БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра технологий программирования

#### ОТЧЕТ

#### по лабораторной работе №1

По курсу “Программирование мобильных и встраиваемых систем”

#### Базовые команды для работы с ОС LINUX для мобильных и встраиваемых платформ

**Подготовил:**  
студент 3 курса 13 группы кафедры ТП  
Петров Андрей Александрович

Минск, 2021 г.

# ЦЕЛЬ РАБОТЫ:

Ознакомление с операционной системой ОС Unix/Linux и основными командами работы с файлами, и каталогами, правами доступа и процессами ОС Unix/Linux.

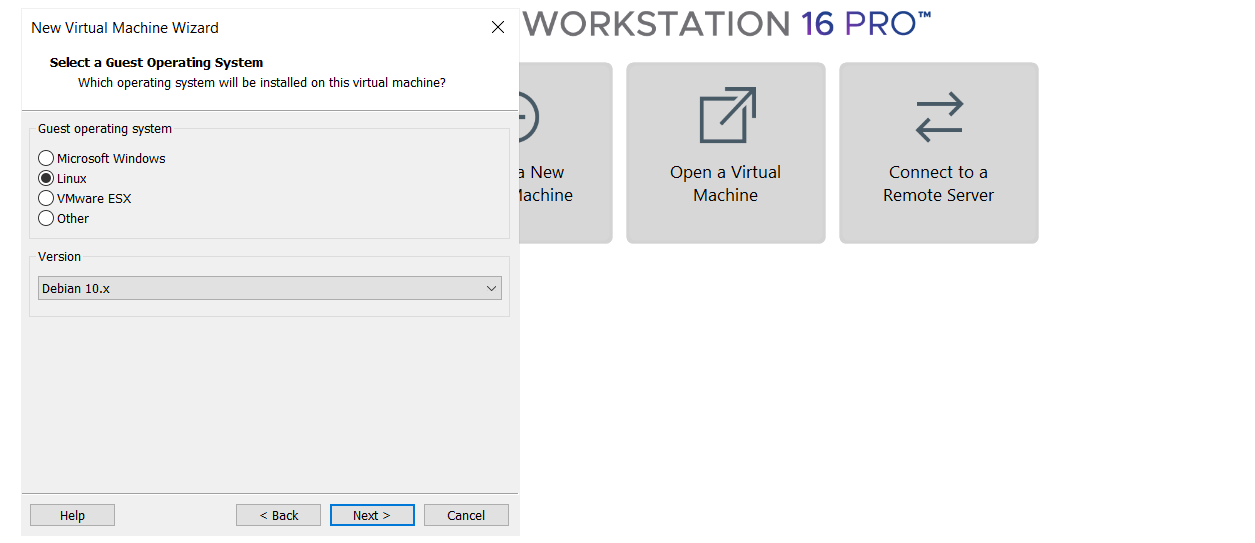
# ЗАДАЧИ РАБОТЫ:

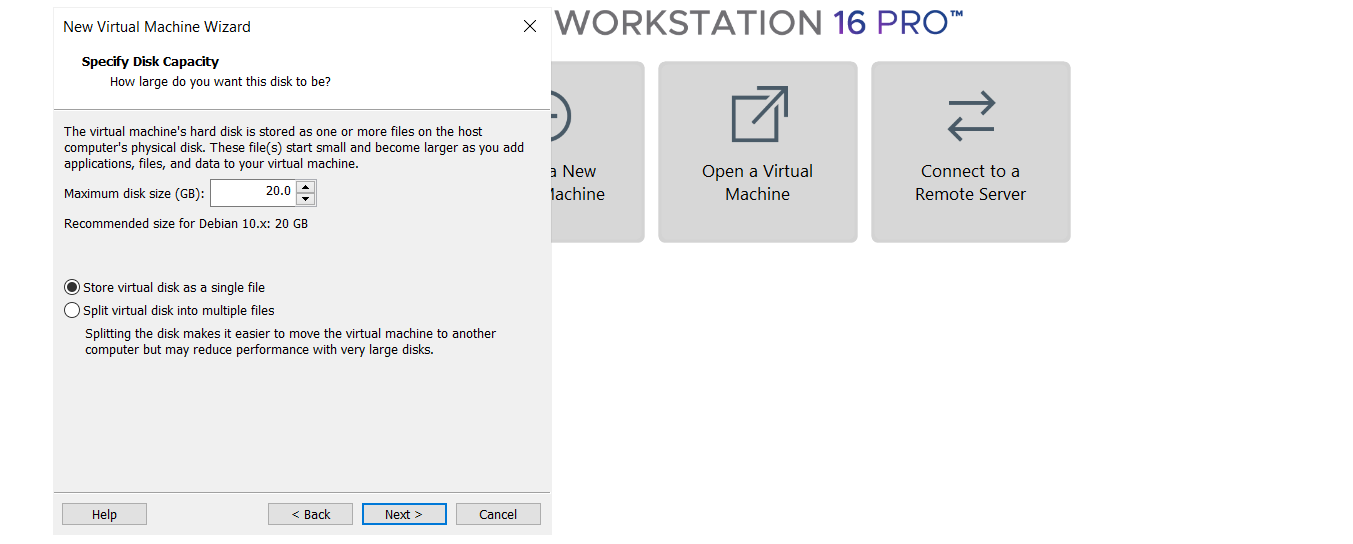
* 1. Изучить процедуры входа в систему, выполнения основных команд, получения подсказки, создания и редактирования текстовых файлов.
  2. Изучить основные принципы работы по протоколу ssh.
  3. Изучить основные команды для создания каталогов, файлов, жестких ссылок и символических ссылок, включая архивацию и синхронизацию данных.
  4. Изучить команды создания управления учетными записями, команды изменения прав доступа и владения chmod и chown.
  5. Изучить работу и ключи команд ps, pidof, fuser — получения сведений о процессах в операционной системе.

