

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет прикладной математики и информатики
Кафедра технологий программирования

Лабораторная работа №1
По курсу “Системное программирование”

Пользовательское окружение ОС LINUX и основные команды

Методические указания по выполнению лабораторной работы

Подготовила: Давидовская М.И.,
Ст. преподаватель кафедры ТП

Минск, 2023 г.

Содержание

Цель работы.....	3
Задачи работы.....	3
Краткие теоретические сведения.....	3
Задания.....	4
Методические указания.....	4
Критерии оценивания.....	5
Содержание отчета.....	5
Задание 1. Развертывание виртуальной машины под управлением ОС Ubuntu.....	6
Задание 2. Настройка ssh-соединений между хостовой системой и гостевой машиной.....	7
Задание 3. Использование команды script для протоколирования команд.....	7
Задание 4. Безопасное копирование файлов и публикация в репозиторий.....	8
Задание 5. Управление файлами и папками.....	8
Задание 6. Сжатие и архивация файлов.....	10
Контрольные вопросы.....	11
Приложение 1.....	12
1. Синтаксис основных команд.....	12
2. Управление файлами, каталогами и права доступа.....	19
2.1. Создание и просмотр файлов.....	19
2.2 Создание и просмотр каталогов.....	20
2.3 Копирование и перемещение файлов.....	20
2.4 Работа со ссылками.....	20
2.6 Удаление файлов и каталогов.....	21
3. Архивация файлов и каталогов в ОС Linux.....	21
3.1 Утилиты zip и unzip.....	21
3.2 Утилиты gzip и gunzip.....	22
3.3 Утилита tar.....	23

Цель работы

Ознакомление с операционной системой ОС Unix/Linux и основными командами работы с файлами и каталогами.

Задачи работы

- 1.1. Изучить процедуры входа в систему, выполнения основных команд, получения подсказки, создания и редактирования текстовых файлов.
- 1.2. Изучить основные принципы работы по протоколу ssh.
- 1.3. Изучить основные команды для создания каталогов, файлов, жестких ссылок и символических ссылок, включая архивацию и сжатие данных.

Краткие теоретические сведения

Лабораторная работа выполняется в среде ОС Ubuntu ОС, исполняемой в виртуальной машине под управлением Virtualbox или KVM/QEMU. Доступ к ОС Ubuntu осуществляется с рабочего места, функционирующего в среде ОС Windows или macOS. При выполнении лабораторной работы Вы получаете доступ к учетной записи администратора и создаете учетные записи пользователей для выполнения лабораторной работы.

По умолчанию приглашением в Unix/Linux является символ '#' или '\$', в ответ на которое Вы можете вводить команды. Если в примере приводится команда и в качестве символа приглашения указан символ '#', то это означает, что это пример командной строки для суперпользователя `root`, и команду мы вводим под учетной записью `root`. Если же в примере указан символ '\$', то это означает, что команда вводится по учетной записью обычного пользователя. Левее символа командной строки в квадратных скобках указана Ваша учетная запись и имя устройства с ОС Linux, а также текущий каталог.

В сеансе работы с Linux Вашим текущим (домашним) каталогом является каталог: `/home/имя`, где `имя` — Ваше сетевое имя пользователя. К этому каталогу вам предоставлен полный доступ, т.е. вы имеете права чтения, записи и выполнения. Вы не имеете права записи к каталогам, не являющимся подкаталогами вашего домашнего каталога, если их владельцы (или системный администратор) не дали вам соответствующих прав. Не забывайте, что в Unix/Linux символ "прямая косая черта" (прямой слэш) — разделитель имен каталогов наклонен вправо: `/`!

Для получения справки по командам и программам служат инструкции `man` и `info`, а также другие команды. Некоторые разделы справки даны на английском языке. Краткое описание применяемых в лабораторной работе команд и программ на русском языке представлено в Приложение 1, стр. 12.

Для окончания сеанса работы с командной оболочкой Linux используем команды `exit` или `logout`.

Задания

Методические указания

- 1.1. Вводите команды ОС только при наличии приглашения оболочки.
- 1.2. В используемой оболочке имеется буфер команд (история команд), перемещение по которому производится стрелками управления курсором (вверх и вниз). Для редактирования команды из буфера используются клавиши <- и ->, <BackSpace> и <Delete>. Кроме того для просмотра истории команд используется команда `history`.
- 1.3. Для окончания работы инструкций `man` и `info` используйте символ `q`.
- 1.4. Для протоколирования результатов выполнения команд используйте команду `script`, которая позволяет создать протокол работы пользователя в командной оболочке. Примеры использования команды `script` описаны в руководстве [script_command_guide.pdf](#). Требуется логгировать не только сами команды, но временные метки. Как это сделать, рассмотрено в 2-м примере указанного выше руководства.
- 1.5. Все результаты лабораторной работы должны быть опубликованы в git-репозитории, ссылка на который доступна в курсе «Системное программирование» или же в перечне ниже:
 - Репозиторий группы 12a — <https://classroom.github.com/a/r5tSroXG>
 - Репозиторий группы 126 — <https://classroom.github.com/a/nLYbob4h>
 - Репозиторий группы 13a — <https://classroom.github.com/a/xGylpkID>
 - Репозиторий группы 136 — <https://classroom.github.com/a/gggejuRC>
- 1.6. Возможно открыть новый терминал в пределах той же сессии, используя клавиши <Alt><F2> — <F6>. В виртуальной машине переключение может не работать.
- 1.7. При работе в файловом менеджере `Midnight Commander` для вызова команд можно применять сочетания клавиш с <Ctrl> или <Alt>. Если указано сочетание трех клавиш, то первые две нажимаются одновременно, третья — отдельно. Если символ после служебной клавиши расположен в верхнем регистре, то одновременно нажимаются все три клавиши, включая <Shift>. Если `Midnight Commander` не установлен, установите.
- 1.8. Для доступа к файлам, созданным в виртуальной машине `Ubuntu`, применяйте протокол `ssh`. Для копирования файлов на хостовую машину (ваше окружение с `Windows` или `macOS`) используйте протокол `ftp` или команду `scp`. Протокол `ftp` доступен в менеджере файлов `Far Manager`. При этом учетная запись и пароль для `ftp` те же, что и при подключении по протоколу `ssh`.

