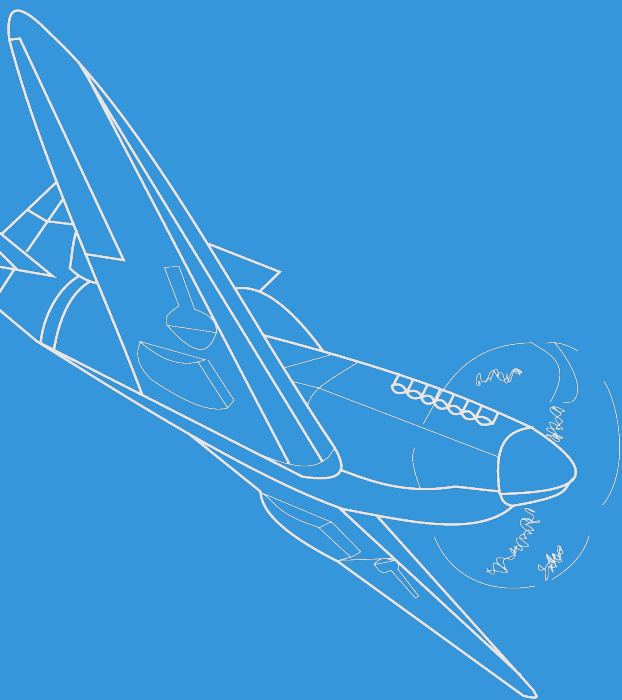


Поисковый робот

на которой расскажут, что такое поисковый робот, как обкачивать веб, как есть подводные камни и про robots.txt тоже расскажут.

Содержание занятия

1. URL, URN, URI
2. Протокол HTTP
3. Поисковый робот
4. Домашнее задание;

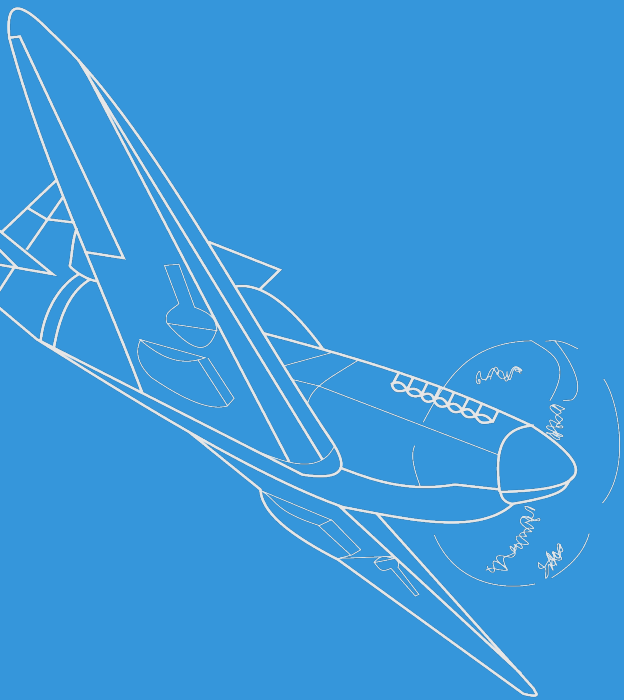


Документы



Документы могут быть:

- Статические
 - Это файлы на дисках сервера;
 - Как правило, обладают постоянным адресом.
- Динамические
 - Создаются на каждый запрос;
 - Содержимое зависит от времени и пользователя;
 - Адрес может быть постоянным или меняться.



URI, URL, URN





- **URI** — Uniform Resource Identifier (унифицированный идентификатор ресурса);
- **URL** — Uniform Resource Locator (унифицированный локатор/указатель ресурса);
- **URN** — Uniform Resource Name (унифицированное имя ресурса).

URI является либо URL, либо URN, либо одновременно обоими.

URN — uniform resource name



`urn:<NID>:<NSS>`

- `<NID>` — идентификатор пространства имён;
- `<NSS>` — строка из определённого пространства имён.

