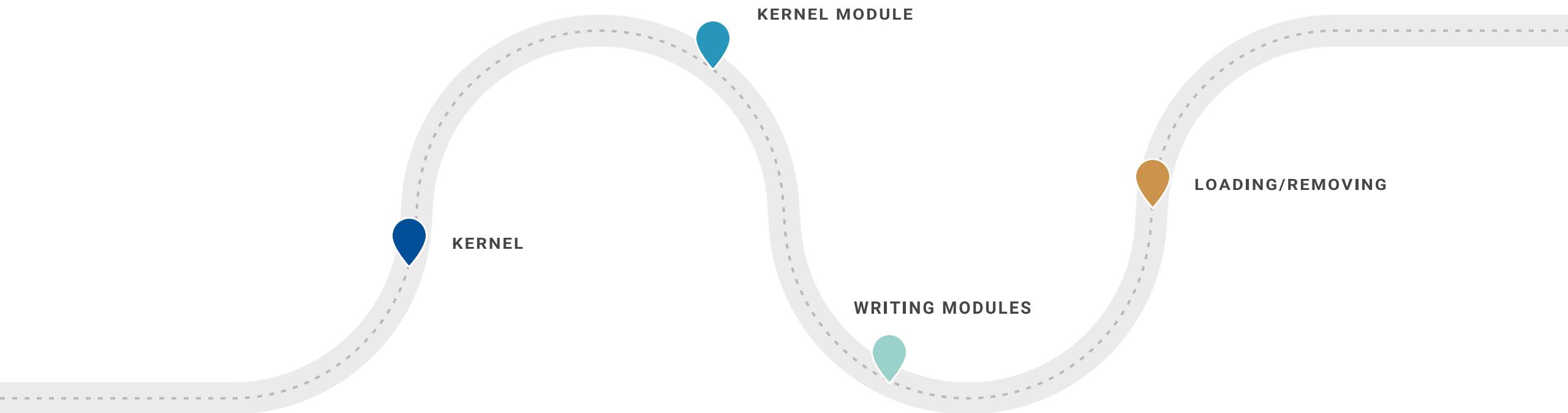


Introduction to : Kernel Module Programming

A KOSS INITIATIVE

Contents



Section 1

KERNEL

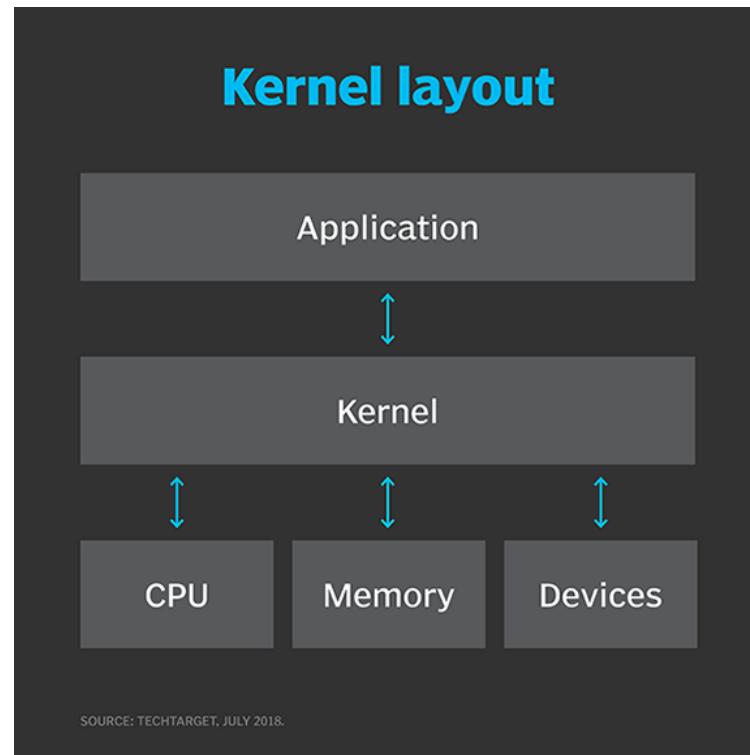
WHAT IS A KERNEL ?

HAS CONTROL OVER EVERYTHING

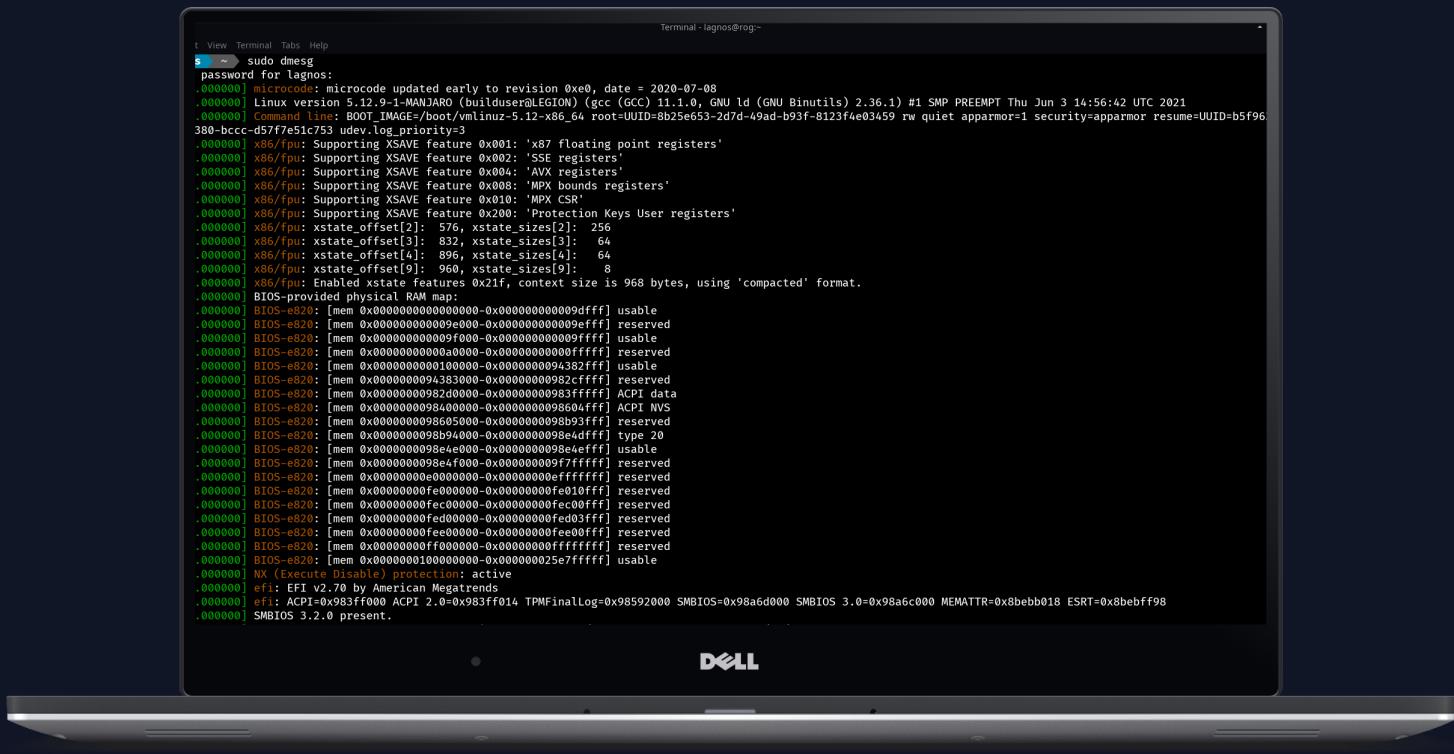
CONNECTS APPLICATION SOFTWARE WITH THE COMPUTER HARDWARE.

MANAGES THE FOLLOWING:

- RAM allocation to programs
- Access to IO devices
- Access to resources like CPU threads
- Access to various privileged processes
- And much more...



The Linux Kernel



● MONOLITHIC

Most of the hardware related programs are present in the kernel space. Syscalls are used to access kernel space from the user space.

● MODULAR

The Linux kernel is also modular which means it allows the kernel to be extended by the use of modules which can be loaded and unloaded without restarting the OS after every change.

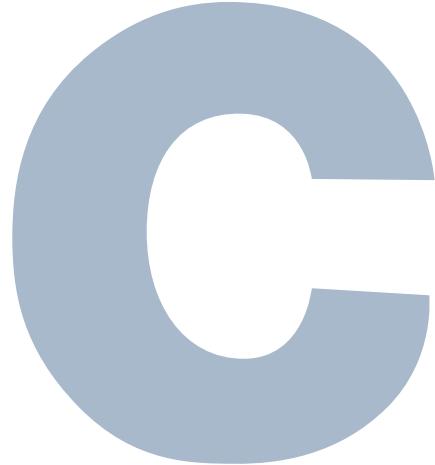
Things to do first...



FOLLOW HIM!

<https://github.com/torvalds>

Things to do first...

THE
PROGRAMMING
LANGUAGE

GET FLUENT IN C

Things to do first...

The Linux Kernel Archives

About Contact us FAQ Releases Signatures Site news

Protocol Location
HTTP <https://www.kernel.org/pub/>
GIT <https://git.kernel.org/>
RSYNC <rsync://rsync.kernel.org/pub/>

Latest Release
5.12.13 

mainline: 5.13-rc7 2021-06-20 [tarball] [patch] [inc. patch] [view diff] [browse]
stable: 5.12.13 2021-06-23 [tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm: 5.10.46 2021-06-23 [tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm: 5.4.128 2021-06-23 [tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm: 4.19.195 2021-06-16 [tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm: 4.14.237 2021-06-16 [tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm: 4.9.273 2021-06-16 [tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm: 4.4.273 2021-06-16 [tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
linux-next: next-20210623 2021-06-23 [tarball] [patch] [inc. patch] [view diff] [browse]

Other resources

Cgit Bugzilla Mirrors	Documentation Patchwork Linux.com	Wikis Kernel Mailing Lists Linux Foundation
---	---	---

