1. Назовите первый компьютер общего назначения и ответьте на вопрос нужна ли была ему операционная система. Что означает «компьютер общего назначения» и как подобный компьютер программировался?

Компьютер общего назначения – это компьютер, который позволяет перепрограммировать себя под разные задачи. ЭНИАК не требовалась операционная система как таковая. Его перепрограммирование заключалось в том, чтобы человек вручную манипулировал физическими переключателями в нужное время и в нужном месте, после чего можно было изучить результат недавно заданной программы. Подобного подхода было достаточно, т.к. главной целью компьютера было точное вычисление.

1. Для решения какой проблемы компьютер научили разделять время между множеством пользователей. Что такое переключение контекста?

С момента создания транзистора компьютеры начали работать на полупроводниках, что дало ощутимый прирост производительности. Это позволило компьютерам работать сразу с несколькими пользователями за одну единицу времени. Но, увы, возникла новая проблема, связанная с задержкой ввода информации каждым пользователем в группе. Т.е. по сути пока один из пользователей с помощью устройства ввода вбивал свою программу в память компьютера, само устройство для вычислений в это время простаивало. Для решения этой проблемы компьютер начал разделять время между множеством пользователей, т.е. он выделял часть своего времени на одного пользователя, чтобы тот смог произвести ввод данных программы. После этого машина сохраняла состояние в специальные разделы памяти, дабы перейти к следующему пользователю, чтобы предоставить ему те же самые возможности. Как итог, компьютер переключал состояния пользователей между собой, дабы сократить время ввода информации. В современном мире это называется переключением контекста.

1. Назовите причины создания операционной системы. Какая операционная система с разделением времени исполнения программ была создана первой?

Время – самый ценный компьютерный ресурс.

Начиная с момента введения переключения контекста, появляется необходимость в создании специальной программы, которая отвечала бы за: разделение времени, контроль оборудования и за загрузку новых программ в область памяти. Этой программой и станет Операционная система, первой из которых станет Multics.

1. Опишите функции операционной системы и приведите для них примеры.

* Исполнение запросов программ (по доступу к данным, запуску/остановке других программ и выделению необходимой памяти);
* Загрузка программ в оперативную память для обеспечения их исполнения;
* Стандартизированный доступ к периферийным устройствам (устройства ввода/вывода);
* Управление объединенным банком оперативной памяти (распределение памяти между процессами);
* Управление доступом к данным на энергонезависимых носителях (HDD, SSD и т.д.), в рамках той или иной файловой системы;
* Обеспечение пользовательского интерфейса;
* Сохранение информации об ошибках системы.

1. Что такое ядро операционной системы и для каких целей оно служит? Можно ли утверждать, что ядро операционной системы – это и есть сама операционная система?

Ядро ОС – это её центральная часть, обеспечивающая приложениям координированный доступ к ресурсам компьютера, таким как процессорное время, внешнее аппаратное обеспечение, внешнее устройство ввода и вывода информации. Т.е. в современных ОС, которые используют ядро, приложения не могут обратиться к ресурсам компьютера напрямую.

1. Перечислите архитектуры ядер ОС и приведите пример ОС, которые их используют. Опишите особенности каждой и назовите чем в основе своей они отличаются друг от друга?

В основном архитектура ядра ОС отличается подходом расположения драйверов системы относительно пространства самого ядра. Чем ближе программы находятся к оборудованию, тем быстрее они работают, но в то же время, чем дальше они находятся от оборудования, тем стабильнее вся система.

* Монолитное ядро (Monolithic kernel);
* Микроядро (Microkernel);
* Гибридное ядро (Hybrid kernel).

Архитектура монолитного ядра является самой производительной, т.к. драйвера системы работают в пространстве ядра. Причина ускорения такой архитектуры является и причиной её недостатка. Если произойдет сбой в драйвере – произойдет сбой во всей системе. Данный подход понижает отказоустойчивость, но повышает быстродействие.

В случае архитектуры микроядра, драйверы расположены обособленно от ядра и работают в своем собственном, независимом пространстве. В данном случае ядро производит переключение из одного пространства в другое.

Данный подход повышает отказоустойчивость, но понижает быстродействие (т.к. процессор постоянно производит переключение контекста между различными программами).

Отличительная черта гибридного ядра заключается в том, что для ускорения быстродействия некоторая необходимая служба может спуститься в пространство ядра, что может существенно ускорить быстродействие исполнения подобной службы, но при этом не сильно понизить отказоустойчивость в продолжительной перспективе.

1. Что такое кольца защиты? Приведите примеры колец защиты и скажите для чего они нужны.

Кольца защиты необходимы для того, чтобы разграничивать ресурсы компьютер между приложениями.

Современные ОС используют 2 кольца защиты:

* Пространство ядра;
* Пространство пользователя.

Во время переключения с одного кольца защиты на другое процессор не производит никаких вычислений, т.е. в нашем понимании он простаивает (т.к. не выполняет наших команд). Поэтому в современных ОС все программы работают в пространстве пользователя, дабы постоянно не тратить время между переключениями режимов, а если возникает необходимость произвести некоторые изменения в драйверах или в самом ядре – только тогда система переключится на кольцо защиты пространства ядра.

1. Кем было создано ядро Linux и какую операционную систему он взял за основу? Существует ли официальная версия ОС Linux?

Ядро Linux было создано в 1991г. Линусом Торвальдсом на основе учебной ОС Minix, версия которой шла вместе с книгой Эндрю Таненбаума (автор Minix).

Официальной версии ОС Linux не существует. Изначально Linux планировалась как конструктор, который мы могли бы использовать для сборки своей версии ОС на базе имеющихся программного обеспечения и ядра.

1. Кто такой Ричард Столлман и чем являлся его проект «GNU»? Как его проект повлиял на появление операционной системы GNU / Linux?

Ядро операционной системы – это часть ОС, которая отвечает за взаимодействие с оборудованием и предоставляет интерфейс (POSIX), с которым начинает общаться программное обеспечение.

Ричард Столлман – создатель проекта GNU (GNU is Not UNIX). Изначально данный проект должен был стать свободной UNIX-подобной ОС, но по итогу было создано лишь окружение операционной системы, а ядро до сих пор не готово.

1. Как полное название носит операционная система Linux? Может ли абсолютно любой человек в мире осуществить изучение и доработку данной ОС?

После объединения ядра Linux с проектом GNU (окружением ОС) получилась полноценная операционная система GNU/Linux, которая является свободным программным обеспечением, распространяемым по лицензии GPL, что позволяет абсолютно любом человеку осуществлять изучение и доработку данного программного обеспечения.

1. Что такое дистрибутив и чем они отличаются друг от друга? Возможно ли создать свой собственный дистрибутив ОС Linux?

Операционная система GNU/Linux – это ядро и набор программ для решения определенного круга задач.

Соответственно, можно собрать своё собственное минималистичное ядро, которое затем можно будет использовать в роутере (без поддержки видеоадаптера и т.д.) или же изменить набор программ, которые комплектно будут предустанавливаться вместе с операционной системой.

1. Что такое TTY и PTS, а также сопоставьте им терминал и консоль. Чем отличается терминал от консоли, а также ответьте каким образом TTY является сокращением от слова Teletypewriter?

TTY (TeleTYpe) и PTS (Pseudo-Terminal Slave) - это специальные устройства, которые представляют собой интерфейс между пользователем и ядром операционной системы Unix-подобных систем. TTY используется для подключения физического устройства, такого как телетайп, терминал или консоль, к системе, тогда как PTS используется для подключения приложения к терминалу.