Пример:

- `urn:isbn:5170224575` — URN книги, идентифицируемой номером ISBN;

URL — uniform resource locator



`<схема>://[[<логин>[:<пароль>]@]<хост>[:<порт>]][/<URL -
путь>][?<параметры>][#<якорь>]`

`http://server.org:8080/path/doc.html?a=1&b=2#part1`

- `http` — протокол;
- `server.org` — DNS имя сервера (может указываться ip-адрес машины);
- `8080` — TCP порт;
- `/path/doc.html` — путь к файлу;
- `a=1&b=2` — параметры запроса;
- `part1` — якорь, положение на странице.

Абсолютные и относительные URL



- `http://server.org/1.html` — абсолютный;
- `//server.org/1.html` — абсолютный (schemeless);
- `/another/page.html?a=1` — относительный (в пределах домена);
- `pictures/cat.png` — относительный (от URL текущего документа);
- `?a=1&b=2` — относительный (от URL текущего документа);
- `#part2` — относительный (в пределах текущего документа);

Правила разрешения URL



`https://site.com/path/page.html` — основной документ

- `http://wikipedia.org` = `http://wikipedia.org`
- `//cdn.org/jquery.js` = `https://cdn.org/jquery.js`
- `/admin/index.html` = `https://site.com/admin/index.html`
- `another.html` = `https://site.com/path/another.html`
- `?full=1` = `https://site.com/path/page.html?full=1`
- `#chapter2` = `https://site.com/path/page.html#chapter2`



HTTP-протокол

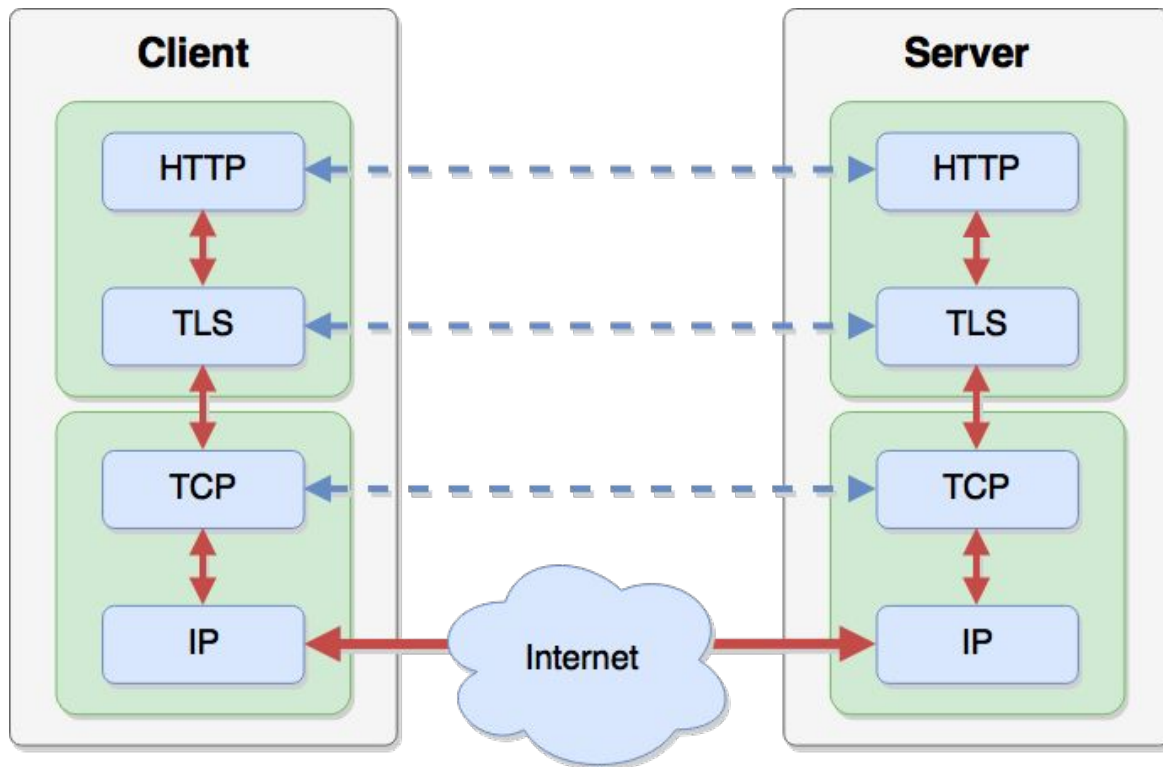
Особенности, методы, версии, заголовки.

