**Методические указания к выполнению практических работ**

При выполнении практических работ необходимо выполнить все задания, указанные в них. Выполнение каждого из них оценивается в баллах. Результат выполнения каждого из заданий необходимо продемонстрировать преподавателю. В случае, если это невозможно – то необходимо оформить отчет о выполнении практического задания.

Требования к оформлению отчетов доступны на странице [[ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ]](https://sp.susu.ru/student/graduate/CourseWork_content_paper.html) сайта кафедры Системного программирования в разделе «Студентам -> Курсовые работы и проекты -> Требования к содержанию и оформлению курсовой работы и проекта».

В состав отчета должны быть включены:

- описание хода выполнения каждого задания со скриншотами выполнения;

- пояснения к каждому действию, выполняемому в ходе задания;

- ответы на контрольные вопросы.

При выполнении практических работ студент должен изучить контрольные вопросы, так как эти вопросы будут на промежуточной аттестации во дисциплине.

**Примечание!** С каждым последующим номером, практические работы будут усложняться, а описания действий будут становиться менее информативными (то есть без описания применения тех или иных команд). Научитесь искать информацию о командах и возможностях Linux самостоятельно!

**Практическая работа №1: Сеанс работы в Linux. Терминал и командная строка.**

**Задание 1.** Используя команду **passwd**, измените пароль для текущего (или дополнительно созданного) пользователя.

**Задание 2.** Примените команду **whoami**, чтобы определить название учетной записи пользователя, от имени которого выполняется данная команда.

**Задание 3.** Переключитесь на другую виртуальную консоль. «*Залогиньтесь*» под своим именем учетной записи. Примените команду **who** и определите, сколько пользователей зарегистрированы в системе (вошли в систему) в настоящий момент времени.

**Задание 4.** Завершите работу второго сеанса виртуальной консоли, используя команду **logout**. Переключитесь на основную виртуальную консоль. Используйте команду **last** для просмотра списка пользователей, работавших в системе в последнее время. Если у вас возникли сложности с просмотром всего списка (например, вы не можете «листать» вывод команды), то обратитесь за помощью к поисковым системам (например, google.com) или задайте вопрос преподавателю.

**Задание 5.** Используйте команду **date** для вывода текущего времени и даты. Примените эту же команду с ключом ***–u***. Сравните результат.

**Задание 6**. Используя команду **cal** выведите в консоль календарь на текущий месяц. Примените команду **cal** с параметрами ***номер\_месяца*** и ***год***. Например, **cal *3 2019***. Примените команду **cal** с ключом ***–m***. Сравните вывод этой командной конструкции с выводом команды **cal** без ключа.

**Задание 7.** Примените команду **man** ***объект*** для просмотра руководства по интересующей пользователя программе. Например, мануал для команды cal: **man** ***cal***.

**Задание 8**. По очереди примените команды **whatis** и **apropos**. Сравните результаты и опишите преподавателю разницу между этими командами.

*Список контрольных вопросов*

1) Что такое UID?

2) Что такое GID?

3) Что такое домашний каталог?

4) Что такое командная оболочка (интерпретатор командной строки)?

5) Какая учетная запись обязательно присутствует в любой операционной системе Linux?

6) Для чего применяют команду **passwd**?

7) Для чего применяют команду **whoami**?

8) Каким способом можно переключаться между виртуальными консолями в Linux? Как обозначаются виртуальные консоли?

9) Кто такие «псевдопользователи»?

10) Для чего применяют команду **last**?

11) При помощи какой команды можно завершить сеанс работы в виртуальной консоли для зарегистрированного в системе пользователя?

12) Что является основным средством общения с Linux? В каком режиме это общение происходит?

13) Что такое терминал?

14) Какую команду необходимо использовать для определения текущего времени и даты? Какой ключ необходимо применить совместно с этой командой для вывода времени по Гринвичу?

15) Какая команда применяется для вывода календаря на текущий месяц? Как вывести календарь на интересующий пользователя месяц? Как выполнить просмотр календаря так, чтобы первым днем недели являлся понедельник, как это принято в России?