Консоль - это устройство, позволяющее пользователю взаимодействовать с операционной системой напрямую, без использования графического интерфейса. Консоль реализуется через TTY и обычно предлагает текстовый интерфейс. Терминал же - это графический интерфейс, который позволяет пользователю работать с командной строкой в графическом режиме.

Отличие терминала от консоли заключается в том, что терминал - это графический интерфейс для работы с командной строкой, в то время как консоль - это текстовый интерфейс. Консоль предоставляет доступ к командной строке непосредственно из операционной системы, без использования графического интерфейса, в то время как терминал обычно является приложением, которое может быть запущено из графического интерфейса.

TTY назван так в честь телетайпов, которые использовались для ввода команд в Unix-подобных операционных системах в прошлом. В настоящее время TTY относится к устройствам, подключенным к системе посредством последовательных интерфейсов, таких как COM-порты или USB-порты, и не имеет отношения к телетайпам, но название TTY все еще используется для обозначения соответствующих средств связи в Unix-системах.

1. Из чего состоит терминал и консоль Linux и для каждой составляющей приведите пример.

Терминал и консоль Linux - это интерфейс командной строки, который предоставляет пользователю доступ к операционной системе Linux для ввода команд и управления системой. Вот составляющие терминала и консоли Linux и примеры:

* Оболочка командной строки (shell) — это программное обеспечение, которое интерпретирует команды, вводимые пользователем, и выполняет их на уровне операционной системы. Примеры таких оболочек в Linux: Bash, Zsh, Ksh.
* Терминал — это графический интерфейс, который позволяет пользователю работать с командной строкой. В терминале могут быть использованы различные шрифты, цветовые схемы и другие опции настройки, которые не доступны из консоли. Примеры терминалов для Linux: GNOME Terminal, Konsole, Xfce Terminal.
* Консоль — это текстовый интерфейс, который предоставляет пользователю доступ к командной строке непосредственно из операционной системы. Консоль работает на уровне ядра Linux и не использует графический интерфейс. Примеры консолей,

1. Что в командной строке означает «user@hostname:~$»? Приведите пример нескольких командных оболочек и расскажите, чем они друг от друга отличаются.

* user – имя текущего пользователя;
* hostname – имя хоста;
* ~ – текущий каталог (в bash символ тильды соответствует домашнему каталогу, путь к нему – /home/user).
* $ – символ, который означает конец командной строки, после которого появится ввод пользователя с клавиатуры.

В ОС Linux существует несколько командных оболочек, которые могут использоваться пользователем. Наиболее распространенными являются:

1. Bash (Bourne-Again SHell) - это наиболее распространенная командная оболочка в Linux. Bash является стандартной оболочкой в большинстве дистрибутивов Linux и поддерживает множество функций, таких как автодополнение, история команд и др.

2. Zsh (Z Shell) - это командная оболочка, которая предоставляет расширенные функции автодополнения и поддерживает множество плагинов. Zsh также обеспечивает более продвинутый механизм истории команд и поддерживает работу с различными кодировками символов.

3. Fish (Friendly Interactive SHell) - это командная оболочка, которая предоставляет более удобный интерфейс для работы с командами. Fish обеспечивает автодополнение, подсветку синтаксиса и подсказки по командам.

Каждая командная оболочка имеет свои особенности и функции. Например, Bash предоставляет более продвинутый механизм истории команд и поддерживает работу с переменными среды, а Zsh обеспечивает более продвинутый механизм автодополнения и поддерживает работу с различными кодировками символов. Fish, в свою очередь, предоставляет более удобный интерфейс для работы с командами. Каждый пользователь может выбрать ту командную оболочку, которая наиболее подходит для его задач и предпочтений.

1. В чем разница между командами «sudo» и «sudo su». Для каких ситуация используется каждая из них?

Есть несколько способов временно под некоторую задачу расширить свои права пользователя в системе.

Sudo - (Super User Do). Позволяет исполнять команды с правами суперпользователя.

Перед переходом в режим root администратора система выполнит запрос его пароля.

Sudo su - Инструмент применяется в ситуации, когда необходимо использовать расширенные права доступа длительное время. Все следующие вводимые команды будут запускаться в режиме суперпользователя до закрытия командной строки.

1. Что такое управляющие операторы и приведите их примеры. Опишите как использовать каждый из приведенных вами.

Управляющие операторы - это конструкции, которые используются в командной строке для управления выполнением команд в операционной системе Linux. Некоторые примеры управляющих операторов в Linux:

1. & - запускает команду в фоновом режиме, что позволяет продолжать работу в терминале.

2. ; - позволяет запускать несколько команд последовательно в одной строке.

3. && - выполняет следующую команду только в том случае, если предыдущая команда выполнена успешно.

4. || - выполняет следующую команду только в том случае, если предыдущая команда выполнена неудачно.

1. В чем разница между абсолютным и относительным путем, а также назовите команду, которая выведет ваше текущее местоположении в файловой системе.

Абсолютный путь - это полный путь к файлу или директории от корневой директории. Например, /home/user/Documents/file.txt - это абсолютный путь к файлу file.txt, который находится в директории Documents, которая находится в директории пользователя user, который находится в директории home, которая является корневой директорией.

Относительный путь - это путь к файлу или директории относительно текущего местоположения в файловой системе. Например, если текущая директория /home/user/Downloads, то путь к файлу file.txt в директории Documents, которая находится в директории пользователя user, будет выглядеть как ../../Documents/file.txt.

Команда, которая выведет текущее местоположение в файловой системе, называется pwd (print working directory). Команда pwd выводит абсолютный путь к текущей директории, в которой находится пользователь.

1. Что такое символ перенаправления и существует ли специальная команда в Linux для создания пустого файла?

Символ перенаправления - это символы '>', '<', '>>', которые используются в командной строке Linux для перенаправления ввода и вывода.

Символ '>' используется для перенаправления вывода команды в файл. Например, команда ls > file.txt выведет список файлов и директорий в текущей директории и перенаправит вывод в файл file.txt. Если файл file.txt уже существует, то его содержимое будет перезаписано.

Символ '<' используется для перенаправления ввода команды из файла. Например, команда cat < file.txt выведет содержимое файла file.txt.

Символ '>>' используется для добавления вывода команды в конец файла. Если файл не существует, то он будет создан. Например, команда echo "Hello world" >> file.txt добавит строку "Hello world" в конец файла file.txt.

Существует специальная команда в Linux для создания пустого файла, она называется touch. Команда touch используется для создания пустого файла или для обновления времени последнего доступа и изменения файла. Например, команда touch file.txt создаст пустой файл с именем file.txt в текущей директории.

1. Приведите пример консольного текстового редактора и опишите как им пользоваться.

Nano - это консольный текстовый редактор, который выполняется через терминал. Для завершения работы с ним необходимо нажать сочетание клавиш Ctrl+X, затем нажать «Y», если желаем сохранить изменения и «N», если не желаем, после чего нажимаем «Enter» на клавиатуре. Горячие клавиши для быстрых действий с данным текстовым редактором описаны в нижней его части.

1. Чем для системы является имя пользователя и важно ли оно ей? Каким образом система идентифицирует пользователь в системе и что именно определяет данный параметр для конкретного пользователя?

Для Linux имя пользователя является лишь псевдонимом, который необходим нам, а не системе. Система же идентифицирует пользователя по уникальному параметру UID (User Identifier), который как правило задается автоматически при создании.

Данный параметр определяет конкретного пользователя, а также права, которыми он обладает в системе.

