Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)

Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»

В.Ю. Мельников

Исследование процесса загрузки ОС Linux

Электронное учебное издание

Методические указания по выполнению лабораторных работ
по дисциплине "Операционные системы"

2019

Введение

Цель работы - исследование процесса загрузки Linux на примере дистрибутива Debian. Освоение работы с интерпретатором командной строки «bash». А так же освоение подключения репозиториев и установки пакетов.

Продолжительность работы - 2 часа.

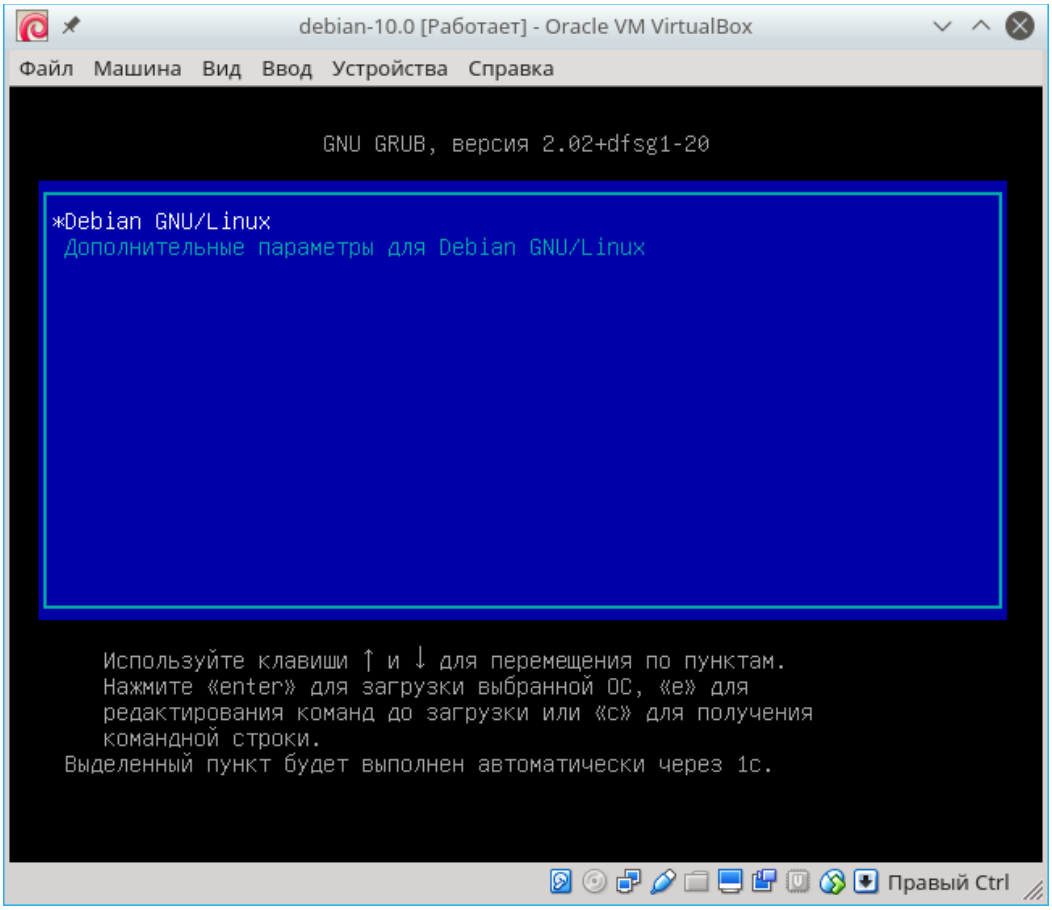
Этапы загрузки компьютера

BIOS

Сначала из ПЗУ копируется в оперативную память BIOS — базовая система ввода вывода. BIOS проверяет исправность процессора, памяти и видеоадаптера. Не экран выводится информация об этом процессе.

Первичный загрузчик

Затем, с заданного в BIOS физического жёсткого диска из 0-сектора считывается Master Boot Record (MBR). В MBR хранится таблица разделов, на которые разбит жесткий диск и первичный загрузчик. Помните, при установке Debian установил первичный загрузчик GRUB в MBR.



Если бы не был установлен GRUB, а более простой первичный загрузчик, он нашёл бы в таблице разделов раздел, помеченный как активный, считал из него 0-сектор и передал управление на вторичный загрузчик.

GRUB устроен сложнее. Первичный загрузчик GRUB считывает несколько секторов, лежащих сразу за MBR — полуторный заргузчик.

Полуторный загрузчик GRUB уже умеет работать с файловой системой. Он читает вторичный загрузчик из определённого файла и передаёт управление ему.

GRUB при установке или настройке подготовил меню выбора операционных систем, установленных в разные разделы. При загрузке он выводит это меню и предлагает пользователю выбрать операционную систему, которую надо загрузить.

По умолчанию будет выбрана заданная в настройках ОС и через несколько секунд она будет автоматически загружена. Для ускорения процесса можно нажать «Enter»

Вторичный загрузчик

Загружает из каталога «/boot» файловой системы образ образа ядра и образ «initrd» (о нем позже). Затем передаёт управление ядру операционной системы.

После изменения параметров в конфигурационных файлах ядра, необходимо дать команду, которая обновит образ ядра.