16) Как посмотреть руководство с информацией о той или иной команде в Linux?

17) Для чего применяют команды **whatis** и **apropos**? Какая между ними разница?

18) Что такое *shell*?

**Практическая работа №2: Структура файловой системы. Работа с файловой системой.**

**Задание 1.** Создайте текстовый файл при помощи команды **touch** ***имя\_файла***:

$ **touch *example1.txt***

Направьте вывод команды echo текст в созданный файл:

$ **echo *‘Hello, World!’ > example1.txt***

Определите, какие данные содержатся в созданном файле:

$ **file *example1.txt***

Просмотрите содержимое созданного файла:

$ **cat *example1.txt***

Кроме созданного файла, выведите описание вашего рабочего каталога:

$ **file */home/имя\_пользователя***

**Задание 2.** Используйте команду **ls *каталог*** (здесь ***каталог*** – полный путь к каталогу), для просмотра его содержимого (корневого каталога):

$ **ls */***

Просмотрите содержимое вашего рабочего каталога:

$ **ls**

**Задание 3.** Определите текущий каталог при помощи команды **pwd**.

Выведите содержимое корневого каталога с ключом ***–F*** и без него. Сравните выводы команд:

$ **ls *–F /***

$ **ls */***

Просмотрите содержимое вашего текущего каталога без ключей и с ключом ***–a***, при этом не забывайте применять ключ ***–F***:

$ **ls -*F***

$ **ls *-aF***

**Задание 4.** Создайте новый каталог в вашем домашнем каталоге при помощи команды **mkdir**:

$ **mkdir *examples***

Просмотрите содержимое домашнего каталога при помощи команды **ls *–F*** и убедитесь, что каталог *examples* был создан.

Внутри каталога examples создайте еще один, с именем ex1:

$ **mkdir *examples/ex1***

Создайте еще один каталог в examples, но используя при этом ключ –p, а в качестве параметра команды укажите путь к создаваемому каталогу:

$ **mkdir *–p examples/ex2/ex3***

Проверьте содержимое каталога examples, а также содержимое вложенных в него каталогов:

$ **ls *–F examples***

$ **ls *–F examples/ex1***

$ **ls *–F examples/ex2***

$ **ls *–F examples/ex2/ex3***

**Задание 5.** Используя команду **cd** переместитесь в созданную ранее папку *ex3*:

$ **cd** ***examples/ex2/ex3***

Определите полный путь к каталогу, в котором вы сейчас находитесь при помощи команды **pwd**.

Переместитесь на уровень выше, используя команду **cd** и дополнительное обозначение родительского каталога:

$ **cd *..***

Снова переместитесь в каталог *ex3* но используйте при этом относительный путь:

$ **cd *./ex3***

Выполните перемещение в домашний каталог, используя его спецобозначение и проверьте ваше местонахождение командой pwd:

$ **cd *~***

**Задание 6.** Используя команду mv переместите созданный вами ранее файл *example1.txt* в каталог *./examples* и проверьте результат с использованием команды **ls**:

$ **mv *example1.txt examples/***

Перейдите в каталог *examples* и используйте команду mv для переименования файла *example1.txt* в *example2.txt*:

$ **mv *example1.txt example2.txt***

Создайте 2 копии файла example2.txt:

$ **cp *example2.txt example1.txt***

$ **cp *example2.txt example1.txt.bak***

**Задание 7.** Создайте жесткую ссылку на файл example1.txt так, чтобы она находилась в домашнем каталоге:

$ **ln *~/examples/example1.txt ~/ex1-hardink***

Выведите содержимое домашнего каталога командой **ls** с ключами ***–l*** и ***–R***. Определите, сколько имен есть у файла *ex1-hardlink* и у созданного ранее *example1.txt*.

