

## *Лекция № 3*

### ***Языки взаимодействия пользователя с операционной системой. Понятие программного интерфейса, его назначение. Виды интерфейсов.***

Современные операционные системы (ОС) служат связующим звеном между пользователем и аппаратным обеспечением компьютера. Чтобы пользователь мог эффективно работать с системой, необходимы средства взаимодействия. В этой лекции мы подробно рассмотрим различные языки взаимодействия с операционной системой, а также понятие программного интерфейса и его виды.

Язык взаимодействия пользователя с ОС — это способ, с помощью которого пользователь может давать команды операционной системе и получать результаты их выполнения. Эти команды могут быть как текстовыми, так и графическими.

#### **Текстовые интерфейсы**

Текстовые интерфейсы, или командные строки, предполагают ввод команд в текстовом формате. Это один из старейших и наиболее мощных способов взаимодействия с ОС. С помощью командной строки пользователи могут напрямую взаимодействовать с системой, используя определенные команды для выполнения различных операций.

- В операционных системах семейства Unix/Linux используется терминал, где пользователь может вводить команды, такие как `ls` для отображения списка файлов в директории или `cd` для перехода между каталогами.

- В Windows существует командная строка (`cmd`), где можно выполнять аналогичные команды, например `dir` для отображения списка файлов.

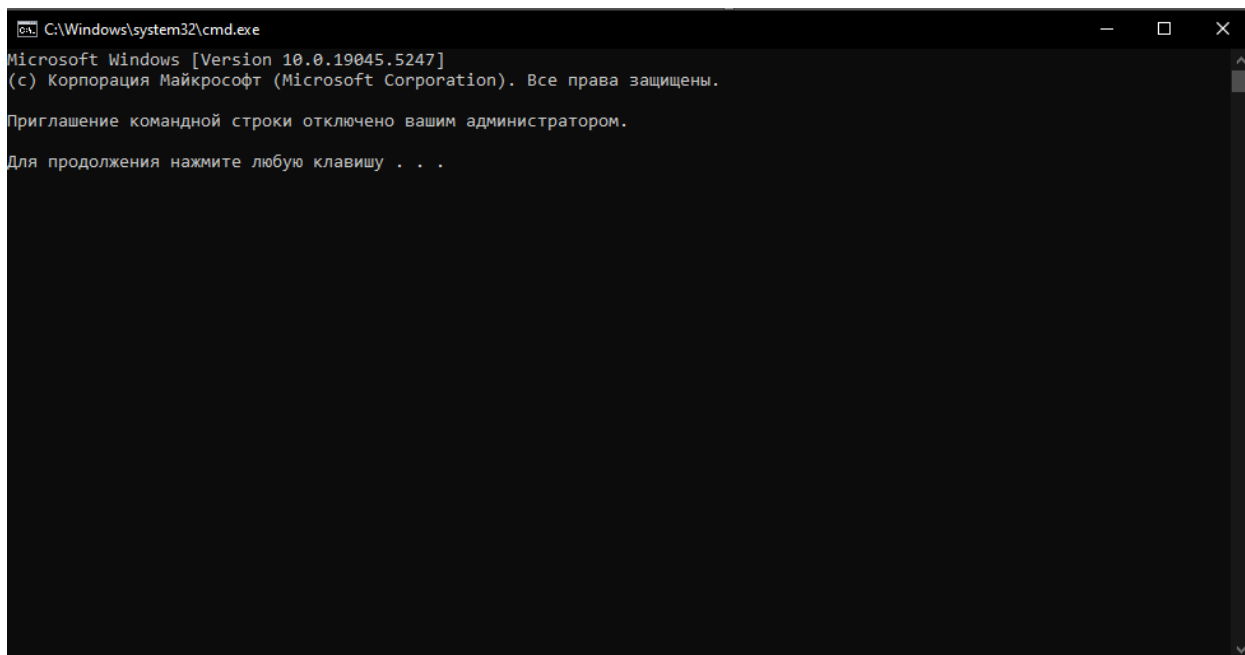


Рисунок 1 – командная строка в Windows

Командные оболочки, такие как Bash в Linux или PowerShell в Windows, предоставляют пользователю мощный инструментарий для автоматизации и администрирования системы.

### **Графические интерфейсы**

Графический пользовательский интерфейс (GUI, Graphical User Interface) представляет собой взаимодействие с ОС через визуальные элементы, такие как окна, кнопки, меню и значки. GUI делает операционную систему доступной для широкой аудитории, включая тех пользователей, кто не обладает специальными техническими знаниями.

- В Windows графический интерфейс включает в себя рабочий стол, панели задач, окна приложений, меню "Пуск" и другие элементы управления.