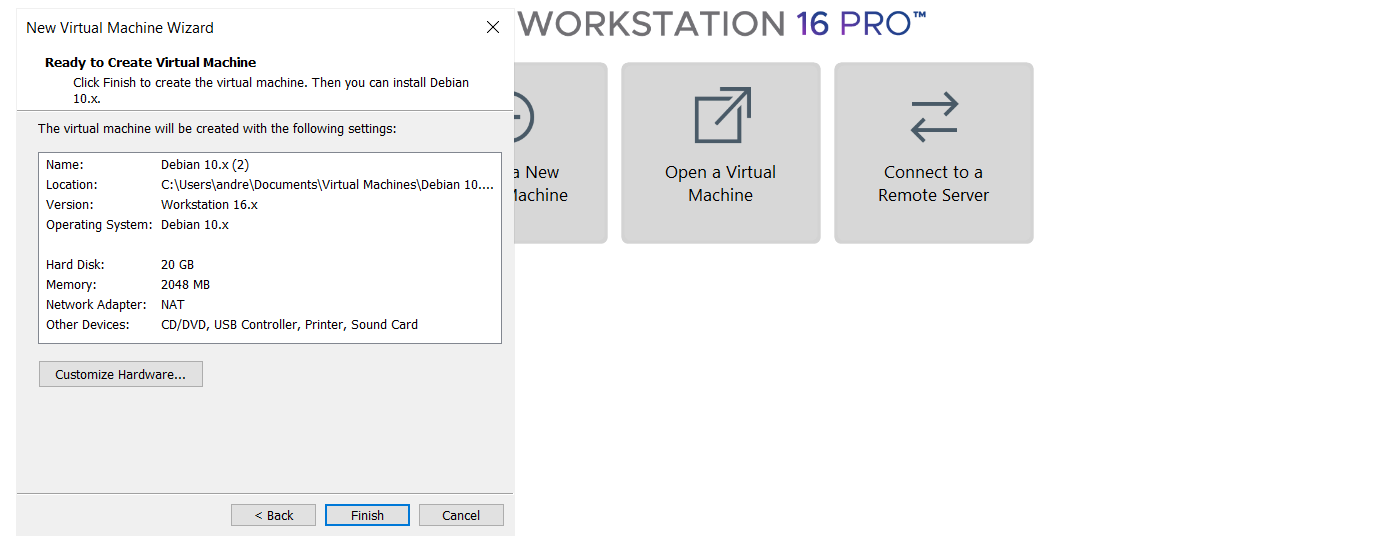
## ЗАДАНИЕ 1. РАЗВЕРТЫВАНИЕ ВИРТУАЛЬНЫХ МАШИН ПОД УПРАВЛЕНИЕМ ОС RASPBERRY PI И ОС UBUNTU

* 1. Изучить материалы учебного курса [Raspbian x86 on VirtualBox on a](https://edufpmi.bsu.by/mod/url/view.php?id=439) [Windows PC](https://edufpmi.bsu.by/mod/url/view.php?id=439) и [Как установить Ubuntu в VirtualBox](https://edufpmi.bsu.by/mod/url/view.php?id=680).
  2. Установить OC Raspberry PI Desktop. Учесть, что в процессе установки создается учетная запись с именем pi и паролем raspberry. Пароль после установки рекомендуется сменить. Созданный пользователь обладает правами администратора. Продемонстрировать процедуру установки скриншотами в отчете.









Изображение выглядит как текст

Автоматически созданное описание



Изображение выглядит как текст, внешний, снимок экрана

Автоматически созданное описание

* 1. Установить OC Ubuntu Mate или Kali Linux. 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.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, монитор, электроника, компьютер

Автоматически созданное описание

## ЗАДАНИЕ 2. НАСТРОЙКА SSH-СОЕДИНЕНИЙ МЕЖДУ ХОСТОВОЙ СИСТЕМОЙ И ГОСТЕВЫМИ МАШИНАМИ

* 1. Изучить материалы из учебного курса по настройке ssh в ОС Raspberry PI и ОС Ubuntu:
  + [Как настроить ssh-соединение в гостевой ОС в Virtualbox](https://edufpmi.bsu.by/mod/resource/view.php?id=443)
  + [Как настроить ssh в ОС Raspbian](https://edufpmi.bsu.by/mod/url/view.php?id=681)
  + [Как настроить ssh ОС UBuntu](https://edufpmi.bsu.by/mod/url/view.php?id=682)
  + [Как пользоваться SSH? А также установка и настройка (Linux, Windows,](https://androidp1.ru/kak-polzovatsja-ssh-a-takzhe-ustanovka-i-nastrojka/) [macOS, Android, iOS)](https://androidp1.ru/kak-polzovatsja-ssh-a-takzhe-ustanovka-i-nastrojka/)
  1. Настроить ssh-соединение в гостевой виртуальной ОС Raspberry PI и Ubuntu. Основные этапы настройки продемонстрировать скриншотами в отчете.



