

## Практика №1

### Работа с IP адресами

Итак IP-адрес - это адрес, используемый узлом на сетевом уровне. Он имеет иерархическую структуру. Что это значит? Это значит, что каждая цифра в его написании несет определенный смысл. Возьмем для примера номер обычного телефона - +74951234567. Первые два символа +7, «+» используется в начале записи номера телефона, «7» говорит о том, что номер принадлежит зоне РФ (в целом для кодирования страны может использоваться от одного до пяти символов). Далее следует 495 - это код Москвы. И последние 7 цифр взяты случайным образом, уникальный номер абонента, но так как не все АТС поддерживают семизначный номер, то в нем может храниться и уточнение региона, района и т.д., минимальное количество символов, отводимых на телефонный номер – пять. Кроме указанного примера могут существовать добавочные номера (например абонентом номера является организация со своим мини АТС, тогда для звонка конечному абоненту требуется указать добавочный номер).

Как видите здесь наблюдается четкая иерархия. То есть по номеру можно понять какой стране, области, району, организации, абоненту он принадлежит. IP адреса придерживаются аналогично строгой иерархии. Контролирует их организация IANA(англ. Internet Assigned Numbers Authority). Если на русском, то это «Администрация адресного пространства Интернет». Запомните, что слово «Интернет» пишется с большой буквы. Мало кто придает этому значение, но существует разница в написании с большой и маленькой буквы. В англоязычной литературе термин «internet» используется для описания нескольких подключённых друг к другу сетей. А термин «Internet» для описания глобальной сети, т. е. неправильное написание данного слова может привести к непониманию вашей документации.

Давайте перейдём к IP адресам. Не для кого, я думаю, не секрет, что IP адрес записывается в десятичном виде, это связано с тем, что мы привыкли воспринимать числовую информацию в десятичном формате. Однако все современные компьютеры воспринимают информацию в двоичном (0 и 1). Представление данных и команд в двоичном коде лежит в основе разработки вычислительных устройств и систем. Поэтому был придуман алгоритм перевода из двоичной системы в десятичную, и обратно.

Итак, как выглядят IP адреса в десятичном формате? Полезным уточнением будет то, что в работе мы рассматриваем IP адреса четвертого поколения.

«193.233.44.12» - типичный пример IP – адреса в десятичной записи. Как вы видите он состоит из 4-х чисел, называемых октетами и разделенных между собой точками.

Каждое такое число (октет) может принимать значение от 0 до 255. То есть одно из 256 значений. Длина каждого октета равна 8 битам, а суммарная

длина IPv4 = 32 бита. Теперь давайте разберемся, как данный адрес воспринимает вычислительная техника и работает с ним.

Как говорилось ранее, информация об IP адресе воспринимается вычислительной техникой исключительно в бинарном виде. Поэтому предложенный адрес требуется перевести в двоичный код. Переводить вручную должен уметь каждый. Если на экзамене по данной дисциплине вам попадется данный тип задач, то у вас не будет под рукой ничего, кроме бумаги и ручки, и полагаться придется только на свои навыки. Поэтому показываю, как это делать вручную. Строится таблица.

128	64	32	16	8	4	2	1
x	x	x	x	x	x	x	x

Вместо «х» записывается либо 1, либо 0. Таблица разделена на 8 колонок, каждая из которых несет в себе 1 бит (8 колонок = 8 бит = 1 октет). Расположены они по старшинству слева направо. То есть первый (левый) бит - самый старший и имеет номер 128, а последний (правый) - самый младший и имеет номер 1. Теперь объясню, откуда эти числа взялись. Так как система двоичная, и длина октета равна 8-ми битам, то каждое число получается возведением числа 2 в степень от 0 до 7. И каждая из полученных цифр записывается в таблицу от большего к меньшему. То есть слева направо. От 2 в 7-ой степени до 2 в 0-ой степени. Приведу таблицу степеней 2-ки.

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
128	64	32	16	8	4	2	1

Давайте теперь разберем приведенный для примера адрес «193.233.44.12» и посмотрим, как он выглядит в двоичном формате. Разберем каждый октет отдельно. Возьмем число 193 и посмотрим, из каких табличных комбинаций оно получается.  $128 + 64 + 1 = 193$ .

