Лабораторная работа №1

Компьютерные науки и технология программирования. Операционные системы

Татур Стефан Андреевич

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Установить виртуальную машину,произвести ее базовую настройку и получить навыки работы с ней.

2 Выполнение лабораторной работы

1)Прежде всего я начал работу со встроенной коснолью. Я проверил папку cd/var/tmp и удостоверившись в том,что она существует,произвел запуск виртуальной машины командой VirtualBox &

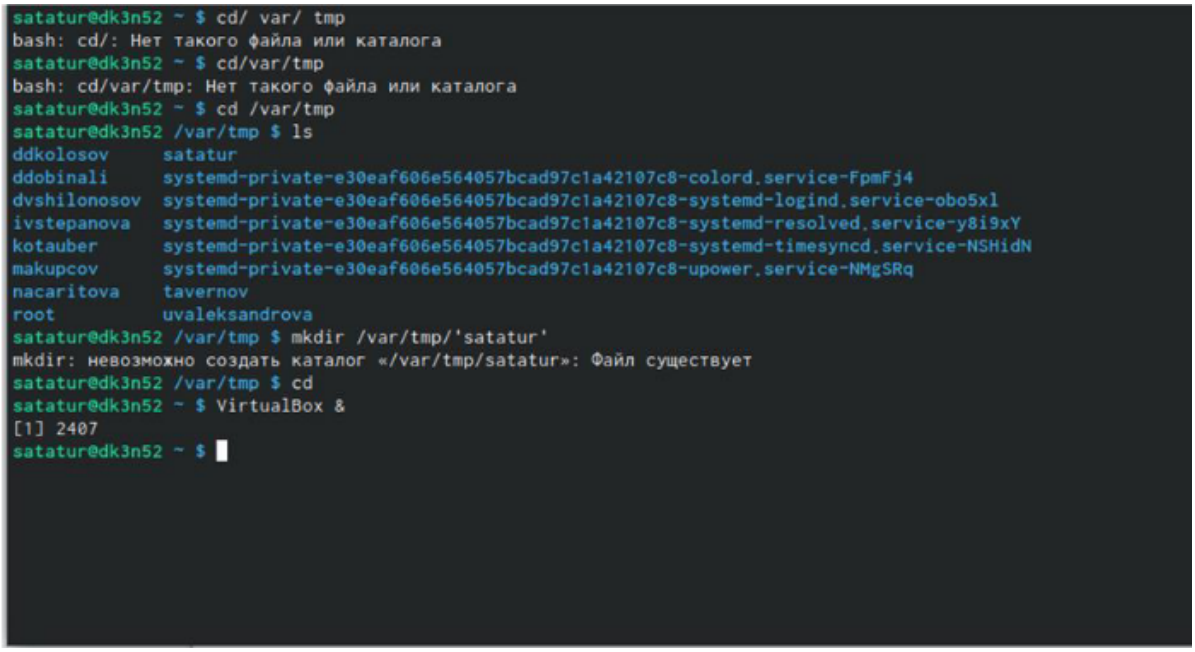


Рис. 2.1: virtual box

2)2. После запуска установщика виртуальной машины я проделал следующие шаги: 2.1 Создал новую виртуальную машинус именем “Linux 2.2 Указал объем памяти равный 4096 мб. 2.3 Создал новый жесткий виртуальный диск,указав тип VDI и выбрал динамический виртуальный диск. 2.4 Выделил размер файла 50 гб и указал имя виртуального жесткого диска. 2.5 Увеличил объем вмидеопамяти