Social

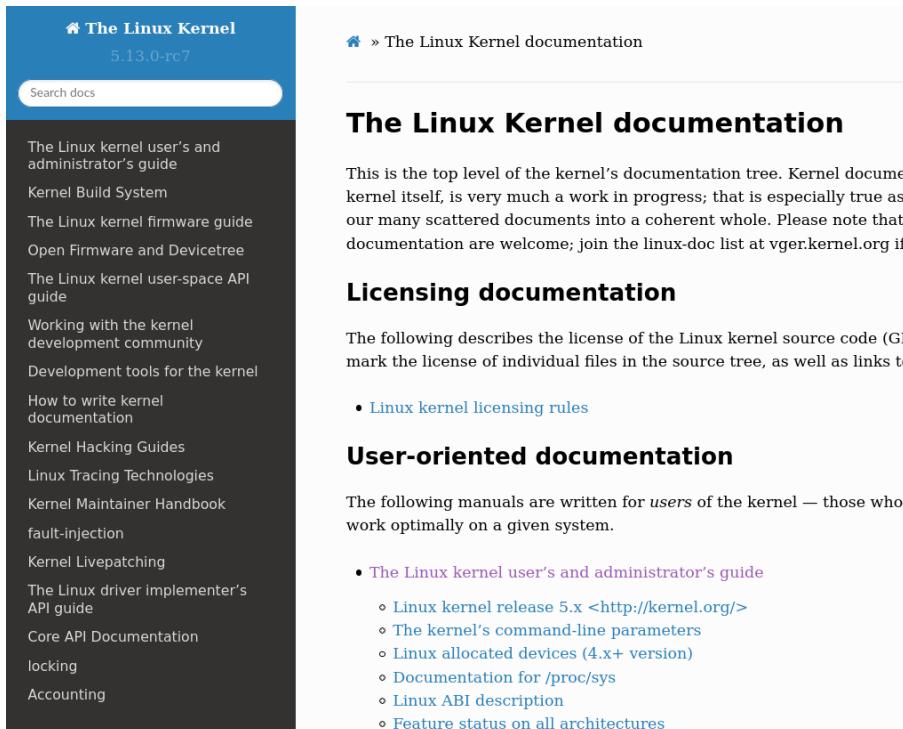
Site Atom feed Releases Atom Feed Kernel Planet

DOWNLOAD THE LINUX KERNEL

<https://www.kernel.org/>

AND BUILD THE SOURCE CODE FROM ANY TUTORIAL AVAILABLE
ONLINE SPECIFIC TO YOUR DISTRO.

Things to do first...



The screenshot shows the homepage of the Linux Kernel documentation. The top navigation bar includes a logo, the text "The Linux Kernel", the version "5.13.0-rc7", and a "Search docs" input field. The main content area has a header "The Linux Kernel documentation". Below it, there's a section titled "Licensing documentation" with a bullet point "Linux kernel licensing rules". Another section titled "User-oriented documentation" lists several items under "The Linux kernel user's and administrator's guide", such as "Linux kernel release 5.x", "The kernel's command-line parameters", and "Documentation for /proc/sys". A sidebar on the left contains links to various kernel-related topics like "The Linux kernel user's and administrator's guide", "Kernel Build System", and "Working with the kernel development community".

READ THE LINUX KERNEL DOCUMENTATION

Read the documentation and coding style. Remember the Linux Kernel maintainers are strict with the coding style and indentations.

Things to do first...

Last 100 messages

Today's messages



Based in The Netherlands and complies with EU-GDPR privacy laws

Latest messages

Jim Mattson	Re: [PATCH 07/54] KVM: x86: Alert userspace that K...
"Gustavo A. R. Silva"	Re: linux-next: build warning after merge of the k...
Hamza Mahfooz	[PATCH] iwlwifi: remove redundant calls to unlikel...
Oded Gabbay	Re: [Linaro-mm-sig] [PATCH v3 1/2] habanalabs: def...
Siddharth Gupta	Re: [PATCH v3 1/4] remoteproc: core: Move cdev add...
Jason Gunthorpe	Re: [Linaro-mm-sig] [PATCH v3 1/2] habanalabs: def...
Drew Fustini	Re: [RFC PATCH 0/1] Adding jh7100 SoC to defconfig
Olivier Langlois	[PATCH v2 2/2] io_uring: Create define to modify a...
Olivier Langlois	[PATCH v2 1/2] io_uring: Fix race condition when s...
Salvatore Bonacorso	Re: Bug#989778: Regression in at least 5.10.y and ...
Olivier Langlois	[PATCH v2 0/2] Minor SQPOLL thread fix and improve...
pr-tracker-bot@kernel ...	Re: [GIT PULL] SPI fixes for v5.13-rc7
Paolo Bonzini	Re: [PATCH 07/54] KVM: x86: Alert userspace that K...
jim.cromie@gmail ...	Re: [V9fs-developer] KCSAN BUG report on p9_client...
Bjorn Andersson	Re: [PATCH] remoteproc: stm32: fix mbox_send_messa...

LKML.ORG?

In case you haven't read the titlebar of this site is the (unofficial) Linux Kernel mailing list is a rather high-volume list, on the design of, and bugs in the Linux scares you, please [read the FAQ](#).

SUBSCRIBE TO THE MAILING LIST

<https://lkml.org/>

Section 2

Kernel Modules

Building Modules

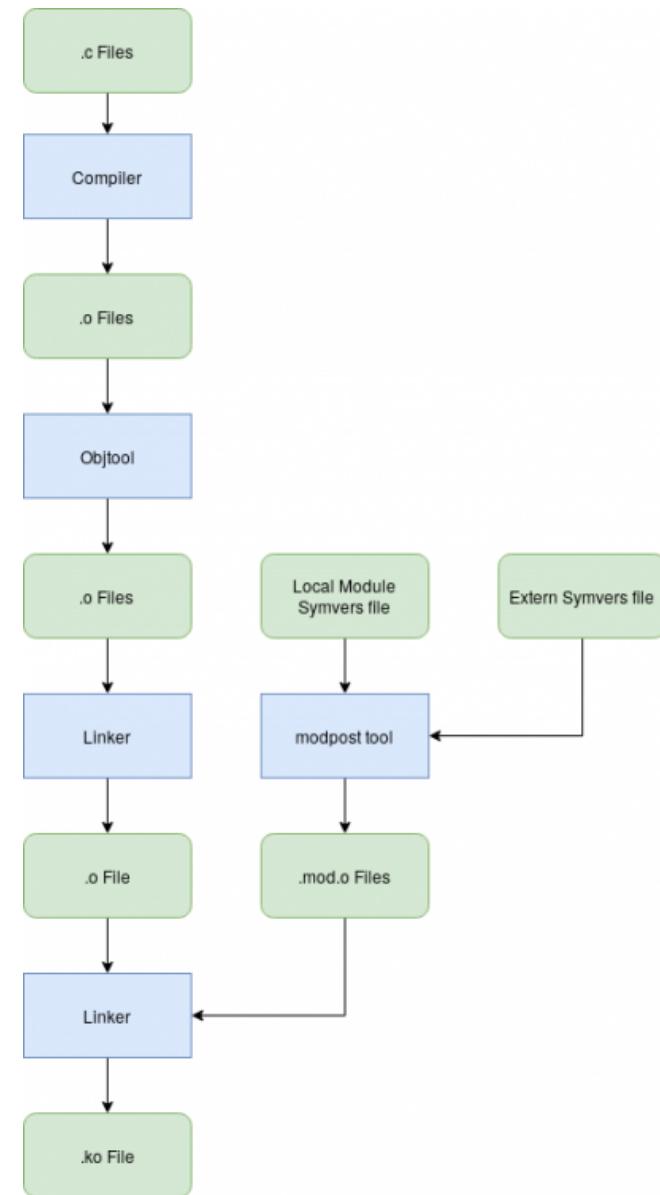
The process of building a .c file to module.

The Linux Modules are pieces of codes in C.

These C files are compiled to object files and then linked. Then the modpost tool is called.

When building an external module, the build system needs access to the symbols from the kernel to check if all external symbols are defined. This is done in the modpost step and a new object file is generated.

At the end a .ko file, i.e a kernel object file is generated. This file is used to load the kernel modules.



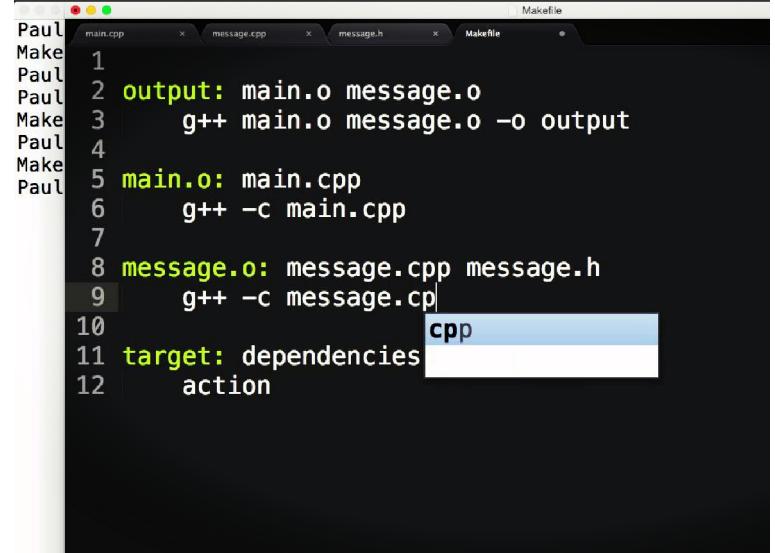
Makefile

As you saw in the previous slide that building a single source code for linux kernel module requires many files to be compiled, linked and recompiled.

Here comes *make*. *Make* uses *Makefile* to automate the process of recompiling a program along with its dependencies when any of the associated files change.

This is very useful for programs written in C/C++ which are not interpreted but compiled.