128	64	32	16	8	4	2	1
1	1	0	0	0	0	0	1

Те числа, которые участвовали в формировании комбинации получают 1, а все остальные получают 0. Берем первый октет 233.  $128 + 64 + 32 + 8 + 1$ .

128	64	32	16	8	4	2	1
1	1	1	0	1	0	0	1

Для 44 - это  $32 + 8 + 4$ .

128	64	32	16	8	4	2	1
0	0	1	0	1	1	0	0

И напоследок 12.  $8 + 4$ .

128	64	32	16	8	4	2	1
0	0	0	0	1	1	0	0

Получается длинная битовая последовательность 11000001.11101001.00101100.00001100. Именно с данным видом работают сетевые устройства. Битовая последовательность обратима. Вы можете так же вставить каждый октет (по 8 символов) в таблицу и получить десятичную запись. Представим совершенно случайную последовательность и приведем ее к десятичному виду. Пусть это будет 11010101.10110100.11000001.00000011. Строим таблицу и заносим в нее первый блок.

128	64	32	16	8	4	2	1
1	1	0	1	0	1	0	1

Получаем  $128 + 64 + 16 + 4 + 1 = 213$ .

Вычисляем второй блок.

128	64	32	16	8	4	2	1
1	0	1	1	0	1	0	0

Считаем:  $128 + 32 + 16 + 4 = 180$ .

Третий блок.

128	64	32	16	8	4	2	1
1	1	0	0	0	0	0	1

$128 + 64 + 1 = 193$ .

И четвертый блок.

128	64	32	16	8	4	2	1
0	0	0	0	0	0	1	1

$2 + 1 = 3$

Собираем результаты вычислений и получаем адрес 213.180.193.3.

**Задание №1, выполните ручной перевод IP адресов представленных десятичной записью в двоичную, а двоичные в десятичную:**

- 1) 10.124.56.220
- 2) 113.72.101.11
- 3) 173.143.32.194
- 4) 200.69.139.217
- 5) 88.212.236.76

- 6) 01011101.10111011.01001000.00110000
- 7) 01001000.10100011.00000100.10100001
- 8) 00001111.11011001.11101000.11110101
- 9) 01000101.00010100.00111011.01010000
- 10) 00101011.11110011.10000010.00111101

Выше мы говорили о структуре телефонных номеров и их иерархии, соответственно мы понимаем, что IP-адреса нужно как-то сгруппировать и контролировать их выдачу. Поэтому было решено разделить все пространство IP-адресов на классы.

Это решение получило название **классовая адресация** (от англ. **Classful**). Она уже давно устарела, но практически в любой книге на нее отводятся целые главы и разделы. Такая компания как Cisco тоже не забывает

Класс	Первые биты	Начальный адрес	Конечный адрес
A	0	0.0.0.0	127.255.255.255
B	10	128.0.0.0	191.255.255.255
C	110	192.0.0.0	223.255.255.255
D	1110	224.0.0.0	239.255.255.255
E	1111	240.0.0.0	255.255.255.255

про это и в своих учебных материалах рассказывает про нее. Пространство было поделено на 5 классов. Каждому классу был назначен блок адресов.

**Класса А.** Если внимательно посмотреть на таблицу, то можно заметить, что этому блоку дан самый большой блок адресов, а если быть точным, то половина всего адресного пространства. Предназначался данный класс для крупных сетей. Структура этого класса выглядит следующим образом.

Класс А			
Сеть	Хост	Хост	Хост

В чем суть. Первый октет, то есть 8 бит, остаются за адресом сети, а 3 последних октета (то есть оставшиеся 24 бита) назначаются хостам. Вот для того, чтобы показать, какой кусок относится к сети, а какой к хостам, используется **маска**. По структуре записи она аналогична записи IP-адреса. Отличие маски от IP-адресов в том, что 0 и 1 не могут чередоваться. Сначала идут 1, а потом 0. Таким образом, там где есть единица, значит это участок сети. Чуть ниже, после разбора классов, я покажу, как с ней работать. Сейчас главное знать, что маска класса А - 255.0.0.0. В таблице еще упомянут какой-то первый бит и для класса А он равен 0. Этот бит как раз нужен для того, чтобы сетевое устройство понимало, к какому классу оно принадлежит. Он же еще задает начальный и конечный диапазон адресов. Если в двоичном виде записать на всех октетах единицы, кроме первого бита в первом октете (там всегда 0), то получится 127.255.255.255, что является границей класса А. Например, возьмем адрес 44.58.63.132. Мы знаем, что у класса А первый октет

отдается под адрес сети. То есть «44» - это адрес сети, а «58.63.132» - это адрес хоста.