до 128 мб

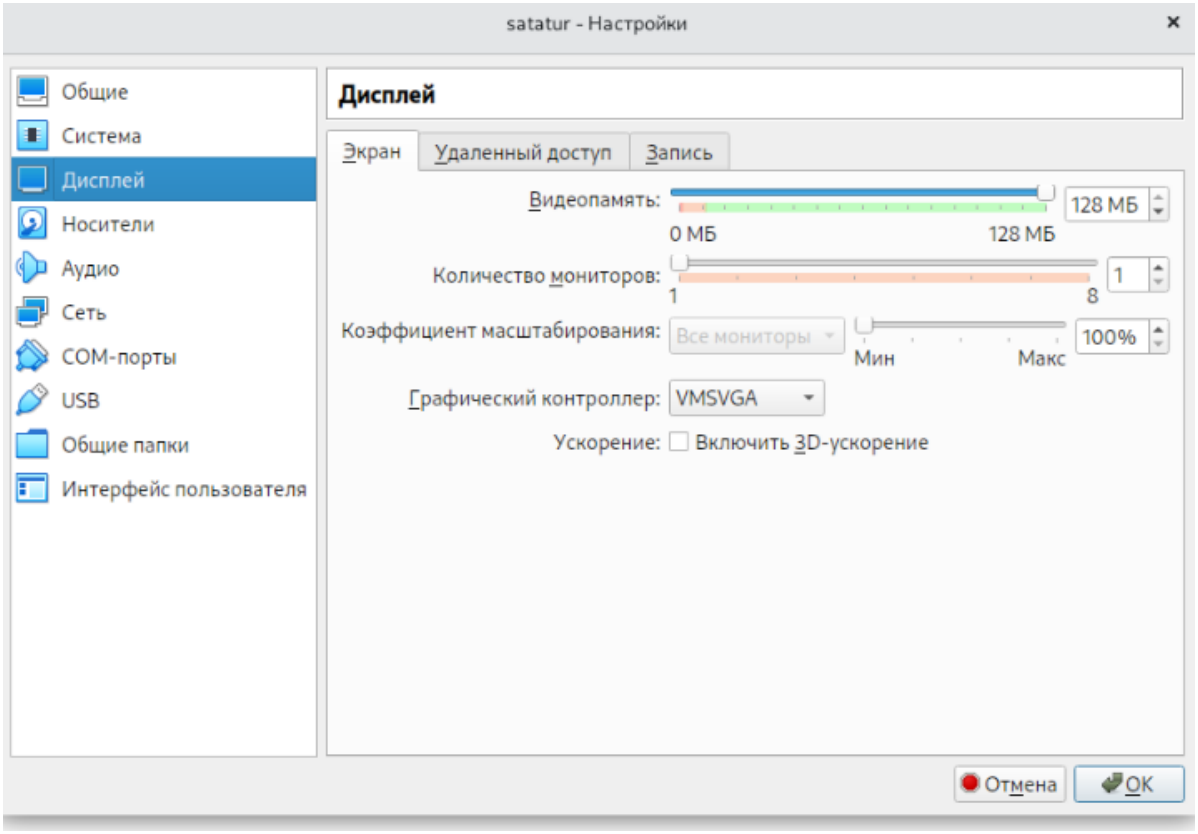


Рис. 2.2: Настройка

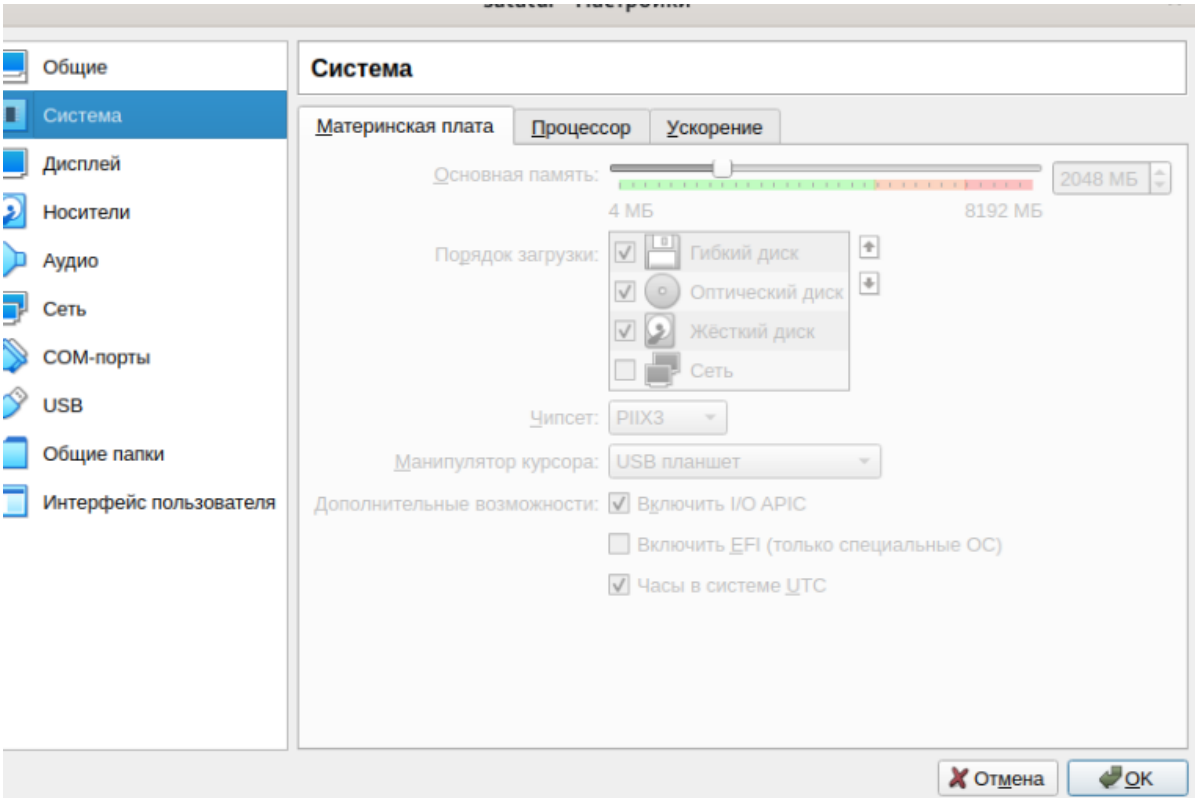


Рис. 2.3: настройка

3. Запустил виртуальную машину и начал установку системы с выбора языка,региона.
4. Начал загрузку операционной системы.
5. После загрузки ввел данные необходимые для входа (имя и пароль)
6. Удостоверившись,что все работает я вышел из системы.