1. На какие три типа делятся все пользователи в ОС Linux и где их всех найти? Для чего нужны системные пользователи?

root (от англ. root - корень) - суперпользователь, аккаунт в UNIX-подобных системах, владелец которого имеет право на выполнение всех операций без исключения. Присутствует в системе по умолчанию. UID = 0

Системные пользователи - системные процессы, у которых есть учетные записи для управления привилегиями и правами доступа к файлам и каталогам. Создаются системой автоматически. UID от 100 до 999

Обычные пользователи - учетные записи пользователей, допущенных к управлению системой. Создаются системным администратором. UID от 1000 до 60000.

1. Что такое «Daemon» в системах класса UNIX? Каким образом они конфигурируются?

Daemon – компьютерная программа в системах класса UNIX, запускаемая самой системой и работающая в фоновом режиме без прямого взаимодействия с пользователем, которая ожидает некоторого события, на которое ей будет необходимо соответствующе отреагировать (к примеру сетевые службы: ssh сервер, ftp сервер, dns сервер и прочие).

1. Какую роль для пользователей выполняют группы и каким параметром идентифицируется каждая группа? Где найти все группы в ОС Linux?

Группы в ОС Linux выполняют роль организации пользователей с общими правами доступа к файлам и ресурсам. Каждая группа идентифицируется уникальным идентификатором (GID). Все группы можно найти в файле /etc/group или командой getent group.

1. Где найти пароли всех пользователей и в каком виде они там хранятся? Что такое аутентификация и авторизация?

Пароли всех пользователей в ОС Linux хранятся в файле /etc/shadow. Однако, этот файл доступен только для чтения и записи суперпользователю (root). Пароли хранятся в зашифрованном виде с использованием функции хеширования. Обычно, пароль хранится в виде строки, начинающейся с знака "$", за которым следует обозначение используемой функции хеширования и соответствующие параметры.

Аутентификация – проверка наличия пользователя в системе.

Авторизация – проверка наличия прав для доступа к ресурсам системы.

Перед тем, как пользователь войдет на сервер, система произведет аутентификацию (проверит пользователя на существование в системе). Если пара логин и пароль прошла проверку, система произведет авторизацию (проверит обладает ли данный пользователем правом доступа к системе).

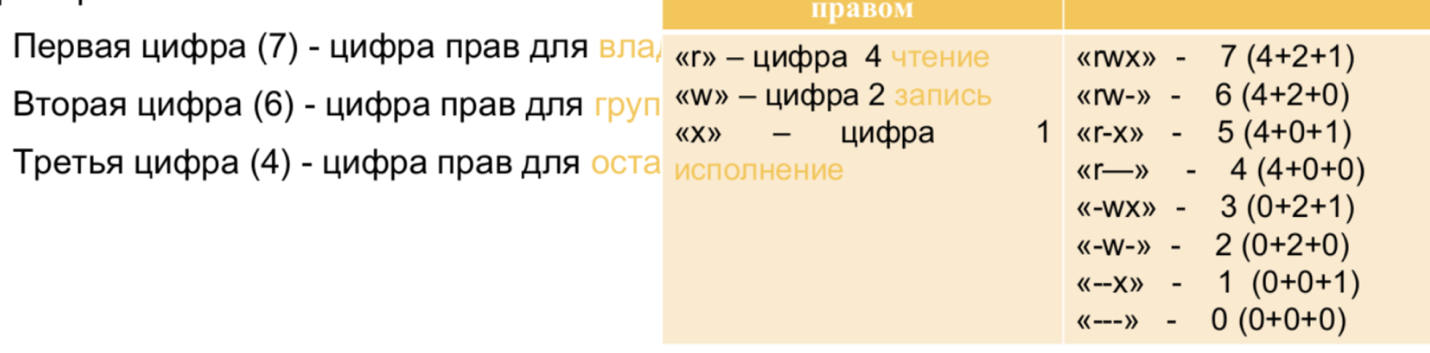
1. Какие три полномочия регулируют всю систему разграничения прав доступа в ОС Linux? Как узнать права у файла, а также как данные права влияют на файл и на каталог?

Поскольку в Linux все есть файл, распределение прав доступа возложено на три полномочия:

* Чтение
* Запись
* Выполнение

Чтобы узнать права доступа к файлу или каталогу в ОС Linux, нужно использовать команду ls -l. Она покажет список файлов и каталогов в текущей директории с указанием их прав доступа, владельца, группы и другой информации.

* u – Владелец (User)
* g – Группа (Group)
* o - Все остальные, кроме владельца и группы (Other)
* a - Владелец и группа владельцев и все остальные вместе взятые (All)
* «+» - Добавляет к текущим правам доступа новое разрешение.
* «-» - Удаляет из текущих прав доступа определенное разрешение.
* «=» - Устанавливает полностью новые разрешения (предыдущие перезаписываются новыми).



1. Каким трем категориям, которые могут взаимодействовать с файлами, можно выдать права доступа на конкретный файл? Какой утилитой можно модифицировать права доступа на файл?

Трем категориям, которые могут взаимодействовать с файлами, можно выдать права доступа на конкретный файл: владельцу файла, группе, которой принадлежит файл, и всем остальным пользователям.

Утилитой для модификации прав доступа на файл является команда chmod. Она позволяет изменять права доступа для каждой из трех категорий пользователей (владелец, группа, все остальные) на чтение (r), запись (w) и выполнение (x) файла. Кроме того, можно использовать символическое представление прав доступа, например, u+r для добавления права чтения владельцу файла.

1. В чем разница между восьмеричным и символьным подходами для разграничения прав доступа на файлы, опишите восьмеричную систему?

Восьмеричная система используется для представления прав доступа на файлы в виде числа, которое состоит из трех цифр. Каждая цифра соответствует правам доступа для каждой из трех категорий пользователей (владелец, группа, все остальные), где первая цифра отвечает за права доступа владельца файла, вторая - за права доступа группы, а третья - за права доступа для всех остальных пользователей.

Каждая цифра восьмеричного числа представляет собой сумму значений прав доступа: 4 для права чтения, 2 для права записи и 1 для права выполнения. Например, число 754 означает, что владелец файла имеет права чтения, записи и выполнения (4+2+1=7), группа имеет права чтения и выполнения (4+1=5), а все остальные пользователи имеют только право на выполнение (1).

Символьный подход использует команду chmod и символы для изменения прав доступа на файлы. Например, u+r означает добавление права чтения владельцу файла, а o-x означает удаление права выполнения для всех остальных пользователей. Разница между восьмеричным и символьным подходами заключается в представлении прав доступа на файлы - числами или символами.

1. Каким образом узнать какие сетевые адаптеры существуют в вашей ОС Linux, а также как узнать какие у них IP-адреса? Что такое «lo» и «enp0s3» интерфейсы?

Для того, чтобы узнать IP-адрес системы, необходимо использовать команду «ip a».

lo (loopback) – имеет адрес 127.0.0.1/8, который сопоставлен имени хоста localhost – это специальный виртуальный сетевой интерфейс , который компьютер использует для связи с самим собой (к примеру если Ethernet или Wi-Fi отключены). Используется в основном для диагностики и устранения неполадок, а также для подключения к серверу, работающему на локальном компьютере.

enp0s3 – имеет адрес 192.168.1.87/24 (ip варьируется от системы к системе). Это сетевое периферийное устройство Ethernet.

1. Что такое шлюз и DNS, и как их узнать? В чем разница между сетевым мостом и общей сетью?

