1. Дайте определение понятию «Интернет».

Сеть на основе TCP/IP.

1. Дайте определение понятию «Служба Интернет».

Это виды услуг, которые оказываются серверами сети Internet (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».

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».

Набор соглашений интерфейса логического уровня, которые определяют обмен данными между различными программами. Эти соглашения задают единообразный способ передачи сообщений и обработки ошибок.

1. Перечислите основные свойства протокола HTTP.

версии HTTP/1.1 – действующий (текстовый), HTTP/2 – черновой (не распространен, бинарный);

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;

TCP-порты: 80, 443;

для адресации используется URI или URN;

поддерживается W3C, описан в нескольких RFC.

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

Запросы содержат следующие элементы:

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

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

Ответы содержат следующие элементы:

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

1. Дайте определение понятию «web-приложение».

Клиент-серверное приложение, у которого клиент и сервер взаимодействуют по протоколу HTTP.

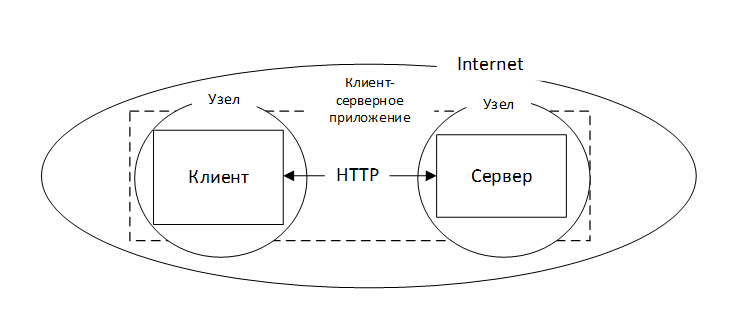
1. Дайте определение понятиям «frontend» и «backend».

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

1. Дайте определение понятию «кроссплатформенное приложение».

Приложение, способное работать на более чем одной программно-аппаратной (аппаратура + операционная система) платформе.

1. Изобразите и поясните общую схему web-приложения.



1. Назовите основные технологии разработки серверных кроссплатформенных приложений.

Кроссплатформенность может быть достигнута различными способами: 1) на уровне компилятора (С, С++); 2) на уровне среды (фреймворка) исполнения (Java/JVM, C#/.NET CORE, 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.

1. Поясните принцип выполнения асинхронного запроса с помощью объекта XMLHTTPRequest и Fetch.
2. Поясните основное назначение сервера NODE.JS.

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

1. Перечислите основные свойства сервера NODE.JS.

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; **http-parser** – 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**;
* четные версии Node.js –версии длительной поддержки (LST), нечетные версии – нестабильные версии, включающие последние разработки.