Ядро Linux

Содержит основные функции, необходимые большинству программ. В том числе функции запуска программ, обеспечение многозадачности.

При загрузке ядро монтирует «initrd» (Initial RAM Disk). Это диск в оперативной памяти, на котором расположена временная файловая система, используемая ядром Linux при начальной загрузке. На диске «initrd» расположен минимальный набор утилит и драйверы используемых файловых систем.

Далее ядро linux монтирует разделы реального диска, демонтирует «initrd» и запускает init

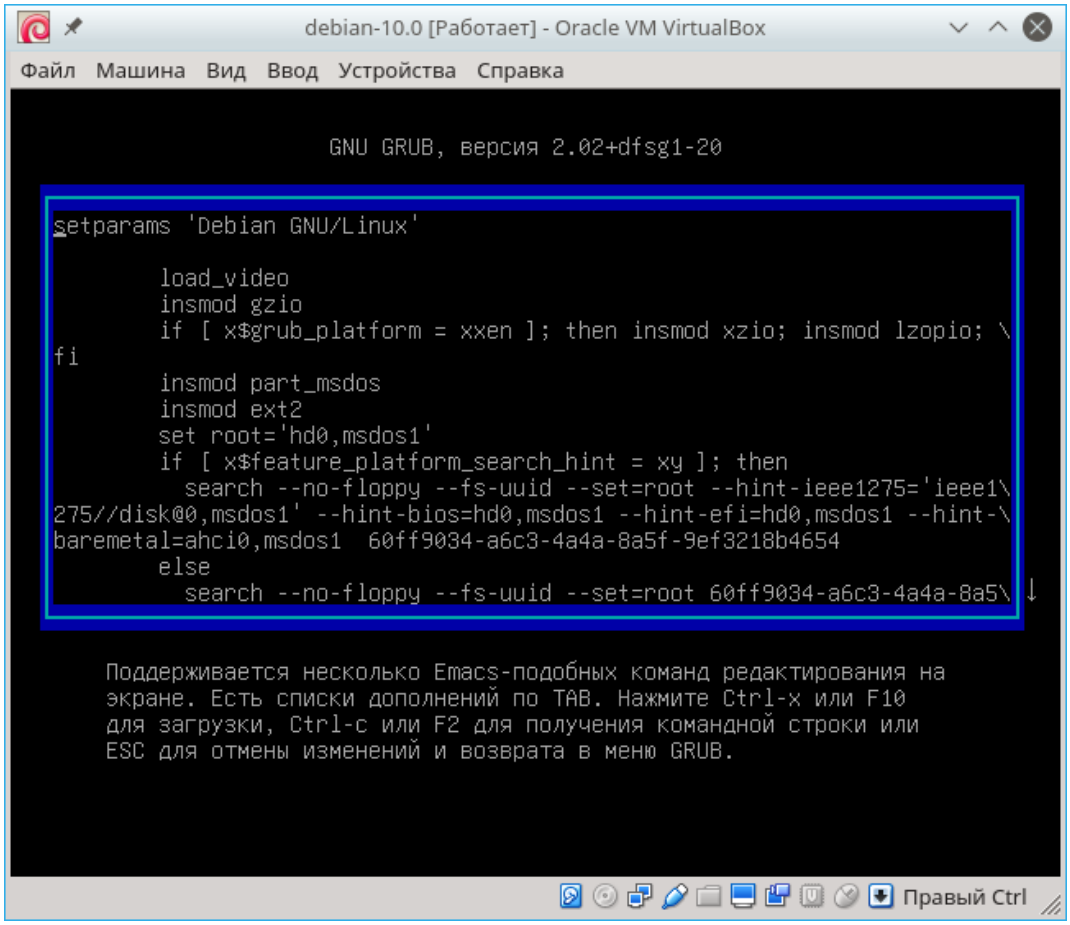
Init

Запускает в несколько этапов службы и переходит в режим ожидания. Он снова активируется при завершении работы компьютера и останавливает службы.

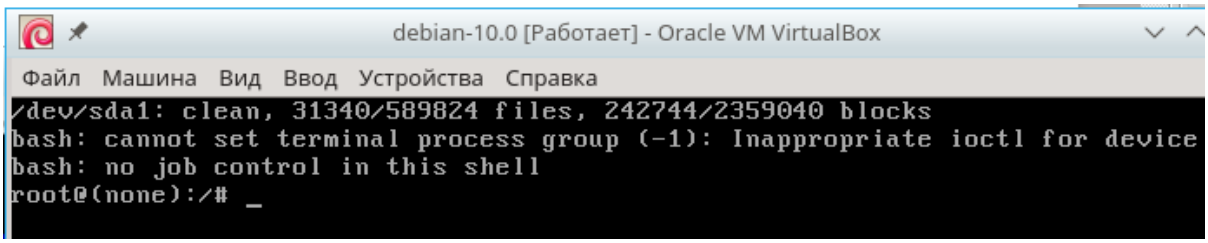
Когда потеряли пароль суперпользователя

В Debian, если есть физический доступ к компьютеру, имеется возможность сменить пароль суперпользователя, если старый пароль утерян.