Критерии оценивания

Оценка 4

Выполнены задания 1-4. Файлы протоколов команд и меток времени с недочетами. Лабораторная работа сдана с задержкой в 2 недели.

Оценка 5-6

Выполнены задания 1-5. Представлен отчет, файлы протоколов и меток времени в git-репозитории. Отчет, файлы протоколов команд и меток времени могут содержать ошибки. Лабораторная работа сдана с задержкой в 1 неделю.

Оценка 7-8

Выполнены задания 1-6 на хорошем уровне (есть недочеты). Представлен отчет, ответы на контрольные вопросы, файлы протоколов и меток времени в git-репозитории. Отчет, файлы протоколов команд и меток времени могут содержать незначительные ошибки. Лабораторная работа сдана с задержкой в 1 неделю.

Оценка 9

Выполнены задания 1-6 на отличном уровне. Представлен отчет, ответы на контрольные вопросы и файлы протоколов и меток времени в git-репозитории. Отчет, файлы протоколов команд и меток времени не содержат ошибок. Лабораторная работа сдана в срок.

Содержание отчета

1. Цель работы.
2. Вариант задания.
3. Протоколы выполненных действий.
4. Ответы на контрольные вопросы.

Отчет должен быть опубликован в git-репозитории на github. Все результаты лабораторной работы должны быть опубликованы в git-репозитории, ссылка на который доступна в курсе «Системное программирование» или же в перечне ниже:

- Репозиторий группы 12а — <https://classroom.github.com/a/r5tSroXG>
- Репозиторий группы 12б — <https://classroom.github.com/a/nLYbob4h>
- Репозиторий группы 13а — <https://classroom.github.com/a/xGylpkID>
- Репозиторий группы 13б — <https://classroom.github.com/a/gggejuRC>

В файле Readme проекта на github должна быть ссылка на отчёт. Отчет опубликовать во внешнем хранилище или в репозитории в каталоге /docs. Если в лабораторной работе необходимо написать программу/ы, то отчёт должен содержать результаты тестов по каждой программе и ответы на контрольные вопросы.

Пример оформления файла Readme может быть таким:

Overview

Report on LabRabota1.

Usage

// Заменить <<link>> и <<folder>> на соответствующие ссылки/названия

To check, please, preview report by <<link>> and script files in <<folder>>.

Author

Your name and group number.

Additional Notes

// СКОПИРОВАТЬ И ВСТАВИТЬ ССЫЛКУ НА **СВОЙ** РЕПОЗИТОРИЙ, НАПРИМЕР

<https://github.com/maryiad/lab3-task1-gr16-david>

Каждая лабораторная работа содержит тексты задач и контрольные вопросы, ответы на которые проверяются преподавателем при приёме работы у студента.

Выполнение студентом лабораторной работы и сдача её результатов преподавателю происходит следующим образом:

1. Студент выполняет разработку программ.
2. В ходе разработки студент обязан следовать указаниям к данной задаче (в случае их наличия). Исходные тексты программ следует разрабатывать в соответствии с требованиями к оформлению, приведёнными в приложении.
3. Студент выполняет самостоятельную проверку исходного текста каждой разработанной программы и правильности её работы, а также свои знания по теме лабораторной работы.

Задание 1. Развертывание виртуальной машины под управлением ОС Ubuntu

1.1. Изучить материалы учебного курса [Как установить Ubuntu в VirtualBox](#) и [How to install Virt-Manager on Ubuntu 22.04 LTS Linux](#).

1.2. Установить ОС Ubuntu (желательно LTS релиз, например Ubuntu 22.04) или Kali Linux (или любая другая ОС Linux). В среде Windows рекомендуется использовать Virtualbox, в среде Linux — KVM/QEMU + Virt-Manager + Cockpit (<https://www.linuxtechi.com/how-to-install-cockpit-on-ubuntu/>). В процессе установки создать пользователя с именем по следующему шаблону `ФамилияN`, где `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13. Созданный пользователь обладает правами администратора. Продемонстрировать процедуру установки скриншотами в отчете.

Задание 2. Настройка ssh-соединений между хостовой системой и гостевой машиной

2.1. Изучить материалы из учебного курса по настройке ssh в ОС Ubuntu:

- [Как настроить ssh-соединение в гостевой ОС в Virtualbox](#)
- [Как настроить ssh ОС Ubuntu](#)

2.2. Настроить ssh-соединение в гостевой виртуальной ОС Ubuntu. Основные этапы настройки продемонстрировать скриншотами в отчете.

2.3. В зависимости от вашей основной операционной системы изучите рекомендации:

- **Для Windows:** «[Как активировать ssh-клиент в Windows 10](#)» или установите клиент Putty для создания терминальных сессий, используя инструкцию «[Как подключиться по ssh из Windows](#)».
- **Для macOS:** «[Как использовать встроенный ssh-клиент в macOS](#)»

2.5. Открыть ssh-клиент в вашей хостовой системе Windows 10 или macOS. Подключиться из хостовой машины в гостевую виртуальную машины под управлением ОС Ubuntu. Процесс подключения продемонстрировать скриншотами в отчете.

Задание 3. Использование команды script для протоколирования команд

3.1. Изучить руководство по команде script в материале [Руководство по команде script](#) из учебного курса. Обратите внимание на пример команды, которая позволяет протолировать команды и вести журнал меток времени (пример 4).

3.2. Подключиться из хостовой ОС по ssh под пользователем с правами администратора в Ubuntu с именем ФамилияN, где Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13.

3.3. Включить ведение протокола командой script с журналом меток времени. **Протокол** назвать по следующему шаблону — taskXФамилияN, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13. **Журнал меток** назвать по следующему шаблону — timelogXФамилияN, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13.

3.4. Выполните команды help, ls, cd, pwd, mkdir, rmdir, rm, history. Если требуется изучите описания команд с помощью man и info.

