



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и Управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №7

«Конфигурирование ядра Linux»

ДИСЦИПЛИНА: «Операционные системы»

Выполнил: студент гр. ИУК4-52Б _____ (Подпись) (Калашников А.С.) (Ф.И.О.)

Проверил: _____ (Подпись) (Красавин Е.В.) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2022

Цель работы: приобретение практических навыков по конфигурированию и компиляции ядра ОС Linux.

Задачи:

1. Получить навыки конфигурирования ядра Linux.
2. Получить навыки компиляции и запуску ядра Linux.

Задание:

Научиться конфигурировать, компилировать и запускать ядро Linux.
Выполнить следующие действия:

1. Подготовка исходного кода ядра. Необходимо заранее подготовить исходники. Скачать ядро Linux можно с официального сайта (<https://www.kernel.org>)
2. Подготовка каталогов с исходными кодами ядра. Необходимо распаковать архив с исходными кодами ядра. (Команды зависят от типа архива)
3. Конфигурирование ядра. Необходимо указать какие компоненты включить в ядро, а какие убрать (можно оставить стандартные параметры).
4. Компиляция ядра и установка модулей. Необходимо выполнить несколько шагов:
 - Подготовка
 - Непосредственно сборка ядра
 - Сборка и установка модулей
5. Перемещение ядра. Необходимо установить как ядро, так и его tar-файл в каталог, где они будут постоянно находиться.
6. Настройка и запуск lilo. Необходимо настроить загрузчик так, чтобы была возможность выбрать ядро для загрузки (старое и новое)

Ход выполнения работы:

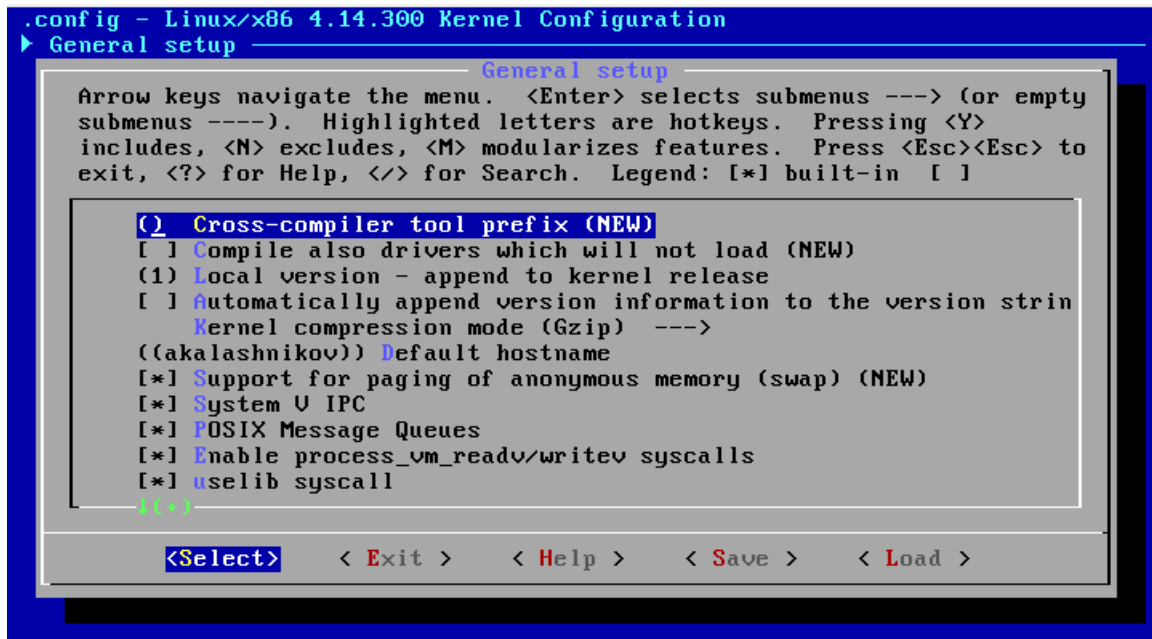


Рис. 1. Пункт основных настроек

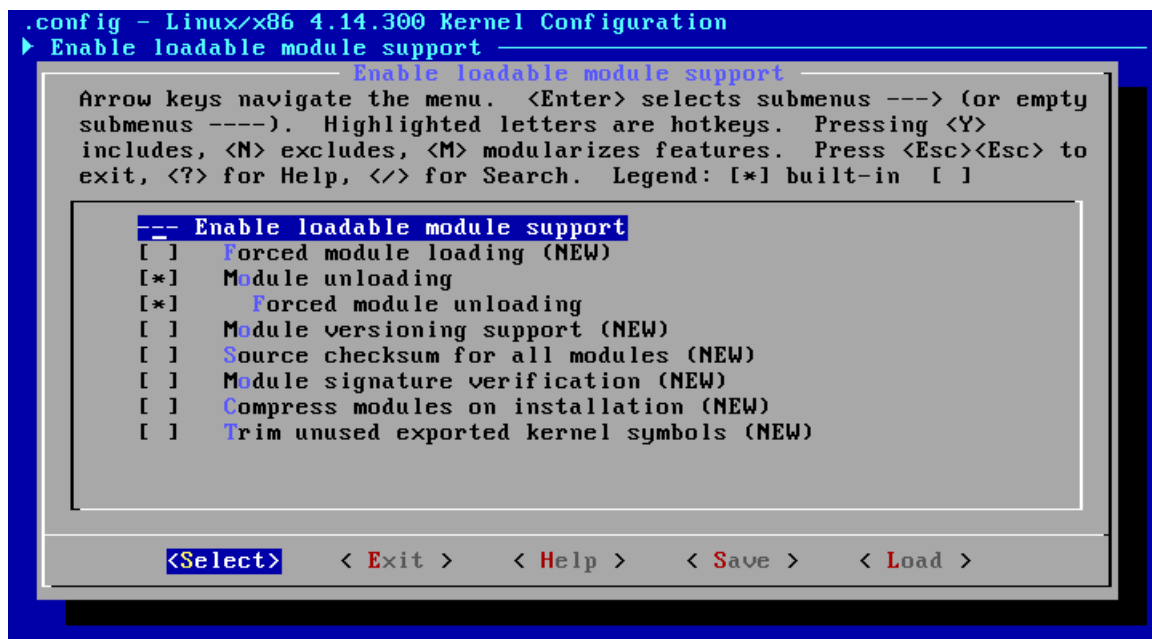


Рис. 2. Настройка загрузки внешних модулей

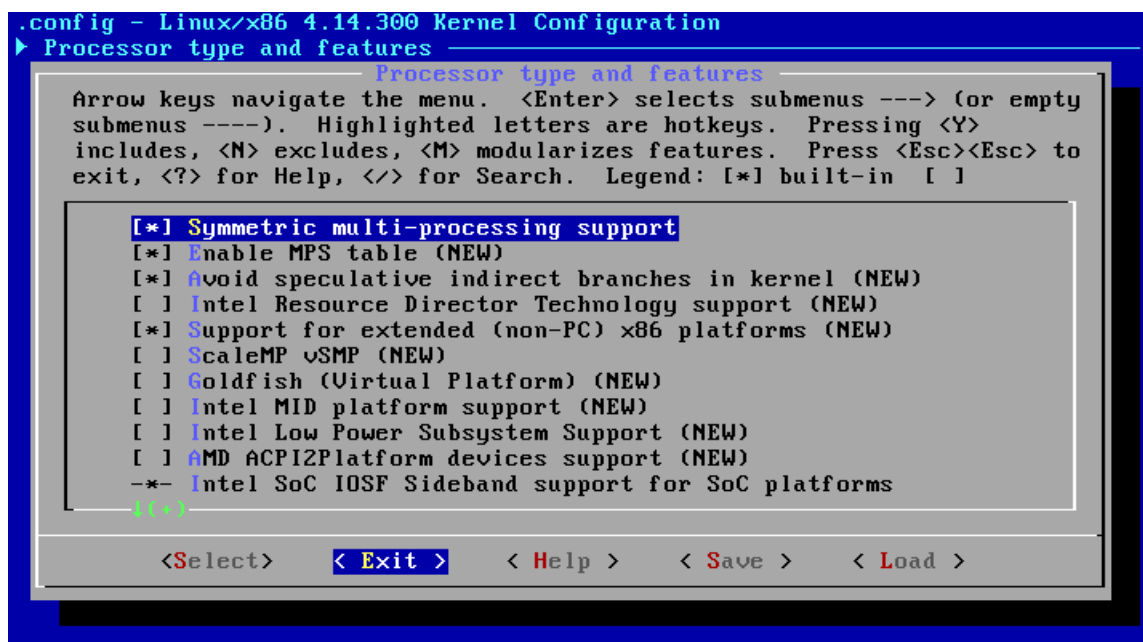


Рис. 3. Настройка типа процессора

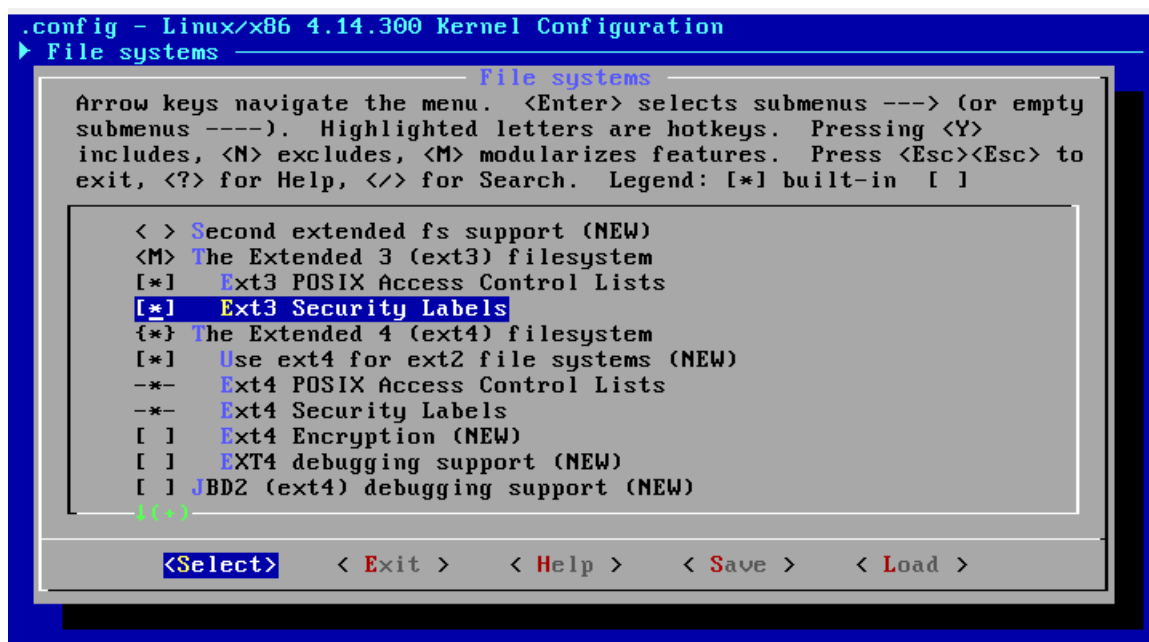


Рис. 4. Настройка файловых систем

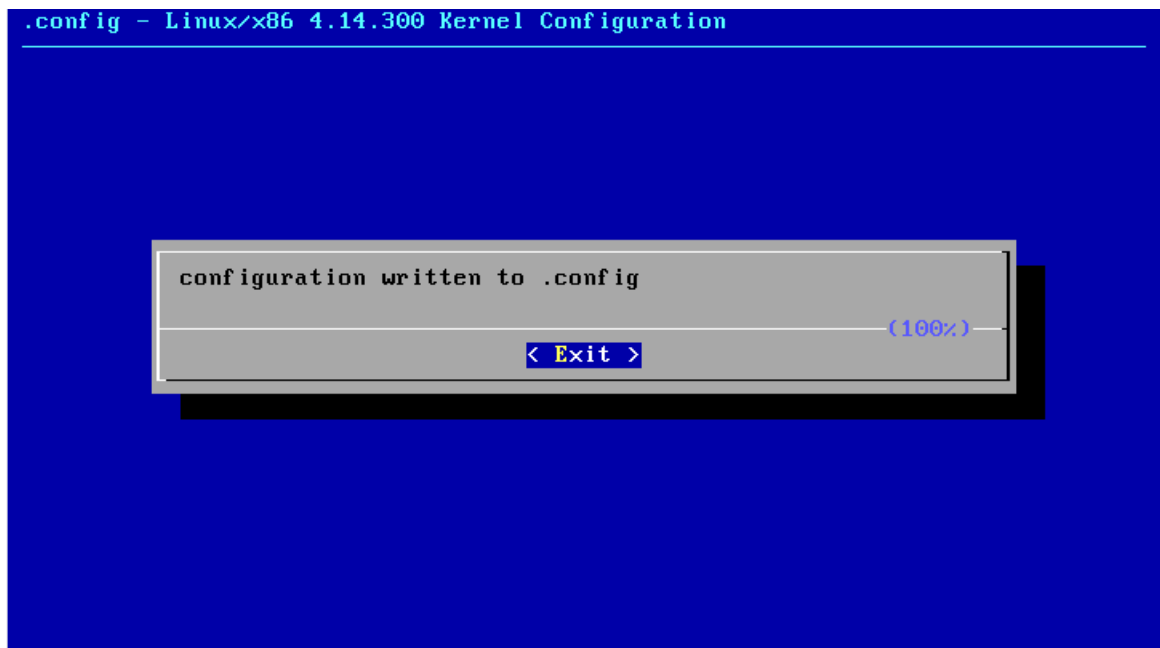


Рис. 5. Сохранение настроек конфигурации

```
root@darkstar:/usr/src/linux-4.14.300# make clean
```

Рис. 6. Подготовка к компиляции (команда make clean)

```
CC      net/ipv4/netfilter/nf_nat_ipv4.mod.o
LD [M]  net/ipv4/netfilter/nf_nat_ipv4.ko
CC      net/ipv4/netfilter/nf_nat_masquerade_ipv4.mod.o
LD [M]  net/ipv4/netfilter/nf_nat_masquerade_ipv4.ko
CC      net/ipv6/netfilter/nf_log_ipv6.mod.o
LD [M]  net/ipv6/netfilter/nf_log_ipv6.ko
CC      net/netfilter/nf_log_common.mod.o
