

Tutorial para o desenvolvimento de módulo USB para o kernel do Linux

Neste tutorial será mostrado como criar e inserir um módulo USB para ler os dados do pen drive inserido. Abaixo é possível visualizar o código fonte do módulo cujo o nome atribuído foi **stick_driver.c**. Na struct `usb_device_id` `pen_table` é necessário que se saiba o `idVendor` e o `idProduct` do pen_drive que pode ser descoberto inserindo o pen_drive, digitando o comando **lsusb**. Após isso basta inserir os valores em `USB_DEVICE` no código dentro da struct `usb_device_id` `pen_table`.

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

//funcao sonda
//chamada na insercao do dispositivo se e somente se nenhum outro driver nos precedeu
static int pen_probe (struct usb_interface *interface , const struct usb_device_id *id){
    printk(KERN_INFO "[%] Pen Drive (%04X:%04X) Plugged\n" , id->idVendor , id->idProduct);
    return 0;
}

//disconecta
static void pen_disconnect(struct usb_interface *interface){
    printk(KERN_SOH "[%] Pen drive removido\n");
}

//usb_device_id
static struct usb_device_id pen_table[] = {
    //1221:3234 idVendor e idProduct
    {USB_DEVICE(0x1221, 0x3234)}, //informacao obtida usando "lsusb" no terminal
    {} /*entrada terminal*/
};
MODULE_DEVICE_TABLE(usb, pen_table);

// usb_driver
static struct usb_driver pen_driver = {
    .name = "Henrique-USB Stick Driver",
    .id_table = pen_table, //usb_device_id
    .probe = pen_probe,
    .disconnect = pen_disconnect,
};

static int __init pen_init(void){
    int ret = -1;
    printk(KERN_SOH "[%] construtor do driver");
    printk(KERN_SOH "\tRegistrando driver com kernel");
    ret = usb_register(&pen_driver);
}
```

```
static struct usb_device_id pen_table[] = {
    //1221:3234 idVendor e idProduct
    {USB_DEVICE(0x1221, 0x3234)}, //informacao obtida usando "lsusb" no terminal
    {} /*entrada terminal*/
};
MODULE_DEVICE_TABLE(usb, pen_table);

// usb_driver
static struct usb_driver pen_driver = {
    .name = "Henrique-USB Stick Driver",
    .id_table = pen_table, //usb_device_id
    .probe = pen_probe,
    .disconnect = pen_disconnect,
};

static int __init pen_init(void){
    int ret = -1;
    printk(KERN_SOH "[%] construtor do driver");
    printk(KERN_SOH "\tRegistrando driver com kernel");
    ret = usb_register(&pen_driver);
    printk(KERN_SOH "\tRegistro completo");
    return ret;
}

static void __exit pen_exit(void){
    //deregister
    printk(KERN_SOH "[%] Desconstrutor do driver");
    usb_deregister(&pen_driver);
    printk(KERN_SOH "\tregistro cancelado!");
}

module_init(pen_init);
module_exit(pen_exit);

