

2020

Задачи по курсу “Информационные системы и сети”.

Оценка за курс выставляется на экзамене по теории.

Сдача заданий на семинарах формирует “промежуточную” оценку за семинарские задания. Для получения оценки “отлично” необходимо набрать не менее 50 баллов, на оценку “хорошо” - не менее 40 баллов, на оценку “удовлетворительно” не менее 30 баллов.

Сдача задания подразумевает также и демонстрацию работоспособности решения. Необходимо умение отвечать на вопросы как по реализации решения, так и по теории. Перед сдачей решения нужно согласовать с семинаристом выбор используемых технических средств, в том числе выбор языка программирования и программных библиотек.

У каждой задачи в названии указан диапазон максимума получаемых за нее баллов. Бонусы, указанные в тексте задачи как “(+X баллов)”, применяются к нижней границе указанного диапазона.

Поощряется выполнение т.н. “творческих заданий”, условие которых студент формулирует самостоятельно. Максимальная стоимость такого задания определяется преподавателем. Допускается сдача заданий, выполненных студентом до начала прохождения курса.

За одну из задач, отмеченных звездой, можно получить вдвое больше баллов. Для этого необходимо реализовать ее на языке C с использованием POSIX socket API и корректно обработать все исключительные ситуации. Перед сдачей нужно согласовать с семинаристом свое намерение и архитектуру приложения.

Задачи разбиты на разделы. Сумма баллов за каждый раздел должна быть равна минимум трем.

После 15 октября запрещается сдавать двухбалльные задачи, после 15 ноября – трехбалльные, после 15 декабря – четырехбалльные.

[Раздел 1: "Возня с сокетами"]

1. Simple proxy (5 баллов, *)

Написать программу, которая на каждое входящее TCP-соединение на указанный порт порождает исходящее и коммутирует их (данные, поступающие по одному соединению, тут же отправляет по другому и наоборот). Отдельно следует отметить, что программа должна поддерживать несколько одновременных входящих соединений.

2. Wake-on-LAN (2–4 балла, *)

Написать программу, генерирующую "волшебный пакет" Wake-on-LAN, позволяющий вывести компьютер из состояния пониженного энергопотребления. Решение, использующее отправку UDP datagram оценивается в 2 балла; формирующее "сырой" Ethernet-пакет - в 4 балла.

3. Простой HTTP-сервер (4–7 баллов, *)

Написать простой HTTP-сервер для отдачи статического содержимого. Сервер должен по запросу отдавать содержимое затребованного файла из определенной директории или ее поддиректорий. (4 балла) Стоимость задачи можно увеличить, реализовав:

- 1) режим для запуска через inetd (+1 балл);
- 2) возможность отдачи клиенту не содержимого файла, но результата его исполнения. т. е. реализовать частичную поддержку CGI (в первую очередь реализовать передачу переменных окружения REMOTE_ADDR, REQUEST_METHOD, QUERY_STRING, PATH_INFO) (+1 – +2 балла).

4. Super-server (inetd) (10 баллов, *)

Написать т. н. super-server: программу, ожидающую соединения на нескольких TCP- или UDP-портах и, при открытии соединения, порождающую соответствующую этому порту дочернюю программу-обработчик, стандартные потоки ввода и вывода которой скомутированы с сокетом входящего соединения. Другими словами, super-server позволяет "поговорить" по сети со стандартными вводом и выводом "обычной" внешней программы. Соответствие программ портам должно задаваться посредством конфигурационного файла произвольного формата. Обязательно реализовать: выбор типа сокета и номера порта, передачу произвольной команде произвольного набора аргументов, опцию, запрещающую или разрешающую запуск нескольких программ-обработчиков одного типа одновременно.

5. Децентрализованный чат с использованием broadcast (4–8 баллов, *)

Написать программу, позволяющую нескольким пользователям обмениваться текстовыми сообщениями с использованием широковещательных пакетов в одноранговой сети без использования выделенного "сервера" (4 балла). Дополнительно можно реализовать поддержку работы в сети сложной топологии с использованием ретрансляции пакетов на узлах с несколькими интерфейсами (+4 балла).

6. Централизованный чат с использованием TCP (6–8 баллов, *)

Написать программу, позволяющую нескольким пользователям обмениваться текстовыми сообщениями через выделенный сервер. Необходимо придумать свой протокол, реализовать авторизацию, отправку сообщений и передачу информации о статусе клиента (как минимум подключен/отключен) (6 баллов). Дополнительно можно реализовать шифрование соединения посредством TLS (+2 балла).

7. Тестер характеристик сетевого подключения (3–10 баллов)

Написать пару программ, производящих двустороннее измерение пропускной способности и задержки сетевого соединения. Полученные результаты объяснить (3 балла). Дополнительные баллы можно получить, реализовав тестирование как поверх TCP, так и поверх UDP (+2 балла); измерив и отобразив скорости передачи данных с субмиллисекундным разрешением (+5 баллов).