Как задачи решает HTTP?



- Передача документов;
- Передача мета-информации;
- Авторизация;
- Поддержка сессий;
- Кеширование документов;
- Согласование содержимого (negotiation);
- Управление соединением.

Как происходит HTTP запрос?



Ключевые особенности HTTP



- Работает поверх TCP/TLS;
- Протокол запрос-ответ;
- Не поддерживает состояние (соединение) — stateless;
- Текстовый протокол;
- Расширяемый протокол.

HTTP запрос состоит из



- строка запроса:
 - метод,
 - URL документа,
 - версия.
- заголовки;
- тело запроса;

HTTP/1.0 запрос



```
GET http://www.ru/robots.txt HTTP/1.0
Accept: text/html, text/plain
User-Agent: telnet/hands
If-Modified-Since: Fri, 24 Jul 2015 22:53:05 GMT
```

Перевод строки — `\r\n`

HTTP/1.1 запрос



```
GET /robots.txt HTTP/1.1
Accept: text/html,application/xhtml+xml
Accept-Encoding: gzip, deflate
Cache-Control: max-age=0
Connection: keep-alive
Host: www.ru
User-Agent: Mozilla/5.0 Gecko/20100101 Firefox/39.0
```

HTTP/1.1 ответ



HTTP/1.1 404 Not Found

Server: nginx/1.5.7

Date: Sat, 25 Jul 2015 09:58:17 GMT

Content-Type: text/html; charset=iso-8859-1

Connection: close

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

<HTML><HEAD>...

Методы HTTP-запроса



- GET — получение документа;
- HEAD — получение только заголовков;
- POST — отправка данных на сервер;
- PUT — отправка документа на сервер;
- DELETE — удаление документа;
- CONNECT, TRACE, OPTIONS — используются редко;
- COPY, MOVE, MKCOL — расширения WebDAV.

HTTP-коды ответов



- 1xx — информационные;
- 2xx — успешное выполнение;
- 3xx — перенаправления;
- 4xx — ошибка на стороне клиента;
- 5xx — ошибка на стороне сервера.

HTTP-коды ответов (1)



200 OK — запрос успешно выполнен;

204 No Content — запрос успешно выполнен, но документ пуст;

301 Moved Permanently — документ сменил URL;

302 Found — повторить запрос по другому URL;

304 Not Modified — документ не изменился, использовать кеш.

HTTP-коды ответов (2)



400 Bad Request — неправильный синтаксис запроса;

401 Unauthorized — требуется авторизация;

403 Forbidden Moved Permanently — нет доступа (неверная авторизация);

404 Not Found — документ не найден;

418 I'm a teapot

500 Internal Server Error — неожиданная ошибка сервера;

502 Bad Gateway — проксируемый сервер отвечает с ошибкой;

504 Gateway Timeout — проксируемый сервер не отвечает;

Заголовки HTTP (общие)



Для управления соединением и форматом сообщения (документа):

- `Content-Type` — mime-тип документа;
- `Content-Length` — длина сообщения;
- `Content-Encoding` — кодирование документа, например, gzip-сжатие;
- `Transfer-Encoding` — формат передачи, например, chunked;
- `Connection` — управление соединением;
- `Upgrade` — смена протокола.

