1. **Аутентификация**: Это процесс проверки подлинности пользователя или системы. Обычно включает предоставление учетных данных (например, имя пользователя и пароль) для проверки.
2. **Авторизация**: Это процесс предоставления доступа к определенным ресурсам или функциям после успешной аутентификации. То есть определяется, какие действия пользователь или система могут выполнять после проверки подлинности.
3. **Дискреционная система безопасности (Discretionary Access Control, DAC)**: Это модель безопасности, в которой владелец ресурса имеет полный контроль над доступом к нему и может делегировать права доступа другим пользователям или группам.
4. **Учетная запись**: Это набор данных, который идентифицирует пользователя или систему в информационной системе. Он может включать имя пользователя, пароль, права доступа и другую информацию.
5. **Пользователь операционной системы**: Это лицо или программа, которая использует ресурсы операционной системы для выполнения задач.
6. **Параметры создания Windows-пользователя**: Обычно включают имя пользователя, пароль, права доступа, принадлежность к группам и другие настройки безопасности.
7. **Группа пользователей**: Это совокупность пользователей, объединенных общими правами доступа или другими характеристиками.
8. **Соотношение пользователи и группы пользователей**: Пользователи могут быть членами одной или нескольких групп пользователей, а группы могут включать одного или несколько пользователей.
9. **Параметр level в функциях API для работы с учетными записями**: Это параметр, определяющий уровень доступа или детализации информации, которую функция должна вернуть или выполнить.
10. **Действия над учетными записями пользователей с помощью API Windows**: Создание, изменение, удаление учетных записей, аутентификация, авторизация и т. д.
11. **Действия над учетными записями групп пользователей с помощью API Windows**: Создание, изменение, удаление групп, добавление и удаление пользователей из групп и т. д.
12. **Callback-функция**: Это функция, передаваемая в качестве аргумента в другую функцию, которая вызывается в определенный момент времени или при наступлении определенного события.
13. **Windows-сервис**: Это приложение, которое работает в фоновом режиме в операционной системе Windows и не имеет интерфейса пользователя.
14. **Компоненты Windows-сервиса**: Основной код службы, управляющий ее жизненным циклом и обработкой запросов, файл конфигурации, дополнительные ресурсы и т. д.
15. **Хранение информации о Windows-сервисах**: Информация о службах хранится в системном реестре Windows и в файловой системе.
16. **Цели использования Windows-сервиса**: Предоставление функций, работающих в фоновом режиме, автоматизация задач, обеспечение постоянного доступа к определенным ресурсам и другие.
17. **SEH (Structured Exception Handling)**: Это механизм обработки исключений в Windows, который позволяет обрабатывать исключения и восстанавливаться от них.
18. **Блоки SEH**: **try**, **except**, **finally**, **leave**.
19. **Значения SEH-фильтра**: Они определяют, как обрабатывать исключение или передавать его на следующий уровень.
20. **Системная функция-фильтр**: Это функция, которая используется для обработки исключений и определяется пользователем.
21. **Стандартный поток операционной системы**: Это поток ввода-вывода, который связан с операционной системой и предоставляет доступ к ее ресурсам.
22. **Windows Console**: Это окружение в Windows, предоставляющее текстовый интерфейс и возможность взаимодействия с пользователем через командную строку.
23. **Основные составляющие Windows Console**: Командный интерпретатор (командная строка), консольное окно, стандартные потоки ввода-вывода.
24. **Действия с консолью с помощью API Windows**: Создание консоли, чтение и запись в стандартные потоки, установка цветов и размеров окна и другие.
25. **Асинхронная операция ввода/вывода**: Это операция ввода или вывода, которая выполняется параллельно с основным потоком программы и не блокирует его выполнение.
26. **Буферизация ввода/вывода**: Это техника оптимизации работы с данными, при которой данные временно сохраняются в буфере, прежде чем быть переданными или записанными.
27. **Файловая система**: Это метод организации и хранения данных на носителях информации, таких как жесткие диски, флеш-накопители и другие.
28. **Отличия разных файловых систем**: Это может быть различие в способе организации данных, поддержке атрибутов файлов, размере файлов и т. д.
29. **Операции в файловой системе с помощью API Windows**: Создание, удаление, чтение, запись, перемещение, переименование файлов и директорий и другие.
30. **Порт завершения асинхронных операций ввода/вывода**: Это объект, который используется для отслеживания завершения асинхронных операций ввода/вывода.
31. **Назначение платформы Docker**: Предоставление среды для разработки, доставки и выполнения приложений в контейнерах.
32. **Два основных компонента Docker**: Docker Engine (для создания и управления контейнерами) и Docker Hub (репозиторий образов контейнеров).
33. **Разница между Docker Image и Docker Container**: Docker Image - это шаблон, используемый для создания контейнера, а Docker Container - это экземпляр Docker Image, который запущен и выполняется.
34. **Устройство Docker UFS**: Это файловая система, используемая в контейнерах Docker для организации и хранения файлов и данных контейнера.
35. **Пользователь**:
    * Пользователь - это лицо, которое использует компьютерную систему или программное обеспечение для выполнения задач.
    * Пользователь может быть человеком, компьютерным программным обеспечением или другой системой, требующей доступа к ресурсам или функциям.
    * В контексте операционной системы, пользователь может взаимодействовать с системой через графический интерфейс, командную строку или приложения.
    * Пользователь может использовать учетные данные (логин и пароль) для аутентификации в системе и получения доступа к ее ресурсам.
36. **Учетная запись**:
    * Учетная запись - это запись в информационной системе, которая содержит информацию о пользователе и его правах доступа к ресурсам.
    * Учетная запись обычно включает в себя уникальное имя пользователя (логин), пароль для аутентификации и набор прав доступа, определяющих, какие действия пользователь может выполнять в системе.
    * Учетная запись также может содержать дополнительную информацию о пользователе, такую как его полное имя, адрес электронной почты, группы, к которым он принадлежит, и т. д.