- Перезагрузите компьютер
- Когда появится меню GRUB нажмите клавишу «e»



- В конце строки, начинающийся с "linux /boot/vmlinuz-...", поставить пробел и дописать «init=/bin/bash» - вместо запуска служб linux запустить интерпретатор командной строки (bash).
- Нажать ctrl+x



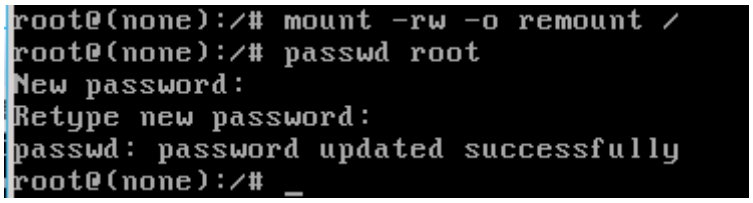
Ну вот, мы вошли в linux без пароля. Мы можем давать команды!

- Но надо сначала перемонтировать файловую систему с правами на запись.

```
mount -rw -o remount /
```

Теперь можно сменить пароль.

- Дайте команду «`passwd`» и введите новый пароль (2 раза)



- Осталось перезагрузиться Для этого дайте из меню «Машина» виртуальной машины команду «Перезапустить».

Запуск программ, командная оболочка «bash»

Когда вы ввели правильный пароль, «login» запустил интерпретатор команд (или командную оболочку) «**bash**». Именно этот процесс обеспечивает ввод команды и её выполнение. Рассмотрим подробнее его возможности.

Нажмите клавишу «↑» на клавиатуре. Высвечивается предыдущая набранная команда. Повторные нажатия «↑» высвечивают предыдущие команды. Всегда можно повторить введённую ранее команду, при необходимости отредактировав её. Историю команд можно получить командой «history»

История набранных вами команд хранится в файле истории в открытом виде поэтому, даже если команда позволяет ввести пароль в командной строке, не стоит пользоваться этой возможностью, а использовать форму, которая запрашивает пароль в процессе выполнения команды.

Команду не обязательно набирать полностью. Наберите «hi» и нажмите клавишу «Tab». Интерпретатор команд допишет за вас команду «history». А вот «h», для авто дополнения недостаточно, поскольку несколько команд начинается на эту букву. Нажмите клавишу «Tab» дважды. Высветится список команд, начинающихся на введённую строку:

Даже, если вам не трудно ввести команду целиком, лучше пользоваться авто дополнением, во избежание ошибок ввода. Поможет оно и если вы забыли, как точно пишется команда. Имена команд представляют собой либо английские слова, либо аббревиатуры, так что первые пару букв можно угадать. Особенно авто дополнение полезно при вводе в команде имени файла.

А ещё, интерпретатор команд «bash» выполняет командные сценарии. Когда администратору надоедает выполнять ежедневно одни и те же команды, он создаёт файл, пишет в него сценарий, и впредь выполняет эти действия одной командой. Сценарий для выполнения несложной задачи написать проще, чем программу на языке «C» и не надо

устанавливать пакеты компилятора и сборщика. Язык сценариев имеет все конструкции языка программирования (циклы, условия, переменные):

```
#!/usr/bin/env bash
s1='строка 1'
s2='строка 2'
if [[$s1 == "$s2"]]
then
  echo 'условие выполняется'
else
  echo 'условие не выполняется'
fi

i=0
until [[$i -eq 10]]
do
  echo "$i"
  i=$(($i+1))
done
```

Но отсутствуют возможности обращения к системным функциям, да и синтаксис неудобный. Так, что если не удалось написать сценарий за 15 минут, бросайте и пишите программу на Вашем любимом языке программирования.

Но вернёмся к выполнению команды. Мы ввели команду и нажали клавишу «ENTER». Интерпретатор команд разбирает введённый текст. Первое слово (до пробела) это имя программы. Может быть задан путь к программе (например, «`/usr/bin/ps`») или только имя программы, (например, «`ps`»).

Программа это файл, помеченный признаком «выполняемый». Если команда начинается с «/», то задан полный путь к программе, начиная с корневого каталога. Интерпретатор команд запускает соответствующую программу.

Если путь к программе не указан, интерпретатор команд ищет программу сначала в текущем каталоге, а затем, в каталогах, перечисленных в переменной окружения «`PATH`». Пути разделяются символом «`:`».

```
root@debian:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Это был суперпользователь. А вот, пути для поиска программ пользователя «user»

```
user@debian:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

Обратите внимание, на отсутствие путей «/sbin», «/usr/sbin» и«/usr/local/sbin». В этих каталогах лежат инструменты системного администратора, часть из них обычный пользователь не имеет прав запускать, но многие команды и обычный пользователь может использовать для получения информации, надо только указать полный путь к команде.

При необходимости, в эту переменную можно добавить нужные Вам пути. Например, команда «`export PATH=$PATH:/home/user/my_programs`» добавляет путь «/home/user/my_programs» к имеющимся на время работы данного интерпретатора команд. Если надо добавить этот путь постоянно, добавьте эту команду в конец файла «`.bashrc`». Мы ещё заглянем в этот файл в работе «Исследование методов защиты»

После имени программы, через пробел следуют параметры, которые уточняют, что должна сделать команда. Например, «`cat /proc/meminfo`» выводит на экран содержимое файла «`/proc/meminfo`».

