Государственное бюджетное образовательное учреждение среднего профессионального образования Новосибирской области

Новосибирский химико- 630102 г. Новосибирск, ул. Садовая, 26,

технологический колледж. Тел./факс: (383) 266-00-44, тел.: (383) 266-00-54,

nhtk@mail.ru, http://nhtk-edu.ru

им. Д.И. Менделеева

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ**

**ПРАКТИЧЕСКОЙ РАБОТЫ**

Разработчик:

Соломенный Ю.Ю.

Новосибирск – 2023

**Практическая работа №3 «Основы ОС Linux»**

**Теоретический материал**

### [Шпаргалка по командам терминала Linux | 100 самых частых команд (serverspace.ru)](https://serverspace.ru/support/help/shpargalka-po-komandam-linux/?ysclid=lm4qbbmh60227590356)

### Загрузка

Алгоритм включения сервера и загрузки Linux в большинстве случаев выглядит следующим образом:

* [BIOS](https://en.wikipedia.org/wiki/BIOS) / [UEFI](https://en.wikipedia.org/wiki/UEFI) → [MBR](https://en.wikipedia.org/wiki/Master_boot_record) / [GPT](https://en.wikipedia.org/wiki/GUID_Partition_Table)  
  Выполняется код, заложенный производителем аппаратного обеспечения. Этот код проводит тестирование системы POST (Power On Self Test) и передает управление загрузчику в MBR (Master Boot Record) / GPT (GUID Partition Table)
* MBR / GPT → [GRUB2](https://en.wikipedia.org/wiki/GNU_GRUB#GRUB_2) (существуют и другие загрузчики)  
  Загрузчик из MBR / GPT очень простой – он способен только найти на диске и запустить следующий загрузчик. Как правило это GRUB2, но существуют и другие загрузчики, например [LILO](https://en.wikipedia.org/wiki/LILO_(boot_loader)) (в настоящее время практически не используется)
* GRUB2 → Kernel  
  GRUB2 расположен на разделе жесткого диска в каталоге /boot. GRUB2 загружает ядро Linux (vmlinuz)
* Kernel → Init  
  Ядро запускает процесс инициализации операционной системы. Как правило это SystemD, но существуют и другие системы инициализации, например [SystemV](https://en.wikipedia.org/wiki/UNIX_System_V) (в настоящее время практически не используется). Процесс инициализации запускает все остальные процессы в системе

### Подключение

#### **Командная оболочка**

Подключиться к Linux для управления можно в интерфейсе командной строки (command-line interface, CLI) или в графическом интерфейсе (graphical user interface, GUI). При работе с серверной инфраструктурой в подавляющем числе случаев GUI отсутствует и взаимодействие с сервером осуществляется в CLI. При входе пользователя на сервер в CLI запускается командная оболочка (в GUI командную оболочку можно запустить через эмулятор, например, Terminal). Командная оболочка (shell) – это программа, которая принимает команды с клавиатуры и передает их операционной системе для выполнения. Наиболее распространенной командной оболочкой в Linux является [GNU bash](https://www.gnu.org/software/bash/) (Bourne Again SHell). bash основывается на другой легковесной оболочке-предшественнике – [sh](https://en.wikipedia.org/wiki/Bourne_shell) (Bourne sh), созданной [Стефеном Борном](https://en.wikipedia.org/wiki/Stephen_R._Bourne).

Команды можно выполнять с помощью командной строки, указав имя двоичного (бинарного, bin) исполняемого файла или сценария. По умолчанию в Linux много команд, которые позволяют перемещаться по файловой системе, устанавливать ПО, конфигурировать его и выполнять другие действия. Каждая запущенная команда является отдельным процессом. Важно отметить, что в Linux (в отличие от Windows) почти всегда учитывается регистр, включая имена файлов и каталогов, команды, аргументы и опции.

