

# La commande modprobe

- Objet : La commande modprobe
- Niveau requis :  
[débutant, avisé](#)
- Commentaires : *charger ou télécharger un module*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :  
[à-placer](#)
  - Création par [mattux](#) le 17/06/2007
  - Testé et complété par lagrenouille.... le 03/05/2023....
- Commentaires sur le forum : [C'est ici<sup>1\)</sup>](#)

## Présentation

La commande modprobe permet de charger ou de télécharger un module.

Les modules sont un morceau de code qui étend les fonctionnalités du noyau Linux

Les modules sont des bibliothèques que l'on peut charger dans le noyau lorsque celui-ci a besoin d'une certaine fonctionnalité.



Normalement, les installations automatisent la création de tels fichiers et vous n'avez rien à faire.

Mais par exemple pour l'installation d'un WIFI vous allez devoir utiliser cette commande.

- Syntaxe

```
modprobe <nom module>
```

la commande **lsmod** liste tous les modules

```
lsmod
Module                Size  Used by
btrfs                 1589248 0
blake2b_generic        20480 0
xor                   24576 1 btrfs
raid6_pq              122880 1 btrfs
ufs                   94208 0
qnx4                   16384 0
hfsplus               126976 0
hfs                    73728 0
minix                  45056 0
msdos                   20480 0
jfs                    212992 0
```