LD [M]  net/netfilter/nf_log_common.ko
CC      net/netfilter/nf_nat.mod.o
LD [M]  net/netfilter/nf_nat.ko
CC      net/netfilter/nf_nat_ftp.mod.o
LD [M]  net/netfilter/nf_nat_ftp.ko
CC      net/netfilter/nf_nat_irc.mod.o
LD [M]  net/netfilter/nf_nat_irc.ko
CC      net/netfilter/nf_nat_sip.mod.o
LD [M]  net/netfilter/nf_nat_sip.ko
CC      net/netfilter/xt_LOG.mod.o
LD [M]  net/netfilter/xt_LOG.ko
CC      net/netfilter/xt_addrtype.mod.o
LD [M]  net/netfilter/xt_addrtype.ko
CC      net/netfilter/xt_mark.mod.o
LD [M]  net/netfilter/xt_mark.ko
CC      net/netfilter/xt_nat.mod.o
LD [M]  net/netfilter/xt_nat.ko
root@darkstar:/usr/src/linux-4.14.300#
```

Рис. 7. Конец сборки ядра

```

CC      net/netfilter/xt_mark.mod.o
LD [M]  net/netfilter/xt_mark.ko
CC      net/netfilter/xt_nat.mod.o
LD [M]  net/netfilter/xt_nat.ko
