1. Основные глобальные объекты Node.js включают:

* global: Это глобальный объект, который предоставляет доступ к другим глобальным объектам и функциям в Node.js. Он аналогичен объекту window в браузере.
* process: Этот объект предоставляет информацию и управление текущим процессом Node.js. Он содержит свойства и методы, которые позволяют взаимодействовать с окружением, а также управлять процессом выполнения.
* console: Этот объект предоставляет методы для вывода информации в консоль. Он позволяет выводить сообщения об ошибках, отладочные сообщения и другую информацию.
* module: Этот объект представляет текущий модуль Node.js. Он содержит информацию о модуле, его экспортируемых функциях и переменных.
* require: Этот объект используется для загрузки модулей в Node.js. Он позволяет импортировать функции и переменные из других модулей.

1. Асинхронная функция - это функция, которая выполняется асинхронно, то есть не блокирует выполнение программы и позволяет продолжать работу сразу после вызова функции, не дожидаясь ее завершения. Вместо этого, асинхронная функция запускает задачу в фоновом режиме и возвращает промис или использует обратный вызов для обработки результата, когда задача завершится.

Асинхронные функции особенно полезны при выполнении операций ввода-вывода (I/O), таких как чтение файлов, отправка запросов к базе данных или выполнение сетевых запросов. Они позволяют программе продолжать работу, пока выполняются эти операции, и обрабатывать результаты, когда они станут доступны.

1. Стандартные системные потоки - это предопределенные потоки ввода-вывода, которые доступны в операционной системе. В Node.js они представлены следующими объектами:

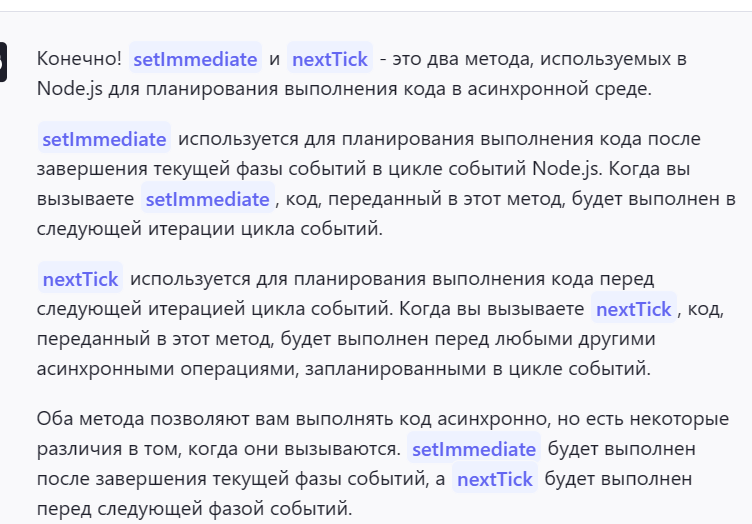
* process.stdin: Поток стандартного ввода (stdin), который позволяет программе получать данные из консоли или другого источника ввода.
* process.stdout: Поток стандартного вывода (stdout), который позволяет программе выводить данные в консоль или другой источник вывода.
* process.stderr: Поток стандартного вывода ошибок (stderr), который позволяет программе выводить сообщения об ошибках или другую отладочную информацию.