We will see only those parts which are required for LKM. For more on Make you can refer : <https://maketutorial.com/>

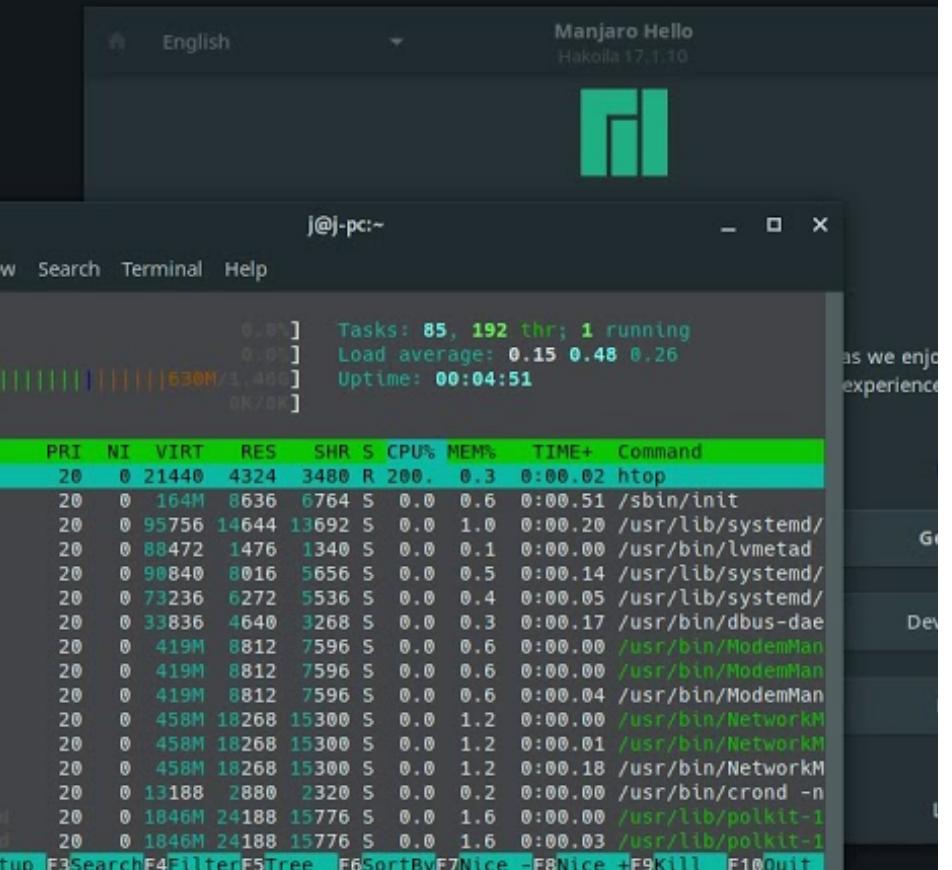


A screenshot of a terminal window titled "Makefile". The window contains a Makefile with the following content:

```
1 2 output: main.o message.o  
3     g++ main.o message.o -o output  
4  
5 main.o: main.cpp  
6     g++ -c main.cpp  
7  
8 message.o: message.cpp message.h  
9     g++ -c message.cpp  
10  
11 target: dependencies  
12     action
```

The code is color-coded: "output", "main.o", "message.o", "main.cpp", "message.h", and "target" are in green; "g++" and "action" are in blue; and "dependencies" is in red. A blue box highlights the word "cpp" in the line "g++ -c main.cpp".

Command Line Tools



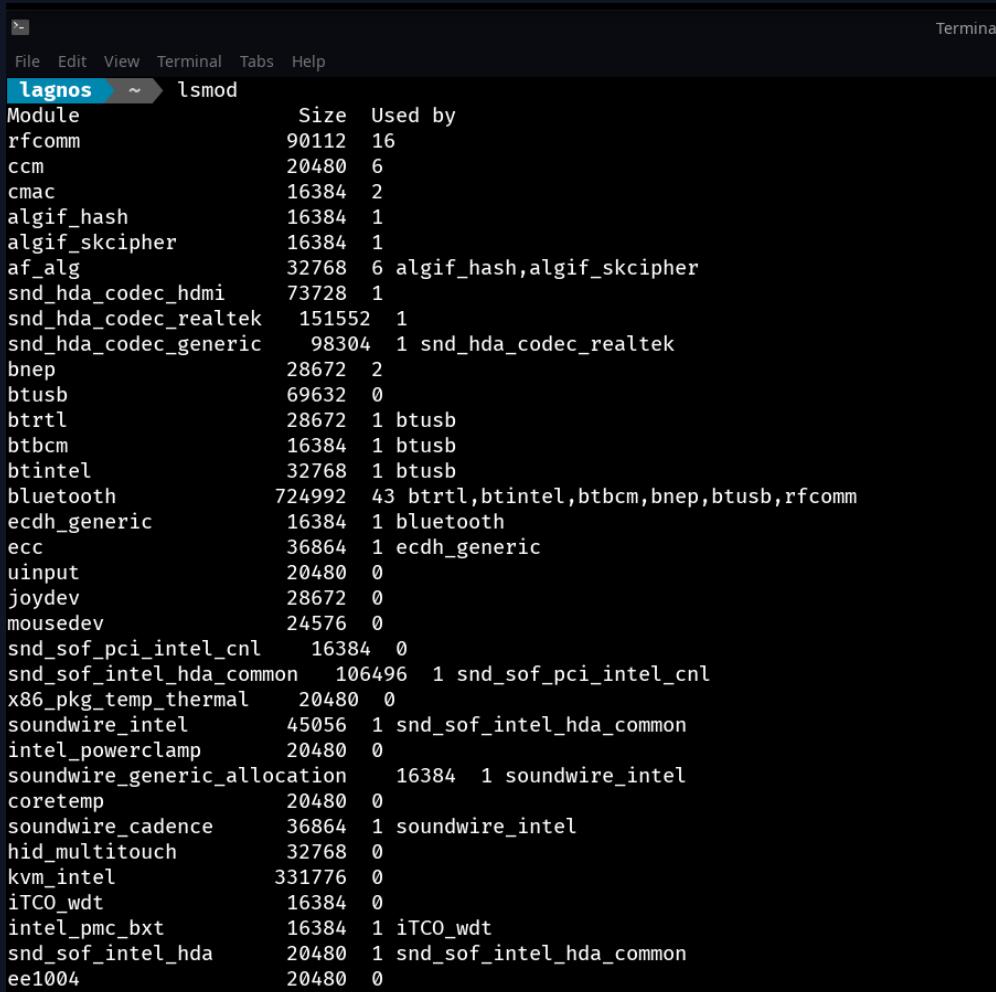
A screenshot of a Manjaro desktop environment. The top bar shows "English" and "Manjaro Hello". A terminal window titled "Terminal" is open, displaying the output of the "htop" command. The output shows system statistics like tasks, load average, and uptime, followed by a list of processes with columns for PRI, NI, VIRT, RES, SHR, CPU%, MEM%, TIME+, and Command. The terminal window has a dark theme with green highlights for selected text.

PRI	NI	VIRT	RES	SHR	CPU%	MEM%	TIME+	Command
20	0	21440	4324	3480	R	200.	0.3	0:00.02 htop
20	0	164M	8636	6764	S	0.0	0.6	0:00.51 /sbin/init
20	0	95756	14644	13692	S	0.0	1.0	0:00.20 /usr/lib/systemd/
20	0	88472	1476	1340	S	0.0	0.1	0:00.00 /usr/bin/lvmetad
20	0	90840	8016	5656	S	0.0	0.5	0:00.14 /usr/lib/systemd/
20	0	73236	6272	5536	S	0.0	0.4	0:00.05 /usr/lib/systemd/
20	0	33836	4640	3268	S	0.0	0.3	0:00.17 /usr/bin/dbus-dae
20	0	419M	8812	7596	S	0.0	0.6	0:00.00 /usr/bin/ModemMan
20	0	419M	8812	7596	S	0.0	0.6	0:00.00 /usr/bin/ModemMan
20	0	419M	8812	7596	S	0.0	0.6	0:00.04 /usr/bin/ModemMan
20	0	458M	18268	15300	S	0.0	1.2	0:00.00 /usr/bin/NetworkM
20	0	458M	18268	15300	S	0.0	1.2	0:00.01 /usr/bin/NetworkM
20	0	458M	18268	15300	S	0.0	1.2	0:00.18 /usr/bin/NetworkM
20	0	13188	2880	2320	S	0.0	0.2	0:00.00 /usr/bin/crond -n
20	0	1846M	24188	15776	S	0.0	1.6	0:00.00 /usr/lib/polkit-1
20	0	1846M	24188	15776	S	0.0	1.6	0:00.03 /usr/lib/polkit-1

There are a few major tools which we will use to perform various functions with kernel modules.

We will also look at various files which store information regarding kernel like module details, messages logged by kernel modules, etc.

lsmod

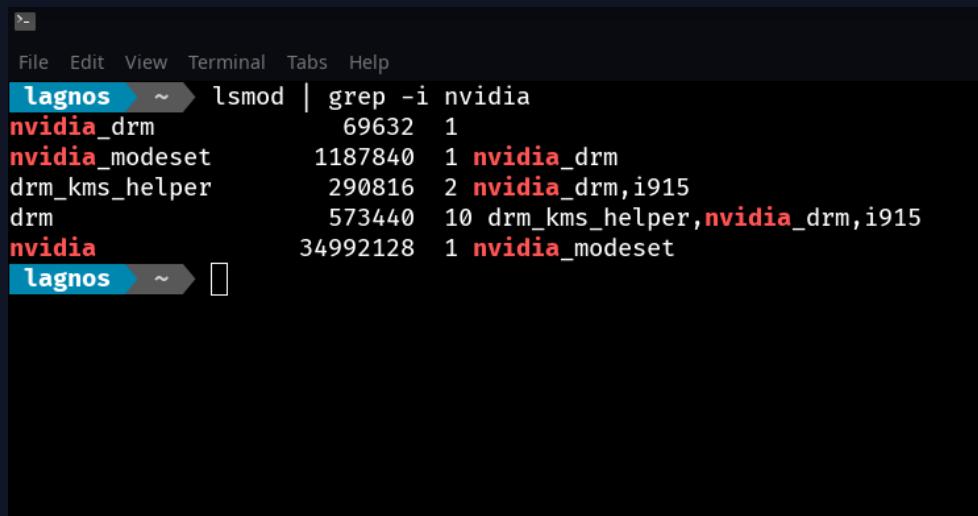
A screenshot of a terminal window titled "Terminal". The window shows the output of the "lsmod" command. The output lists various kernel modules along with their sizes and dependencies. The modules listed include rfcomm, ccm, cmac, algif_hash, algif_skcipher, af_alg, snd_hda_codec_hdmi, snd_hda_codec_realtek, snd_hda_codec_generic, bnep, btusb, btrtl, btbcm, btintel, bluetooth, ecdh_generic, ecc, uinput, joydev, mousedev, snd_sof_pci_intel_cnl, snd_sof_intel_hda_common, x86_pkg_temp_thermal, soundwire_intel, intel_powerclamp, soundwire_generic_allocation, coretemp, soundwire_cadence, hid_multitouch, kvm_intel, iTCO_wdt, intel_pmc_bxt, snd_sof_intel_hda, and ee1004. The "Module" column lists the module names, "Size" shows the memory footprint, and "Used by" indicates which other modules depend on them.

```
Module           Size  Used by
rfcomm          90112  16
ccm            20480   6
cmac           16384   2
algif_hash      16384   1
algif_skcipher 16384   1
af_alg          32768   6 algif_hash,algif_skcipher
snd_hda_codec_hdmi 73728   1
snd_hda_codec_realtek 151552   1
snd_hda_codec_generic 98304   1 snd_hda_codec_realtek
bnep            28672   2
btusb           69632   0
btrtl           28672   1 btusb
btbcm           16384   1 btusb
btintel          32768   1 btusb
bluetooth       724992  43 btrtl,btintel,btbcm,bnep,btusb,rfcomm
ecdh_generic    16384   1 bluetooth
ecc             36864   1 ecdh_generic
uinput          20480   0
joydev          28672   0
mousedev        24576   0
snd_sof_pci_intel_cnl 16384   0
snd_sof_intel_hda_common 106496  1 snd_sof_pci_intel_cnl
x86_pkg_temp_thermal 20480   0
soundwire_intel 45056   1 snd_sof_intel_hda_common
intel_powerclamp 20480   0
soundwire_generic_allocation 16384   1 soundwire_intel
coretemp         20480   0
soundwire_cadence 36864   1 soundwire_intel
hid_multitouch  32768   0
kvm_intel        331776   0
iTCO_wdt         16384   0
intel_pmc_bxt   16384   1 iTCO_wdt
snd_sof_intel_hda 20480   1 snd_sof_intel_hda_common
ee1004          20480   0
```