Используя команду **ls** и ключ ***–i*** выведите номера индексных дескрипторов для файла *example1.txt* и *ex1-hardlink*:

$ **ls *–i ex1-hardlink ./examples/example1.txt***

Для файла *example1.txt* создайте символьную ссылку в домашнем каталоге:

$ **ln *–s examples/example1.txt ex1-symlink***

Выведите содержимое домашнего каталога командой **ls *–li*** и определите, чем отличается вывод имени символьной ссылки от других файлов, а также определите, отличается ли номер индексного дескриптора (inode) символьной ссылки от жесткой ссылки.

**Задание 8.** Удалите файл *example1.txt* при помощи команды **rm**:

$ **rm *examples/example1.txt***

Выведите подробное описание жесткой ссылки, созданной на удаленный файл ранее:

$ **ls *–l ex1-hardlink***

Убедитесь, что жесткая ссылка на файл *example1.txt* продолжает существовать, а количество жестких ссылок на этот файл уменьшилось на 1.

Удалите все жесткие ссылки на файл *example1.txt*.

В каталоге ex1 создайте новый пустой файл с выбранным вами именем. Используя команду rm удалите каталог ex1:

*Список контрольных вопросов*

1) Что является файл?

2) Что такое файловая система?

3) Какой командой можно создать пустой файл?

4) Какой командой можно вывести в консоль текстовую строку?

5) Какая команда применяется при определении типа данных, содержащихся в файлах и каталогах?

6) Какой командой можно вывести в консоль содержимое файлов?

7) При помощи какой команды можно узнать содержимое каталога?

8) Что такое полный путь и относительный путь?

9) Что такое домашний каталог?

10) Как определить текущий каталог?

11) Какой ключ для команды **ls** необходимо применить для того, чтобы отличить файлы от каталогов? Какой ключ этой команды необходимо применить для вывода полного списка содержимого каталога?

12) Что такое родительский каталог?

13) Какая команда позволяет создавать каталоги? Какой ключ необходимо применить, чтобы создать путь к создаваемому каталогу?

14) Какая команда позволяет вам изменять текущий каталог (перемещаться между каталогами)?

15) Какая команда позволяет переместить файл из одного каталога в другой?

16) Какая команда позволяет создавать копии файлов?

17) Что такое жесткие ссылки? Какая команда позволяет их создавать? Для чего применяются жесткие ссылки?

18) Какой ключ команды **ls** необходимо использовать для вывода содержимого каталога с подробным описанием?

19) Что такое символьная ссылка? Для чего она применяется? Как создать символьную ссылку?

20) Какими командами можно удалять файлы и каталоги? Какие ключи необходимо при этом использовать?

**Практическая работа №3: Доступ процессов к файлам и каталогам и права доступа**

**Задание 1.** Изучите команду **ps** и **pstree**. Продемонстрируйте выполнение утилит на примере.

**Задание 2.** Изучите вопрос активных и фоновых процессов в Linux. Продемонстрируйте действие команд fg и bg на примерах.

**Задание 3.** Изучите утилиты top и htop. Продемонстрируйте их работу на примерах.

**Задание 4.** Используя команды kill и killall, продемонстрируйте их выполнение на примерах: завершение процессов по PID и имени процесса.

**Задание 5.** Изучите команды chmod и chown. Продемонстрируйте их работу на примерах.

*Список контрольных вопросов*

1) Что такое процесс?

2) Что такое PID?

3) Что такое PPID?

4) Что такое активный процесс?

5) Что такое фоновый процесс?

6) Для чего применяют утилиту **top**?

7) Как активный процесс сделать фоновым и наоборот, фоновый – активным?

8) Что такое сигнал?

9) Для чего применяют команду kill?

10) Что такое сценарий?

11) Кто такой «суперпользователь»?

12) Какая команда позволяет пользователю определить права доступа к файлу или каталогу?

13) На какие группы разделены индивидуальные права доступа к файлам и каталогам?

14) Какая команда применяется для изменения прав доступа к файлам и/или каталогам?

15) Какая команда применяется для изменения владельца файла/каталога?

