Вопросы к очному туру олимпиады «Инфотелеком» (НИИ «Масштаб»)

1. Гламурная блондинка Ефросинья купила смартфон и установила в него две SIM-карты разных операторов связи в надежде, что теперь сможет смотреть любимые онлайн-сериалы с суммарной скоростью двух Интернет-подключений (по 1 Мбит/с каждое). Однако при попытке вручную выполнить пакетную балансировку IP-пакетов между двумя подключениями она выяснила, что TCP-приложения получили не сумму пропускных способностей (2 Мбит/с), а всего лишь 500 кбит/с, т.е. меньше, чем если бы она использовала всего один Интернет-канал без всякой балансировки. Чем это можно объяснить?

**Ответ**: Используемые Ефросиньей Интернет-подключения имели существенно асимметричные значения RTT, что не позволило протоколу ТСР корректно выбрать размер окна при пакетной балансировке.

**Пояснения.** Проблема могла возникнуть, только если каналы связи Ефросиньи имели асимметричные характеристики (например, в одном канале RTT-задержка равна 500 мс, а в другом 10 мс или в одном потери 1%, а в другом 10% и т.п.). Причина проблемы – в особенностях реализации механизма окна в ТСР: окно позволяет подстроить скорость передачи только под стабильный канал связи, в котором характеристики почти не меняются (т.е. RTT почти постоянен и мало колеблется). А при пакетной балансировке половина пакетов идут по каналу с большой RTT, а другая половина – по каналу с маленькой RTT, что не позволяет TCP правильно настроить окно. Первый способ частично решить проблему Ефросиньи – это использовать балансировку по потокам, а не по пакетам (тогда каждый ТСР-поток будет все свои пакеты отправлять по одинаковому пути). Второй способ решить проблему – использовать вместо ТСР протоколы SCTP, MP-TCP, в которых каждый используемый канал связи настраивается независимо от другого, что позволяет успешно агрегировать пропускную способность нескольких каналов связи. Если олимпиадник приведёт хотя бы один из этих вариантов решения проблемы, стоит зачесть ответ.

**Баллы: 30.**

1. В параллельной вселенной программисты додумались, как и мы, хранить и передавать числа с помощью бинарных кодов, состоящих из нолей и единиц. Однако при этом наши антиподы используют гипотетическую фибоначчиевую систему счисления вместо привычной нам двоичной. В результате, у них код «1100» означает не двенадцать, а восемь. Каким образом можно передать в параллельную вселенную сообщение, содержащее ответ на главный вопрос жизни (т.е. передать число 42)?

**Ответ**: 10010000

**Пояснения.** Правильным ответом являются коды 10010000, 10001100, 10001011, 1110000, 1101100, 1101011, каждый из которых является корректной записью числа 42. Участники олимпиады должны по подсказке в задании (1100b=8Ф) догадаться, что в фибоначчиевой системе счисления вместо двоичных весов разрядов «1, 2, 4, 8, 16…» используются числа Фиббоначи «1, 2, 3, 5, 8, …». Если участник олимпиады привёл только один из указанных кодов, то поставить ему 50% баллов. Если привёл от двух до четырёх, поставить 90%. Если привёл от пяти до шести, поставить 100%.

**Баллы: 20.**

1. Программист очнулся в вагоне поезда, состоящего из большого количества одинаковых вагонов, между которыми можно свободно переходить. Кондуктор сообщил программисту, что поезд находится на круговой железной дороге и первый вагон соединяется с последним. В каждом вагоне есть стоп-кран, который может находиться в одном из двух положений, но начальное состояние стоп-кранов неизвестно. После десятой кружки кофе в вагоне-ресторане программист придумал алгоритм, позволяющий посчитать количество вагонов в поезде. Опишите этот алгоритм (словесно или на любом языке программирования), зная, что программисту разрешено использовать стоп-краны?

**Ответ**: Нужно попеременно идти то влево, то вправо от исходного вагона с каждым разом увеличивая количество проходимых вагонов, при этом поднимая стоп-краны в одном направлении и опуская в противоположном. При обнаружении неверного положения стоп-крана в посещаемом вагоне нужно остановиться и посчитать количество вагонов.

