

Лабораторная работа номер 1

Выполнение лабораторной работы

Титков Ярослав Максимович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	12
6	Контрольные вопросы	13

Список иллюстраций

4.1	Установка операционной системы	8
4.2	Обновление операционной системы	8
4.3	Делаем систему более комфортной	9
4.4	Снятие защиты через mc	9
4.5	Установка dmsk	10
4.6	Работа с ядром	10
4.7	Установка Pandoc	10
4.8	Проверяем установку нужных утилит для работы с файлами	11

Список таблиц

1 Цель работы

Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

2 Задание

1. Установка операционной системы
2. Действия после установки
3. Установка имени пользователя и названия хоста
4. Установка программного обеспечения для создания документации

3 Теоретическое введение

Лабораторная работа подразумевает установку на виртуальную машину VirtualBox (<https://www.virtualbox.org/>) операционной системы Linux (дистрибутив Fedora).

Выполнение работы возможно как в дисплейном классе факультета физико-математических и естественных наук РУДН, так и дома. Описание выполнения работы приведено для дисплейного класса со следующими характеристиками техники:

- Intel Core i3-550 3.2 GHz, 4 GB оперативной памяти, 80 GB свободного места на жёстком диске;
- ОС Linux Gentoo (<http://www.gentoo.ru/>);
- VirtualBox версии 7.0 или новее.

Для установки в виртуальную машину используется дистрибутив Linux Fedora (<https://getfedora.org>), вариант с менеджером окон sway (<https://fedoraproject.org/spins/sway/>).

При выполнении лабораторной работы на своей технике вам необходимо скачать необходимый образ операционной системы (<https://fedoraproject.org/spins/sway/download/index>).

В дисплейных классах можно воспользоваться образом в каталоге `/afs/dk.sci.pfu.edu.ru/comp`.

Для определённости в описании будем использовать версию Fedora-Sway-Live-x86_64-41-1.4.iso.

