1. Что такое поток?

Поток — это наименьшая единица выполнения внутри процесса. Каждый поток представляет собой последовательность команд, которые выполняются независимо от других потоков в процессе, хотя они могут разделять ресурсы (память, файлы) с другими потоками того же процесса. Потоки позволяют программе выполнять задачи параллельно, что полезно для многозадачности, особенно на многоядерных процессорах.

2. Что такое Thread Local Storage (TLS)?

Thread Local Storage (TLS) — это область памяти, выделенная для каждого потока отдельно, что позволяет потокам хранить свои локальные данные, недоступные другим потокам. TLS используется, когда необходимо иметь уникальные данные для каждого потока, избегая при этом глобальных переменных и возможных конфликтов доступа.

3. Когда лучше использовать потоки вместо дополнительных процессов для многозадачности?

Потоки обычно используются, когда:

- Нужно разделять память и данные между задачами (потоки внутри одного процесса имеют доступ к общей памяти).

- Задачи тесно связаны между собой, и требуется высокоскоростной обмен данными.

- Важно минимизировать накладные расходы на создание и управление (создание потока быстрее, чем создание процесса).

- Применяется параллельная обработка данных, например, при выполнении вычислительных задач на многоядерных процессорах.

Использование дополнительных процессов может быть более подходящим, если задачи изолированы и требуется большая безопасность или независимость выполнения, так как каждый процесс имеет своё отдельное адресное пространство.

4. Какие функции WinAPI для работы с потоками вы знаете? POSIX?

**В WinAPI:**

- CreateThread — создает новый поток.

- ExitThread — завершает поток.

- TerminateThread — принудительно завершает поток (рекомендуется избегать).

- WaitForSingleObject / WaitForMultipleObjects — ожидают завершения потока.

- GetCurrentThread / GetCurrentThreadId — возвращают дескриптор или ID текущего потока.

- Sleep — приостанавливает выполнение потока на определенное время.

- TlsAlloc, TlsSetValue, TlsGetValue, TlsFree — управление TLS.

**В POSIX (pthread):**

- pthread\_create — создает новый поток.

- pthread\_exit — завершает поток.

- pthread\_join — ожидает завершения потока.

- pthread\_cancel — отправляет запрос на завершение потока.

- pthread\_detach — отделяет поток, позволяя ему освобождать ресурсы после завершения.

- pthread\_self — возвращает ID текущего потока.

- pthread\_key\_create, pthread\_setspecific, pthread\_getspecific, pthread\_key\_delete — управление TLS.

5. К каким возможным последствиям может привести выполнение потоком следующего кода?

pthread\_join(pthread\_self(), NULL);

или

WaitForSingleObject(GetCurrentThread(), INFINITE);

Оба вызова приведут к взаимоблокировке (deadlock), так как поток пытается ожидать завершения самого себя. В случае pthread\_join(pthread\_self(), NULL) поток будет ждать своего завершения, что невозможно, так как он находится в активном состоянии. Точно так же WaitForSingleObject(GetCurrentThread(), INFINITE) заблокирует поток на неопределенное время, ожидая, что он завершится. Эти ошибки приводят к бесконечному ожиданию и могут вызвать зависание программы.