|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Теоретической и прикладной математики | | |
|  | | |
| Лабораторная работа № 1 | | |
| по дисциплине «ОСНОВЫ WEB-ПРОГРАММИРОВАНИЯ» | | |
|  | | |
| **HTTP-запросы** | | |
|  | | |
|  |
| Факультет: | ПМИ |
| Группа: | ПМИ-02 |
| Студент: | Сидоров Даниил |
|  |  |
| Преподаватель: | Цыгулин Алексей Александрович. |
|  |  |
|
|  |  |
|  | | |
| Новосибирск | | |
| 2022 | | |

1. **Цель р****а****боты**

Познакомиться с протоколом HTTP, увидеть внутреннюю структуру обмена клиента и сервера.

1. **Задание 1. Telnet**

HTTP – текстовый протокол взаимодействия. Браузер работает аналогично терминалу связи. Мы используем сетевой протокол **telnet** для того, чтобы увидеть, что происходит при взаимодействии клиента и сервера. Для обмена данными между сервером и браузером воспользуемся сайтом международного института компьютерных технологий [**http://www.iict.ru/**](http://www.iict.ru/).

В командной строке выполним команду **telnet iict.ru 80**:



В открывшемся окне терминала набираем следующее:

**GET / HTTP/1.1**

**Host: iict.ru**

После отправки запроса получили ответ:



Разберем полученный ответ:

**HTTP/1.1 200 OK** - код состояния и пояснение к нему.

**Date: Sat, 01 Oct 2022 06:53:29 GMT** - дата создания ответа от сервера.

**Content-Type: text/html; charset=utf-8** - указание типа текстовых данных и используемой кодировки.

**Transfer-Encoding: chunked** - механизм передачи данных в протоколе передачи гипертекста, надёжно доставляющий данные от сервера клиенту.

**Connection: keep-alive** - использование одного TCP-соединения для отправки и получения многократных HTTP-запросов и ответов.

**Keep-Alive: timeout=10** - как долго сервер готов держать соединение открытым, прежде чем принудительно закроет соединение.

**Vary: Accept-Encoding -** информирует о поведении сервера в отношении кэширования представления запрошенного ресурса.

**Server:** **Apache -** используемый веб-сервер.

**Set-Cookie: -** заголовки, которые используются для отправки cookies с сервера.

**P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM "** – заголовок, позволяющий сайту информировать браузер о предполагаемом получении личных данных пользователя.

**Cache-Control: private, no-store, no-cache, must-revalidate, post – check=0, pre – check=0** - заголовок используется для задания инструкций кеширования как для запросов, так и для ответов.

**private** - ответ сервера является специфическим для конечного пользователя и не должен кэшироваться различными промежуточными прокси

**no-store** - кэш не должен хранить никакую информацию о запросе и ответе.

**no-cache** - необходимость отправить запрос на сервер для валидации ресурса перед использованием закэшированных данных.

**must-revalidate** - кэш должен проверить статус устаревших ресурсов перед их использованием).

**Post - check=0** - Определяет интервал времени в секундах, после которого ресурс должен быть проверен на актуальность.

**Pre - check=0** - Определяет интервал времени в секундах, после которого проверка актуальности ресурса должна быть произведена перед его отображением для пользователя.

**Expires: Wed, 17 Aug 2005 00:00:00 GMT** - заголовок, который содержит дату/время, по истечении которой ответ сервера считается устаревшим.