Заголовки HTTP-запросов

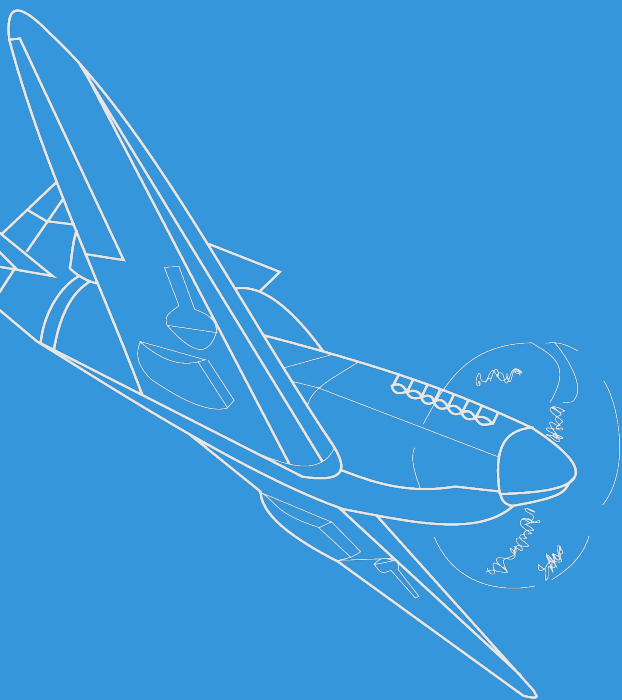


- **Authorization** — авторизация, чаще всего логин/пароль;
- **Cookie** — передача состояния (сессии) на сервер;
- **Referer** — URL предыдущего документа, контекст запроса;
- **User-Agent** — описание web-клиента, версия браузера;
- **If-Modified-Since** — условный GET запрос;
- **Accept-*** — согласование (negotiation) содержимого.

Заголовки HTTP-ответов



- `Location` — новый URL документа при перенаправлениях (коды 301, 302);
- `Set-Cookie` — установка состояния (сессии) в браузере;
- `Last-Modified` — дата последнего изменения документа;
- `Date` — Дата на сервере, для согласования кешей;
- `Server` — описание web-сервера, название и версия.



Поисковый робот



- Без поиска сложно найти нужный контент,
- Без поиска нет стимула создавать новый контент,
 - Зачем что-то публиковать, если никто не сможет прочитать?
 - Зачем что-то публиковать, если нельзя заработать на рекламе?
- Кто-то должен платить за веб,
 - Сервера, инфраструктура веб, создание контента.

Какого размера веб?



- Сложности
 - Веб на самом деле бесконечный
 - Динамический контент
 - Статический веб содержит синтаксическое дублирование, в основном, зеркала (30%)
 - Некоторые сервера редко или плохо доступны
- Кого это заботит?
 - Медиа, и соответственно, пользователей
 - Архитектуру поисковой системы
 - Алгоритмы обкатки, влияние на полноту

Новое определение



- Статический индексируемый веб - это есть индекс поисковой системы
- Различные поисковые системы имеют различные характеристики
- Различные поисковые системы индексируют различные части одного и того же URL:
 - frames, meta-keywords, document restrictions, document extensions, ...

Основные задачи поискового робота



- Изначально имеет набор известных URL;
- Скачать и распарсить страницы с них:
 - Извлечь ссылки со страницы;
 - Добавить ссылки в очередь на выкачку;
- Скачать все URL из очереди и повторить цикл.

Всё ли так просто?



- Обычные страницы тоже доставляют неприятности
 - Скорость доступа/Ширина канала до удалённых серверов различаются
 - Все сайты отличаются по своей структуре
 - Насколько глубоко робот должен обходить сайт?
 - Сайты-зеркала и дубликаты страниц
- Вежливость - не делать запросы слишком часто

Первоначальный анализ сайта



- Проходимся по нескольким страницам сайта прямо в браузере
- Пытаемся определить:
 - Есть ли редирект
 - Что возникает при “человеческом поведении”

Что любой робот должен сделать



- Быть вежливым: соблюдать явные или неявные соглашения
 - Обходить только разрешённые страницы
 - Соблюдать robots.txt
- Быть умным: устойчивость к ловушкам (spider traps) и некорректному и зловредному поведению веб-серверов

Что любой робот должен бы сделать (1)



- Поддерживать выполнение распределённых операций: иметь возможность запускаться на нескольких машинах;
- Быть масштабируемым: добавление серверов ведёт к увеличению производительности
- Производительность/эффективность: использовать все ресурсы железа и сети

Что любой робот должен бы сделать (2)



- Скачивать в первую очередь “качественные” страницы
- Постоянное обновление: скачивать свежие копии ранее скаченных страниц
- Расширяемость: поддерживать новые форматы и протоколы

Очередь URL'ов (URL frontier)



- Может содержать множество страниц с одного хоста;
- Не должна пытаться скачать их все в одно и то же время;
- Должна по возможности загружать все потоки на выкачку.