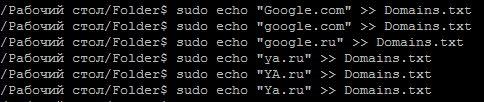
16) Какая команда применяется для изменения группы владельцев файла/каталогов?

**Практическая работа №4: Работа с текстовыми данными**

**Задание 1.** Выполните указанные далее действия и изучите работу утилиты **grep**.

Используя команду *mkdir*, создайте на рабочем столе папку “Folder”.

В этой папке создайте текстовый файл “Domains.txt”, как показано далее. Сохраните при этом регистр – это важно.



Используйте команду *cat*, чтобы вывести содержимое файла “Domains.txt”. Убедитесь, что регистр для имен был сохранен.

Используйте команду *grep*, чтобы вывести все строки, в которые входит заданная текстовая конструкция.





Изучите результат и сравните с использованным ранее выводом команды *cat*.

Измените использование команды *grep*, добавив ключ –i.





Изучите результат, сравните его с выводом команды *cat*.

Используйте команду *grep* в следующем варианте.





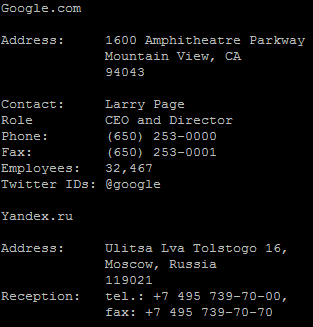
Дополните команду grep ключом –w.



Измените команду *grep*, дополнив ее ключом –n.



Измените содержимое файла “Domains.txt”, используя текстовый редактор.



Примените команду *grep* к измененному файлу.



Дополните команду *grep* параметром –A.



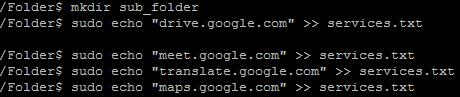
Используйте команду *grep* с параметром –B.



Используйте команду *grep* с параметром –C.



Создайте в каталоге “Folder” два новых файла.



Используйте поиск по всему содержимому каталога “Folder”.

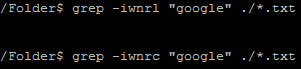


Создайте во вложенном каталоге “sub\_folder” новый файл.



Используйте поиск по всему содержимому каталога “Folder”, включая вложенные каталоги.





Используйте команду *pip3 freeze*.



Используйте фильтрацию *grep*.



Используйте регулярные выражения для поиска.





Запишите в новый файл “patterns.txt” регулярное выражение “\d{3}-\d{2}-\d{2}”.

Удалите файл “tel.txt”.

Примените *grep*, используя регулярное выражение из файла “patterns.txt”.



**Практическая работа №5**

**Задание 1.** Изучите команды **alias и unalias**. Продемонстрируйте их действия на примере.

**Задание 2**. Выполните указанные далее действия и изучите вариант применения команд и переменных при создании сценариев.

1. В вашем рабочем каталоге создайте папку “MyScripts” с использованием команды *mkdir*.
2. Перейдите в каталог “MyScripts”, используя команду *cd*.
3. Создайте в каталоге “MyScripts” файл “MyScript1.sh” используя команду *nano*:



1. В открывшемся файле напишите ваш первый bash-скрипт.



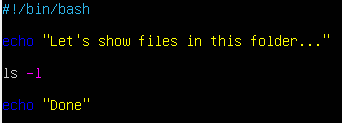
1. Сохраните и закройте файл вашего первого bash-скрипта.
2. Запустите ваш bash-скрипт:



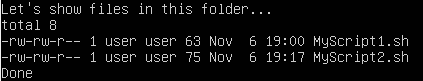
1. Оцените результат выполнения команды.
2. Запустите ваш bash-скрипт, используя команду *bash*:



1. Сравните результат выполнения команды с предыдущем вызовом (см. п.6).
2. Создайте в каталоге “MyScripts” файл “MyScript2.sh” используя команду *nano*.
3. В теле нового bash-скрипта запишите следующий код.