MODULE_LICENSE("GPL");
MODULE_AUTHOR("HenriqueAnnicchino");
MODULE_DESCRIPTION("USB Pen Drive registration");
```

Abaixo é possível visualizar o **Makefile** utilizado.

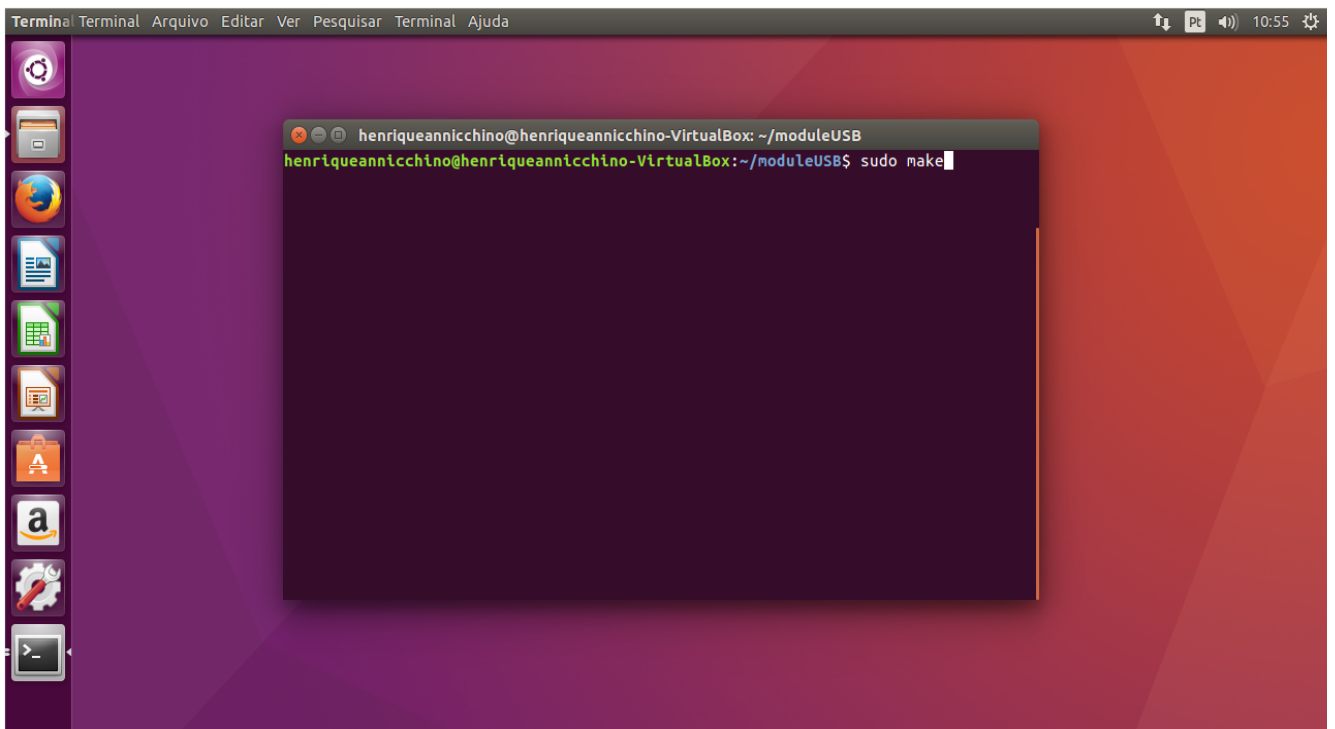
```
obj-m := stick_driver.o

KERNEL_DIR = /lib/modules/$(shell uname -r)/build
PWD = $(shell pwd)

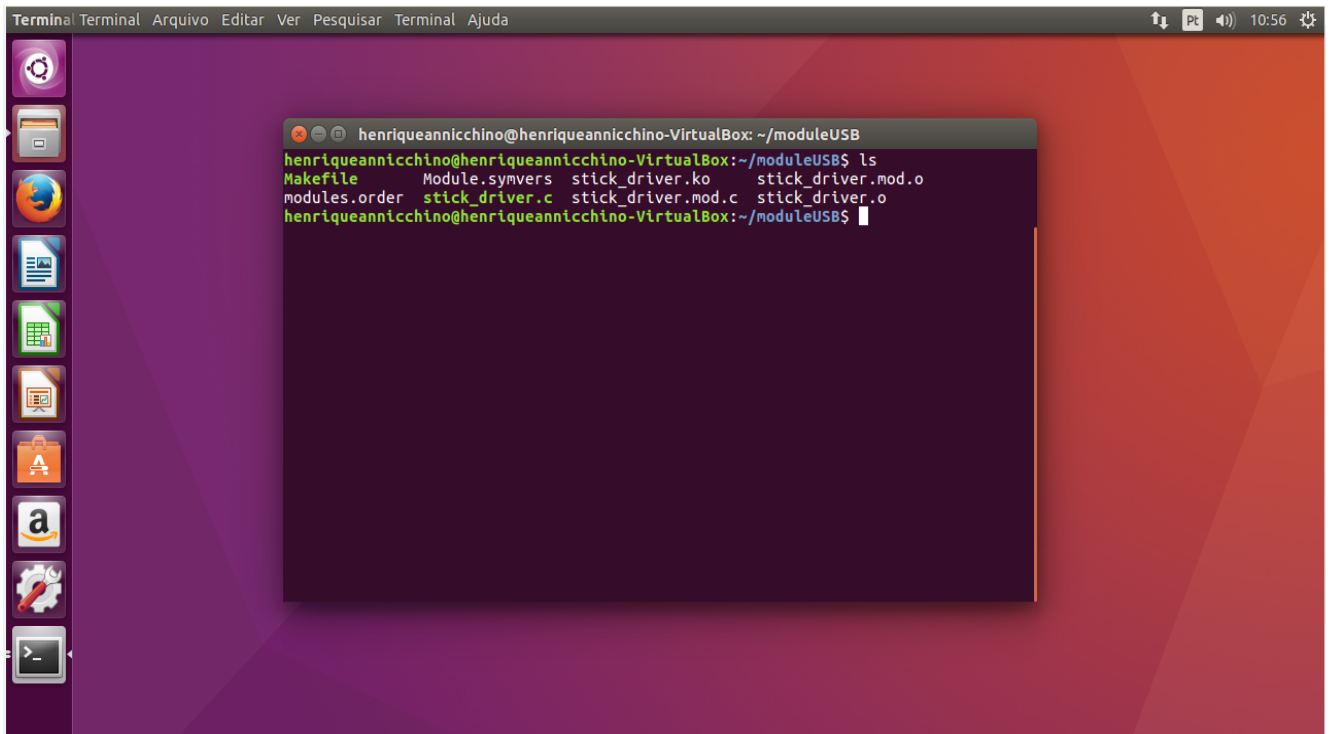
all:
    $(MAKE) -C $(KERNEL_DIR) SUBDIRS=$(PWD) modules