Изображение выглядит как текст, монитор, ноутбук, снимок экрана

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

* 1. Подключиться по ssh из ОС Raspberry PI в Ubuntu и продемонстрировать скриншотами в отчете.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

* 1. В зависимости от вашей основной операционной системы изучите рекомендации:
  + [Как пользоваться SSH? А также установка и настройка (Linux, Windows,](https://androidp1.ru/kak-polzovatsja-ssh-a-takzhe-ustanovka-i-nastrojka/)
  + [macOS, Android, iOS)](https://androidp1.ru/kak-polzovatsja-ssh-a-takzhe-ustanovka-i-nastrojka/)
  + Для Windows: «[Как активировать ssh-клиент в Windows 10](https://edufpmi.bsu.by/mod/resource/view.php?id=442)» или установите клиент Putty для создания терминальных сессий, используя инструкцию «[Как подключиться по ssh из Windows](https://beget.com/ru/articles/ssh_windows)».
  + Для macOS: «[Как использовать встроенный ssh-клиент в macOS](http://www.prostomac.com/2017/06/kak-ispolzovat-vstroennyj-v-terminal-ssh-klient/)»
  1. Открыть ssh-клиент в вашей хостовой системе Windows 10 или macOS. Подключитесь из хостовой машины в гостевые виртуальные машины под управлением ОС Raspberry PI и Ubuntu. Процесс подключения продемонстрировать скриншотами в отчете.

Изображение выглядит как текст

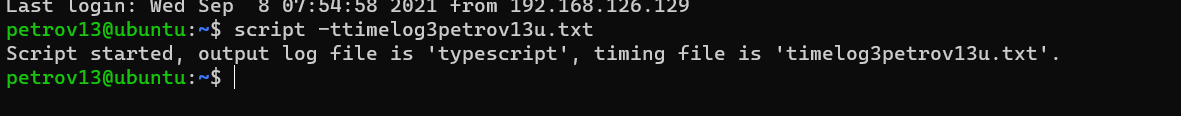
Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

## ЗАДАНИЕ 3. ИСПОЛЬЗОВАНИЕ КОМАНДЫ SCRIPT ДЛЯ ПРОТОКОЛИРОВАНИЯ КОМАНДАМ

* 1. Изучить руководство по команде script в материале [Руководство по](https://edufpmi.bsu.by/mod/url/view.php?id=675) [команде script](https://edufpmi.bsu.by/mod/url/view.php?id=675) из учебного курса. Обратить внимание на пример команды, которая позволяет протоколировать команды и вести журнал меток времени (пример 4).
  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. Включить ведение протокола командой script с журналом меток времени. **Протокол** назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu. **Журнал меток** назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu.



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

Изображение выглядит как текст

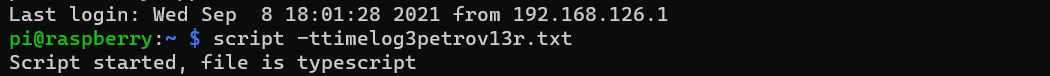
Автоматически созданное описание

* 1. Завершить протоколирование команд в Ubuntu, т. е.ввести команду exit.
  2. Не выключая Ubuntu, включите гостевую ОС Raspberry PI. Подключитесь по ssh к гостевой виртуальной ОС Raspberry PI под учетной записью пользователя pi.

Изображение выглядит как текст

Автоматически созданное описание

* 1. Включить ведение протокола командой script с журналом меток времени. **Протокол** назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu. **Журнал меток** назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu.



* 1. Создайте каталог, задав в качестве имени вашу фамилию латиницей строчными буквами. Данные каталог использовать для размещения результатов лабораторной работы в ОС Raspberry PI и сохраняйте в него результаты работы.
  2. Выполнить команды cal, date, who, whoami.

Изображение выглядит как текст

Автоматически созданное описание

* 1. Создать текстовый файл в ОС Raspberry PI с произвольным содержанием при помощи команды tee. Просмотреть содержимое файла при помощи команды cat и исправить в нем ошибки при помощи текстового редактора vi.
  2. Изучить справку на команды cp, unlink.
  3. Создать копию файла, созданного в п. 3.10, командой cp, удалить ее командой unlink.
  4. Завершить протоколирование команд в Raspberry PI, т. е. ввести команду exit.

Изображение выглядит как текст

Автоматически созданное описание

## ЗАДАНИЕ 4. БЕЗОПАСНОЕ КОПИРОВАНИЕ ФАЙЛОВ И ПУБЛИКАЦИЯ В РЕПОЗИТОРИЙ

* 1. Если время действия ssh-сессии к гостевой ОС Ubuntu истекло, подключиться по ssh к ней.
  2. Включить ведение протокола командой script с журналом меток времени. **Протокол** назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu. **Журнал меток** назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu.