1. Сохраните и закройте ваш новый скрипт. Вызовите его выполнение, используя команду *bash*.
2. В результате, у вас должен появиться список файлов в каталоге “MyScripts” с назначенными правами доступа.



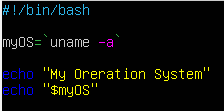
1. Измените права доступа к этим двум файлам, используя команду *chmod*.



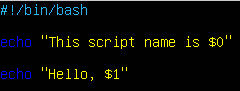
1. Запустите bash-скрипт “MyScript2.sh”.



1. Оцените результат выполнения команды из п.14.
2. В bash-скрипте “MyScript3.sh” введите указанный ниже код. Затем выполните этот bash-скрипт.



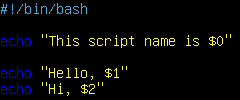
1. Оцените результат выполнения bash-скрипта.
2. Создайте bash-скрипт “MyScript4.sh”. В его теле укажите следующий код.



1. Выполните bash-скрипт, указав входное значение свое имя. Например, так.



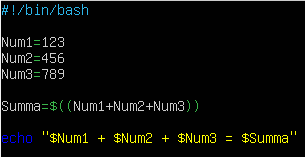
1. Модифицируйте код, как показано ниже.



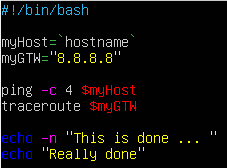
1. Запустите bash-скрипт, указав уже два входных значения. Например, так.



1. Создайте bash-скрипт “MyScript5.sh”. В его теле укажите следующий код.



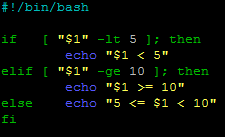
1. Выполните bash-скрипт. Оцените результат.
2. Создайте bash-скрипт “MyScript6.sh”. В его теле укажите следующий код.



1. Выполните bash-скрипт и оцените результат.

**Задание 3.** Выполните указанные далее действия и изучите вариант применения операций сравнения, циклов и функций при создании сценариев.

1. Создайте в каталоге “MyScripts” файл “Script1.sh”. В теле bash-скрипта запишите указанный код.

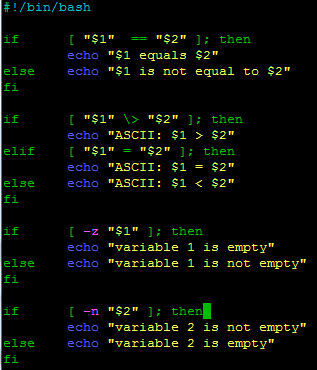


Сохраните bash-скрипт и примените его с одним целочисленным параметром несколько раз (например, с параметром 2, 7, 12).

Изучите результат для того, чтобы понять, как работает код bash-скрипта.

Создайте новый или модифицируйте этот bash-скрипт, используя следующие операции сравнения: -eq, -ne, -gt, -ge, -lt, -le.

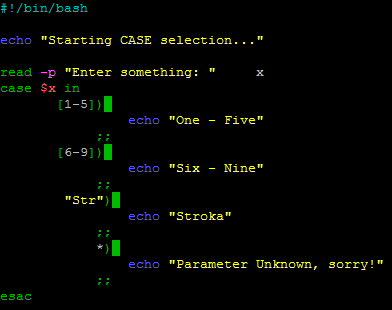
1. Создайте в каталоге “MyScripts” файл “Script2.sh”. В теле bash-скрипта запишите указанный код.



Сохраните bash-скрипт и примените его с двумя строковыми параметрами несколько раз (например, с параметрами “Str” и “str”, “str” и “Str”, “STR1” и “STR”). Примените bash-скрипт с одним параметром, например, с “STR”.

Изучите результат для того, чтобы понять, как работает код bash-скрипта.

