

Módulos

Los módulos se asocian al **kernel** (Linux).

Son programas que se cargan en memoria (en '*kernel-space*') para que el SO pueda **administrar** el sistema (gestionar el procesador, la memoria, el fs, y las I/O) [1].

Los módulos fundamentales se cargan de una en memoria (integrados o **built-in**), y otros se cargan bajo demanda (**dinámicamente**), como los drivers de los dispositivos de E/S [2].

Estos módulos son parte del conjunto de programas del kernel [3].

Los archivos que representan módulos del kernel suelen ser de la extensión '**.ko**' [4].

```

root@porteus:/home/guest# more /lib/modules/$(uname -r)/modules.builtin
kernel/arch/x86/events/intel/intel-uncore.ko
kernel/arch/x86/platform/intel/iosf_mbi.ko
kernel/mm/zsmalloc.ko
kernel/fs/binfmt_script.ko
kernel/fs/mbcache.ko
kernel/fs/aufs/aufs.ko
kernel/fs/crypto/fscrypto.ko
kernel/fs/exportfs/exportfs.ko
kernel/fs/ext4/ext4.ko
kernel/fs/fat/fat.ko
kernel/fs/fat/vfat.ko
kernel/fs/fuse/fuse.ko
kernel/fs/isofs/isofs.ko
kernel/fs/jbd2/jbd2.ko
kernel/fs/nls/nls_base.ko
kernel/fs/nls/nls_cp437.ko
kernel/fs/nls/nls_iso8859-1.ko
kernel/fs/nls/nls_utf8.ko
kernel/fs/reiserfs/reiserfs.ko
kernel/fs/squashfs/squashfs.ko
kernel/fs/xfs/xfs.ko
kernel/security/keys/encrypted-keys/encrypted-keys.ko
kernel/crypto/crypto.ko
kernel/crypto/crypto_wq.ko
kernel/crypto/crypto_algapi.ko
kernel/crypto/aead.ko
kernel/crypto/crypto_blkcipher.ko
kernel/crypto/seqiv.ko
kernel/crypto/crypto_hash.ko
kernel/crypto/akcipher.ko
kernel/crypto/kpp.ko
kernel/crypto/cryptomgr.ko
kernel/crypto/hmac.ko
kernel/crypto/crypto_null.ko
kernel/crypto/sha256_generic.ko
kernel/crypto/gf128mul.ko
kernel/crypto/ecb.ko

root@porteus:/home/guest# lsmod
Module                  Size  Used by
ip6v6                   228313  18
btusb                   22658  0
btrtl                   4032  1 btusb
btbcm                   4928  1 btusb
btintel                 5184  1 btusb
bluetooth               225081  5 btrtl,btintel,btcm,btusb
uvcvideo                55256  0
videobuf2_vmalloc       2880  1 uvcvideo
videobuf2_memops        1216  1 videobuf2_vmalloc
videobuf2_v4l2          7304  1 uvcvideo
videobuf2_core          18948  2 uvcvideo,videobuf2_v4l2
wl                      6225308  0
asus_nb_wmi             14000  0
asus_wmi                12545  1 asus_nb_wmi
sparse_keymap           2048  1 asus_wmi
snd_hda_codec_hdmi      23937  1
intel_rapl              7776  0
coretemp                4740  0
kvm_intel               143484  0
kvm                     226297  1 kvm_intel
irqbypass              2176  1 kvm
crct10dif_pclmul        3264  0
crc32_pclmul            2048  0
crc32c_intel            7296  0
ghash_clmulni_intel     2944  0
aesni_intel            138904  0
aes_x86_64              7040  1 aesni_intel
lrw                     2816  1 aesni_intel
glue_helper             3456  1 aesni_intel
ablk_helper             1472  1 aesni_intel
cryptd                  4872  3 ablk_helper,ghash_clmulni_intel,aesni_intel
intel_cstate            4201  0
snd_hda_codec_realtek   47824  1
snd_hda_codec_generic   36288  1 snd_hda_codec_realtek
intel_rapl_perf         5088  0

```

modulos-integrados-kernel

modulos-dinamicos-kernel

Nota: en Porteus (que es la distribución con la que estuve investigando todo esto), le llaman módulos a los archivos '**.xzm**'. Estos archivos son como .RAR, o sea, archivos comprimidos. Dentro de estos .xzm puede haber cualquier cosa, módulos del kernel, el contenido de un paquete, o archivos nuestros [5].