clean:
    rm -rf *.o *.ko *.mod.* *.symvers *.order *
```

Após já ter criado ambos os arquivos citados acima basta usar o comando **sudo make** no diretório onde os mesmos se encontram.

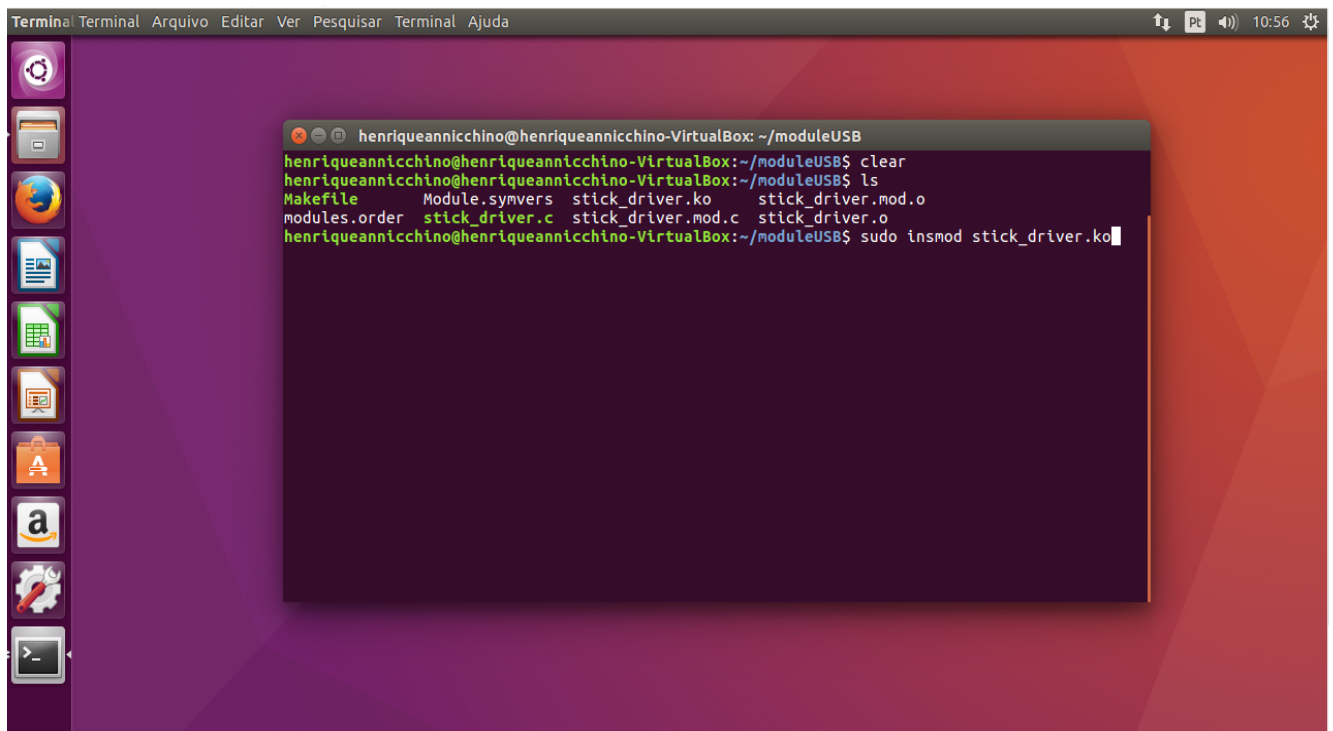


Caso o comando tenha sido executado corretamente será possível mais alguns arquivos no diretório usando o comando **ls** como pode ser visto abaixo.