LISTING MODULES

lsmod is used to list all the loadable modules that are currently loaded in the kernel.

lsmod



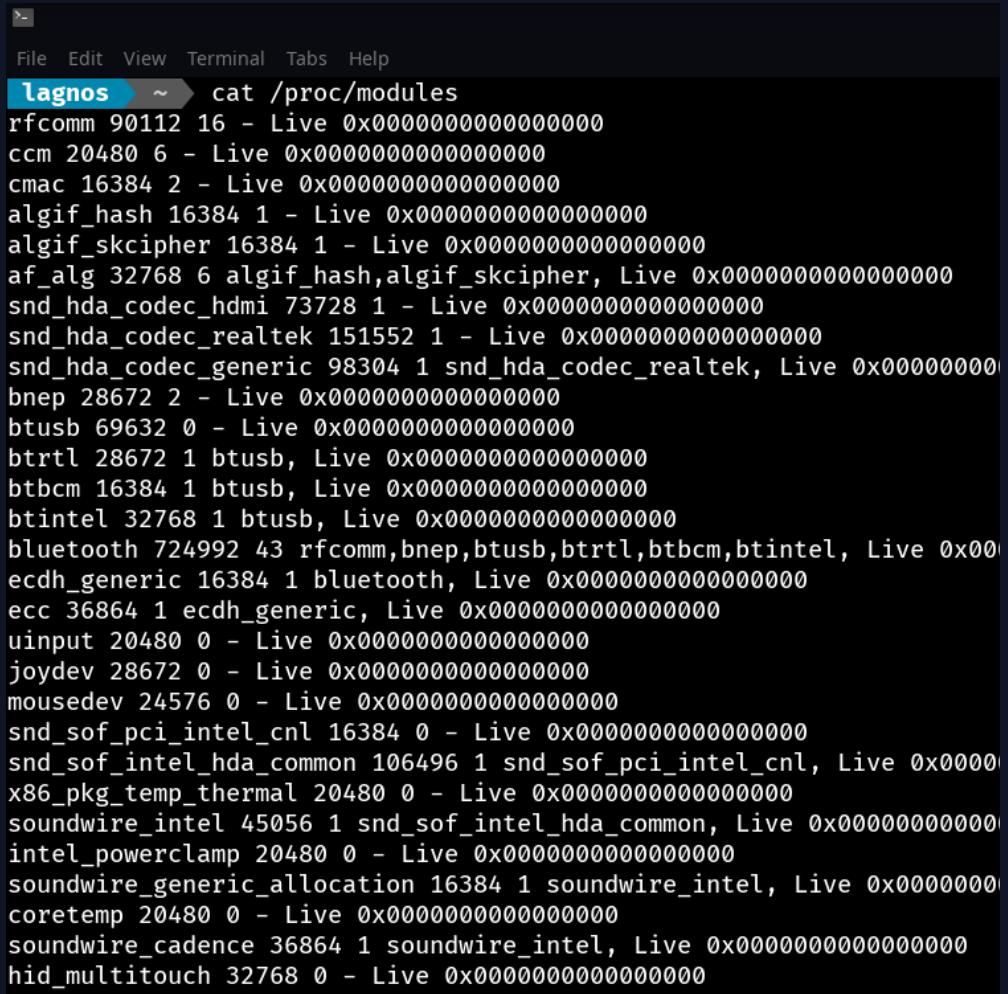
```
File Edit View Terminal Tabs Help
lagnos ~ ➔ lsmod | grep -i nvidia
nvidia_drm          69632  1
nvidia_modeset      1187840 1 nvidia_drm
drm_kms_helper      290816  2 nvidia_drm,i915
drm                573440  10 drm_kms_helper,nvidia_drm,i915
nvidia             34992128  1 nvidia_modeset
lagnos ~ ➔
```

FILTERING MODULES

The output from *lsmod* can be piped (|) to *grep* to filter out data according to your need as illustrated here. This lists all the modules which contain “nvidia” in their module name.

The *-i* command is used to ignore case to output matching lower and uppercase strings.

lsmod



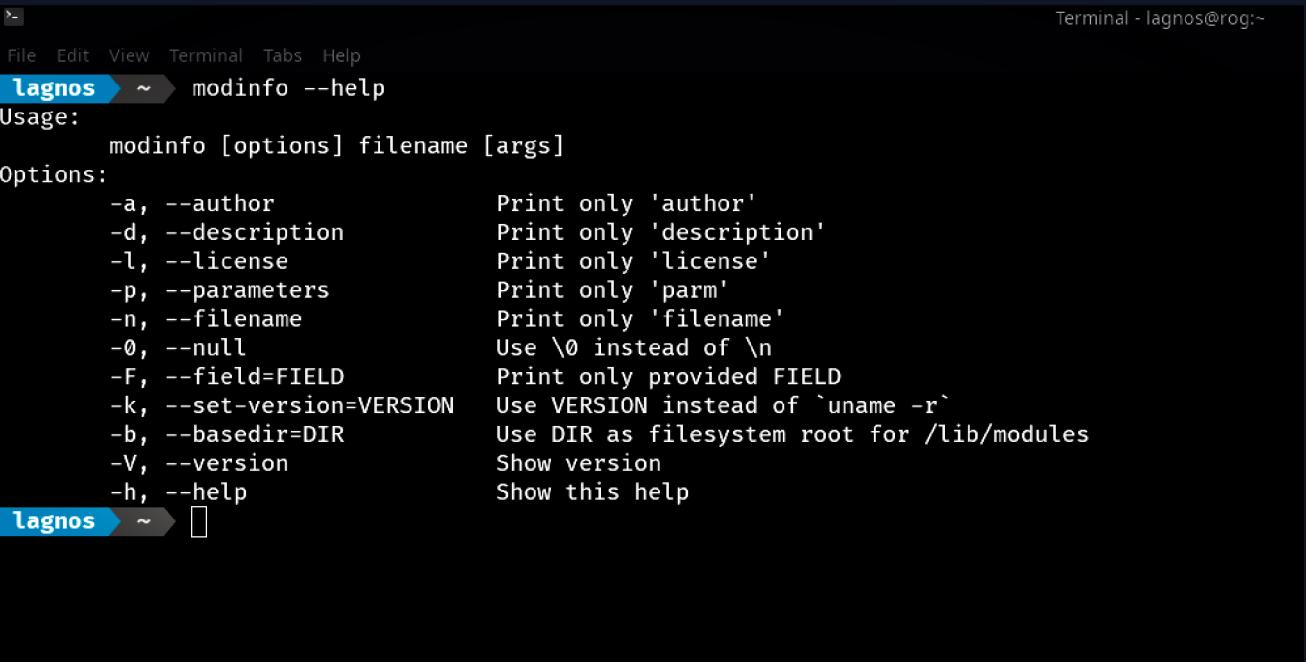
```
File Edit View Terminal Tabs Help
lagnos ~ cat /proc/modules
rfcomm 90112 16 - Live 0x0000000000000000
ccm 20480 6 - Live 0x0000000000000000
cmac 16384 2 - Live 0x0000000000000000
algif_hash 16384 1 - Live 0x0000000000000000
algif_skcipher 16384 1 - Live 0x0000000000000000
af_alg 32768 6 algif_hash,algif_skcipher, Live 0x0000000000000000
snd_hda_codec_hdmi 73728 1 - Live 0x0000000000000000
snd_hda_codec_realtek 151552 1 - Live 0x0000000000000000
snd_hda_codec_generic 98304 1 snd_hda_codec_realtek, Live 0x00000000
bnep 28672 2 - Live 0x0000000000000000
btusb 69632 0 - Live 0x0000000000000000
btrtl 28672 1 btusb, Live 0x0000000000000000
btbcm 16384 1 btusb, Live 0x0000000000000000
btintel 32768 1 btusb, Live 0x0000000000000000
bluetooth 724992 43 rfcomm,bnep,btusb,btrtl,btbcm,btintel, Live 0x00000000
ecdh_generic 16384 1 bluetooth, Live 0x0000000000000000
ecc 36864 1 ecdh_generic, Live 0x0000000000000000
uinput 20480 0 - Live 0x0000000000000000
joydev 28672 0 - Live 0x0000000000000000
mousedev 24576 0 - Live 0x0000000000000000
snd_sof_pci_intel_cnl 16384 0 - Live 0x0000000000000000
snd_sof_intel_hda_common 106496 1 snd_sof_pci_intel_cnl, Live 0x00000000
x86_pkg_temp_thermal 20480 0 - Live 0x0000000000000000
soundwire_intel 45056 1 snd_sof_intel_hda_common, Live 0x000000000000
intel_powerclamp 20480 0 - Live 0x0000000000000000
soundwire_generic_allocation 16384 1 soundwire_intel, Live 0x00000000
coretemp 20480 0 - Live 0x0000000000000000
soundwire_cadence 36864 1 soundwire_intel, Live 0x0000000000000000
hid_multitouch 32768 0 - Live 0x0000000000000000
```

UNDER THE HOOD

lsmod under the hood, nicely formats the contents of the file /proc/modules. As you can see in the image and compare the first few lines with the lsmod ouput.

modinfo

modinfo displays the information about a kernel module which may be loaded or a module which has been built already locally.