xfs	1802240	0	
dm_mod	163840	0	
cpuid	16384	0	
nf_tables	253952	0	
tcp_diag	16384	0	
udp_diag	16384	0	
inet_diag	24576	2	tcp_diag,udp_diag
binfmt_misc	24576	1	
battery	24576	0	
nls_ascii	16384	0	
nls_cp437	20480	0	
vfat	20480	0	
fat	86016	2	msdos,vfat
uas	32768	1	
usb_storage	81920	1	uas
nfnetlink	20480	1	nf_tables
8021q	40960	0	
garp	16384	1	8021q
stp	16384	1	garp
mrp	20480	1	8021q
llc	16384	2	stp,garp
iptables_nat	16384	0	
nf_nat	57344	1	iptables_nat
nf_conntrack	176128	1	nf_nat
nf_defrag_ipv6	24576	1	nf_conntrack
nf_defrag_ipv4	16384	1	nf_conntrack
libcrc32c	16384	5	nf_conntrack,nf_nat,btrfs,nf_tables,xfs
iptables_mangle	16384	0	
iptables_filter	16384	0	
bluetooth	757760	2	
jitterentropy_rng	16384	1	
drbg	40960	1	
ansi_cprng	16384	0	
ecdh_generic	16384	1	bluetooth
rftkill	32768	3	bluetooth
ecc	36864	1	ecdh_generic
snd_hda_codec_hdmi	73728	1	
snd_hda_codec_realtek	167936	1	
intel_rapl_msr	20480	0	
intel_rapl_common	28672	1	intel_rapl_msr
snd_hda_codec_generic	98304	1	snd_hda_codec_realtek
snd_sof_pci	24576	0	
snd_sof_intel_byt	24576	1	snd_sof_pci
x86_pkg_temp_thermal	20480	0	
snd_sof_intel_ipc	20480	1	snd_sof_intel_byt
intel_powerclamp	20480	0	
snd_sof_intel_hda_common	102400	1	snd_sof_pci
coretemp	20480	0	
snd_sof_xtensa_dsp	16384	2	snd_sof_intel_hda_common,snd_sof_intel_byt
snd_sof	139264	4	
snd_sof_pci,snd_sof_intel_hda_common,snd_sof_intel_byt,snd_sof_intel_ipc			

```

kvm_intel          327680  0
snd_sof_intel_hda  20480  1 snd_sof_intel_hda_common
snd_soc_hdac_hda   24576  1 snd_sof_intel_hda_common
kvm                937984  1 kvm_intel
snd_hda_ext_core   36864  3
snd_sof_intel_hda_common,snd_soc_hdac_hda,snd_sof_intel_hda
snd_soc_acpi_intel_match 45056  2 snd_sof_pci,snd_sof_intel_hda_common
snd_soc_acpi       16384  3
snd_soc_acpi_intel_match,snd_sof_intel_hda_common,snd_sof_intel_byt
ledtrig_audio      16384  2 snd_hda_codec_generic,snd_sof
snd_hda_intel      57344  2
irqbypass          16384  1 kvm
snd_intel_dspcfg    28672  3
snd_hda_intel,snd_sof_pci,snd_sof_intel_hda_common
soundwire_intel     45056  2 snd_sof_intel_hda_common,snd_intel_dspcfg
soundwire_generic_allocation 16384  1 soundwire_intel
ghash_clmulni_intel 16384  0
snd_soc_core        319488  4
soundwire_intel,snd_sof,snd_sof_intel_hda_common,snd_soc_hdac_hda
snd_compress        32768  1 snd_soc_core
soundwire_cadence   36864  1 soundwire_intel
aesni_intel         372736  0
snd_hda_codec       176128  5
snd_hda_codec_generic,snd_hda_codec_hdmi,snd_hda_intel,snd_hda_codec_realtek
,snd_soc_hdac_hda
libaes              16384  2 bluetooth,aesni_intel
crypto_simd         16384  1 aesni_intel
cryptd              24576  2 crypto_simd,ghash_clmulni_intel
mei_hdcp            24576  0
glue_helper         16384  1 aesni_intel
snd_hda_core        110592  9
snd_hda_codec_generic,snd_hda_codec_hdmi,snd_hda_intel,snd_hda_ext_core,snd_
hda_codec,snd_hda_codec_realtek,snd_sof_intel_hda_common,snd_soc_hdac_hda,snd
d_sof_intel_hda
rapl                20480  0
snd_hwdep           20480  1 snd_hda_codec
soundwire_bus       94208  3
soundwire_intel,soundwire_generic_allocation,soundwire_cadence
intel_cstate        20480  0
snd_pcm             143360  9
snd_hda_codec_hdmi,snd_hda_intel,snd_hda_codec,soundwire_intel,snd_sof,snd_s
of_intel_hda_common,snd_compress,snd_soc_core,snd_hda_core
intel_uncore        180224  0
snd_timer           49152  1 snd_pcm
joydev              28672  0
iTCO_wdt            16384  0
mei_me              45056  1
snd                 110592  14
snd_hda_codec_generic,snd_hda_codec_hdmi,snd_hwdep,snd_hda_intel,snd_hda_cod
ec,snd_hda_codec_realtek,snd_timer,snd_compress,snd_soc_core,snd_pcm
intel_pmc_bxt       16384  1 iTCO_wdt