Таким образом, основное различие между пользователем и учетной записью заключается в том, что пользователь - это субъект, который использует систему, в то время как учетная запись - это сущность, представляющая пользователя в системе и содержащая информацию о его идентификации и правах доступа.

**Идентификация** — это процесс предоставления пользователем своих уникальных данных для определения его личности в системе. Примеры идентификаторов: логин, номер телефона, адрес электронной почты.

**Аутентификация** — это проверка подлинности предоставленных пользователем данных. Это подтверждение, что пользователь действительно является тем, за кого себя выдает. Методы аутентификации включают пароли, одноразовые коды (OTP), биометрию.

**Авторизация** — это процесс определения прав и привилегий пользователя после его аутентификации. На этом этапе система решает, какие ресурсы и операции доступны пользователю. Примеры: доступ к файлам, выполнение команд, использование приложений.

1. **GetUserName**
   * Описание: Эта функция получает имя текущего пользователя.
   * Сигнатура: BOOL *GetUserName(LPTSTR lpBuffer, LPDWORD pcbBuffer);*
   * Параметры:
     + **lpBuffer**: Указатель на буфер, который будет содержать имя пользователя.
     + **pcbBuffer**: Указатель на размер буфера. После выполнения функции в этой переменной будет содержаться фактический размер имени пользователя.
   * **Возвращаемое значение**: Если функция выполнена успешно, возвращает ненулевое значение. В случае ошибки возвращает нуль.
2. **NetUserGetLocalGroups**
   * Описание: Эта функция получает локальные группы, в которые входит указанный пользователь.
   * Сигнатура: *NET\_API\_STATUS NetUserGetLocalGroups(LPCWSTR servername, LPCWSTR username, DWORD level, DWORD flags, LPBYTE \*bufptr, DWORD prefmaxlen, LPDWORD entriesread, LPDWORD totalentries);*
   * Параметры:
     + **servername**: Имя сервера (если **NULL**, используется локальный компьютер).
     + **username**: Имя пользователя, для которого запрашиваются группы.
     + **level**: Уровень информации (0 или 1).
     + **flags**: Флаги, определяющие, какие группы включать.
     + **bufptr**: Указатель на буфер, который будет содержать информацию о группах.
     + **prefmaxlen**: Предпочтительный максимальный размер буфера.
     + **entriesread**: Указатель на количество записей, возвращенных в буфере.
     + **totalentries**: Указатель на общее количество записей.
   * **Возвращаемое значение**: Возвращает код состояния **NET\_API\_STATUS**. **NERR\_Success** (0) означает успешное выполнение.
3. **NetApiBufferFree**
   * Описание: Эта функция освобождает буфер, выделенный функциями сети.
   * Сигнатура: *NET\_API\_STATUS NetApiBufferFree(LPVOID Buffer);*
   * Параметры:
     + **Buffer**: Указатель на буфер, который нужно освободить.
   * **Возвращаемое значение**: Возвращает код состояния **NET\_API\_STATUS**. **NERR\_Success** (0) означает успешное выполнение.