- Явные правила: веб-мастер сам определяет, какие части сайта надо обойти
 - robots.txt
- Неявные правила: даже без указаний от веб-мастеров избегать выполнения частых запросов

robots.txt (1)



- Протокол для работы робота, ограничивающий доступ к сайту, разработан в 1994 году;
- <http://www.robotstxt.org/>
- Сайт определяет, что не надо качать
 - robots.txt
 - <meta>

robots.txt (2)



User-agent: *

Allow: /article*

Disallow: /auth*

Clean-param: s /forum/showthread.php

Crawl-delay: 30

Sitemap: <https://solo.nabiraem.ru/sitemap.xml>

robots.txt (3)



Какие из этих документов мы можем скачать?

- <http://site.ru/>
- <http://site.ru/auth/>
- <http://site.ru/auth/article/>
- <http://site.ru/article/>
- <http://site.ru/post/>

User-agent: *

Allow: /article*

Disallow: /auth*

robots.txt (4)



Какие из этих документов мы можем скачать?

- <http://site.ru/>
- ~~<http://site.ru/auth/>~~
- ~~<http://site.ru/auth/article/>~~
- <http://site.ru/article/>
- <http://site.ru/post/>

User-agent: *

Allow: /article*

Disallow: /auth*

Примеры robots.txt



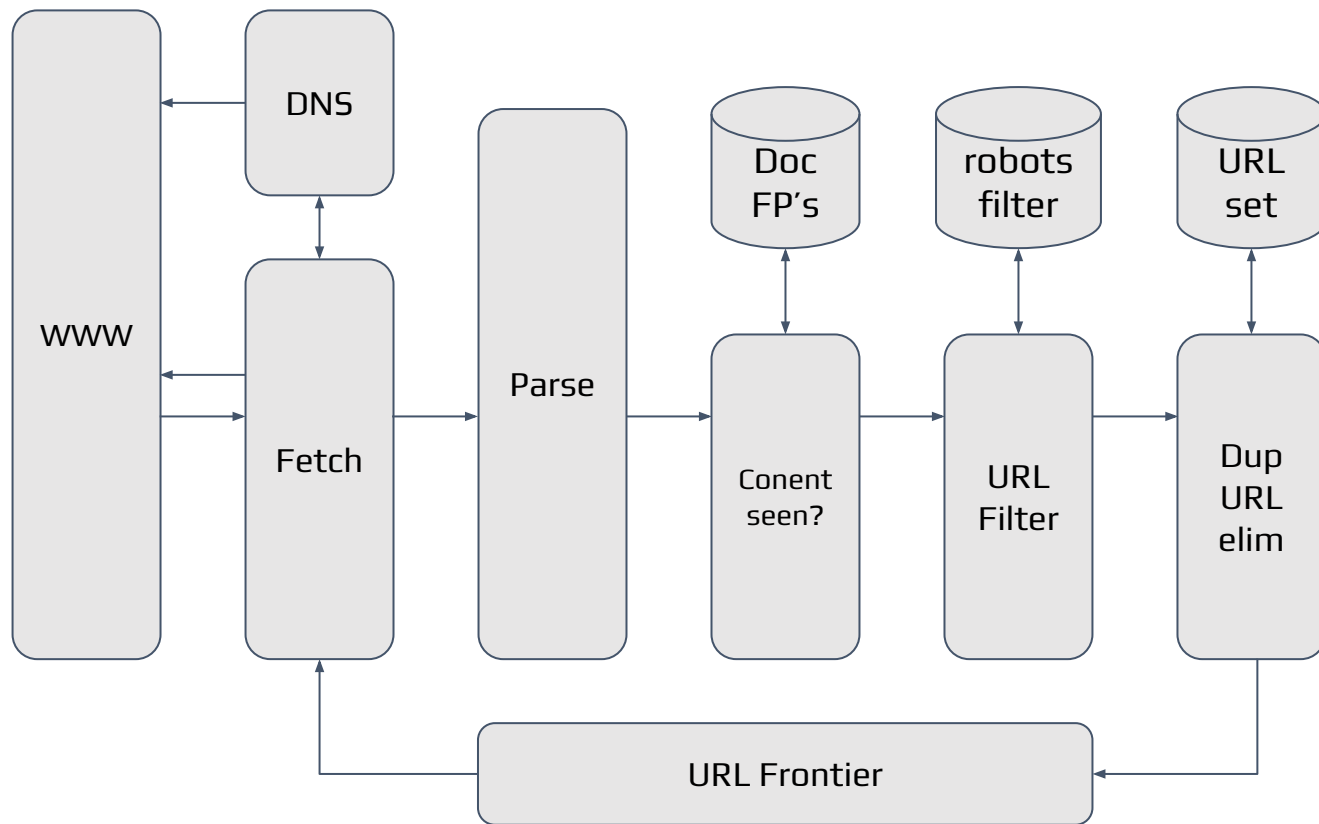
- <https://solo.nabiraem.ru/robots.txt> (любит всех)
- <https://music.yandex.ru/robots.txt> (очистка параметров)
- <https://lenta.ru/robots.txt> (эталон)
- <http://tamqui.com/robots.txt> (не любит майл)
- <https://directmobile.ru/robots.txt> (вообще никого не любит)
- <https://tnt-online.ru/robots.txt> (delay)