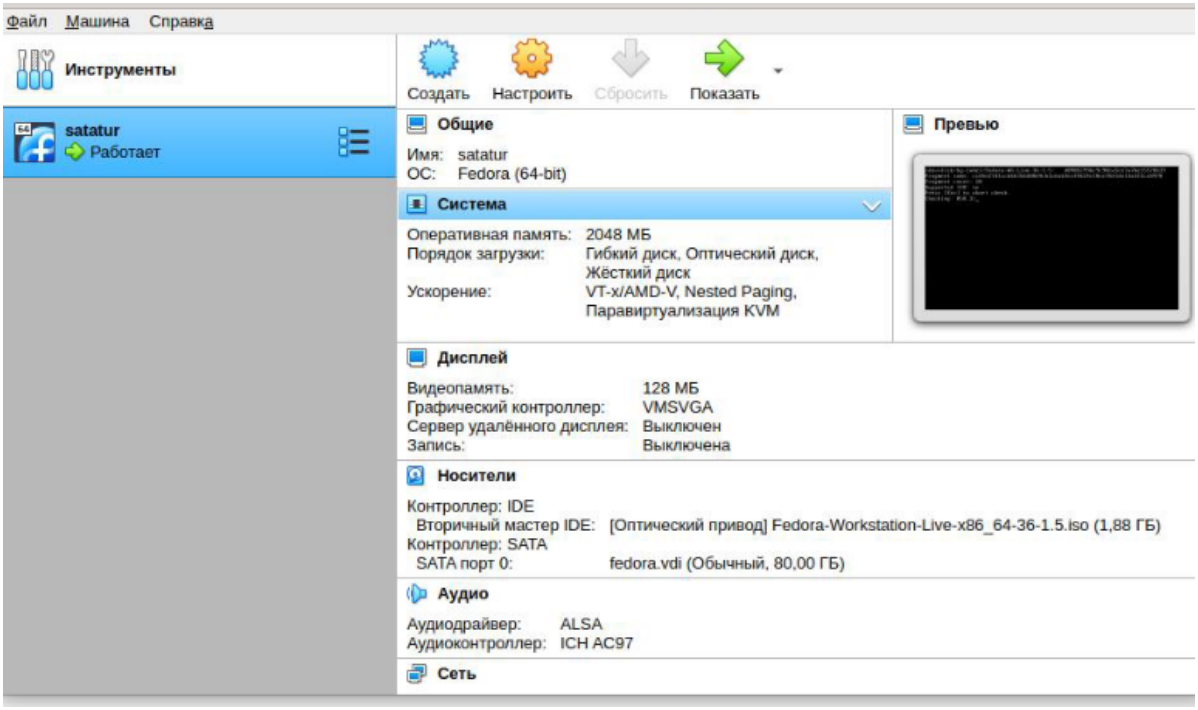


Рис. 2.4: Название рисунка

7. Используя консоль я установил Midninght Commander (mc),так же благодаря команде mc проверил его работу.

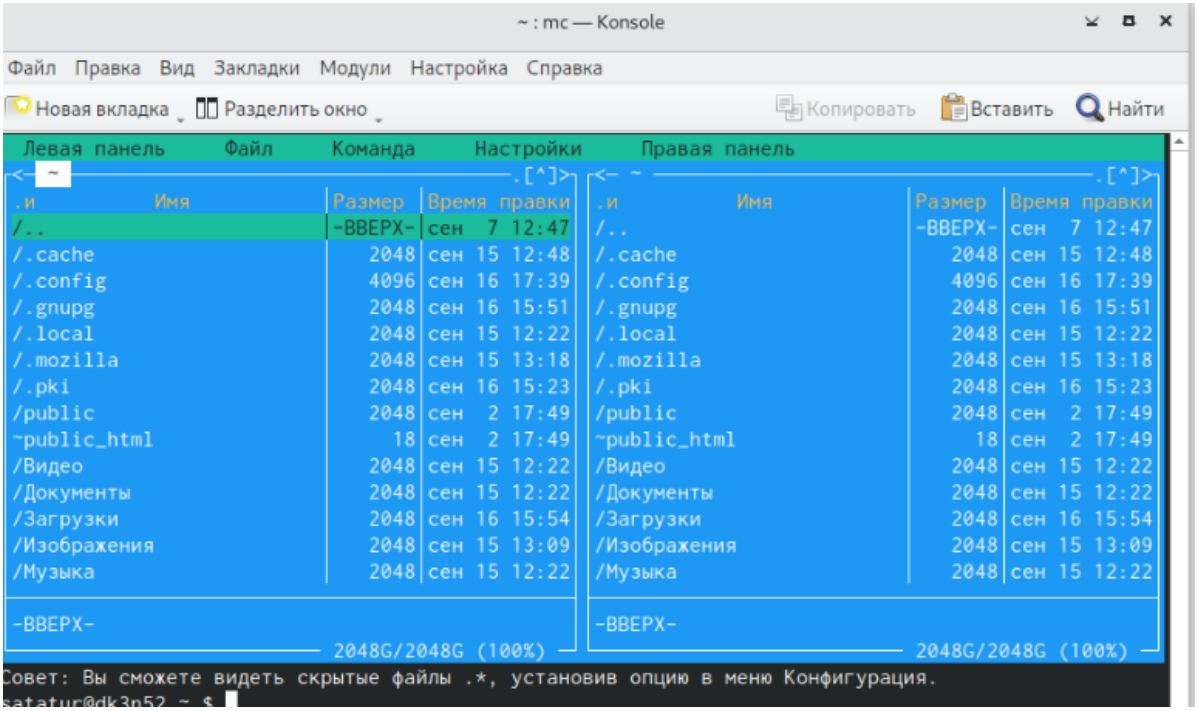


Рис. 2.5: Midninght Commander

8. Проверил наличие системы Git. Благодаря команде “git” вывел данные в консоле.