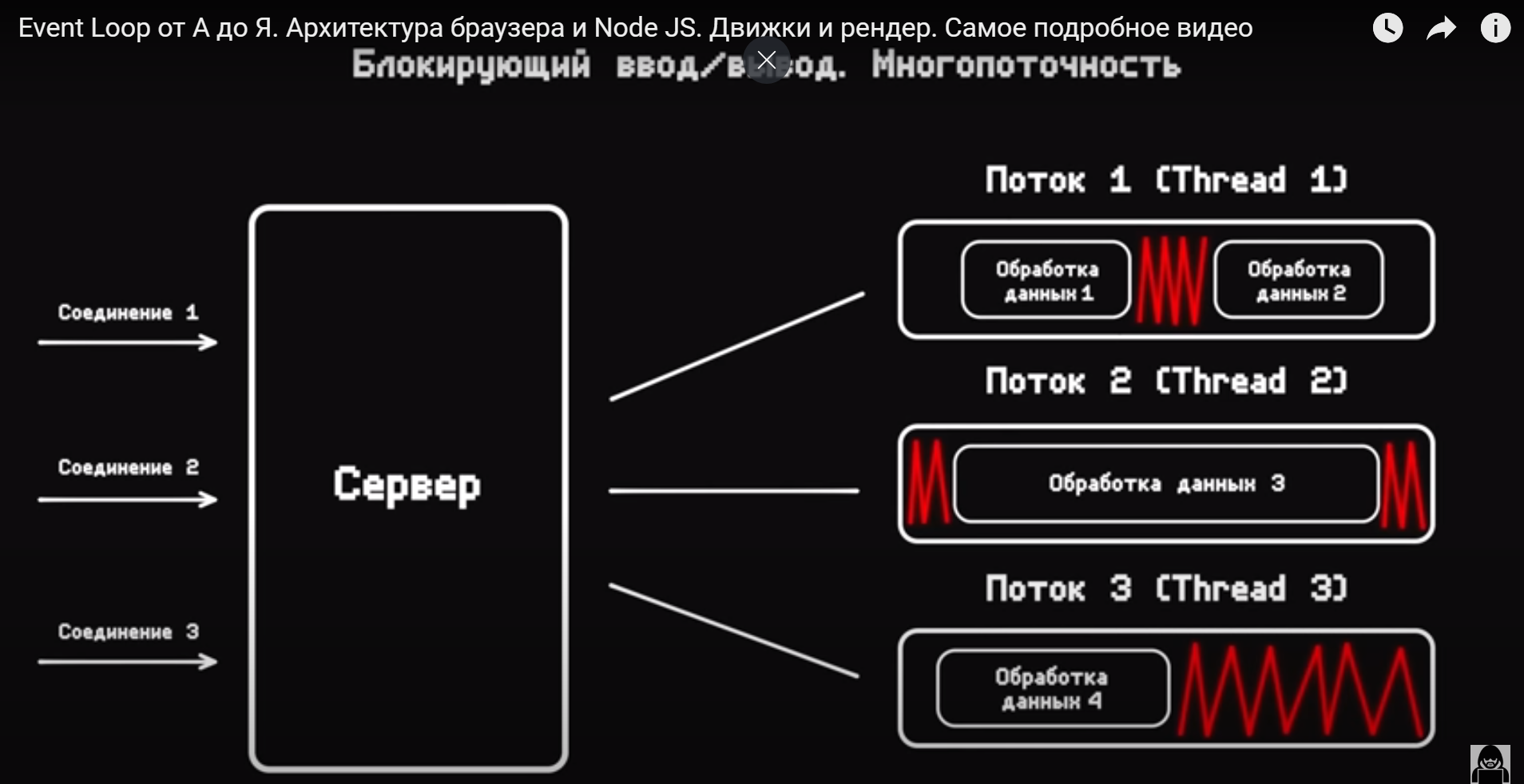
Эти потоки могут быть использованы для чтения ввода пользователя, вывода результатов работы программы или вывода сообщений об ошибках.

1. Функция process.nextTick используется для планирования выполнения функции в конце текущей итерации цикла событий. Она позволяет выполнить функцию до следующей итерации цикла событий, что делает ее особенно полезной для выполнения задач с высоким приоритетом.