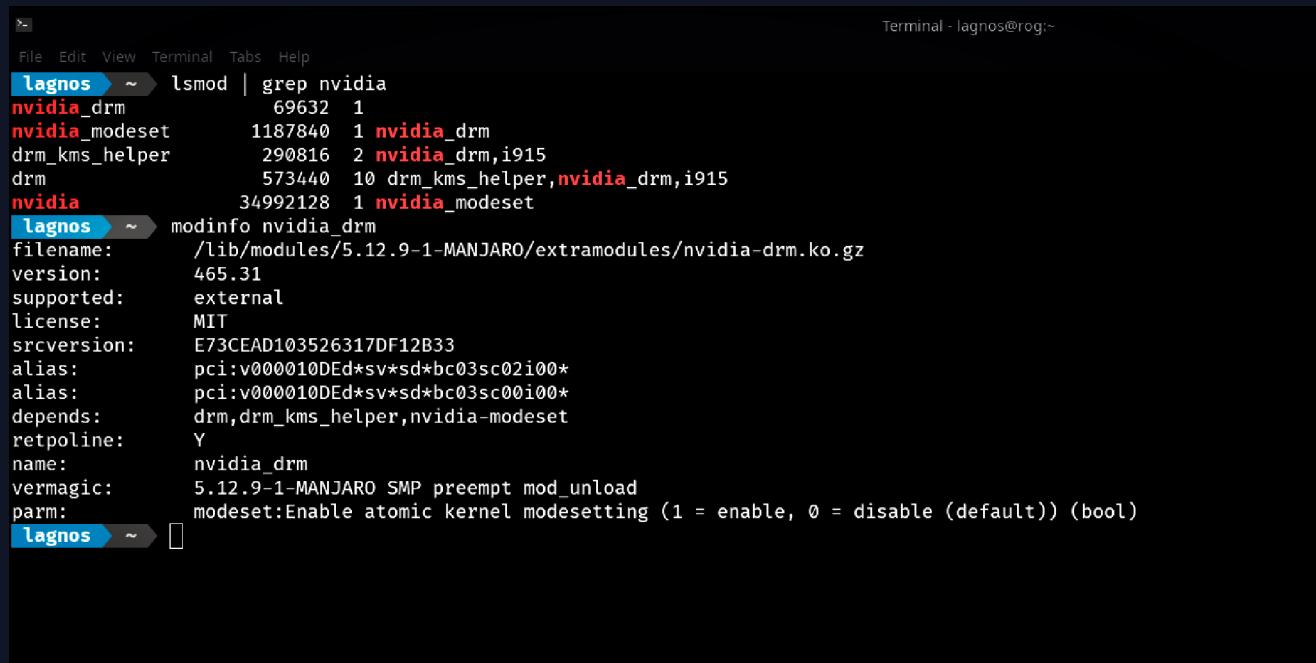


The screenshot shows a terminal window with a dark theme. The title bar reads "Terminal - lagnos@rog:~". The command "modinfo --help" is entered, followed by its usage and option descriptions:

```
File Edit View Terminal Tabs Help
lagnos ~ modinfo --help
Usage:
    modinfo [options] filename [args]
Options:
    -a, --author           Print only 'author'
    -d, --description      Print only 'description'
    -l, --license          Print only 'license'
    -p, --parameters       Print only 'parm'
    -n, --filename         Print only 'filename'
    -\0, --null             Use \0 instead of \n
    -F, --field=FIELD      Print only provided FIELD
    -k, --set-version=VERSION Use VERSION instead of `uname -r`
    -b, --basedir=DIR       Use DIR as filesystem root for /lib/modules
    -V, --version           Show version
    -h, --help              Show this help
lagnos ~
```

modinfo

Here is an example of *nvidia* module we search using *lsmod* command and use that to display its properties like licensing, version, etc.

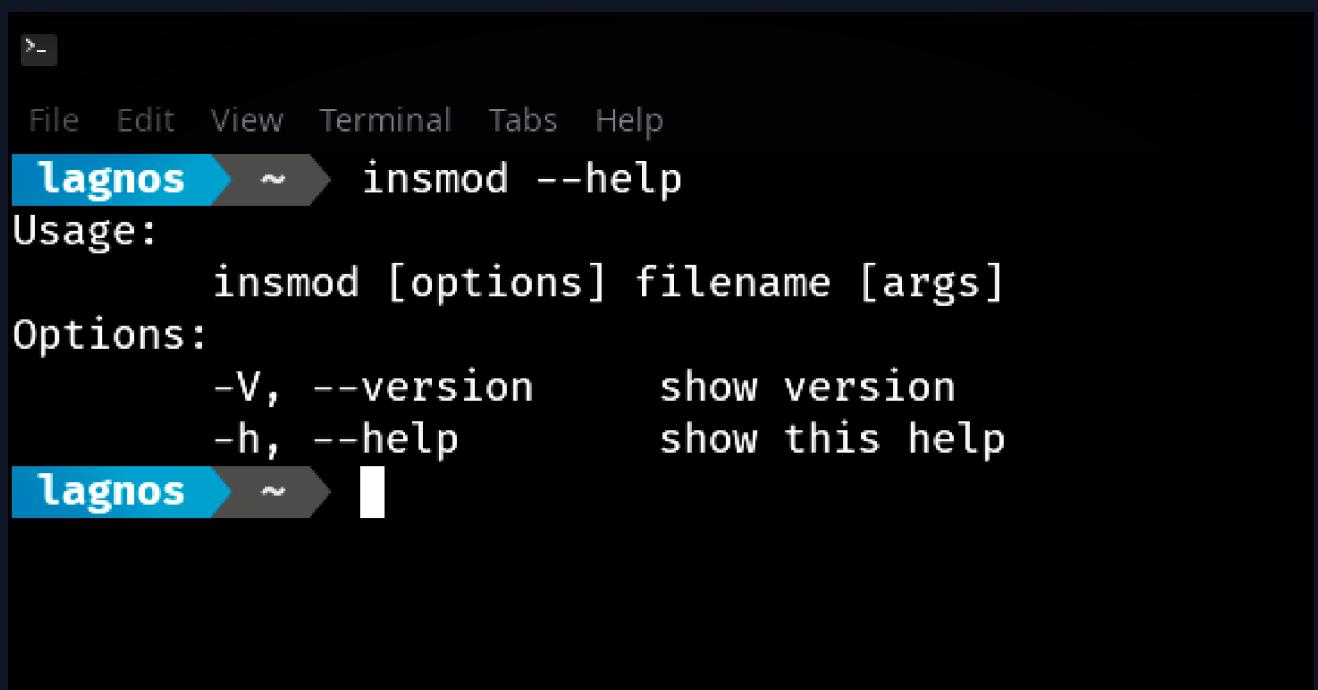


```
File Edit View Terminal Tabs Help
lagnos ~ lsmod | grep nvidia
nvidia_drm          69632  1
nvidia_modeset      1187840  1 nvidia_drm
drm_kms_helper      290816  2 nvidia_drm,i915
drm                 573440  10 drm_kms_helper,nvidia_drm,i915
nvidia              34992128  1 nvidia_modeset
lagnos ~ modinfo nvidia_drm
filename:           /lib/modules/5.12.9-1-MANJARO/extramodules/nvidia-drm.ko.gz
version:            465.31
supported:         external
license:             MIT
srcversion:        E73CEAD103526317DF12B33
alias:              pci:v000010DEd*sv*sd*bc03sc02i00*
alias:              pci:v000010DEd*sv*sd*bc03sc00i00*
depends:            drm,drm_kms_helper,nvidia-modeset
retpoline:          Y
name:               nvidia_drm
vermagic:          5.12.9-1-MANJARO SMP preempt mod_unload
parm:               modeset:Enable atomic kernel modesetting (1 = enable, 0 = disable (default)) (bool)
```

insmod

insmod is used to insert(load) the loadable linux kernel module.

We will use *insmod* later to insert(load) the coded modules.



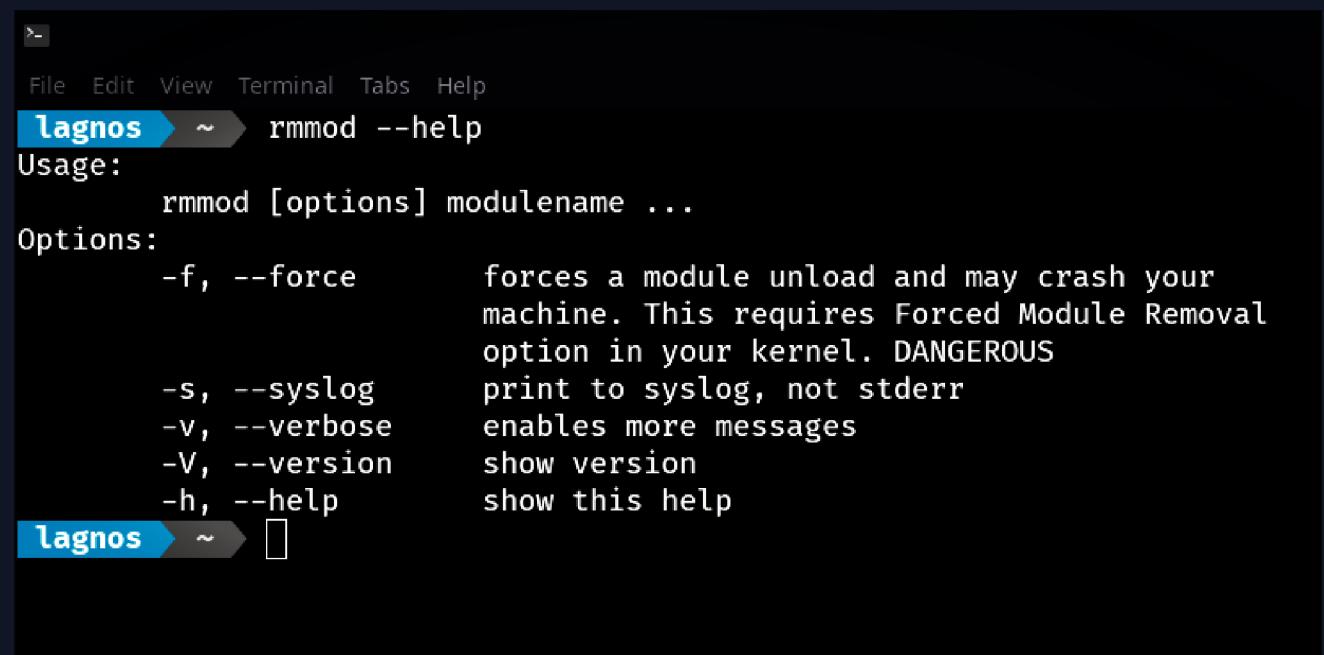
```
File Edit View Terminal Tabs Help
lagnos ~ insmod --help
Usage:
    insmod [options] filename [args]
Options:
    -V, --version      show version
    -h, --help         show this help
lagnos ~
```

A screenshot of a terminal window with a dark background. The window title bar shows 'File Edit View Terminal Tabs Help'. The user's prompt is 'lagnos ~'. The command 'insmod --help' is entered and its usage information is displayed. The usage information includes the command syntax 'insmod [options] filename [args]' and two options: '-V, --version' which shows the version, and '-h, --help' which shows this help message. The terminal window has a blue header bar and a white body area where the text is displayed.

rmmod

rmmod is used to unload kernel modules.