* 1. Создайте каталог, задав в качестве имени вашу фамилию латиницей строчными буквами. Данные каталог использовать для размещения результатов лабораторной работы в ОС Ubuntu.
  2. Установить git, если требуется.
  3. Перейдите в каталог, созданный в п 4.3. Склонируйте репозиторий из github.

Изображение выглядит как текст

Автоматически созданное описание

* 1. Скопируйте в каталог, созданный в п. 4.3, результаты работы в Ubuntu, полученные в ходе выполнения задания 3.
  2. Скопируйте с помощью scp из Raspberry PI сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.

Изображение выглядит как текст

Автоматически созданное описание

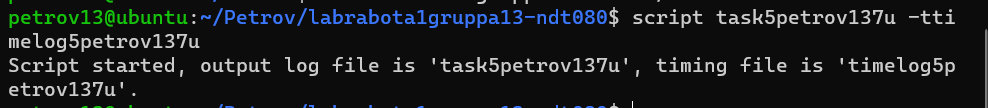
* 1. Завершите протоколирование команд с помощью script, т. е. введите команду exit.
  2. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения заданий 3 и 4, в удаленный.

Изображение выглядит как текст

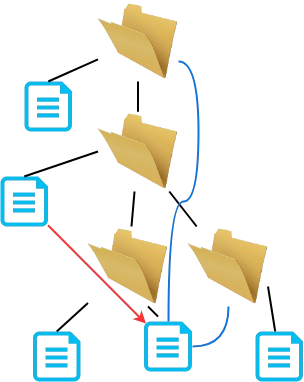
Автоматически созданное описание

## ЗАДАНИЕ 5. УПРАВЛЕНИЕ ФАЙЛАМИ И ПАПКАМИ

* 1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой script с журналом меток времени. **Протокол** назвать по следующему шаблону — taskXФамилияNKM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, K — номер варианта, M — **r** — для Raspberry PI, **u** — для Ubuntu. **Журнал меток** назвать по следующему шаблону — timelogXФамилияNKM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, K — номер варианта, M — **r** — для Raspberry PI, **u** — для Ubuntu.



* 1. 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:



Вариант 7

Создадим каталоги:



Изображение выглядит как текст

Автоматически созданное описание

Создадим файлы в каталогах:



* 1. 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.

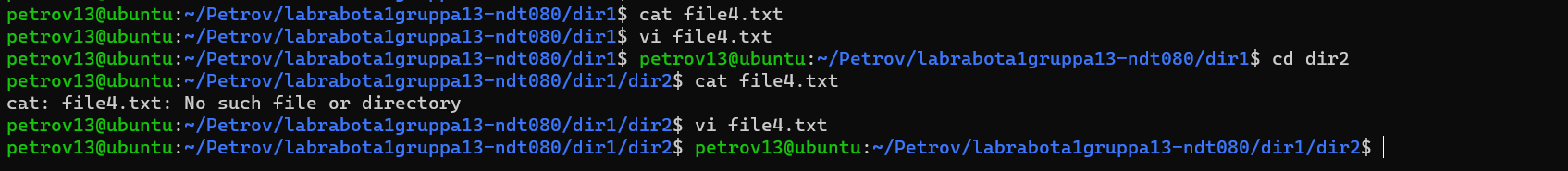
Командой *ln* создаем ссылки, символические ссылки - *ln -s* . Убирает ссылки команда *unlink*



Изображение выглядит как текст

Автоматически созданное описание

Экспериментируем с доступом:

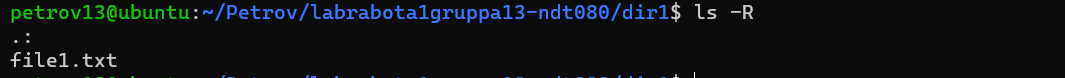


Изображение выглядит как текст

Автоматически созданное описание

Производим очистку:



* 1. Завершите протоколирование команд с помощью script, т. е. введите команду exit.
  2. Подключитесь по ssh из Ubuntu в ОС Raspberry PI и выполните шаги 5.1-5.4.
  3. Скопируйте с помощью scp из Raspberry PI сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.



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

Изображение выглядит как текст

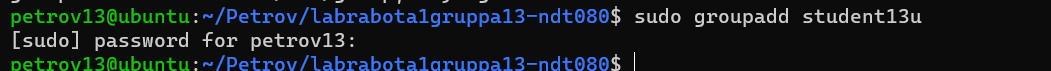
Автоматически созданное описание

## Задание 6. Управление пользователями и права доступа

* 1. Изучить справку к командам useradd, groupadd, chmod и chown.
  2. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой script с журналом меток времени. **Протокол** назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu. **Журнал меток** назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu.



* 1. Добавить группу studentNM в ОС Ubuntu, где N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu.



* 1. Добавить своего пользователя для каждой ОС в созданную группу.





* 1. Открыть для своей основной группы доступ к своему домашнему каталогу 3 — для поиска в каталоге и к рабочей копии файла в домашнем каталоге — для чтения и записи.