root@darkstar:/usr/src/linux-4.14.300# make modules_install
INSTALL drivers/thermal/x86_pkg_temp_thermal.ko
INSTALL fs/efivarfs/efivarfs.ko
INSTALL net/ipv4/netfilter/ipt_MASQUERADE.ko
INSTALL net/ipv4/netfilter/iptable_nat.ko
INSTALL net/ipv4/netfilter/nf_log_arp.ko
INSTALL net/ipv4/netfilter/nf_log_ipv4.ko
INSTALL net/ipv4/netfilter/nf_nat_ipv4.ko
INSTALL net/ipv4/netfilter/nf_nat_masquerade_ipv4.ko
INSTALL net/ipv6/netfilter/nf_log_ipv6.ko
INSTALL net/netfilter/nf_log_common.ko
INSTALL net/netfilter/nf_nat.ko
INSTALL net/netfilter/nf_nat_ftp.ko
INSTALL net/netfilter/nf_nat_irc.ko
INSTALL net/netfilter/nf_nat_sip.ko
INSTALL net/netfilter/xt_LOG.ko
INSTALL net/netfilter/xt_addrtype.ko
INSTALL net/netfilter/xt_mark.ko
INSTALL net/netfilter/xt_nat.ko
DEPMOD  4.14.3001
root@darkstar:/usr/src/linux-4.14.300#

