config fonts-dejavu-core g++ g++-12 gcc gcc-12 gnupg gnupg-l10n gnupg-utils gpg  
  gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libabsl20220623 libalgorithm-diff-perl  
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libaom3 libasan8 libassuan0 libatomic1 libavif15  
  libbinutils libc-bin libc-dev-bin libc-devtools libc-l10n libc6 libc6-dev libcc1-0 libcrypt-dev  
  libctf-nobfd0 libctf0 libdav1d6 libde265-0 libdeflate0 libdpkg-perl libfakeroot  
  libfile-fcntllock-perl libfl-dev libfl2 libfontconfig1 libgav1-1 libgcc-12-dev libgd3 libgomp1  
  libgprofng0 libheif1 libisl23 libitm1 libjbig0 libjpeg62-turbo libksba8 liblerc4 liblsan0 libmpc3  
  libmpfr6 libncurses6 libnpt0 libnsl-dev libnuma1 libquadmath0 librav1e0 libstdc++-12-dev  
  libsvtavlenc1 libtiff6 libtirpc-dev libtsan2 libubsan1 libwebp7 libx265-199 libxpm4 libyuv0  
  linux-libc-dev locales m4 make manpages-dev patch pinentry-curses rpcsvc-proto zlib1g-dev
```

Identificación de la versión del kernel

1. Vemos la versión actual del kernel con el comando:

```
uname -r
```

```
yankees@yankees:~$ uname -r  
6.1.0-29-amd64
```

Descarga del código fuente del kernel

1. Descargamos el código del kernel (la versión 6.13) usando el comando:

```
wget https://www.kernel.org/pub/linux/kernel/v6.x/linux-6.13.tar.gz
```

```
yankees@yankees:~$ wget https://www.kernel.org/pub/linux/kernel/v6.x/linux-6.13.tar.gz  
--2025-03-24 19:44:11-- https://www.kernel.org/pub/linux/kernel/v6.x/linux-6.13.tar.gz  
Resolviendo www.kernel.org (www.kernel.org)... 151.101.181.55, 2a04:4e42:2b::311  
Conectando con www.kernel.org (www.kernel.org)[151.101.181.55]:443... conectado.  
Petición HTTP enviada, esperando respuesta... 200 OK  
Longitud: 240912392 (230M) [application/x-gzip]  
Grabando a: «linux-6.13.tar.gz»  
  
linux-6.13.tar.gz 13%[====>] 31.61M 8.49MB/s eta 23s
```

Extracción del código fuente del kernel

1. Movemos lo que descargamos a `/usr/src` con el comando:

```
sudo mv linux-6.13.tar.gz /usr/src/
```

2. Luego nos movemos a `/usr/src`

```
yankees@yankees:~$ sudo mv linux-6.13.tar.gz /usr/src/  
yankees@yankees:~$ cd /usr/src/  
yankees@yankees:/usr/src$ ls  
linux-6.13.tar.gz
```

- Después, extraemos lo que descargamos usando el comando:

```
sudo tar -xvf linux-6.13.tar.gz
```

- Y finalmente, nos movemos a la carpeta que se creó.

```
linux-6.13/virt/lib/irqbypass.c
yankees@yankees:/usr/src$ ls
linux-6.13  linux-6.13.tar.gz
yankees@yankees:/usr/src$ cd linux-6.13/
```

Configuración del kernel

- Copiamos el archivo de configuración que se usó para compilar este kernel a nuestro directorio actual con el comando:

```
sudo cp /boot/config-6.1.0-29-amd64 .config
```

```
yankees@yankees:/usr/src/linux-6.13$ sudo cp /boot/config-6.1.0-29-amd64 .config
yankees@yankees:/usr/src/linux-6.13$
```

- Posteriormente, para cargar la configuración anterior (**.config**) y ver sobre nuevas opciones que tiene el kernel, ejecutamos el comando:

```
sudo make oldconfig
```

```
yankees@yankees:/usr/src/linux-6.13$ sudo make oldconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
HOSTCC scripts/kconfig/confdata.o
HOSTCC scripts/kconfig/expr.o
LEX scripts/kconfig/lexer.lex.c
YACC scripts/kconfig/parser.tab.[ch]
HOSTCC scripts/kconfig/lexer.lex.o
HOSTCC scripts/kconfig/menu.o
HOSTCC scripts/kconfig/parser.tab.o
HOSTCC scripts/kconfig/preprocess.o
HOSTCC scripts/kconfig/symbol.o
HOSTCC scripts/kconfig/util.o
HOSTLD scripts/kconfig/conf
```

- Notemos que le damos Enter a todas las opciones/preguntas, ya que eso usa la opción por default. Además, para acelerar este proceso podemos usar el siguiente comando para siempre seleccionar la opción por default.:

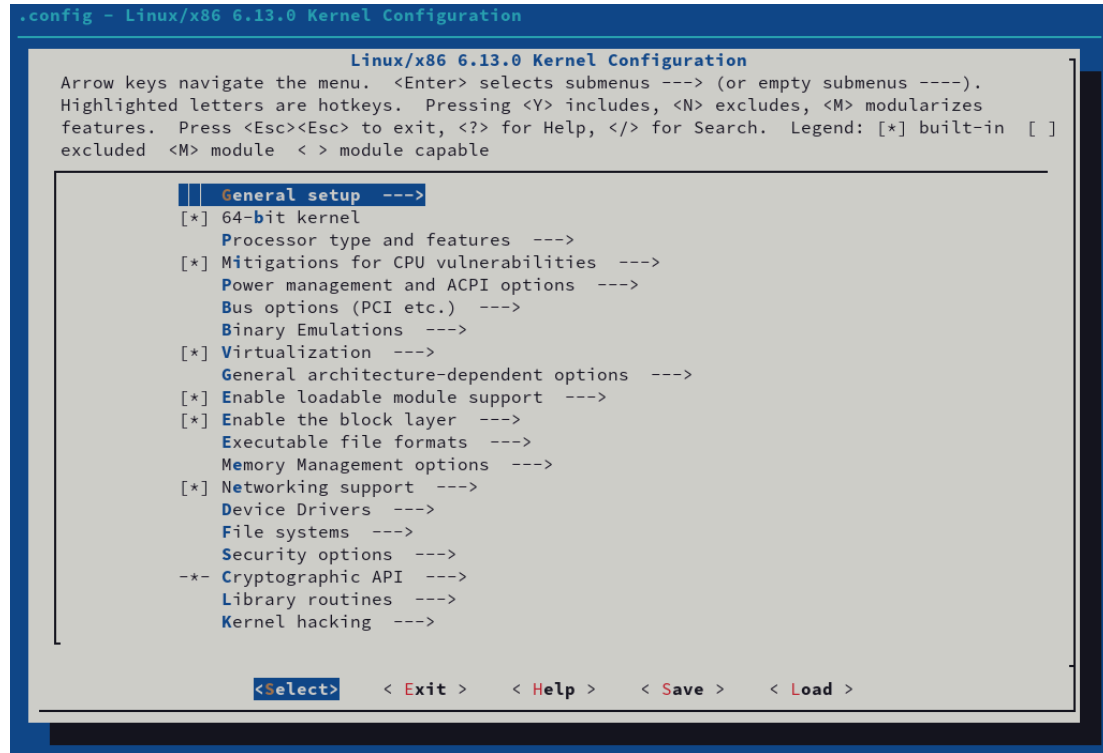
```
yes "" | sudo make oldconfig
```