Links útiles Módulos:**[1]**

- <https://web.mit.edu/rhel-doc/3/rhel-sag-es-3/ch-kernel-modules.html>
- <http://www.it.uc3m.es/amarin/curs/lao/node59.html>
- https://linux-kernel-labs.github.io/refs/heads/master/labs/kernel_modules.html
- <https://tldp.org/LDP/lkmpg/2.6/lkmpg.pdf>
- <https://unix.stackexchange.com/questions/47208/what-is-the-difference-between-kernel-drivers-and-kernel-modules>

[2]

- [Difference-between-linux-loadable-and-built-in-modules](#)
- <https://tldp.org/HOWTO/Module-HOWTO/x73.html> (**una joyita**)
- <https://www.xmodulo.com/find-information-builtin-kernel-modules-linux.html> (comandos)
- <https://www.networkworld.com/article/3391362/looking-into-linux-modules.html> (comandos)

[3]

<https://boxmatrix.info/wiki/Kernel-Modules> *(no necesariamente el kernel Linux de una distribución tendrá todos estos módulos; es un listado de todos los módulos que podría usar)*

[4]

<https://fileinfo.com/extension/ko>

[5]

[Porteus - What is a xzm linux kernel module?](#)

Paquetes

Los paquetes representan a un **programa** o a un conjunto de **utilidades**. En su interior están todos los archivos compilados para poder ejecutar el programa **[1]**.

Estos archivos se ubican en diferentes partes del filesystem. El **gestor de paquetes** de cada distribución es quien se encarga de 'abrir' los paquetes y ubicar los archivos donde correspondan, para que así quede 'instalado' en el sistema la nueva función/programa *(o también podríamos llegar a hacerlo nosotros manualmente)* **[2]**.

Hay grandes **servidores/repositorios** de paquetes, de donde podemos descargar paquetes. Por ejemplo, el de Porteus es <https://slackware.pkgs.org/>. O desde el gestor de paquetes de la distribución, podemos solicitar que haga una descarga (se conecta al servidor de paquetes y lo descarga e instala en nuestra computadora) **[3]**.

Están destinados a que los usen los **usuarios**, y no el kernel. Es decir, el usuario es quien solicita ejecutar estos programas, por lo que se cargan en memoria en 'user-space'.

*(de todas formas muchos de estos programas necesitan modo kernel para ejecutarse (su/root))
(igual no descarto que pueda llegar a haber paquetes que solo usa el kernel... no se bien).*

Los paquetes se suelen asociar con el **Sistema Operativo** como tal, y no tanto con el kernel. O sea, a grandes rasgos, el SO = **kernel** + **entorno grafico** (si lo hay) + **otros_programas**. Entonces los **'otros_programas'** vienen en paquetes.

Hay **paquetes** más **'técnicos'** (implementación de los comandos de consola) y paquetes que representan **aplicaciones** (como Firefox, VLC, etc).