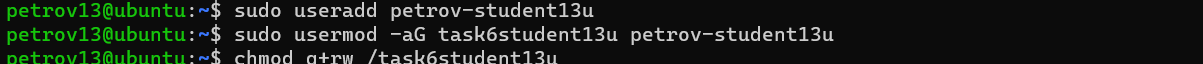


* 1. Создать пользователя studentNM в ОС Ubuntu, где N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu. Открыть доступ на чтение и запись к одному из свиох каталогов и одному из своих файлов для созданного пользователя.

Создадим группу для каталога, в которую поместим нашего пользователя.



Создадим пользователя и его домашнюю директорию.







Присвоим группу каталогу и настроим права.

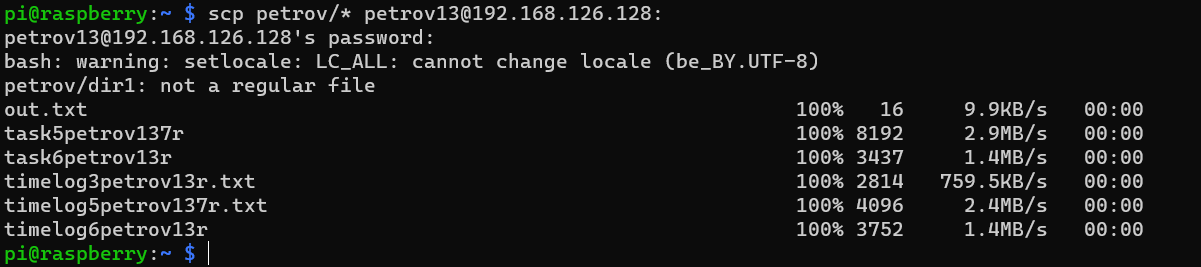




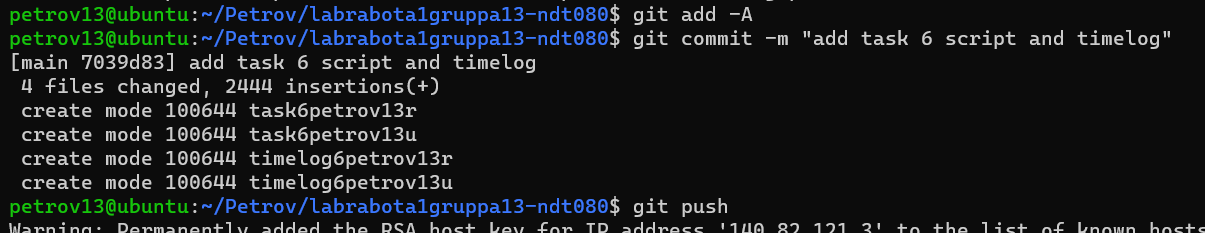




* 1. Подключиться по ssh с созданной учетной записью или авторизоваться в гостевой ОС. И проверить права доступа на свой каталог и файл, внеся в них изменения.
  2. Переключиться в свою учетную запись. Проверить изменения.
  3. Закрыть доступ к своему домашнему каталогу.
  4. Завершите протоколирование команд с помощью script, т. е. введите команду exit.
  5. Подключитесь по ssh из Ubuntu в ОС Raspberry PI и выполните шаги 6.2-6.10.
  6. Скопируйте с помощью scp из Raspberry PI сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.



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



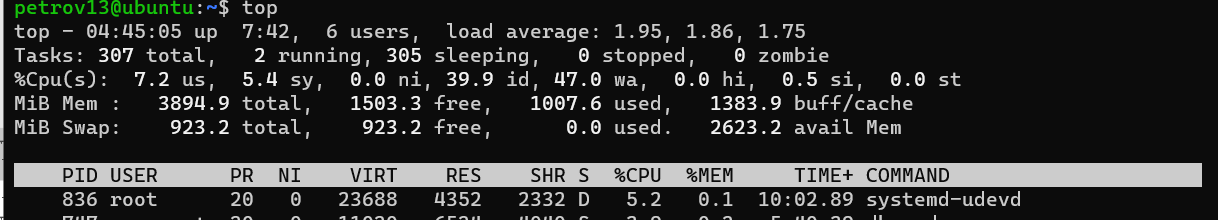
## Задание 7. Базовые команды для управления процессами

* 1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой script с журналом меток времени. **Протокол** назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu. **Журнал меток** назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu.
  2. Изучите справку к команде ps и статью [https://www.tecmint.com/ps-](https://www.tecmint.com/ps-command-examples-for-linux-process-monitoring/) [command-examples-for-linux-process-monitoring/](https://www.tecmint.com/ps-command-examples-for-linux-process-monitoring/), выполнить ее с ключами -a, -e, a, x, ax в ОС Raspberry PI, записать результаты в файл, например: ps -e > ps.log.

Изображение выглядит как текст

Автоматически созданное описание

* 1. Авторизуйтесь в ОС Ubuntu по ssh. Изучите справку к команде pidof и статью <https://www.tecmint.com/find-process-name-pid-number-linux/>. Запустите команду top. А в другом терминале, используя pidof определите ее идентификатор процесса. Завершите выполнение команды top.