```
yankees@yankees:/usr/src/linux-6.13$ yes "" | sudo make oldconfig
.config:887:warning: symbol value '0' invalid for BASE_SMALL
.config:6364:warning: symbol value 'm' invalid for FB_BACKLIGHT
.config:8020:warning: symbol value 'm' invalid for VFIO_VIRQFD
.config:9250:warning: symbol value 'm' invalid for ANDROID_BINDER_IPC
.config:9410:warning: symbol value 'm' invalid for FSCACHE
*
* Restart config...
*
*
* General setup
*
Compile also drivers which will not load (COMPILE_TEST) [N/y/?] n
Compile the kernel with warnings as errors (WERROR) [N/y/?] n
```

3. Después, para poder seleccionar las opciones que queremos para compilar nuestro kernel ejecutamos el comando:

`sudo make menuconfig`

Para esta práctica veremos las opciones de **Processor type and features**, **Networking support**, **Device Drivers** y **File systems**.



4. De Processor type and features seleccionamos:

```
[*] Symmetric multi-processing support
-- Support x2apic
[ ] Enable MSI and MSI-x delivery by posted interrupts
[*] Enable MPS table
[*] x86 CPU resource control support
[ ] Flexible Return and Event Delivery
[ ] Support for extended (non-PC) x86 platforms
[*] Intel Low Power Subsystem Support
[*] AMD ACPI2Platform devices support
-- Intel SoC IOSF Sideband support for SoC platforms
[ ] Enable IOSF sideband access through debugfs
[*] Single-depth WCHAN output
[*] Linux guest support --->
    Processor family (Generic-x86-64) ---->
[ ] Supported processor vendors ----
-- Enable DMI scanning
[*] Old AMD GART IOMMU support
[*] Enable Maximum number of SMP Processors and NUMA Nodes
[*] Cluster scheduler support
[*] Multi-core scheduler support
[*] CPU core priorities scheduler support
[*] Reroute for broken boot IRQs
[*] Machine Check / overheating reporting
[ ] Support for deprecated /dev/mcelog character device
[*] Intel MCE features
v(+)
```

```

[*] Intel MCE features
[*] AMD MCE features
<M> Machine check injector support
    Performance monitoring --->
[*] Enable support for 16-bit segments
[*] Enable vsyscall emulation
[*] IOPERM and IOPL Emulation
[ ] Late microcode loading (DANGEROUS)
<M> /dev/cpu/*/msr - Model-specific register support
<M> /dev/cpu/*/cpuid - CPU information support
[*] Enable 5-level page tables support
[*] Enable statistic for Change Page Attribute
[*] AMD Secure Memory Encryption (SME) support
[*] NUMA Memory Allocation and Scheduler Support
[*] Old style AMD Opteron NUMA detection
[*] ACPI NUMA detection
[ ] Enable sysfs memory/probe interface
<M> Support non-standard NVDIMMs and ADR protected memory
[*] Check for low memory corruption
[*] Set the default setting of memory_corruption_check (NE
[*] MTRR (Memory Type Range Register) support
[*] MTRR cleanup support
(0) MTRR cleanup enable value (0-1)
(1) MTRR cleanup spare reg num (0-7)
[*] x86 PAT support
v(+)

```

```

[*] x86 PAT support
[*] User Mode Instruction Prevention
[ ] Indirect Branch Tracking
[*] Memory Protection Keys
    TSX enable mode (off) --->
[*] Software Guard eXtensions (SGX)
[ ] X86 userspace shadow stack
[*] EFI runtime service support
[*] EFI stub support
[*] EFI handover protocol (DEPRECATED)
[*] EFI mixed-mode support
--*-- Export EFI runtime maps to sysfs
    Timer frequency (250 HZ) --->
(0x1000000) Physical address where the kernel is loaded
--*-- Build a relocatable kernel
[*] Randomize the address of the kernel image (KASLR)
(0x200000) Alignment value to which kernel should be aligned
[*] Randomize the kernel memory sections
(0xa) Physical memory mapping padding
[ ] Disable the 32-bit vDSO (needed for glibc 2.3.3)
    vsyscall table for legacy applications (None) --->
[*] Built-in kernel command line
( ) Built-in kernel command string (NEW)
[*] Enable the LDT (local descriptor table)
[ ] Enforce strict size checking for sigaltstack
v(+)

```

```

[*] Built-in kernel command line
() Built-in kernel command string (NEW)
[*] Enable the LDT (local descriptor table)
[ ] Enforce strict size checking for sigaltstack
[*] Kernel Live Patching
[*] Split Lock Detect and Bus Lock Detect support

```

5. De Networking Support seleccionamos:

```

-- Networking support
Networking options --->
[*] Amateur Radio support --->
<M> CAN bus subsystem support --->
<M> Bluetooth subsystem support --->
{M} RxRPC session sockets
[*] IPv6 support for RxRPC
[ ] Inject packet loss into RxRPC packet stream
[ ] Inject delay into packet reception
[*] RxRPC dynamic debugging
[*] RxRPC Kerberos security
< > RxRPC test service
<M> KCM sockets
[*] MCTP core protocol support ----
-- Wireless --->
<M> RF switch subsystem support --->
<M> Plan 9 Resource Sharing Support (9P2000) --->
< > CAIF support ----
{M} Ceph core library
[ ] Include file:line in ceph debug output
[ ] Use in-kernel support for DNS lookup
<M> NFC subsystem support --->
{M} Packet-sampling netlink channel ----
{M} Inter-FE based on IETF ForCES InterFE LFB ----
-- Network light weight tunnels
v(+)

```

```

-- Network light weight tunnels
[*] Execute BPF program as route nexthop action
[*] Page pool stats
{M} Generic failover module
[*] Netlink interface for ethtool

```


6. De Device Drivers seleccionamos:

```
[ ] EISA support ----
[*] PCI support --->
<M> PCCard (PCMCIA/CardBus) support --->
< > RapidIO support ----
    Generic Driver Options --->
    Bus devices --->
    Cache Drivers ----
{*} Connector - unified userspace <-> kernelspace linker --->
    Firmware Drivers --->
<M> GNSS receiver support --->
<M> Memory Technology Device (MTD) support --->
[*] Device Tree and Open Firmware support --->
<M> Parallel port support --->
-* Plug and Play support --->
[*] Block devices --->
    NVME Support --->
    Misc devices --->
    SCSI device support --->
<M> Serial ATA and Parallel ATA drivers (libata) --->
[*] Multiple devices driver support (RAID and LVM) --->
<M> Generic Target Core Mod (TCM) and ConfigFS Infrastructure
[*] Fusion MPT device support --->
    IEEE 1394 (FireWire) support --->
[*] Macintosh device drivers --->
-* Network device support --->
v(+)
```

```
-* Network device support --->
    Input device support --->
    Character devices --->
    I2C support --->
< > I3C support ----
[*] SPI support --->
< > SPMI support ----
< > HSI support ----
-* PPS support --->
    PTP clock support --->
-* Pin controllers --->
-* GPIO Support --->
<M> Dallas's 1-wire support --->
[ ] Board level reset or power off ----
< > Power Sequencing support ----
-* Power supply class support --->
{*} Hardware Monitoring support --->
-* Thermal drivers --->
[*] Watchdog Timer Support --->
{M} Sonics Silicon Backplane support --->
{M} Broadcom specific AMBA --->
    Multifunction device drivers --->
-* Voltage and Current Regulator Support --->
<M> Remote Controller support --->
    CEC support --->
v(+)
```



```

    CEC support --->
<M> Multimedia support --->
    Graphics support --->
[ ] Compute Acceleration Framework ----
<M> Sound card support --->
[*] HID bus support --->
[*] USB support --->
<M> MMC/SD/SDIO card support --->
<M> Universal Flash Storage Controller --->
<M> Sony MemoryStick card support --->
*- LED Support --->
[*] Accessibility support --->
<M> InfiniBand support --->
<*> EDAC (Error Detection And Correction) reporting --->
[*] Real Time Clock --->
[*] DMA Engine support --->
    DMABUF options --->
{M} Userspace I/O drivers --->
<M> VFIO Non-Privileged userspace driver framework --->
[*] Virtualization drivers --->
[*] Virtio drivers --->
< > vDPA drivers ----
[*] VHOST drivers --->
||| Microsoft Hyper-V guest support --->
    Xen driver support --->
v(+)

```

```

    Xen driver support --->
< > Greybus support ----
<M> Data acquisition support (comedi) --->
[*] Staging drivers --->
[ ] Platform support for Goldfish virtual devices ----
[*] Platform support for Chrome hardware --->
[ ] Platform support for CZ.NIC's Turris hardware ----
[ ] Platform support for Mellanox hardware ----
[*] Microsoft Surface Platform-Specific Device Drivers --->
*- X86 Platform Specific Device Drivers --->
*- Common Clock Framework --->
[ ] Hardware Spinlock drivers ----
    Clock Source drivers ----
*- Mailbox Hardware Support --->
[*] IOMMU Hardware Support --->
    Remoteproc drivers --->
    Rpmmsg drivers --->
<M> SoundWire support --->
    SOC (System On Chip) specific Drivers --->
    PM Domains --->
*- Generic Dynamic Voltage and Frequency Scaling (DVFS) support --->
<M> External Connector Class (extcon) support --->
[*] Memory Controller drivers ----
<M> Industrial I/O support --->
< > Non-Transparent Bridge support ----
v(+)

```

```

< > Non-Transparent Bridge support ----
[*] Pulse-Width Modulation (PWM) Support --->
    IRQ chip support --->
< > IndustryPack bus support ----
-* Reset Controller Support --->
    PHY Subsystem --->
-* Generic powercap sysfs driver --->
< > MCB support ----
    Performance monitor support --->
-* Reliability, Availability and Serviceability (RAS) feature
<M> Unified support for USB4 and Thunderbolt --->
    Android --->
{M} NVDIMM (Non-Volatile Memory Device) Support --->
-* DAX: direct access to differentiated memory --->
-* NVMEM Support --->
    HW tracing support --->
< > FPGA Configuration Framework ----
< > FSI support (NEW) ----
< > Trusted Execution Environment support ----
< > Eckelmann SIOX Support ----
<M> SLIMbus support --->
[ ] On-Chip Interconnect management support ----
< > Counter support ----
< > MOST (Media Oriented Systems Transport) support -----
< > PECI support ----
v(+)

```

```

< > MOST (Media Oriented Systems Transport) support ----
< > PECI support ----
[*] Hardware Timestamping Engine (HTE) Support ----

```

7. De File systems seleccionamos:

```

[ ] Validate filesystem parameter description
< > Second extended fs support (DEPRECATED)
< > The Extended 3 (ext3) filesystem
<M> The Extended 4 (ext4) filesystem
[*] Use ext4 for ext2 file systems
[*] Ext4 POSIX Access Control Lists
[*] Ext4 Security Labels
[*] Ext4 debugging support
[ ] JBD2 (ext4) debugging support
<M> JFS filesystem support
[*] JFS POSIX Access Control Lists
[*] JFS Security Labels
[ ] JFS debugging
[ ] JFS statistics
<M> XFS filesystem support
[*] Support deprecated V4 (crc=0) format
[*] Support deprecated case-insensitive ascii (ascii-ci=1) f
[*] XFS Quota support
[*] XFS POSIX ACL support
[*] XFS Realtime subvolume support
[ ] XFS online metadata check support
[ ] XFS Verbose Warnings
[ ] XFS Debugging support
<M> GFS2 file system support
[*] GFS2 DLM locking
v(+)

```

```

[*] GFS2 DLM locking
<M> OCFS2 file system support
<M> O2CB KernelSpace Clustering
<M> OCFS2 Userspace Clustering
[*] OCFS2 statistics
[*] OCFS2 logging support
[ ] OCFS2 expensive checks
<M> Btrfs filesystem support
[*] Btrfs POSIX Access Control Lists
[ ] Btrfs will run sanity tests upon loading
[ ] Btrfs debugging support
[ ] Btrfs assert support
[ ] Btrfs experimental features
[ ] Btrfs with the ref verify tool compiled in
<M> NILFS2 file system support
<M> F2FS filesystem support
[*] F2FS Status Information
-- F2FS extended attributes
[*] F2FS Access Control Lists
[*] F2FS Security Labels
[ ] F2FS consistency checking feature
[ ] F2FS fault injection facility
[*] F2FS compression feature
[*] LZ0 compression support
[*] LZ0-RLE compression support
v(+)

```

```

[*] LZ0-RLE compression support
[*] LZ4 compression support
[*] LZ4HC compression support
[*] ZSTD compression support
[*] F2FS IO statistics information
[ ] F2FS unfair rw_semaphore
< > bcache filesystem support (EXPERIMENTAL)
<M> zonefs filesystem support
[*] File system based Direct Access (DAX) support
-- Enable filesystem export operations for block IO
[*] Enable POSIX file locking API
[*] FS Encryption (Per-file encryption)
[*] FS Verity (read-only file-based authenticity protection)
[*] FS Verity builtin signature support
[*] Dnotify support
[*] Inotify support for userspace
[*] Filesystem wide access notification
[*] fanotify permissions checking
-- Quota support
[*] Report quota messages through netlink interface
[ ] Additional quota sanity checks
<M> Old quota format support
<M> Quota format vfstv0 and vfstv1 support
<M> Kernel automounter support (supports v3, v4 and v5)
<M> FUSE (Filesystem in Userspace) support
v(+)

```

```

<M> Old quota format support
<M> Quota format vfstv0 and vfstv1 support
<M> Kernel automounter support (supports v3, v4 and v5)
<M> FUSE (Filesystem in Userspace) support
<M> Character device in Userspace support
<M> Virtio Filesystem
[*] Virtio Filesystem Direct Host Memory Access support
[*] FUSE passthrough operations support
<M> Overlay filesystem support
[ ] Overlayfs: turn on redirect directory feature by default
[*] Overlayfs: follow redirects even if redirects are turned
[ ] Overlayfs: turn on inodes index feature by default
[ ] Overlayfs: auto enable inode number mapping
[ ] Overlayfs: turn on metadata only copy up feature by default
[ ] Overlayfs: turn on extra debugging checks
Caches --->
CD-ROM/DVD Filesystems --->
DOS/FAT/EXFAT/NT Filesystems --->
Pseudo filesystems --->
--*-- Miscellaneous filesystems --->
[*] Network File Systems --->
--*-- Native language support --->
<M> Distributed Lock Manager (DLM) --->
<*> UTF-8 normalization and casefolding support
< > Test UTF-8 normalization support

```

Identificación del kernel

1. Usamos:

sudo nano .config

Para poder agregar:

CONFIG_BUILD_SALT="_ASUL"

Y así poder identificar nuestro kernel.

```

CONFIG_LOCALVERSION=""
# CONFIG_LOCALVERSION_AUTO is not set
CONFIG_BUILD_SALT="_ASUL"
CONFIG_HAVE_KERNEL_GZIP=y

```

Compilación del kernel

1. Compilamos el kernel y los módulos con el comando:

sudo make -j\$(nproc)

- *Nota: Durante el proceso hubo un error y tuvimos que ejecutar el comando:*

sudo apt install bc

```

/bin/sh: 1: bc: not found
CHKSHA1 include/linux/atomic/atomic-arch-fallback.h
CHKSHA1 include/linux/atomic/atomic-instrumented.h
make[2]: *** [Kbuild:24: include/generated/timeconst.h] Error 127
make[2]: *** Se espera a que terminen otras tareas....
CHKSHA1 include/linux/atomic/atomic-long.h
make[1]: *** [/usr/src/linux-6.13/Makefile:1263: prepare0] Error 2
make[1]: *** Se espera a que terminen otras tareas....
LD      /usr/src/linux-6.13/tools/objtool/objtool-in.o
LINK    /usr/src/linux-6.13/tools/objtool/objtool
make: *** [Makefile:251: __sub-make] Error 2
yankees@yankees: /usr/src/linux-6.13$

```

```

yankees@yankees:/usr/src/linux-6.13$ sudo make -j$(nproc)
mkdir -p /usr/src/linux-6.13/tools/objtool && make O=/usr/src/linux-6.13 subdir=tools/objtool --no-print-directo
ry -C objtool
  UPD      include/generated/timeconst.h
  UPD      include/generated/bounds.h
  INSTALL  libsubcmd_headers
  CC       arch/x86/kernel/asm-offsets.s
  UPD      include/generated/asm-offsets.h
  CALL     scripts/checksyscalls.sh
  LDS      scripts/module.lds
  HOSTCC   usr/gen_init_cpio
  CC       init/main.o
  CC       certs/system_keyring.o
  CC       ipc/compat.o
  CC       mm/filemap.o
  CC       arch/x86/coco/sev/core.o
  CC       fs/notify/dnotify/dnotify.o

```