[Раздел 2: "Применение готовых библиотек"]

8. Уведомления по XMPP (3–4 балла)

Реализовать программу, периодически уведомляющую о чем-либо посредством передачи XMPP-сообщений. В качестве источника данных использовать сервис наподобие weather.nsu.ru, поощряется выбор других сервисов сети Интернет. Подразумевается использование готовых библиотек для работы как с XMPP, так и с протоколом выбранного сервиса. Дополнительный балл можно получить, если программа будет поддерживать постоянное соединение XMPP, а не устанавливать новое для каждого нового уведомления (+1 балл).

9. Туннелирование TCP поверх XMPP (6–12 баллов)

Написать программу или несколько программ, выступающих клиентами протокола XMPP и реализующих туннелирование TCP-соединения между двумя своими экземплярами. Выбор конкретного способа передачи payload в рамках протокола XMPP производится и обосновывается студентом. Подразумевается использование готовой библиотеки для работы с XMPP. Программа должна корректно передавать любую последовательность данных, а не только символы ASCII. Должны поддерживаться стандартные для XMPP возможности сжатия и шифрования соединения (+2 балла). Дополнительно можно заложить в протокол и реализовать открытие, закрытие и мультиплексирование нескольких TCP-соединений (+4 балла).

10. ZeroMQ echo-сервер с балансировкой нагрузки (2–3 балла)

Реализовать с помощью ZeroMQ сервер, отправляющий клиенту назад его же запрос (echo-сервер). Продемонстрировать балансировку нагрузки на нескольких запущенных одновременно серверов (2 балла). Дополнительный балл можно получить, если клиент и сервер будут написаны на разных языках (+1 балл).

11. FTP-сервер "с поддержкой CGI" (8–10 баллов)

Реализовать FTP сервер, отдающий клиенту вместо содержимого исполняемых файлов результат их исполнения. Дополнительные два балла можно получить, не используя готовых библиотек для работы с протоколом FTP (+2 балла).

12. IMAP – FTP гейт (10 баллов или 6 баллов)

Написать FTP-сервер, выступающий в то же время IMAP-клиентом и предоставляющий доступ к иерархии почтовых ящиков IMAP и их содержимому посредством протокола FTP. Предполагается использование готовой библиотеки как минимум для работы с IMAP (10 баллов). Допускается сдача решения, состоящего из комбинации нескольких готовых программ (6 баллов).

13. Индикатор сетевой активности (7 баллов)

Написать программу, перехватывающую сетевой трафик и графически отображающую временную зависимость сетевой активности (количества захваченных данных в единицу времени). Предполагается использование библиотеки pcap. Также необходимо реализовать возможность задания произвольного фильтра на учитываемый сетевой трафик (см. pcap-filter). Возможна сдача решений, представляющих собой комбинации готовых утилит; такие решения принимаются на усмотрение преподавателя, оцениваются в первую очередь адекватность подхода и выбранных средств, корректность, краткость и элегантность.

[Раздел 3: "Администрирование"]

14. Аналог программы simple proxy, прямо в конфигурационном файле inetd (2 балла)

Реализовать аналог программы simple proxy (см. задачу 1) в одну строку конфигурационного файла классической реализации программы inetd.

15. Настроить SSH-сервер и SSH port forwarding, запускаемые через inetd (4 балла)

Настроить SSH-сервер, запускаемый из демона inetd только по факту установки подключения со стороны клиента. Настроить перенаправление порта через SSH в конфигурационном файле демона inetd (TCP-подключение inetd хоста А следует через SSH-подключение к хосту В перенаправить на порт хоста С, доступного с хоста В).

16. Настроить DNS-сервер (4 балла)

Требуется настроить первичный DNS, отвечающий за какую-нибудь зону, в которой будет два почтовых сервера, запись А, CNAME и WKS.

17. Настроить iptables и SSH port forwarding (2 балла)

Настроить iptables так, чтобы были запрещены все входящие соединения с внешних интерфейсов, кроме соединений на 22 порт TCP. Запустить и настроить, если это необходимо, SSH-сервер. Продемонстрировать удаленный доступ к локальному ресурсу посредством SSH port forwarding.

18. LAMP (2 балла)

Настроить LAMP (Linux, Apache, MySQL/MariaDB, PHP/Perl/Python) и какой-нибудь стандартный сервис, его использующий. Решения, использующие аналогичные компоненты, не присутствующие в этом списке, принимаются по предварительному согласованию с преподавателем. Не следует ожидать полного балла за установку готового пакета для WAMP, или специализированного дистрибутива Linux.

19. Загрузка по сети (5 баллов)

Настроить серверы DHCP и TFTP, продемонстрировать бездисковую загрузку ОС по сети посредством PXE.