Необязательные параметры называют опциями. В команде «`free -h`» опция «-h» говорит команде, что надо выводить объём памяти в килобайтах и мегабайтах. Без этого параметра команда «free» выводит объём памяти в байтах.

Параметров может быть несколько, и разделяются они пробелами. Если параметр (например, имя файла) содержит пробел, параметр следует заключить в апострофы или кавычки. Например, команда «`mkdir 'Операционные системы'`» создаст каталог «Операционные системы». Если апострофы не поставить, интерпретатор команд передаст программе 2 параметра и «mkdir» создаст 2 каталога «Операционные» и «системы».

Если имя файла содержит служебные символы, «& | > < ; $ ( )» следует использовать апострофы, в этом случае служебные символы рассматриваются, как обычные. И наоборот, если вам надо подставить в команду значение переменной окружения, следует использовать кавычки. Сравните:

```
root@debian:~# echo "*** $USER ***"
*** root ***
root@debian:~# echo '*** $USER ***'
*** $USER ***
```

Параметры у каждой программы свои. Можно получить краткую справку по большинству программ, указав параметр «--help». Например, команда «`free --help`» выдаёт справку по команде «free». Часто даже краткая справка не умещается на экране. Можно просмотреть её постранично, добавив «`|more`». Например, «`cat –help|more`». Для перелистывания страниц используется клавиша «пробел».

Более подробную информацию по параметрам команд можно найти в Интернете или используя команду «`man`» (от слова «manual» - руководство). Например, «`man cat`»

Установка новых программ в Debian

Репозитории

Программы для linux хранятся в сетевых репозиториях. У каждого релиза и каждой

версии Linux имеется свой официальный репозиторий в которых лежат самые свежие версии программ, скомпилированные под конкретную версию. Эти репозитории используются для установки обновлений и программ которые не вошли на установочный диск.

Кроме того имеется множество репозиториев от сторонних разработчиков. Они содержат множество полезных программ. Желательно пользоваться репозиториями, рекомендованными разработчиками вашего релиза Linux и остерегаться подмены репозиториев злоумышленниками.

Настроим наш Debian на работу с официальным репозиторием.

Для Linux характерно хранение настроек в текстовых файлах. В Debian перечень репозиториев хранится в файле /etc/apt/sources.list

Редактор vi

Заодно, освоим доисторический редактор «vi». Этот редактор имеет жутко непривычный интерфейс (хотя есть фанаты этого интерфейса). Хотелось бы, чтобы вы умели пользоваться этим редактором, потому что он включён абсолютно во все релизы Linux и в условиях недоступного интернета на сервере, рано или поздно вам придётся им воспользоваться. Кроме того, это единственный редактор, который не загружает весь файл в память и позволяет отредактировать очень большой файл.

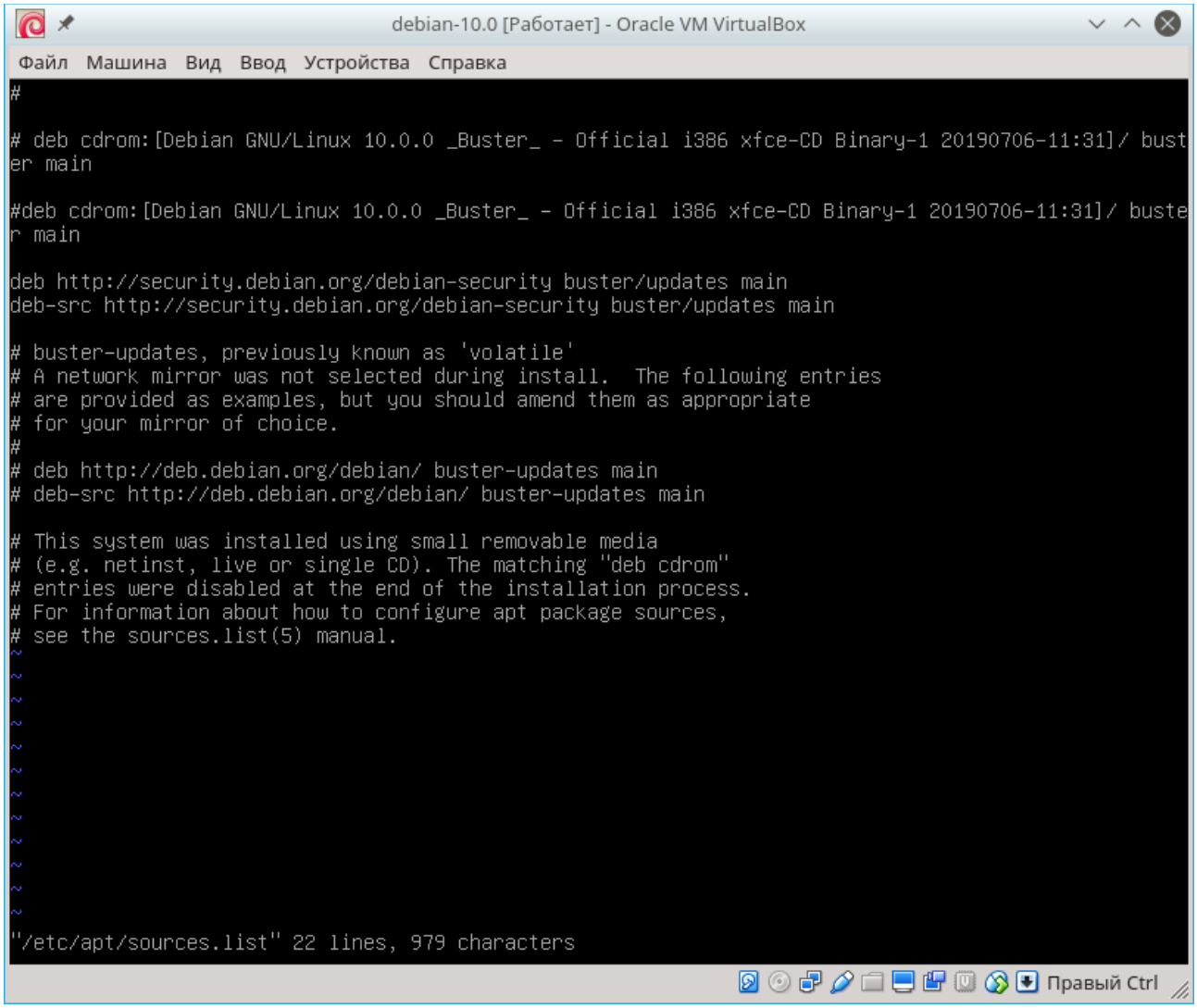
Чтобы отредактировать файл «/etc/apt/sources.list» в редакторе vi надо дать команду:

```
vi /etc/apt/sources.list
```

