

1. Дайте определение понятию «Интернет».

Интернет – это всемирная компьютерная сеть, объединяющая между собой компьютеры, предназначенная для хранения, обработки и передачи информации через стандартизированные протоколы, такие как TCP/IP.

2) стандарты Internet (RFC, STD); 3) службы Интернет (DNS, SMTP/POP3/IMAP, WWW, FTP, Telnet, SSH,…); 4) организации, управляющие сетью Internet (ISOC, IETF, ICANN, IANA, W3C, …).

1. Дайте определение понятию «Служба Интернет».

Служба интернет - это системы, предоставляющие услуги пользователям Интернета(DNS, SMTP/POP3/IMAP, WWW, FTP, Telnet, SSH,…);

1. Дайте определение понятию «Узел сети Интернет».

Устройство, имеющее IP-адрес и подключенное сети Интернет (обычно к сети Интернет-провайдера). Каждый узел характеризуется своей программно-аппаратной платформой – аппаратурой и операционной системой.

1. Дайте определение понятию «клиент-серверное приложение».

Клиент-серверное приложение = приложение (программа, архитектурный шаблон) с клиент-серверной архитектурой: приложение, состоящее из двух компонент – клиента и **сервера**; клиент и сервер взаимодействуют между собой в соответствии с заданными правилами (спецификациями, **протоколами**); для взаимодействия между клиентом и сервером в соответствии с правилами (спецификацией, протоколом) должно быть установлено **соединение**; **инициатором соединения всегда является клиент**.

1. Дайте определение понятию «сетевой протокол».

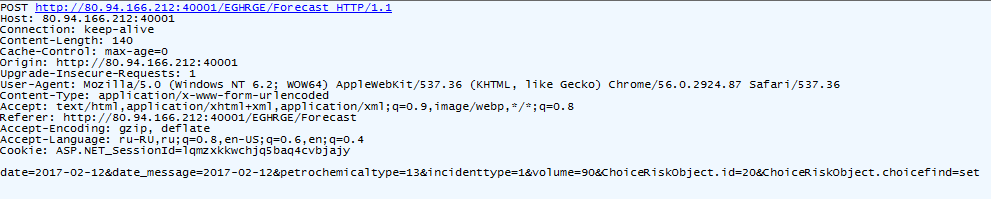
Сетевой протокол - набор правил, определяющий порядок взаимодействия устройств в сети (просто протокол – определяет порядок взаимодействия двух соседних уровней модели iso/osi).

1. Перечислите основные свойства протокола HTTP.

* версии HTTP/1.1 – действующий (текстовый), HTTP/2 – черновой (не распространен, бинарный), HTTP/3 (раньше HTTP-over-QUIC, стадия черновика);
* два типа абонентов: клиент и сервер;
* два типа сообщений: request и response;
* от клиента к серверу – request;
* от сервера к клиенту – response;
* на один request всегда один response, иначе ошибка;
* одному response всегда один request, иначе ошибка;
* stateless;
* TCP-порты: 80, 443;
* для адресации используется URI или URL;
* поддерживается W3C, описан в нескольких RFC.

1. Перечислите состав информации, пересылаемой в HTTP-запросе.

* метод;
* URI;
* версия протокола (HTTP/1.1);
* заголовки (пары: имя/заголовок);
* *параметры (пары: имя/заголовок);*
* расширение.



1. Перечислите состав информации, пересылаемой в HTTP-ответе.

* версия протокола (HTTP/1.1);
* код состояния (1xx, 2xx, 3xx, 4xx, 5xx);
* пояснение к коду состояния;
* заголовки (пары: имя/заголовок);
* расширение.



1. Перечислите группы заголовков HTTP.

* **1xx**: информационные сообщения;
* **2xx**: успешный ответ;
* **3xx**: переадресация;
* **4xx**: ошибка клиента;
* **5xx**: ошибка сервера.

1. Дайте определение понятию «web-приложение».

