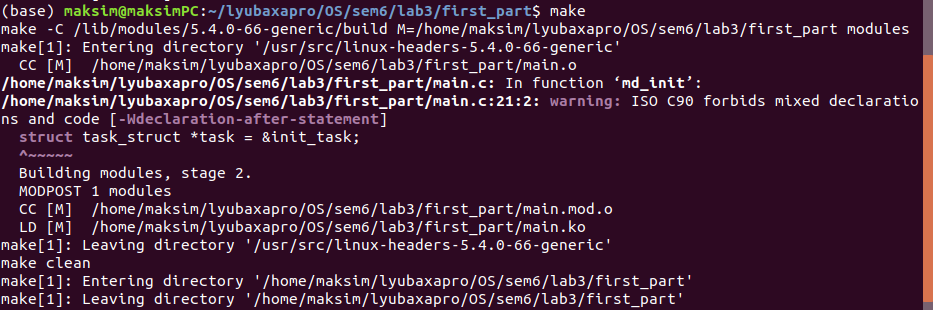
## ЧАСТЬ 1



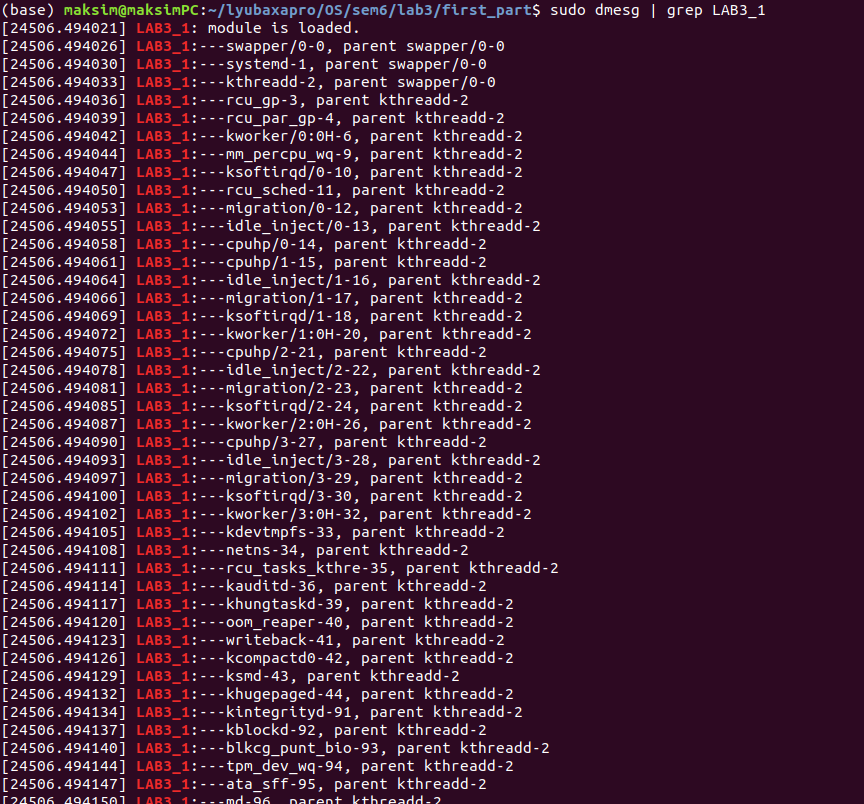
Загружает модуль в ядро из конкретного файла, если модуль зависит от других модулей, которые не загружены в ядро, то выдает ошибку и не загружает модуль. Только суперпользователь может загрузить модуль в ядро.