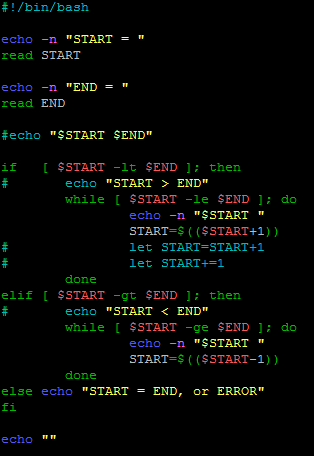
1. Создайте в каталоге “MyScripts” файл “Script3.sh”. В теле bash-скрипта запишите указанный код.



Сохраните bash-скрипт и примените его. В качестве переменной после начала работы скрипта введите один параметр. Выполните процедуру несколько раз (например, с параметром 2, 7, “Str”, “LabuLabuDapDap”).

Изучите результат для того, чтобы понять, как работает код bash-скрипта.

1. Создайте в каталоге “MyScripts” файл “Script4.sh”. В теле bash-скрипта запишите указанный код.



Сохраните bash-скрипт и примените его несколько раз. В первый раз – первый аргумент должен быть больше второго (например, 12 и 22). Во второй раз – второй аргумент больше первого (например, 16 и 4). В третий раз – первый и второй аргументы равны. В четвертый раз – вместо одного из аргументов введите символы (например, 12 и “STR”, или “STOP” и 1).

Изучите результат для того, чтобы понять, как работает код bash-скрипта.

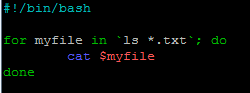
1. В окне терминала создайте несколько текстовых файлов (не меньше 3х), как показано в примере.



Просмотрите список созданных вами текстовых файлов.

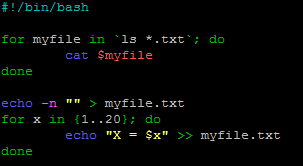


Создайте в каталоге “MyScripts” файл “Script5.sh”. В теле bash-скрипта запишите указанный код.



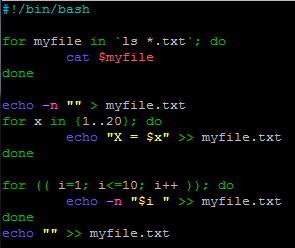
Сохраните bash-скрипт и примените его. Изучите результат для того, чтобы понять, как работает код bash-скрипта.

Модифицируйте код bash-скрипта.

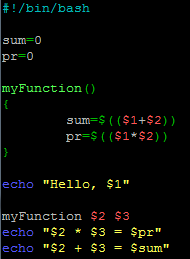


Сохраните bash-скрипт и примените его. Изучите результат для того, чтобы понять, как работает код bash-скрипта.

Модифицируйте bash-скрипт еще раз. Примените его. Изучите результат.



1. Создайте в каталоге “MyScripts” файл “Script6.sh”. В теле bash-скрипта запишите указанный код.



Сохраните bash-скрипт и примените его, задав 3 входных значения (например, значения “Mr.Freeman”, 6, 16). Изучите результат для того, чтобы понять, как работает код bash-скрипта.

*Список контрольных вопросов*

1) Зачем в bash-скрипте в первой строке всегда указывают “#!/bin/bash”?

2) К какому результату приводит использование конструкции echo в bash-скрипте?

3) Как в bash-скрипте прописать выполнение какой-либо команды, например, *ls –l*?

4) К какому результату приводит выполнение команды *chmod a+x*?

5) Как в bash-скрипте задать переменную и присвоить ей вывод какой-либо команды?

6) Как в bash-скрипте вывести на экран значение переменной?

7) Какое значение хранит в себе переменная “$0”?

8) Как использовать переменные в bash-скрипте так, чтобы им присваивались значения, указанные при выполнении bash-скрипта?

9) Как в bash-скрипте выполнять арифметические действия?

10) Как в bash-скрипте выполнить команду echo таким образом, чтобы курсор не переходил на строку ниже?

11) Как в bash-скриптах работает конструкция if-elif-else-fi? Что такое fi в этой конструкции?

12) Какие операции сравнения можно выполнить в конструкции if-elif-else-fi?

13) Какой код bash-скрипта позволяет пользователю вводить значения параметров при его (скрипта) выполнении?