* 1. Авторизуйтесь в ОС Ubuntu по ssh. Изучите справку к команде fuser и статью [https://www.tecmint.com/learn-how-to-use-fuser-command-with-examples-in-](https://www.tecmint.com/learn-how-to-use-fuser-command-with-examples-in-linux/) [linux/](https://www.tecmint.com/learn-how-to-use-fuser-command-with-examples-in-linux/). Откройте любой текстовый файл на редактирование. Определите, какие процессы используют ваш файл.

Изображение выглядит как текст

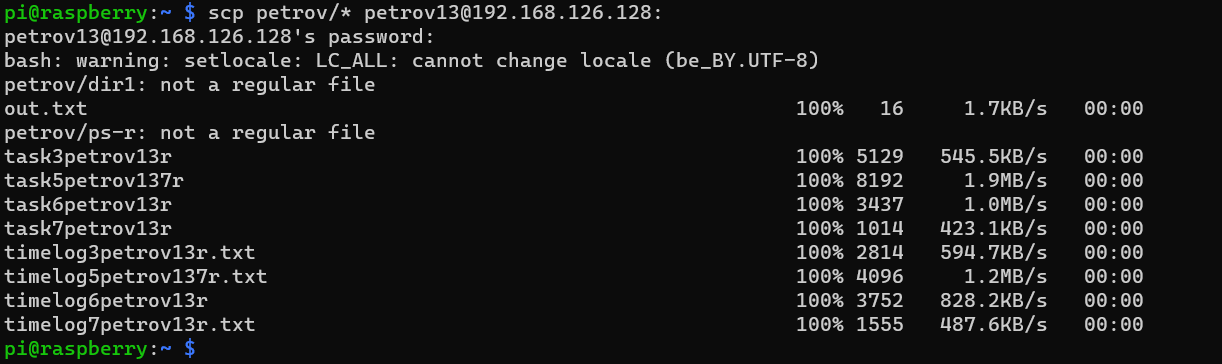
Автоматически созданное описание



* 1. Авторизуйтесь в ОС Ubuntu по ssh. Изучите справку к команде vmstat. Результаты вывода команды запишите в файл с указанием в названии файла вашу фамилию и имя и название используемой команды.



* 1. Авторизуйтесь в ОС Ubuntu по ssh. Изучите справку к команде lsof. Результаты вывода команды запишите в файл с указанием в названии файла вашу фамилию и имя и название используемой команды.
  2. Скопируйте с помощью scp из Raspberry PI сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.



* 1. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 7, в удаленный.

Изображение выглядит как текст

Автоматически созданное описание

## Задание 8. Сжатие и архивация файлов

* 1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой script с журналом меток времени. **Протокол** назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu. **Журнал меток** назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu.

#### Утилиты gzip и gunzip

* 1. Включить запись протокола команд и меток времени командой script.
  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.



* 1. Используя программу gzip, сжать файл любой текстовый файл в созданном каталоге согласно варианту из задания 5, чтобы исходный файл остался в текущем каталоге.



* 1. Проверить целостность сжатого файла.



* 1. Скопировать полученный архив с помощью scp в домашний каталог своего пользователя в ОС Raspberry PI. Распаковать файл архивы \*.gz в домашний каталог в ОС Raspberry PI.



Изображение выглядит как текст

Автоматически созданное описание

* 1. Завершить протоколирование команд.
  2. Проверить содержимое домашнего каталога в ОС Raspberry PI.

#### Утилита tar

* 1. Включить в ОС Ubuntu запись протокола команд и меток времени командой script.
  2. Создать в домашнем каталоге архив формата tar, включающий каталог и файлы.

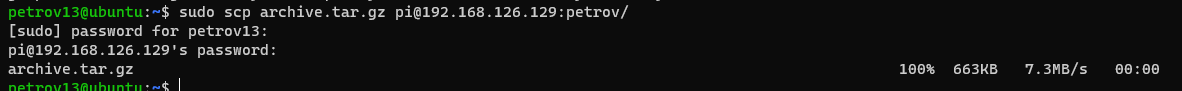
Изображение выглядит как текст

Автоматически созданное описание

* 1. Сжать полученный архив.



* 1. Скопировать полученный архив с помощью scp в домашний каталог своего пользователя в ОС Raspberry PI.



* 1. Завершить протоколирование команд.
  2. Подключиться к ОС Raspberry PI и распаковать архив.



* 1. Скопируйте сохранённые протоколы работы и добавить в git- репозиторий, созданный в ОС Ubuntu в п.4.5.

Изображение выглядит как текст

Автоматически созданное описание

* 1. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 8, в удаленный.