**Pragma: no-cache** - аналогично заголовку cache-control: no-cache.

**Last-modified:** **Sat, 01 Oct 2022 06.53.29 GMT** - дата последних изменений.

1. **Задание 2. Браузер**

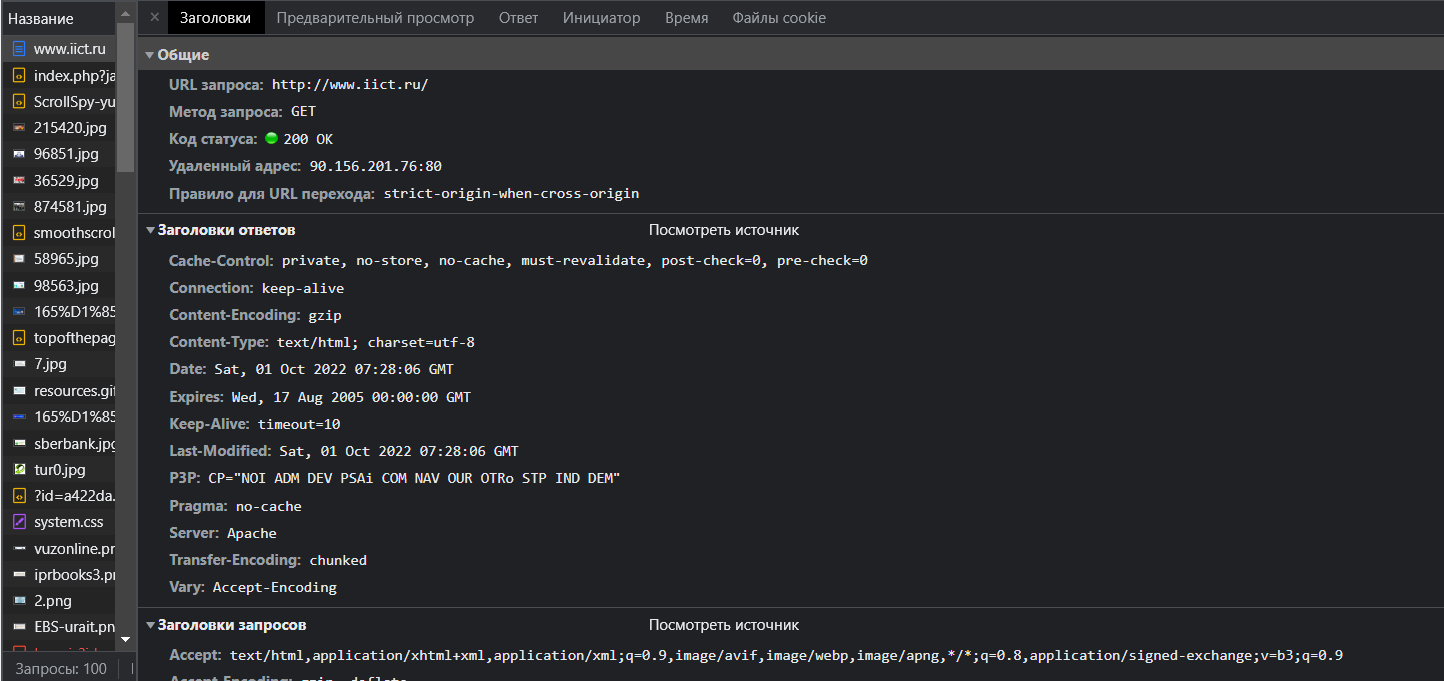
Проделаем предыдущее задание с использованием браузера

