**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Программа ДПО Intaro

ЛАБОРАТОРНАЯ РАБОТА №2

“Работа с файловой системой ОС Linux”

Студент \_\_\_\_\_\_\_\_\_\_ Целищев А.Е.

ПМ-21-2

Руководитель \_\_\_\_\_\_\_\_\_\_ Кургасов В.В.

Доцент каф. АСУ

Липецк 2023 г

Оглавление

[Цели работы 3](#_Toc9114)

[Задание к лабораторной работе 3](#_Toc12389)

[Установка VM VirtualBox 4](#_Toc27089)

[Создание виртуальной машины, ее настройка 5](#_Toc4632)

[Установка гостевой ОС Linux (Ubuntu Server) 10](#_Toc32722)

[Настройка текстового редактора Vim, разработка на Linux 15](#_Toc5737)

[Выводы 20](#_Toc14593)

[Ответы на контрольные вопросы 21](#_Toc19725)

**Цель работы**

Приобрести опыт работы с файлами и каталогами в ОС Linux, настройки прав на доступ к файлам и каталогам.

1. **Общая часть**

1.2. FHS (Filesystem Hierarchy Standard) - стандарт, унифицирующий местонахождение файлов и каталогов с общим назначением в файловой системе UNIX. Большинство UNIX-подобных систем так или иначе следует этому стандарту, хотя есть Linux-дистрибутивы, отвергающие этот стандарт, например, GoboLinux. В то же время, например, Mac OS X, использует собственные имена /Library/, /Applications/, /Users/ вместе с традиционные именами UNIX-иерархии.

В FHS все файлы и каталоги находятся внутри корневого каталога, даже если они расположены на различных физических носителях.

1.3. Перечень основных каталогов согласно FHS

|  |  |
| --- | --- |
| Путь | Описание |
| / | Корневой каталог, содержащий всю файловую иерархию. |
| /bin | Основные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям. |
| /boot | Загрузочные файлы (файлы загрузчика, ядро, initrd, System.map) |
| /dev | Основные файлы устройств |
| /etc | Общесистемные конфигурационные файлы |
| /etc/opt | Файлы конфигурации для opt |
| /etc/X11 | Файлы конфигурации X Windows System версии 11 |
| /etc/sgml | Файлы конфигурации SGML |
| /etc/xml | Файлы конфигурации XML |
| /home | Содержит домашние каталоги пользователей, которые в свою очередь содержат персональные настройки и данные пользователя. |
| /lib | Основные библиотеки, необходимые для работы программ из /bin и /sbin |
| /media | Точки монтирования для сменных носителей (CD-ROM, DVD-ROM и тд) |
| /mnt | Содержит временно монтируемые файловые системы |
| /opt | Дополнительное программное обеспечение |
| /proc | Виртуальная файловая система, представляющая состояние ядра операционной системы и запущенных процессов в виде файлов |
| /root | Домашний каталог пользователя root |
| /run | Информация о системе с момента ее загрузки, в том числе данные, необходимые для работы демонов (pid-файлы, UNIX-сокеты и тд.) |
| /sbin | Основные системные программы для администрирования и настройки системы |
| /srv | Данные для сервисов, предоставляемых системой (например, www или ftp) |
| /sys | Содержит информацию об устройствах, драйверах, а также некоторых свойствах ядра |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя. Содержит большинство пользовательских приложений и утилит, используемых в многопользовательском режиме. Может быть смонтирована по сети только для чтения и быть общей для нескольких машин |
| /usr/bin | Дополнительные программы для всех пользователей, не являющиеся необходимыми в однопользовательском режиме |
| /usr/include | Стандартные заголовочные файлы |
| /usr/lib | Библиотеки для программ, находящихся в usr/bin и usr/sbin |
| /usr/local | Третичная иерархия для данных, специфичных для данного хоста. Обычно содержит такие подкаталоги, как bin, lib, share |
| /usr/sbin | Дополнительные системные программы (например, демоны сетевых сервисов) |
| /usr/share | Архитектурно-независимые общие данные |
| /usr/src | Исходные коды (например, здесь располагаются исходные коды ядра) |
| /var | Изменяемые файлы, такие как файлы регистрации, временные почтовые файлы и тд |
| /var/cache | Кэш приложений. Такие данные генерируются локально в результате ресурсозатратных вычислений или операций ввода-вывода. Приложение обязано уметь регенерировать эти данные. Данные файлы могут быть удалены без потери данных |
| /var/lib | Информация о состоянии. Постоянные данные, изменяемые программами в процессе работы (например, базы данных, метаданные пакетного менеджера и др.) |
| /var/lock | Файлы блокировки, указывающие на занятость некоторого ресурса |
| /var/log | Различные файлы регистрации |
| /var/mail | Почтовые ящики пользователей |
| /var/run | Каталог для хранения информации о системе с момента её загрузки, на данный момент устарел. Допускается делать его ссылкой на /run. Каталог оставлен для обратной совместимости с программами, которые всё ещё используют каталог /var/run |
| /var/spool | Задачи, ожидающие обработки (например, очереди печати, непрочитанные или неотправленные письма) |
| /var/spool/mail | Местоположение пользовательских почтовых ящиков (устаревшее) |
| /var/tmp | Временные файлы, которые должны быть сохранены между перезагрузками |

1.4. Зайдем в терминал под root, введя sudo bash и пароль.