```

iTCO_vendor_support	16384	1	iTCO_wdt
wmi_bmf	16384	0	
pcspkr	16384	0	
mei	139264	3	mei_hdcp,mei_me
watchdog	32768	1	iTCO_wdt
soundcore	16384	1	snd
tpm_crb	20480	0	
ee1004	20480	0	
tpm_tis	16384	0	
sg	36864	0	
tpm_tis_core	28672	1	tpm_tis
tpm	73728	3	tpm_tis,tpm_crb,tpm_tis_core
intel_pmc_core	45056	0	
rng_core	16384	1	tpm
evdev	28672	15	
acpi_pad	184320	0	
acpi_tad	20480	0	
ib_iser	49152	0	
rdma_cm	131072	1	ib_iser
iw_cm	57344	1	rdma_cm
ib_cm	135168	1	rdma_cm
ib_core	413696	4	rdma_cm,iw_cm,ib_iser,ib_cm
iscsi_tcp	28672	0	
libiscsi_tcp	36864	1	iscsi_tcp
libiscsi	73728	3	libiscsi_tcp,iscsi_tcp,ib_iser
scsi_transport_iscsi	143360	5	libiscsi_tcp,iscsi_tcp,ib_iser,libiscsi
parport_pc	40960	0	
ppdev	24576	0	
lp	20480	0	
parport	73728	3	parport_pc,lp,ppdev
fuse	167936	7	
configfs	57344	2	rdma_cm
ip_tables	36864	3	iptables_filter,iptables_nat,iptables_mangle
x_tables	53248	3	iptables_filter,ip_tables,iptables_mangle
autofs4	53248	2	
ext4	942080	3	
crc16	16384	2	bluetooth,ext4
mbcache	16384	1	ext4
jbd2	151552	1	ext4
crc32c_generic	16384	0	
hid_logitech_hidpp	49152	0	
hid_logitech_dj	28672	0	
hid_generic	16384	0	
usbhid	65536	1	hid_logitech_dj
hid	151552	4	
usbhid,hid_generic,hid_logitech_dj,hid_logitech_hidpp			
i915	2777088	18	
sr_mod	28672	0	
sd_mod	61440	6	
cdrom	73728	3	hfsplus,hfs,sr_mod
i2c_algo_bit	16384	1	i915

```

drm_kms_helper      278528  1 i915
nvme                 49152  2
xhci_pci            24576  0
nvme_core           131072  4 nvme
xhci_hcd            311296  1 xhci_pci
ahci                 40960  2
cec                 61440  2 drm_kms_helper,i915
libahci             45056  1 ahci
t10_pi              16384  2 sd_mod,nvme_core
crc_t10dif          20480  1 t10_pi
libata              299008  2 libahci,ahci
drm                 630784  9 drm_kms_helper,i915
r8169               102400  0
usbcore             331776  5 xhci_hcd,usbhid,usb_storage,xhci_pci,uas
realtek             24576  1
mdio_devres         16384  1 r8169
scsi_mod            270336  10
sd_mod,scsi_transport_iscsi,usb_storage,iscsi_tcp,ib_iser,uas,libiscsi,libat
a,sg,sr_mod
libphy              155648  3 r8169,mdio_devres,realtek
crc32_pclmul        16384  0
crc32c_intel        24576  7
i2c_i801            32768  0
crct10dif_generic   16384  0
crct10dif_pclmul    16384  1
i2c_smbus           20480  1 i2c_i801
crct10dif_common    16384  3
crct10dif_generic,crc_t10dif,crct10dif_pclmul
usb_common          16384  2 xhci_hcd,usbcore
wmi                 36864  1 wmi_bmf
fan                 20480  0
video               61440  1 i915
button              24576  0

```

le nom - la taille - liste de modules séparés par une virgule

les modules de kernel Les modules correspondent à des drivers que l'on peut charger et décharger dynamiquement de la mémoire. Vous trouverez le ou les modules dans le répertoire/lib/modules/(version kernel)

```

ls -lha /lib/modules/5.10.0-21-amd64/
kernel/                modules.dep
modules.alias          modules.dep.bin
modules.alias.bin      modules.devname
modules.builtin        modules.order
modules.builtin.alias.bin modules.softdep
modules.builtin.bin    modules.symbols
modules.builtin.modinfo modules.symbols.bin

```

```

ls -lha /lib/modprobe.d/
total 28K

```

```
drwxr-xr-x  2 root root 4,0K 30 avril 18:55 .
drwxr-xr-x 130 root root 12K 30 avril 18:55 ..
-rw-r--r--  1 root root  655  8 janv. 2021 aliases.conf
-rw-r--r--  1 root root  390 22 déc. 12:55 fbdev-blacklist.conf
-rw-r--r--  1 root root  773  2 févr. 2021 systemd.conf
```

```
ls -lha /lib/modules/5.10.0-21-amd64/kernel/
total 48K
drwxr-xr-x 12 root root 4,0K 23 janv. 23:02 .
drwxr-xr-x  3 root root 4,0K 23 janv. 23:02 ..
drwxr-xr-x  3 root root 4,0K 23 janv. 23:02 arch
drwxr-xr-x  2 root root 4,0K 23 janv. 23:02 block
drwxr-xr-x  4 root root 4,0K 23 janv. 23:02 crypto
drwxr-xr-x 79 root root 4,0K 23 janv. 23:02 drivers
drwxr-xr-x 60 root root 4,0K 23 janv. 23:02 fs
drwxr-xr-x  7 root root 4,0K 23 janv. 23:02 lib
drwxr-xr-x  2 root root 4,0K 23 janv. 23:02 mm
drwxr-xr-x 53 root root 4,0K 23 janv. 23:02 net
drwxr-xr-x 15 root root 4,0K 23 janv. 23:02 sound
drwxr-xr-x  3 root root 4,0K 23 janv. 23:02 virt
```