3.5. Создайте каталог, задав в качестве имени вашу фамилию латиницей строчными буквами. Данный каталог использовать для размещения результатов лабораторной работы и сохраняйте в него результаты работы.

3.6. Выполнить команды cal, date, who, whoami.

3.7. Создать текстовый файл с произвольным содержанием при помощи команды tee. Просмотреть содержимое файла при помощи команды cat и исправить в нем ошибки при помощи текстового редактора vi.

3.8. Изучить справку на команды cp, unlink.

3.9. Создать копию файла, созданного в п. 3.8, командой `cp`, удалить ее командой `unlink`.

3.13. Завершить протоколирование команд, введя команду `exit`.

Задание 4. Безопасное копирование файлов и публикация в репозиторий

4.1. Установить в ОС Ubuntu `git`, если требуется.

4.2. Склонируйте репозиторий из `github`.

4.3. Скопируйте в каталог репозитория результаты работы в Ubuntu, полученные в ходе выполнения задания 3.

4.4. Скопируйте с помощью `scp` из хостовой системы в гостевую систему любой файл. Копирование выполнить на стороне гостевой системы.

4.5. Завершите протоколирование команд с помощью `script`, т. е. введите команду `exit`.

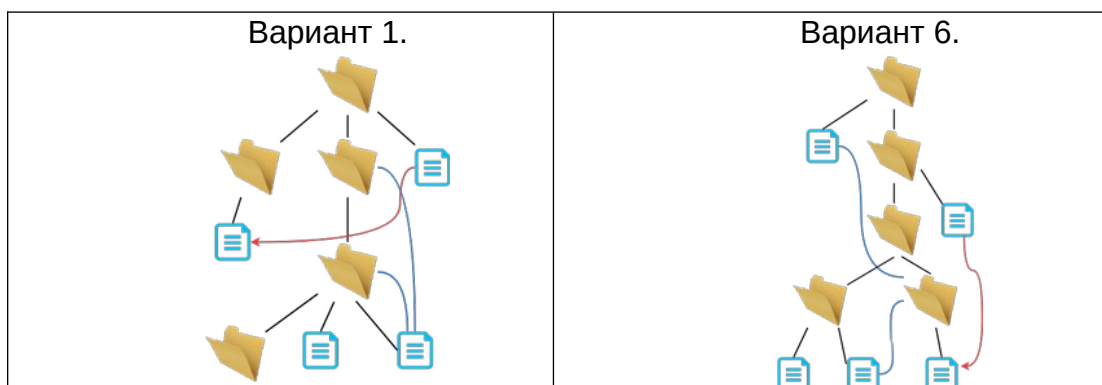
4.6. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения заданий 3 и 4, в удаленный.

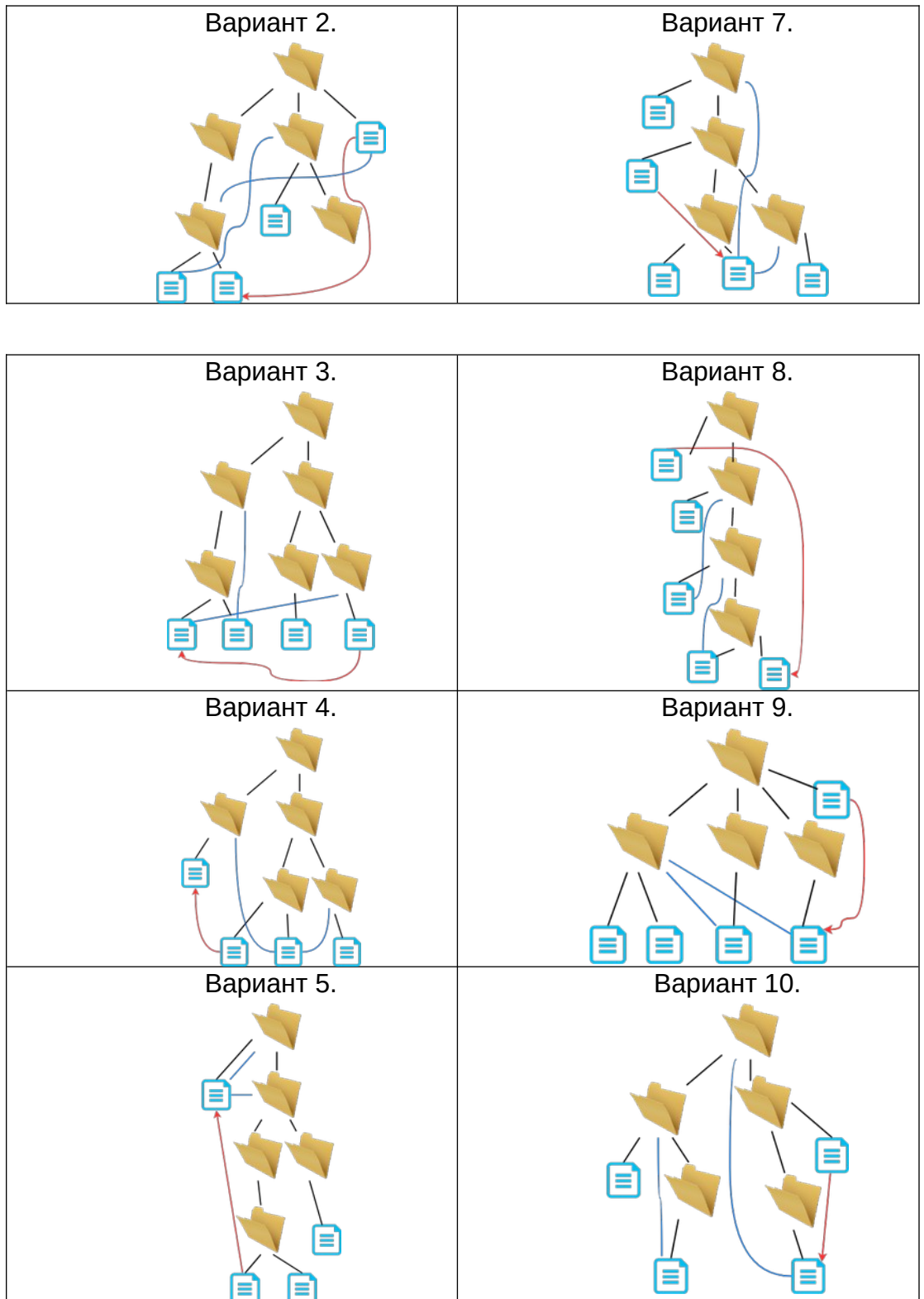
Задание 5. Управление файлами и папками

5.1. Задание выполнить в ОС Ubuntu. Вести протокол командой `script` с журналом меток времени. **Протокол** назвать по следующему шаблону — `taskXФамилияNK`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `K` — номер варианта. **Журнал меток** назвать по следующему шаблону — `timelogXФамилияNK`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13, `K` — номер варианта.