```
henriqueannicchino@henriqueannicchino-VirtualBox: ~/moduleUSB
henriqueannicchino@henriqueannicchino-VirtualBox:~/moduleUSB$ ls
Makefile      Module.symvers  stick_driver.ko  stick_driver.mod.o
modules.order  stick_driver.c  stick_driver.mod.c  stick_driver.o
henriqueannicchino@henriqueannicchino-VirtualBox:~/moduleUSB$
```

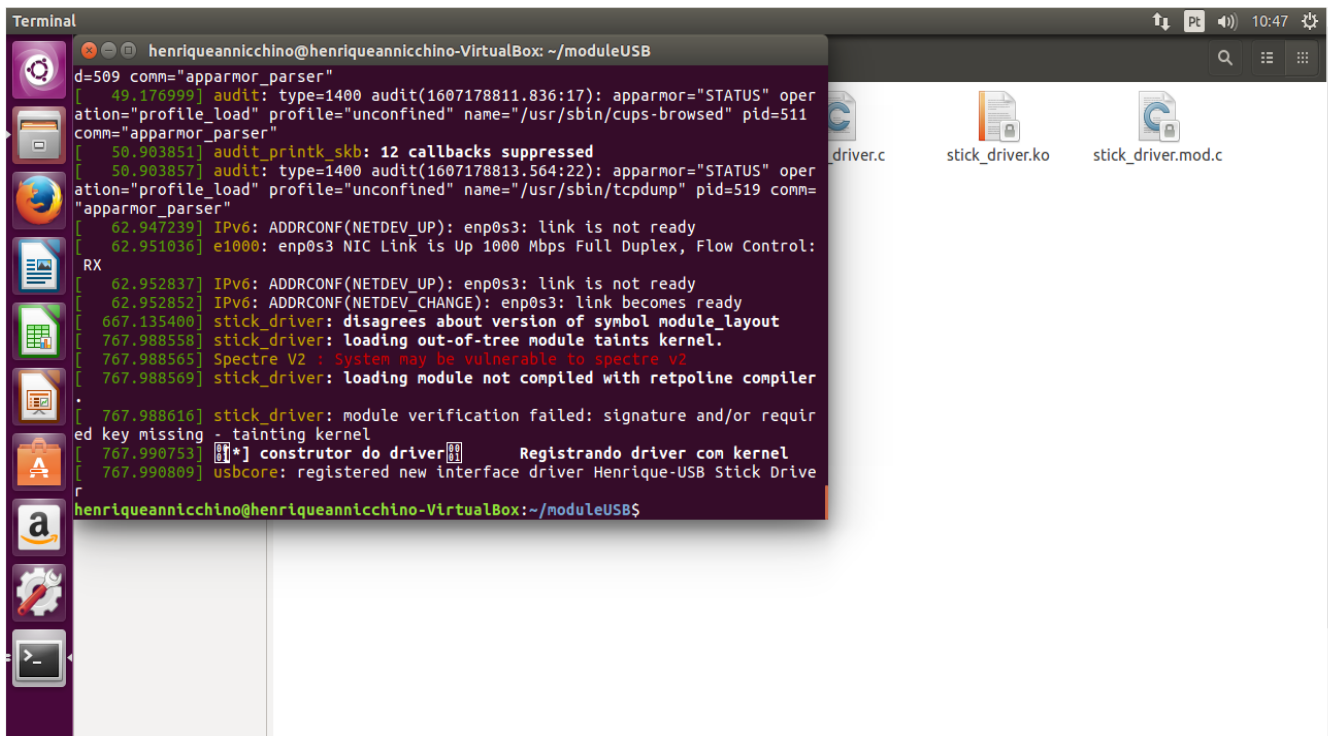
Para inserir o módulo é necessário utilizar o comando **sudo insmod (nome do arquivo).ko**