#### **SSH**

Подключаться к Linux и работать с командной оболочкой можно локально (например, включив ПК дома или подойдя к серверу в центре обработки данных), однако гораздо чаще работать с системой требуется удаленно. Для этого необходимо настроить SSH и подключаться через него.

[SSH](https://en.wikipedia.org/wiki/SSH) (Secure SHell) – это протокол, позволяющий производить удаленное управление операционной системой и туннелирование TCP-соединений (например, для копирования файлов). SSH основан на клиент-серверной архитектуре, которая организует защищенное (зашифрованное) соединение поверх небезопасных каналов связи. Серверная часть устанавливается на удаленном сервере, а клиентская на компьютере, с которого осуществляется подключение.

Существует много клиентов для удаленного подключения к серверу по SSH. Вот несколько примеров для различных операционных систем:

* Windows

[PuTTY](https://www.putty.org/) – выбор новичка, PowerShell (команда ssh)

* Linux  
  Terminal (команда ssh) – выбор новичка.
* MacOS  
  Terminal (команда ssh) – выбор новичка.

### Установка программ (утилит) пакетным менеджером

#### **Зачем нужны пакетные менеджеры?**

На заре развития Linux установить приложение (утилиту) можно было только путем скачивания исходного кода программы и компиляции. Это не практично и не слишком удобно для пользователей, поэтому были разработаны [пакетные менеджеры](https://en.wikipedia.org/wiki/Package_manager). Установка приложений в них производится из пакетов – архивов с файлами скомпилированной программы. Большинство популярных дистрибутивов Linux содержат пакетные менеджеры, способные устанавливать любое программное обеспечение. Пакетные менеджеры имеют свой список репозиториев – серверов с базой пакетов. Во время установки алгоритм менеджера находит необходимый пакет в базе и производит автоматическое скачивание, установку и настройку.

Существует несколько форматов пакетов, однако наибольшее распространение получили .deb и .rpm. Рассмотрим операционные системы и менеджеры пакетов для данных форматов:

* DEB (.deb)  
  ОС – DEB-based, например Debian, Ubuntu, AstraLinux  
  Система управления пакетами – [DPKG](https://en.wikipedia.org/wiki/Dpkg) (работает только с локальными пакетами)  
  Пакетный менеджер – [apt](https://en.wikipedia.org/wiki/APT_(software))
* RPM (.rpm)  
  ОС – RPM-based, например RedHat Enterprise Linux, Fedora, CentOS  
  Система управления пакетами – [RPM](https://en.wikipedia.org/wiki/RPM_Package_Manager) (работает только с локальными пакетами)   
  Пакетный менеджер – [yum](https://en.wikipedia.org/wiki/Yum_(software)) (в последних дистрибутивах заменен на [dnf](https://en.wikipedia.org/wiki/DNF_(software)))

**Структура файловой системы и работа с файлами**

Типы файлов

Все объекты в Linux являются файлами. Существуют следующие типы файлов:

* Обычные файлы **-**Символьные и двоичные данные (текст, картинки, программы и др.)
* Каталог (директория) **d**Список ссылок на файлы или другие каталоги
* Символьные ссылки **l**Ссылки на другие файлы по имени
* Блочные устройства **b,**символьные устройства **c**Интерфейсы для взаимодействия с аппаратным обеспечением (диски, терминалы, клавиатуры, принтеры и др.). Когда происходит обращение к файлу устройства, ядро операционной системы передает запрос драйверу этого устройства
* Сокеты **s** и каналы **p**  
  Интерфейсы для взаимодействия процессов

Структура файловой системы

Структура файловой системы представляет собой дерево, корнем которой является каталог /.