**Chrome**. Используем вкладку «Сеть» (Network) в средствах

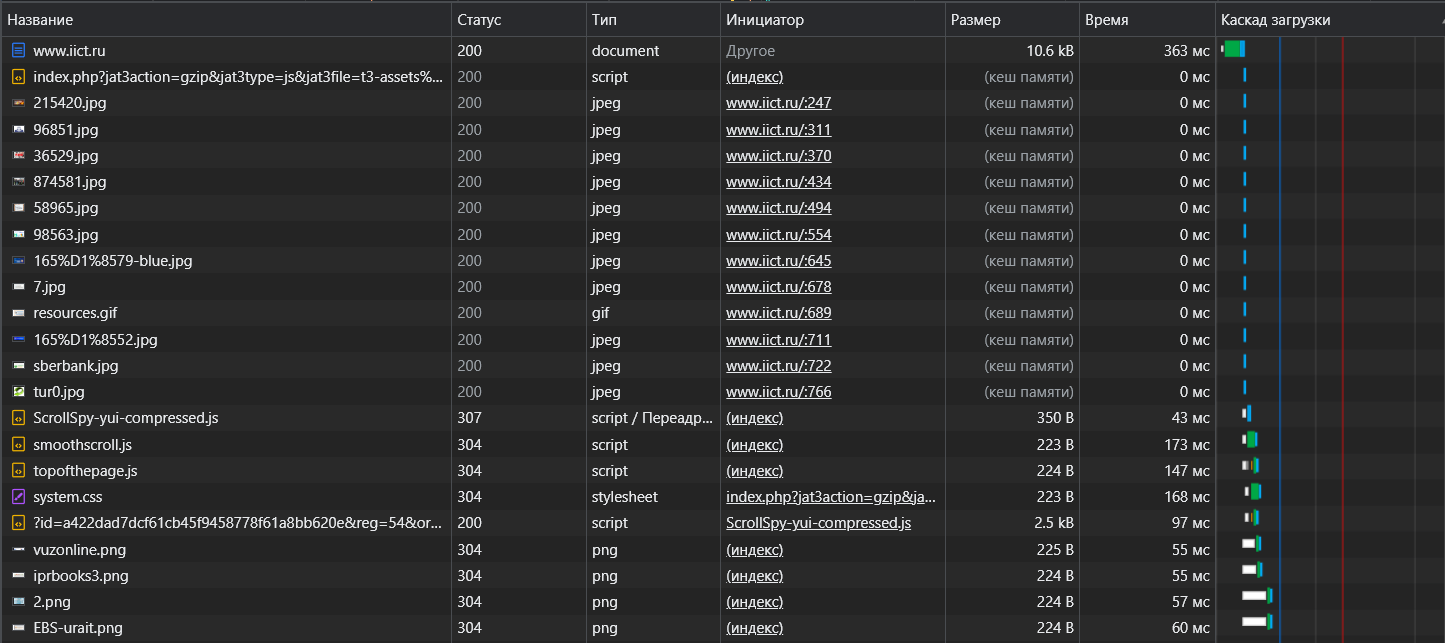
разработки для того, чтобы отследить передаваемый запрос и ответ на

него.

Откроем сайт **iict.ru** и посмотрим заголовки ответов.



Все заголовки совпадают с ранее полученными. Так же появился новый заголовок **Content-Encoding: gzip -** согласование того, какой формат кодирования использовать для передачи тела.



В самом начале загрузился документ **www.iict.ru** за 363 мс, его размер составил 10.6 kB, статус документа – 200.