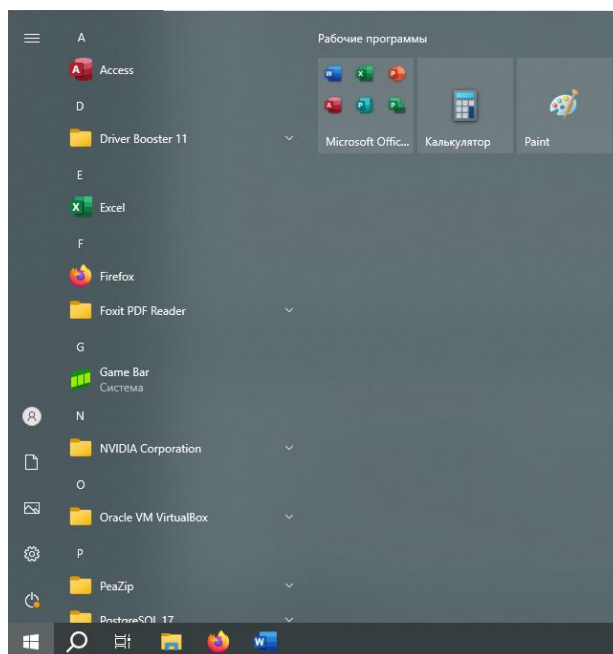


Рисунок 2 – графический интерфейс Windows

- В macOS аналогичные функции выполняют элементы интерфейса, такие как док, меню в верхней части экрана, и различные всплывающие окна.



Рисунок 3 – графический интерфейс в macOS

Графический интерфейс значительно упрощает взаимодействие, делая его более интуитивно понятным и визуально привлекательным.

## Голосовые интерфейсы

Голосовые интерфейсы — это новая форма взаимодействия, где пользователь может давать команды ОС с помощью речи. Голосовые системы используют технологии распознавания речи и могут выполнять команды, такие как поиск информации, запуск приложений или управление настройками.

- Голосовой помощник Siri в Apple, Google Assistant на Android и Cortana в Windows — это примеры голосовых интерфейсов, которые позволяют пользователям взаимодействовать с ОС с помощью голоса.

## **Программный интерфейс (API)**

Программный интерфейс (API, Application Programming Interface) — это набор средств и протоколов, который позволяет программному обеспечению взаимодействовать с операционной системой или другими программами. API является основным инструментом для разработчиков, создающих приложения, которые должны работать на разных платформах.

API включает в себя:

- Функции или методы, которые предоставляет операционная система или другие программы.
- Описание того, как и в каком порядке нужно вызывать эти функции.
- Формат данных, который ожидает система или приложение.

API позволяет скрывать сложность работы с операционной системой и упрощает разработку приложений. Разработчик использует API для выполнения определенных операций, таких как доступ к файловой системе, управление памятью, создание сетевых соединений и многое другое.

Главной целью API является создание абстракции над сложной внутренней реализацией операционной системы или другой программы.

Это позволяет программистам не беспокоиться о низкоуровневых деталях и сосредоточиться на решении задач своей программы.

- Если приложение хочет открыть файл, оно не взаимодействует напрямую с аппаратной частью, а использует функцию API ОС, например, `fopen` в языке C для открытия файла.

- Для работы с сетью можно использовать API сокетов, который абстрагирует детали сетевых соединений.

## Виды интерфейсов

### Пользовательский интерфейс (UI)

Пользовательский интерфейс (User Interface, UI) — это совокупность всех средств, с помощью которых пользователи взаимодействуют с системой. Это может быть как командная строка (CLI), так и графический интерфейс (GUI). Важнейшими аспектами UI являются удобство, доступность и эффективность взаимодействия.