4 Выполнение лабораторной работы

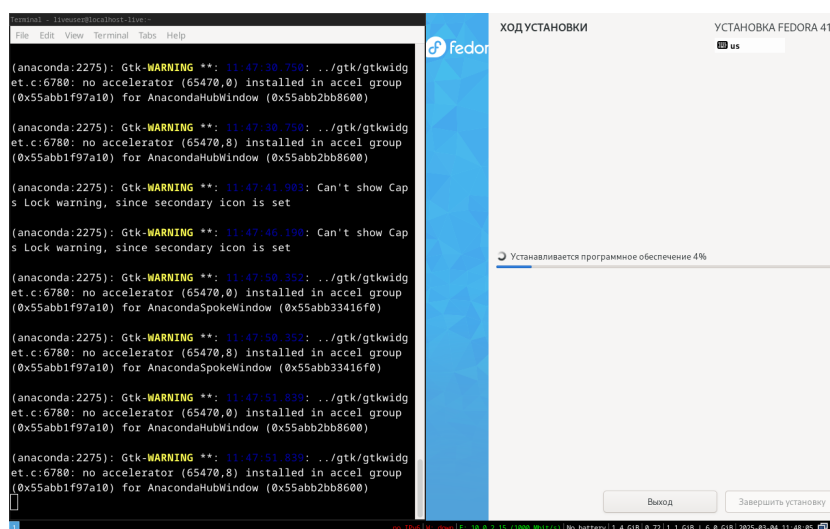


Рис. 4.1: Установка операционной системы

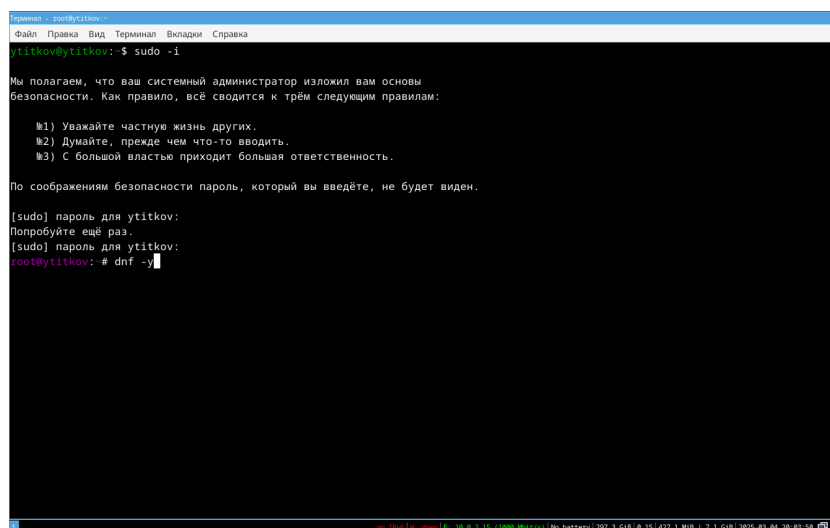


Рис. 4.2: Обновление операционной системы


```
Erasing fedora-release-identity-basic-0:41-25.noarch      100% | 200.0 B/s | 1.0 B | 00m00s
Erasing libx11-common-0:1.8.10-2.fc41.noarch             100% | 23.0 KiB/s | 259.0 B | 00m00s
Erasing crypto-policies-0:20241010-1.git8baf557.fc41.noarch 100% | 16.9 KiB/s | 190.0 B | 00m00s
Erasing tzdata-0:2024a-9.fc41.noarch                    100% | 152.0 KiB/s | 1.8 KiB | 00m00s
Erasing coreutils-common-0:9.5-10.fc41.x86_64           100% | 22.4 KiB/s | 252.0 B | 00m00s
Erasing adobe-mappings-cmap-0:20230622-4.fc41.noarch     100% | 33.7 KiB/s | 287.0 B | 00m00s
Erasing grub2-common-1:2.12-10.fc41.noarch               100% | 686.0 B/s | 57.0 B | 00m00s
Erasing setroubleshoot-plugins-0:3.14-10.fc41.noarch     100% | 12.5 KiB/s | 205.0 B | 00m00s
Erasing m17n-db-0:1.8.8-1.fc41.noarch                    100% | 27.8 KiB/s | 455.0 B | 00m00s
Erasing elfutils-default-yama-scope-0:0.191-8.fc41.noarch 100% | 200.0 B/s | 1.0 B | 00m00s
Erasing glibc-0:2.40-3.fc41.x86_64                      100% | 5.9 KiB/s | 103.0 B | 00m00s
Erasing glibc-all-langpacks-0:2.40-3.fc41.x86_64        100% | 5.7 KiB/s | 41.0 B | 00m00s
Erasing glibc-gconv-extra-0:2.40-3.fc41.x86_64          100% | 38.1 KiB/s | 624.0 B | 00m00s
>>> Running post-uninstall scriptlet: glibc-gconv-extra-0:2.40-3.fc41.x86_64warning: posix.fork(): .fork(), .exec(), .wait()
and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.wait(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.exec(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
Erasing glibc-gconv-extra-0:2.40-3.fc41.x86_64          100% | 13.2 KiB/s | 624.0 B | 00m00s
Erasing glibc-common-0:2.40-3.fc41.x86_64               100% | 6.3 KiB/s | 52.0 B | 00m00s
Erasing libgcc-0:14.2.1-3.fc41.x86_64                   100% | 0.0 B/s | 11.0 B | 00m00s
>>> Running post-uninstall scriptlet: libgcc-0:14.2.1-3.fc41.x86_64warning: posix.fork(): .fork(), .exec(), .wait() and .red
irect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.wait(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.exec(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
Erasing libgcc-0:14.2.1-3.fc41.x86_64                   100% | 0.0 B/s | 11.0 B | 00m00s
Erasing libgcc-0:14.2.1-3.fc41.x86_64                   100% | 0.0 B/s | 11.0 B | 02m16s
Complete!
root@yitikov: #
root@yitikov: # dns install tmux mc
```

