

Base64

维基百科，自由的百科全书

Base64是一种基于64个可打印字符来表示二进制数据的表示方法。由于**2⁶ = 64**，所以每6个位元为一个单元，对应某个可打印字符。3个字节有24个位元，对应于4个Base64单元，即3个字节可由4个可打印字符来表示。它可用来作为电子邮件的传输编码。在Base64中的可打印字符包括字母A-Z、a-z、数字0-9，这样共有62个字符，此外两个可打印符号在不同的系统中而不同。一些如uuencode的其他编码方法，和之后BinHex的版本使用不同的64字符集来代表6个二进制数字，但是不被稱為Base64。

Base64常用于在通常处理文本数据的场合，表示、传输、存储一些二进制数据，包括MIME的电子邮件及XML的一些复杂数据。

目录

- MIME
 - 例子
- UTF-7
- IRCu
- 在URL中的应用
- 其他应用
- 相关事件
- 参见
- 参考资料
- 外部链接

MIME

在MIME格式的电子邮件中，Base64可以用来将binary的字节序列数据编码成ASCII字符序列构成的文本。使用时，在传输编码方式中指定Base64。使用的字符包括大小写拉丁字母各26个、数字10个、加号+和斜杠/，共64个字符，等号=用来作为后綴用途。

完整的Base64定义可见RFC 1421和RFC 2045。编码后的数据比原始数据略长，为原来的**4⁄3**。在电子邮件中，根据RFC 822规定，每76个字符，还需要加上一个回车换行。可以估算编码后数据长度大约为原长的135.1%。

转换的时候，将3字节的数据，先后放入一个24位元的缓冲区中，先来的字节占高位。数据不足3字节的话，於緩衝區中剩下的位元用0补足。每次取出6位元（因为**2⁶ = 64**），按照其值选择ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/中的字符作为编码后的输出，直到全部输入数据转换完成。

若原数据长度不是3的**倍數**时且剩下1個輸入數據，則在編碼結果後加2個=；若剩下2個輸入數據，則在編碼結果後加1個=。

记数系统
印度-阿拉伯数字系统
<div> <div>西方阿拉伯数字</div> <div>阿拉伯文数字</div> <div>高棉數字</div> </div> <div> <div>印度數字</div> <div>波羅米數字</div> <div>泰语数字</div> </div>
汉字文化圈記數系統
<div> <div>中文数字</div> <div>閩南語數字</div> <div>越南语数字</div> <div>算筹</div> </div> <div> <div>日語數字</div> <div>朝鲜文数字</div> <div>苏州码子</div> </div>
字母記數系統
<div> <div>阿拉伯字母數字</div> <div>亞美尼亞數字</div> <div>西里爾數字</div> <div>吉茲數字</div> </div> <div> <div>希伯來數字</div> <div>希腊数字</div> <div>阿利耶波多數字</div> </div>
其它記數系統
<div> <div>雅典數字</div> <div>巴比倫數字</div> <div>古埃及數字</div> <div>伊特拉斯坎數字</div> </div> <div> <div>玛雅数字</div> <div>罗马数字</div> </div>
依底数区分的进位制系统
<div>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 20</div> <div>24 30 32 36 60 64</div>

例子

舉例來說，一段引用自托馬斯·霍布斯《利維坦》的文句：

Man is distinguished, not only by his reason, but by this singular passion from other animals, which is a lust of the mind, that by a perseverance of delight in the continued and indefatigable generation of knowledge, exceeds the short vehemence of any carnal pleasure.

經過Base64編碼之後變成：

TWFuIGlziGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpYByZWZb24sIGJ1dCBieSB0aGlzIHNPbmd1bGFyIHh3c3Npb24gZnJvbSBvdGhlciBhbmltYWxzLCB3aG1jaCBpcyBhIGx1c3Qgb2YgdGhlIG1pbmQsIHROeXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpb0aGUy29udGluZGVkIGFuZCBpbmRlZmF0aWdhYmxiIGd1bmV5YXRpb24gb2Yga25vd2x1ZGd1LCBleGNlZWZlIHROZSBzaG9ydCB2ZWh1bWVuY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS4=

■ 编码「Man」

文本	M								a								n							
ASCII编码	77								97								110							
二进制位	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
索引	19								22								5							
Base64编码	T								W								F							

在此例中，Base64算法将3个字节编码为4个字符。

Base64索引表：

数值	字符	数值	字符	数值	字符	数值	字符
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

如果要编码的字节数不能被3整除，最后会多出1个或2个字节，那么可以使用下面的方法进行处理：先使用0字节值在末尾补足，使其能够被3整除，然后再进行Base64的编码。在编码后的Base64文本后加上一个或两个=号，代表补足的字节数。也就是说，当最后剩余两个八位字节（2个byte）时，最后一个6位的Base64字节块有四位是0值，最后附上两个等号；如果最后剩余一个八位字节（1个byte）时，最后一个6位的base字节块有两位是0值，最后附加一个等号。参考下表：