```

  STUBCOPY drivers/firmware/efi/libstub/tpm.stub.o
  STUBCOPY drivers/firmware/efi/libstub/vsprintf.stub.o
  AR       drivers/firmware/imx/built-in.a
  AR       drivers/firmware/psci/built-in.a
  AR       drivers/firmware/qcom/built-in.a
  AR       drivers/firmware/smccc/built-in.a
  AR       drivers/firmware/tegra/built-in.a
  AR       drivers/firmware/xilinx/built-in.a
  CC       drivers/firmware/dmi_scan.o
  CC [M]   drivers/crypto/ccp/ccp-dev-v3.o
  AR       drivers/crypto/stm32/built-in.a
  AR       drivers/crypto/xilinx/built-in.a
  AR       drivers/crypto/hisilicon/built-in.a
  STUBCOPY drivers/firmware/efi/libstub/smbios.stub.o
  STUBCOPY drivers/firmware/efi/libstub/unaccepted_memory.stub.o
  CC [M]   drivers/crypto/ccp/ccp-dev-v5.o
  AR       drivers/crypto/intel/keembay/built-in.a
  AR       drivers/crypto/intel/ixp4xx/built-in.a
  CC [M]   drivers/crypto/intel/qat/qat_common/adf_cfg.o

```

```

  LD [M]   net/vmw_vsock/vsock_diag.ko
  LD [M]   net/vmw_vsock/vmw_vsock_vmci_transport.ko
  LD [M]   net/vmw_vsock/vmw_vsock_virtio_transport.ko
  LD [M]   net/vmw_vsock/vmw_vsock_virtio_transport_common.ko
  LD [M]   net/vmw_vsock/hv_sock.ko
  LD [M]   net/vmw_vsock/vsock_loopback.ko
  LD [M]   net/nsh/nsh.ko
  LD [M]   net/qrtr/qrtr.ko
  LD [M]   net/qrtr/qrtr-mhi.ko
yankees@yankees:/usr/src/linux-6.13$ █

```

2. Después de cómo unas horas y terminar de compilar, ahora tenemos que compilar los módulos con el comando:

sudo make modules_install

```

yankees@yankees:/usr/src/linux-6.13$ sudo make modules_install
[sudo] contraseña para yankees:
  SYMLINK  /lib/modules/6.13.0/build
  INSTALL  /lib/modules/6.13.0/modules.order
  INSTALL  /lib/modules/6.13.0/modules.builtin
  INSTALL  /lib/modules/6.13.0/modules.builtin.modinfo
  INSTALL  /lib/modules/6.13.0/kernel/arch/x86/events/amd/power.ko
  SIGN     /lib/modules/6.13.0/kernel/arch/x86/events/amd/power.ko
  INSTALL  /lib/modules/6.13.0/kernel/arch/x86/events/intel/intel-uncore.ko
  SIGN     /lib/modules/6.13.0/kernel/arch/x86/events/intel/intel-uncore.ko
  INSTALL  /lib/modules/6.13.0/kernel/arch/x86/events/intel/intel-cstate.ko

```

```
INSTALL /lib/modules/6.13.0/kernel/net/nsh/nsh.ko
SIGN    /lib/modules/6.13.0/kernel/net/nsh/nsh.ko
INSTALL /lib/modules/6.13.0/kernel/net/qrtr/qrtr.ko
SIGN    /lib/modules/6.13.0/kernel/net/qrtr/qrtr.ko
INSTALL /lib/modules/6.13.0/kernel/net/qrtr/qrtr-mhi.ko
SIGN    /lib/modules/6.13.0/kernel/net/qrtr/qrtr-mhi.ko
DEPMOD  /lib/modules/6.13.0
```

Instalación del kernel

1. Instalamos el kernel con el comando:

sudo make install

```
yankees@yankees:/usr/src/linux-6.13$ sudo make install
INSTALL /boot
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 6.13.0 /boot/vmlinuz-6.13.0
update-initramfs: Generating /boot/initrd.img-6.13.0
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 6.13.0 /boot/vmlinuz-6.13.0
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.13.0
Found initrd image: /boot/initrd.img-6.13.0
Found linux image: /boot/vmlinuz-6.1.0-29-amd64
Found initrd image: /boot/initrd.img-6.1.0-29-amd64
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
```

Actualización de la configuración

1. Actualizamos la configuración del grub con el comando:

sudo update-grub

```
yankees@yankees:/usr/src/linux-6.13$ sudo update-grub
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.13.0
Found initrd image: /boot/initrd.img-6.13.0
Found linux image: /boot/vmlinuz-6.1.0-29-amd64
Found initrd image: /boot/initrd.img-6.1.0-29-amd64
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
```

2. Después, reiniciamos el sistema con el comando:

sudo reboot

```
yankees@yankees:/usr/src/linux-6.13$ sudo reboot

Broadcast message from root@yankees on pts/1 (Mon 2025-03-24 21:15:17 CST):

The system will reboot now!

yankees@yankees:/usr/src/linux-6.13$ Connection to 192.168.23.46 closed by remote host.
```

Verificación del sistema

1. Por último, revisamos que el nuevo kernel esté en uso con el comando:

```
uname -r  
yankees@yankees:~$ uname -r  
6.13.0  
yankees@yankees:~$
```

Sistema Ada

Según la información mencionada en clase y recopilada en internet, el sistema ADA (Ambientes Digitales de Aprendizaje), los ADA surgieron con la emergencia por COVID-19. Pueden fortalecer la educación a distancia y presencial. En la UNAM se han creado más de 300 ambientes digitales de aprendizaje (ADA) que ofrecen propuestas didácticas completas para el estudio de diversas.

Particularmente en esta práctica lo usamos para conectarnos a través de un usuario y contraseña asignados por la facultad para tener un entorno virtual de tipo página que permite y habilita el acceso a los sistemas de las computadoras de la facultad sin necesidad de estar físicamente en ellos.

Para conectarnos a Ada desde la facultad podemos usar la IP local 192.168.181.180 (aunque algunas veces hemos tenido problemas con esta IP) o podemos usar la IP pública 132.248.181.180 para poder conectarnos desde nuestras casitas, ya dentro de Ada solo tenemos que conectarnos con ssh a la computadora del laboratorio, la cual si IP es 192.168.23.46.

Características del kernel

- Describe todas las características que agregaste o eliminaste del kernel y explica por qué lo hiciste, detallando cómo pueden impactar en el rendimiento, compatibilidad o seguridad de tu sistema.

Las características que agregamos fueron las siguientes:

- Para **processor type and features**:

- *Built-in kernel command line*

Esta opción permite definir una línea de comandos del kernel que se incluirá de forma estática dentro del propio kernel, en lugar de depender de un bootloader (como GRUB) para proporcionar opciones de arranque.

Agregamos esta característica porque es útil en sistemas embebidos o dispositivos que no usan un gestor de arranque configurable, además de que nos permite definir parámetros críticos del kernel sin depender de la configuración externa.

- Impacto: Esta característica afecta la compatibilidad, ya que limita la capacidad de ajustar parámetros sin recompilación.
- *Check for low memory corruption*

Esta opción habilita una comprobación periódica para detectar corrupción de memoria en la parte baja de la RAM.

Agregamos esta característica porque ayuda a detectar errores de memoria causados por hardware defectuoso o malware que intenta modificar áreas críticas de la memoria.

- Impacto: Afecta la seguridad y estabilidad del sistema, pero puede reducir el rendimiento.
- *Enable statistic for Change Page Attribute*

Habilita la recopilación de estadísticas sobre los cambios en los atributos de las páginas de memoria (como permisos y caché).

Agregamos esta característica porque es útil para la depuración y análisis de rendimiento en entornos de desarrollo o pruebas.

- Impacto: Afecta el rendimiento, ya que recopilar estadísticas consume recursos adicionales.
- Para **networking support**:

- *Page pool stats*

Habilita la recopilación de estadísticas sobre el uso de pools de páginas en la pila de red, optimizando la asignación y liberación de memoria.

Agregamos esta característica porque es útil para diagnosticar problemas de rendimiento en redes de alta velocidad, además de que ayuda a optimizar el uso de memoria en servidores con tráfico intensivo.

- Impacto: Afecta el rendimiento, pero solo en casos de redes de alta demanda.
- *MCTP core protocol support*

Proporciona soporte para MCTP, un protocolo de comunicación utilizado en servidores y sistemas de administración remota.

Agregamos esta característica porque es requerido en sistemas con firmware avanzado o dispositivos que dependen de BMC (Baseboard Management Controller).

- Impacto: Afecta la compatibilidad, pero no el rendimiento ni la seguridad en la mayoría de los casos.

- Para **device drivers**:

- *Device Tree and Open Firmware support*

Proporciona soporte para Device Tree y Open Firmware, utilizados en arquitecturas como ARM y PowerPC para describir el hardware sin necesidad de usar ACPI.

Agregamos esta característica porque es esencial en sistemas embebidos y dispositivos ARM, además de que permite mejor compatibilidad con hardware sin modificar el kernel.

- Impacto: Afecta la compatibilidad, pero no el rendimiento o la seguridad.

- *Hardware Timestamping Engine (HTE) Support*

Proporciona soporte para el motor de sellado de tiempo en hardware, lo que mejora la precisión de eventos en sistemas que requieren sincronización precisa.

Agregamos esta característica porque es útil en redes de baja latencia o sistemas industriales donde la sincronización es clave, además de que puede mejorar la precisión en registros de eventos de hardware.

- Impacto: Afecta el rendimiento solo si se usa hardware compatible con esta función.

- Para **file systems**:

- *Ext4 debugging support*

Habilita opciones adicionales de depuración para el sistema de archivos Ext4.

Agregamos esta característica porque es útil para detectar errores en el sistema de archivos durante el desarrollo o pruebas, además de

que permite encontrar problemas en discos o almacenamiento afectado.

- Impacto: Afecta el rendimiento, pero mejora la capacidad de diagnóstico en sistemas en desarrollo.

Nota: Nótese que procuramos no quitar tantas opciones (o ninguna), esto debido para evitar que quitemos una opción la cual sea importante para que todo funcione correctamente (en un intento de compilar el kernel de una máquina virtual nos pasó y obtuvimos errores de kernel panic), esto principalmente ya que como son tantas opciones es algo intimidante, durante la semana el profesor nos dio una idea que podemos quitar los drivers de red y cpu que no usa la computadora, esto es una buena idea pero no la aplicamos ya que compilar el kernel toma algo de tiempo y prender la computadora que usamos tiene un truco (tenemos que estar físicamente en la computadora) así que apagarla es algo riesgoso, pero las principales ventajas de quitar opciones sería que el kernel pesara menos, compilara más rápido y fuera algo más eficiente la computadora (lo cual no es tan importante ya que actualmente la computadora no realizar procesos muy intensos).

En resumen agregamos algunas opciones que nos sonaron familiares e interesantes, no tanto para que el kernel fuera más eficiente o pesara menos, si no porque es lo que encontramos familiar y pensamos que agregarlo no podría traer problemas.

- Investiga también aquellas características que te parecieron más interesantes (dentro de las secciones mencionadas), no necesariamente aquellas que modificaste y explica su función.

- Para **processor type and features**:

- *Enable MPS table*