Шлюз (gateway) – это маршрутизатор (если он есть) или сервер провайдера (если маршрутизатора нет). Он необходим в ситуации, если пакет, в качестве адреса назначения имеет адрес, не принадлежащий к нашей локальной сети, то он отправляется в шлюз (маршрутизатор). Именно маршрутизатор позволяет общаться нескольким разным сетям между собой. «ip r» - узнать IP-fдрес маршрутизатора.

DNS сервер (Domain name server) – сервер, который позволяет связать ip-адрес сервера (77.88.55.55) с именем домена (Yandex.ru). В сети интернет все обращение основано на ip адресах, которые проблематично использовать пользователям. Для удобства использования сети интернет существуют DNS сервера, которые позволяют найти ip-адрес сайта по вводимому пользователем имени домена, после чего происходит адресация и сайт открывается. «resolvectl status» - узнать днс.

Разница между сетевым мостом и общей сетью заключается в том, что первый соединяет две или более отдельные сети на физическом уровне, а вторая позволяет нескольким компьютерам обмениваться данными и ресурсами в рамках одной сети.

1. Что такое порт? Как узнать список сетевых служб и портов, а также портов, которые прослушиваются в системе в данным момент времени?

Порт – числовой идентификатор программы или процесса, который позволяет определить получателя на заданном IP-адресе.

Файл /etc/services содержит в себе список сетевых служб, а также порты и сетевые протоколы, которые они используют.

С помощью команды «sudo ss -tnlp» можно посмотреть прослушиваемые порты программами, которые сейчас активны.

1. Что такое netplan, какой язык использует и каким особенностями обладает данный язык?

Netplan - это инструмент для настройки сети в операционных системах на базе Linux, таких как Ubuntu, Debian и другие. Он использует язык YAML (YAML Ain't Markup Language) для конфигурации сетевых интерфейсов и связанных с ними параметров.

Особенностью языка YAML является его удобочитаемость и простота синтаксиса. Он использует отступы для определения блоков кода, что делает его более понятным для человека. Кроме того, YAML поддерживает комментарии, что позволяет добавлять пояснения к конфигурационному файлу.

1. Что такое SSH и для каких целей его можно использовать? Какие криптографические техники используются при соединении по SSH, опишите каждую из них.

SSH (Secure SHell - защищенная оболочка) — сетевой протокол прикладного уровня, предназначенный для безопасного удаленного доступа к UNIX-системам (к примеру Linux). Данный протокол эффективен тем, что шифрует всю передаваемую информацию по сети. По умолчанию, используется 22-й порт. В основном он нужен для удаленного управления данными пользователя на сервере, запуска служебных команд, работы в консольном режиме с базами данных.

КРИПТОГРАФИЧЕСКИЕ ТЕХНИКИ:

Симметричное шифрование использует один единственный секретный ключ для шифрования и дешифровки сообщения как клиентом (компьютером), так и хостом (сервером).

Симметричные ключи используются в SSH для шифрования всего соединения, что позволяет защитить парольную аутентификацию от отслеживания.

Асимметричное шифрование использует два отдельных ключа для шифрования и дешифровки. Эти два ключа также известны как приватный (красный на схеме, хранится зашифрованным у пользователя) и публичный ключи (синий на схеме, хранится на сервере в открытом виде). Вместе они формируют пару публичных-приватных ключей.

Асимметричные пары ключей используются только для аутентификации, а не для шифрования соединения.

Хеширование не предназначено для дешифровки. Главный принцип подобного способа заключается в том, чтобы создать уникальное значение фиксированной длинны для каждого ввода, которое невозможно будет преобразовать обратно в понятную нам формулировку.

Хеширование в основном используются для подтверждения целостности данных и проверки подлинности связи. Основное использование хэширования в SSH – это хэш-коды аутентификации сообщений. Они позволяют подтвердить, что текст принятого сообщения не был изменен или поврежден.

1. Перечислите принципы из философии UNIX, которые вобрала в себя ОС Linux.

* Файловая система древовидная – в Linux вся файловая система начинается с одного каталога root (/) и в сравнении с Windows, в Linux отсутствуют диски (Локальный диск (C:));
* Широкое использование текстовых файлов для хранения настроек системы, что позволяет её конфигурировать;
* «Все есть файлы» - в Unix все представляется в виде файла (включая каталоги), а само понятие «Устройство» отсутствует. Есть файл, который связан с некоторым устройством, будь то Жесткий диск, порты или Ethernet карта и благодаря данному файлу мы сможем произвести конфигурирование. Главное преимущество данного подхода – отсутствие необходимости в написании отдельных интерфейсов для отдельных устройств, т.к. с этим могут справиться стандартные утилиты Linux;
* Файлы не нуждаются в расширении имен файлов (.txt), вместо этого система понимает, что это за файл проверяя первые несколько байт файла.

1. Что такое файловая система (ФС) и какая ФС является самой популярной для большинства дистрибутивов Linux? В чем её особенность?

Файловая система (ФС) - это способ организации и хранения файлов на жестком диске или других устройствах хранения данных. Она определяет, как данные будут записываться, читаться и удаляться.

Самой популярной файловой системой для большинства дистрибутивов Linux является ext4 (четвертое расширенное поколение файловой системы ext). Её особенностью является высокая производительность, надежность и поддержка больших объёмов файлов. Она также поддерживает функции журналирования, которые позволяют быстро восстановить данные в случае сбоя системы или отключения питания.

1. Чему равен классический блок в файловой системе ext4 и что будет, если информация, записываемая в него, будет меньше или больше его максимального объема? Можно ли варьировать объем одного блока и к каким последствиям это приведет?

Классический блок в файловой системе ext4 равен 4 килобайтам. Если информация, записываемая в блок, будет меньше его максимального объема, то оставшееся место в блоке останется неиспользованным. Если же информации будет больше, чем максимальный объем блока, то она будет разбита на несколько блоков.

Объем одного блока можно варьировать, но это может привести к неэффективному использованию пространства на диске. Например, если увеличить размер блока, то файлы, которые меньше нового размера блока, будут занимать больше места на диске из-за неиспользования оставшейся части блока. Кроме того, изменение размера блока может повлиять на производительность файловой системы и время доступа к файлам. Поэтому рекомендуется использовать стандартный размер блока для конкретной файловой системы.

1. Что такое inode и какой целью служит в файловой системе (ФС) ext4? Могут ли закончиться inode в ФС до заполнения полного объема диска и если да, то что в такое ситуации делать?

Inode (index nodes – индексный узел) содержит метаданные о файле, т.е. некоторую дополнительную информацию о нем, к примеру: размер, дату создания, права доступа, тип файла и владельца/группу. Inode позволяет зарезервировать на диске место за файлом, дав понять файловой системе, что заполненные только что блоки использовать другими данными нельзя.

Да, инод может закончиться, в такой ситуации можно попробовать увеличить количество доступных inode, используя специальные инструменты или параметры при создании файловой системы. Однако, если это невозможно, то единственным решением будет создание новой файловой системы с большим количеством доступных inode и перемещение данных на нее. Поэтому при проектировании и настройке файловых систем необходимо учитывать возможное количество inode и выбирать соответствующие параметры.

1. Чем является каталог в файловой системе (ФС) ext4 и какую роль он играет для всей ФС? каталог является специальным типом файлов, но какова его функция?

Функцией каталога выступает хранение в себе имен файлов или других каталогов, а сама каталог – представляет собой, по сути, соответствие имени файла с его inode. Директория является особым видом файла, который содержит таблицу соответствия между именами файлов и структурами inode (Командой «ls -ali» можно увидеть данную таблицу).

1. Чем выступает имя файла для файловой системы ext4? Как реализовано удаления файла с устройства и причем тут каталог?