### **Класс В**

Класс В			
Сеть	Сеть	Хост	Хост

Этому классу был дан блок поменьше. И адреса из этого блока предназначались для сетей средних масштабов. 2 октета отданы под адрес сети, и 2 - под адрес хостов. Маска у В класса - 255.255.0.0. Первые биты строго 10. А остальные меняются. Перейдем к примеру: 172.16.105.32. Два первых октета под адрес сети - «172.16». А 3-ий и 4-ый под адрес хоста - «105.32».

### **Класс С**

Класс С			
Сеть	Сеть	Сеть	Хост

Этот класс обделили адресами и дали ему самый маленький блок. Он был предназначен для мелких сетей. Зато этот класс отдавал целых 3 октета под адрес сети и только 1 октет - под хосты. Маска у него - 255.255.255.0. Первые биты 110. На примере это выглядит так - 192.168.1.5. Адрес сети «192.168.1», а адрес хоста «5».

### **Классы D и E.**

Адреса из этих блоков зарезервированы и не могут назначаться сетям и хостам. Класс D предназначен для многоадресной рассылки. Аналогию можно привести с телевидением. Телеканал вещает группе лиц свой эфир. И те, кто подключены, могут смотреть телепередачи. То есть в распоряжение администраторов могут попасть только 3 первых класса.

Напомню, что первые биты у класса D - это 1110. Пример адреса - 224.0.0.5.

А первые биты у класса E - это 1111. Поэтому, если вдруг увидите адрес вида 240.0.0.1, смело говорите, что это адрес E класса.

### **Маски**

По идее IP адреса хосты понимают в каком они классе. Но суть использования маски вот в чем. Например, у вас есть маленький офис, и вам нужен блок IP-адресов. Никто не будет вам выдавать все адреса класса C. А дадут только его кусок. Например, 192.168.1.0 с маской 255.255.255.0. Так вот эта маска и будет определять вашу границу. Мы уже говорили, что октет варьируется в значении от 0 до 255. Вот этот 4 октет полностью в вашем распоряжении. За исключением первого адреса и последнего, то есть 0 и 255 в данном случае. Первый адрес - это адрес сети (в данном случае 192.168.1.0), а последний адрес - широковещательный адрес (192.168.1.255).

Напомним, что широковещательный адрес используется в том случае, когда надо передать информацию всем узлам в сети. Поэтому есть правило. Если вам надо узнать номер сети, то все биты, относящиеся к хосту обращаете

в 0, а если широковещательный, то все биты - в 1. Поэтому, если из 256 адресов забирается 2 адреса, то на назначение хостам остается 254 адреса (256 - 2). На собеседованиях часто любят задавать следующие два вопроса: «Количество IP-адресов в сети?» и «Сколько доступных IP-адресов в сети для назначения хостам?». Это два разных вопроса, хотя и могут показаться вам одинаковыми. Ответом на первый будет - все адреса, включая адрес сети и широковещательный адрес, а на второй вопрос - все адреса, кроме адреса сети и широковещательного адреса. Теперь углубимся в изучении маски.

Запишем адрес класса C 192.168.1.1 с маской 255.255.255.0 в

Десятичная запись IP-адреса	192	168	1	1
Двоичная запись IP-адреса	11000000	10101000	00000001	00000001
Десятичная запись маски	255	255	255	0
Двоичная запись маски	11111111	11111111	11111111	00000000

десятичном и двоичном формате. Обратите внимание на то, как выглядит IP-адрес и маска в двоичном формате. Если в IP-адресе 0 и 1 чередуются, то в маске сначала идут 1, а потом 0. Эти биты фиксируют адрес сети и задают размер. По таблице выше можно сделать вывод, что в двоичном виде маска представлена последовательностью 24 единиц подряд. Это говорит о том, что целых 3 октета выделено под сеть, а 4 октет свободен под адресацию для хостов. Здесь ничего необычного. Это стандартная маска класса C.