```
henriqueannicchino@henriqueannicchino-VirtualBox: ~/moduleUSB
henriqueannicchino@henriqueannicchino-VirtualBox:~/moduleUSB$ clear
henriqueannicchino@henriqueannicchino-VirtualBox:~/moduleUSB$ ls
Makefile      Module.symvers  stick_driver.ko  stick_driver.mod.o
modules.order  stick_driver.c  stick_driver.mod.c  stick_driver.o
henriqueannicchino@henriqueannicchino-VirtualBox:~/moduleUSB$ sudo insmod stick_driver.ko
```

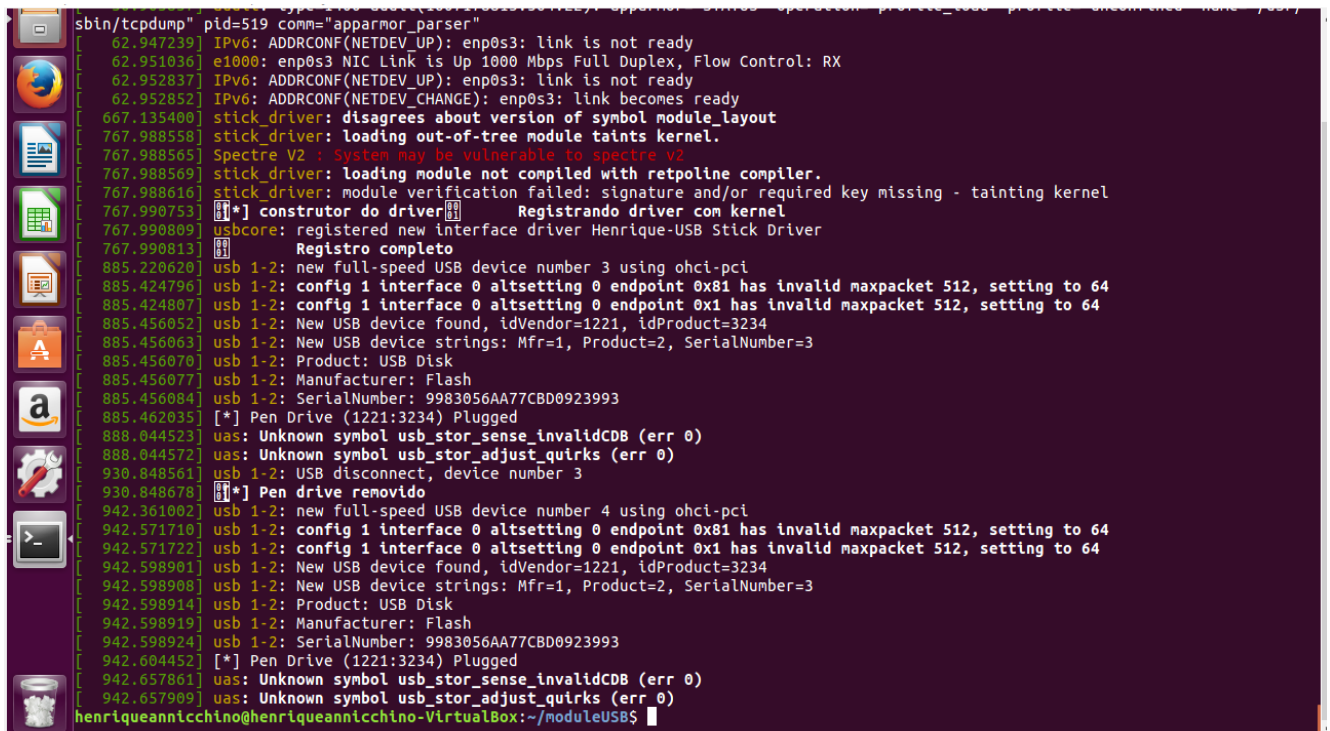
Para visualizar se o módulo carregado corretamente basta utilizar o comando **dmesg**.

OBS: Para que o módulo funcione como esperado é preciso remover o módulo `usb_storage` utilizando o comando **sudo rmmod usb_storage**.