Si vous avez installé “tree” les répertoires et les modules du noyau sont répertoriés sous forme d'arborescence.

ci dessous un extrait de la commande :

```
tree /lib/modules/5.10.0-21-amd64/kernel/
/lib/modules/5.10.0-21-amd64/kernel/
├── arch
│   └── x86
│       ├── crypto
│       │   ├── aegis128-aesni.ko
│       │   ├── aesni-intel.ko
│       │   ├── blake2s-x86_64.ko
│       │   ├── blowfish-x86_64.ko
│       │   ├── camellia-aesni-avx2.ko
│       │   ├── camellia-aesni-avx-x86_64.ko
│       │   ├── camellia-x86_64.ko
│       │   ├── cast5-avx-x86_64.ko
│       │   ├── cast6-avx-x86_64.ko
│       │   ├── chacha-x86_64.ko
│       │   ├── crc32c-intel.ko
│       │   ├── crc32-pclmul.ko
│       │   ├── crct10dif-pclmul.ko
│       │   ├── curve25519-x86_64.ko
│       │   ├── des3_edc-x86_64.ko
│       │   ├── ghash-clmulni-intel.ko
│       │   ├── glue_helper.ko
│       │   ├── poly1305-x86_64.ko
│       │   ├── serpent-avx2.ko
│       │   └── serpent-avx-x86_64.ko
```

```
├── serpent-sse2-x86_64.ko
├── sha1-ssse3.ko
├── sha256-ssse3.ko
├── sha512-ssse3.ko
├── twofish-avx-x86_64.ko
├── twofish-x86_64-3way.ko
├── twofish-x86_64.ko
├── events
│   ├── amd
│   │   └── power.ko
│   ├── intel
│   │   ├── intel-cstate.ko
│   │   └── intel-uncore.ko
│   └── rapl.ko
├── kernel
│   ├── cpu
│   │   └── mce
│   │       └── mce-inject.ko
│   ├── cpuid.ko
│   └── msr.ko
├── kvm
│   ├── kvm-amd.ko
│   ├── kvm-intel.ko
│   └── kvm.ko
├── oprofile
│   └── oprofile.ko
└── block
    ├── bfq.ko
    ├── kyber-iosched.ko
    └── t10-pi.ko
```