5.2. Создайте структуру каталогов в соответствии с вариантом в ОС Ubuntu. Черными линиями представлена вложенность файлов/подкаталогов в каталоги. Синими линиями представлены жесткие ссылки. Красными линиями — символические ссылки. Стрелка на красной линии указывает на целевой файл ссылки. Файлы создаются копированием ранее созданного файла командой `cp` с внесением в копии некоторых изменений. Ссылки создаются командой `ln`, символические ссылки — ей же, но с ключом `-s`:

Таблица 1.





5.3. Для всех вариантов выполнить следующие действия:

- Создать ссылки (синие линии).
- Создать символические ссылки (красные линии).
- Провести ряд экспериментов, иллюстрирующих доступ к файлам по основным именам, по ссылкам и по символическим ссылкам. Для доступа использовать команду `cat` или редактор `vi`.
- Провести ряд экспериментов, иллюстрирующих реакцию системы на удаление файла, на который имеются ссылки, и файла, на который имеются символические ссылки. Проверять результаты командой `ls -la`.

- Уничтожить созданные подкаталоги и файлы в них, используя команды `rmdir` и `unlink`, сохранив, однако, файл, созданный в пункте 5.2 и одну его рабочую копию в домашнем каталоге.

5.4. Завершите протоколирование команд с помощью `script`, т. е. введите команду `exit`.

5.5. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 5, в удаленный.

Задание 6. Сжатие и архивация файлов

6.1. Задание выполнить в ОС Ubuntu. Вести протокол командой `script` с журналом меток времени. **Протокол** назвать по следующему шаблону — `taskXФамилияN`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13. **Журнал меток** назвать по следующему шаблону — `timelogXФамилияN`, где `X` — номер выполняемого задания, `Фамилия` — заменить на вашу фамилию латиницей и строчными буквами, `N` — номер группы, например 12 или 13.

Утилиты `gzip` и `gunzip`

6.2. Подключиться по `ssh` со своей учетной записью к ОС Ubuntu или авторизоваться в виртуальной машине. Используя программу `gzip`, сжать файл любой текстовый файл в созданном каталоге согласно варианту из задания 5.

6.3. Включить запись протокола команд и меток времени командой `script`.

6.4. Используя программу `gzip`, сжать файл любой текстовый файл в созданном каталоге согласно варианту из задания 5, чтобы исходный файл остался в текущем каталоге. Продемонстрировать два примера команд, выполняющих сжатие с сохранением исходного файла.

6.5. Проверить целостность сжатого файла.

6.6. Завершить протоколирование команд.

Утилита `tar`

6.7. Включить запись протокола команд и меток времени командой `script`.

6.8. Создать в домашнем каталоге архив формата `tar`, включающий каталог и файлы.

6.9. Сжать полученный архив.

6.10. Проверить целостность архива и сжатого архива.

6.11. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 6, в удаленный.

Контрольные вопросы.

1. Какие ключи команды `ls` Вы знаете? Какой ключ используется для рекурсивного обхода каталогов?
2. Чем отличаются `man` и `info`? Как с ними работать? Какая команда позволяет найти документацию по ключевому слову в каждом случае?
3. Команда `script` — назначение и применение.
4. Какая команда используется для генерации `ssh`-ключа?
5. Для каких задач применяется команда `scp`?
6. Команды `tee` и `cat`. Назначение и применение. Чем `cat` отличается от `more` и `less`?
7. Основные команды редактора `vi`.
8. Жесткие ссылки и символические ссылки. Создание и применение.
9. Создание и копирование файлов и папок в `Linux`.
10. Перемещение и удаление файлов и папок в `Linux`.
11. Как вы считаете, почему в `Linux` разделены команды для сжатия и команды архивирования? Приведите их примеры.
12. Какой алгоритм сжатия использует `gzip`?
13. В каком году был создан архиватор `xz`? Для каких задач применяется? Какой алгоритм сжатия использует?
14. Какой архиватор, созданный в 2015 году, поддерживает 22 уровня сжатия.

Приложение 1

1. Синтаксис основных команд

ls — вывод содержимого каталога. **Синтаксис:** `ls [опции] [файл...]`

Описание: Команда `ls` сначала выводит список всех файлов (не каталогов), перечисленных в командной строке, а затем выводит список всех файлов, находящихся в каталогах, перечисленных в командной строке. Если не указано ни одного файла, то по умолчанию аргументом назначается '.' (текущий каталог). Опция `-d` заставляет `ls` не считать аргументы-каталоги каталогами. Будут отображаться только файлы, которые не начинаются с '.' или все файлы, если задана опция `-a`.

Результаты печатаются на стандартный вывод, по одному файлу на строку, если с помощью опции `-C` не задан многоколоночный вывод.

Каждый список файлов (для файлов, которые не являются каталогами и для каждого каталога, содержащего список файлов) сортируется отдельно в алфавитной последовательности.

Опции:

`-l` В дополнение к имени каждого файла, выводятся тип файла, права доступа к файлу, количество ссылок на файл, имя владельца, имя группы, размер файла в байтах и временной штамп (время последней модификации файла, если не задано другое).

`-a` Выдавать все файлы в каталогах, включая все файлы и подкаталоги, имена которых начинаются с '.'.

`-d` Выдавать имена каталогов, как будто они обычные файлы, вместо того, чтобы показывать их содержимое.

