|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**ОТЧЕТ**

*к лабораторной работе №3*

*По курсу: «Операционные системы»*

*На тему: «Загружаемые модули ядра»*

Студент ИУ7-66Б

Нечитайло Д.В.

Преподаватель

Рязанова Н.Ю.

*Москва, 2021 г.*

**Часть 1**

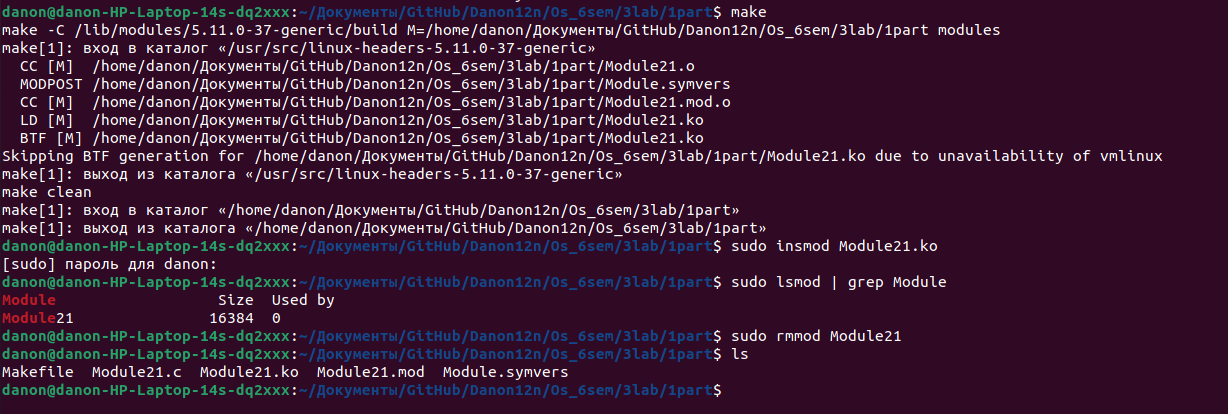
**Условие:**

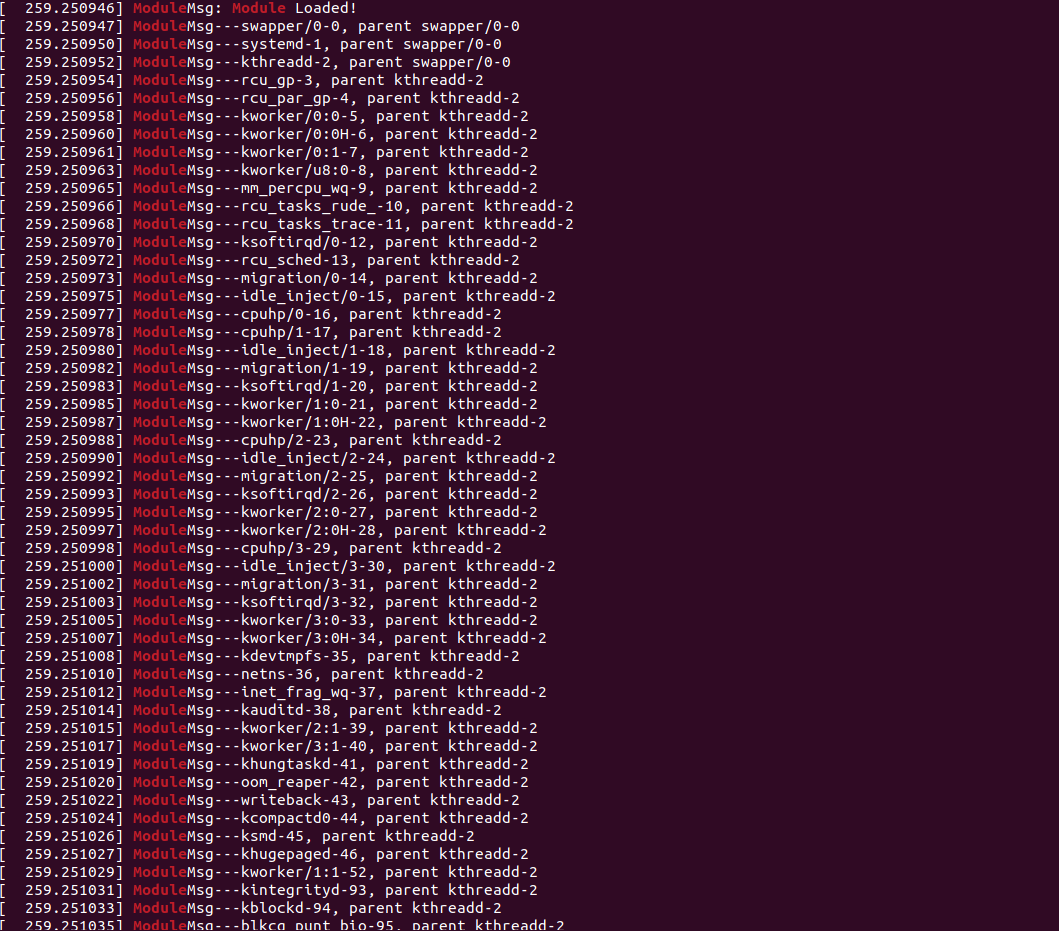
Реализовать загружаемый модуль ядра, который при загрузке записывает в системный журнал информацию о процессах. О каждом процессе запи-сать: название, идентификатор, название предка, идентификатор предка. Модуль должен собираться при помощи make файла.