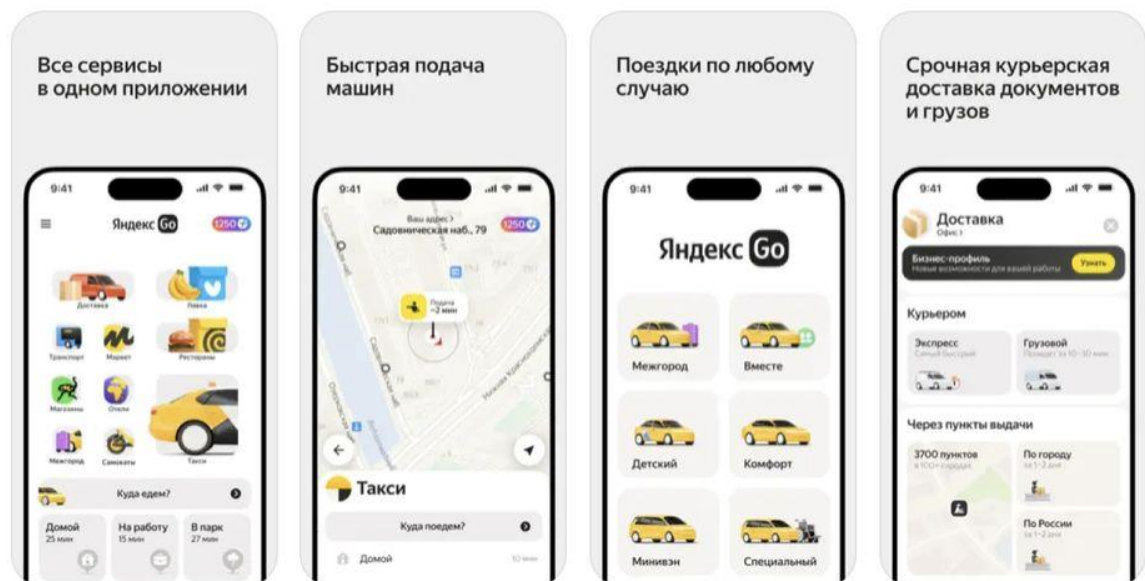


Рисунок 4 – пример пользовательского интерфейса

Примером эффективного использования UI является интерфейс приложения для заказа такси. В этом приложении есть кнопки для вызова машины, меню выбора типа автомобиля и поля ввода адреса. Они расположены таким образом, чтобы пользователю не нужно было тратить много времени на поиск необходимой функции. Это делает взаимодействие с приложением быстрым и удобным.

### Программный интерфейс (API)

API, как уже обсуждалось, — это средство взаимодействия программ с операционной системой. В отличие от UI, API ориентирован на разработчиков, а не на конечных пользователей.

## **Аппаратный интерфейс (Hardware Interface)**

Аппаратный интерфейс — это набор протоколов и стандартов, которые используются для взаимодействия операционной системы с аппаратным обеспечением. Это включает драйверы устройств, шины передачи данных и другие низкоуровневые механизмы. Например, операционная система использует интерфейс для взаимодействия с видеокартой, жестким диском или клавиатурой. USB (Universal Serial Bus): USB является универсальным и широко используемым интерфейсом для подключения и питания периферийных устройств к компьютерам и микроконтроллерам. Он поддерживает передачу данных и электропитание через один и тот же кабель, обеспечивая удобство и простоту использования. USB интерфейс отличается высокой скоростью передачи данных, гибкостью в подключении различных устройств и возможностью горячей замены, что делает его идеальным для множества приложений, от флеш-накопителей до внешних жестких дисков и периферийных устройств.

## **Сетевой интерфейс**

Сетевой интерфейс — это механизм, который позволяет системе и приложениям обмениваться данными по сети. Сетевые интерфейсы могут быть реализованы через сокеты или более высокоуровневые API.

## Примеры интерфейсов в операционных системах

### UNIX/Linux

В UNIX и Linux операционных системах существует несколько типов интерфейсов:

- Командная оболочка (Shell): основное средство взаимодействия с системой через командную строку. Это может быть Bash, Zsh, Fish и другие оболочки.

- Графический интерфейс: в Linux существует несколько графических оболочек, таких как GNOME, KDE, Xfce, которые обеспечивают визуальное взаимодействие с пользователем.



Рисунок 5 – графическая оболочка GNOME

- Системные вызовы и API: программное взаимодействие с ядром системы осуществляется через API, предоставляемые операционной системой.

### Windows

В Windows операционной системе:

- Командная строка (Command Prompt) и PowerShell: командные интерфейсы для выполнения различных задач.

- Graphical User Interface (GUI): графический интерфейс, в котором все основные функции, такие как управление окнами, запуск программ, управление файлами, выполняются с помощью визуальных элементов.

- Windows API: набор функций, предоставляемых ОС для взаимодействия с приложениями, например, для работы с окнами, графикой, сетью и т. д.

Языки взаимодействия пользователя с операционной системой являются важной частью ее функционала и определяют удобство работы с компьютером. От текстовых интерфейсов до сложных графических и голосовых систем — каждый способ взаимодействия предоставляет свои преимущества в зависимости от задач и потребностей пользователя. Понимание программных интерфейсов (API) позволяет разработчикам создавать более эффективные и совместимые приложения. Знание этих интерфейсов — ключевая часть понимания того, как работают современные операционные системы.

1. Чем отличаются текстовые и графические интерфейсы взаимодействия с ОС?
2. Что такое API и какое его назначение в программировании?
3. Приведите примеры API в популярных операционных системах.
4. Какие виды интерфейсов существуют и в чем их отличие?