`-L` Выдавать информацию о файлах, на которые указывают символические ссылки, вместо информации о самих символических ссылках.

`-R` Рекурсивно выдавать список содержимого всех каталогов.

`-h` Добавлять к каждому размеру файла букву размера, например, М (мегабайт).

`-X` Производить сортировку в алфавитном порядке по расширениям файлов (символы после последней '.'); файлы без расширений будут показаны первыми.

`-S` Производить сортировку по размеру файла, вместо сортировки по алфавиту. Таким образом, наибольшие файлы будут показаны сначала.

`-c` Сортировать содержимое каталога в соответствии с временем изменения состояния файла. Если с помощью опции `-l` задан длинный формат, то выдавать время изменения состояния файла вместо времени его модификации.

`-t` Сортировать по времени последней модификации вместо того, чтобы производить сортировку по алфавиту. Самые свежие файлы будут отображаться первыми.

`-u` Сортировать по времени последнего доступа к файлу, вместо времени последней модификации.

Типы файлов (первая буква в строке при задании опции `-l`) могут принимать следующие значения: — для обычного файла, `d` для каталога, `b` для блочного устройства, `c` для символьного устройства, `l` для символической ссылки, `r` для FIFO и `s` для гнезда (socket)

Права доступа составляют 9 символов и делятся на три группы по три символа: права доступа владельца, других пользователей из его группы, всех прочих пользователей. Права обозначаются следующим образом:

`r` Право на чтение.

`w` Право на запись.

`x` Право на выполнение (поиск в каталоге).

– Данное право доступа отсутствует.

Для каталога под правом на выполнение подразумевается право на просмотр в поисках требуемого файла.

cd — смена текущего каталога. **Синтаксис:** cd [каталог]

Описание: cd изменяет текущий каталог на каталог. Имя каталог может задаваться абсолютным (от корневого каталога) — в этом случае оно начинается с символа '/' — или относительным (от текущего каталога) — в этом случае оно начинается с символов './' или '../'. Если каталог не указан, текущим становится "домашний" каталог пользователя, определяемый значением переменной окружения \$HOME.

pwd — выдача имени текущего каталога. **Синтаксис:** pwd

Описание: Команда pwd выдает имя текущего (рабочего) каталога.

mkdir — создание каталога. **Синтаксис:** mkdir [опции] каталог...

Описание: Команда mkdir создает каталоги с заданными именами.

По умолчанию права доступа к каталогам устанавливаются в 0777 ('a+rwX').

Опции:

-m права Устанавливает права доступа к создаваемым каталогам. Эти права могут быть заданы либо в символьном виде, либо в виде восьмеричного числа, как описано в

rmdir — удаление пустых каталогов. **Синтаксис** rmdir [опции] каталог...

Описание. Команда rmdir удаляет пустые каталоги. Если какой-либо из аргументов каталог не указывает на существующий пустой каталог, то будет выдано сообщение об ошибке.

Опции:

-r Если каталог включает более, чем один компонент пути, то удаляется каталог, затем убирается последний компонент пути и удаляется получившийся каталог и т.д. до тех пор, пока все компоненты не будут удалены. Таким образом, команда rmdir -r a/b/c эквивалентна rmdir a/b/c; rmdir a/b; rmdir a.

man — форматирование и отображение онлайн-справочных страниц.

Синтаксис: man [раздел] имя...

Описание: Команда man выполняет форматирование и отображение онлайн-справочных страниц Unix. Если задан раздел, то man ищет только в заданном разделе руководства. имя — это обычно имя страницы руководства, которое, как правило, является именем команды, функции или файла.

Имеющиеся справочные страницы разбиты на несколько разделов. Важнейшими являются разделы:

1 — команды Unix;

2, 3 — системные вызовы Unix;

4, 5 — форматы файлов Unix.

Когда команда отображает страницу подсказки, в нижней строке экрана выводится приглашение man — символ ':'. После приглашения можно вводить внутренние команды man. В кратком руководстве следует упомянуть только две внутренние команды man:

h получение подробной информации о внутренних командах man;

q выход из man или переход к следующей странице, если команда man была введена с указанием нескольких имен команд.

Двигаться по отображаемой странице можно при помощи клавиш управления курсором.

Для получения более подробной информации о команде man введите: man man

info — отображение онлайн-справочных страниц. **Синтаксис:** info имя...

Описание: Команда info выполняет форматирование и отображение онлайн-справочных страниц Linux.

В текстах, отображаемых командой, могут быть наборы строк, озаглавленные "*" Menu", каждая строка такого набора начинается с символа "*". Выбрав курсором пункт меню и нажав клавишу Enter, можно получить страницу подсказки по этому пункту.

Независимо от положения курсора, после приглашения можно вводить внутренние команды info. В кратком руководстве следует упомянуть только две внутренние команды info:

h получение подробной информации о внутренних командах info;

q выход из info или переход к следующей странице, если команда info была введена с указанием нескольких имен команд.

Двигаться по отображаемой странице можно при помощи клавиш управления курсором.

script — протоколирование сеанса. **Синтаксис:** script [-a] файл

Описание: Команда script начинает "вложенный" сеанс и протоколирует весь терминальный ввод и вывод в заданном файле. Завершение вложенного сеанса и выполнения команды script происходит по нажатию комбинации клавиш Ctrl+D.

Опции:

-a добавление протокола нового сеанса к содержимому файла, если эта опция не задана, то файл создается заново.

who — кто в системе? **Синтаксис:** who [опции]

Описание: Команда who сообщает имя пользователя, имя терминальной линии, астрономическое время начала сеанса, продолжительность бездействия терминальной линии с момента последнего обмена, идентификатор процесса для каждого из пользователей, работающих в системе.

Сообщения команды who имеют следующий формат:

NAME STATE LINE TIME IDLE PID COMMENT

где NAME — входное имя пользователя; LINE — имя терминальной линии, под которым она фигурирует в каталоге /dev; TIME — время начала сеанса; IDLE — время (часы и минуты), протекшее с последнего момента активизации данной линии. Точка (.) свидетельствует о том, что это действующий терминал. PID — идентификатор процесса интерпретатора shell, обслуживающего данного пользователя; COMMENT — комментарий, характеризующий данную линию.