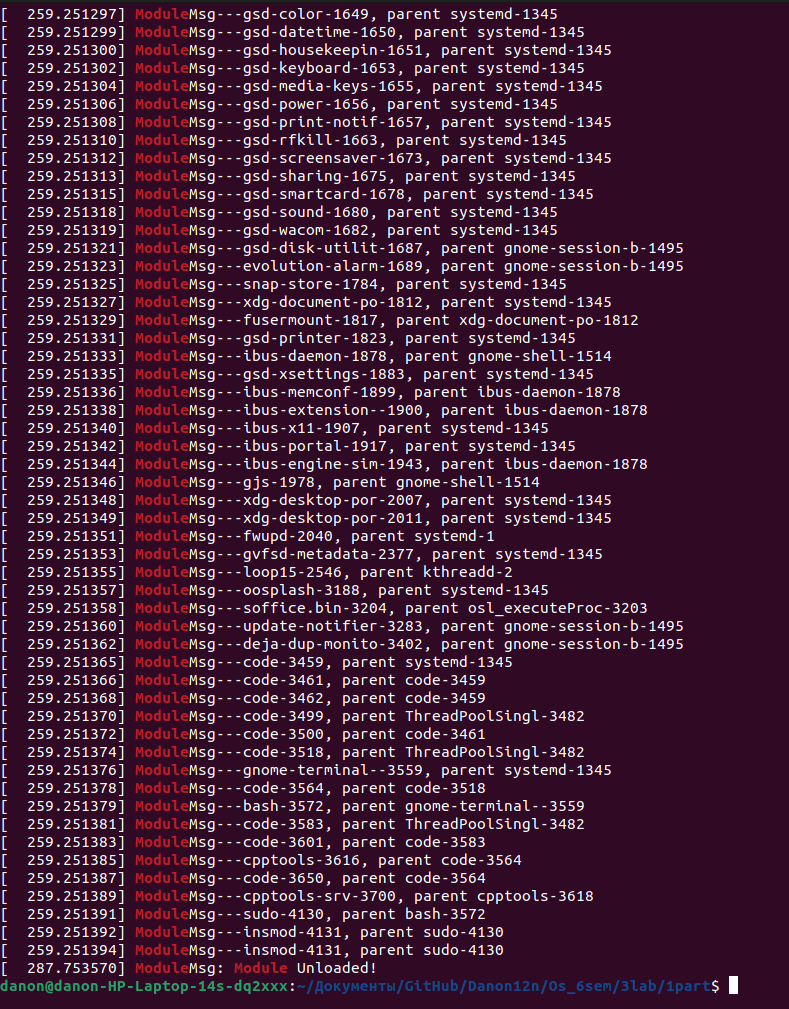
**Листинг кода:**

|  |
| --- |
| #include <linux/module.h>  #include <linux/kernel.h>  #include <linux/init.h>  #include <linux/sched.h>  #include <linux/init\_task.h>  MODULE\_LICENSE("GPL");  MODULE\_AUTHOR("Nechitaylo D.V.");  MODULE\_DESCRIPTION("Module Descrioption");  struct task\_struct \*task = &init\_task;  static int \_\_init Module\_init(void)  {  printk("ModuleMsg: Module Loaded!\n");    do  {  printk("ModuleMsg---%s-%d, parent %s-%d", task->comm,  task->pid, task->parent->comm, task->parent->pid);  } while ((task = next\_task(task)) != &init\_task);  printk("ModuleMsg---%s-%d, parent %s-%d", current->comm,  current->pid, current->parent->comm, current->parent->pid);  return 0;  }  static void \_\_exit Module\_exit(void)  {  printk("ModuleMsg: Module Unloaded!\n");  }  module\_init(Module\_init);  module\_exit(Module\_exit); |

**Результат работы программы**

Компиляция с помощью makefile, загрузка модуля в ядро (insmod), просмотр списка загруженных модулей (lsmod), выгрузка модуля из ядра (rmmod): 

**Вывод модуля ядра:** 



**Часть 2**

**Условие:**

Реализовать три загружаемых модуля ядра:

* Вызываемый модуль md1
* Вызывающий модуль md2
* «Отладочный» модуль md3

Каждый загружаемый модуль должен содержать указание лицензии GPL и указание автора.

Загружаемые модули должны собираться при помощи Make-файла (сборка командой make). Вызов каждой функции модуля должен сопровождаться записью в системный журнал информации, какая функция какого модуля была вызвана.

**Листинг кода:**

**Module31.c**

|  |
| --- |