Рис. 4.3: Делаем систему более комфортной

```
config [----] 15 L: [ 3+ 2 5/ 30] *(175 /1187b) 0101 0x065 [*] [X]
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
# See also:
# https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#getting-started-with-selinux-selinux-states-a
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux during boot. If you need a system with SELinux
# fully disabled instead of SELinux running with no policy loaded, you
# need to pass selinux=0 to the kernel command line. You can use grubby
# to persistently set the bootloader to boot with selinux=0:
#
# grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
# grubby --update-kernel ALL --remove-args selinux
#
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mis - Multi Level Security protection.
SELINUXTYPE=targeted

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Перем-тить 7Линск 8Удалить 9ЧениМС 10Выход
[0] 8:mc*~M "mc [yitikov@yitikov.y" 20:21 04-мар-25
```

Рис. 4.4: Снятие защиты через mc

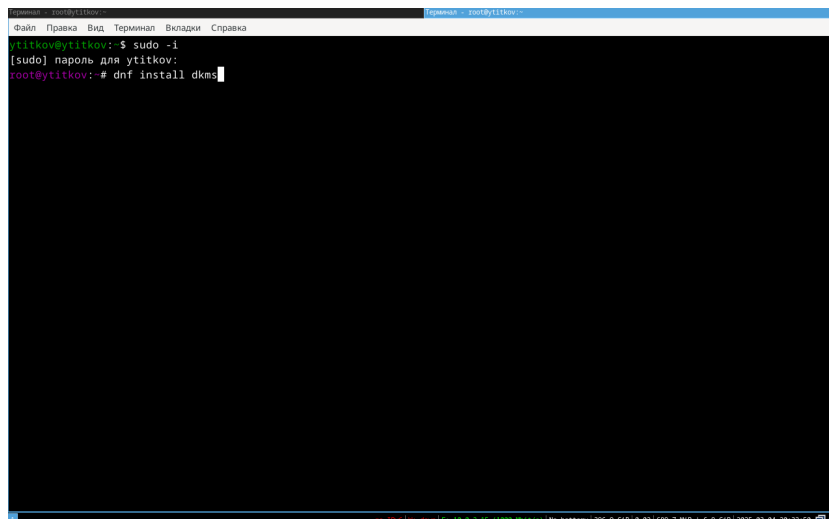


Рис. 4.5: Установка dmsk

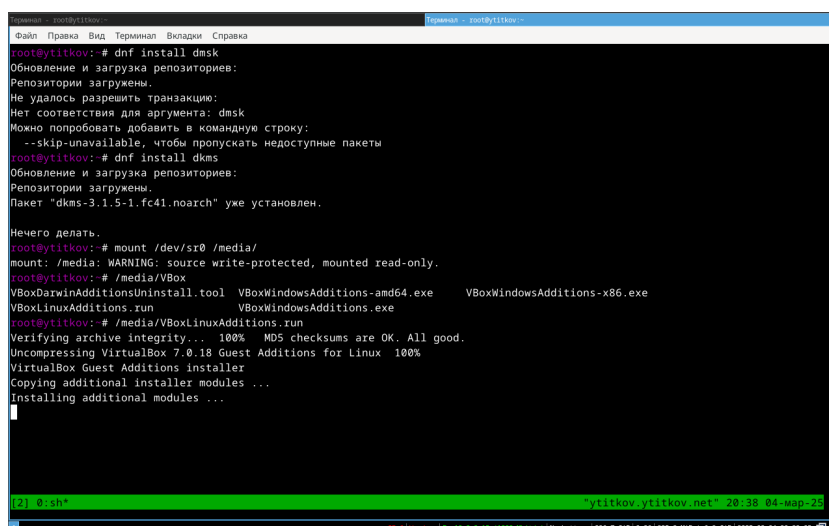


Рис. 4.6: Работа с ядром

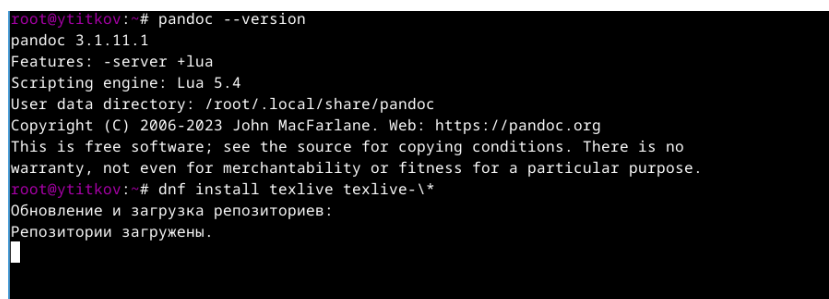


Рис. 4.7: Установка Pandoc

```

\pdfxchar{texlive\texmf-dist\tex-generic\hyph-utf8\loadhyph\loadhyph-pt.tex}
%#-8 Portuguese hyphenation patterns
\pdfxchar{texlive\texmf-dist\tex-generic\hyph-utf8\patterns\tex-hyph-pt.tex}
! Missing number, treated as zero.
%to be read again-
\chardef
%to be read again-
\rm@arabic
\addlanguage ... \chardef \csname rnm@lencsname
                                     %%< ifdefined <directlua ...
\endgroup
\if 70 \addlanguage(arabic)(zerohyph.tex){}{(){}
}
! Emergency stop.
%to be read again-
\chardef
%to be read again-
\rm@arabic
\addlanguage ... \chardef \csname rnm@lencsname
                                     %%< ifdefined <directlua ...
\endgroup
\if 70 \addlanguage(arabic)(zerohyph.tex){(){}}
{
%% Fatal error occurred, no output PDF file produced!
! Transcript written on lutex.log
!> fatal [INFO]: log file copied to : /home/yttikov/texlive2023/texmf-var/webzcr/lutex/lutex_log