```

Рис. 8. Сборка и установка модулей (make modules)

```

LD [M]  net/netfilter/xt_nat.ko
root@darkstar:/usr/src/linux-4.14.300# make modules_install
INSTALL drivers/thermal/x86_pkg_temp_thermal.ko
INSTALL fs/efivarfs/efivarfs.ko
INSTALL net/ipv4/netfilter/ipt_MASQUERADE.ko
INSTALL net/ipv4/netfilter/iptable_nat.ko
INSTALL net/ipv4/netfilter/nf_log_arp.ko
INSTALL net/ipv4/netfilter/nf_log_ipv4.ko
INSTALL net/ipv4/netfilter/nf_nat_ipv4.ko
INSTALL net/ipv4/netfilter/nf_nat_masquerade_ipv4.ko
INSTALL net/ipv6/netfilter/nf_log_ipv6.ko
INSTALL net/netfilter/nf_log_common.ko
INSTALL net/netfilter/nf_nat.ko
INSTALL net/netfilter/nf_nat_ftp.ko
INSTALL net/netfilter/nf_nat_irc.ko
INSTALL net/netfilter/nf_nat_sip.ko
INSTALL net/netfilter/xt_LOG.ko
INSTALL net/netfilter/xt_addrtype.ko
INSTALL net/netfilter/xt_mark.ko
INSTALL net/netfilter/xt_nat.ko
DEPMOD  4.14.3001
root@darkstar:/usr/src/linux-4.14.300# cp System.map /boot/System.map-4.14.300
root@darkstar:/usr/src/linux-4.14.300# cp arch/x86/boot/bzImage /boot/bzImage-4.14.300
root@darkstar:/usr/src/linux-4.14.300#

