

Workshop 1:

Workshop1涉及到的主题:

- 二进制
- 十六进制
- “与”操作

1: 二进制数学

作为了解网络是如何工作的，你需要对二进制算法有很好的理解。这是为什么呢？因为网络设备所呈现出来的一些操作是通过二进制算法来完成的，比如一下应用就会使用到二进制数学的知识：

- 解析网络首部字段
- 使用计算机的子网掩码
- 确定一个分组是否应当被转发给目的IP地址

所以，让我们来了解基本的二进制算法，然后做一些练习。

1.1 引言

任何数字都可以通过无限多的方式表示出来，而不需要改变数字本身。比如，一打鸡蛋的数量总是相同的(12个)。然而，将数字写在纸上的方式可以有很多种。比如，鸡蛋的数目是：

- 一打(汉语)
- 12(十进制数)
- XII(罗马数字)
- 1100(二进制)

上述所表达的都是同一个数字。我们之所以在计算机中非常频繁的使用二进制来表达数字，这是由计算机存储和处理数字的方式所决定的。
二进制表示法和十进制表示法有一些相似之处

数的十进制表示	数的二进制表示
最右边的列是有意义的	最右边的列是有意义的
每一列的值是其右边列的值的10倍	每一列的值是其右边列的值的2倍
有固定数目的标识符：0, 1, 2, 3, 4, 5, 6, 7, 8, 9.	有固定数目的标识符：0, 1.
0代表这一列没有值。最前面的0是可选的	0代表这一列没有值。最前面的0是可选的

1.2 二进制表示法

基于上面的介绍，现在我们可以看到，为了计算出一个二进制数的值，就像在十进制中所做的一样，我们只需要将列的值相加即可。例如：

1
2 6 3 1
8 4 2 6 8 4 2 1

 0 1 1 0 1 0 1 1<-- 二进制数字

请注意，我已经在数字上方标示出了每一列的值。数字01101011是一个普通的数字，它只是恰巧用二进制的方式表达了出来。如果我们想将它转换成十进制，我们只需将各列的值相加即可：

01101011 in binary = $64 + 32 + 8 + 2 + 1 = 107$ （十进制）

你应当至少熟悉下面的8列数值，并且你应当注意到每一个值都是2的若干次幂：

$2^0 = 1$
 $2^1 = 2$
 $2^2 = 4$
 $2^3 = 8$
 $2^4 = 16$
 $2^5 = 32$
 $2^6 = 64$
 $2^7 = 128$
 $2^8 = 256$
 $2^9 = 512$
 $2^{10} = 1024$
 $2^{11} = 2048$
 $2^{12} = 4096$ 等等.

你还需要牢记的是一个所有位都是二进制1的八位位组(即一个包含8比特位的位组)等价于十进制的255，也就是 $1111\ 1111$ (二进制) = 255(十进制)

1.3 将十进制转换为二进制

最简单的方法是用十进制数减去适当的2的若干次幂，直到结果为零时停止，然后计算出哪一列需要标识1，而哪一列没有使用到，用0对其进行标记。比如，将217转换成二进制：

不能够减去256(这个数字太大了)，因此可以减去128：

$217 - 128 = 89$
 $89 - 64 = 25$
 $25 - 16 = 9$
 $9 - 8 = 1$
 $1 - 1 = 0$

所使用到的2的幂有： $128 + 64 + 16 + 8 + 1$

故所表示的二进制数是： $1101\ 1001$

通常，我们会在四个比特位组之间增加一个空格，来增加可读性。

1.4 用二进制计数

你或许经常需要用二进制来计数。一个解决方法是将序列用十进制写出来，然后手动将其转换

为二进制。当然还有一个更好的方法，就是观察在二进制计数时的循环模式。

- 最右列以0开始，就改变每一个数字
- 下一列以0开始，就每两个数字变换一次
- 下一列以0开始，就每四个数字变换一次

所以：

0000 0000= 0
0000 0001= 1
0000 0010= 2
0000 0011= 3
0000 0100= 4
0000 0101= 5
0000 0110= 6
0000 0111= 7
0000 1000= 8
0000 1001= 9
0000 1010= 10
0000 1011= 11
0000 1100= 12
0000 1101= 13
0000 1110= 14
0000 1111= 15 等等.