Имя файла – это идентификатор, по которому можно найти inode данного файла. Удаление файла реализовано в виде удаления его имени из каталога, т.е. сами данные из памяти не стираются, но его запись из каталога исчезает. Данный подход был реализован с целью быстрого удаления информации.

1. Что такое HardLink и SoftLink? Опишите их особенности и скажите какая между ними разница.

HardLink (жесткая ссылка) представляют собой дополнительную запись в файловой таблице раздела для существующего файла. Таким образом жесткая ссылка ничем не отличается от исходного файла, но файл при этом не копируется физически и остается на диске в единственном экземпляре.

SoftLink (символическая ссылка) – это файл (с уникальной inode), который содержит адрес на другой файл, для которого создавалась ссылка.

1. Опишите особенность структуры каталогов в Linux и назовите чем она отличается от структуры каталогов в Windows.

Структура каталогов в Linux полностью подчиняется логике, задуманной разработчиками данной ОС. В отличие от Windows, операционная система Linux не станет спрашивать у пользователя, где будут размещены установленные программы, где хранить временные файлы, какие файлы будут являться конфигурационными и т.д.

Linux самостоятельно размещает все файлы по каталогом в зависимости от их принадлежности, руководствуясь собственной структурой каталогов.

1. Опишите каждый вариант установки программного обеспечения в ОС Linux. Назовите для каждого свои плюсы и минусы.

