

Попытка спользования в md2 функции, определённой в md1

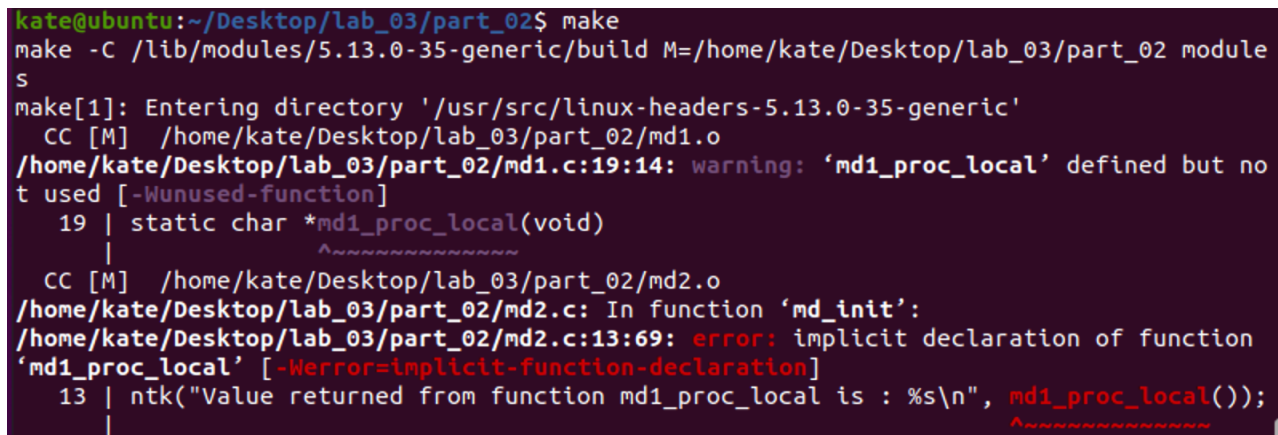
При попытке использования локальной функции md1 в md2 возникает ошибка компиляции md2, так как в md2 нет ни объявления, ни определения функции (рис. 1).

Листинг 1: md1.c

```
1 #include ...
2
3 static char *md1_proc_local(void)
4 {
5     return md1_data;
6 }
7 ...
```

Листинг 2: md2.c

```
1 #include <..>
2 static int __init md_init(void) {
3
4     printk("Module md2 was loaded\n");
5     printk("Value returned from function md1_proc_local is : %s\n",
6         md1_proc_local());
7     return 0;
8 }
```



```
kate@ubuntu:~/Desktop/lab_03/part_02$ make
make -C /lib/modules/5.13.0-35-generic/build M=/home/kate/Desktop/lab_03/part_02 module
s
make[1]: Entering directory '/usr/src/linux-headers-5.13.0-35-generic'
CC [M] /home/kate/Desktop/lab_03/part_02/md1.o
/home/kate/Desktop/lab_03/part_02/md1.c:19:14: warning: 'md1_proc_local' defined but no
t used [-Wunused-function]
   19 | static char *md1_proc_local(void)
      |
CC [M] /home/kate/Desktop/lab_03/part_02/md2.o
/home/kate/Desktop/lab_03/part_02/md2.c: In function 'md_init':
/home/kate/Desktop/lab_03/part_02/md2.c:13:69: error: implicit declaration of function
'md1_proc_local' [-Werror=implicit-function-declaration]
   13 | ntk("Value returned from function md1_proc_local is : %s\n", md1_proc_local());
      |
```

Рис. 1: Результат

Попытка экспортировать функцию из md1 в md2 без использования EXPORT_SYMBOL

Без EXPORT_SYMBOL функция из md1 не была экспортирована в md2, так как на шаге MODPOST возникла ошибка отсутствия определения этой функции (рис. 3).

Из документации про MODPOST:

--- 6.2 Symbols and External Modules

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. modpost obtains the symbols by reading Module.symvers from the kernel source tree. If a Module.symvers file is present in the directory where the external module is being built, this file will be read too. During the MODPOST step, a new Module.symvers file will be written containing all exported symbols that were not defined in the kernel.