Опции:

-H отображение заголовков столбцов в выводимой информации

-i отображается поле IDLE

-q отображение только имен и количества пользователей, работающих в системе в данный момент; все прочие опции при этом игнорируются

-T аналогично -s, но при этом отображается также поле STATE, как:

+ терминал, на который можно передавать сообщения

- терминал, на который нельзя передавать сообщения

? терминал неисправен

-s выводятся только поля NAME, LINE и TIME; это опция по умолчанию.

write — передача сообщения другому пользователю. **Синтаксис:** write *адресат*

Описание: *Адресат* задается как сетевое имя пользователя. После запуска команда write устанавливает связь с адресатом и переходит в режим ожидания ввода. В момент установки связи на терминал адресата выводится сообщение:

Message from *отправитель* ...

Отправитель вводит любой текст, который отображается на терминале получателя.

Отправитель заканчивает сообщение нажатием комбинации клавиш Ctrl+D в начале строки. У адресата окончание сообщения индицируется строкой: EOF

Получатель может заблокировать/разблокировать вывод сообщений на свой экран при помощи команды `mesg`. При попытке передать сообщение на заблокированный терминал отправитель получает диагностику:
`write: адресат has messages disabled`

tee — ответвление канала. **Синтаксис:** `tee [опции]... [файл]...`

Описание: Команда `tee` переписывает стандартный ввод на стандартный вывод и делает копии в файлах. Признаком окончания ввода является комбинация клавиш `Ctrl+D`.

Опции:

-а добавлять выводимую информацию в файлы, а не переписывать их с начала.

cat — слияние и вывод файлов. **Синтаксис:** `cat [-опции] файл ...`

Описание: Команда `cat` по очереди читает указанные файлы и выдает их содержимое на стандартный вывод. Так, например, `cat f` распечатывает содержимое файла `f`, а `cat f1 f2 > f3` сливает первые два файла и помещает результат в третий. Чтобы добавить файл `f1` к файлу `f2`, надо выполнить команду `cat f1 >> f2`. Если не указан ни один файл или среди аргументов встретился `-`, команда `cat` читает данные со стандартного ввода.

Опции:

-b Нумеруются непустые строки файла.

-s Нумеруются все строки файла. (Поле номера отделяется от текста символом табуляции).

-v Визуализация непечатных символов. Управляющие символы изображаются в виде `^X` (`CTRL+X`); символ `DEL` (восьмеричное `0177`) — в виде `^?`. Символы, не входящие в набор `ASCII` (то есть с восьмым битом, установленным в 1) выдаются в виде `M-x`, где `x` — определяемый младшими семью битами символ.

С опцией `-v` можно использовать следующие опции:

-t Визуализация символов табуляции в виде `^I`.

-e Визуализация символов перевода строки в виде `$` (строка при этом все же переводится).

Если опция `-v` не указана, то опции `-t` и `-e` игнорируются.

vi — текстовый редактор. **Синтаксис:** `vi имя_файла`

Описание: Редактор `vi` имеет три режима:

1. Командный — в этом режиме можно перемещаться по файлу и выполнять редактирующие команды над текстом. Команды вызываются ОБЫЧНЫМИ ЛАТИНСКИМИ БУКВАМИ.
2. Ввода текста — в этом режиме обычные латинские буквы будут вставляться в текст.
3. Режим строчного редактора `vi` используется для управления файлами (типа сохранить файл, зачитать файл и т.д.)

VI в КОМАНДНОМ РЕЖИМЕ.

ЧТОБЫ ВЫЙТИ ИЗ ФАЙЛА БЕЗ СОХРАНЕНИЯ, нажмите:

`ESC : q ! Enter`

чтобы выйти из файла, сохранив изменения, нажмите:

`ESC : w ! Enter`

`ESC : q Enter`

выйти из файла с сохранением, одной командой:

`ESC : wq Enter`

для перехода В РЕЖИМ ВВОДА нужно нажать команды типа:

`i` вставлять здесь

`A` вставлять с конца строки

заданного каталога, включая и заданный каталог (без ограничения на глубину дерева).

Опции:

-f Игнорировать несуществующие файлы и никогда не запрашивать подтверждение на удаления.

-i Выдавать запрос на удаление каждого файла. (Принята по умолчанию).

-r Рекурсивно удалять содержимое каталогов.

-v Выдавать имя каждого файла перед его удалением.

ln — создание ссылки на файл. **Синтаксис:** ln [-f] файл1 [файл2 ...] целевой_файл

Описание: Команда ln делает целевой_файл ссылкой на файл1. Файл1 не должен совпадать с целевым_файлом. Если целевой_файл является каталогом, то в нем создаются ссылки на файл1, файл2,... с теми же именами. Только в этом случае можно указывать несколько исходных файлов.

Если целевой_файл существует и не является каталогом, его старое содержимое теряется.

Опции:

-f удаление существующего целевого файла

-s создание символической ссылки (по умолчанию создается жесткая ссылка)

chmod — изменение режима доступа к файлам. **Синтаксис:** chmod режим файл...

Описание: Команда chmod изменяет права доступа к указанным файлам (среди которых могут быть каталоги) в соответствии с указанным режимом. Режим может быть задан в абсолютном или символьном виде.

Абсолютный вид — восьмеричное число, являющееся поразрядным ИЛИ следующих режимов (названы не все режимы):

00400 Доступен для чтения владельцем.

00200 Доступен для записи владельцем.

00100 Доступен для выполнения (в случае каталога — для просмотра) владельцем.

00040 Доступен для чтения членами группы.

00020 Доступен для записи членами группы.

00010 Доступен для выполнения (просмотра) членами группы.

00004 Доступен для чтения прочими пользователями.

00002 Доступен для записи прочими пользователями.

00001 Доступен для выполнения (просмотра) прочими пользователями.