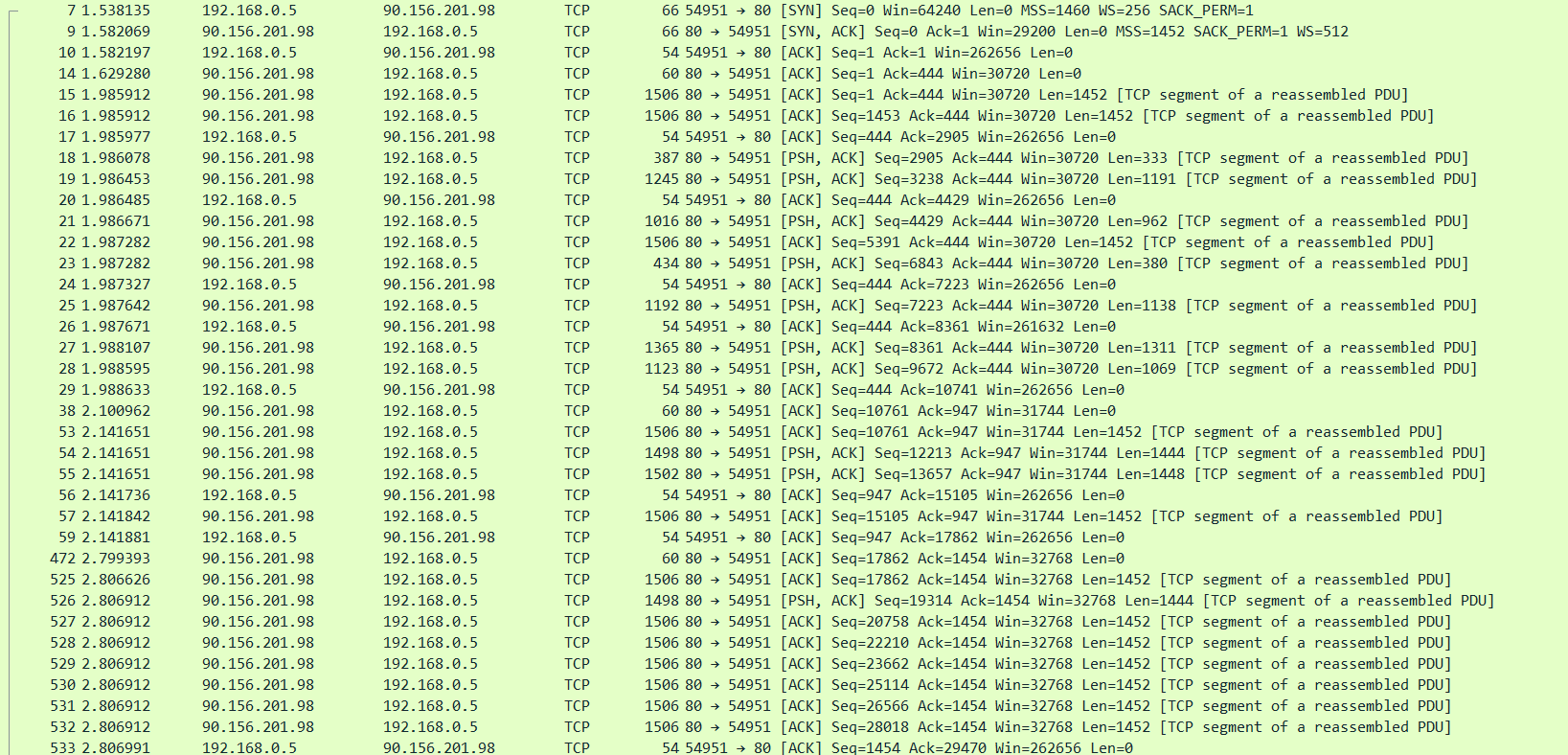
Далее имеем много запросов типа **script**, **jpeg/gif/png**, а также **stylesheet** – это запросы для загрузки файлов **JS**, **картинок** и **CSS** соответственно. Загрузка запросов типа **script** заняло от 0 до 173 мс, загрузка **картинок** от 0 до 60 мс, загрузка **таблицы стилей** 168 мс.

1. **Задание 3. Wireshark**

Для выполнения задания запустим программу **wireshark**, выберем активный интерфейс **Ethernet**, включим запись сетевого трафика, зайдем на сайт **iict.ru** и выключим запись.

Видим пакеты с протоколом **TCP**. Изначально идут пакеты, направленные к серверу, а после от сервера к браузеру. Время выполнения указано во 2 колонке (от 1 до 3 мс).

Пакет **TCP** — это блок данных, который передаётся в потоке. Порядок пакетов строго определён и не может быть нарушен. Задача протокола **TCP** - управлять отправкой данных и следить за тем, чтобы они были гарантированно приняты получателем. **TCP** разбивает данные на пакеты и нумерует их.



Если последуем за **TCP** потоком, получим информацию, схожую с заданием 1 и 2.



Далее видим пакеты с протоколом **HTTP**. Сначала произошел **http** запрос, далее загрузились запросы типа **png/jpg, css, js**. Время выполнения указано во 2 колонке (от 1 до 3.2 мс). После выполнения каждого запроса получаем ответ 200.



Если мы последуем за **HTTP** потоком, то увидим такие же результаты, как и в прошлых заданиях:



1. **Вывод**

В ходе выполнения заданий познакомились с протоколами HTTP и TCP с помощью **telnet**, **панели разработчика** **браузера** и **wireshark**.