Рис. 2: MODPOST step

Таким образом, на шаге MODPOST проверяется, определены ли все внешние символы (система сборки обращается к символам из ядра). Так как макрос EXPORT_SYMBOL не был применён к символу md1_proc_noexport, этот символ не попал в локальный Module.symvers файл, а значит не попал и в соответствующий файл ядра, поэтому проверка определения этого символа провалилась.

```
kate@ubuntu:~/Desktop/lab_03/part_02$ make
make -C /lib/modules/5.13.0-35-generic/build M=/home/kate/Desktop/lab_03/part_02
modules
make[1]: Entering directory '/usr/src/linux-headers-5.13.0-35-generic'
CC [M] /home/kate/Desktop/lab_03/part_02/md1.o
MODPOST /home/kate/Desktop/lab_03/part_02/Module.symvers
ERROR: modpost: "md1_proc_noexport" [/home/kate/Desktop/lab_03/part_02/md2.ko] u
ndefined!
make[2]: *** [scripts/Makefile.modpost:150: /home/kate/Desktop/lab_03/part_02/Mo
dule.symvers] Error 1
make[2]: *** Deleting file '/home/kate/Desktop/lab_03/part_02/Module.symvers'
make[1]: *** [Makefile:1794: modules] Error 2
make[1]: Leaving directory '/usr/src/linux-headers-5.13.0-35-generic'
make: *** [Makefile:14: default] Error 2
kate@ubuntu:~/Desktop/lab_03/part_02$
```

Рис. 3: Результат

Листинг 3: md.h

```
1 extern char *md1_proc_noexport(void);
```

Листинг 4: md1.c

```
1 #include "md.h"
2 #include ...
3
4 char *md1_proc_noexport(void)
5 {
6     return md1_data;
7 }
8 ...
```

Листинг 5: md2.c

```
1 #include "md.h"
2 <..>
3 static int __init md_init(void) {
4
5     printk("Module md2 was loaded\n");
6     printk("Value returned from function md1_proc_noexport is : %s\n",
7         md1_proc_noexport());
8     return 0;
9 }
```

Использование в md2 данных и функций, экспортированных с помощью EXPORT_SYMBOL из md1

При использовании EXPORT_SYMBOL данные и функция были успешно экспортированы из md1 в md2, что подтверждается выводом в лог соответствующих сообщений (рис. 5).

Листинг 6: md.h

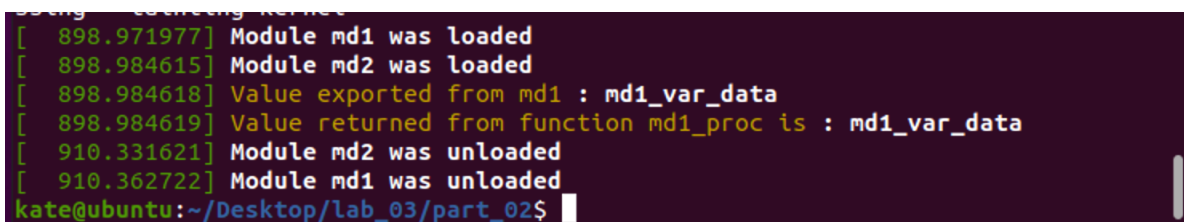
```
1 extern char *md1_data;  
2 extern char *md1_proc(void);
```

Листинг 7: md1.c

```
1 #include "md.h"  
2 #include ...  
3  
4 char *md1_data = "md1_var_data";  
5  
6 char *md1_proc(void)  
7 {  
8     return md1_data;  
9 }  
10 EXPORT_SYMBOL(md1_data);  
11 EXPORT_SYMBOL(md1_proc);  
12  
13 ...
```

Листинг 8: md2.c

```
1 #include "md.h"  
2 <..>  
3 static int __init md_init(void) {  
4  
5     printk("Module md2 was loaded\n");  
6     printk("Value exported from md1 : %s\n", md1_data);  
7     printk("Value returned from function md1_proc is : %s\n", md1_proc());  
8     return 0;  
9 }  
10 ...
```



```
Building matching kernel  
[ 898.971977] Module md1 was loaded  
[ 898.984615] Module md2 was loaded  
[ 898.984618] Value exported from md1 : md1_var_data  
[ 898.984619] Value returned from function md1_proc is : md1_var_data  
[ 910.331621] Module md2 was unloaded  
[ 910.362722] Module md1 was unloaded  
kate@ubuntu:~/Desktop/lab_03/part_02$
```

Рис. 4: Результат

__init возвращает -1

В md3 используется код md2 из предыдущего пункта с изменением возвращаемого значения функции md_init с 0 на -1, md1.c и md.h остались такими же. Сборка модулей прошла успешно, однако при загрузке возникла ошибка (рис. ??).

```
kate@ubuntu:~/Desktop/lab_03/part_02$ make load
sudo insmod md1.ko
sudo insmod md2.ko
sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Operation not permitted
make: *** [Makefile:26: load] Error 1
kate@ubuntu:~/Desktop/lab_03/part_02$
```

Рис. 5: Результат