Клиент-серверное приложение, у которого клиент и сервер взаимодействуют по протоколу HTTP называется **web-приложением**.

1. Дайте определение понятиям «frontend» и «backend».

Когда говорят о разработке web-приложения, говорят о разработке **frontend** (клиента) и **backend** (сервера)

1. Дайте определение понятию «кроссплатформенное приложение».

**Кроссплатформенное приложение**: приложение, способное работать на более чем одной программно-аппаратной (аппаратура + операционная система) платформе. Кроссплатформенность может быть достигнута различными способами: 1) на уровне компилятора (С, С++); 2) на уровне среды (фреймворка) исполнения (Java/JVM, C#/.NET CORE/CLR, JS/Node.js/V8).

1. Изобразите и поясните общую схему web-приложения.

Общая схема web-приложения включает в себя клиентскую часть (frontend), которая взаимодействует с пользователем через браузер, и серверную часть (backend), которая обрабатывает запросы, взаимодействует с базой данных и обеспечивает бизнес-логику. Клиент и сервер обмениваются данными через HTTP-запросы и HTTP-ответы.



1. Назовите основные технологии разработки серверных кроссплатформенных приложений.

1) на уровне компилятора (С, С++); 2) на уровне среды (фреймворка) исполнения (Java/JVM, C#/.NET CORE/CLR, JS/Node.js/V8).

1. Поясните понятие «асинхронная операция».

Операция называется асинхронной, если ее выполнение осуществляется в 2 фазы: 1) заявка на исполнение; 2) получение результата; при этом участвуют два механизма: A-механизм, формирующий заявку и потом получающий результат; B-механизм, получающий заявку от A, исполняющий операцию и отправляющий результат A; продолжительность исполнения операции B-механизмом, как правило, непредсказуемо; в то время пока B-механизм исполняет операцию, А-механизм выполняет собственную работу. Применение асинхронности не противоречит применению многопоточности.

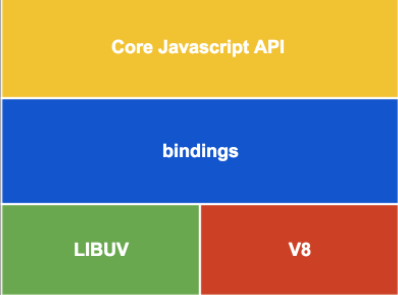
Это операция или задача, которая выполняется независимо от основного потока выполнения программы. В асинхронной модели выполнения операции могут запускаться параллельно или в фоновом режиме, что позволяет программе эффективно использовать ресурсы и не блокировать основной поток.

1. Поясните основное назначение NODE.JS.

Программная платформа для разработки серверных web-приложений на языке JS/V8.

1. Перечислите основные свойства NODE.JS.
2. основные свойства:

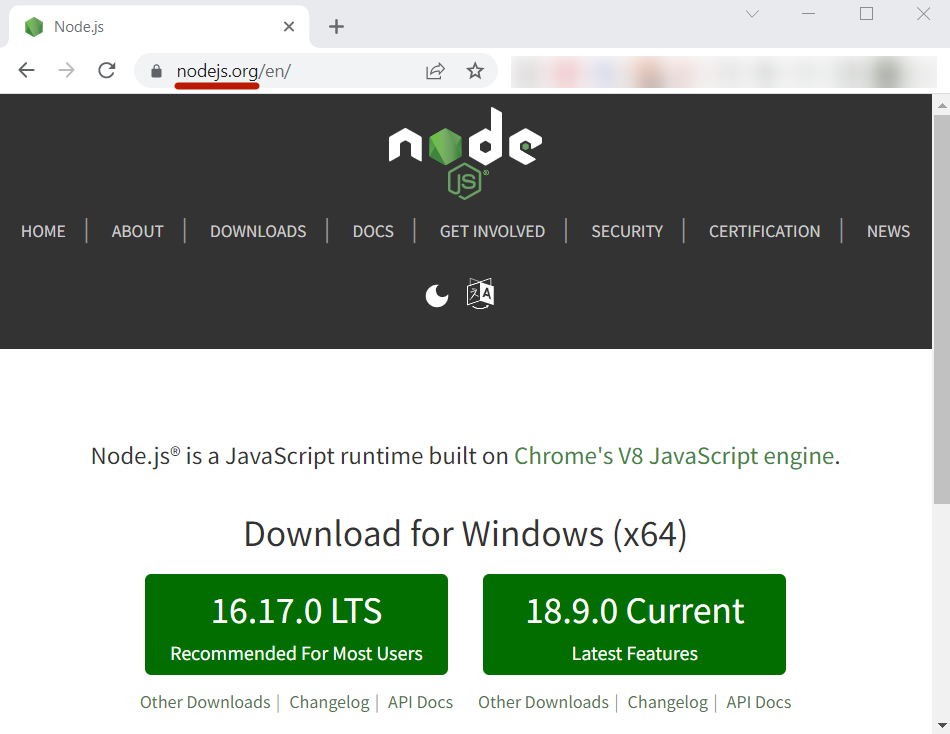
* основан на **Chrome V8**;
* **среда (контейнер) исполнения** приложений на JavaScript;
* поддерживает механизм **асинхронности**;
* ориентирован на **события**;
* **однопоточный** (код приложения исполняется только в одном потоке, один стек вызовов); обычно в серверах для каждого соединения создается свой поток, в Node.js все соединения обрабатываются в одном JS-потоке;
* **не блокирует** выполнение кода при вводе/выводе (в файловой системе до 4х одновременно);
* в состав Node.js входят инструменты: **npm** – пакетный менеджер; **gyp** - Python-генератор проектов; **gtest** – Google фреймворк для тестирования С++ приложений;
* использует библиотеки: **V8** – библиотека V8 Engine, **libuv** – библиотека для абстрагирования неблокирующих операций ввода/вывода (представляет собой обертку над epoll, kqueue, IOCP); **llhttp** – легковесный парсер http-сообщений (написан на C и не выполняет никаких системных вызовов); **c-ares** -библиотека для работы с DNS; **OpenSSL** – библиотека для криптографии; **zlib** – сжатие и распаковка.



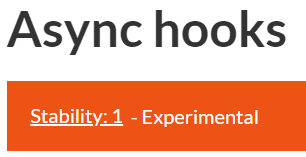
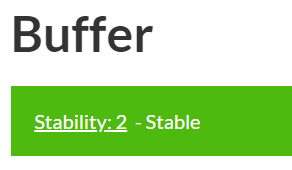
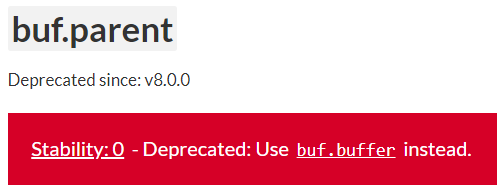
* разработчик: **Райан Дал**:

****

* первая версия: **2009 г**.;
* стабильные версии: с **2015 г., Node.js 4.0.0;**
* официальный сайт:



* основная сфера применения: **разработка web-серверов**;
* версионирование: **две ветки** 16.x.x – версии длительной поддержки (LST, Long Term Support), 18.x.x – нестабильные версии, включающие последние разработки (Current).
* документация: <https://nodejs.org/api/>
* Stability Index - Deprecated (0), Experimental (1), Stable (2), Locked (3)

* **Асинхронность**: Поддержка асинхронного программирования с использованием колбэков или обещаний (promises).
* **Высокая производительность**: Возможность обработки большого количества одновременных подключений и запросов.
* **Модульность**: Поддержка модульной структуры кода с использованием CommonJS.
* **Скорость выполнения кода**: Использование движка JavaScript V8 от Google.
* **Широкое сообщество и экосистема**: Богатая библиотека модулей и фреймворков для разработки.