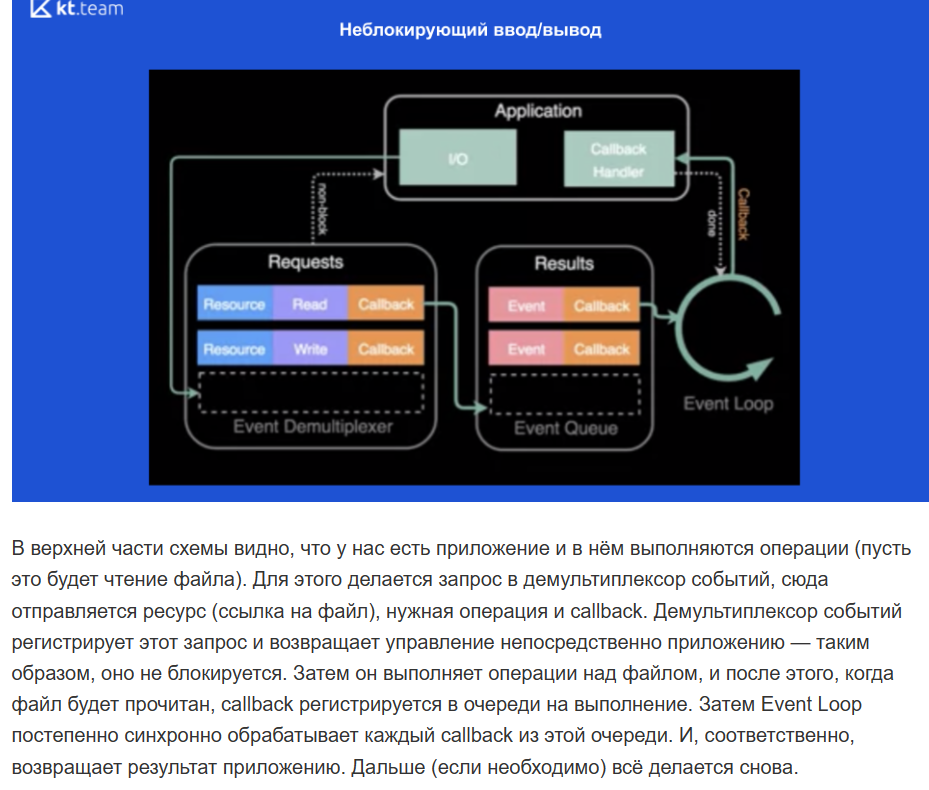
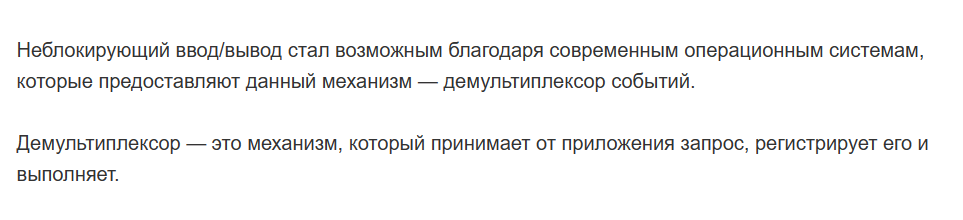
Функция setImmediate также используется для планирования выполнения функции, но она планирует ее выполнение в начале следующей итерации цикла событий. Это позволяет выполнить функцию после завершения текущей итерации цикла событий.

Разница между process.nextTick и setImmediate заключается в порядке выполнения. process.nextTick выполняет функцию перед любыми другими запланированными операциями, в то время как setImmediate выполняет функцию после выполнения всех операций, запланированных на текущую итерацию цикла событий.

