

¿Qué es un módulo del kernel?

Un **módulo del kernel de Linux** es una extensión dinámica del núcleo del sistema operativo. Estos módulos permiten añadir funcionalidades al kernel sin necesidad de recompilarlo ni reiniciar el sistema.

Características:

- **Dinámicos** : Se cargan/descargan en tiempo de ejecución.
 - **Modulares** : Mantienen el kernel base ligero.
 - **Especializados** : Drivers, sistemas de archivos, protocolos, etc.
 - **Escritos en C** : Requieren herramientas de desarrollo y headers del kernel.
-

Procedimiento básico: Listado, info, carga y descarga (usando solo modprobe)

Acción	Comando
Listar módulos cargados	<code>lsmod</code>
Información de un módulo	<code>modinfo <nombre_modulo></code>
Cargar un módulo	<code>sudo modprobe <nombre_modulo></code>
Descargar un módulo	<code>sudo modprobe -r <nombre_modulo></code>
Mostrar dependencias	<code>modprobe --show-depends <nombre_modulo></code>

✓ Usar `modprobe` es preferible porque gestiona automáticamente las dependencias entre módulos y respeta las reglas definidas en `/etc/modprobe.d/`.

Instalación de herramientas de desarrollo

En Red Hat / Rocky Linux 9.x

```
sudo dnf groupinstall "Development Tools"
sudo dnf install kernel-devel kernel-headers
```

En Debian / Ubuntu Server 24.04

```
sudo apt update
sudo apt install build-essential linux-headers-$(uname -r)
```



Ejercicio paso a paso: Crear, compilar e instalar un módulo del kernel usando únicamente modprobe

Paso 1: Crear el código fuente en C

Crema un archivo llamado `hola_mundo.c`:

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Tu Nombre");
MODULE_DESCRIPTION("Un modulo simple del kernel que imprime un mensaje.");
MODULE_VERSION("1.0");

static int __init hola_inicio(void) {
    printk(KERN_INFO "Hola Mundo! Modulo cargado.\n");
    return 0;
}

static void __exit hola_salida(void) {
    printk(KERN_INFO "Adiós Mundo! Modulo descargado.\n");
}

module_init(hola_inicio);
module_exit(hola_salida);
```

Paso 2: Crear el archivo Makefile

Crema un archivo `Makefile` en el mismo directorio:

```
obj-m += hola_mundo.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean

install:
    sudo cp hola_mundo.ko /lib/modules/$(shell uname -r)/extra/
    sudo depmod -a
```

```
uninstall:  
sudo rm /lib/modules/$(shell uname -r)/extra/hola_mundo.ko  
sudo depmod -a
```

OJO!!!! Crear el directorio extra en la ruta (no existe)
sudo mkdir -p /lib/modules/\$(uname -r)/extra

Paso 3: Compilar el módulo

```
make
```

Se generará el archivo `hola_mundo.ko`.

Paso 4: Instalar el módulo en el sistema

```
sudo make install
```

Este comando copia el módulo `.ko` a la carpeta de módulos del kernel y actualiza la base de datos de dependencias con `depmod`.

Paso 5: Cargar el módulo usando modprobe

```
sudo modprobe hola_mundo
```

Verifica los mensajes del kernel:

```
dmesg | tail -20
```

Deberías ver algo como:

```
[XXXXXX.XXXXXX] Hola Mundo! Modulo cargado.
```

Paso 6: Verificar si está cargado

```
lsmod | grep hola_mundo
```

Paso 7: Descargar el módulo usando modprobe

```
sudo modprobe -r hola_mundo
```

Verifica nuevamente con:

```
dmesg | tail -20
```

Aparecerá:

```
[XXXXXX.XXXXXX] Adiós Mundo! Modulo descargado.
```

Paso 8: Limpiar (opcional)

```
make clean
```

Y si deseas desinstalarlo permanentemente:

```
sudo make uninstall
```



Resumen de herramientas instaladas por distribución

Distribución	Comandos para instalar herramientas
Rocky Linux	<pre>dnf groupinstall "Development Tools"+dnf install kernel-devel kernel-headers</pre>
Ubuntu Server 24.04	<pre>apt install build-essential linux-headers-\$(uname -r)</pre>