

# **Сборка и установка драйверов в ОС Linux**

**Драйвер не может быть собран и установлен в WSL!**

## **Установка необходимых пакетов**

В виртуальной машине (гостевой ОС) обновить списки пакетов в репозиториях:

```
sudo apt update
```

Опционально можно обновить установленные в ОС пакеты:

```
sudo apt upgrade
```

Далее необходимо установить заголовочные файлы ядра:

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

И утилиты **make** и **gcc**:

```
sudo apt install gcc  
sudo apt install make
```

## Исходный код простейшего драйвера

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

MODULE_LICENSE("Dual MIT/GPL");

int __init test_drv_init(void)
{
    printk("test_drv_init called\n");
    return 0;
}

void __exit test_drv_exit(void)
{
    printk("test_drv_exit called\n");
}

module_init(test_drv_init);
module_exit(test_drv_exit);
```

## Сборка драйвера

Сборку драйвера удобно осуществлять при помощи *Makefile*. Для этого в директории с файлами исходного кода драйвера необходимо создать файл с именем *Makefile* и следующим содержанием:

```
TARGET := DRIVER_NAME
PWD := $(shell pwd)
KERNEL := $(shell uname -r)
obj-m := $(TARGET).o
$(TARGET)-objs += drv_src.o
all:
    make -C /lib/modules/$(KERNEL)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(KERNEL)/build M=$(PWD) clean
```

где *DRIVER\_NAME* — имя исполняемого файла собираемого драйвера

*drv\_src* — имя файла исходного кода драйвера

*drv\_src.o* — имя объектного файла, соответствующего файлу исходного кода драйвера

При наличии в проекте нескольких файлов исходного кода, необходимо добавить имена соответствующих им объектных файлов в параметр *\$(TARGET)-objs*. Например, при наличии в проекте помимо дополнительного файла исходного кода с именем *fun.c*, изменить в *Makefile* следующую строку:

```
$(TARGET)-objs += drv_src.o fun.o
```

Для сборки драйвера в директории с файлами исходного кода и *Makefile*'ом необходимо выполнить команду:

```
make all
```

Собранный исполняемый файл драйвера имеет имя ***DRIVER\_NAME*** и расширение ***.ko***.

Получить информацию об исполняемом файле драйвера можно при помощи команды:

```
modinfo DRIVER_NAME.ko
```

где ***DRIVER\_NAME*** – имя собранного драйвера.

Для удаления собранных файлов (исполняемого файла драйвера и созданных служебных файлов) можно воспользоваться командой:

```
make clean
```

## Загрузка и выгрузка драйвера

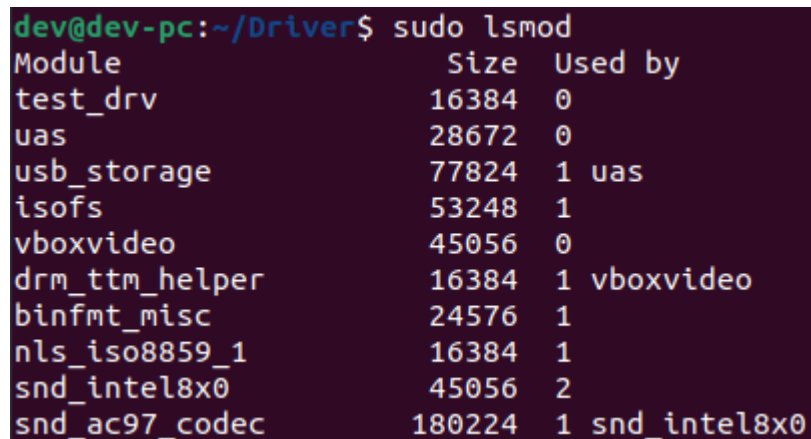
Для установки (загрузки в память) драйвера из директории с исполняемым файлом драйвера необходимо выполнить команду:

```
sudo insmod DRIVER_NAME.ko
```

где ***DRIVER\_NAME*** – имя собранного драйвера.

Список установленных модулей режима ядра можно получить при помощи команды:

```
sudo lsmod
```



```
dev@dev-pc:~/Driver$ sudo lsmod
Module                  Size  Used by
test_drv                16384  0
uas                    28672  0
usb_storage             77824  1 uas
isofs                   53248  1
vboxvideo               45056  0
drm_ttm_helper          16384  1 vboxvideo
binfmt_misc             24576  1
nls_iso8859_1           16384  1
snd_intel8x0            45056  2
snd_ac97_codec          180224 1 snd_intel8x0
```

Для выгрузки драйвера из памяти необходимо выполнить команду:

```
sudo rmmod DRIVER_NAME
```

## Просмотр отладочных сообщений драйвера

Просмотреть отладочные сообщения драйвера, выводимые функцией `printk` можно при помощи команды:

```
sudo dmesg
```

```
[ 319.343947] test_drv_init called  
[ 345.287127] test_drv_exit called
```

Для форматированного вывода отладочных сообщений с именем драйвера, в исходный код необходимо добавить следующие макросы:

```
#define DEVICE_NAME "MyTestDevice"  
#define DBGMSG(fmt, ...) printk(DEVICE_NAME": "fmt,##__VA_ARGS__)
```

Пример использования:

```
...  
int __init test_drv_init(void)  
{  
    DBGMSG("test_drv_init called\n");  
  
    return 0;  
}  
  
void __exit test_drv_exit(void)  
{  
    DBGMSG("test_drv_exit called\n");  
}  
...
```

Пример отладочного сообщения:

```
[ 588.225859] MyTestDevice: test_drv_init called  
[ 591.999667] MyTestDevice: test_drv_exit called
```