# СОДЕЖРАНИЕ

[1 Перечень используемых сокращений 6](#_Toc134887365)

[2 ВВЕДЕНИЕ 7](#_Toc134887366)

[3 ОБЗОР МЕТОДОВ И АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ 8](#_Toc134887367)

[4 ОБОСНОВАНИЕ МЕТОДОВ И АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ 9](#_Toc134887368)

[5 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПРОГРАММИСТА 10](#_Toc134887369)

[6 ОПИСАНИЕ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ 11](#_Toc134887370)

[7 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 12](#_Toc134887371)

[8 ЗАКЛЮЧЕНИЕ 13](#_Toc134887372)

[9 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14](#_Toc134887373)

# Перечень используемых сокращений

1. ПО – Программное обеспечение.
2. ОС – Операционная система.
3. ЭВМ – Электронно-вычислительная машина.
4. СУБД – Система управления базами данных.
5. ПК – Персональный Компьютер.
6. DNS – Domain Name Service, система доменных имен.
7. FPT - File Transfer Protocol, протокол передачи данных.
8. IP – Internet Protocol, Протокол Интернета.
9. SNMP - Simple Network Management Protocol, простой протокол управления сетями.
10. URL – Universal Resource Locator, универсальный указатель ресурсов.

# ВВЕДЕНИЕ

## Программное обеспечение.

Программное обеспечение (ПО) представляет собой комплекс программ, которые хранятся на запоминающих устройствах компьютера и выполняются процессором. Существует несколько групп ПО, каждая из которых имеет свои задачи и функциональность:

* Системное ПО отвечает за корректное функционирование компьютера. Оно включает в себя операционные системы (ОС), драйверы и другие программы, необходимые для обеспечения работы аппаратных устройств.
* Прикладное ПО предназначено для выполнения определенных задач пользователем. К нему относятся программы и пакеты, которые используются для решения конкретных задач, например, текстовые редакторы, графические редакторы, программы для управления базами данных и другие.
* Инструментальное ПО используется для разработки системного и прикладного ПО. К нему относятся языки программирования, компиляторы, трансляторы и другие средства разработки.

Для данного курсового проекта также важно отметить программы для управления базами данных (СУБД), которые предназначены для хранения, обработки и выдачи данных по запросу пользователя. Кроме того, следует учитывать, что существуют различные версии программного обеспечения, которые могут отличаться по функциональности, стабильности и совместимости с другими программами.

## Операционная система.

Операционная система (ОС) – совокупность программных модулей, обеспечивающих поддержку работы всем программ ЭВМ, их взаимодействие с аппаратными средствами ЭВМ, и общее управление аппаратными средствами ЭВМ.

ОС включает в себя:

* Файловую систему;
* Систему Ввода/Вывода;
* Командный процессор.

Стоит дополнительно указать важный компонент ОС – драйверы. Драйвер – специальная программа, поддерживающая и управляющая работой устройств, подключенных к ЭВМ.

Операционная система представляет из себя связующее звено между пользователем и аппаратным обеспечением ЭВМ.

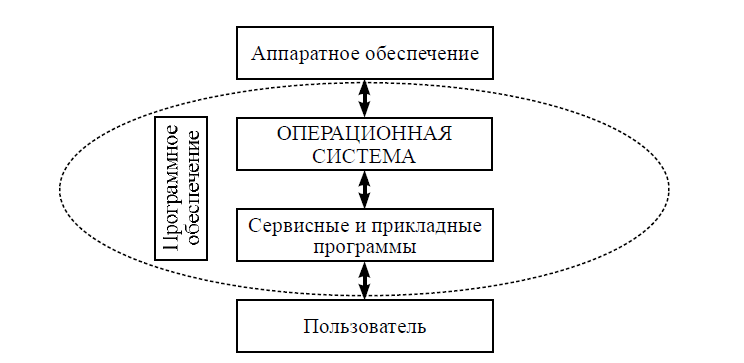


Рисунок 2.1 – Роль ОС во взаимодействии пользователя с ЭВМ.

Среди множества функций выполняемых ОС стоит обозначить инициация, загрузка программ и их исполнение, обеспечение работы системы управления файлами, планирование и диспетчеризация задач, работа с командами ввода/вывода, управление памятью и организация виртуальной памяти.