Изображение выглядит как текст

Автоматически созданное описание

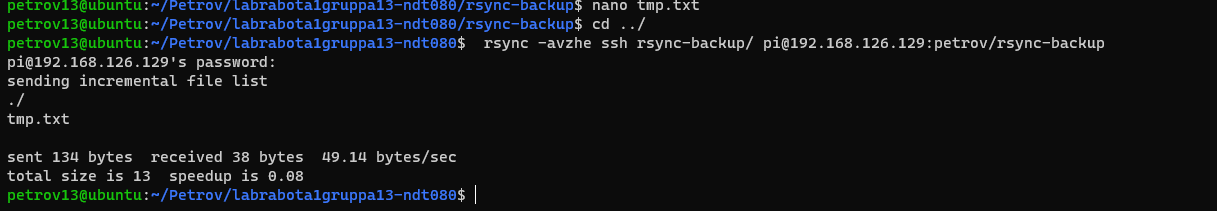
## Задание 9. Синхронизация в Linux с помощью утилиты rsync

* 1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой script с журналом меток времени. **Протокол** назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu. **Журнал меток** назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — **r** — для Raspberry PI, **u** — для Ubuntu.
  2. Создать в домашнем каталоге каталог rsync-backup в ОС Ubuntu и в ОС Raspberry PI.
  3. Скопировать несколько файлов в каталог rsync-backup в ОС Ubuntu.
  4. Используя rsync, скопировать содержимое каталога rsync-backup из ОС Ubuntu в каталог rsync-backup в ОС Raspberry PI.

Изображение выглядит как текст

Автоматически созданное описание

* 1. Внести изменения в любой из файлов в каталоге rsync-backup в ОС Ubuntu и синхронизировать изменения с каталогом в ОС Raspberry PI.



* 1. Скопируйте сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.



* 1. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 9, в удаленный.

Изображение выглядит как текст

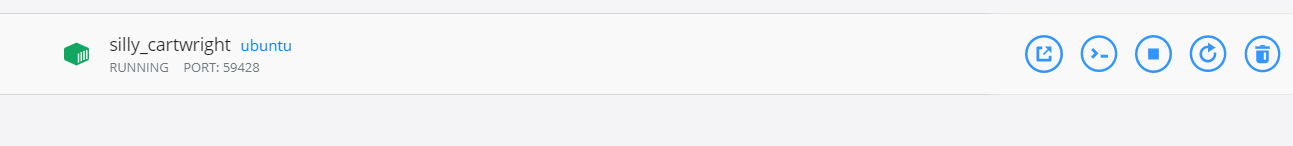
Автоматически созданное описание

## Задание 10. Подключение по ssh к контейнеру Docker

* 1. Задание выполнить в **хостовой** системе. Для Windows и macOS найти документацию по установке контейнера Docker.
  2. Просмотреть видео <https://youtu.be/V7lTLVzsK5U> с установкой docker на ОС Ubuntu 18.04.
  3. Выполнить установку docker согласно документации, например для ОС Ubuntu <https://docs.docker.com/engine/install/ubuntu/>.
  4. Установить docker и развернуть контейнер для ubuntu.







* 1. Изучить основные команды для работы с docker <https://youtu.be/XtO3hUWLKF8>.
  2. Установить Docker Compose согласно документации <https://docs.docker.com/compose/install/>.
  3. Изучить статью <https://phoenixnap.com/kb/how-to-ssh-into-docker-container> и <https://andreyex.ru/linux/kak-podklyuchitsya-po-ssh-k-kontejneru-docker/>.
  4. Для контейнера ubuntu настроить подключение по ssh, скопировать ключ ssh с хостовой системы и подключиться к контейнеру docker ubuntu по ssh. Продемонстрировать подключение по ключу ssh (без применения пароля) и по нестандартному порту для ssh (стандартный пор для ssh — 22).









Изображение выглядит как текст

Автоматически созданное описание

* 1. Изучить статьи для создания контенейра Docker для тестирования сборки для Android:

[https://andresand.medium.com/android-emulator-on-docker-container-](https://andresand.medium.com/android-emulator-on-docker-container-f20c49b129ef) [f20c49b129ef](https://andresand.medium.com/android-emulator-on-docker-container-f20c49b129ef)

[https://betterprogramming.pub/build-a-lightweight-docker-container-for-](https://betterprogramming.pub/build-a-lightweight-docker-container-for-android-testing-2aa6bdaea422) [android-testing-2aa6bdaea422](https://betterprogramming.pub/build-a-lightweight-docker-container-for-android-testing-2aa6bdaea422)

<https://andresand.medium.com/building-android-with-docker-8dbf717f54d4>

* 1. Настроить контейнер docker для Android и продемонстировать запуск эмулятора.

## Контрольные вопросы.

1. Какие ключи команды ls Вы знаете? Что они дают?

Команда ls имеет следующие ключи:

* -A - не отображать ссылку на текущую папку и корневую папку . и ..;
* -a - отображать все файлы, включая скрытые, это те, перед именем которых стоит точка;
* --author - выводить создателя файла в режиме подробного списка;
* -b - выводить Escape последовательности вместо непечатаемых символов;
* --block-size - выводить размер каталога или файла в определенной единице измерения, например, мегабайтах, гигабайтах или килобайтах;
* -B - не выводить резервные копии, их имена начинаются с ~;
* -c - сортировать файлы по времени модификации или создания, сначала будут выведены новые файлы;
* -C - выводить колонками;
* --color - включить цветной режим вывода, автоматически активирована во многих дистрибутивах;
* -d - выводить только директории, без их содержимого, полезно при рекурсивном выводе;
* -D - использовать режим вывода, совместимый с Emacs;
* -f - не сортировать;
* -F - показывать тип объекта, к каждому объекту будет добавлен один из специализированных символов \*/=>@|;
* --full-time - показывать подробную информацию, плюс вся информация о времени в формате ISO;
* -g - показывать подробную информацию, но кроме владельца файла;
* --group-directories-first - сначала отображать директории, а уже потом файлы;
* -G - не выводить имена групп;
* -h - выводить размеры папок в удобном для чтения формате;
* -H - открывать символические ссылки при рекурсивном использовании;
* --hide - не отображать файлы, которые начинаются с указанного символа;
* -i - отображать номер индекса inode, в которой хранится этот файл;
* -l - выводить подробный список, в котором будет отображаться владелец, группа, дата создания, размер и другие параметры;
* -L - для символических ссылок отображать информацию о файле, на который они ссылаются;
* -m - разделять элементы списка запятой;
* -n - выводить UID и GID вместо имени и группы пользователя;
* -N - выводить имена как есть, не обрабатывать контролирующие последовательности;
* -Q - брать имена папок и файлов в кавычки;
* -r - обратный порядок сортировки;
* -R - рекурсивно отображать содержимое поддиректорий;
* -s - выводить размер файла в блоках;
* -S - сортировать по размеру, сначала большие;
* -t - сортировать по времени последней модификации;
* -u - сортировать по времени последнего доступа;
* -U - не сортировать;
* -X - сортировать по алфавиту;
* -Z - отображать информацию о расширениях SELinux;
* -1 - отображать один файл на одну строку.

1. Чем отличаются man и info? Как с ними работать?

Команда info – это альтернативая команда команде man. info предоставляет информацию в гипертекстовом формате, что позволяет просматривать различные разделы, не выходя из оболочки, предоставляемой командой info, man – нет.

1. Команда script — назначение и применение.

Логирование всех вводимых команд с возможностью указания меток.

1. Какая команда используется для генерации ssh-ключа?

ssh-keygen -t rsa

1. Для каких задач применяется команда scp?

Передача файлов посредством протокола ssh.

1. Команды tee и cat. Назначение и применение. Чем cat отличается от more и less?

Команда tee считывает стандартный ввод (stdin), после чего записывает его в стандартный вывод (stdout) и одновременно копирует его в подготовленный файл или переменную tee [OPTIONS] [FILE]

Команда cat позволяет вывести на стандартный вывод (на экран) содержимое любого файла, однако она используется для этих целей очень редко, разве что для вывода очень небольших по объему файлов.

Команда more позволяет выводить содержимое больших файлов постранично. Перемещаться по страницам можно только вперёд.

Команда less работает аналогично, но позволяет двигаться по файлу в обоих направлениях.

1. Основные команды редактора vi.

:q – выход из программы

:!command – выполнение команды (например, git)

:p – вставка из буфера обмена

:w [filename] – сохранение

1. Ссылки и символические ссылки. Создание и применение.

Команда ln предназначена для создания ссылок на файл.

ln [-f] file1 [file1 ...] target\_file

Команда ln делает target\_file ссылкой на file1. file1 не должен совпадать с target\_file. Если target\_file является каталогом, то в нем создаются ссылки на file1, file1,... с теми же именами. Только в этом случае можно указывать несколько исходных файлов. Если целевой\_ target\_file существует и не является каталогом, его старое содержимое теряется.

Опции:

-f удаление существующего целевого файла

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

1. Создание и копирование файлов и папок в Linux.

touch [option] file\_name(s)

cp [option]... source... directory

mkdir [options...] [directories ...]

1. Перемещение и удаление файлов и папок в Linux.

mv [filename] [new\_filename]

rm [OPTION]... FILE...

rmdir [OPTION]... DIRECTORY...

1. Команды useradd, adduser, userdel, deluser, groupadd, addgroup, groupdel, delgroup . Назначение и применение.

useradd [options] username – добавление нового пользователя в систему;

userdel [options] LOGIN – удаление пользователя и принадлежащие ему файлы;

groupadd [options] group – добавление группы пользователей;

groupdel group – удаление группы

delgroup [options] [--only-if-empty] group – удаление группы пользователей.

1. Команды chmod и chown. Назначение и применение.

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.

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:

*owner group other-users*

*(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).

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

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.

1. Какие права доступа Вы имеете к своему домашнему каталогу, каталогам /home и / ?

Полные права.

1. Как вы считаете, почему в Linux разделены команды для сжатия и команды архивирования? Приведите их примеры.

Потому что в linux архивирование не есть одно и тоже, что и сжатие.

tar опцииf файл\_для\_записи /папка\_файлами\_для\_архива

gzip опции файл

1. Команда ps. Назначение и применение. Ключи команды.

Команда ps выводит в стандартный вывод информацию о текущем

состоянии процессов.

Ключи:

-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.

-p 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.

1. Команда pidof. Назначение и применение. Ключи команды.

pidof [options] command - ищет id процесса

1. Команда fuser. Назначение и применение. Ключи команды

fuser [options] [file|socket] - ищет процесс по файлам или сокетам