**Пояснения.** Идя налево от исходного вагона, программист опускает стоп-краны в посещаемых вагонах; идя направо – поднимает. При этом он поочерёдно двигается то налево, то направо, увеличивая на один количество проходимых вагонов, считая от исходного в каждую сторону. Для этого он держит в уме пару чисел i и j, которые изменяются следующим образом: +1, -1, +2, -2, +3, -3, …, где знак означает направление движения (налево или направо), а числа означают количество проходимых вагонов, считая от исходного. Программист точно знает количество пройденных вагонов и знает, какое он в них установил положение стоп-кранов. По мере увеличения i и j в какой-то момент самый левый посещённый вагон совпадёт с самым правым (круг замкнётся), при этом программист обнаружит неверное положение стоп-крана в уже пройденном им вагоне. Для подсчёта количества вагонов достаточно будет в этот момент сложить модули переменных i и j.

**Баллы: 20.**

1. Встретились два программиста на дискотеке:

– Сколько у тебя компьютеров?

– Три. К сожалению, все однопроцессорные.

– А сколько ядер на процессорах?

– Если количества ядер перемножить, получится 36.

– Мало информации, не могу посчитать.

– Суммарное количество ядер равно возрасту ди-джея.

– Всё еще мало информации, хотя ди-джей мой родной брат.

– Самый слабый процессор не поддерживает SSE4.

– О, теперь знаю!

После этого был правильно назван ответ (количество ядер в процессорах). Каков этот ответ?

**Ответ**: 1, 6, 6.

**Пояснения.** 1, 6, 6 (или любая перестановка этих чисел). Из условия понятно, что решение надо искать в целых числах. Зная делители числа 36 (1, 2, 3, 4, 6, 9, 12, 18), можно методом перебора найти уникальные разложения на множители числа 36, см. в таблице:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Проц1 | Проц2 | Проц3 | Сумма | Произведение |
| 1 | 1 | 36 | 38 | 36 |
| 1 | 2 | 18 | 21 | 36 |
| 1 | 3 | 12 | 16 | 36 |
| 1 | 4 | 9 | 14 | 36 |
| 1 | 6 | 6 | 13 | 36 |
| 2 | 2 | 9 | 13 | 36 |
| 2 | 3 | 6 | 11 | 36 |
| 3 | 3 | 4 | 10 | 36 |

Уточнение про ди-джея будет недостаточным для ответа, только если сумма ядер равна повторяющемуся в таблице числу 13 (иначе сразу после этого уточнения был бы найден ответ по таблице, т.к. возраст брата точно известен). Третье уточнение позволяет отбросить вариант 2, 2, 9, т.к. в нём присутствуют сразу два одинаково слабых процессора. Остаётся ответ. 1, 6, 6.

**Баллы: 10.**

1. В заголовке пакета IP (версия 4) для указания размера пакета выделено 16 бит, что позволяет передавать пакеты размером вплоть до 65535 байт. Однако, наблюдая за размерами реально курсирующих IP-пакетов в Интернете, можно убедиться, что их размеры практически никогда не превышают 2000 байт. Чем это можно объяснить?

**Ответ**: Наиболее часто IP-пакеты передаются по Ethernet-сетям, в которых предельно допустимый размер фрагмента передаваемых данных равен примерно 1500 байт.

**Пояснения.** Наиболее часто IPv4-пакеты в современных сетях передаются по Ethernet-сетям, в которых предельно допустимый размер фрагмента данных (кадр) равен примерно 1500 байт (значение MTU). Поэтому при попытке передать IP-пакет более 1500 байт он будет фрагментирован на несколько кадров по 1500 байт, каждый из которых будет передан независимо. Затем на другом конце эти фрагменты придётся собирать воедино. Все эти операции по фрагментированию и сборке являются нежелательными (замедляется RTT), поэтому стараются их избегать, не давая размеру пакетов выходить за 1500.

**Баллы: 10.**

1. В односвязном списке из n элементов записаны целые числа от 1 до n в произвольном порядке (каждое число встречается ровно один раз). Один элемент списка случайным образом удаляется. Каким образом можно за один проход полученного списка обнаружить удалённый элемент?

**Ответ**: Нужно из суммы арифметической прогрессии от 1 до n (с шагом 1) вычесть сумму элементов списка, рассчитанную при однократном проходе списка.

**Баллы: 10.**