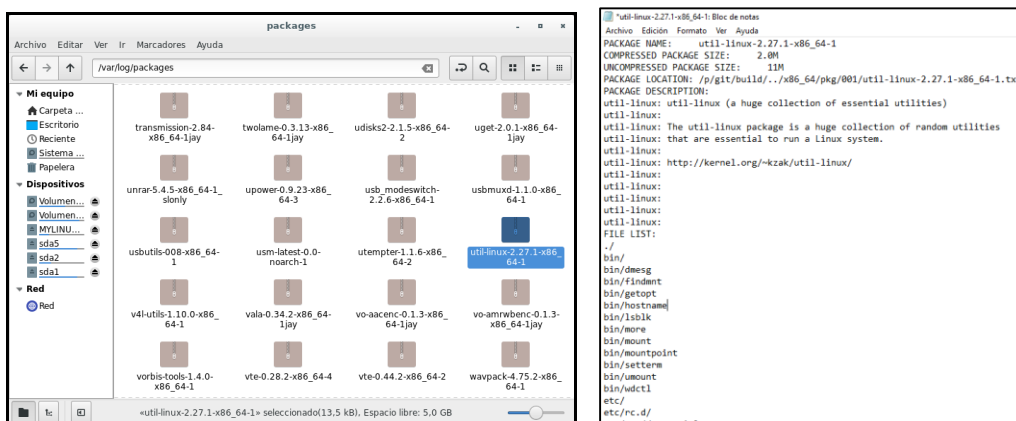
- ➔ Las distribuciones de Linux derivan del sistema operativo GNU/Linux (GNU es el SO y Linux su Kernel). Por ende, hay varios **paquetes 'base'** que vienen con **GNU**, como **'coreutils'** que implementa muchos comandos básicos de consola, como mkdir, ls, etc (*GNU es un gran proyecto de código abierto, en el cual integró muchos de sus paquetes para conformar un SO*). De todas formas, recordemos que cada distribución es libre de sacar algunos de esos paquetes GNU, o agregar paquetes propios de la distro que contengan implementaciones de paquetes GNU [4].

También contamos con **paquetes** más de **Linux** que de GNU, como **'e2fsprogs'** (que implementa comandos como mke2fs y e2fsck) y **'util-linux'** (que implementa comandos como mkfs, fsck, mount y su). Pero lo que no estoy seguro, es que si estos 2 paquetes vienen dentro del kernel (Linux) o están dentro de los **'otros_programas'**. Los repos de estos 2 paquetes están en el gran repo de Linux (<https://git.kernel.org/>), así que supongo que sí pero no se... porque también es posible descargar estos paquetes desde los repositorios de las distribuciones por si no llegaron a venir incluidos [5].

- ➔ Y por último, están los **paquetes** que nos agrega la **distribución**, ya sean funcionalidades propias de la distro, como aplicaciones (editores de texto, navegadores, calculadora, etc).

Las **extensiones** de los **archivos** que representan a los **paquetes**, varían según la distribución que tengamos. Por ejemplo, acá en Porteus, los paquetes vienen dentro de los **.txz** (distinto a los **.xzm**, pero pueden convertirse a ellos). En Debian **.deb** (creo). Etc.

En `/var/log/packages` podemos ver los paquetes que tenemos instalados. Igual esos no son los paquetes en sí, sino que son una especie de 'descriptor' de los paquetes (si los abrimos con el bloc de notas, veremos que nos dice toda la info sobre el paquete y dónde están los archivos del mismo).



Links útiles Paquetes:**[1]**

- https://en.wikipedia.org/wiki/Package_manager
- <https://ricardovargas.me/es/bitacora-web/articulos/item/que-son-manejadores-de-paquetes-package-manager-en-linux>

[2]

- <https://forum.porteus.org/viewtopic.php?t=6264> (Instalar y desinstalar aplicaciones en Porteus + **Info muy piola** en general sobre los paquetes)

[3]

- <http://www.porteus.org/tutorials/9-modules/149-usm.html> (Gestor de paquetes de Porteus)
- <http://www.porteus.org/tutorials/9-modules.html> (Descargar un paquete manualmente en Porteus)

[4]