20. Автоматическое обнаружение прокси-сервера (10–15 баллов)

Настроить прокси-сервер и "убедить" какой-нибудь браузер использовать его, пользуясь исключительно механизмами автоматического обнаружения прокси-сервера. Дополнительно до пяти баллов можно получить, реализовав на уровне прокси-сервера инспекцию и/или модификацию запросов и ответов (пример: два балла, если вместо запрашиваемой страницы с вероятностью $\frac{1}{5}$ будет открываться поиск mail.ru; пять, если продемонстрируете перехват пароля для какого-либо сервиса или вертикальное отражение изображений с запрошенной страницы (+5 баллов).

[Раздел 4: "Организация сетей"]

21. Настройка Wi-Fi маршрутизатора с NAT и пробросом портов (3 балла)

Настроить обычный Wi-Fi маршрутизатор "потребительского класса". Необходимо продемонстрировать подмену MAC-адреса внешнего интерфейса, настройку NAT и проброс портов "внутрь" сети. Маршрутизатор может быть предоставлен преподавателем по предварительному запросу.

22. VLAN (3 балла)

Организовать в изначально одноранговой сети несколько изолированных подсетей при помощи подинтерфейсов VLAN.

23. VPN-туннель и маршрутизация (5 баллов)

Настроить VPN-туннель и настроить маршрутизацию между двумя изолированными подсетями. Стоит учесть, что для решения этой задачи отдельно от предыдущей могут потребоваться дополнительное оборудование, а возможно и дополнительные усилия, согласия и договоренности.

[Раздел 5: "Перехват и обход"]

24. Выяснить ключ WEP-сети (4 балла)

Организовать беспроводную сеть, использующую шифрование WEP и восстановить ключ шифрования по захваченному трафику.

25. Перехват и анализ данных при помощи программы Wireshark (2 балла)

Перехватить трафик, генерируемый браузером при входе на сайт, не использующий шифрование, и вычленив из него логин и пароль пользователя. В самом простом варианте подразумевается применение анализатора сетевого трафика Wireshark "вручную".

26. Подслушивание изображений (3–8 баллов)

Написать программу, перехватывающую сетевой трафик, вычленившую из него изображения в одном или нескольких популярных форматах графических файлов и отображающую их на экране по мере обработки. Дополнительные баллы ставятся в зависимости от полноты и обстоятельности решения: приветствуются поддержка нескольких форматов, рекомбинация пакетов и т.п. (+5 баллов)

27. Выход в Интернет через гостевую сеть ИЯФ без регистрации (10 баллов)

Продemonстрировать возможность доступа к произвольным публично доступным сервисам сети Интернет при использовании в качестве единственного подключения гостевой беспроводной сети ИЯФ и без использования регистрационных данных для "входа" в нее.

[Раздел 6: "Крупные задачи"]

28. Обратная разработка посредством анализа трафика (7 баллов, *)

Разработать аналог предоставленной преподавателем программы с закрытым исходным кодом посредством анализа сетевого трафика.

29. Port knocking daemon, открывающий порты firewall'a (5–8 баллов)

Написать программу, слушающую определенные порты и, в случае получения правильной последовательности подключений или пакетов с одного адреса на разные порты в течение фиксированного промежутка времени, открывающую на некоторое время доступ к "секретному порту" с IP-адреса отправителя. Дополнительный балл можно получить, если на этих же портах при этом принимают и обрабатывают соединения полностью функционирующие и не вызывающие подозрений сервисы (+3 балла).

30. Написать свой fail2ban, закрывающий порты firewall'a (4 балла)

Написать программу, отслеживающую неудачные попытки авторизации посредством чтения log-файла (например, SSH- или HTTP-сервер и, в случае определенного количества неудачных авторизаций с одного и того же IP-адреса, временно закрывающую доступ с этого адреса к порту "атакуемого" сервиса.

31. Распределённое хранилище файлов (10-20 баллов)

Написать программу, реализующую избыточное и распределенное хранилище файлов на нескольких узлах одной сети (10 баллов). Дополнительные баллы можно получить, к примеру, за что-либо из перечисленного далее: отсутствие выделенного сервера; степень избыточности, не равную числу узлов; реализацию избыточности с использованием XOR вместо дублирования; возможность примонтировать хранилище как файловую систему посредством FUSE или аналога; предоставление доступа к данным посредством какого-либо общепринятого протокола передачи файлов (+10 баллов).

32. Голосовое общение (5 баллов)

Написать программу для (одно-)двухсторонней передачи голоса между двумя или более пользователями без выделенного сервера в одноранговой сети. Для компрессии речи необходимо использовать существующий кодек, например Speex, Opus или Vorbis. Использование протоколов SIP и RTP принесет автору дополнительные баллы (сверх 5 баллов).

33. Протокольный демультимплексор (7 баллов, *)

Написать программу, слушающую один TCP-порт и перенаправляющую соединения на другие порты в зависимости от протокола, используемого подключающейся программой-клиентом.

Написать какой-нибудь существенный патч в upstream open-source сетевой утилиты (по дополнительной договоренности)

Творческие задания (по дополнительной договоренности)