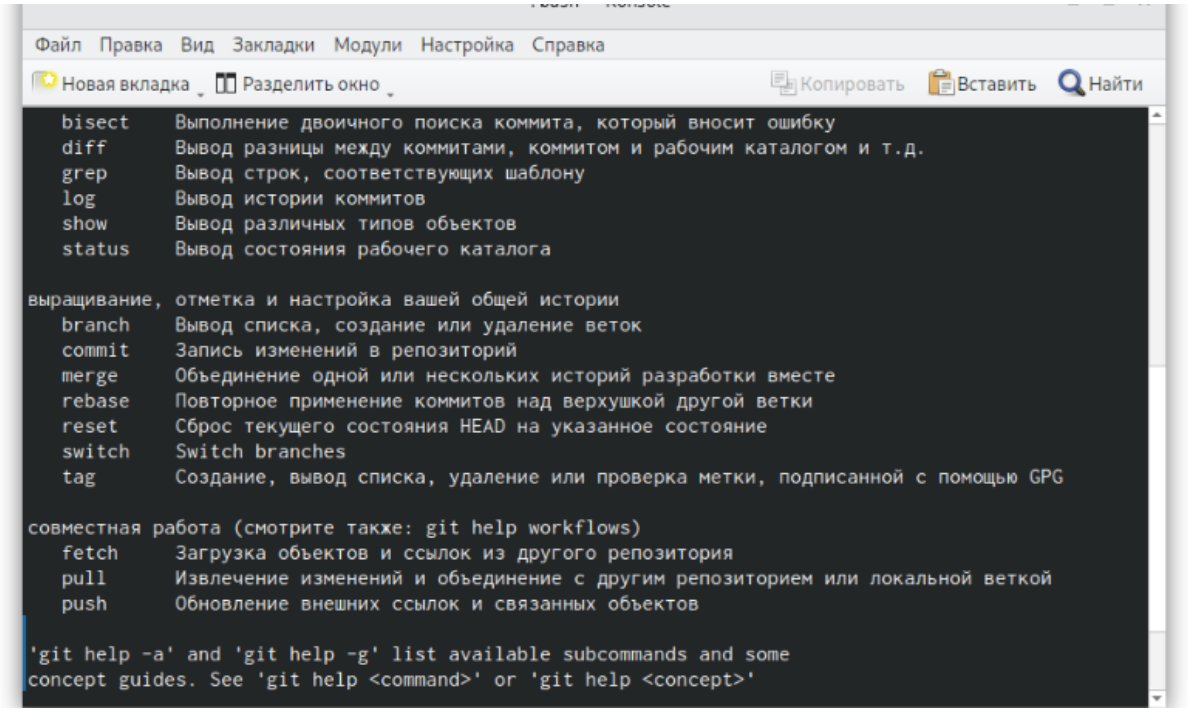


Рис. 2.6: git

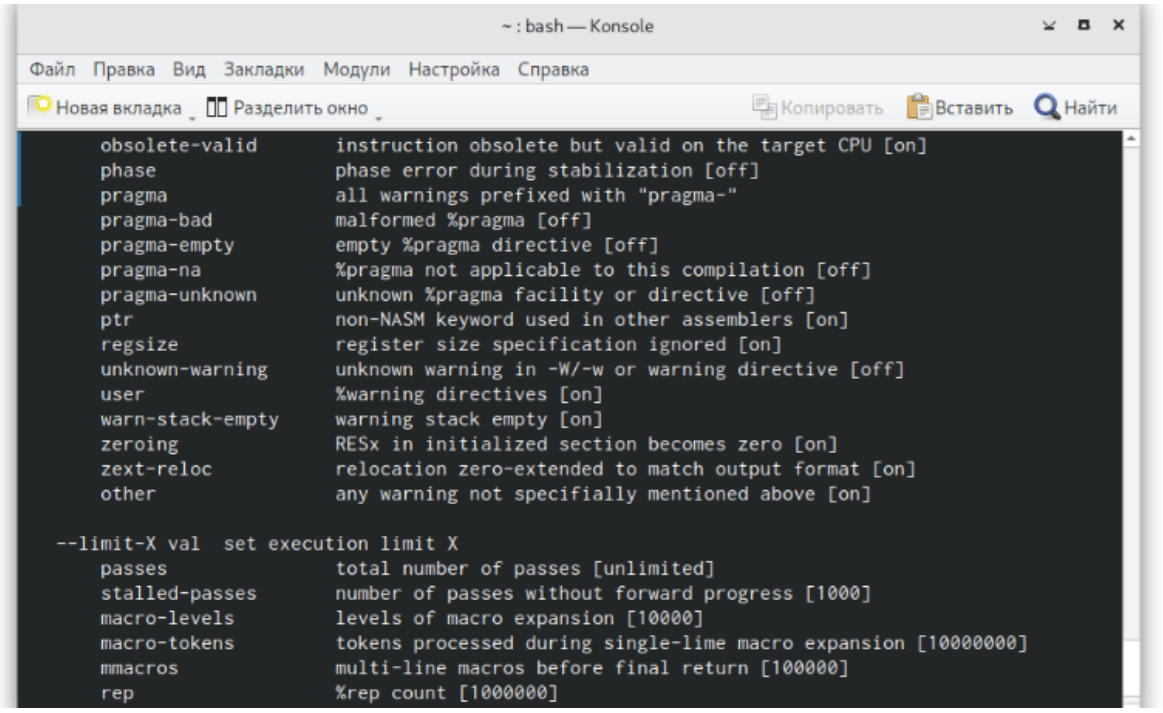


Рис. 2.7: git

9. Далее я перешел в аккаунт супер-пользователь,для того чтобы прописать команду sudo.