We will use *rmmod* later to unload the coded modules.



```
File Edit View Terminal Tabs Help
lagnos ~ ➔ rmmod --help
Usage:
    rmmod [options] modulename ...
Options:
    -f, --force      forces a module unload and may crash your
                     machine. This requires Forced Module Removal
                     option in your kernel. DANGEROUS
    -s, --syslog     print to syslog, not stderr
    -v, --verbose    enables more messages
    -V, --version    show version
    -h, --help       show this help
lagnos ~ ➔
```

modprobe

modprobe is used to load, unload or show details and config of modules

modprobe is much more intelligent than *insmod*. *insmod* simply adds the module whereas *modprobe* looks for the dependencies and loads them too.

modprobe looks for .ko modules in the standard directory - */lib/modules/\$(uname -r)* whereas *insmod* can be used to load modules from any path.

```

File Edit View Terminal Tabs Help
lagnos ~ modprobe --help
Usage:
modprobe [options] [-i] [-b] modulename
modprobe [options] -a [-i] [-b] modulename [modulename...]
modprobe [options] -r [-i] modulename
modprobe [options] -r -a [-i] modulename [modulename...]
modprobe [options] -c
modprobe [options] --dump-modversions filename

Management Options:
-a, --all
Consider every non-argument to
be a module name to be inserted
or removed (-r)
Remove modules instead of inserting
Also remove modules depending on it
Only lookup and print alias and exit
Fail if module already inserted or removed
Ignore install commands
Ignore remove commands
Apply blacklist to resolved alias.
Force module insertion or removal.
implies --force-modversions and
--force-vermagic
Ignore module's version
Ignore module's version magic

--force-modversion
--force-vermagic

Query Options:
-D, --show-dependencies
Only print module dependencies and exit
-c, --showconfig
Print out known configuration and exit
-c, --show-config
Same as --showconfig
--show-modversions
Dump module symbol version and exit
--dump-modversions
Same as --show-modversions
--show-exports
Only print module exported symbol versions and

General Options:
-n, --dry-run
Do not execute operations, just print out
-n, --show
Same as --dry-run
-C, --config=FILE
Use FILE instead of default search paths
-d, --dirname=DIR
Use DIR as filesystem root for /lib/modules
-S, --set-version=VERSION
Use VERSION instead of `uname -r`
-s, --syslog
print to syslog, not stderr
-q, --quiet
disable messages

```

Blacklisting

● MODULES CAUSING ERRORS

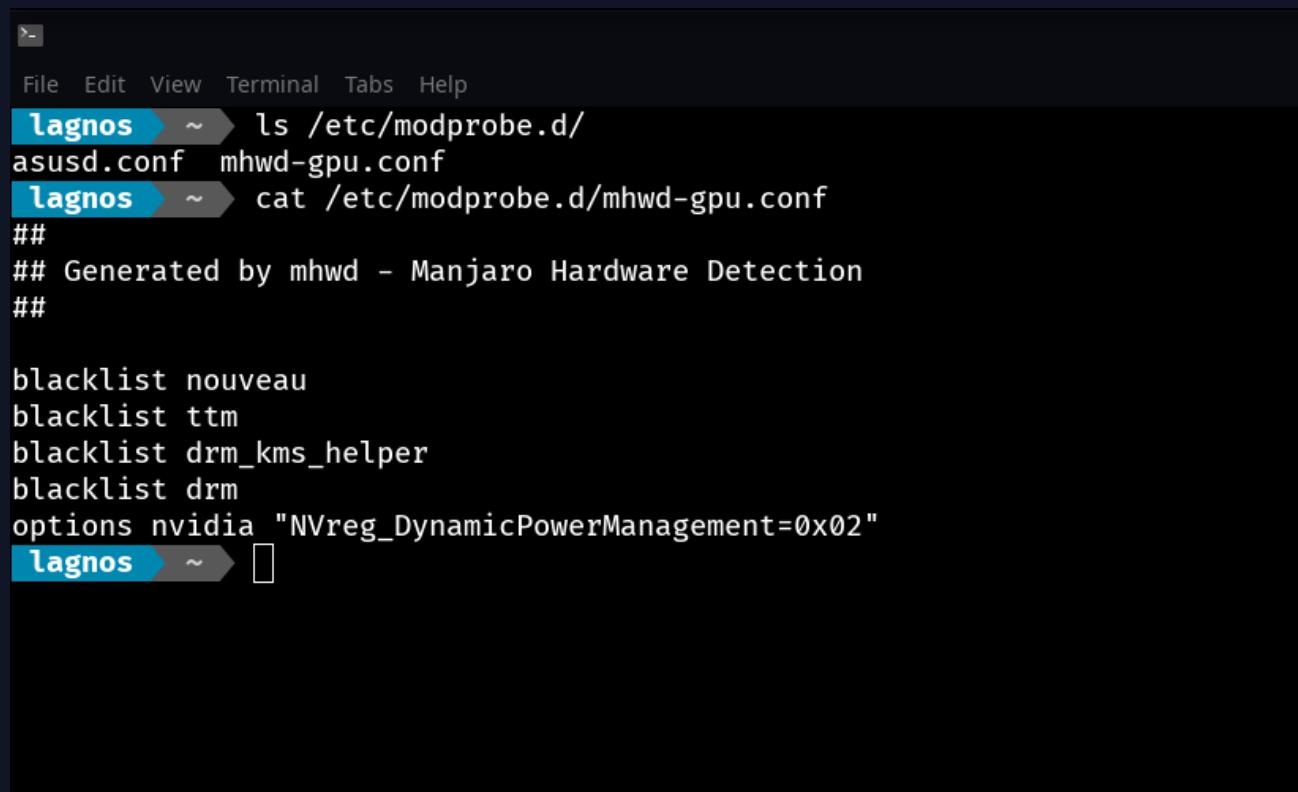
Some modules can cause error due to bugs or memory problems which can cause the related hardware to run incorrectly.

● CONFLICT AMONG MODULES

It may happen that two drivers try to access the same hardware resource.

● PRACTICAL CASE

Manually installing *NVIDIA* *proprietary* drivers in Linux. To install the *proprietary* driver we first need to disable the *open source driver - Nouveau* so that the official driver can get access to the nvidia graphics card and set it as the default renderer.



```
File Edit View Terminal Tabs Help
lagnos ~ ls /etc/modprobe.d/
asusd.conf mhwd-gpu.conf
lagnos ~ cat /etc/modprobe.d/mhwd-gpu.conf
##
## Generated by mhwd - Manjaro Hardware Detection
##

blacklist nouveau
blacklist ttm
blacklist drm_kms_helper
blacklist drm
options nvidia "NVreg_DynamicPowerManagement=0x02"
lagnos ~
```

Files containing logs and messages

```
File Edit View Terminal Tabs Help
lagos ~ sudo dmesg
[sudo] password for lagos:
[    0.000000] microcode: microcode updated early to revision 0xe0, date = 2020-07-08
[    0.000000] Linux version 5.12.9-1-MANJARO (builduser@LEGION) (gcc (GCC) 11.1.0, GNU ld
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.12-x86_64 root=UUID=8b25e653-2d7d-4
cabf-4380-bccc-d57f7e51c753 udev.log_priority=3
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x008: 'MPX bounds registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x010: 'MPX CSR'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x200: 'Protection Keys User registers'
[    0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[    0.000000] x86/fpu: xstate_offset[3]: 832, xstate_sizes[3]: 64
[    0.000000] x86/fpu: xstate_offset[4]: 896, xstate_sizes[4]: 64
[    0.000000] x86/fpu: xstate_offset[9]: 960, xstate_sizes[9]: 8
[    0.000000] x86/fpu: Enabled xstate features 0x21f, context size is 968 bytes, using 'co
[    0.000000] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009ffff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009e000-0x000000000009efff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000000009f000-0x0000000000000009ffff] usable
[    0.000000] BIOS-e820: [mem 0x000000000000a000-0x000000000000ffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x00000000094382ffff] usable
[    0.000000] BIOS-e820: [mem 0x00000000094383000-0x000000000982cfffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000982d0000-0x000000000983ffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x00000000098400000-0x00000000098604ffff] ACPI NVS
[    0.000000] BIOS-e820: [mem 0x00000000098605000-0x00000000098b93ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000098b94000-0x00000000098e4dff] type 20
[    0.000000] BIOS-e820: [mem 0x00000000098e4e000-0x00000000098e4effff] usable
[    0.000000] BIOS-e820: [mem 0x00000000098e4f000-0x0000000009f7ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000e0000000-0x00000000effffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fe000000-0x00000000fe010ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fed00000-0x00000000fed03ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee0ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000ff000000-0x00000000ffffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000010000000-0x0000000025e7ffff] usable
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] efi: EFI v2.70 by American Megatrends
[    0.000000] efi: ACPI=0x983ff000 ACPI 2.0=0x983ff014 TPMFinalLog=0x98592000 SMBIOS=0x98a
[    0.000000] SMBIOS 3.2.0 present.
```

OMESG

mesg is used to examine or control the kernel ring buffer. Generally it is used to display all messages from the kernel ring buffer.

