1. Аутентификация - процесс проверки подлинности пользовательской учетной записи, чтобы убедиться, что пользователь является тем, за кого себя выдает.
2. Авторизация - процесс предоставления прав доступа пользователю после успешной аутентификации. Он определяет, какие действия и ресурсы могут быть доступны пользователю.
3. Дискреционная система безопасности - это система управления доступом, в которой доступ к ресурсам определяется владельцем ресурса. Владелец имеет полный контроль над доступом и может предоставлять или ограничивать доступ другим пользователям.
4. Учетная запись - это запись, содержащая информацию о пользователе или группе пользователей, такую как имя пользователя, пароль, права доступа и другие атрибуты.
5. Пользователь операционной системы - это лицо или программа, которая использует операционную систему для выполнения задач и обращения к ресурсам компьютера.
6. При создании Windows-пользователя задаются параметры, такие как имя пользователя, пароль, группа пользователей, права доступа, домен и другие атрибуты.
7. Группа пользователей - это совокупность пользователей, объединенных по какому-либо признаку или для выполнения определенной задачи. Группа может иметь свои права доступа и привилегии.
8. Пользователи могут быть членами групп пользователей. Группа пользователей может включать в себя одного или нескольких пользователей.
9. Параметр level в функциях API для работы с учетными записями пользователей и групп пользователей определяет версию структуры данных, используемую для передачи информации о пользователях или группах пользователей.
10. С помощью API Windows можно создавать, удалять, изменять учетные записи пользователей, устанавливать и изменять пароли, управлять правами доступа, получать информацию о пользователях и другие операции.
11. С помощью API Windows можно создавать, удалять, изменять группы пользователей, добавлять или удалять пользователей из группы, управлять правами доступа группы и другие операции.
12. Callback-функция - это функция, передаваемая в качестве аргумента другой функции, которая будет вызвана в определенный момент или по событию.
13. Windows-сервис - это программа, которая работает в фоновом режиме без прямого взаимодействия с пользователем. Она выполняет определенные задачи или предоставляет функциональность другим программам или пользователям.
14. Windows-сервис состоит из исполняемого файла, который выполняет код сервиса, и сервисного контроллера, который управляет жизненным циклом сервиса, включая запуск, остановку и перезапуск.
15. Информация о Windows-сервисах хранится в реестре Windows.
16. Windows-сервисы применяются для выполнения задач в фоновом режиме, автоматического выполнения определенных операций, предоставления функциональности другим программам или пользователям, а также для обеспечения непрерывной работы системы.
17. SEH расшифровывается как Structured Exception Handling и представляет собой механизм обработки исключений в Windows.
18. Блоки SEH включают блок try, блокcatch и блок finally, которые используются для обработки исключений.
19. Значения SEH-фильтра могут указывать, что нужно сделать с исключением, например, продолжить выполнение программы, обработать исключение в блоке catch или передать его выше по стеку вызовов.
20. Системная функция-фильтр - это функция, которая используется в блоке SEH для обработки исключений и принятия решения о том, как их обработать.
21. Стандартный поток операционной системы - это поток ввода-вывода, который связан с консолью операционной системы и используется для взаимодействия с пользователем через командную строку.
22. Windows Console - это окружение командной строки в операционной системе Windows, где пользователь может вводить команды и получать вывод от программ.
23. Основные составляющие Windows Console включают командную строку, окно консоли, стандартные потоки ввода-вывода и другие элементы интерфейса.
24. С помощью API Windows можно выполнять действия с консолью, такие как создание и управление окнами консоли, чтение и запись в стандартные потоки, управление цветами и шрифтами консоли, управление курсором и другие операции.
25. Асинхронная операция ввода/вывода - это операция, которая выполняется параллельно с основной программой, без блокировки ее выполнения. Это позволяет программе продолжать работу, пока операция ввода/вывода выполняется.
26. Буферизация ввода/вывода - это механизм, который используется для временного хранения данных в памяти, прежде чем они будут записаны на диск или переданы по сети. Это позволяет увеличить производительность операций ввода/вывода.
27. Файловая система - это способ организации и хранения файлов на компьютере или другом устройстве. Она определяет структуру директорий, формат файлов, правила доступа и другие атрибуты файлов.
28. Разные файловые системы могут отличаться по формату файлов, максимальному размеру файла, поддерживаемым атрибутам и правам доступа, поддержке журналирования и другим характеристикам.
29. С помощью API Windows можно выполнять операции в файловой системе, такие как создание и удаление файлов и директорий, чтение и запись данных в файлы, управление атрибутами файлов, получение информации о файлах и другие операции.
30. Порт завершения асинхронных операций ввода/вывода - это механизм, который используется для координации и управления асинхронными операциями ввода/вывода, включая отслеживание их завершения и обработку результатов.
31. Платформа Docker предназначена для упаковки, доставки и запуска приложений в контейнерах. Она обеспечивает изоляцию и портативность приложений, позволяя запускать их на разных операционных системах и хостах.
32. Две основных компоненты платформы Docker - Docker Engine, который управляет и запускает контейнеры, и Docker Registry, который служит для хранения и распространения Docker Images.
33. Docker Image - это снимок (snapshot) состояния контейнера, который содержит все необходимые компоненты для запуска приложения, включая код, зависимости, конфигурацию и другие файлы. Docker Image является основой для создания и запуска контейнеров.

Docker Container - это выполняющийся экземпляр Docker Image. Контейнер обеспечивает изоляцию приложения и его зависимостей от хостовой системы и других контейнеров, позволяя запускать приложения в однородной среде, независимо от операционной системы и конфигурации хоста.

1. Docker UFS (Union File System) - это механизм файловой системы, который используется Docker для создания и управления контейнерами. UFS предоставляет механизм объединения (union) нескольких файловых систем в одну, позволяя контейнерам иметь свою собственную изолированную файловую систему, независимую от хостовой системы.

Устройство Docker UFS состоит из следующих основных компонентов:

Base Image (Базовый образ): Это основа для создания Docker Image. Базовый образ содержит минимальный набор файлов и директорий, необходимых для работы приложения. Он может быть основан на другом базовом образе или создан с нуля.

Docker Image (Образ Docker): Образ Docker - это неизменяемый снимок состояния контейнера, который содержит все необходимые компоненты для запуска приложения, включая файлы, зависимости, конфигурацию и другие ресурсы. Docker Image создается на основе базового образа и может быть использован для запуска нескольких контейнеров.

Container Layer (Слой контейнера): Каждый контейнер создает свой собственный слой, который является записываемым и содержит только изменения, внесенные в файловую систему контейнера. Слой контейнера строится на основе Docker Image и может быть изменен или удален без влияния на другие слои или контейнеры.

Overlay File System (Система объединения файлов): Docker использует OverlayFS в качестве механизма объединения файловых систем. OverlayFS позволяет объединить несколько слоев файловой системы в одну общую файловую систему. Каждый слой представляет собой отдельную файловую систему, и изменения, внесенные в один слой, видны в контейнере через общую файловую систему.

Copy-on-write (Копирование при записи): При записи в файловую систему контейнера, Docker использует механизм копирования при записи, чтобы не изменять исходные файлы. Это позволяет экономить место на диске и повышает производительность, так как только измененные или добавленные файлы копируются на записываемый слой.

Docker UFS позволяет контейнерам быть изолированными и независимыми друг от друга, а также от хостовой системы. Каждый контейнер имеет свою собственную файловую систему, которая может быть изменена или удалена без влияния на другие контейнеры или хостовую систему.