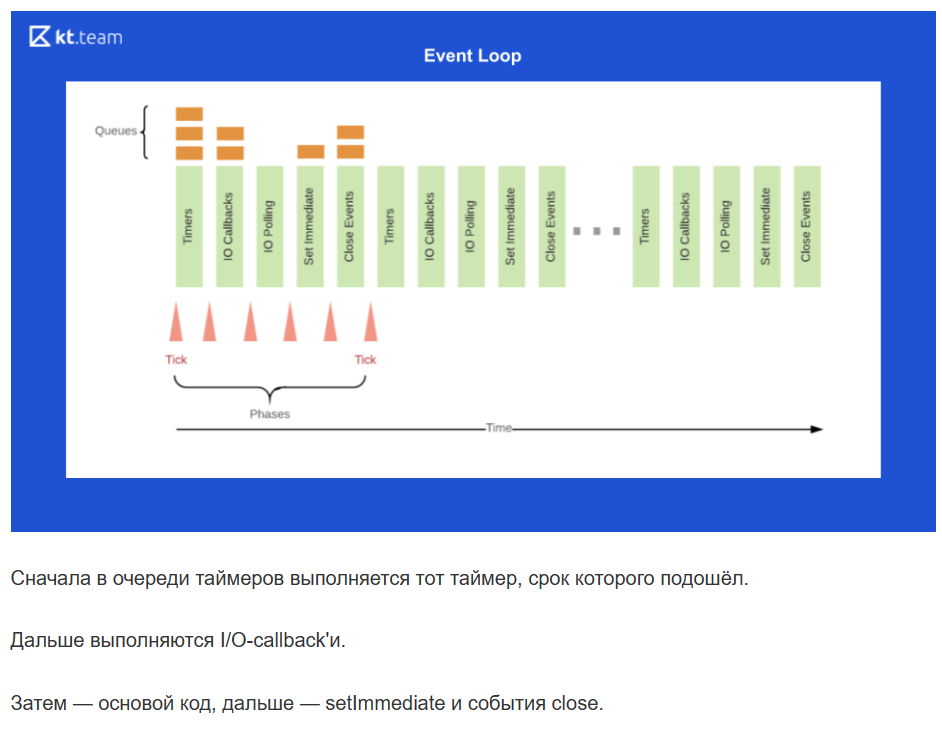
****

****

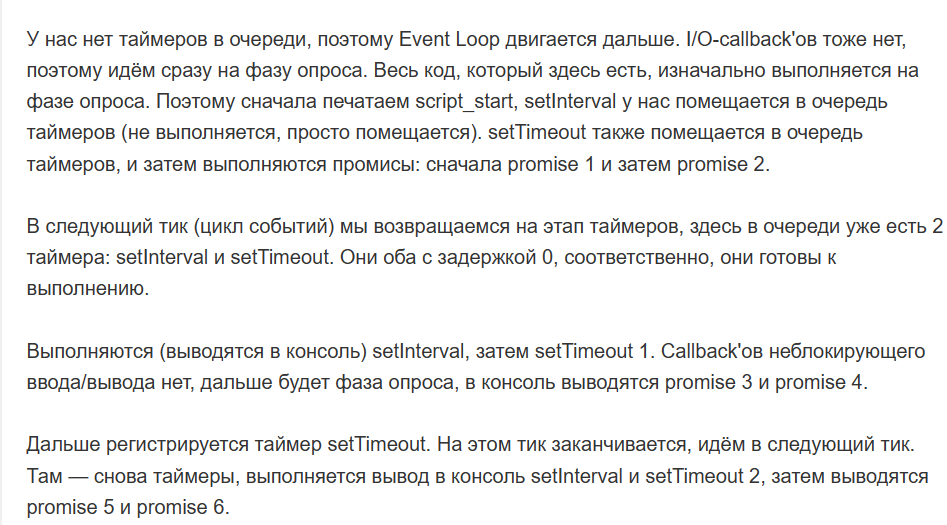
****

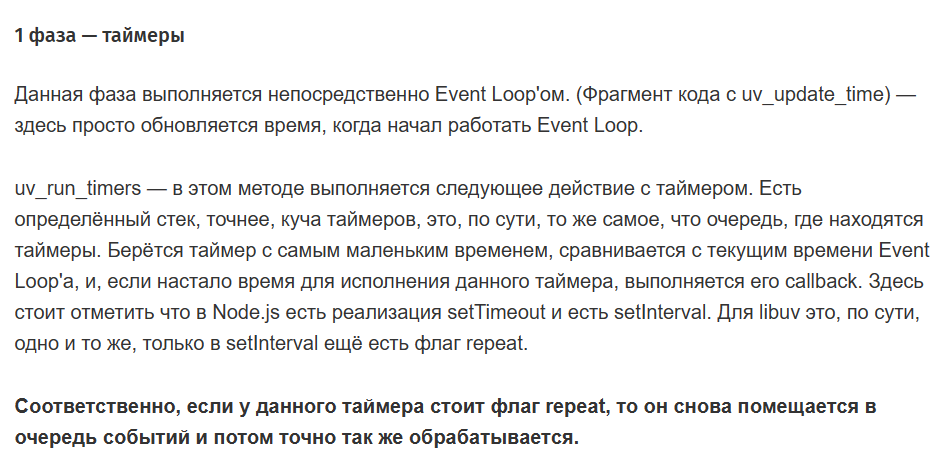
****

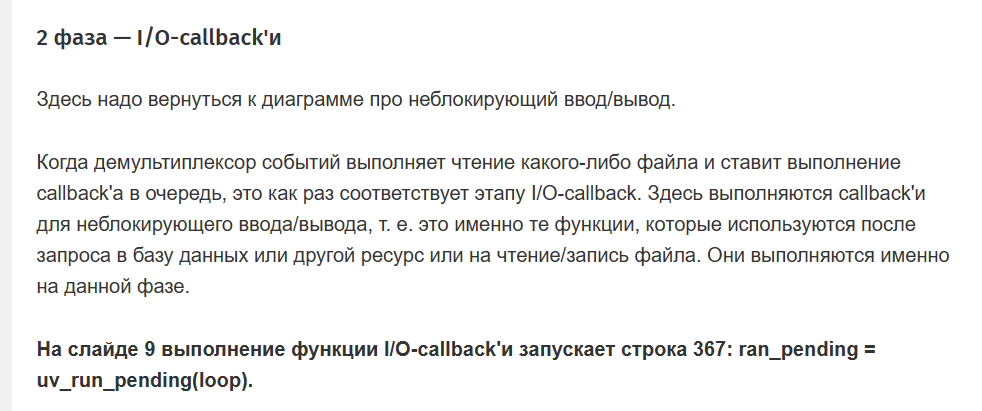
****

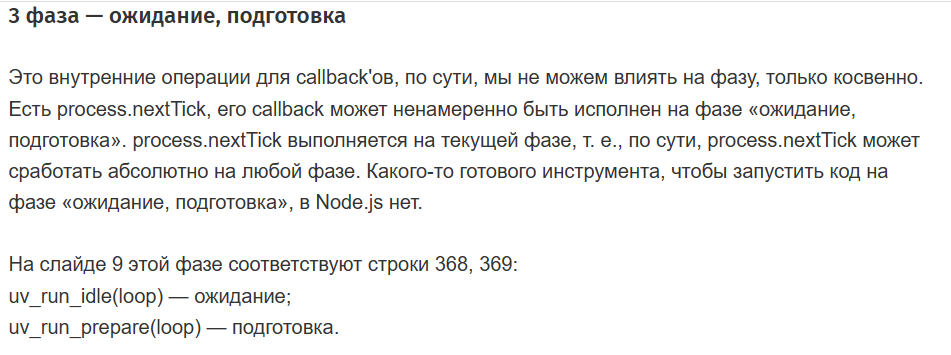