La MPS (MultiProcessor Specification) table es utilizada en sistemas multiprocesador antiguos para describir la configuración de CPUs y dispositivos de interrupción en la BIOS. Esta característica permite la compatibilidad con sistemas antiguos que no usan ACPI para administrar múltiples procesadores y proporciona información al kernel sobre la topología del sistema (CPUs, buses y controladores de interrupción).

Ventajas: La MPS Table es útil en sistemas antiguos, ya que proporciona una manera sencilla de gestionar múltiples procesadores y dispositivos de interrupción sin requerir tecnologías más avanzadas como ACPI. Facilita la configuración de la topología del sistema, lo que ayuda en la administración de recursos en sistemas con hardware más limitado.

Desventajas: La MPS Table es obsoleta en comparación con tecnologías más modernas como ACPI, que ofrecen una gestión más

avanzada y flexible del hardware y la energía. Además, no es adecuada para configuraciones de hardware complejas y carece de flexibilidad para adaptarse a sistemas modernos.

- *Intel Low Power Subsystem Support*

Este es el soporte para el subsistema de baja potencia de Intel (LPSS), presente en procesadores Intel modernos (especialmente en plataformas móviles y embebidas). Esta característica gestiona controladores de baja potencia en hardware de Intel (controladores de GPIO, I2C, SPI, UART) y permite reducir el consumo energético en dispositivos móviles y laptops con procesadores Intel.

Ventajas: LPSS optimiza el consumo de energía en dispositivos móviles y embebidos, gestionando eficientemente los controladores de baja potencia como GPIO, I2C, SPI y UART. Esto resulta en una mayor duración de la batería y mejor eficiencia energética en dispositivos portátiles, lo que es esencial para las plataformas móviles y sistemas embebidos.

Desventajas: Esta tecnología está limitada a procesadores Intel, lo que la hace incompatible con plataformas que utilizan otros fabricantes de chips. Además, su implementación puede ser compleja, especialmente en sistemas embebidos, donde se requieren ajustes adicionales para garantizar una gestión eficiente de la energía.

- *CPU Frequency Scaling*

Función: Esta característica permite que la frecuencia del procesador se ajuste dinámicamente según la carga de trabajo. Si el sistema está inactivo, la frecuencia del procesador se reduce para ahorrar energía, y si la carga aumenta, la frecuencia se incrementa para mejorar el rendimiento.

Ventajas: Mejora la eficiencia energética, especialmente en sistemas móviles o laptops, ayudando a prolongar la duración de la batería sin sacrificar demasiado el rendimiento.

Desventajas: Puede haber una ligera pérdida de rendimiento cuando el procesador no está funcionando a su frecuencia máxima, lo que puede afectar a las aplicaciones sensibles al rendimiento.

- **Para networking support:**

- *Netlink interface for ethtool*

Esta habilita una interfaz basada en Netlink para la herramienta ethtool, utilizada para configurar y diagnosticar interfaces de red. Esta característica permite que ethtool use Netlink en lugar de las llamadas ioctl tradicionales, mejorando la eficiencia y flexibilidad en la configuración de interfaces de red y facilita el ajuste dinámico de parámetros como velocidad, duplex, offloading y estadísticas de red.

Ventajas: La interfaz Netlink para ethtool mejora la eficiencia y flexibilidad al permitir que ethtool se comunique con el kernel de Linux mediante Netlink en lugar de las tradicionales llamadas ioctl. Esto facilita la configuración dinámica de parámetros de red como velocidad, dúplex, offloading y estadísticas, sin necesidad de reiniciar o detener las interfaces. Además, Netlink permite una mejor integración y escalabilidad, lo que es útil en entornos de red complejos.

Desventajas: Aunque Netlink ofrece ventajas en términos de eficiencia y flexibilidad, puede ser más complejo de implementar y administrar en comparación con las tradicionales llamadas ioctl. También, los programas que dependen de ethtool con Netlink podrían no ser totalmente compatibles con sistemas más antiguos que solo soportan ioctl, lo que podría limitar su adopción en ciertos entornos.

- *IPv6 Address Autoconfiguration*

Función: Habilita la capacidad de los dispositivos para autoconfigurar direcciones IPv6 sin necesidad de un servidor DHCP, utilizando un proceso conocido como SLAAC (Stateless Address Autoconfiguration).

Ventajas: Facilita la implementación de IPv6 en redes grandes sin necesidad de configurar manualmente las direcciones IP o usar un servidor DHCP.

Desventajas: Puede ser menos seguro si no se implementan correctamente las políticas de privacidad y seguridad en la red, ya que cualquier dispositivo en la red puede autoconfigurar una dirección IPv6.

- **Para device drivers:**

- *Watchdog Timer Support*

Soporte para watchdogs, dispositivos que reinician el sistema si este deja de responder. Esta característica se usa en sistemas críticos (servidores, dispositivos embebidos) para evitar bloqueos prolongados. Si el sistema deja de responder, el watchdog lo reinicia automáticamente.

Ventajas: El soporte para el watchdog timer es esencial en sistemas críticos, como servidores y dispositivos embebidos, ya que ayuda a prevenir bloqueos prolongados. Si el sistema deja de responder o se congela, el watchdog reinicia automáticamente el sistema, asegurando que la operación continúe sin intervención manual, lo que mejora la fiabilidad y disponibilidad del sistema.

Desventajas: Una desventaja del soporte para watchdog es que, si no está correctamente configurado o si el sistema se reinicia debido a fallos menores, podría interrumpir servicios o procesos importantes sin la oportunidad de corregir el error. Además, requiere una configuración adecuada para evitar reinicios innecesarios, lo que podría agregar complejidad al mantenimiento del sistema.

- *USB 3.0 Support*

Función: Habilita el soporte para el estándar USB 3.0, que ofrece una mayor velocidad de transferencia de datos en comparación con USB 2.0, así como mejor manejo de energía para dispositivos conectados.

Ventajas: Mejora el rendimiento de dispositivos USB al permitir mayores velocidades de transferencia de datos y mejor administración de energía.

Desventajas: Requiere que tanto el hardware como el sistema operativo lo soporten, y en algunos casos, puede no ser completamente compatible con versiones anteriores de USB o dispositivos más antiguos.

- **Para file systems:**

- *Kernel automounter support*

Soporte para automounting, permitiendo montar sistemas de archivos automáticamente cuando se acceden y desmontarlos cuando no se usan. Esta característica facilita el acceso a sistemas de archivos remotos o unidades externas sin intervención manual y es común en servidores para montar sistemas NFS o particiones de red según demanda.