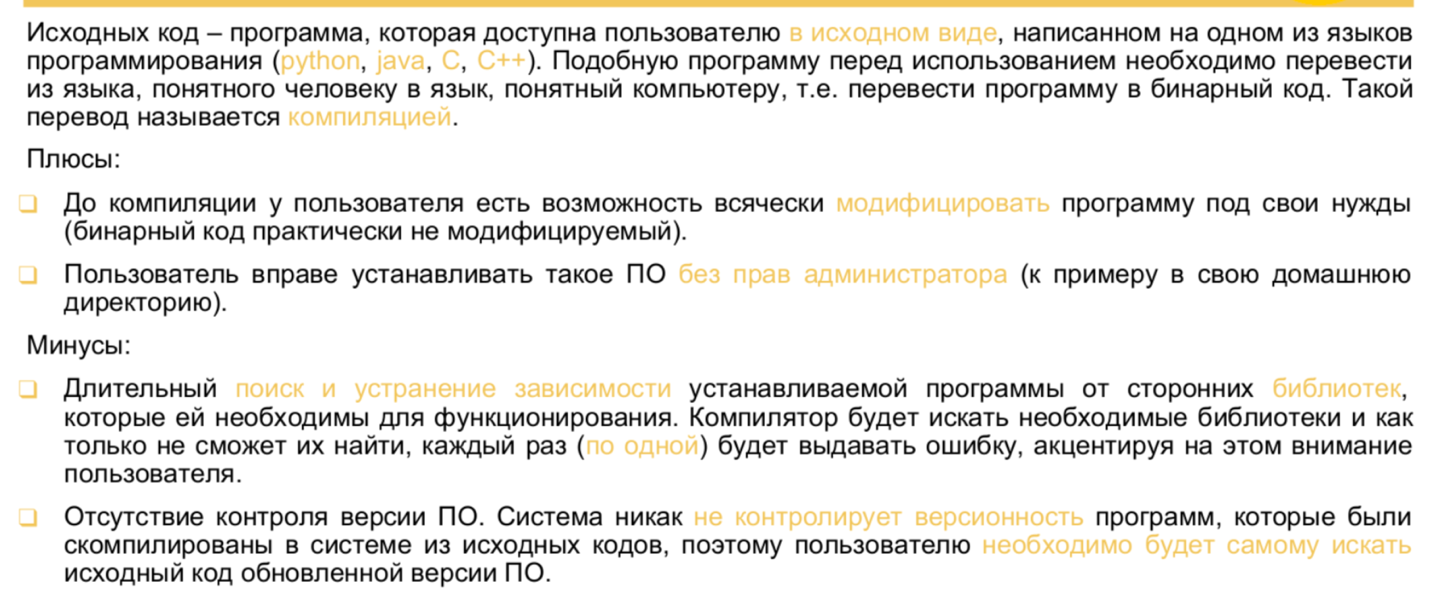


Рисунок 1 ИСХОДНЫЙ КОД

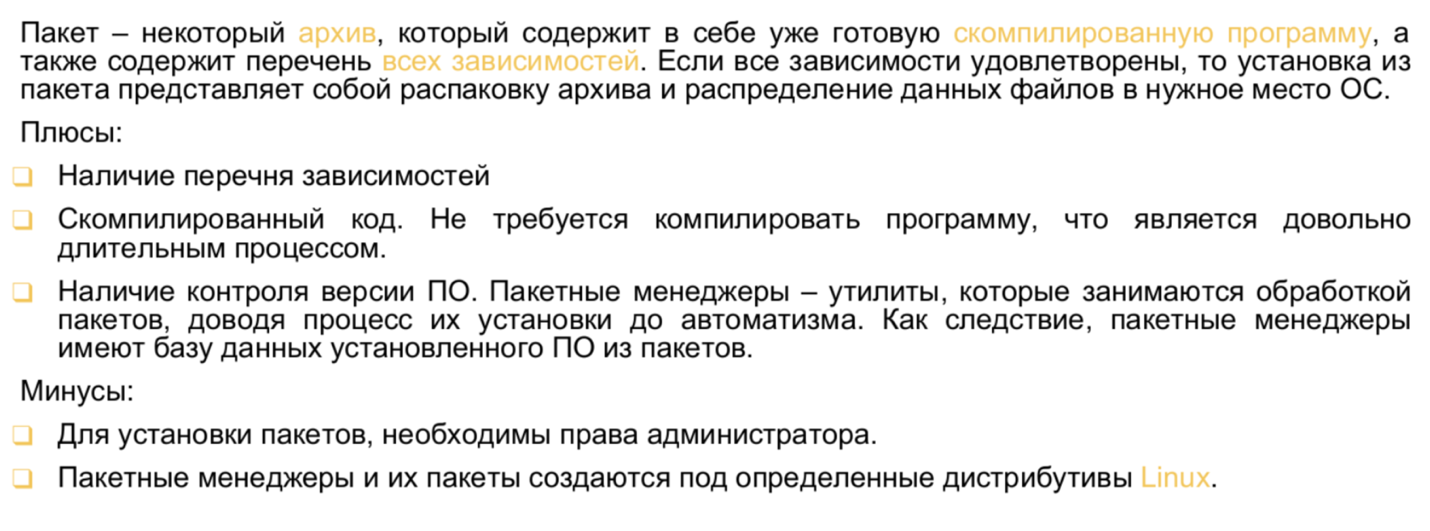


Рисунок 2 ПАКЕТ

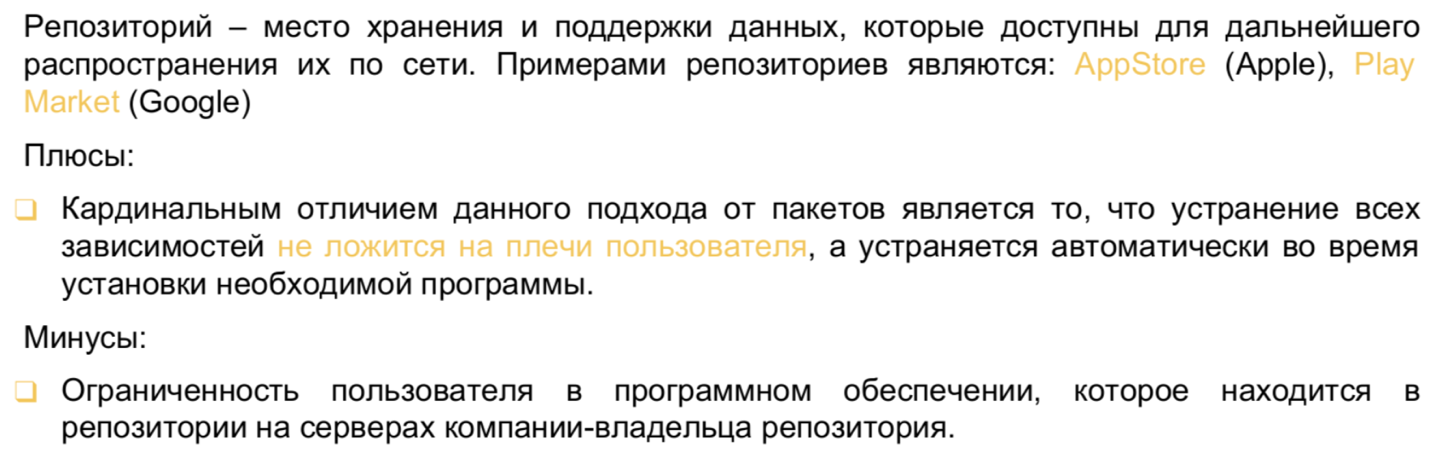


Рисунок 3 РЕПОЗИТОРИЙ

1. Что такое пакеты и пакетные менеджеры? Почему существуют низкоуровневые и высокоуровневые пакетные менеджеры? На примере дистрибутива Ubuntu приведите их примеры, а также назовите формат пакетов, который используется в данном дистрибутиве.

Пакеты - это архивы, содержащие программное обеспечение, необходимое для установки и работы приложений в операционной системе. Пакетные менеджеры - это программы, которые управляют установкой, обновлением и удалением пакетов в операционной системе.

Низкоуровневые пакетные менеджеры работают непосредственно с файлами пакетов, выполняя их установку и удаление. Высокоуровневые пакетные менеджеры работают на более высоком уровне абстракции, предоставляя пользователю удобный интерфейс для управления пакетами.

В дистрибутиве Ubuntu примером низкоуровневого пакетного менеджера является dpkg, который работает с пакетами в формате .deb. Примером высокоуровневого пакетного менеджера является apt, который предоставляет пользователю команды для поиска, установки и удаления пакетов, а также автоматическую установку зависимостей.

Формат пакетов, используемый в Ubuntu - это формат .deb. Он содержит исполняемые файлы, библиотеки, конфигурационные файлы и другие ресурсы, необходимые для работы приложения.

1. Для чего в Apt первым делом используют команду «update» и только затем команду «upgrade»?

APT (Advanced Packaging Tool) – консольная утилита, выполняющая роль «поисковика» и загрузчика пакетов из репозиториев.

Команда «sudo apt update» позволяет загрузить последнюю обновленную информацию о метапакетах ПО, которая содержит: названия, версии и т.д., тем самым обновляя информацию о списке пакетов из удаленных репозиториев.

После загрузки обновленной информации можно воспользоваться командой «sudo apt list --upgradable» с целью вывода списка пакетов, для которых доступны обновления.

Чтобы убедиться, что информация о доступных обновлениях актуальна, необходимо сначала выполнить команду "update", а затем уже выполнять команду "upgrade".

1. Для чего нужны VFS и драйвер файловой системы? Что такое монтирование и как оно связано с VFS?

VFS (Virtual File System) - это абстрактный слой, который предоставляет единый интерфейс для работы с файловыми системами различных типов. Он позволяет приложениям работать с файлами и каталогами, не зависимо от того, какая файловая система используется на диске.

Драйвер файловой системы - это программный модуль, который позволяет операционной системе работать с определенной файловой системой. Он обеспечивает интерфейс между файловой системой и ядром операционной системы.

Монтирование - это процесс подключения файловой системы к определенной точке монтирования в файловой системе операционной системы. В процессе монтирования, VFS использует драйвер файловой системы для доступа к файлам и каталогам, которые находятся на диске.

Таким образом, VFS и драйвер файловой системы необходимы для того, чтобы операционная система могла работать с файлами и каталогами, независимо от типа используемой файловой системы. Монтирование связано с VFS, потому что оно использует этот слой для доступа к файлам и каталогам при монтировании файловой системы.

1. С какой целью и из какого конфигурационного файла берет информацию система инициализации?

Система инициализации берет информацию из конфигурационного файла /etc/inittab с целью определения процессов, которые должны быть запущены при загрузке операционной системы и управления ими во время работы системы.

1. Дайте определение bat-файлам и объясните, как их создать. Какие команды можно использовать внутри bat-файлов?

Для того, чтобы создать bat-файл, достаточно создать файл блокнота и изменить ему расширение с «.txt» на «.bat» или перейти в приложение «блокнот (англ. Notepad)» и после нажатия «Сохранить как» изменить расширение на «.bat», а ниже выбрать пункт «Все файлы \*». Также подобный пакетный файл можно создать с помощью CMD.

* 1. echo - выводит текст на экран
* 2. set - устанавливает значение переменной среды
* 3. cd - изменяет текущую директорию
* 4. dir - выводит список файлов и директорий в текущей директории
* 5. copy - копирует файлы
* 6. del - удаляет файлы
* 7. move - перемещает файлы
* 8. start - запускает программу или файл
* 9. pause - приостанавливает выполнение скрипта до нажатия клавиши
* 10. if - выполняет условный оператор
* 11. for - выполняет циклические операции
* 12. goto - переходит к указанной метке в скрипте.

1. Смена кодировки в bat-файлах. Способы блокировки вывода команд в терминал во время исполнения bat-файла.

Для смены кодировки в bat-файлах можно использовать команду chcp (change code page), например: chcp 65001.

Чтобы блокировать вывод команд в терминал во время исполнения bat-файла, можно использовать оператор >nul, который перенаправляет вывод на нулевое устройство (никуда). Например: echo "Hello, world!" >nul

Эта команда выведет текст "Hello, world!" в никуда и не будет отображать его в терминале.

1. Создание переменных в bat-файлах. Способ использования. Ввод данных в переменную через консоль. Переменные выражения и доступные операторы.

В bat-файлах переменные создаются с помощью оператора set, например: set var=Hello. Здесь создается переменная var со значением "Hello".

Для использования переменной в bat-файле нужно указать ее имя вместе с символом % в начале и в конце, например: echo %var%. Эта команда выведет значение переменной var в терминал.

Для ввода данных в переменную через консоль можно использовать оператор set /p, например: set /p var=Enter your name:

В bat-файлах также доступны переменные выражения, которые позволяют выполнять математические операции, например: set /a x=2+3 Эта команда создает переменную x со значением 5, полученным в результате сложения 2 и 3.

Доступные операторы для переменных выражений:

* + (сложение)
* - (вычитание)
* \* (умножение)
* / (деление)
* % (остаток от деления)

1. Алгоритм обхода ограничения на изменение значений внутри циклов и операторов условий.

Существует ограничение на изменение значения переменной внутри группы команд в рамках задаваемых скобок, например IF или FOR. Для того, чтобы обойти данной ограничения используются:

* Запуск командного процессора - Setlocal EnableDelayedExpansion;
* Восклицательные знаки (!per!) вместо процентов (%per%), для получения значения переменной.

1. Синтаксис FOR - перебор значений. Синтаксис FOR - перебор файлов.