14) Как работает в bash-скриптах конструкция case-esac? Что такое esac?

15) Как в bash-скриптах реализуется цикл while?

16) Как в bash-скриптах реализуется цикл for?

17) Как в bash-скриптах можно записать echo в файл?

18) Как задать функцию в bash-скрипте? Как передавать в нее параметры?

**Практическая работа №6: Управление пакетами**

**Задание 1.** Используя менеджер пакетов **apt**, выполните установку 2 разных СУБД: mysql и postgresql.

1. Установите файлы сервера и клиента MySQL, используя команду *apt*.



Задайте для суперпользователя баз данных root пароль и подтвердите его (если понадобится).

Войдите в СУБД MySQL.



Используя команду *show*, выведите в терминал список всех существующих баз данных.



Создайте тестовую базу данных, используя команду *create database*.



Снова выведите на экран список всех существующих баз данных, чтобы убедиться, что тестовая база была успешно создана.

Удалите тестовую базу данных, используя команду *drop database*.



Выведите на экран список всех существующих баз данных, чтобы убедиться, что тестовая база была успешно удалена.

1. Используя команду *apt*, установите сервер PostgreSQL.



Выполните вход в базу данных PostgreSQL через консоль.



Задайте пароль для пользователя psql.



Создайте тестовую базу данных, используя команду *create database*.



Выведите на экран список всех существующих баз данных, используя команду *\l*.



Удалите тестовую базу данных, используя команду *drop database*.



Выведите на экран список всех существующих баз данных, чтобы убедиться, что тестовая база данных была удалена.

1. Удалите установленные пакеты mysql и postgresql при помощи утилиты **apt**.

**Задание 2.** Изучите работу утилиты tar, продемонстрируйте на примерах как:

а) создать архив для одного файла/каталога;

б) создать архив для нескольких файлов/каталогов;

в) просмотреть содержимое архивов;

г) как разархивировать созданные ранее архивы.

**Задание 3.** В вашей графической оболочке создайте ярлык на указанное далее приложение.

1. Установите в вашей ОС Linux пакет anaconda3.

<https://www.anaconda.com/products/individual>

1. В установленном пакете обязательно должен обязательно содержаться веб-приложение The Jupyter Notebook. Проверьте его работу.
2. Создайте на рабочем столе вашей ОС Linux ярлык, который будет запускать данное приложение. При этом, окно терминала при запуске должно быть скрытым, а в качестве иконки укажите лого приложения Jupyter:

<https://upload.wikimedia.org/wikipedia/commons/thumb/3/38/Jupyter_logo.svg/800px-Jupyter_logo.svg.png>

*Список контрольных вопросов*

1) Для чего применяют утилиту **apt**?

2) Как создавать архивы в терминале?

**Практическая работа №7: Сеть TCP/IP и сетевые возможности**

**Задание 1.** Разверните на вашей машине с Linux сервер telnet. Продемонстрируйте преподавателю работу сервиса.

**Задание 2.** Разверните на вашей машине с Linux сервер SSH. Продемонстрируйте преподавателю работу сервиса:

а) с входом по паролю;

б) с входом по паролю и ключу;

в) с входом только по ключу.

**Задание 3.** Изучите следующие команды и продемонстрируйте их работу на примерах:

а) ping и traceroute;

б) ifconfig и ip addr;

в) route и ip route;

г) host и dig;

д) netstat.

*Список контрольных вопросов*

1) Для чего применяют telnet и ssh?

2) Какую команду вы примените для того, чтобы узнать информацию о настройках сетевого адаптера/интерфейса?

3) Как узнать присвоенные IP для ваших сетевых адаптеров?

4) Как проверить целостность и качество соединений в сети на основе TCP/IP? Как ограничить число запросов?

5) Какую команду вы будете использовать, чтобы отследить путь к интересующему вас сетевому ресурсу?

6) Как определить ip-адрес сервера по его имени?

7) Как узнать информацию о сетевых подключениях компьютера? Как увидеть все порты, порты TCP, порты UDP, список всех интерфейсов?