Рис 1. Вход в терминал под root.

1.5. Введем lsblk чтобы вывести содержимое каталога файлов физических устройств:

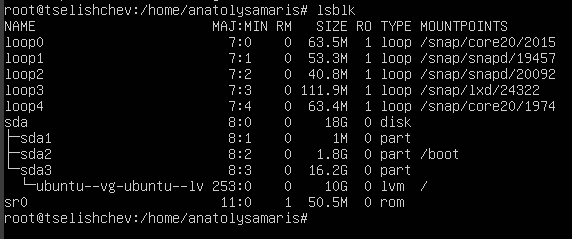


Рис 2. Вывод содержимого каталога файлов физических устройств.

В списке устройств видим имена loopX, sda, sr0.

- sda: это обозначение для первого блочного устройства, обычно это жесткий диск или твердотельный накопитель. "sdb" будет вторым устройством, "sdc" - третьим и так далее.

- loop: это обозначение для устройств loopback, которые позволяют монтировать файлы и образы как блочные устройства. Это может использоваться, например, для монтирования образов дисков или файловых систем в файле.

- sr0: это обозначение для устройств CD/DVD-ROM. "sr0" обычно является первым устройством CD/DVD-ROM в системе. Если есть несколько устройств CD/DVD-ROM, то они могут быть обозначены как "sr1", "sr2" и так далее.

1.6. Перейдем в директорию пользователя root, просмотрим содержимое. Нас интересует местонахождение файла vmlinuz, который должен быть в каталоге boot. Убедимся в этом:

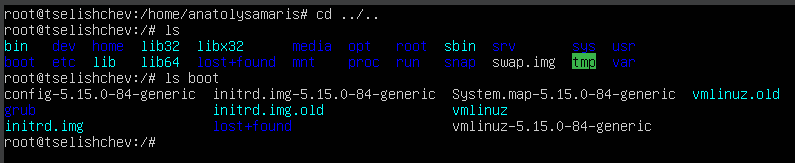


Рис 3. Переход в директорию пользователя и проверка директории boot.

Откроем файл vmlinuz, используя текстовый редактор Vim.

Командой ls -l vmlinuz просмотрим права доступа к файлу. Видим вывод lrwxrwxrwx. Здесь последовательно выводятся права доступа сначала для владельца файла, потом для группы пользователей, привязанных к файлу, и затем для всех остальных пользователей. Символ «l» перед правами доступа означает тип файла. Утилита ls обозначает символом «l» символические ссылки. Таким образом, каждый имеет доступ на чтение (“r”), запись (“w”) и исполнение (“x”) символической ссылки vmlinuz.



Рис 4. Просмотр прав доступа к файлу.

1.7. Создадим нового пользователя user командой adduser user

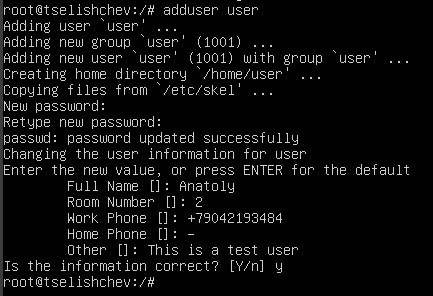


Рис 5. Создание пользователя user.

1.8. Перейдем в профиль пользователя user и создадим 3 текстовых файла, используя touch, cat, и vim

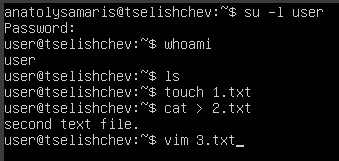


Рис 6. Вход в профиль user и создание текстовых файлов различными способами.

Используя команду ls -l мы можем посмотреть тип файла и права доступа к файлу. Как можем видеть, все созданные файлы имеют -rw-rw-r--. «-» в начале означает, что это обычный файл (а расширение .txt могло бы подсказывать системе, какой именно утилитой/программой обрабатывать файл), дальше следуют разрешения на чтение и запись для владельца (создателя) файла и группы пользователей, к которой принадлежит владелец (в нашем случае группа пользователей, в которую входит user, состоит только из него), и разрешение лишь на чтение для всех остальных пользователей.

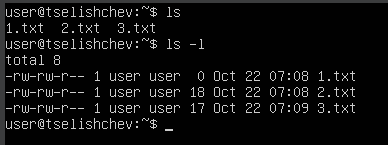


Рис 7. Просмотр типа созданных файлов прав доступа к ним.

1.9. Перейдем к директории пользователя root. Видим то же, что в пункте 1.6.



Рис 8. Директория пользователя root.

1.10. Изменим права доступа к файлу 1.txt командой chmod ugo+rwx 1.txt, которая выдаст все права всем пользователям (u - владелец, g - группа владельца, o - другие пользователи; + - добавить права, r - чтение, w - запись, x - исполнение)

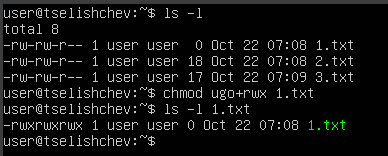


Рис 9. Изменение прав доступа к файлу 1.txt.