Символьный вид основан на однобуквенных обозначениях, которые определяют класс доступа и права доступа для членов данного класса. Права доступа к файлу зависят от идентификатора пользователя и идентификатора группы, в которую он входит. Режим в целом описывается в терминах трех последовательностей, по три буквы в каждой:

Владелец	Группа	Прочие
----------	--------	--------

(u)	(g)	(o)
-----	-----	-----

rwX	rwX	rwX
-----	-----	-----

Для задания режима доступа в символьном виде используется синтаксис:

[кому] операция права

Часть кому есть комбинация букв u, g и o (владелец, члены группы и прочие пользователи соответственно). Если часть кому опущена или указано a, то это эквивалентно ugo.

Операция может быть: + (добавить право), — (лишить права), = (в пределах данного класса присвоить права абсолютно, то есть добавить указанные права и отнять неуказанные).

Права — любая осмысленная комбинация следующих букв (не все):

r Право на чтение.

w Право на запись.

x Право на выполнение (поиск в каталоге).

Опустить часть права можно только если операция есть = (для лишения всех прав).

Если надо сделать более одного указания об изменении прав, то при использовании символьного вида в правах не должно быть пробелов, а указания должны разделяться запятыми. Например, команда

```
chmod u+w,go+x f1
```

добавит для владельца право писать в файл f1, а для членов группы и прочих пользователей — право выполнять файл. Права устанавливаются в указанном порядке.

Изменить режим доступа к файлу может только его владелец или суперпользователь.

Для просмотра прав доступа и контроля при их изменении используется команда ls с флагом -l.

chown — изменение владельца и группы файлов.

Синтаксис: chown [опции] пользователь[:группа] файл...

Описание: Команда chown изменяет владельца и/или группу для каждого заданного файла. В качестве имени владельца/группы берется первый аргумент, не являющийся опцией. Если задано только имя пользователя (или числовой идентификатор пользователя), то данный пользователь становится владельцем каждого из указанных файлов, а группа этих файлов не изменяется. Если за именем пользователя через двоеточие следует имя группы (или числовой идентификатор группы), без пробелов между ними, то изменяется также и группа файла.

Опции:

-R Рекурсивное изменение владельца для каталогов и их содержимого.

ps — вывод информации о состоянии процессов. **Синтаксис:** ps [опции]

Описание: Команда ps выводит в стандартный вывод информацию о текущем состоянии процессов.

Опции:

-a все процессы, кроме лидеров групп и процессов, не ассоциированных с терминалом.

-d все процессы, кроме лидеров групп.

-e все процессы.

-g*список* выбирать процессы по *списку* лидеров групп.

-r*список* выбирать процессы по *списку* идентификаторов процессов.

-t*список* выбирать процессы по *списку* терминалов.

-u*список* выбирать процессы по *списку* идентификаторов пользователей.

-f генерировать полный листинг.

-l генерировать листинг в длинном формате.

Результат команды ps:

Ниже приводятся заголовки колонок выдачи команды ps и объясняется смысл их содержимого. Буквы l или f означают, что эта колонка появляется соответственно при длинном или полном формате выдачи; отсутствие букв означает, что данная колонка выводится всегда.

F **l** **Флаги** (шестнадцатеричные), логическая сумма которых дает следующие сведения о процессе:

00 процесс терминирован; элемент таблицы процессов свободен;

01 системный процесс: всегда в основной памяти;

02 процесс трассируется родительским процессом;

04 родительский трассировочный сигнал остановил процесс; родительский процесс ждет;

08 процесс не может быть разбужен сигналом;

10 процесс в основной памяти;

20 процесс в основной памяти; блокирован до завершения события;

40 идет сигнал к удаленной системе;
 80 процесс в очереди на ввод/вывод.
 S I статус процесса:
 O активный: обрабатывается процессором;
 S спящий: ожидает завершения события;
 R готов: стоит в очереди на выполнение;
 I рождающийся: процесс создается;
 Z состояние "зомби": процесс завершен, но родительский процесс не ждет этого;
 T трассируемый: процесс остановлен сигналом, так как родительский процесс трассирует его;
 X растущий: процесс ожидает получения большего объема основной памяти.
 UID f,l идентификатор владельца процесса; при указании опции -f выдается входное имя пользователя.
 PID идентификатор процесса .
 PPID f,l идентификатор родительского процесса.
 C f,l доля выделенного планировщиком времени ЦП.
 STIME f время запуска процесса (часы:минуты:секунды); если процесс запущен более чем 24 часа назад, выдается месяц и день запуска.
 PRI I приоритет процесса.
 NI I поправка к приоритету.
 ADDR I адрес процесса в памяти.
 SZ I размер (в блоках по 512 байт) образа процесса в памяти.
 WCHAN I адрес события, которого ожидает процесс.
 TTY I управляющий терминал.
 TIME I использованное процессом время.
 COMMAND I имя программы; если указана опция -f, то выводится полное имя команды и ее аргументы.

2. Управление файлами, каталогами и права доступа

2.1. Создание и просмотр файлов

1. Входим в систему по имени student и создаем в текущем

каталоге пустой файл с именем report.txt:

```
$ touch report.txt
```

2. Открываем файл текстовым редактором vim или nano и набираем в нем произвольный текст

```
$ vim report.txt
```

3. Выводим содержимое файла report.txt на консоль

```
$ cat report.txt
```

4. Выводим на консоль первых десять строк файла report.txt

```
$ head report.txt
```

5. Выводим на консоль первых две строки файла report.txt

```
$ head -n 2 report.txt
```

6. Выводим на консоль последних десять строк файла report.txt

```
$ tail report.txt
```

7. Выводим на консоль последних две строки файла report.txt

```
$ tail -n 2 report.txt
```

2.2 Создание и просмотр каталогов

1. Создаем в текущем каталоге новый каталог с именем documents:

```
$ mkdir documents
```

2. Просматриваем содержимое текущего каталога

```
$ ls -l
```

3. Просматриваем содержимое текущего каталога, включая скрытые файлы

```
$ ls -a
```

4. Просматриваем файлы текущего каталога в порядке возрастания времени их последней модификации:

```
$ ls -ltr
```

2.3 Копирование и перемещение файлов

1. Копируем файл report.txt из текущего каталога в подкаталог documents текущего каталога:

```
$ cp report.txt documents
```