```
henriqueannicchino@henriqueannicchino-VirtualBox: ~/moduleUSB
d=509 comm="apparmor_parser"
[ 49.176999] audit: type=1400 audit(1607178811.836:17): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/sbin/cups-browsed" pid=511 comm="apparmor_parser"
[ 50.903851] audit_printk_skb: 12 callbacks suppressed
[ 50.903857] audit: type=1400 audit(1607178813.564:22): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/sbin/tcpdump" pid=519 comm="apparmor_parser"
[ 62.947239] IPv6: ADDRCONF(NETDEV_UP): enp0s3: link is not ready
[ 62.951036] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 62.952837] IPv6: ADDRCONF(NETDEV_UP): enp0s3: link is not ready
[ 62.952852] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 667.135400] stick_driver: disagrees about version of symbol module_layout
[ 767.988558] stick_driver: loading out-of-tree module taints kernel.
[ 767.988565] Spectre V2 : System may be vulnerable to spectre v2
[ 767.988569] stick_driver: loading module not compiled with retpoline compiler
[ 767.988616] stick_driver: module verification failed: signature and/or required key missing - tainting kernel
[ 767.990753] [*] construtor do driver[*] Registrando driver com kernel
[ 767.990809] usbcore: registered new interface driver Henrique-USB Stick Drive
henriqueannicchino@henriqueannicchino-VirtualBox: ~/moduleUSB$
```

Na imagem abaixo é possível visualizar as mensagens exibidas pela módulo ao inserir, remover e inserir novamente um pen_drive que possua idVendor e idProduct listados no código.



```
sbins/tcpdump" pid=519 comm="apparmor_parser"
[ 62.947239] IPv6: ADDRCONF(NETDEV_UP): enp0s3: link is not ready
[ 62.951036] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 62.952837] IPv6: ADDRCONF(NETDEV_UP): enp0s3: link is not ready
[ 62.952852] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 667.135400] stick_driver: disagrees about version of symbol module_layout
[ 767.988558] stick_driver: loading out-of-tree module taints kernel.
[ 767.988565] Spectre V2 : System may be vulnerable to spectre v2
[ 767.988569] stick_driver: loading module not compiled with retpoline compiler.
[ 767.988616] stick_driver: module verification failed: signature and/or required key missing - tainting kernel
[ 767.990753] [*] construtor do driver[*] Registrando driver com kernel
[ 767.990809] usbcore: registered new interface driver Henrique-USB Stick Driver
[ 767.990813] Registro completo
[ 885.220620] usb 1-2: new full-speed USB device number 3 using ohci-pci
[ 885.424796] usb 1-2: config 1 interface 0 altsetting 0 endpoint 0x81 has invalid maxpacket 512, setting to 64
[ 885.424807] usb 1-2: config 1 interface 0 altsetting 0 endpoint 0x1 has invalid maxpacket 512, setting to 64
[ 885.456052] usb 1-2: New USB device found, idVendor=1221, idProduct=3234
[ 885.456063] usb 1-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 885.456070] usb 1-2: Product: USB Disk
[ 885.456077] usb 1-2: Manufacturer: Flash
[ 885.456084] usb 1-2: SerialNumber: 9983056AA77C8D0923993
[ 885.462035] [*] Pen Drive (1221:3234) Plugged
[ 888.044523] uas: Unknown symbol usb_stor_sense_invalidCDB (err 0)
[ 888.044572] uas: Unknown symbol usb_stor_adjust_quirks (err 0)
[ 930.848561] usb 1-2: USB disconnect, device number 3
[ 930.848678] [*] Pen drive removido
[ 942.361002] usb 1-2: new full-speed USB device number 4 using ohci-pci
[ 942.571710] usb 1-2: config 1 interface 0 altsetting 0 endpoint 0x81 has invalid maxpacket 512, setting to 64
[ 942.571722] usb 1-2: config 1 interface 0 altsetting 0 endpoint 0x1 has invalid maxpacket 512, setting to 64
[ 942.598901] usb 1-2: New USB device found, idVendor=1221, idProduct=3234
[ 942.598908] usb 1-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 942.598914] usb 1-2: Product: USB Disk
[ 942.598919] usb 1-2: Manufacturer: Flash
[ 942.598924] usb 1-2: SerialNumber: 9983056AA77C8D0923993
[ 942.604452] [*] Pen Drive (1221:3234) Plugged
[ 942.657861] uas: Unknown symbol usb_stor_sense_invalidCDB (err 0)
[ 942.657909] uas: Unknown symbol usb_stor_adjust_quirks (err 0)
henriqueannicchino@henriqueannicchino-VirtualBox: ~/moduleUSB$
```