Допустим, что в офисе 100 компьютеров, и расширяться вы не планируете. Зачем плодить сеть из 250+ адресов, которые вам не нужны?! На помощь приходит разделение на подсети. Это очень удобная вещь. Объясню принцип на примере того же класса C. Как бы вы не хотели, но трогать 3 октета нельзя. Они фиксированы. Но вот 4 октет свободен под хосты, поэтому его можно трогать. Заимствуя биты из хостового куска, вы дробите сеть на n-ое количество подсетей и, соответственно, уменьшаете в ней количество адресов для хостов.

Попробуем это воплотить в реальность. Меняем маску. Заимствую первый бит из хостовой части (то есть 1-ый бит 4-ого октета выставляем в единицу). Получается следующая маска.

Двоичная запись маски	11111111.11111111.11111111.10000000
Десятичная запись маски	255.255.255.128

Данная маска делит сеть на 2 части. Если до дробления у сети было 256 адресов (от 0 до 255), то после дробления у каждого куска будет по 128 адресов (от 0 до 127 и от 128 до 255). Теперь посмотрю, что изменится в целом с адресами.

Двоичная запись IP-адреса 192.168.1.1	11000000.10101000.00000001.00000001
Двоичная запись маски 255.255.255.128	11111111.11111111.11111111.10000000

Красным цветом показаны те биты, которые зафиксированы и не могут изменяться. То есть маска ей задает границу. Соответственно биты помеченные черным цветом определены для адресации хостов. Теперь вычислим эту границу. Чтобы определить начало, надо все свободные биты

(помеченные черным цветом) обратить в ноль, а для определения конца обратить в единицы.

Начало сети или адрес сети	11000000.10101000.00000001.00000000
Конец сети или широковещательный адрес	11000000.10101000.00000001.01111111

То есть в четвертом октете меняются все биты, кроме первого. Он жестко фиксирован в рамках этой сети. Посмотрим на вторую половину сети и вычислим ее адреса. Деление у нас производилось заимствованием первого бита в 4-ом октете, значит он является делителем (разделителем, разграничителем). Первая половина сети получалась, когда этот бит принимал значение 0, а значит вторая сеть образуется, когда этот бит примет значение 1. Меняем этот бит в 1 и посмотрим на границы.

Начало сети или адрес сети	11000000.10101000.00000001.10000000
Конец сети или широковещательный адрес	11000000.10101000.00000001.11111111

Приведем в десятичный вид.

Начало сети или адрес сети в 10-ом формате	192.168.1.128
Конец сети или широковещательный адрес в 10-ом формате	192.168.1.255

Соответственно .128 и .255 назначать хостам нельзя. Значит в доступности 128-2=126 адресов. Вот таким образом можно при помощи маски управлять размером сети. Каждый заимствованный бит делит сеть на 2 части. Если откусить 1 бит от хостовой части, то поделим на 2 части (по 128 адресов), 2 бита = 4 части (по 64 адреса), 3 бита = 8 (по 32 адреса) и так далее.

Если вы рассчитали количество бит, отдаваемые под хосты, то количество доступных IP-адресов можно вычислить по формуле  $2^H - 2$ .

Существует хорошая формула для расчета битов, отдаваемых на подсеть и хосты:

$$N + S + H = 32,$$

где **N** - кол-во битов сети (класс А - 8 бит, В - 16 бит, С - 24 бита), **S** - кол-во заимствованных битов на подсеть (это то, что мы делали выше, когда заимствовали 1 бит из хостовой части), **H** - кол-во бит отводимых хостам.

Внесем ясность, как и где применять эти формулы.

Возьмем пример: Нам выдали сеть 172.16.0.0 и попросили создать 120 подсетей со 180 хостами и записать маску.

В качестве шпаргалки, и для быстроты вычисления, ниже подготовлена таблицу степеней двойки.

Двигаемся дальше. Первое главное условие, при использовании классовой адресации - это то, что должна использоваться одна маска для всех подсетей. То есть, если у вас для одной подсети маска 255.255.255.0, то для другой подсети она не может быть 255.255.255.128.