И установил пакеты pandoc,а также texlive.

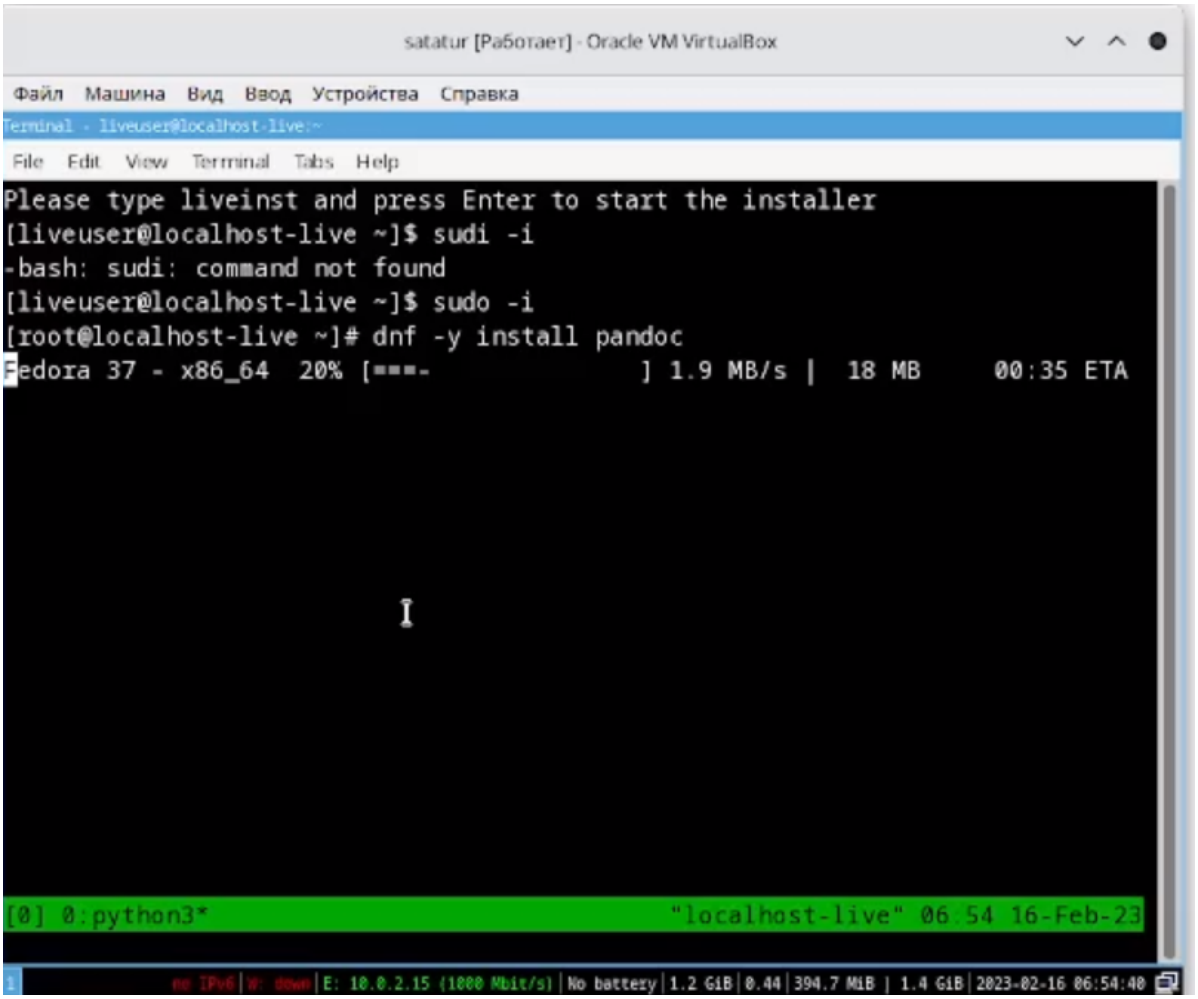


Рис. 2.8: Название рисунка

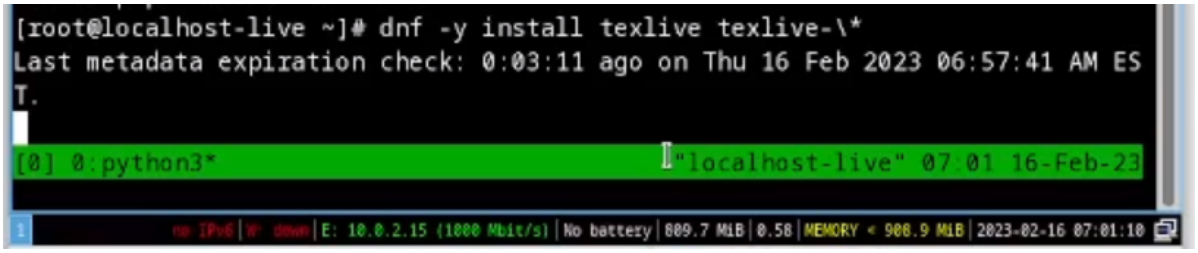


Рис. 2.9: Название рисунка

3 Домашняя работа к лабораторной работе No1

1). Дождался загрузки графического окружения и открыла терминал. В окне терминала проанализировала последовательность загрузки системы, выполнив команду dmesg. Можно просто просмотреть вывод этой команды: dmesg | less