The kernel ring buffer is a data structure that records messages related to the operation of the kernel. It is always constant in size, removing the oldest messages when new messages are received.

```
[  OK ] Reached target Remote File Systems.
[  OK ] Listening on Delayed Shutdown Socket.
[  OK ] Listening on /dev/initctl Compatibility Named Pipe.
[  OK ] Reached target Encrypted Volumes.
[  OK ] Listening on udev Kernel Socket.
[  OK ] Listening on udev Control Socket.
[  OK ] Set up automount Arbitrary Executable File Formats F...automount.
Expecting device dev-disk-by\x2duuid-6038ea52\x2d80a...ce4c9.d...
[  OK ] Listening on Journal Socket.
Starting File System Check on Root Device...
Starting udev Kernel Device Manager...
Mounting Debug File System...
Starting Journal Service...
[  OK ] Started Journal Service.
Starting Apply Kernel Variables...
Starting udev Coldplug all Devices...
Mounting Huge Pages File System...
Mounting POSIX Message Queue File System...
Starting Setup Virtual Console...
Starting Set Up Additional Binary Formats...
Mounting Configuration File System...
[  OK ] Started Apply Kernel Variables.
[  OK ] Started udev Kernel Device Manager.
Mounting Arbitrary Executable File Formats File System...
[  OK ] Started udev Coldplug all Devices.
[  OK ] Mounted POSIX Message Queue File System.
[  OK ] Mounted Debug File System.
[  OK ] Mounted Configuration File System.
[  OK ] Mounted Huge Pages File System.
[  OK ] Mounted Arbitrary Executable File Formats File System.
[  OK ] Started Set Up Additional Binary Formats.
[  OK ] Started Setup Virtual Console.
[  OK ] Found device /dev/ttyS0.
systemd-fsck[53]: fedora: clean, 319575/983040 files, 2914206/3932160 b...
[  OK ] Started File System Check on Root Device.
Starting Remount Root and Kernel File Systems...
[  OK ] Started Remount Root and Kernel File Systems.
[  OK ] Reached target Local File Systems (Pre).
Mounting Temporary Directory...
Starting Load Random Seed...
[  OK ] Mounted Temporary Directory.
[  OK ] Reached target Local File Systems.
Starting Recreate Volatile Files and Directories...
[  OK ] Found device QEMU HARDDISK.
Activating swap /dev/disk/by-uuid/6038ea52-80a2-42c5...d0a22e1...
[  OK ] Started Load Random Seed.
[  OK ] Activated swap /dev/disk/by-uuid/6038ea52-80a2-42c5-8902-d0a22...
[  OK ] Reached target Swap.
[  OK ] Started Recreate Volatile Files and Directories.
[  OK ] Reached target System Initialization.
Starting Restore Sound Card State...
[  OK ] Listening on SSH Socket for Per-Connection Servers.
[  OK ] Listening on Avahi mDNS/DNS-SD Stack Activation Socket.
[  OK ] Listening on D-Bus System Message Bus Socket.
[  OK ] Reached target Sockets.
[  OK ] Reached target Basic System.
Starting Network Manager...
Starting Network Time Service...
Starting Command Scheduler...
[  OK ] Started Command Scheduler.
Starting Login Service...
Starting Avahi mDNS/DNS-SD Stack...
Starting Permit User Sessions...
Starting D-Bus System Message Bus...
[  OK ] Started Restore Sound Card State.
[  OK ] Started Network Time Service.
[  OK ] Reached target Network Time Protocol.
[  OK ] Started Permit User Sessions.
Starting Serial Getty on ttyS0...
[  OK ] Started Serial Getty on ttyS0.
Starting Getty on tty1...
[  OK ] Started Getty on tty1.
[  OK ] Reached target Login Prompts.
[  OK ] Started D-Bus System Message Bus.
[  OK ] Started Avahi mDNS/DNS-SD Stack.
[  OK ] Started Login Service.
[  OK ] Started Network Manager.
[  OK ] Reached target Multi-User.
```

Files containing logs and messages

JOURNALCTL

`journalctl` is used to query the contents of the systemd journal.

systemd is a system and service manager for Linux operating systems. It also acts as init system that brings up and maintains *userspace* services.

You must have seen long list of messages when booting in Linux similar to the one beside. That was systemd starting all the services!

```
Welcome to Fedora 17 (Beefy Miracle)!

[ OK ] Reaching target Local File Systems.
[ OK ] Listening on Delayed Shutdown Socket.
[ OK ] Listening on /dev/initctl Compatibility Named Pipe.
[ OK ] Reached target Encrypted Volumes.
[ OK ] Listening on udev Kernel Socket.
[ OK ] Listening on udev Control Socket.
[ OK ] Set up automount Arbitrary Executable File Formats F...automount.
Expecting device dev-ttyS0.device...
[ OK ] Listening on Journal Socket.
Starting File System Check on Root Device...
Starting udev Kernel Device Manager...
Mounting Debug File System...
Starting Journal Service...
[ OK ] Started Journal Service.
Starting Apply Kernel Variables...
Starting udev Coldplug all Devices...
Mounting Huge Pages File System...
Mounting POSIX Message Queue File System...
Starting Setup Virtual Console...
Starting Set Up Additional Binary Formats...
Mounting Configuration File System...
[ OK ] Started Apply Kernel Variables.
[ OK ] Started udev Kernel Device Manager.
Mounting Arbitrary Executable File Formats File System...
[ OK ] Started udev Coldplug all Devices.
[ OK ] Mounted POSIX Message Queue File System.
[ OK ] Mounted Debug File System.
[ OK ] Mounted Configuration File System.
[ OK ] Mounted Huge Pages File System.
[ OK ] Mounted Arbitrary Executable File Formats File System.
[ OK ] Started Set Up Additional Binary Formats.
[ OK ] Started Setup Virtual Console.
[ OK ] Found device /dev/ttyS0.
stremd-fsck[53]: fedora: clean, 319575/983040 files, 2914206/3932160 b...
[ OK ] Started File System Check on Root Device.
Starting Remount Root and Kernel File Systems...
[ OK ] Started Remount Root and Kernel File Systems.
[ OK ] Reached target Local File Systems (Pre).
Mounting Temporary Directory...
Starting Load Random Seed...
[ OK ] Mounted Temporary Directory.
[ OK ] Reached target Local File Systems.
Starting Recreate Volatile Files and Directories...
[ OK ] Found device OEMU HARDDISK.
Activating swap /dev/disk/by-uuid/6038ea52-80a2-42c5...d0a22e1c.
[ OK ] Started Load Random Seed.
[ OK ] Activated swap /dev/disk/by-uuid/6038ea52-80a2-42c5-8902-d0a22e1c.
[ OK ] Reached target Swap.
[ OK ] Started Recreate Volatile Files and Directories.
[ OK ] Reached target System Initialization.
Starting Restore Sound Card State...
[ OK ] Listening on SSH Socket for Per-Connection Servers.
[ OK ] Listening on Avahi mDNS/DNS-SD Stack Activation Socket.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Basic System.
Starting Network Manager...
Starting Network Time Service...
Starting Command Scheduler...
[ OK ] Started Command Scheduler.
Starting Login Service...
Starting Avahi mDNS/DNS-SD Stack...
Starting Permit User Sessions...
Starting D-Bus System Message Bus...
[ OK ] Started Restore Sound Card State.
[ OK ] Started Network Time Service.
[ OK ] Reached target Network Time Protocol.
[ OK ] Started Permit User Sessions.
Starting Serial Getty on ttyS0...
[ OK ] Started Serial Getty on ttyS0.
Starting Getty on ttyl...
[ OK ] Started Getty on ttyl.
[ OK ] Reached target Login Prompts.
[ OK ] Started D-Bus System Message Bus.
[ OK ] Started Avahi mDNS/DNS-SD Stack.
[ OK ] Started Login Service.
[ OK ] Started Network Manager.
[ OK ] Reached target Multi-User.
```

Section 3

Coding Modules

File Edit View Terminal Tabs Help

```
1 #include<linux/kernel.h>
2 #include<linux/module.h>
3
4 MODULE_AUTHOR("Mohammad Shehar Yaar Tausif");
5 MODULE_LICENSE("GPL");
6 MODULE_DESCRIPTION("Hello World program with default init and exit");
7
8 int init_module(void)
9 {
10     printk(KERN_INFO "Hello world -1\n");
11     return 0;
12 }
13
14 void cleanup_module(void)
15 {
16     printk(KERN_INFO "Goodbye world -1\n");
17 }
18
```

LINUX HEADER FILES FOR KERNEL MODULES THAT CONTAINS VARIOUS KERNEL FUNCTIONS

VARIOUS MODULE PROPERTIES LIKE AUTHOR, DESCRIPTION, LICENSING, ETC.

INIT FUNCTION (DEFAULT) IS CALLED WHEN A MODULE IS LOADED USING MODPROBE OR INSMOD

CLEANUP FUNCTION (DEFAULT) IS CALLED WHEN A MODULE IS UNLOADED USING MODPROBE OR RMMOD

KERNEL VERSION OF PRINTF.

IT REQUIRES VARIOUS MODES OF MESSAGE LIKE INFO, ALERT, ERROR, ETC.
RETURN 0 DENOTES A SUCCESSFUL RETURN
WHEREAS A NON ZERO RETURNS DENOTES AN ERROR
AND PREVENTS LOADING OF MODULE

File Edit View Terminal Tabs Help

```
1 #include<linux/kernel.h>
2 #include<linux/module.h>
3 #include<linux/init.h> • INIT.H IS INCLUDED IN ORDER TO DECLARE CUSTOM INIT AND EXIT FUNCTIONS
4
5 MODULE_AUTHOR("Mohammad Shehar Yaar Tausif");
6 MODULE_LICENSE("GPL");
7 MODULE_DESCRIPTION("Hello World program with custom init and exit");
8
9 static int __init hello_world(void) • A USER DEFINED FUNCTION WHICH WILL
10 { BE USED AS INIT FUNCTION
11     printk(KERN_INFO "Hello world -2\n");
12     return 0;
13 }
14
15 static void __exit goodbye(void) • A USER DEFINED FUNCTION TO BE USED
16 { AS CLEANUP FUNCTION
17     printk(KERN_INFO "Goodbye world -2\n");
18 }
19
20 module_init(hello_world); • THESE LINES OF CODES ARE USED TO PASS
21 module_exit(goodbye); THE INIT AND EXIT FUNCTIONS TO THE MODULE
```

File Edit View Terminal Tabs Help

```
1 obj-m += hello-1.o
2 obj-m += hello-2.o
3
4 all:
5     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
6 clean:
7     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

TELLS MAKE THAT THE FOLLOWING OBJECT FILES HAVE TO BE BUILT
AND THEY ARE OF TYPE M (MODULES)

THIS WILL BE EXECUTED WHEN
MAKE CLEAN WILL BE CALLED

THIS LINE TELLS MAKE TO CALL USE THE
BUILD FILE LOCATED IN THE GIVEN FOLDER.
M=\$(PWD) TELLS THAT THE LOCATION OF
MODULE IS IN THE PRESENT WORKING
DIRECTORY

Section 4

Loading and Unloading

File Edit View Terminal Tabs Help

```
[lagnos@rog hello]$ make  
make -C /lib/modules/5.10.42-1-MANJARO/build M=/home/lagnos/Desktop/kernel-programming/hello modules  
make[1]: Entering directory '/usr/lib/modules/5.10.42-1-MANJARO/build'  
CC [M] /home/lagnos/Desktop/kernel-programming/hello/hello-1.o  
CC [M] /home/lagnos/Desktop/kernel-programming/hello/hello-2.o  
MODPOST /home/lagnos/Desktop/kernel-programming/hello/Module.symvers  
CC [M] /home/lagnos/Desktop/kernel-programming/hello/hello-1.mod.o  
LD [M] /home/lagnos/Desktop/kernel-programming/hello/hello-1.ko  
CC [M] /home/lagnos/Desktop/kernel-programming/hello/hello-2.mod.o  
LD [M] /home/lagnos/Desktop/kernel-programming/hello/hello-2.ko  
make[1]: Leaving directory '/usr/lib/modules/5.10.42-1-MANJARO/build'
```

```
[lagnos@rog hello]$ ls  
hello-1.c    hello-1.mod      hello-1.mod.o  hello-2.c    hello-2.mod      hello-2.mod.o  Makefile      Module.symvers  
hello-1.ko   hello-1.mod.c   hello-1.o       hello-2.ko   hello-2.mod.c   hello-2.o       modules.order  
[lagnos@rog hello]$
```

CALLING MAKE FROM THE PROGRAM FOLDER

DEBUG STATEMENTS

INTERMEDIATE .O FILES AS WELL AS FINAL .KO FILES ARE CREATED BY MAKE

File Edit View Terminal Tabs Help