| #include <linux/init.h>  #include <linux/module.h>  #include "Module.h"  MODULE\_LICENSE("GPL");  MODULE\_AUTHOR("Nechitaylo D.V.");  char\* Module31\_string = "Hello world!";  int Module31\_int = 5;  extern char\* Module31\_proc1(void)  {  return Module31\_string;  }  extern int Module31\_proc2(void)  {  return Module31\_int \* Module31\_int;  }  static char\* Module31\_local(void)  {  return Module31\_string;  }  extern char\* Module31\_noexport(void)  {  return Module31\_string;  }  EXPORT\_SYMBOL(Module31\_string);  EXPORT\_SYMBOL(Module31\_int);  EXPORT\_SYMBOL(Module31\_proc1);  EXPORT\_SYMBOL(Module31\_proc2);  static int \_\_init Module31\_init(void)  {  printk("ModuleMsg: Module31 Loaded!\n");  return 0;  }  static void \_\_exit Module31\_exit(void)  {  printk("ModuleMsg: Module31 Unloaded!\n");  }  module\_init(Module31\_init);  module\_exit(Module31\_exit); |

**Module32.c**

|  |
| --- |
| #include <linux/init.h>  #include <linux/module.h>  #include "Module.h"  MODULE\_LICENSE("GPL");  MODULE\_AUTHOR("Nechitaylo D.V.");  static int \_\_init Module32\_init(void)  {  printk("ModuleMsg: Module32 Loaded!\n");  printk("ModuleMsg: string exported from Module31 : %s\n", Module31\_string);  printk("ModuleMsg: string returned Module31\_proc1() is : %s\n", Module31\_proc1());  printk("ModuleMsg: int exported from Module31 : %d\n", Module31\_int);  printk("ModuleMsg: int returned by Module31\_proc2 is : %d", Module31\_proc2());  return 0;  }  static void \_\_exit Module32\_exit(void)  {  printk("ModuleMsg: Module32 Unloaded!\n");  }  module\_init(Module32\_init);  module\_exit(Module32\_exit); |