!> fatal [ERROR]: running lutex -ini -j none-live progname=lutex lutex.ini =>2</dev/null' return status: 1
!> fatal [ERROR]: returning error due to option -strict
!> fatal [INFO]: not selected format: 64
!> fatal [INFO]: failed to build: 3 {lutex:lutex}
!> fatal [INFO]: total format: 55
!> fatal [INFO]: mixing with status 1
! can't find the format file 'lutex.fat'!
!> showmyinfo: 4
!> showmyinfo: 4 select
This is XeTeX, Version 3.141592653-2.6.0-999995 (TeX Live 2023) (preloaded format=xelatex)
!> fatal [FATAL] fatal: fatal
!> fatal [FATAL] fatal

```

Рис. 4.8: Проверяем установку нужных утилит для работы с файлами

5 Выводы

В ходе выполнения лабораторной работы мы приобрели практические навыки выполнения лабораторной работы.

6 Контрольные вопросы

1. **Учётная запись пользователя** содержит имя пользователя, UID, GID, домашний каталог, оболочку (shell), пароль (хранится в зашифрованном виде).

2. **Команды терминала:**

- Получение справки: `man` команда (пример: `man ls`).
- Перемещение по файловой системе: `cd` путь (пример: `cd /home/user`).
- Просмотр содержимого каталога: `ls` (пример: `ls -la`).
- Определение объёма каталога: `du -sh` каталог (пример: `du -sh /var/log`).
- Создание / удаление каталогов / файлов:
 - `mkdir` каталог (пример: `mkdir test`).
 - `rmdir` каталог (пример: `rmdir test`).
 - `touch` файл (пример: `touch file.txt`).
 - `rm` файл (пример: `rm file.txt`).

- Задание прав на файл / каталог:
 - `chmod 755 файл` (пример: `chmod 755 script.sh`).
 - `chown user:group файл` (пример: `chown user:user file.txt`).
- Просмотр истории команд: `history`.

3. **Файловая система** – способ организации, хранения и управления данными на диске. Примеры:

- `ext4` – стандарт для Linux, журналируемая.
- `XFS` – высокопроизводительная, хороша для больших файлов.
- `Btrfs` – поддержка снимков, дедупликация.
- `NTFS` – используется в Windows, поддерживается в Linux.

4. **Просмотр смонтированных файловых систем:**

- `mount`
- `df -T`
- `lsblk -f`

5. **Удаление зависшего процесса:**

- `kill PID` (пример: `kill 1234`).
- `kill -9 PID` (жёсткое завершение).
- `htop` → выбор процесса → F9.