```

Рис. 9. Конец сборки и установки модулей

```
nano 2.6.0                               File: lilo.conf                             Modified
# VESA framebuffer console @ 640x480x64k
#vga=785
# VESA framebuffer console @ 640x480x32k
#vga=784
# VESA framebuffer console @ 640x480x256
#vga=769
# End LILO global section
# Linux bootable partition config begins
image = /boot/vmlinuz
  root = /dev/sda1
  label = Linux
  read-only

image = /boot/bzImage-4.14.300
  root = /dev/sda2
  label = linux-4.14.300
  read-only
  append = "debug=2 noapic nosmp"

# Linux bootable partition config ends

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^_ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Рис. 10. Настройка файла lilo.conf, позволяющая выбирать старое или новое ядро для загрузки



Рис. 10. Выбор ядра для загрузки

Вывод: в результате выполнения лабораторной работы были приобретены практические навыки по конфигурированию и компиляции ядра ОС Linux.

Ответы на контрольные вопросы

1. Раскройте понятие ядра операционной системы.

Ядро операционной системы — центральная часть ОС, обеспечивающая приложениям координированный доступ к ресурсам компьютера, таким как процессорное время, память, внешнее аппаратное обеспечение, внешнее устройство ввода и вывода информации.

2. Опишите архитектуру операционной системы Linux.

Операционную систему можно условно разделить на два уровня. На верхнем уровне находится пользовательское пространство (пространство приложений). Здесь исполняются приложения пользователя. Под пользовательским пространством располагается пространство ядра. Здесь функционирует ядро Linux. Имеется также библиотека GNU C (glibc). Она предоставляет интерфейс системных вызовов, который обеспечивает связь с ядром и дает механизм для перехода от приложения, работающего в пространстве пользователя, к ядру.

3. Перечислите основные этапы сборки ядра.

- Приобретение исходных кодов ядра.
- Подготовка каталогов с исходниками ядра.
- Конфигурирование ядра.
- Компиляция ядра и установка модулей.
- Перемещение ядра.
- Настройка и запуск lilo.

4. Назовите актуальную версию ядра Linux на данный момент.

На данный момент последняя версия ядра Linux – 6.0.7.

5. Опишите принцип нумерации версий ядра Linux.

Номер версии ядра Linux до 30 мая 2011 содержал четыре числа. Число А обозначает версию ядра. Число В обозначает старшую версию ревизии ядра. Число С обозначает младшую версию ревизии ядра. Число D впервые появилось после случая, когда в коде ядра версии 2.6.8 была обнаружена грубая, требующая незамедлительного исправления ошибка, связанная с NFS.

30 мая 2011 Линус Торвалдс выпустил ядро версии 3.0-rc1. Вместе с ним изменена политика нумерации версий ядра. Отменено использование чётных

и нечётных номеров для обозначения стабильности ядра, а третье число означает стабильность ядра.

6. Опишите по какой лицензии распространяется ОС Linux.

Linux распространяется на условиях лицензии GNU General Public License.

7. Опишите шаг приобретение исходных кодов ядра.

Исходники последней версии ядра для Linux нужно найти на <ftp://ftp.kernel.org>. Там ядра лежат в /pub/linux/. Далее загрузить ядро и поместить его в каталог /usr/src. Там же создать каталог для файлов и каталогов ядра командой `mkdir linux-4.X.X`. После этого создать связь с каталогом `linux`. Если каталог `linux-4.X.X` уже существует, то его предварительно надо удалить.

8. Опишите шаг подготовка каталогов с исходниками ядра.

Необходимо распаковать архив с исходниками. Пример необходимых команд:
`unxz linux-4.X.X.tar.xz => tar -xvf linux-4.X.X.tar`

При выполнении этих команд содержимое файлов развернется в каталог, определенный связью `linux`.

9. Опишите шаг конфигурирование ядра.

Находясь в каталоге, где распаковали архив, вводим любую из 3-х команд:

- `make config` – это наиболее простой пошаговый сценарий;
- `make menuconfig` – это более удобный метод (использовать если `xconfig` не запускается);
- `make xconfig` – это графическая утилита для настройки ядра. Перед тем, как ей воспользоваться необходимо перейти в среду X Window.

На этом этапе необходимо выбрать опции, которые будут использоваться в новом ядре. Далее нужно пройти по разделам и включить драйвера для своего оборудования. После выполнения всех действий ядро готово к сборке.

10. Опишите шаг компиляция ядра и установка модулей.

В свою очередь этот этап делится на шаги:

- Подготовка (`make clean`);
- Непосредственно сборка ядра (`make bzImage`);
- Сборка и установка модулей (`make modules, make modules_install`).

Первый из них - `make clean` - является подготовкой. Команда `make clean` удалит все лишние (вспомогательные) файлы, созданные от предыдущих процессов компиляции. Далее идет шаг, при котором необходимо непосредственно собрать ядро. Третьим шагом является сборка и установка модулей ядра. Этот процесс выполняется с помощью 2-х команд `make modules` и `make modules_install`. При выполнении команды `make modules` происходит сборка модулей, которые соответствуют ядру, созданному на предыдущем этапе. Команда `make modules_install`, в свою очередь, перемещает созданные модули из исходного дерева ядра в каталог `/lib/modules/<kernel-version>/kernel/<module-type>`.

11.Опишите шаг перемещение ядра.

После завершения процесса сборки необходимо установить как ядро, так и его `map`-файл в каталог, где они будут постоянно находиться (`/boot`). Надо переместить файл `System.map` в каталог `/boot`, добавив номер версии ядра в конец строки: `cp System.map /boot/System.map-2.X.X`.

Затем точно так же нужно переместить и само ядро: `cp arch/i386/boot/bzImage /boot/bzImage-2.X.X`.

12.Опишите шаг настройка и запуск lilo.

Вставив новый раздел образа в файл `/etc/lilo.conf` можно получить дополнительный образ загрузки. Для настройки дополнительного образа загрузки необходимо продублировать последние 4 строки файла и изменить адрес до загружаемого ядра. После конфигурации файла `/etc/lilo.conf` следует переустановить диспетчер начальной загрузки системы (`lilo`).

13.Приведите пример команды для конфигурирования ядра.

```
make menuconfig
```

14.Приведите пример команд для компиляции ядра и установки модулей.

```
make bzImage
```

```
make modules, make modules_install
```

15.Приведите пример модифицированного файла lilo.conf с возможностью выбора ядра для запуска.

```
image = /boot/<Уже стоящее в системе ядро>  
label = linux.orig
```

```
root = /dev/hda1
read-only
image = /boot/<Новое ядро>
lable = linux
root = /dev/hda1
read-only
append = "debug=2 noapic nosmp"
```