Когда будете набирать имя файла, обязательно воспользуйтесь автодополнением. Наберите «vi /etc/apt/s» и нажмите «Tab». Если на указанную букву начинается только один файл, его имя автоматически допишется в командную строку. Если вы ошиблись при наборе пути, имя не дополнится, а если правильно, то меньше набирать символов.

Для выполнения команды нажмите клавишу «Enter».

Запускается редактор vi, который выводит текст заданного файла.



Но не кидайтесь ого редактировать. Про vi говорят, что он может только портить текст и бибикать. Прежде всего, научимся выходить без сохранения.

Когда разрабатывали «vi» на клавиатуре не было функциональных клавиш. Поэтому команды отдаются обычными символами. Редактор «vi» может находится в 3 режимах:

- Командный
- Ввода команд
- Ввода текста

Нажмите клавишу «:». В левом нижнем углу экрана должен был появиться символ «:». Это значит вы перешли в режим ввода команд. Если «:» не появилось, то вы находились в режиме ввода текста. Значит надо нажать клавишу «ESC» и снова «:».

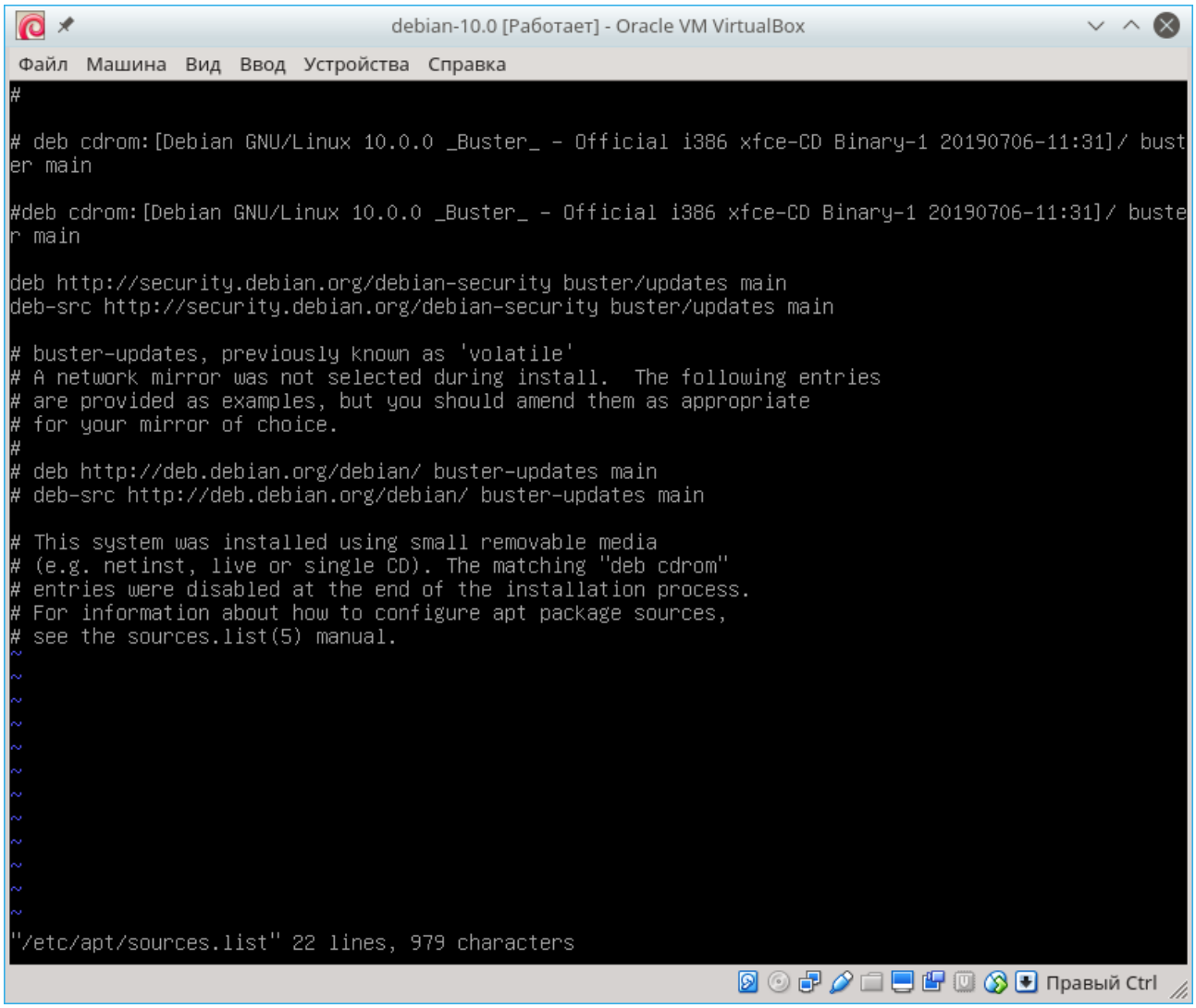
Продолжим ввод команды. Надо набрать «:q!» и нажать «Enter».