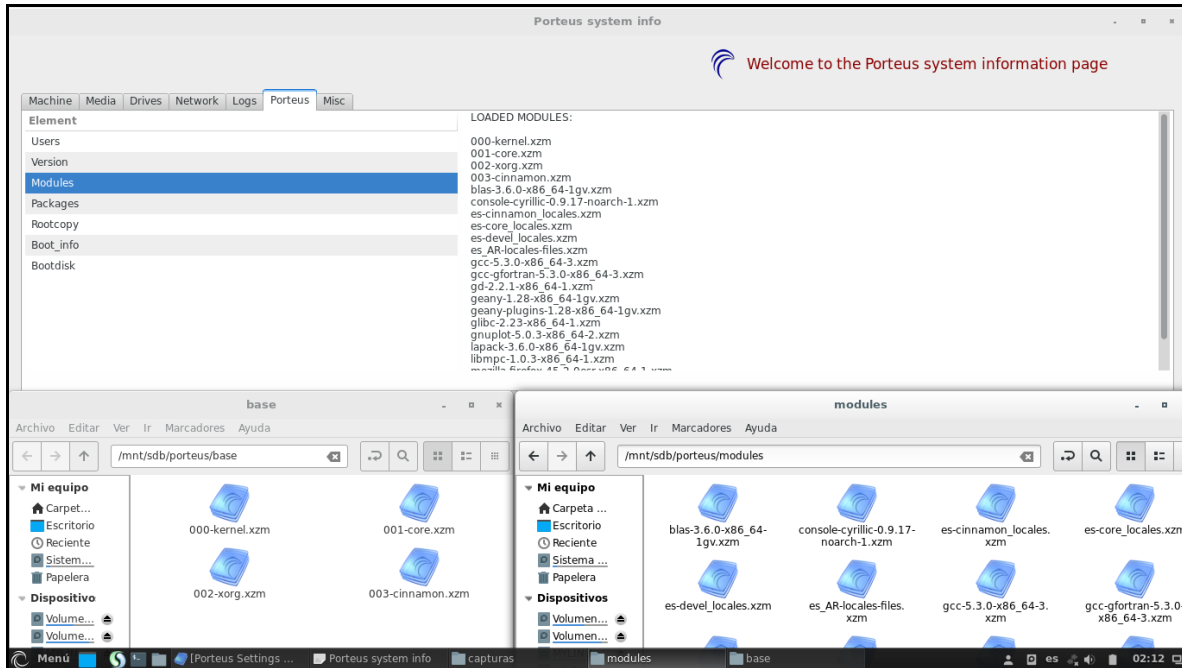
- <https://en.wikipedia.org/wiki/GNU>
- <https://es.wikipedia.org/wiki/GNU/Linux>
- https://en.wikipedia.org/wiki/List_of_GNU_packages

[5]

- <https://en.wikipedia.org/wiki/E2fsprogs>
- <https://en.wikipedia.org/wiki/Util-linux>
- <https://unix.stackexchange.com/questions/278522/mkfs-ext4-command-not-found-in-debian-jessie>

Extras

1. Porteus trae varios módulos que se ‘activan’ cuando iniciamos el Sistema (recordemos que los ‘módulos’ de Porteus son como archivos comprimidos, y el proceso de ‘activar’ sería como descomprimirlos en la raíz ‘/’ del filesystem).



En `/base` están los módulos (`.xzm`) que conforman al sistema operativo como tal: dentro de ‘kernel’ está todo lo referido a Linux, como los módulos built-in, drivers, etc (o sea, son archivos que el kernel requerirá en algún momento, pero el proceso inicial del kernel se crea en tiempo de booteo, al cargarse en memoria el archivo [vmlinuz](#) ubicado en `/boot`); dentro de ‘core’ hay entre otras cosas, todos los paquetes que vienen con la distribución, o sea, paquetes base de GNU + paquetes propios de Porteus + paquetes de aplicaciones como un navegador web, etc; y en ‘xorg’ y ‘cinnamon’ hay cosas referidas al entorno gráfico.

En `/modules` agregamos los módulos (`.xzm`) que también queremos que se activen al iniciar Porteus. En este caso, son módulos que nos agregaron los profes de análisis numérico.

2. Página recomendada entre otras cosas para ver la documentación de comandos Linux:

<https://man7.org/linux/man-pages/man1/su.1.html> (ejemplo de la docu de ‘su’)

3. Diferencia entre mkfs.ext2 y mke2fs:

Básicamente mkfs.ext2 / mkfs.ext3 / mkfs.ext4 es como que invoca a mke2fs con ciertos parámetros para formatear tal filesystem. De todas formas, mkfs.ext2 no es una implementación o algo así del paquete 'util-linux', es un 'symbolic link' que igual no se bien de qué paquete sale je (en 'util-linux' solo estan mkfs.bfs, mkfs.cramfs y mkfs.minix).