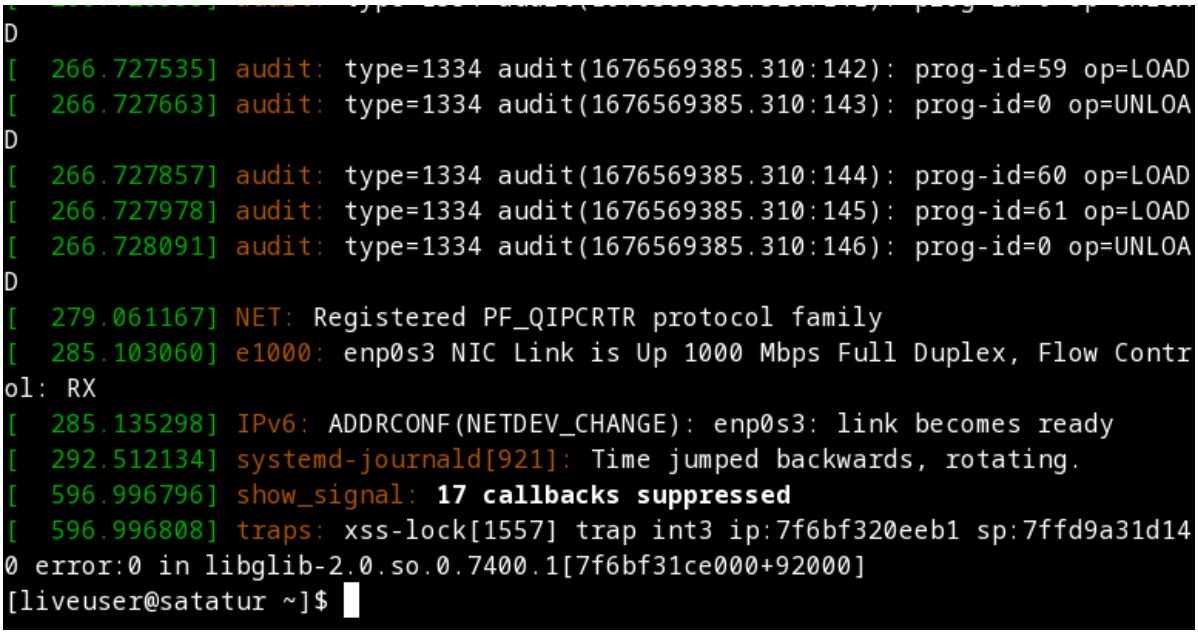
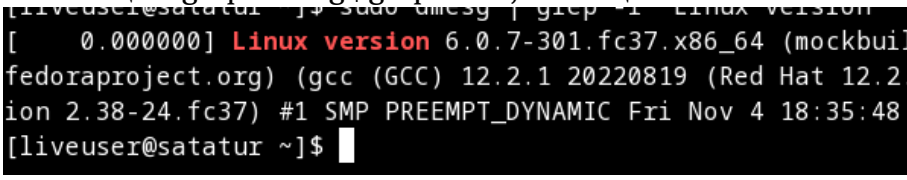


Рис. 3.1: Название рисунка

2). Можно использовать поиск с помощью grep:dmesg | grep -i”то, что ищем” а. Версия ядра Linux (Linux version).



b. Частота процессора (Detected Mhz processor).

```
[liveuser@satatur ~]$ sudo dmesg | grep -i "MHz
[    0.000063] tsc: Detected 1703.998 MHz proce
[    9.170476] e1000 0000:00:03.0 eth0: (PCI:33
[liveuser@satatur ~]$
```

c. Модель процессора (CPU0)

```
[liveuser@satatur ~]$ sudo dmesg | grep -i "CPU0"
[    0.248366] smpboot: CPU0: Intel(R) Core(TM) i5-8400T CPU @ 1
y: 0x6, model: 0x9e, stepping: 0xa)
[liveuser@satatur ~]$
```

d. Объем доступной оперативной памяти (Memory available)