Снова появилась подсказка интерпретатора команд

Снова запускаем редактор. Не спешите набирать заново команду.

Воспользуемся историей команд. Нажмите клавишу «↑». Появится предыдущая команда. Осталось нажать «Enter»

Посмотрим содержимое файла «/etc/apt/sources.list».



Строки, которые начинаются с символа «#» это комментарии.

Обратите внимание на строки, начинающиеся с «`deb cdrom:`» эти строки задают использование установочного CD в качестве репозитория. В данной версии ини закомментированы. Но в более ранних версиях Debian их надо закромментировать. Установочный CD извлечён из привода и любая попытка установить программу будет выдавать предупреждение «репозиторий недоступен». В Debian 10 для тренировки допишем второй символ «#» в начало строки.

Клавишами «↑» «↓» «←» «→» подведите курсор в начало строки «`deb cdrom:`» и нажмите клавишу «i». Мы перешли в режим вставки текста. Теперь можно вводить текст. В нашем случае «#». Если начнёте вводить текст, не перейдя в режим вставки — испортите текст. Большинство символов соответствуют командам (включая, удалить словл, удалить строку...). Когда закончите вводить текст, нажмите клавишу «Esc» (или клавишу

перемещения курсора).

Сохраняем текст. Для этого наберите «:w» и нажмите «Enter»

Файл записан.