Si vous faites un less

```
tree /lib/modules/5.10.0-21-amd64/kernel/ | less
```

vous pouvez faire une recherche par nom dans votre less

## /drivers

de quels modules dépendent les modules du noyau drm.

```
modinfo drm
```

```
filename:      /lib/modules/5.10.0-22-amd64/kernel/drivers/gpu/drm/drm.ko
license:       GPL and additional rights
description:   DRM shared core routines
author:        Gareth Hughes, Leif Delgass, José Fonseca, Jon Smirl
license:       GPL and additional rights
description:   DRM bridge infrastructure
author:        Ajay Kumar <ajaykumar.rs@samsung.com>
license:       GPL and additional rights
```

```

description:    DRM panel infrastructure
author:        Thierry Reding <treding@nvidia.com>
depends:
retpoline:     Y
intree:        Y
name:          drm
vermagic:      5.10.0-22-amd64 SMP mod_unload modversions
sig_id:        PKCS#7
signer:        Debian Secure Boot CA
sig_key:       32:A0:28:7F:84:1A:03:6F:A3:93:C1:E0:65:C4:3A:E6:B2:42:26:43
sig_hashalgo:  sha256
signature:     07:81:04:67:F4:36:E3:B8:BA:6C:35:38:E7:11:88:FE:14:9E:02:31:
               BC:EA:08:68:AB:84:B7:5F:B0:30:2A:0D:11:DF:41:03:30:F1:13:CF:
               BF:5C:3C:DA:CC:30:70:69:C3:72:B6:F0:4F:C4:E5:0E:24:74:A2:4B:
               39:36:4F:5A:38:74:5D:3F:CD:88:24:17:85:59:D7:56:B6:89:3D:23:
               E1:BF:6E:F3:65:C8:0C:9E:D5:29:57:9A:D1:AC:E1:2B:41:B8:54:3B:
               06:90:44:57:DF:D4:9F:BB:25:81:47:4E:D9:F2:1F:23:23:1D:EA:C5:
               46:93:D8:33:F9:A9:2E:C5:80:A8:9E:4D:B8:66:5B:81:A3:47:67:68:
               09:25:48:11:6D:A7:A9:C6:E2:53:7F:AD:E5:73:E2:CC:BC:BB:7E:90:
               37:0A:43:44:B5:35:ED:1B:BB:81:14:0F:2A:66:A8:6D:42:C8:66:73:
               E8:44:17:92:62:78:BA:D4:9E:EE:B9:66:D2:DD:4A:6D:C8:5F:62:E5:
               EE:8F:AF:42:4C:FC:D4:50:00:EE:74:4A:86:50:53:2C:DA:C6:23:BD:
               85:CE:67:82:AC:C8:92:02:06:8F:75:AB:2D:69:78:A2:0C:32:EE:FE:
               3B:BC:05:0F:6A:58:DD:FC:B4:89:65:87:D2:92:9C:2F
parm:          edid_firmware:Do not probe monitor, use specified EDID blob
from built-in data or /lib/firmware instead. (string)
parm:          vblankoffdelay:Delay until vblank irq auto-disable [msecs]
(0: never disable, <0: disable immediately) (int)
parm:          timestamp_precision_usec:Max. error on timestamps [usecs]
(int)
parm:          debug:Enable debug output, where each bit enables a debug
category.
               Bit 0 (0x01) will enable CORE messages (drm core code)
               Bit 1 (0x02) will enable DRIVER messages (drm controller code)
               Bit 2 (0x04) will enable KMS messages (modesetting code)
               Bit 3 (0x08) will enable PRIME messages (prime code)
               Bit 4 (0x10) will enable ATOMIC messages (atomic code)
               Bit 5 (0x20) will enable VBL messages (vblank code)
               Bit 7 (0x80) will enable LEASE messages (leasing code)
               Bit 8 (0x100) will enable DP messages (displayport code) (int)
parm:          edid_fixup:Minimum number of valid EDID header bytes (0-8,
default 6) (int)

```

Pour une raison donnée ; on peut aussi décharger un module. Pour cela, on utilise la commande `rmmod`

```
# rmmod module
```

depmod - Gestion des dépendances entre les modules du noyau Les utilitaires `depmod` et `modprobe` ont été conçus pour rendre le noyau Linux modulaire, configurable pour tous les utilisateurs, administrateurs et concepteurs de distribution. `depmod` crée un fichier de dépendances de type «



Makefile », basé sur les symboles qu'il trouve dans les modules indiqués sur la ligne de commande ou dans les répertoires indiqués par le fichier de configuration. Ce fichier est ensuite utilisé par modprobe pour charger automatiquement le bon module ou l'ensemble de modules. L'invocation normale de depmod se fait en incluant la ligne

à utiliser avec précaution

**insmod** - Charger des modules dans le noyau en cours d'utilisation. **insmod** essaie de lier un module dans le noyau en cours d'exécution en résolvant les noms de symboles à partir de la table des symboles exportés par le noyau.

<sup>1)</sup>

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:systeme:modprobe>



Last update: **14/05/2023 11:03**