

Задачи по курсу «Протоколы Интернета»

Для допуска экзамену в течение семестра надо сдать 3 задачи из каждой темы.

Тема 1 – низкий уровень, задачи 1 – 3.

Тема 2 – система доменных имен, задача 4.

Тема 3 – прикладные протоколы и API, задачи 5 – 8.

Можно сдавать больше задач, набирая баллы за работу в семестр.

Общие замечания

0. Язык реализации любой.
1. Документация, справка по использованию.
2. Тестирование (примеры запуска).
3. Не использовать готовые модули, решающие существенную часть задачи.
4. Элегантная обработка стандартных ошибок (нет доступа в Интернет, сервер долго не отвечает, не удается разрешить доменное имя в IP адрес...)

Задачи

1. Стоимость = 8 баллов. Трассировка автономных систем. Пользователь вводит доменное имя или IP адрес. Осуществляется трассировка до указанного узла (например, с использованием `tracert`), т. е. мы узнаем IP адреса маршрутизаторов, через которые проходит пакет. Необходимо определить к какой автономной системе относится каждый из полученных IP адресов маршрутизаторов. Для определения номеров автономных систем можно обращаться к базам данных региональных интернет регистраторов.

Выход: для каждого IP-адреса – вывести результат трассировки (или кусок результата до появления ***), для "белых" IP-адресов из него указать номер автономной системы.

В итоге должна получиться такая таблица

№ по порядку	IP	AS

Стоимость = 10 баллов. То же, что на 8 баллов + определять страну и провайдера.

2. Стоимость = 8 баллов. SNTP сервер точного времени, который «врет» на заданное в своем конфигурационном файле число секунд. Сервер прослушивает 123 порт UDP. Время сервер узнает либо у ОС, либо у другого надежного SNTP сервера точного времени, например, `time.windows.com`. В конфигурационном файле сервера указано, на сколько секунд он должен «врать», т. е. из точного времени сервер вычитает или прибавляет указанное число секунд.

Проверять работоспособность надо стандартным SNTP клиентом (можно взять любой по желанию студента).

3. Сканер TCP и UDP портов.

Стоимость = 5 баллов. Определить, какие TCP порты открыты в заданном диапазоне.

Стоимость = 10 баллов. То же, что на 5 баллов + параллельная реализация + проверять UDP порты. Если послать UDP-пакет на закрытый порт, система ответит сообщением ICMP «порт недоступен». Отсутствие такого сообщения можно истолковать как сигнал того, что порт открыт. С UDP портами может возникнуть проблема – если порт блокируется брандмауэром, метод неверно покажет, что порт открыт. Если заблокированы ICMP-сообщения о недоступности порта, все порты будут казаться открытыми. Также, может быть установлено ограничение на частоту использования ICMP-пакетов, что также влияет на результаты, даваемые методом. Такие проблемы (в реальной среде) не являются препятствием для сдачи задачи, но в специально подготовленной среде (например, сервер поднят на localhost) сканер UDP портов должен правильно работать.

Стоимость = 15 баллов. То же, что на 10 баллов + определять протокол, который работает на открытом порте. Имеется в виду не «приписывать» номеру хорошо известных портов имена соответствующих протоколов, а реально определять протокол, посылая запрос (возможно некорректный) и анализируя ответ. Если http сервер будет запущен на 110 TCP порту, это надо уметь понимать. Достаточно распознавать http, smtp, pop3, dns, snmp.

4. Стоимость = 20 баллов. Кэширующий DNS сервер. Сервер прослушивает 53 порт. При первом запуске кэш пустой. Сервер получает от клиента рекурсивный запрос и выполняет разрешение запроса. Получив ответ, сервер разбирает пакет ответа, извлекает из него ВСЮ (в том числе из полей Authority и Additional) полезную информацию, т. е. все ресурсные записи, а не только то, о чем спрашивал клиент. Полученная информация сохраняется в кэше сервера. Например, это может быть два хэш-массива.

Доменное имя	IP адрес

IP адрес	Доменное имя

Сервер регулярно просматривает кэш и удаляет просроченные записи (использует поле TTL). Сервер не должен терять работоспособность (уходить в бесконечное ожидание, падать с ошибкой и т. д.), если старший сервер почему-то не ответил на запрос. Во время штатного выключения сервер сериализует данные из кэша, сохраняет их на диск. При повторных запусках сервер считывает данные с диска и удаляет просроченные записи, инициализирует таким образом свой кэш.

Достаточно уметь обрабатывать записи типа A, AAAA, NS, PTR.

5. Стоимость = 10 баллов. SMTP клиент. В отдельной папке лежит конфигурационный файл, текстовый файл с письмом и файлы-аттачменты (картинки, документы и т. д.). В текстовом файле пользователь пишет письмо (plain text) на английском или русском языке. В конфигурационном файле пользователь задает адрес получателя(лей), тему (возможно на русской языке) и имена файлов-аттачментов для отсылки в виде вложения.

На сервера mail, yandex, google письма с вложениями должны доходить и быть читабельными.

6. Стоимость = 10 баллов. POP3 клиент. Пользователю было послано письмо (на русском или английском языках) на почтовый сервер mail, yandex или google. В письмо было сделано вложение, например, пара картинок.

Пользователь запускает программу для скачивания письма с почтового сервера. Может запросить заголовки: тему, дату, отправителя и т. п. Может запросить TOP (несколько строк) сообщения. Может скачать письмо с вложением.

7. Стоимость = 10 баллов. HTTP проху. Прокси сервер слушает порт, например, 8080. Браузер пользователя настроен работать через прокси сервер (в настройках прописать). Пользователь обращается к некоторому веб сайту, запрос отправляется прокси серверу. Прокси сервер, проанализировав запрос пользователя, передает запрос нужному веб серверу, а обработанный ответ перенаправляет пользователю. Обработка заключается, например, в том, чтобы вырезать с сайта рекламные баннеры. Иные варианты обработки можно согласовать с преподавателем.

Поскольку произвольного сайта затруднительно понять, где именно находится реклама, прокси может вырезать её, например, только на с сайта e1.ru или только с сайта vk.com.

8. Стоимость = 10 баллов. Использование API. Требуется, используя, API В Контакте (или Facebook, yandex) получить информацию и вывести ее в удобочитаемом виде. Список конкретных задач будет уточнен позже. Пример задачи: вывести список друзей указанного пользователя, вывести названия фотоальбомов указанного пользователя и т. п.