```
[lagnos@rog hello]$ modinfo ./hello-1.ko
filename:      /home/lagnos/Desktop/kernel-programming/hello./hello-1.ko
description:   Hello World program with default init and exit
license:       GPL
author:        Mohammad Shehar Yaar Tausif
srcversion:    97B468C174442732C30BEAC
depends:
retpoline:     Y
name:          hello_1
vermagic:      5.10.42-1-MANJARO SMP preempt mod_unload
[lagnos@rog hello]$ modinfo ./hello-2.ko
filename:      /home/lagnos/Desktop/kernel-programming/hello./hello-2.ko
description:   Hello World program with custom init and exit
license:       GPL
author:        Mohammad Shehar Yaar Tausif
srcversion:    E48C3F6F9557D4A0E7A9FB9
depends:
retpoline:     Y
name:          hello_2
vermagic:      5.10.42-1-MANJARO SMP preempt mod_unload
[lagnos@rog hello]$
```

MODULE INFORMATION OF HELLO-1

MODULE INFORMATION FOR HELLO-2

```
File Edit View Terminal Tabs Help
```

```
[lagnos@rog hello]$ sudo insmod ./hello-1.ko
[sudo] password for lagnos:
[lagnos@rog hello]$ sudo insmod ./hello-2.ko
[lagnos@rog hello]$ lsmod
Module           Size  Used by
hello_2          16384  0
hello_1          16384  0
intel_rapl_msrs 20480  0
intel_rapl_common 28672  1 intel_rapl_msrs
intel_powerclamp 20480  0
crc32_pclmul     16384  1
crc32_pclmul     16384  0
ghash_clmulni_intel 16384  0
aesni_intel      372736 0
vboxsf           40960  1
joydev            28672  0
crypto_simd      16384  1 aesni_intel
squashfs          73728  0
rfkill             28672  2
loop               36864  0
cryptd            24576  2 crypto_simd,ghash_clmulni_intel
mousedev          24576  0
glue_helper        16384  1 aesni_intel
rapl               16384  0
vboxvideo          32768  1
drm_vram_helper   20480  1 vboxvideo
drm_ttm_helper    16384  1 drm_vram_helper
snd_intel8x0       49152  2
ttm                114688 2 drm_vram_helper,drm_ttm_helper
snd_ac97_codec    180224  1 snd_intel8x0
ac97_bus           16384  1 snd_ac97_codec
16384  0
```

OUR LOADED MODULE

```
File Edit View Terminal Tabs Help
```

```
Module           Size  Used by
snd              114688 8 snd_intel8x0,snd_timer,snd_ac97_
sysimgblt       16384  1 drm_kms_helper
soundcore        16384  1 snd
fb_sys_fops     16384  1 drm_kms_helper
i2c_piix4       28672  0
mac_hid          16384  0
video            53248  0
drm              569344 7 drm_kms_helper,drm_vram_helper,v
fuse             163840  3
uinput            20480  0
crypto_user      20480  0
agpgart          53248  4 intel_agp,intel_gtt,ttm,drm
ip_tables         32768  0
x_tables          53248  1 ip_tables
ext4             921600 1
crc32c_generic   16384  0
crc16             16384  1 ext4
mbcache           16384  1 ext4
jbd2              147456 1 ext4
sr_mod            28672  0
cdrom             73728  1 sr_mod
usbhid            65536  0
ata_generic       16384  0
pata_acpi         16384  0
crc32c_intel     24576  2
serio_raw          20480  0
ata_piix          36864  0
[lagnos@rog hello]$ lsmod | grep -i hello
hello_2          16384  0
hello_1          16384  0
[lagnos@rog hello]$
```

FINDING OUR MODULE USING
GREP

File Edit View Terminal Tabs Help

```
[lagnos@rog hello]$ sudo rmmod hello-1
[lagnos@rog hello]$ sudo rmmod hello-2
[lagnos@rog hello]$ lsmod | grep -i hello
[lagnos@rog hello]$
```

UNLOADING BOTH THE MODULES

NO OUTPUT DENOTES NO MORE
MODULES LOADED WITH NAME
CONTAINING HELLO

Terminal - lagnos@rog:~/Desktop/kernel-pro

```
File Edit View Terminal Tabs Help  
[lagnos@rog hello]$ sudo dmesg | grep -i world  
[ 1228.214484] Hello world -1  
[ 1235.597134] Hello world -2  
[ 1322.759327] Goodbye world -1  
[ 1327.062006] Goodbye world -2  
[lagnos@rog hello]$ █
```

DMESG LOGS THE KERNEL RING BUFFER
WHICH INCLUDES OUR LOADED MODULE AND
THE MESSAGE ON UNLOADING OF THE MODULE

JOURNALCTL SHOWS ALL THE OCCURENCES
OF THE MODULES PRESENT.
I TRIED LOADING MANY TIMES HENCE SO MANY LINES
PRESENT.

Terminal - lagnos@rog:~/Desktop/kernel-pro

```
File Edit View Terminal Tabs Help  
[lagnos@rog hello]$ sudo journalctl | grep -i world  
Jun 23 14:28:16 rog kernel: Hello World  
Jun 23 14:29:05 rog kernel: Goodbye World  
Jun 23 14:34:14 rog kernel: Hello World  
Jun 23 14:34:34 rog kernel: Hello World  
Jun 23 14:35:00 rog kernel: Hello World  
Jun 23 14:35:03 rog kernel: Hello World  
Jun 23 14:35:08 rog kernel: Hello World  
Jun 23 14:35:13 rog kernel: Hello World  
Jun 23 14:35:54 rog kernel: Hello World  
Jun 23 14:36:15 rog kernel: Hello World  
Jun 23 14:36:17 rog kernel: Hello World  
Jun 23 14:36:17 rog kernel: Hello World  
Jun 23 14:36:18 rog kernel: Hello World  
Jun 23 14:36:18 rog kernel: Hello World  
Jun 23 14:36:18 rog kernel: Hello World  
Jun 23 14:39:47 rog kernel: Hello World  
Jun 23 14:40:45 rog kernel: Hello World  
Jun 23 14:45:24 rog kernel: Hello World  
Jun 23 14:46:31 rog kernel: Hello World  
Jun 23 14:46:43 rog kernel: Goodbye World  
Jun 23 15:22:40 rog kernel: Hello World with init  
Jun 23 15:23:07 rog kernel: Good Bye world from exit  
Jun 23 16:09:35 rog kernel: Hello World!  
Jun 23 16:10:00 rog kernel: Good Bye World!  
Jun 23 18:45:37 rog kernel: Hello world -1  
Jun 23 18:46:26 rog kernel: Goodbye world -1  
Jun 23 18:51:48 rog kernel: Hello world -1  
Jun 23 18:52:19 rog kernel: Goodbye world -1  
Jun 24 12:42:44 rog kernel: Hello world -1  
Jun 24 12:42:48 rog kernel: Hello world -2  
Jun 24 12:42:55 rog kernel: Goodbye world -1  
Jun 24 12:42:58 rog kernel: Goodbye world -2  
Jun 24 14:14:59 rog kernel: Hello world -1  
Jun 24 14:15:07 rog kernel: Hello world -2  
Jun 24 14:16:34 rog kernel: Goodbye world -1  
Jun 24 14:16:38 rog kernel: Goodbye world -2  
[lagnos@rog hello]$ █
```

Terminal - lagnos@rog:~/...

MADE WITH

beautiful.ai

File Edit View Terminal Tabs Help

[lagnos@rog hello]\$ ls

hello-1.c hello-1.mod hello-1.mod.o hello-2.c hello-2.mod hello-2.mod.o Makefile Module.symvers
hello-1.ko hello-1.mod.c hello-1.o hello-2.ko hello-2.mod.c hello-2.o modules.order

[lagnos@rog hello]\$ make clean

make -C /lib/modules/5.10.42-1-MANJARO/build M=/home/lagnos/Desktop/kernel-programming/hello clean
make[1]: Entering directory '/usr/lib/modules/5.10.42-1-MANJARO/build'

CLEAN /home/lagnos/Desktop/kernel-programming/hello/Module.symvers

make[1]: Leaving directory '/usr/lib/modules/5.10.42-1-MANJARO/build'

[lagnos@rog hello]\$ ls

hello-1.c hello-2.c Makefile

[lagnos@rog hello]\$ █

ALL THE FILES THAT WERE CREATED BY
MAKE HAVE BEEN REMOVED

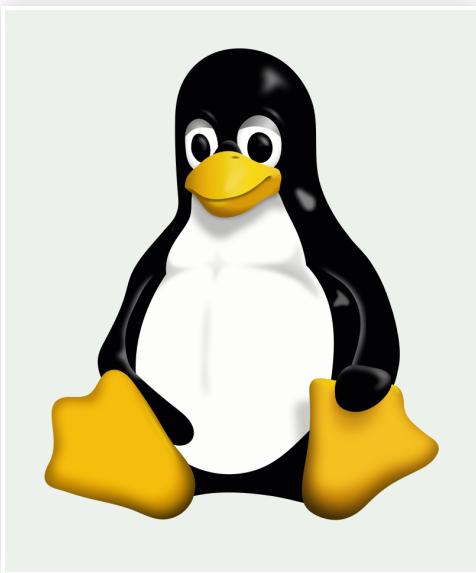
NOW WE RUN *MAKE CLEAN* WHICH WAS
WRITTEN IN MAKEFILE

THE OBJECTIVE FILES ARE PRESENT EVEN
AFTER UNLOADING THE MODULE

Now you can spread your wings and fly...

Thus we conclude the introductory session.

Codes used in the slides are present at : <https://github.com/sheharyaar/koss-kernel>



1 KERNEL

- Linux Kernel
- Things to do before beginning LKM programming

2 KERNEL MODULES

- Building of kernel
- Use of makefiles
- Command line tools - *lsmod*, *modinfo*, *insmod*, *rmmod*, *modprobe*
- Logging programs - *dmesg*, *journalctl*

3 CODING MODULES

- Hello world code with default init and cleanup calls
- Hello world code with user defined init and cleanup functions
- Makefile

4 LOADING/UNLOADING MODULES

- Building module using *make*
- Using *modinfo* to view information of the modules
- Using *insmod* to load and *lsmod* to list
- Using *rmmod* to unload
- Using *dmesg* and *journalctl* to verify our module logs.



{Thank You}

This is Mohammad Shehar Yaar Tausif

Follow me on Github : <https://github.com/sheharyaar/>

@ sheharyaar48@gmail.com

 https://twitter.com/lagnos_andras

 +91-9934180769