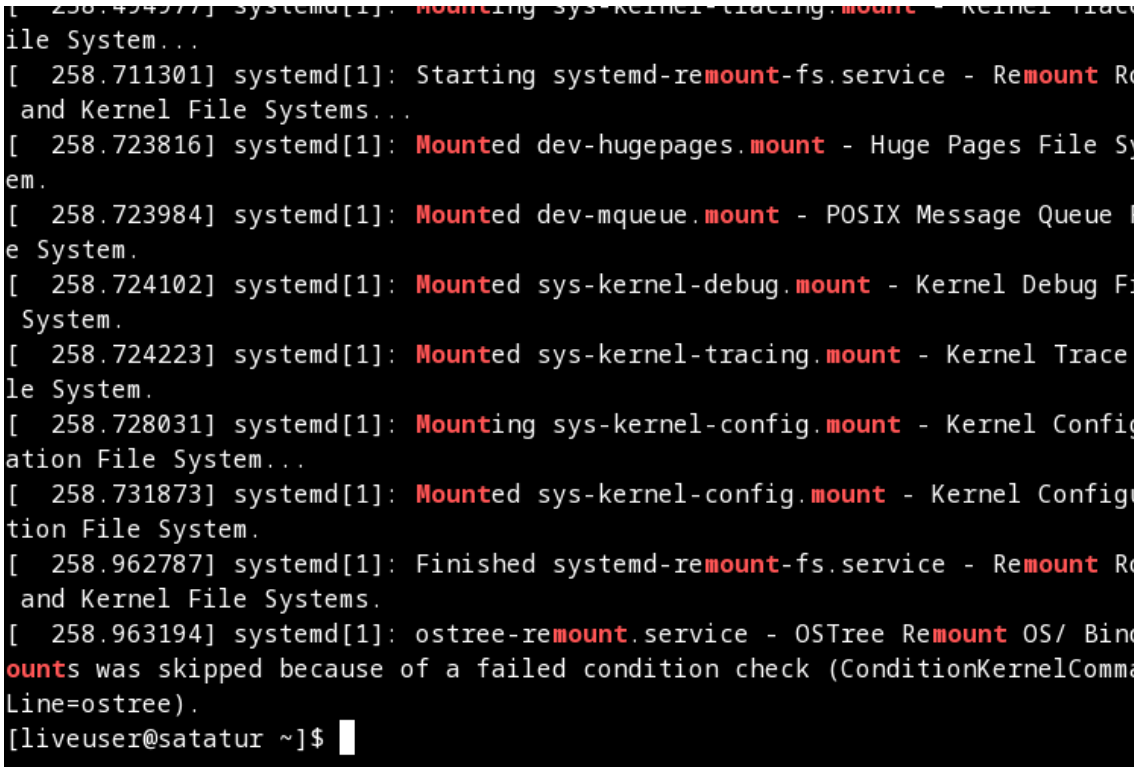
```
[    0.022918] PM: hibernation: Reg
00effff]
[    0.022919] PM: hibernation: Reg
00fffff]
[    0.045070] Memory: 1942500K/209
 rwdata, 12820K rodata, 3024K init
ved)
[    0.142635] Freeing SMP alterna
[    0.249098] x86/mm: Memory bloc
[    4.825418] Freeing initrd memo
[    4.838335] Non-volatile memory
[    5.169972] Freeing unused decr
[    5.170880] Freeing unused kern
[    5.171912] Freeing unused kern
[    5.172981] Freeing unused kern
[  258.470102] systemd[1]: Listeni
f-Memory (OOM) Killer Socket.
[  264.309875] vmwgfx 0000:00:02.0
kB, FIFO = 2048 kB, surface = 5079
[  264.309884] vmwgfx 0000:00:02.0
4 kiB
[liveuser@satatur ~]$
```

e. Тип обнаруженного гипервизора (Hypervisor detected).

```
[    0.000000] Hypervisor detected: KV
[liveuser@satatur ~]$
```

f. Тип файловой системы корневого раздела. Последовательность монтирования

файловых систем



4 Контрольные вопросы:

1) Учетная запись пользователя – это необходимая для системы информация о пользователе, хранящаяся в специальных файлах. Информация используется Linux для аутентификации пользователя и назначения ему прав доступа. Аутентификация – системная процедура, позволяющая Linux определить, какой именно пользователь осуществляет вход. Вся информация о пользователе обычно хранится в файлах /etc/passwd и /etc/group. Учётная запись пользователя содержит: · Имя пользователя (user name) · Идентификационный номер пользователя (UID) · Идентификационный номер группы (GID). · Пароль (password) · Полное имя (full name) · Домашний каталог (home directory) · Начальную оболочку (login shell)
2) Команды терминала:

- Для получения справки по команде: man. Например, команда «man ls» выведет справку о команде «ls».
- Для перемещения по файловой системе: cd. Например, команда «cd newdir» осуществляет переход в каталог newdir.
- Для просмотра содержимого каталога: ls. Например, команда «ls –a ~/newdir» отобразит имена скрытых файлов в каталоге newdir.
- Для определения объёма каталога: du. Например, команда «du –k ~/newdir» выведет размер каталога newdir в килобайтах.
- Для создания / удаления каталогов / файлов: mkdir / rmdir/ rm. Например, команда «mkdir –p ~/newdir1/newdir2» создаст иерархическую цепочку подкаталогов, создав каталоги newdir1 и newdir2; команда «rmdir -v ~/newdir»

удалит каталог newdir; команда «rm –r ~/newdir» так же удалит каталог newdir.

- Для задания определённых прав на файл / каталог: chmod [опции] [путь]. Например, команда «chmod g+r ~/text.txt» даст группе право на чтение файла text.txt.
- Для просмотра истории команд: history. Например, команда «history 7» покажет список последних 7 команд.

3) Файловая система имеет два значения: с одной стороны – это архитектура хранения битов на жестком диске, с другой – это организация каталогов в соответствии с идеологией Unix. Файловая система (англ. «file system») – это архитектура хранения данных в системе, хранение данных в оперативной памяти и доступа к конфигурации ядра. Файловая система устанавливает физическую и логическую структуру файлов, правила их создания и управления ими. В физическом смысле файловая система Linux представляет собой пространство раздела диска, разбитое на блоки фиксированного размера. Их размер кратен размеру сектора: 1024, 2048, 4096 или 8120 байт. Существует несколько типов файловых систем:

- XFS – начало разработки 1993 год, фирма Silicon Graphics, в мае 2000 года предстала в GNU GPL, для пользователей большинства Linux систем стала доступна в 2001-2002 гг. Отличительная черта системы – прекрасная поддержка больших файлов и файловых томов, 8 эксбибайт (8*260 байт) для 64-х битных систем.
- ReiserFS (Reiser3) – одна из первых журналируемых файловых систем под Linux, разработана Namesys, доступна с 2001 г. Максимальный объём тома для этой системы равен 16 тебибайт (16*240 байт).
- JFS (Journaled File System) – файловая система, детище IBM, явившееся миру в далёком 1990 году для ОС AIX (Advanced Interactive eXecutive). В виде первого стабильного релиза, для пользователей Linux, система стала доступна в 2001 году. Из плюсов системы – хорошая масштабируемость. Из минусов

– не особо активная поддержка на протяжении всего жизненного цикла. Максимальный рамер тома 32 пэбибайта (32*250 байт).