1.5 二进制练习

在下面的表格中，将左列中的二进制数转换为十进制数，然后将右边的十进制数转换为二进制数。

Binary	Decimal
0001 1011	
0101 0110	
1001 1110	
0010 0011	
1101 0100	
	27
	206
	191
	88

1.6 IP地址：点分十进制和点分二进制表示法

继续深入今天的主题，我们会看到所有连接到Internet上的计算机都有一个以数字表示的地址，称为网络地址或IP地址。这些网络地址的长度是32位(bit)。为了能够很容易的阅读这些地址，人们将整个地址划分为4个字节(每个字节8位)，中间用圆点分割。
如果数字使用十进制表示的，我们称这种表示法为“点分十进制表示法”，
如果数字使用二进制表示的，我们称这种表示法为“点分二进制表示法”
要注意的是，十进制数字绝不能超过255. 你能够解释为什么吗？
在有些场合，我们不得不将二进制表示的IP地址转换成十进制表示的IP地址，或者反过来做。
这里有一些用点分法表示IP地址的例子。对于用“点分十进制表示法”的地址，请将其转换

为“点分二进制表示法”的地址。反之亦然。

十进制IP地址	二进制IP地址
10. 12. 3. 45	
131. 236. 21. 1	
212. 95. 255. 255	
18. 107. 240. 0	
	1011 1100 . 0000 0110 . 0101 1101 . 0111 1111
	0010 0111 . 1001 1101 . 0011 0111 . 1110 1011
	1101 1000 . 0010 0001 . 0111 1101 . 1001 1010

2：十六进制数学

因为计算机通常使用二进制，有些时候直接使用二进制会很方便，而不需要先将二进制转换成十进制，然后再将其转换回二进制。但是我们用肉眼去阅读二进制是非常困难的，所以早先的计算机开发者们想出来一个不同的表示法——“十六进制表示法”，这种表示法和二进制类似，但是对人们阅读会更容易。

简单的说，十六进制就是将二进制中的每个4比特位组转换成一个特殊的标识符。因为有4比特位，所以我们需要 $2^4=16$ 个标识符；这也是它为什么叫做十六进制的原因。

Binary	Hexadecimal	Binary	Hexadecimal
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

如果比特位多于4个，我们可以对它们进行4bit为一组的方式进行分组。比如以太网卡有一个内置的48位的标识，即MAC地址。如果将MAC地址用二进制表示出来，如下所示：

0000000000000000011000000100011110101010111111000

然而如果对它们进行4位一组进行分组并进行转换的话，我们能够对上述地址用十六进制进行重写，这样阅读起来就非常容易了：

0000C08F55F8

事实上，以太网地址经常用冒号（“:”）或者减号（“-”）来帮助人们更加方便的阅读：

00:00:C0:8F:55:F8

2.1 十六进制练习

完成以下二进制和十六进制之间的转换

二进制	十六进制
00101101	

10110011	
11111000	
00011000	
	27
	BD0F
	C300A6

3: 使用“AND”屏蔽比特

当处理IP地址的时候，我们经常会从一个IP地址中分割出若干位，因为这若干位有着特定的意义：这个IP地址所在的网络地址。我们希望将非网络地址位设置为零。可以通过在地址的每一位上使用网络掩码（子网掩码）和AND操作来达到这个目的。

AND操作有这些属性：

0 AND 0 = 0
0 AND 1 = 0
1 AND 0 = 0
1 AND 1 = 1

可以看到，只有当AND左边的操作数和右边的操作数同时为1时结果才为1. 更重要的是，从我们的角度来看，结果是这样的：

X AND 0 总是 0
X AND 1 总是 X

通过复习上一个表格的内容可以验证我们的推测是正确的。现在，要看仔细了。我们有一个32位的IP地址，并且我们知道左手边的16位代表了这个IP地址的网络地址。我们想保留这16位网络地址，而将其他位填充为零。通过将IP地址和一个前16位都是1，后16位都是0的网络掩码进行AND操作，就可以达到我们的目的：

IP地址	1010 0001 1111 1101 0001 0110 1001 0110
网络掩码	1111 1111 1111 1111 0000 0000 0000 0000
AND结果	1010 0001 1111 1101 0000 0000 0000 0000

可以看到，通过使用网络掩码和AND操作已经可以将我们不需要的位全部填充为零了。当然，我们可以通过用十六进制表示法来使书写更加简短：

IP地址	A1FD 1696
网络掩码	FFFF 0000
AND结果	A1FD 0000

3.1 “AND”操作练习

应用下面练习中提供的网络掩码来计算出每个IP地址所属的网络地址

IP地址	0001 0010 0110 1011 1000 1110 1000 0001
网络掩码	1111 1111 1111 0000 0000 0000 0000 0000
AND结果	
IP地址	0010 1110 0100 0000 1011 0001 0100 0111
网络掩码	1111 1111 1111 1111 1000 0000 0000 0000
AND结果	

IP地址	83EC 0156
网络掩码	FF00 0000
AND结果	
IP地址	4743 724E
网络掩码	FFFF C000
AND结果	