Перебор значений for /L %%переменная in (начало, шаг, конец) do (оператор) Где,

* /L – ключ, говорящий о том, что происходит перебор данных
* %%переменная – имя переменной, в которую записываются данны
* Начало – начальное значение
* Шаг – значение шага, с которым будет происходит увеличение значения
* Конец – конечное значение
* Оператор – задает параметры командной строки, которые используются
* с указанной командой.

Перебор файлов for %%переменная in (набор файлов) do (оператор) Где,

* %%переменная – имя переменной, в которую записываются данные
* Набор файлов – переменная, которая может содержать:
* Отдельный файл, например, my\_file.txt
* Групповой набор документов, например, \*.txt список файлов, который разделяется пробелами, причем каждый элемент списка тоже может содержать шаблон, например, my\_file.txt \*.doc my\*.avi
* Оператор – задает параметры командной строки, которые используются с указанной командой.

1. Способы запуска сторонних файлов через пакетные файлы. Описать их различия. Добавление комментариев в bat-файлы.

С помощью команды Call можно запустить другой bat-файл, но с оговоркой, что пока свою работу не завершит файл, запущенный через Call, изначальный файл не прекратит свою работу. Пример, call 2.bat Для запуска и параллельной работы с другим файлом или программой, используется команда Start.

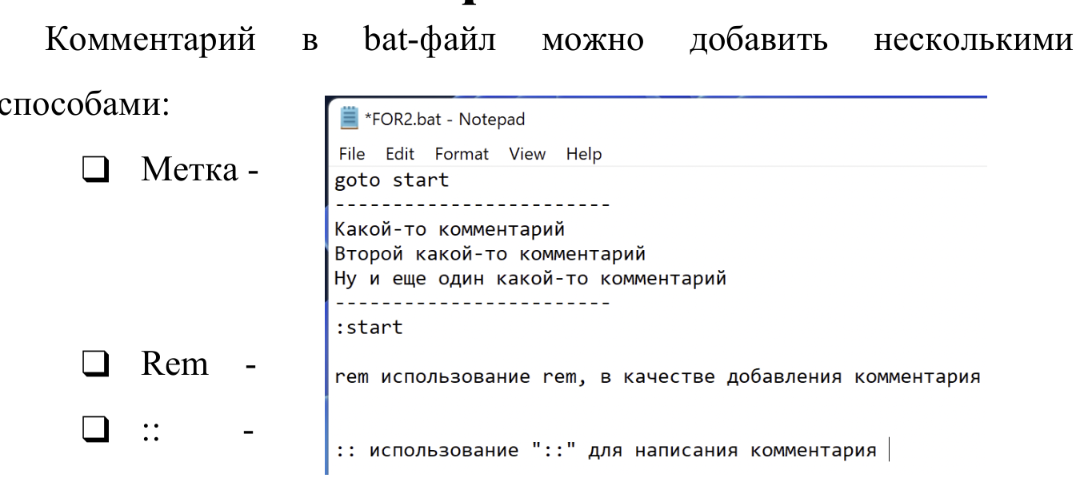


Рисунок 4 КОММЕНТАРИИ В БАТ-ФАЙЛАХ

1. Определение реестра Windows. Описать основные разделы реестра Windows и параметры, используемые во время работы с реестром.

Реестр Windows – представляет собой иерархическую древовидную, постоянно изменяющуюся базу данных, где хранятся все основные настройки операционной системы, многих установленных программ, параметров пользователя и используемого оборудования. Он обеспечивает связанную работу операционной системы как единого целого.

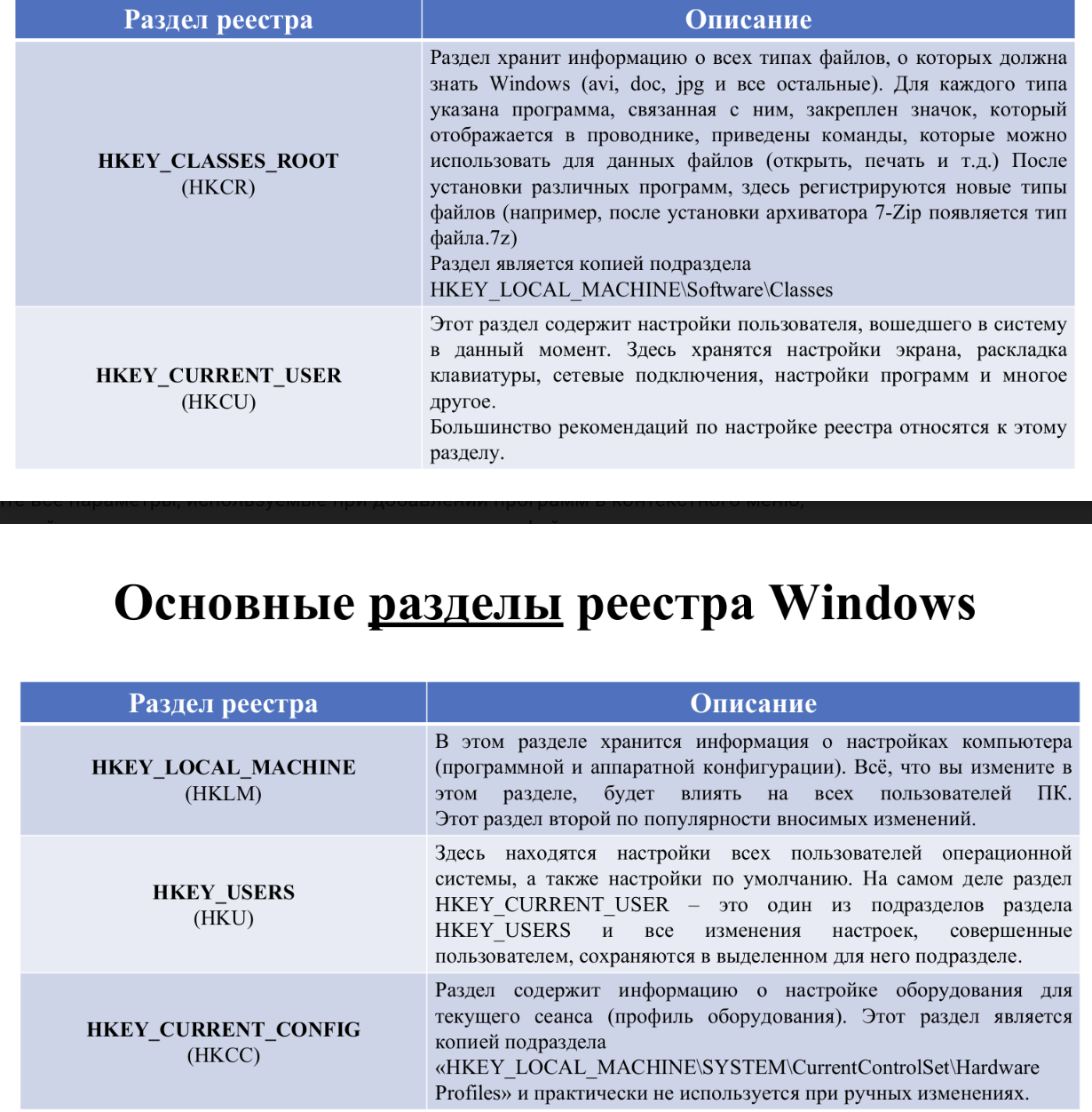


Рисунок 5 ОСНОВНЫЕ РАЗДЕЛЫ РЕЕСТРА

1. Разделы, отвечающие за автозагрузку программ для текущего пользователя и для всех пользователей системы. Какие разделы отвечают за постоянный и единоразовый автозапуск программ.