- ext (extended filesystem) – появилась в апреле 1992 года, это была первая файловая система, изготовленная специально под нужды Linux ОС. Разработана Remy Card с целью преодолеть ограничения файловой системы Minix.
- ext2 (second extended file system) – была разработана Remy Card в 1993 году. Не журналируемая файловая система, это был основной её недостаток, который исправит ext3.
- ext3 (third extended filesystem) – по сути расширение исконной для Linux ext2, способное к журналированию. Разработана Стивеном Твиди в 1999 году, включена в основное ядро Linux в ноябре 2001 года. На фоне других своих сослуживцев обладает более скромным размером пространства, до 4 тебибайт (4*240 байт) для 32-х разрядных систем. На данный момент является наиболее стабильной и поддерживаемой файловой системой в среде Linux.
- Reiser4– первая попытка создать файловую систему нового поколения для Linux. Впервые представленная в 2004 году, система включает в себя такие передовые технологии как транзакции, задержка выделения пространства, а так же встроенная возможность кодирования и сжатия данных. Ханс Рейзер (Hans Reiser) – главный разработчик системы. -xt4 – попытка создать 64-х битную ext3 способную поддерживать больший размер файловой системы (1 эксбибайт). Позже добавились возможности – непрерывные области дискового пространства, задержка выделения пространства, онлайн дефрагментация и прочие. Обеспечивается прямая совместимость с системой ext3 и ограниченная обратная совместимость при недоступной способности к непрерывным областям дискового пространства.
- Btrfs (B-tree FS или Butter FS)– проект изначально начатый компанией Oracle, впоследствии поддержанный большинством Linux систем. Ключевы-

ми особенностями данной файловой системы являются технологии: copy-on-write, позволяющая сделать снимки областей диска (снапшоты), которые могут пригодится для последующего восстановления; контроль за целостностью данных и метаданных (с повышенной гарантией целостности); сжатие данных; оптимизированный режим для накопителей SSD (задаётся при монтировании) и прочие. Немаловажным фактором является возможность перехода с ext3 на Btrfs. С августа 2008 года данная система выпускается под GNU GPL.

- Tux2 – известная, но так и не анонсированная публично файловая система. Создатель Дэниэл Филипс (Daniel Phillips). Система базируется на алгоритме «Фазового Дерева», который как и журналирование защищает файловую систему от сбоев. Организована как надстройка на ext2.
- Tux3 – система создана на основе FUSE (Filesystem in Userspace), специального модуля для создания файловых систем на Unix платформах. Данный проект ставит перед собой цель избавиться от привычного журналирования, взамен предлагая версионное восстановление (состояние в определённый промежуток времени). Преимуществом используемой в данном случае версионной системы, является способ описания изменений, где для каждого файла создаётся изменённая копия, а не переписывается текущая версия.
- Xiafs–задумка и разработка данной файловой системы принадлежат Frank Xia, основана на файловой системе MINIX. В настоящее время считается устаревшей и практически не используется. Наряду с ext2 разрабатывалась, как замена системе ext. В декабре 1993 года система была добавлена в стандартное ядро Linux. И хотя система обладала большей стабильностью и занимала меньше дискового пространства под контрольные структуры –она оказалась слабее ext2, ведущую роль сыграли ограничения максимальных размеров файла и раздела, а так же способность к дальнейшему расширению.
- ZFS (Zettabyte File System)–изначально созданная в Sun Microsystems файло-

вая система, для небезызвестной операционной системы Solaris в 2005 году. Отличительные особенности –отсутствие фрагментациииданных как таковой, возможности по управлению снапшотами (snapshots), пулами хранения (storage pools), варьируемый размер блоков, 64-х разрядный механизм контрольных сумм, а так же способность адресовать 128 бит информации. В Linux системах может использоваться посредствам FUSE.

4) Команда «findmnt» или «findmnt–all» будет отображать все подмонтированные файловые системы или искать файловую систему.
5) Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса:

- SIGINT –самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш Ctrl+C. Процесс правильно завершает все свои действия и возвращает управление;
- SIGQUIT –это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дамп памяти. Сочетание клавиш Ctrl+/;
- SIGHUP –сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом;
- SIGTERM –немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;
- SIGKILL –тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными.Также для

передачи сигналов процессам в Linux используется утилита kill, её синтаксис:kill [-сигнал][pid_процесса](PID–уникальный идентификатор процесса). Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса. Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды psигrep. Команда ps предназначена для вывода списка активных процессов всистеме иинформациионих. Команда grep запускается одновременно cps (в канале) и будет выполнять поиск по результатам команды ps.Утилита pkill –это оболочка для kill, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать его имя. killall работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории /proc. Но эта утилита обнаружит все процессы с таким именем и завершит их. # Выводы

Я приобрел практические навыки по установки операционной системы на виртуальную машину. Также, установил все настройки сервисов для дальнейшей работы.

Список литературы