2. Переименовываем файл report.txt в файл report.bak

```
$ mv report.txt report.bak
```

3. Перемещаем файл report.bak в каталог documents

```
$ mv report.bak documents
```

4. Просматриваем содержимое каталога documents

```
$ ls -l documents
```

2.4 Работа со ссылками

1. Создаем в текущем каталоге жесткую ссылку на файл report.txt, который находится в каталоге documents:

```
$ ln ~/documents/report.txt
```

2. Создаем в текущем каталоге жесткую ссылку с именем rep.txt на файл report.txt, который находится в каталоге documents:

```
$ ln ~/documents/report.txt rep.txt
```

3. Создаем в текущем каталоге символическую ссылку на файл report.bak, который находится в каталоге documents:

```
$ ln -s ~/documents/report.bak
```

4. Создаем в текущем каталоге мягкую ссылку doc на каталог documents:

```
$ ln -s ~/documents/ doc
```

5. Просматриваем содержимое мягкой ссылки doc:

```
$ readlink doc
```

6. Открываем в текстовом редакторе vim файл report.txt через жесткую ссылку, созданную в текущем каталоге:

```
$ vim report.txt
```

Вносим изменения в файл и закрываем его.

7. Открываем в текстовом редакторе vim файл rep.txt через мягкую ссылку на каталог documents, созданную в текущем каталоге:

```
$ vim ~/doc/report.txt
```

и просматриваем изменения в этом файле.

8. Удаляем созданные мягкие и жесткие ссылки:

```
$ unlink report.txt
```

```
$ unlink report.back
```

9. Просматриваем содержимое каталога documents через ссылку:

```
$ ln -l doc
```

10 Удаляем созданную мягкую ссылку на каталог documents:

```
$ unlink doc
```

11. Создаем в текущем каталоге файл message.txt.

```
$ nano message.txt
```

И добавьте 5 строк с любым текстом.

12. Создаем новый файл и добавьте в него две любых строки:

```
$ touch testmask.txt
```

```
$ ls -l
```

2.6 Удаление файлов и каталогов

1. Удаляем файлы message.txt и testmask.txt:

```
# rm message.txt testmask.txt
```

2. Входим в систему под именем student и удаляем файл report.txt из каталога documents:

```
$ rm rep.txt
```

3. Удаляем файл report.bak из каталога documents с запросом на подтверждение удаления:

```
$ rm -i ~/documents/report.bak
```

Проверяем, что каталог documents пустой.

4. Удаляем пустой каталог documents из текущего каталога:

```
$ rmdir documents
```

3. Архивация файлов и каталогов в ОС Linux

3.1 Утилиты zip и unzip

1. Создаем в текущем каталоге файлы report.txt и message.txt

```
$ touch report.txt
```

```
$ touch message.txt
```

С помощью текстового редактора nano вводим в созданные файлы произвольный текст

```
$ nano report.txt
```

```
$ nano message.txt
```

2. Создаем zip архив arch1 с файлами report.txt и message.txt

```
$ zip arch1 report.txt message.txt
```

3. Создаем каталог documents и перемещаем в него файлы report.txt и message.txt

```
$ mkdir documents
```

```
$ mv report.txt message.txt documents
```

4. Сжимаем каталог documents в архив arch2

```
$ zip arch2 documents
```

5. Создаем в текущем каталоге файл temp.txt и вводим в него произвольный текст.

```
$ touch temp.txt
```

```
$ nano temp.txt
```

6. Добавляем файл temp.txt в архивы arch1 и arch2

```
$ zip arch1 temp.txt
```

```
$ zip arch2 temp.txt
```

7. Создаем zip архив arch3, в который помещаем все файлы из текущего каталога

```
$ zip arch3 *
```

8. Просматриваем содержимое архивов arch1 и arch2

```
$ unzip -l arch1
```

```
$ unzip -l arch2
```

9. Проверяем целостность архива arch3

```
$ unzip -t arch3
```

10. Удаляем текстовые файлы и каталог documents из текущего каталога

```
$ rm *.txt
```

```
$ rm -r documents
```

11. Распаковываем архив arch1 и arch2 в текущий каталог

```
$ unzip arch1
```

```
$ unzip arch2
```

12. Проверяем содержимое текущего каталога

```
$ ls -l
```

3.2 Утилиты gzip и gunzip

1. Сжимаем файл report.txt в текущем каталоге

```
$ gzip report.txt
```

Просматриваем текущий каталог

```
$ ls -l
```

Как видим, команда `gzip` удалила исходный файл `report.txt` из текущего каталога.

2. Сжимаем файл `temp.txt` в текущем каталоге

```
$ gzip -c temp.txt > temp.txt.gz
```

Просматриваем текущий каталог

```
$ ls -l
```

Как видим, в этом случае в текущем каталоге сохраняется файл `temp.txt`.

3. Проверяем целостность сжатого файла `report.txt.gz`

```
$ gunzip -t report.txt.gz
```

4. Распаковываем файл `report.txt.gz` в текущий каталог

```
$ gunzip report.txt.gz
```

Просматриваем текущий каталог

```
$ ls -l
```

Как видим, команда `gunzip` удалила исходный сжатый файл `report.txt.gz` из текущего каталога.

5. Распаковываем файл `temp.txt.gz` в текущий каталог

```
$ gunzip -c temp.txt.gz > test.txt
```

Просматриваем текущий каталог

```
$ ls -l
```

Как видим, команда исходный сжатый файл `test.txt.gz` остался в текущем каталоге.

3.3 Утилита `tar`

1. Создаем в текущем каталоге `tar` архив `atar1`, включающий файлы `report.txt`, `message.txt` и `temp.txt`, сжатых программой `gzip`

```
$ tar -czf atar1.tar.gz report.txt message.txt temp.txt
```

2. Проверяем архив `atar1`, сжатый программой `gzip`

```
$ tar -tvzf atar1.tar.gz
```

3. Распаковываем в текущий каталог архив `atar1`, сжатый программой `gzip`

```
$ tar -xvzf atar1.tar.gz
```