****

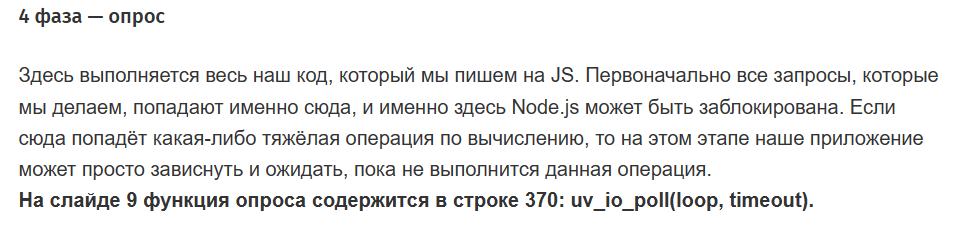
****

****

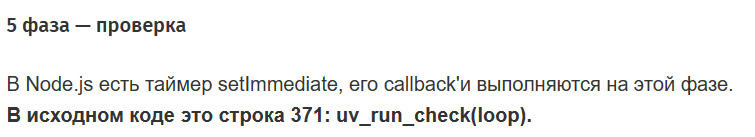
****

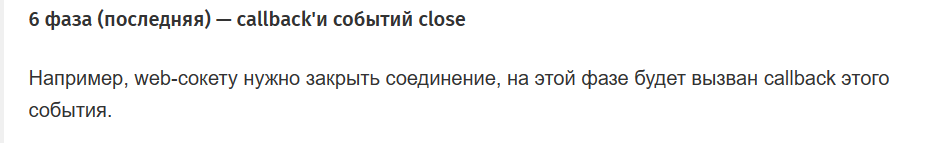
****

****

****

**+ на этой фазе происходит получение новых событий ввода/вывода**

****

****