文本 (1 Byte)	A																							
二进制位	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
二进制位 (補0)	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Base64编码	Q							Q					=				=							
文本 (2 Byte)	B								C															
二进制位	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0
二进制位 (補0)	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Base64编码	Q							k					M				=							

UTF-7

UTF-7是一个修改版Base64 (**Modified Base64**)。主要是将UTF-16的数据，用Base64的方法编码为可打印的ASCII字符序列。目的是传输Unicode数据。主要的区别在于不用等号=补餘，因为该字符通常需要大量的转译。

标准可见 [RFC 2152](#)，《A Mail-Safe Transformation Format of Unicode》。

IRCu

在IRCu等软件所使用的P10 IRC服务器间协议中，对客户与服务器的消息类型号（client/server numerics）和二进制IP地址采用了Base64编码。消息类型号的长度固定为3字节，故可直接编码为4个字节而不需要加填充。对IP地址进行编码时，则需要在地址前添加一些0比特，使之可以编码为整数个字节。这里所用的符号集与前述MIME的也有所不同，将+/改成了[]。

在URL中的应用

Base64编码可用于在HTTP环境下传递较长的标识信息。例如，在Java持久化系统Hibernate中，就采用了Base64来将一个较长的唯一标识符（一般为128-bit的UUID）编码为一个字符串，用作HTTP表单和HTTP GET URL中的参数。在其他应用程序中，也常常需要把二进制数据编码为适合放在URL（包括隐藏表单域）中的形式。此时，采用Base64编码不仅比较简短，同时也具有不可读性，即所编码的数据不会被人用肉眼所直接看到。

然而，标准的Base64并不适合直接放在URL裡传输，因为URL编码器会把标准Base64中的/和+字符变为形如%XX的形式，而这些%号在存入数据库时还需要再进行转换，因为ANSI SQL中已将%号用作通配符。

为解决此问题，可采用一种**用于URL的改进Base64**编码，它不在末尾填充=号，并将标准Base64中的+和/分别改成了-和_，这样就免去了在URL编解码和数据库存储时所要作的转换，避免了编码信息长度在此过程中的增加，并统一了数据库、表单等处对象标识符的格式。

另有一种**用于正则表达式的改进Base64**变种，它将+和/改成了!和-，因为+，*以及前面在IRCu中用到的[和]在正则表达式中都可能具有特殊含义。

此外还有一些变种，它们将+/改为_-或._（用作编程语言中的标识符名称）或.-（用于XML中的Nmtoken）甚至_：（用于XML中的Name）。

为了克服Base64由于输出内容包括两个以上“符号类”字符（+, /, =等）而带来的互不兼容多变种问题，一种输出内容无符号的Base62x编码方案被引入软件工程领域，Base62x被视为无符号化的Base64改进版本。

其他应用

- 垃圾訊息傳播者用Base64來避過反垃圾郵件工具，因為那些工具通常都不會翻譯Base64的訊息。
- 在LDIF檔案，Base64用作编码字符串。

相关事件

- 2018年2月电子邮件程序 Exim 发现重大漏洞，编号为 [CVE-2018-6789](#) (<https://exim.org/static/doc/security/CVE-2018-6789.txt>) 的緩衝區溢位漏洞允许攻击者在服务器上远程执行恶意代码。漏洞位于 base64 解码函数中，影响 Exim v4.90.1 之前的所有版本，多达 40 万服务器受到影响。^{[1][2]}

参见

- [Radix-64](#)
- [Ascii85](#)

- [Quoted-printable](#)
- [uuencode](#)
- [yEnc](#)
- [8BITMIME](#)
- [URL](#)

参考资料

1. <https://devco.re/blog/2018/03/06/exim-off-by-one-RCE-exploiting-CVE-2018-6789-en/>
2. <https://www.solidot.org/story?sid=55710>

外部链接

- [RFC 1421](#) (Privacy Enhancement for Electronic Internet Mail)
- [RFC 2045](#) (MIME)
- [RFC 3548](#) (The Base16, Base32, and Base64 Data Encodings) 在 [RFC 4648](#) 中被替代
- [Home of the Base64 specification, with an online decoder and C99 implementation \(http://josefsson.org/base-encoding/\)](#)
- [Base64在线解码工具 \(https://codebeautify.org/base64-decode\)](https://codebeautify.org/base64-decode)

取自“<https://zh.wikipedia.org/w/index.php?title=Base64&oldid=54048492>”

本页面最后修订于2019年4月16日 (星期二) 23:24。

本站的全部文字在知识共享 署名-相同方式共享 3.0协议之条款下提供，附加条款亦可能应用。（请参阅[使用条款](#)）
Wikipedia®和维基百科标志是维基媒体基金会的注册商标；维基™是维基媒体基金会的商标。
维基媒体基金会是按美国国内税收法501(c)(3)登记的[非营利慈善机构](#)。