Выполнение шагов при выкачки



- Выбрать URL из очереди
- Скачать документ с этого URL
- Распарсить документ
 - Извлечь ссылки на другие документы
- Проверить, что такой документ уже есть в базе
 - Если нет, то добавить в индекс
- Для каждой извлечённой ссылки
- Убедиться, что она проходит различные фильтры
- Проверить, что её ещё нет в очереди (избегать дублирование ссылок)

Основная архитектура робота



DNS (Domain Name Service)



- Сервис получения данных о хосте в интернете
 - для данного URL получить IP адрес (resolving);
 - Сервер предоставляет распределённым набором серверов, время запроса (lookup latency) может быть большим (больше секунды!)

Схема работы DNS

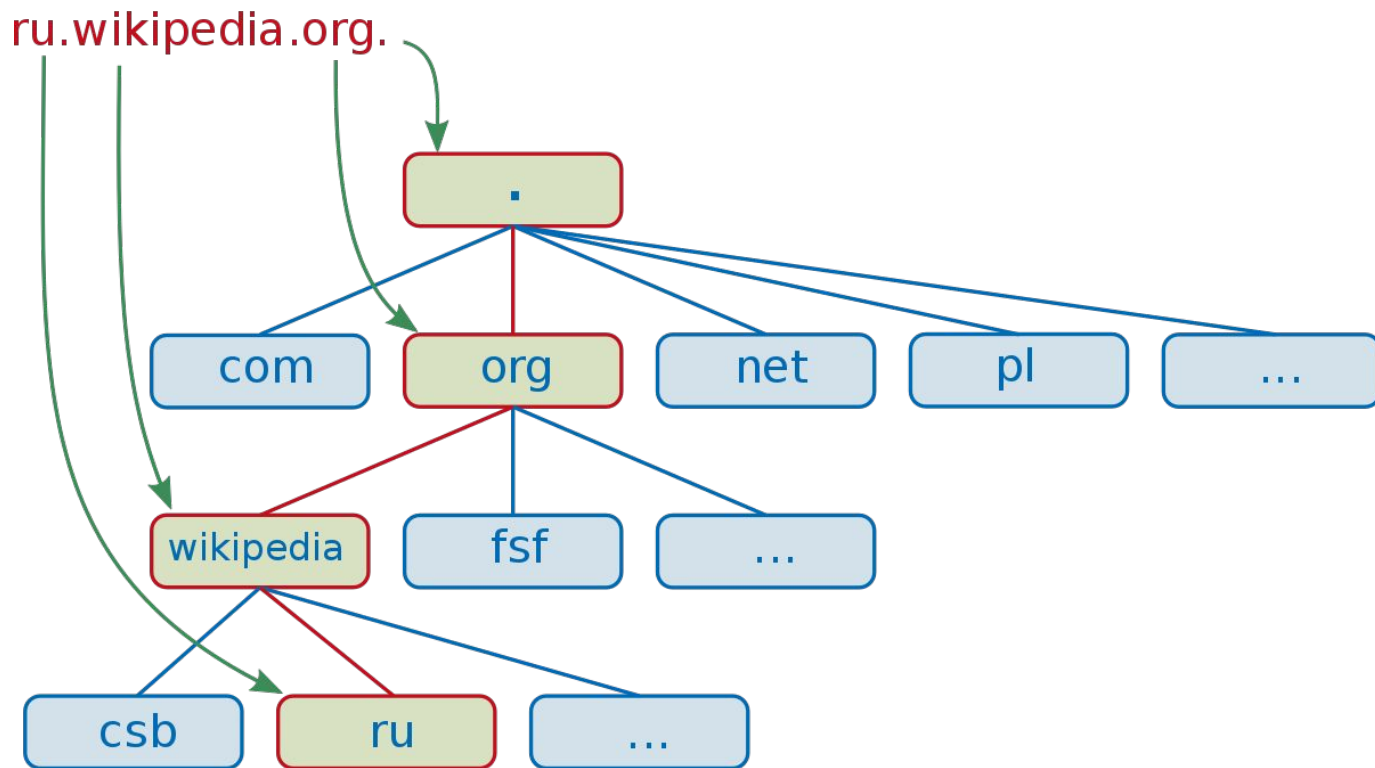


Схема работы DNS (2)



- Зачем несколько адресов?
 - Балансировка нагрузки
 - Failover
 - GeoDNS
 - IPv6 подходит лучше

Схема работы DNS (3)



- PTR-записи
 - позволяют сделать резолвинг наоборот
 - используются фаерволами, netstat и ... админами!

```
$ host 217.69.135.248
```

```
248.135.69.217.in-addr.arpa domain name pointer
```

```
reco-vklive-recommend3.g.smailru.net.
```


Парсинг: нормализация URL



- При парсинге документа часть ссылок являются относительными URL;
- Во время парсинга URL'ы должны нормализоваться

Определение кодировки



- Правильней всего следовать популярным браузерам
 - Не пытаемся быть умнее браузера
1. HTML: Type-Content?
 2. HTML: Meta?
 3. UTF-8?



- Sitemaps содержат списки URL'ов и информацию про эти URL'ы
 - Например, время модификацию и частоту обновления
- Генерируются на стороне сервера
- Дают роботам подсказку, как часто обкачивать ту или иную страницу

Sitemap. Формат



URL: https://www.bfm.ru/sitemap_main.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<url>
