**1. что такое поток упавления**

Поток управления в Windows является объектом ядра операционной

системы, которому выделяется процессорное время для выполнения

приложения. Каждому потоку принадлежат следующие ресурсы:

- код исполняемой функции;

- набор регистров процессора;

- область оперативной памяти;

- стек для работы приложения;

- стек для работы операционной системы;

- маркер доступа, содержащий информацию для системы безопасности.

Все эти ресурсы образуют так называемый контекст потока в Windows.

**2. что такое гланый поток**

Процесс c именем main, является главным процессом сервера, который получает управление от операционной системы. Каждое приложение, работающее в среде Windows, имеет, по крайней мере, один поток, который называется первичным или главным потоком.

Это поток, от которого порождаются все "дочерние" потоки. Часто он должен быть последним потоком, завершающим выполнение, так как он предпринимает различные завершающие действия.

**3. параллельный, однопоточный**

Будем говорить, что программа является многопоточной, если в ней существуют одновременно несколько потоков управления. Сами потоки в этом случае называются параллельными. Если в программе может существовать только один поток, то такая программа называется однопоточной.

**4. безопасность для потока, какими двумя функциями она управляет**

1) свойством реентерабельности;

2) функция обеспечивает блокировку доступа к критическим ресурсам, которые она использует

В общем случае функция называется реентерабельной, если она не изменяет собственный код или собственные статические данные. Другими словами программный код реентерабельной функции должен допускать корректное его использование несколькими потоками одновременно. Блокировка требуется в том случае, если функцией используется ресурс, доступ к которому может быть только упорядоченным (критический ресурс). Примером критического ресурса может служить изменяемые функцией статические и глобальные переменные.

**5. синхронизация сервера**

В арсенале Windows имеется много различных механизмов синхронизации: критические секции, мьютексы, семафоры, события, асинхронные процедуры, ожидающие таймеры и.д. В тех случаях, когда требуется совместная работа нескольких потоков с одной общей переменной, может быть применен специальный механизм упрощенной синхронизации – атомарные операции. Синхронизация будет осуществляться между двумя потоками: AcceptServer и GarbageCleaner

**6. критическая секция**

Критические секции, является одним из самых простых механизмов синхронизации и в нашей модели могут быть использованы для исключения совместного использования списка ListContact потоками AcceptServer и GarbageCleaner. Критическая секция является объектом операционной системы типа CRITICAL\_SECTION.

**7. асинхронная процедура**

Асинхронной процедурой называется функция, которая выполняется асинхронно в контексте какого-нибудь потока. Для исполнения асинхронной процедуры необходимо определить асинхронную процедуру, указать поток, в контексте которого она будет выполняться, и дать разрешение на выполнение асинхронной процедуры.

**8. ожидающий таймер**

Ожидающим таймером в Windows, называется объект синхронизации, который переходит в сигнальное состояние при наступлении заданного момента времени. Если ожидающий таймер ждет момента перехода в сигнальное состояние, то говорят, что он находится в активном состоянии. Другое состояние ожидающего таймера пассивное – из этого состояния он не может перейти в сигнальное состояние.

**9. какие три состояния у него есть**

Сигнальное, активное, пассивное

**10. сигнальный, активный, пассивный что занчит**

сигнальное состояние при наступлении заданного момента времени. Если ожидающий таймер ждет момента перехода в сигнальное состояние, то говорят, что он находится в активном состоянии. Другое состояние ожидающего таймера пассивное – из этого состояния он не может перейти в сигнальное состояние.

**11. таймер с ручным и автоматическим сбосом что значит**

По способу перехода из сигнального стояния в несигнальное, ожидающие таймеры разделяются на таймеры с ручным сбросом и таймеры с автоматическим сбросом, иначе называемые таймерами синхронизации.

**12. переодическое и непериодическое состояние**

По способу перехода из несигнального состояния в сигнальное, ожидающие таймеры бывают периодические и непериодические. Периодические таймеры работают по циклу: активное состояние – сигнальное состояние – активное состояние. Непериодические таймеры могут только один раз перейти из активного состояния в сигнальное.

**13. атомарные операции**

Блокирующие функции выполняют несколько элементарных операций, которые объединяются в одну неделимую операцию, называемую атомарной операцией.

**14. режим блокировки и без нее что значит**

Алгоритм работы функции accept, который рассматривался выше, был обусловлен режимом блокировки (blocking mode), установленным (по умолчанию) для сокета. Переключение сокета в режим без блокировки ( nonblocking mode), позволяет избежать приостановки программы. В режиме без блокировки выполнение accept, не приостанавливает выполнение потока, как это было прежде, а возвращает значение нового сокета, если обнаружен запрос на создание канала (функция connect, выполненная клиентом), или значение INVALID\_SOCKET, если запроса на создание канала нет в очереди запросов или возникла ошибка. Для того, чтобы отличить ошибку от отсутствия запроса, используется уже рассмотренная выше функция WSAGetLastError, которая в последнем случае возвращает значение WSAEWOULDBLOCK.