Функции ОС как правило группируются в соответствии с типами локальных ресурсов, которыми управляет система. Такие группы принято называть подсистемами. Наиболее значимые подсистемы ЭВМ:

* Подсистема управления процессами
* Подсистема управления памятью
* Подсистема управления файлами
* Подсистема управления внешними устройствами
* Подсистема пользовательского интерфейса
* Подсистема защиты данных и администрирования

По числу одновременно исполняемых задач операционные системы делятся на однозадачные и многозадачные. Примером многозадачной ОС является – Linux и дистрибутивы основанные на ядре Linux.

Многозадачные ОС выполняют функции предоставления пользователю «виртуальной машины, включают средства управления периферийными устройствами, средства управления файлами, средства взаимодействия пользователя с ОС и ЭВМ. Так же управляют разделением совместно используемых ресурсов, таких как процессор, оперативная память, файлы, внешние устройства.

## Сети и Сервер.

Сеть сама по себе это замкнутая система, обеспечивающая движение ресурсов между участниками сети и удовлетворяющая потребности пользователей в ресурсе. Компьютерные сети, называемые так же сетями передачи данных, являются логическим результатом эволюции двух важнейших научно-технических отраслей современной цивилизации – вычислительной техники и телекоммуникационных технологий.

В своей истории развития компьютерные сети прошли несколько стадий. Значимым для рассмотрения этапом стоит взять первые глобальные сети. Они решали задачу по доступу с отдельных локальных компьютеров (терминалов) к центральному крупной ЭВМ (мейнфрейму). Соединение происходило через телефонные сети с помощью модемов, открыв для многочисленных пользователей возможность получать удаленный доступ к разделяемым вычислительным ресурсам суперкомпьютеров. Развитие вычислительной техники, рост вычислительной мощности процессоров с обратно пропорциональным снижением их стоимости позволил перейти от модели Терминал-Мейнфрейм к модели ПК-ПК.

Распределенные территориально компьютеры получили возможность обмениваться данными в автоматическом режиме, что является базовым признаком любой вычислительной сети. На основе подобных механизмов в первых сетях были реализованы службы обмена файлами, синхронизации баз данных, электронной почты и другие, ставшие теперь традиционными сетевые службы. Именно в построении глобальных вычислительных сетей были впервые предложены и отработаны многие основные идеи, лежащие в основе современных вычислительных сетей: многоуровневое построение коммуникационных протоколов, концепции коммутации и маршрутизации пакетов.

Эксперименты и математическое моделирование показали что пульсирующий и в значительной степени не чувствительный к задержкам компьютерный трафик гораздо эффективнее передается сетями, работающими по принципу коммутации пакетов, когда данные разделяются на порции – пакеты, которые самостоятельно перемещаются по сети благодаря наличию адреса конечного узла в заголовке пакета.

В 1969 году появление сети APRANET стало отправной точной для создания самой известной в мире сети – Internet. Эта сеть объединяла компьютеры разных типов, работающие под управлением различных ОС с дополнительными модулями, реализующими коммуникационные протоколы общие для всех компьютеров сети. ОС этих компьютеров можно считать первыми сетевыми операционными системами.

Необходимость развития сетей и их функций росла с ростом вычислительных мощностей персональных компьютеров. Одним из этапов развития сетей стали локальные вычислительные сети, построенные изначально на не стандартизованных сетевых технологиях, а с развитием индустрии персональных компьютеров и стандартизированных сетевых технологиях типа Ethernet, Arcnet, Token Ring, Token Bus и т.п.

Сетевая технология – согласованный набор программных и аппаратных средств, механизмов передачи данных по линиям связи достаточный для построения вычислительной сети.

Персональные компьютеры стали преобладать в локальных вычислительных сетях, при этом не только в роли клиентских компьютеров но и в роли центров хранения и обработки данных, то есть сетевых серверов, тем самым вытеснив привычные мейнфреймы.

Большой вклад в сближение локальных и глобальных вычислительных сетей внесло доминирование протокола IP, который может работать поверх любых технологий и локальных и глобальных сетей объединяя различные подсети в единую составную сеть.

Долгие годы Ethernet была преимущественно технологией локальных сетей, однако с развитием новых функций и расширение уровней скоростей она стала преобладать на линиях связи и глобальных сетей.

Результатом эволюции сетей стало создание клиент-серверной сетевой архитектуры, где Сервер – это программный компонент вычислительной системы, выполняющий сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определённым ресурсам или услугам. Сервер в качестве центрального звена сети берет на себя задачи маршрутизации, функционирования сетевых служб, информационных служб, файлового сервера.