**Module33.c**

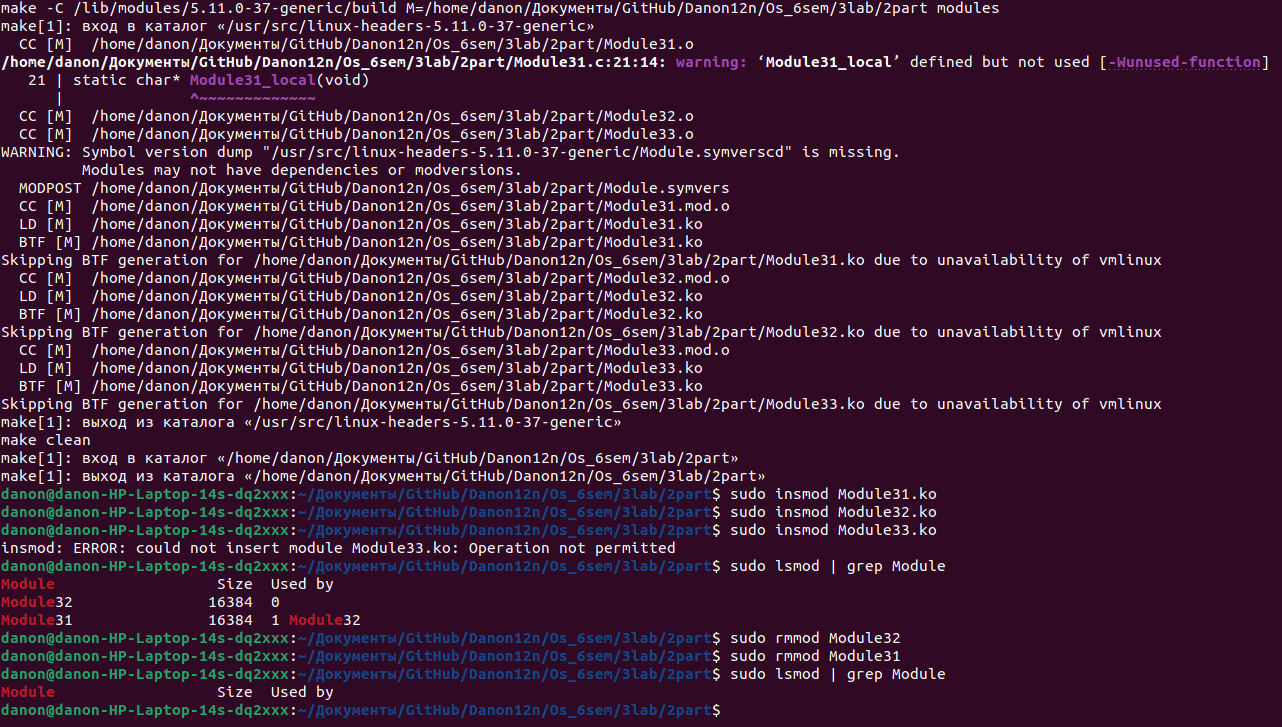
|  |
| --- |
| #include <linux/init.h>  #include <linux/module.h>  #include "Module.h"  MODULE\_LICENSE("GPL");  MODULE\_AUTHOR("Nechitaylo D.V.");  static int \_\_init Module32\_init(void)  {  printk("ModuleMsg: Module33 Loaded!\n");  printk("ModuleMsg: string exported from Module31 : %s\n", Module31\_string);  printk("ModuleMsg: string returned Module31\_proc1() is : %s\n", Module31\_proc1());  printk("ModuleMsg: int exported from Module31 : %d\n", Module31\_int);  printk("ModuleMsg: int returned by Module31\_proc2 is : %d", Module31\_proc2());  return -1;  }  static void \_\_exit Module32\_exit(void)  {  printk("ModuleMsg: Module33 Unloaded!\n");  }  module\_init(Module32\_init);  module\_exit(Module32\_exit); |

**Module.h**

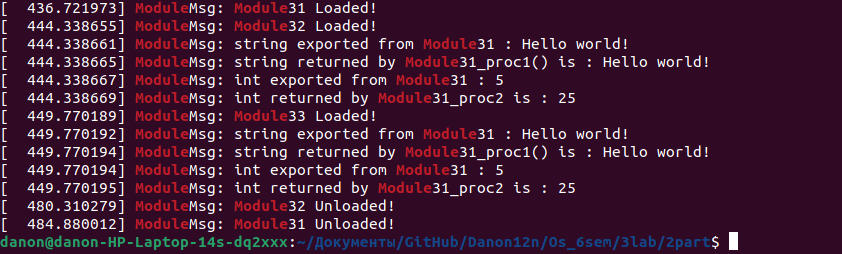
|  |
| --- |
| extern char\* Module31\_string;  extern int Module31\_int;  extern char\* Module31\_proc1( void );  extern int Module31\_proc2( void );  extern char\* Module31\_noexport( void ); |

**Результат работы модулей:**

Компиляция модулей через makefile, загрузка в ядро, отображения их в списке модулей и выгрузка

****

**Вывод модулей**

****