1.11. Создадим символическую и жесткую ссылки на файл 2.txt командами

ln -s 2.txt <имя\_ссылки> и ln 2.txt <имя ссылки> соответственно.

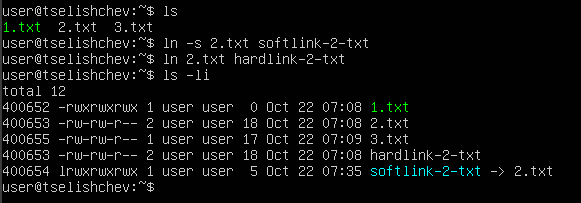


Рис 10. Создание символической и жесткой ссылок на файл 2.txt.

1.12. Создадим каталог new командой mkdir <имя\_каталога>

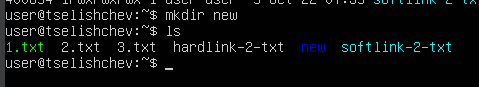


Рис 11. Создание каталога new.

1.13. Копируем файл 1.txt в каталог new с новым именем copy-1.txt с помощью команды cp 1.txt new/copy-1.txt



Рис 12. Копирование файла 1.txt в каталог new с новым именем copy-1.txt.

1.14. Переместим файл 2.txt в каталог new командой mv 2.txt new/



Рис 13. Перемещение файла 2.txt в каталог new.

1.15. Зайдем в профиль root-пользователя и сделаем его владельцем файла 3.txt и каталога new chown <имя\_пользователя> <имя\_файла\_или\_каталога>

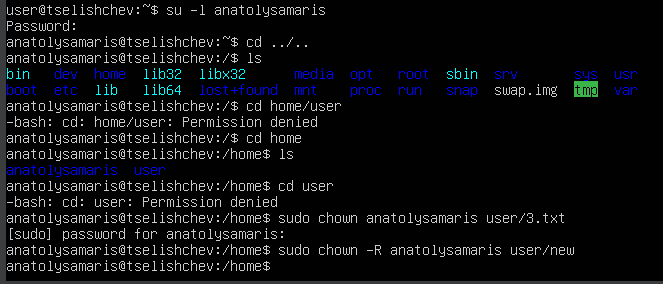


Рис 14. Изменение владельца файла 3.txt и каталога new.

1.16-1.17. Удалим копию файла 1.txt из каталога new, а затем и сам каталог new командами rm <имя\_файла> и rm -Rf <имя\_каталога>. Здесь для удаления каталога используем rm -Rf вместо rmdir чтобы рекурсивно (-R) пройти по всем имеющимся в каталоге файлам и удалить их, не спрашивая разрешения (-f). Затем будет удален и сам каталог.

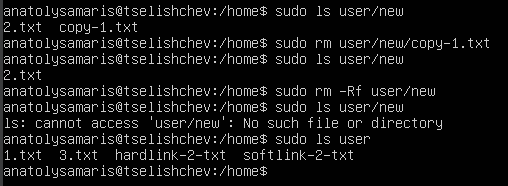
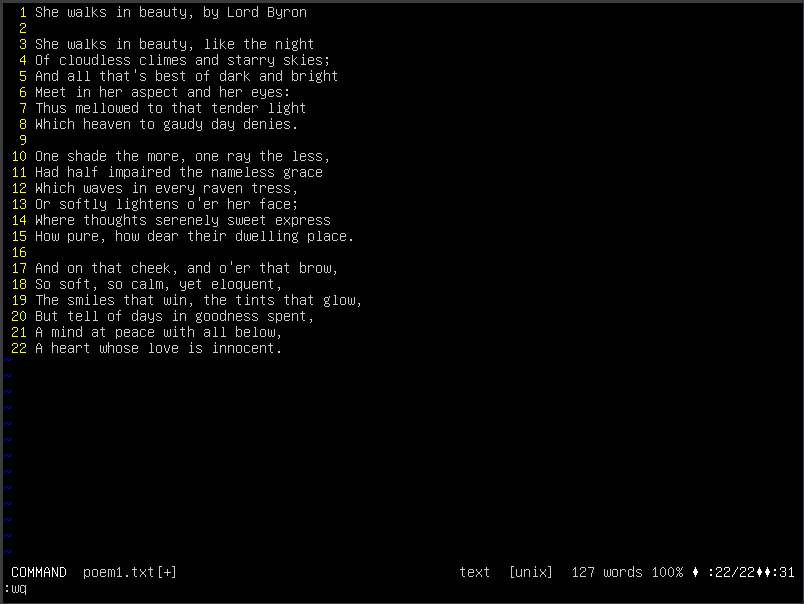


Рис 15. Удаление файла copy-1.txt и каталога new.

1. **Файлы и каталоги**
   1. Создадим 3 текстовых файла разными способами. Для создания с помощью редактора vim используем команду vi <имя\_файла>0



Прежде чем использовать mc (Midnight Commander), установим соответствующую программу: sudo apt install mc