```

```
  <loc>https://www.bfm.ru/</loc>
```

```
  <lastmod>2021-09-23T14:34:36+03:00</lastmod>
```

```
  <changefreq>hourly</changefreq>
```

```
  <priority>1.0</priority>
```

```
</url>
```

...

Sitemap: примеры



- <https://music.yandex.ru/sitemap-new.xml>
- <http://toshcorp.ru/sitemap.xml>
- <https://www.bfm.ru/sitemap.xml>

Что кроме sitemap?



- <http://apple.com/ru/sitemap>
- https://www.bfm.ru/news.rss?container_breaking=8
- SEO



- Требования для системы хранения документов
 - Random access
 - Доступ к документу на основе его URL'а,
 - Обычно используется hash от URL.
 - Большие файлы
 - Не надо хранить и открывать кучу файлов,
 - Уменьшает seek time.
 - Сжатие
 - Снижает требования по месту на дисках и увеличивает эффективность доступа
 - Текст сильно избыточен
 - Соседние документы часто похожи
 - Обновление
 - Иметь возможность добавлять и обновлять новый контент



АEROSPIKE



Используется для сканирования и bulk-load:

- 4 PiB, 150MM ключей,
- daily: 1MM, 15 TiB

АEROSPIKE

Используется как KV для вспомогательных нужд.

Что же делать?



- Райан Митчелл “Современный скрапинг веб-сайтов с помощью Python”;
- Про Scrapy на Python;
- Так же есть [Selenium](#), [Phantomjs](#), [puppeteer](#) позволяющие выполнять js;
-

Домашнее задание №2



1. Написать обкатку документов и сохранить их в базе данных
2. При сохранении документов нужно только необходимый текст (удалять навигационную обвязку и т.д.)
3. Подготовить отчёт

Рекомендуемая литература

Введение в информационный поиск
I Маннинг Кристофер Д., Шютце
Хайнрих

Для саморазвития (опционально)
Чтобы не набирать двумя
пальчиками



Спасибо за
внимание!

Антон Кухтичев



a.kukhtichev@mail.ru



[@toshunster](https://t.me/toshunster)