Разделы, отвечающие за автозагрузку программ для текущего пользователя:

1.HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run

2.HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce

Разделы, отвечающие за автозагрузку программ для всех пользователей системы:

1.HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

2.HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce

Разделы Run и RunOnce отвечают за постоянный автозапуск программ, т.е. программы, которые запускаются каждый раз при запуске операционной системы или при входе в систему пользователя.

Разделы RunOnce и RunOnceEx отвечают за единоразовый автозапуск программ, т.е. программы, которые запускаются один раз при следующей загрузке операционной системы или при входе в систему пользователя.

1. Определение контекстного меню. Способы применения и его виды. Параметры для программ в контекстном меню.

Контекстное меню - это меню, которое появляется при нажатии правой кнопкой мыши на определенный объект, такой как файл, папка, ярлык, текстовый документ, изображение и т.д. Контекстное меню содержит команды, которые можно выполнить для выбранного объекта, а также другие опции, такие как открытие свойств объекта, переименование, копирование, вставка и т.д.

Параметры для программ в контекстном меню – это дополнительные команды, которые можно добавить в контекстное меню определенных объектов. При этом программа должна иметь поддержку создания параметров командной строки.

Примерами параметров для программ в контекстном меню могут быть:

1. Опции открытия файлов в программе: "Открыть с помощью" или "Отправить в".

2. Команды для работы с файлами, такие как "Копировать в", "Вырезать в", "Переместить в".

3. Команды для работы с текстом, такие как "Перевести выделенный текст с помощью Google Translate", "Распечатать выделенный текст" и т.д.

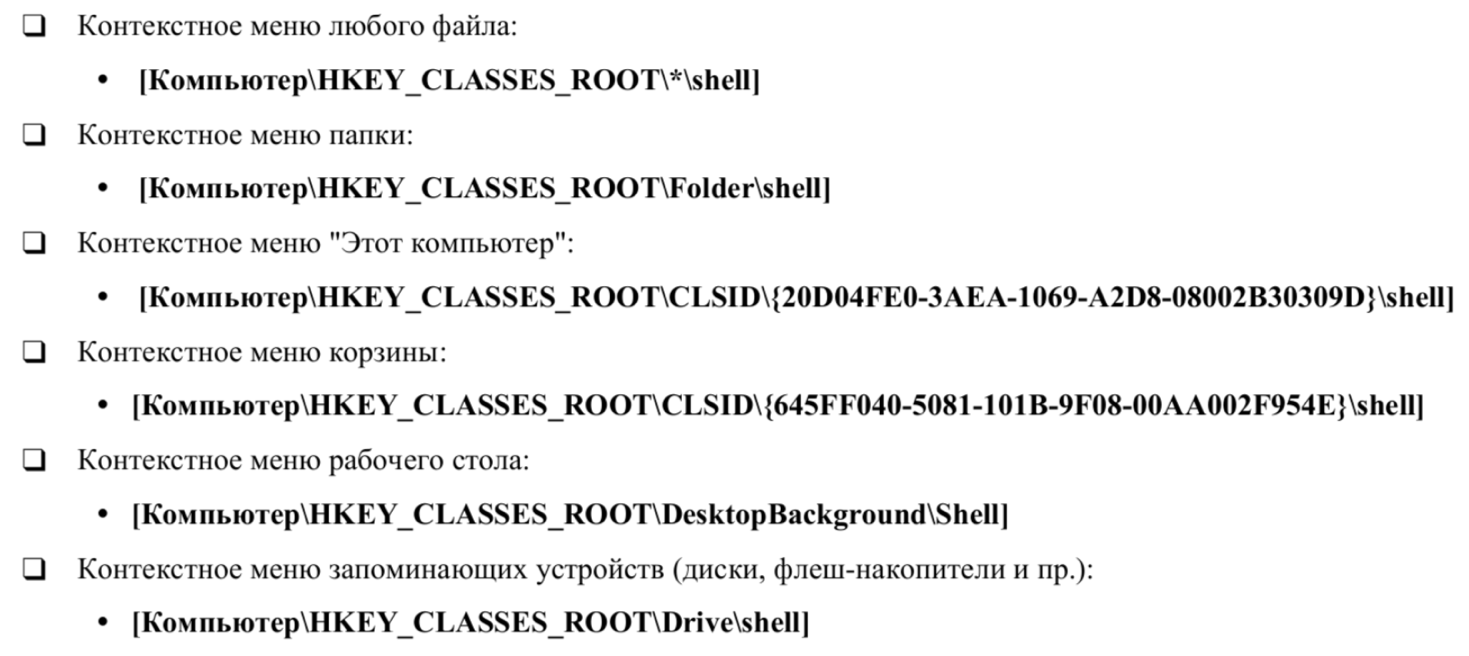


Рисунок 6 КОНТЕКСТНОЕ МЕНЮ

1. Алгоритм формирования выпадающего списка в рамках контекстного меню всех файлов.

Алгоритм формирования выпадающего списка в рамках контекстного меню всех файлов может выглядеть следующим образом:

1. Создание ключа реестра в разделе HKEY\_CLASSES\_ROOT\*\shell.

2. Создание подключей для каждого элемента выпадающего списка.

3. Настройка команд, которые будут выполняться при выборе каждого элемента списка.

4. Определение значений параметров команд, которые могут передаваться в выбранные программы.

5. Установка иконки, которая будет отображаться для каждого элемента списка.

1. Блокировка диспетчера задач, контекстного меню и скрытие системных дисков.

Воспользуемся следующим разделом: [Компьютер\HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Policies].

Для блокировки диспетчера задач необходимо:

Если в нём отсутствует подраздел с именем System, создайте его;

В созданном разделе (System) создаём параметр «DWORD 32 бита» с названием «DisableTaskMgr». Данному параметру меняем значение на 1 – отключить или 0 – включить диспетчер задач.

1. Алгоритм формирования файла для запрета запуска определенных программ, используя слияние.

Для этого понадобится блокнот, поэтому открываем его и заносим в него

текст. Вместо «program.exe», «program2.exe» и т. д., введите имена программ, запуск которых нужно заблокировать на компьютере, например, «chrome.exe», «edge.exe». Далее делаем следующее:

* Нажимаем «Сохранить как…»;
* В поле «Тип файла» выбираем «Все файлы»;
* В поле «Имя файла» вводим «DisallowRun.reg», а затем нажимаем на кнопку
* «Сохранить»;
* Для применения параметров на компьютере, щелкните по файлу «DisallowRun.reg» правой кнопкой мыши, в контекстном меню выберите «Слияние», согласитесь на применение изменений в реестре Windows. Перезагружаем систему.

1. Алгоритм формирования файла для запрета запуска всех программ с добавлением списка исключений, используя слияние.

Обратное действие блокировки определенных программ «DisallowRun» есть блокировка всех программ «RestrictRun» и создания списка исключения – тех программ, которые продолжат работать. Таким образом можно одним простым действием заблокировать абсолютно все программы на компьютере, поэтому будьте аккуратнее! Чтобы это сделать, необходимо проделать все те же действия, что и при использовании «DisallowRun», но в этот раз заменить данный параметр на «RestrictRun», а в список уже будут заноситься программы, которые продолжат работать, поэтому не забывайте первым делом вносить в этот список сам РЕЕСТР «regedit.exe», иначе выйдет не очень хорошо.

Для этого понадобится блокнот, поэтому открываем его и заносим в него

текст.