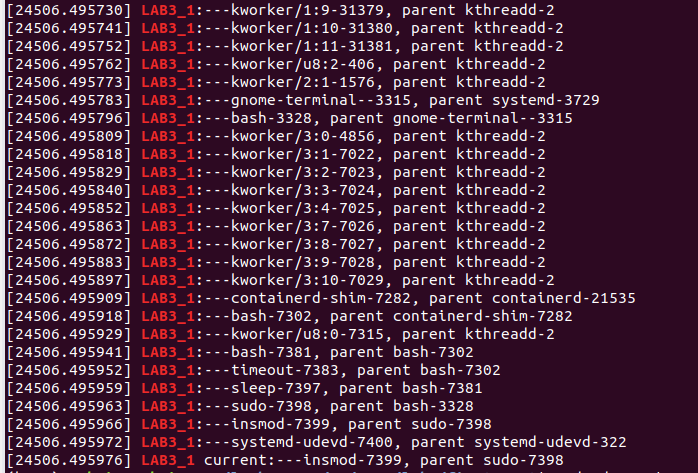


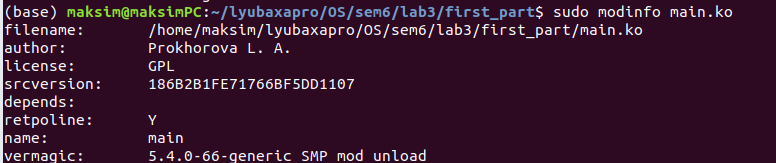
lsmod

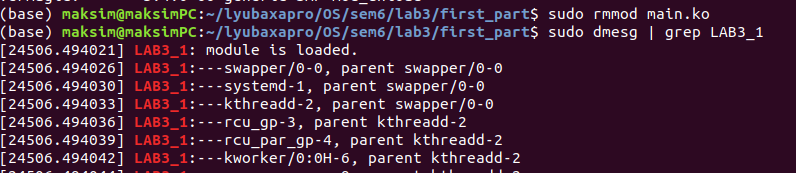
Выводит список модулей, загруженных в ядро.

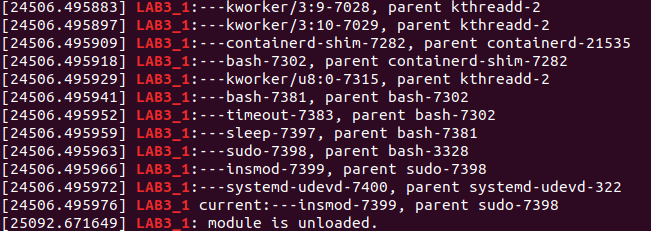










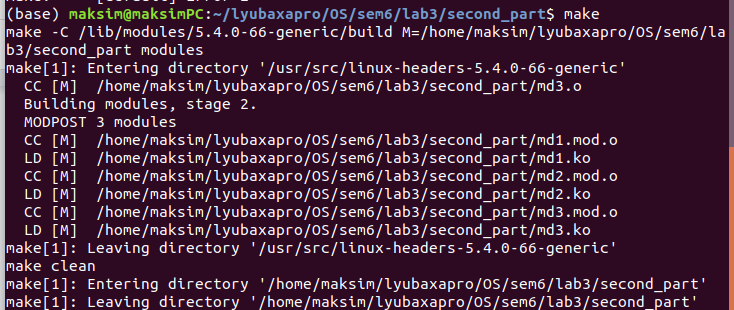


* Откуда мы берем информацию о файлах?
  + Из связанного списка дескрипторов процессов.
  + Инициируем начало списка в строке struct task\_struct \*task = &init\_task;
  + В цикле идем по списку task = next\_task(task)
  + Заканчиваем, когда доходим до первого процесса (task = next\_task(task)) != &init\_task
* Зачем используем current?
  + чтобы получить информацию о текущем процессе
* Что у вас выводит current?
  + insmod
* Почему текущий процесс insmod?
  + Так как была вызвана функция insmod. Наш модуль main загружается в контексте insmod.
* Покажите куда вы выводите информацию о процессах
  + dmesg | grep LAB3\_1

про makefile

<http://dmilvdv.narod.ru/Translate/LDD3/ldd_compiling_loading.html>

## ЧАСТЬ 2



Попробуем инсталлировать (загрузить) один из новых модулей в системе:



Возникла ошибка, потому что модуль **содержит** ссылки на неизвестные ядру имена (хотя известно, что эти недостающие имена определены в другом модуле md1). В системном журнале об этой ошибке будет выведена следующая информация:

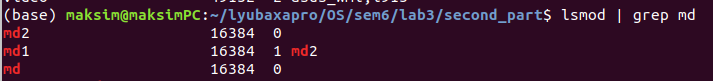


Следует выполнить загрузку модулей в другом, на этот раз правильном, порядке:

****

**модуль md1 используется модулем md2**

* **на модуль md1 ссылается некоторые другие модули или объекты ядра: цифра 1 — это число таких ссылающихся модулей, которое называется счётчиком ссылок;**
* **дальше за счётчиком ссылок указывается список тех модулей, откуда исходят такие ссылки, в данном случае, это один модуль md2;**
* **до тех пор, пока число ссылок на любой модуль в системе не станет нулевым, модуль не может быть выгружен;**
* **бывают случаи (как будет показано дальше), когда модуль вообще не может быть выгружен, в частности, когда счётчик ссылок для модуля не может быть сделан нулевым в силу каких-то причин.**

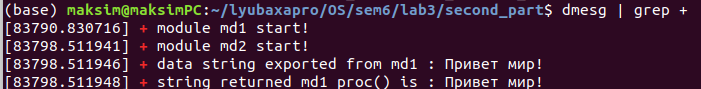


Выгрузим в правильном порядке









модуль 3 нельзя загрузить так как возвращает -1

****

при этом функция инициализации успела записать в журнал



Нельзя выгрузить md1 раньше чем md2 так как md2 использует данные из md1.