Теперь надо добавить официальный репозиторий Debian

Для этого надо добавить строку

Для Debian 10:

```
deb http://deb.debian.org/debian/ buster main
```

Для Debian 9:

```
deb http://ftp.debian.org/debian/ stretch main
```

Для Debian 8:

```
deb http://ftp.debian.org/debian/ jessie main
```

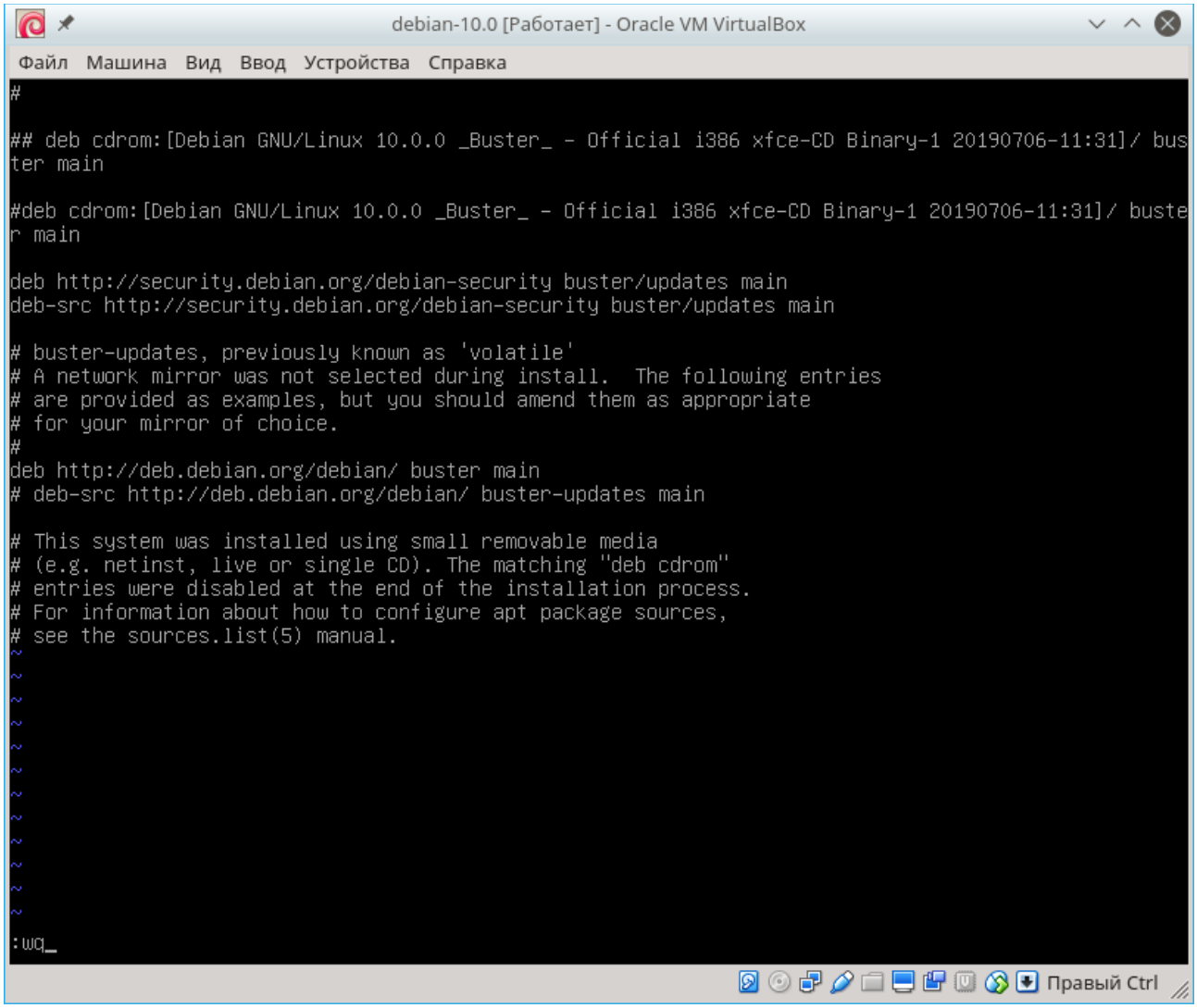
Каждая версия имеет собственное имя

Не спешите вводить новую строку в тексте есть закомментированная похожая строка:

```
# deb http://deb.debian.org/debian/ buster-updates main
```

Подведите курсор и удалите (клавишей «Delete») «#» и «-updates»

Сохраняем и выходим «:wq»



В репозиториях устанавливаемые программы хранятся в составе пакетов.

Надо загрузить (обновить) списки пакетов в подключённых репозиториях командой `apt-get update`

```
root@debian10:~# apt-get update
Пол:1 http://deb.debian.org/debian buster InRelease [118 kB]
Пол:2 http://security.debian.org/debian-security buster/updates InRelease [39,1 kB]
Пол:3 http://deb.debian.org/debian buster/main i386 Packages [7 854 kB]
Пол:4 http://deb.debian.org/debian buster/main Translation-ru [516 kB]
Пол:5 http://deb.debian.org/debian buster/main Translation-en [5 967 kB]
Получено 14,5 MB за 27с (535 kB/s)
Чтение списков пакетов… Готово
```

Если этого не сделать, Debian скажет, что пакет отсутствует и не установит его.

Просмотрите вывод команды «apt-get update» и убедитесь что отсутствуют сообщения об ошибках. Возможны 2 варианта — не доступна сеть или вы ошиблись при редактировании файла. В случае ошибки дайте команду «ping -c 1 deb.debian.org» и убедитесь, что сервер доступен по сети. Часто просто сетевой кабель не подключён.

Установка пакетов

Установим несколько пакетов, содержащие команды, не входящие ныне в стандартные

утилиты, но которые нам понадобятся.

```
apt-get install psmisc net-tools w3m
```

Кроме запрошенных пакетов будут установлены ещё несколько пакетов, содержащих нужные им библиотеки. Команда спрашивает разрешение на установку.

Удобнее вводить не «Д» а «у».

```
Обновлено 0 пакетов, установлено 5 новых пакетов, для удаления отмечено 0 пакетов, и 0
новлено.
Необходимо скачать 1 697 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 4 492 kB.
Хотите продолжить? [Д/н] y
Пол:1 http://deb.debian.org/debian buster/main i386 libgc1c2 i386 1:7.6.4-0.4 [227 kB]
Пол:2 http://deb.debian.org/debian buster/main i386 libgpm2 i386 1.20.7-5 [35,8 kB]
Пол:3 http://deb.debian.org/debian buster/main i386 net-tools i386 1.60+git20180626.aeb
B]
Пол:4 http://deb.debian.org/debian buster/main i386 psmisc i386 23.2-1 [127 kB]
Пол:5 http://deb.debian.org/debian buster/main i386 w3m i386 0.5.3-37 [1 056 kB]
Получено 1 697 kB за 10c (169 kB/s)
Выбор ранее не выбранного пакета libgc1c2:i386.
(Чтение базы данных … на данный момент установлено 27304 файла и каталога.)
Подготовка к распаковке …/libgc1c2_1%3a7.6.4-0.4_i386.deb …
Распаковывается libgc1c2:i386 (1:7.6.4-0.4) …
Выбор ранее не выбранного пакета libgpm2:i386.
Подготовка к распаковке …/libgpm2_1.20.7-5_i386.deb …
Распаковывается libgpm2:i386 (1.20.7-5) …
Выбор ранее не выбранного пакета net-tools.
Подготовка к распаковке …/net-tools_1.60+git20180626.aebd88e-1_i386.deb …
Распаковывается net-tools (1.60+git20180626.aebd88e-1) …
Выбор ранее не выбранного пакета psmisc.
Подготовка к распаковке …/psmisc_23.2-1_i386.deb …
Распаковывается psmisc (23.2-1) …
Выбор ранее не выбранного пакета w3m.
Подготовка к распаковке …/archives/w3m_0.5.3-37_i386.deb …
Распаковывается w3m (0.5.3-37) …
Настраивается пакет net-tools (1.60+git20180626.aebd88e-1) …
Настраивается пакет libgpm2:i386 (1.20.7-5) …
Настраивается пакет psmisc (23.2-1) …
Настраивается пакет libgc1c2:i386 (1:7.6.4-0.4) …
Настраивается пакет w3m (0.5.3-37) …
Обрабатываются триггеры для libc-bin (2.28-10) …
Обрабатываются триггеры для man-db (2.8.5-2) …
Обрабатываются триггеры для mime-support (3.62) …
```