Ventajas: El soporte para automounting facilita el acceso a sistemas de archivos remotos o unidades externas, montándolos automáticamente cuando se acceden y desmontándolos cuando no se usan. Esto ahorra tiempo y esfuerzo, ya que elimina la necesidad de intervención manual para montar o desmontar sistemas de archivos, y es especialmente útil en servidores que gestionan

recursos de red como NFS o comparticiones de red bajo demanda, mejorando la eficiencia operativa.

Desventajas: Una desventaja del automounting es que puede causar una sobrecarga si los puntos de montaje no se gestionan adecuadamente, lo que puede llevar a un uso innecesario de recursos cuando los sistemas de archivos remotos o unidades externas no son realmente necesarios. Además, si se configuran incorrectamente, podría haber problemas de acceso o pérdidas de datos en el caso de desconexiones inesperadas o fallos de red.

- *NTFS Support*

Función: Permite el acceso y la escritura en particiones NTFS, un sistema de archivos utilizado principalmente en sistemas Windows.

Ventajas: Facilita la interoperabilidad entre sistemas Linux y Windows, permitiendo el acceso y la manipulación de archivos en particiones NTFS.

Desventajas: El soporte para NTFS en Linux no es tan eficiente como en Windows, lo que puede causar un rendimiento inferior al trabajar con particiones NTFS, especialmente con archivos grandes.

Función de los paquetes instalados al inicio

Los paquetes que se instalaron al inicio son los siguientes:

- **build-essential**, este paquete incluye herramientas y bibliotecas útiles para la compilación de software de Linux. Por ejemplo, tiene el compilador gcc y el comando make que usamos varias veces en esta práctica.
- **libncurses-dev**, este paquete tiene herramientas para poder hacer interfaces de usuario basadas en texto como las que se usan en el comando make menuconfig.
- **bison**, este paquete sirve para generar parsers que usan LALR(1), que es utilizado en la compilación del kernel para poder interpretar el código de la configuración del kernel.
- **flex**, este paquete sirve para generar lexers, lo cuales son usados junto con **bison** para la compilación del kernel.
- **libssl-dev**, este paquete sirve para el desarrollo de OpenSSL, el cual es utilizado para la seguridad y cifrado relacionado al kernel.
- **libelf-dev**, este paquete sirve para poder manipular archivos ELF (el formato estándar para los archivos ejecutables y de objetos en Linux).

Preguntas

1. ¿En qué directorio se instalan los módulos del kernel?

Los módulos se encuentran en `/lib/modules` o `/usr/lib/modules`, por lo general se guardan en `/lib/modules/$(uname -r)/kernel/`, también se van agrupando o guardando en subdirectorios dependiendo de su función por ejemplo esta `/lib/modules/$(uname -r)/kernel/drivers/` donde se guardan módulos relacionados a módulos de drivers de hardware o `/lib/modules/$(uname -r)/kernel/fs/` para módulos relacionados a sistemas de archivos.

2. ¿Qué hace el comando `make oldconfig`?

Sirve para actualizar el archivo de configuración del kernel de Linux con las nuevas características que tiene el nuevo kernel, lo que pasa es que cuando cambiamos a una nueva versión del kernel es posible que existan nuevas opciones de configuración entonces este comando mantiene la configuración existente y solo pregunta sobre configuraciones nuevas.

3. ¿Cuáles son las características nuevas que implementa la versión del kernel que instalaste?

Algunas de las nuevas cosas del kernel 6.13.0 es que tiene un nuevo modelo de preempción perezosa (lazy preemption) el cual es un modelo de preempción más eficiente que equilibra la capacidad de respuesta del sistema con el rendimiento, también los sistemas de archivos XFS y Ext4 ahora admiten escrituras atómicas, mejoras en la virtualización para la arquitectura ARM64, mejoras en el rendimiento de la gestión de las redes y dispositivos, entre otras mejoras, pero pensamos que esas son algunas de las más importantes.

Referencias

Las referencias que se usaron fueron las siguientes:

- ada archivos -- Portal TIC UNAM. (s. f.). Portal TIC UNAM. <https://www.tic.unam.mx/tag/ada/>
- Debian -- Details of package build-essential in sid. (2025). Debian.org. <https://packages.debian.org/sid/build-essential>
- Debian -- Details of package libncurses-dev in sid. (2025). Debian.org. <https://packages.debian.org/sid/libncurses-dev>
- Debian -- Details of package bison in sid. (2025). Debian.org. <https://packages.debian.org/sid/bison>

- Debian -- Details of package flex in sid. (2025). Debian.org. <https://packages.debian.org/sid/flex>
- Debian -- Details of package libssl-dev in sid. (2025). Debian.org. <https://packages.debian.org/sid/libssl-dev>
- Debian -- Details of package libelf-dev in sid. (2025). Debian.org. <https://packages.debian.org/sid/libelf-dev>
- Wikipedia Contributors. (2025, February 1). Loadable kernel module. Wikipedia; Wikimedia Foundation.
- 3.8. Carga de módulos del kernel en tiempo de ejecución del sistema | Red Hat Product Documentation. (2024). Redhat.com. https://docs.redhat.com/es/documentation/red_hat_enterprise_linux/8/html/managing_monitoring_and_updating_the_kernel/loading-kernel-modules-at-system-runtime_managing-kernel-modules
- Seguí, D. (2020, March 31). Actualizando el Kernel de Linux (Parte II) - SeguiNET. SeguiNET. <https://www.seguinet.es/actualizando-el-kernel-de-linux-parte-ii/>
- Rohan Timalina. (2024, December 18). Linux Kernel 6.13 RC1 and RC2: New Features and Key Highlights. TuxCare. <https://tuxcare.com/blog/linux-kernel-6-13-rc1-and-rc2-new-features-and-key-highlights/>
- Linux_6.13 - Linux Kernel Newbies. (2025). Kernelnewbies.org. https://kernelnewbies.org/Linux_6.13
- Borisov, B. (2025, January 20). Linux Kernel 6.13 Released, Here's What's New. Linuxiac. <https://linuxiac.com/linux-kernel-6-13-released/>

Conclusión, comentarios y observaciones

Usar las computadoras del laboratorio para la práctica es algo pesado y complicado, ya que no todos los equipos funcionan, por lo que esto nos puede retrasar en la realización de la práctica, principalmente en esta donde se realiza un proceso algo tardado.

Otro aspecto no tan bueno de la práctica es que en clase no se explicaron las posibles configuraciones que podemos elegir para el kernel, entonces puede pasar (y pasó) que compilamos el kernel con alguna combinación de opciones de configuración, la cual causa que al intentar iniciar el equipo con el nuevo kernel no funcionen del todo bien y tengamos problemas de kernel panic, entonces decir cuáles elegir podría ser útil.