Рассмотрим подробно структуру и назначение каталогов:

* /bin (binaries) – исполняемые файлы самых необходимых утилит. Может быть символьной ссылкой на /usr/bin
* /boot – файлы, необходимые для самого первого этапа загрузки – загрузки ядра (и обычно само ядро)
* /dev (devices) – блочные и символьные файлы устройств (диски, терминалы, клавиатуры, принтеры и др.)
* /etc (etcetera) – конфигурационные файлы системы и различных программ
* /home – домашние каталоги пользователей для хранения «личных» файлов
* /lib (libraries) – файлы библиотек (стандартных функций, необходимых многим программам), необходимых для работы утилит. Может быть символьной ссылкой на /usr/bin
* /mnt (mount) – каталог для подключения файловых систем (съемных носителей и др.)
* /opt (optional) – каталог для дополнительных программ (проприетарных драйверов, агентов мониторинга и др.)
* /proc (process) – файлы в оперативной памяти, в которых содержится информация о выполняемых в системе процессах
* /root – домашний каталог пользователя root
* /sbin (system binaries) – файлы системных утилит, необходимые для загрузки, резервного копирования и восстановления системы. Может быть символьной ссылкой на /usr/sbin
* /sys (system) – виртуальная файловая система sysfs, которая содержит информацию об аппаратном обеспечении (ЦПУ, ОЗУ, дисках, сетевых устройствах), драйверах, ядре системы и др.
* /tmp – каталог для временных файлов, обычно зачищается при каждой загрузке системы
* /usr – пользовательский каталог, который содержит каталоги исполняемых файлов и конфигурационных файлов
* /var (variable) – файлы, создаваемые или используемые различными программами (логи, очереди, идентификаторы процессов, БД и др.)

#### **Права доступа**

В Linux права доступа к файлам (в том числе к каталогам) задаются для трех видов пользователей – владельца, группы владельца и остальных. Также есть три типа доступа к файлу – чтение **r** (Read), запись **w** (Write) и исполнение **x** (eXecution), которые задаются для каждого из видов пользователей. Прочерк **-** означает отсутствие доступа.

Таким образом, права доступа к файлу выглядят следующим образом:

* права для владельца (u, user) – read, write, execution
* права для группы владельца (g, group) – read, write, execution
* права для остальных пользователей (o, other) – read, write, execution

Пример: rwx r-- --- означает, что у владельца есть права на все, у группы владельца доступ только на чтение, а у остальных, доступа нет. В двоичной системе счисления эти права выглядят как три группы цифр – 111 100 000, что равносильно трем цифрам 7 4 0 в восьмеричной и десятичной системах счисления.

**Порядок выполнения практической работы**

1. Практическая работа выполняется в ***отдельном*** создаваемом студентом ***документе***!
2. Титульный лист с указанием наименования организации, ФИО студента и номер группы
3. Выполнить задания, в отчете написать номер задания, добавить скриншот, описать результат.
4. Ответить на контрольные вопросы
5. Вывод

**Практическое задание**

Задание 1. Выполнить задачи в терминале Linux:

1. Посмотреть под каким пользователем в системе в данный момент вы работаете Linux.
2. Перейти в корневой каталог.
3. Посмотреть содержимое каталога.
4. Определить типы команд ls, cd.
5. Посмотреть из корневого каталога содержимое директории Home.
6. Перейти в директорию /Рабочий стол (Desktop).
7. Создать директорию /pt.
8. Создать 3 текстовых документа 1.txt 2.txt 3.txt.
9. Изменить или добавить содержимое текстового документа с помощью редакторов nano и vim.
10. Скопировать и переименовать директорию с файлами.
11. Заархивировать директорию.
12. Разархивировать директорию.
13. Посмотреть содержимое домашней директории и каталога usr одной командой.
14. Посмотреть содержимое каталога в обратном порядке.
15. Посмотреть права на директорию.
16. Посмотреть права на текстовые файлы.
17. Вывести краткое описание команды и утилиты ls, nano, rm.

**Контрольные вопросы**

* 1. Для чего стоит учится пользоваться терминалом в Linux?
  2. Что такое CLI?
  3. Что такое SSH?
  4. Зачем нужны пакетные менеджеры?
  5. Из чего состоит структура файловой системы Linux?