Зачем каждую выводимую строку предварять строкой "**+**"? Это **маркер**, отмечающий вывод из **собственных**модулей. В качестве него можно выбрать любой символ или вообще отказаться от него (что чаще всего и происходит). Но если настройки Linux таковы, что работают различные сервисы аудита или подобные службы, то они могут «засыпать» системный журнал достаточно плотным потоком своих сообщений, а сообщения собственных модулей будут сильно разрежены таким потоком. Так что их придётся потом разыскивать в этом потоке. Заблаговременно предварять сообщения собственных модулей фиксированными маркерами — это простейший способ позже осуществить их отбор и группировку, по крайней мере, в иллюстрационных целях, как и было показано. Отобрать собственные сообщения можно с помощью команд, подобных приведенной ниже:

|  |  |
| --- | --- |
|  | $ dmesg | tail -n60 | grep +   1. В каком порядке загружаются модули?    * Сначала загружается модуль, экспортирующий данные. Затем модуль, использующий данные. 2. Вы пробовали использовать вместо md1\_proc md1\_local и md1\_noexport? Какие при этом возникают ошибки?    * При md1\_local проект не компилируется. Возникает ошибка implicit declaration of function ‘md1\_local’.    * При md1\_noexport проект компилируется(Если он прописан в .h файле). Возникает ошибка при попытке загрузить модуль. Ошибка insmod: ERROR: could not insert module md2.ko: Unknown symbol in module 3. В каком порядке должны быть выгружены модули? Почему?    * Сначала должен быть выгружен модуль, использующий данные. Затем модуль, экспортирующий данные.    * Из Цирюлика: 4. На модуль md1 ссылается некоторые другие модули или объекты ядра: цифра 1 — это число таких ссылающихся модулей, которое называется счётчиком ссылок; Дальше за счётчиком ссылок указывается список тех модулей, откуда исходят такие ссылки, в данном случае, это один модуль md2; До тех пор, пока число ссылок на любой модуль в системе не станет нулевым, модуль не может быть выгружен; Другими словами, модуль может быть выгружен только после того, как будут выгружены все ссылающиеся на него другие модули (загруженные после него — здесь не может возникнуть цикличности или перекрёстности ссылок); Бывают случаи (как будет показано дальше), когда модуль вообще не может быть выгружен, в частности, когда счётчик ссылок для модуля не может быть сделан нулевым в силу каких-то причин. 5. Что получает второй(использующий данные) модуль по имени экспортируемых данных? Что получает, чтобы обращаться к данным?    * абсолютный адрес    * Из Цирюлика: 6. Модуль md1 экспортирует для использования другими модулями имя процедуры md1\_proc() и, что далеко не так очевидно, имя структуры данных md1\_data. Любой другой модуль (md2) может использовать в своём коде любые экспортируемые имена. Это могут быть имена, экспортируемые ранее загруженными модулями, но гораздо чаще это имена, экспортируемые ядром. Это множество экспортируемых имён ядра далее будет называться API ядра. Примером одного из вызовов из набора API ядра в показанных фрагментах кода является вызов printk(). Модуль md2, использующий экспортируемое имя, связывается с этим именем по абсолютному адресу. Как следствие этого, любые изменения (новая сборка), вносимые в ядро или экспортирующие модули, делают собранный модуль непригодным для использования. Именно поэтому бессмысленно предоставлять модуль в собранном виде — он должен собираться только на месте использования. 7. Что такое абсолютный адрес? Что такое относительный?    * абсолютный адрес - это реальный физический адрес ячейки памяти    * относительный адрес может быть представлен как сегмент:смешение, через страницы и тд 8. Какой командой можно проверить, что модули загружены в систему?    * lsmod    * lsmod | grep md    * md2 ссылается на md1 9. Покажите, что модули загружены в ситсему.    * lsmod | grep md 10. Почему третий модуль не загружен в систему?     * Так как он возвращает -1. 11. Функция инициализации этого модуля, выполнив все предписанные ей действия, преднамеренно возвращает ненулевое значение, что означает ошибку инициализации модуля; Тогда такой модуль не будет подгружен к ядру, но произойдёт это уже после выполнения кода инициализирующей функции модуля в пространстве ядра; А если такой модуль по замыслу не загружается, то он, в принципе, может не иметь функции выгрузки. |