В качестве централизованной службы маршрутизации сервер берет на себя задачи распределения системы доменных имен. Система доменных имен (DNS) представляет сетевую службу, состоящую из серверов и клиентов. Сервера DNS решают задачи поддержки распределенной базы отражения имен, DNS клиенты обращаются к серверам с запросами на отображение доменного имени для IP адреса. Программа, реализующая функции DNS клиента, называется револьвером.

В иерархической организации службы DNS находятся сервера – корневые серверы, хранящие текстовые файлы имен и IP адресов DNS-серверов верхнего уровня. Сервера верхнего уровня хранят данные об именах и адресах имен, входящих в домены верхнего уровня ( com, by, eu), а так же об именах серверов DNS которые обслуживают домены второго уровня иерархии.

## Клиент-серверная архитектура.

Клиент/серверная модель представляет собой архитектуру разработки приложений, предназначенную для отделения уровня представления данных от их обработки и хранения.

Клиентский хост запрашивает у хост-сервера обслуживание запросов, и последний их обеспечивает. Запрос передается от клиента на сервер по сети, обработка выполняется на сервере и скрыта от клиента. Один сервер может обслуживать множество клиентов. Важно отметить, что сервер и клиент не обязательно являются отдельными компонентами оборудования, они могут быть реализованы программами, которые работают как на одной, так и на различных машинах. Серверная часть клиент/серверного приложения выполняет управление ресурсами, распределенными среди нескольких пользователей, получающих доступ к серверу вместе с другими клиентами.

Для коммуникации в клиент/серверной модели, компоненты используют коммуникационные протоколы, представляющие из себя регламенты коммуникации. При обмене данными компоненты сети пересылают друг другу пакеты, протоколы же определяют процедуру упаковки данных в пакеты, скорость передачи и преобразование данных в исходный вид. Сетевое программное обеспечение реализуется несколькими уровнями протоколов, расположенных слоями.

Сервер Internet представляет набор сокетов, через которые предоставляется доступ к сервисам. Примером сервиса можно обозначить протокол передачи данных (FPT). Каждый сервис представлено портом имеющим числовой идентификатор (адрес), через который обрабатывается запрошенный запрос.

Протокол TCP требует предоставление IP адреса и порта. Интернет-протокол (IP) обеспечивает логический адрес, называемый IP-адресом сетевого устройства. Каждому доменному имени компьютера в системе доменных имен DNS соответствует IP-адрес, представленный 32 битным числом, разделенным на 4 байта со значеним от 0 до 255 каждое. В случае если номер порта не указывается, то используется номер порта по умолчанию для сервиса. Для FTP сервера по умолчанию зарезервирован порт номер 21.

TCP/IP протокол обеспечивает надежный канал связи соединения точка-точка для клиент/серверных приложений, с помощью которого программы клиента и сервера устанавливают соединение и связывают сокеты. *Сокет* – программный интерфейс хоста для обеспечения информационного обмена между процессами. Таким образом сокеты используются для управления каналом связи между приложениями, установленным через сеть, каждое TCP соединение может быть однозначно идентифицировано своими двумя конечными точками. Так могут быть обеспечены множественные соединения клиент и сервера и после создания сокета через него будет выполнено дальнейшие взаимодействие клиента и сервера.

## Стек протоколов TCP/IP и FTP-протокол.

Стек TCP/IP называемый так же стеком Интернет, инкапсулирует данные верхних уровней протокола в пакеты нижних уровней. Стек делится на 4 уровня: прикладной (application), транспортный(transport), межсетевой(internet), уровень доступа к среде передачи(network access).

Самый нижний уровень (IV) - уровень доступа к среде передачи, не регламентирован, но поддерживает все популярные стандарты физического и канального уровней.

Уровень III – уровень межсетевого взаимодействия, который занимается передачей дейтаграмм с использованием различных локальных сетей, линий спецсвязи и т.п. В качестве основного протокола в стеке используется протокол IP, который изначально проектировался как протокол передачи пакетов в составных сетях, состоящих из большого количества локальных сетей, объединенными как локальными, так и глобальными связями.

Уровень II – транспортный протокол. Протоколы этого уровня обеспечивают прозрачну, сквозную доставку данных между двумя прикладными процессами. Процесс, получающий или отправляющий данные с помощью транспортного уровня, идентифицируется на этом уровне номером, который называется уровнем порта. Таким образом роль адреса отправителя и получателя на транспортном уровне выполняет номер порта.

Анализируя заголовок своего пакета, полученного от межсетевого уровня, транспортный модуль определяет по номеру порта получателя, какому из прикладных процессов направлены данные, и передает эти данные соответствующему прикладному процессу. Номера портов получателя и отравителя записываются в заголовок транспортным модулем, отправляющим данные: заголовок транспортного уровня содержит так же и другую служебную информацию: формат заголовка зависит от используемого транспортного протокола.