**15. какие приоритеты потоков и процессов есть?**

дается потоку с высшим приоритетом. Приоритеты потоков в Windows определяются относительно приоритета процесса, в рамках которого они исполняются. Приоритеты потоков изменяются от 0 (низший приоритет) до 31 (высший приоритет).

При диспетчеризации процессов и потоков квант времени выделяется потоку. Приоритет потока, который учитывается операционной системой при выделении процессорного времени, называется базовым или основным приоритетом потока. Всего существует 32 базовых приоритетов – от 0 до 31. Для каждого базового приоритета существует очередь потоков. При диспетчеризации квант процессорного времени выделяется потоку, который стоит первым в очереди с наивысшим приоритетом. Базовый приоритет потока определятся исходя из приоритета процесса и уровня приоритета потока. Уровень приоритета потока может быть: низший, ниже нормального, нормальный, выше нормального, высший приоритет, приоритет фонового потока и приоритет потока реального времени.

****

**16. задача оповещения и условных процедур что такое?**

Выше уже упоминался механизм событий, позволяющий оповестить поток о некотором выполненном действии, произошедшем за пределами потока. Саму задачу оповещения часто называют задачей условной синхронизации.

**17. на какие группы разбиты сервера?**

**Администратора и пользователей(?)**

**Итеративные и параллельные**

**18. что такое динамические библиотеки?**

Динамические библиотеки представляют собой программный модуль, который может быть загружен в виртуальную память процесса как статически, во время создания исполняемого модуля процесса, так и динамически во время исполнения процесса операционной системой.

**19. как загрузить и импортировать библиотеку?**

Для создания dll-библиотеки в среде Visual Studio необходимо выбрать проект типа Win32 Dynamic-Link Library. Как и любая программа на языке С++, динамически подключаемая библиотека имеет главную функцию, которая отмечает точку входа программы при ее исполнении операционной системой. Главная функция dll-библиотеки называется DllMain – ее шаблон автоматически создается Visual Studio.

**20. что такое decline?**

Квалификатор \_\_declspec(dllexport) применяется для обозначения экспортируемых dll-библиотекой функций.

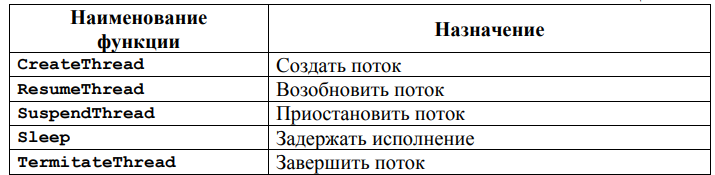
**21. как освободить ожидающие потоки?**

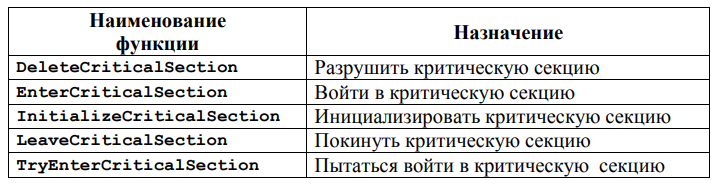
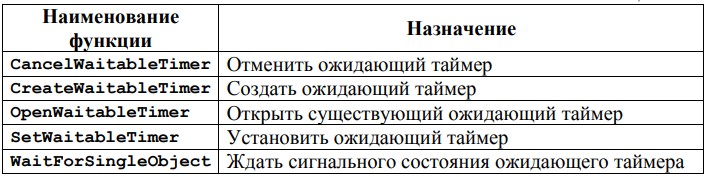
PulseEvent

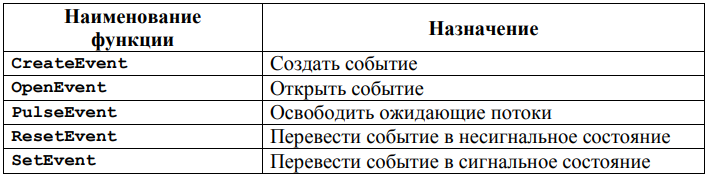
Квалификатор \_\_declspec(dllexport) применяется для обозначения экспортируемых dll-библиотекой функций.

**КОНТЕКСТ ПОТОКА (ВСЕ РЕСУРСЫ)**

Каждому потоку принадлежат следующие ресурсы: - код исполняемой функции; - набор регистров процессора; - область оперативной памяти; - стек для работы приложения; - стек для работы операционной системы; - маркер доступа, содержащий информацию для системы безопасности. Все эти ресурсы образуют так называемый контекст потока в Windows.

Все функции спрашивала 



Типо этих, просто название и значения, параметры не спрашивала