Теперь смотрим на выданную сеть. Путем логических размышлений понимаем, что это адрес класса В. А значит его **N** (кол-во битов сети) = 16. Ок. Значит на хосты выделено тоже 16 бит. Вспоминаем условия задачи. Нужно



создать 120 подсетей. «Откусывать» биты от сетевой части запрещено, значит кусаем от хостовой части.

n	2^n	n	2^n
0	1	16	65536
1	2	17	131072
2	4	18	262144
3	8	19	524288
4	16	20	1048576
5	32	21	2097152
6	64	22	4194304
7	128	23	8388608
8	256	24	16777216
9	512	25	33554432
10	1024	26	67108864
11	2048	27	134217728
12	4096	28	268435456
13	8192	29	536870912
14	16384	30	1073741824
15	32768	31	2147483648

Теперь нужно взять такое кол-во бит, чтобы хватило для 120 подсетей, однако оставляло достаточное кол-во под биты для хоста. Смотрим на таблицу выше. Если взять 7 бит, то получим 128.  $128 > 120$ , следовательно попадаем под условие. Если возьмем 6 бит, то получим 64.  $64 < 128$ , поэтому не попадаем под условие и отбрасываем этот вариант.

Выяснили, что S надо выделить не меньше 7 бит. Теперь посмотрим, что «осталось» под хосты.

$$\text{Если } N + S + H = 32 \Rightarrow H = 32 - (N + S) \Rightarrow H = 32 - (16 + 7) = 9.$$

Смотрим на таблицу выше (или возводим 2 в 9 степень в уме. На калькуляторе или в ручную) и получаем число 512. Отнимаем 2 (адрес сети и широковещательный адрес) и получаем 510 адресов. Нам нужно 180, а значит под условие соблюдается с большим запасом.

В таких случаях вам предоставляется право выбора. Сделать больше подсетей или хостов на подсеть. Объясняю, что это значит. У нас есть 9 бит на хосты. Если мы возьмем 8 бит, то получим число 256.  $256 - 2 = 254$  адреса. Этот вариант нам тоже подходит. Возьмем 7 бит. Получаем 128. Даже не отнимая 2 адреса, становится понятно, что это меньше 180  $\Rightarrow$  данный вариант отбрасывается сразу.

Итого получаем, что минимальное количество для подсети - 7 бит, а для хостов - 8 бит. Поэтому свободный бит можно отдать либо на подсеть, либо на хосты.

Маска получается сложением N и S. В нашем случае получаем, если под подсеть отдаем 7 бит, то получаем 23. В десятичном виде маска будет выглядеть 255.255.254.0. А если отдадим под подсеть 8 бит, то получим 24 (или в десятичном виде 255.255.255.0).



Иногда бывает, что под задачу существует всего одна маска. Ну и, конечно, могут быть случаи, когда маска не попадает не под какие условия. В этих случаях нужно брать сеть другого класса или доказывать заказчику, что это невозможно.

Таким образом работает классовая адресация, и связанные с ней расчеты. Для закрепления целесообразно решить следующие задачи:

#### **Задание №2**

- 1) Записать маску для проекта: сеть 172.16.0.0. 250 подсетей и 220 хостов.
- 2) Записать маску для проекта: сеть 10.0.0.0. 2000 подсетей и 1500 хостов.
- 3) Записать маску для проекта: сеть 192.168.0.0. 4 подсети и 60 хостов.

Таким образом мы получили теоретические основы и некоторые навыки по работе с IP адресами и масками.

В рамках данной практической работы следует выполнить не только расчеты в ручном режиме, но и программную реализацию данных расчетов.

#### **Задание №4**

Реализовать программное средство для представления IP адресов с десятичного и двоичного кодирования. В качестве входных данных задается десятичное представление IP адреса, программное средство выдает двоичное представление или пользователь может ввести двоичное представление IP адреса и получить десятичное представление.

#### **Задание № 5**

Реализовать программное средство, способное определять класс сети, начальный и конечный адрес и маску подсети, на основании заданного IP адреса.

#### **Задание № 6**

Разработать программное средство, которое по заданной сети, количеству подсетей и хостов формирует: маску для проекта сети, а также определяет класс сетей, начало и конец сети, количество IP-адресов в сети, количество доступных IP-адресов в сети для назначения хостам, стек первых 5 допустимых IP-адресов, стек последних 5 допустимых IP-адресов.

Отчет оформляется в соответствии с требованиями:

Наличие титульного листа;

Наличие ручных вычислений по заданиям 1 и 2;

Наличие программного решения заданий 3-6;

Скриншот работающих программных средств;

Листинг программной реализации;

Выводы.