Поиск пакета в котором есть программа

Если в очередной версии Linux пропала привычная вам программа, можно найти пакет, в котором она лежит командой

```
apt-cache search КОМАНДА
```

Если в при выполнении этих лабораторных работ выдаётся сообщение «команда не найдена» (и команду вы указали правильно) найдите пакет и установите его.

Текстовый браузер w3m.

Текстовый браузер w3m появился в 1995 году. Несмотря на то, что текстовые браузеры, как правило, предназначены только для просмотра текста в однооконном режиме, w3m предлагает достаточно широкие возможности:

вкладки, загрузку изображений, работу с мышкой, контекстное меню.

Для запуска w3m из консоли необходимо просто ввести в консоли: w3m

Для запуска w3m сразу с конкретного адреса необходимо дописать этот адрес в команде, например:

w3m bmstu.ru

На рисунке 8 показан пример отображения страницы в браузере w3m.

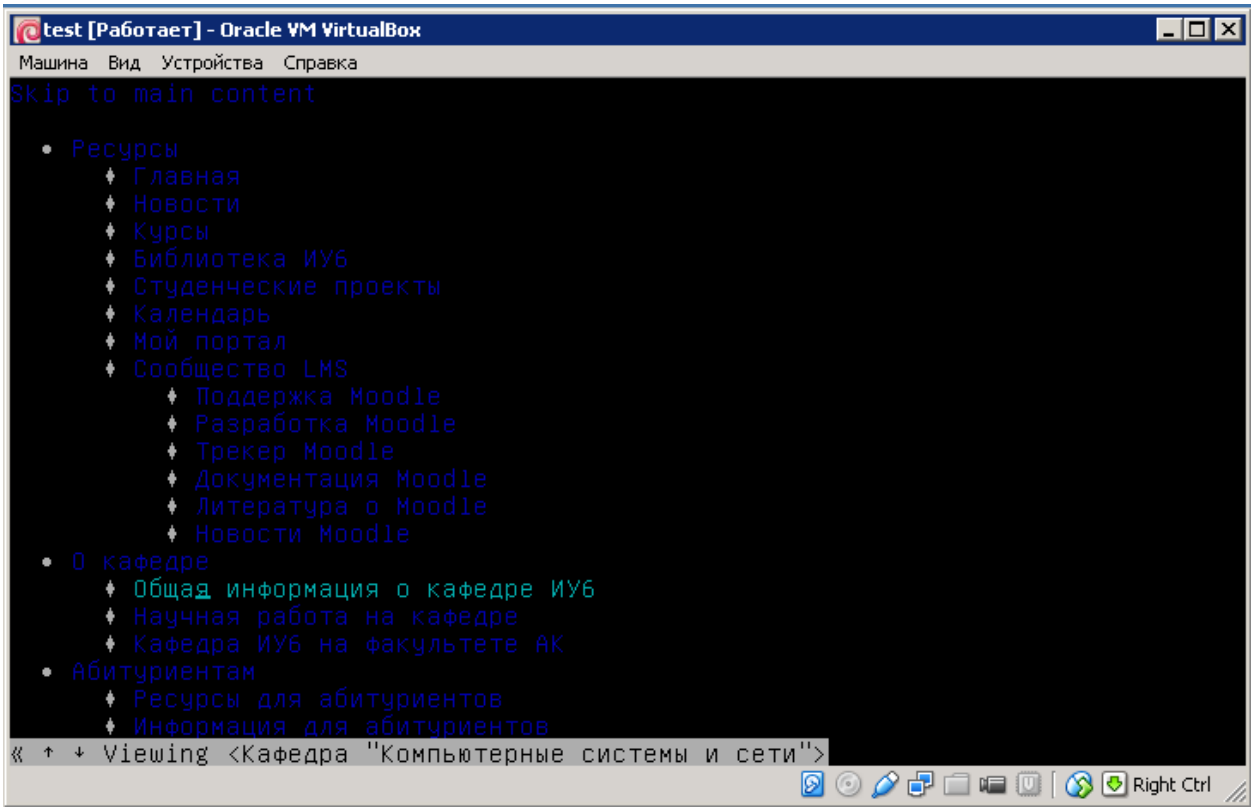


Рисунок 8 – Сайт кафедры ИУ6 в текстовом браузере w3m.

Основные команды w3m:

Shift-U – открывает строку ввода URL

Shift-B – переход на предыдущую страницу

Shift-T – открыть новую вкладку

Shift-[– предыдущая вкладка

Shift-] – следующая вкладка

Shift-H – страница помощи

q – выход

Поле для ввода текста выбирается путём наведения на него курсора и нажатия клавиши ввода, после чего внизу экрана появляется приглашение на

ввод текста.