Уровень I – прикладной уровень. Протокол TCP/IP имеет множество связанных протоколов и сервисов прикладного уровня. Для пересылки данных другому приложению, приложение обращается к тому или иному модулю транспортного уровня.

Протокол SNMP (Simple Network Management Protocol) – используется для организации сетевого управления, решая две задачи. Первая задача связана с передачей информации, требует определения процедуры взаимодействия сервера с программой-клиентом, работающей на хосте клиента через протоколы, определяя формат сообщения которыми обмениваются клиент и сервер, форматы имен и адресов. Вторая задача состоит в регламентации данных, которые должны сохраняться и накапливаться в шлюзах, имена этих данных и синтаксис имен.

Протокол пересылки файлов FTP реализует удаленный доступ к файлу. Для обеспечения надежной передачи FTP использует в качестве транспорта протокол с установлением соединений – TCP. Кроме пересылки файлов протокол FTP предоставляет возможность интерактивной работы с удаленной машиной, позволяет пользователю указывать тип и формат запоминаемых данных, выполняет аутентификацию пользователей по средствам пользовательского логина и пароля.

Служба FTP предназначена для обмена файлами и построена на архитектуре клиент/сервер, была разработана для обмена большими файлами между машинами с разной архитектурой. Стержнем службы является протокол FTP.

В модели клиент/серверного взаимодействия реализуемой службой FTP клиент отправляет запросы серверу, и принимает файлы, а сервер обрабатывает запросы на получение файла.

Служба FPT базируется на 2 стандартах:

1. URL – универсальный способ передачи ресурсов в сети.
2. FTP – протокол передачи файлов.

Поскольку протокол FTP как протокол прикладного уровня использует в качестве транспорта протокол TCP/IP, протокол TCP разбивает информацию на пакеты и нумерует их чтобы при получении можно было правильно восстановить данные, протокол IP обеспечивает передачу всех пакетов получателю, который с помощью протокола TCP проверяет факт получения всех пакетов. После получения TCP располагает пакеты в нужном порядке и восстанавливает данные.

Протокол FTP использует в работе сразу два TCP соединения:

1. Управляющее соединение – соединение для посылки команд серверу и получение ответов от него.
2. Соединение данных – соединение для передачи файлов.

Соединение инициируется интерпретатором протокола пользователя. Управление обменом осуществляется по каналу управления в стандарте протокола Telnet. Команды FTP генерируются интерпретатором протокола пользователя и передаются на сервер. Ответы сервера отправляются пользователю так же по каналу управления. Команды FTP определяются параметрами канала передачи данных и самого процесса передачи. Они так же определяют характер работы с удаленной и локальной файловыми системами. При организации канала передачи данных последовательность действий иная, отличная от организации канала управления. В этом случае сервер инициирует обмен данными в соответствии с согласованными в сессии управления паромерами.

Канал данных устанавливается для того же хоста, что и канал управления, через который ведется установка канала данных. Канал данных может быть использован как для приема, так и для передачи данных.

Канал управления должен быть открыт при передаче данных между машинами, в случае закрытия передача данных будет прервана.

Протокол FTP определяет несколько этапов взаимодействия клиента и сервера:

1. Идентификация (ввод имени и пароля)
2. Выбор каталога
3. Определение режима обмена (поблочный, поточный, двоичный, ascii)
4. Выполнение команд обмена
5. Завершение процедуры

В современных версиях FTP серверов для канала данных сервером могут назначаться порты от N>1024.

# ОБЗОР МЕТОДОВ И АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ

# ОБОСНОВАНИЕ МЕТОДОВ И АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ

# ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПРОГРАММИСТА

# ОПИСАНИЕ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Т. П. Куль, Операционные системы: учебное пособие / Т. П. Куль. – Минск : РИПО, 2019. – 312 с.,
2. В. Олифер, Компьютерные сети. Принципы, технологии, протоколы: Юбилейное издание / Олифер В., Олифер Н. СПб.: Питер, 2020. — 1008 с..
3. Дубаков А.А. Сетевое программирование: учебное пособие / А.А. Дуба-ков – СП: НИУ ИТМО, 2013. – 248 с.
4. Гайсина Л.Ф. FTP-клиент: Методические указания к лабораторному практикуму / Гайсина Л.Ф. – Оренбург: ГОУ УГУ, 2005 – 35с.