- Paquete 'util-linux':
https://slackware.pkgs.org/current/slackware-x86_64/util-linux-2.37.1-x86_64-1.txz.html
- Paquete 'e2fsprogs':
https://slackware.pkgs.org/current/slackware-x86_64/e2fsprogs-1.46.3-x86_64-1.txz.html

Docu mkfs: <https://man7.org/linux/man-pages/man8/mkfs.8.html>

Docu mke2fs: <https://man7.org/linux/man-pages/man8/mke2fs.8.html>

Y por lo que entiendo, hay muchos tipos de comandos mkfs.*, que lo que hacen es invocar al programa que crea cierto filesystem (formatea). Por ejemplo, mkfs.ntfs invoca al comando mkntfs perteneciente al driver del filesystem NTFS (Paquete 'ntfs-3g' que antes se llamaba 'ntfsprogs' https://slackware.pkgs.org/current/slackware-x86_64/ntfs-3g-2017.3.23-x86_64-4.txz.html) (de todas formas, el driver oficial de ntfs en linux es [este](#), que fue escrito originalmente por [The Linux-NTFS Project](#), pero creo que ahora también lo lleva adelante junto con [Tuxera](#), la empresa de 'ntfs-3g', o sea por un lado está el driver incluido en el kernel [módulo built-in] y por otro está 'ntfs-3g', que es un driver [FUSE](#). Esto está bien explicado en la *joyita*).

- (ntfs) <https://unix.stackexchange.com/questions/503325/how-can-i-format-a-partition-into-a-filesystem-quickly>
- (ntfs) [man/8/mkntfs](#)
- (ntfs) <https://www.enmimaquinafunciona.com/pregunta/77679/como-instalo-ntfs-3g-ntfsprogs-manualmente-en-ubuntu>
- (ntfs) <https://arstechnica.com/gadgets/2021/08/paragon-is-working-to-get-its-ntfs3-filesystem-into-the-linux-kernel/> (*joyita*)



AiwendilH · 3a · editado 3a

mkfs is part of the util-linux package. Source-code packages are [here](#), browseable git repository [here](#), mkfs in specific is [here](#).

I wouldn't trust the github mirrors are the moment...it seems at least the kernel mirror on github is a bit out of sync.

Edit: But probably this is not what you are looking for. The mkfs program is just a wrapper to call the appropriate mkfs.<filesystem> executable..and those are maintained usually by the specific filesystem tools.

13 ↓ Compartir ...

https://www.reddit.com/r/linux/comments/9qcf9o/how_do_i_find_the_source_code_of_linux_for_the/

Más links:

- <https://forums.gentoo.org/viewtopic-t-1009142-start-0.html>
- <https://www.linuxquestions.org/questions/linux-newbie-8/technical-difference-between-mkfs-and-mke2fs-4175580239/>
- <https://stackoverflow.com/questions/5221071/mkfs-for-a-new-file-system>
- <https://www.thegeekstuff.com/2013/01/mke2fs-examples/>

4. Links extras:

- <https://blog.carreralinux.com.ar/2018/01/uso-de-depmod-dependencias-modulos/>
- <https://man7.org/linux/man-pages/man8/depmod.8.html>
- <https://www.factor.mx/portal/base-de-conocimiento/los-modulos-kernel/> (Comandos módulos dinámicos kernel)
- <https://medium.com/@admin101/linux-el-directorio-sbin-bf871a72462a>
- <https://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/sbin.html>
- <https://eslinux.com/foro/1542/diferencia-sbin-bin/>
- <https://forum.porteus.org/viewtopic.php?t=1389> (Planteamiento muy interesante sobre las **zonas horarias** - Porteus)
- <https://forum.porteus.org/viewtopic.php?t=9023> (Date and Time - Porteus)