情報技術の基礎 3

情報技術の基礎 3

第 3 章　文字コード

3-1 ASCII

(a)　データ交換のための標準文字コード

コンピュータの歴史を振り返ってみると、その黎明期においては多くのメーカがお互いに互換性のない独自のコンピュータを開発していた。コンピュータが 1 つの会社、1 つの研究所内でだけ用いられている間は何も混乱は起きなかったが、異なるコンピュータを用いている会社間あるいは研究所間でデータの交換をしようとすると、お互いのコンピュータで取り決めている「文字コード」が異なっているために素直にデータ交換ができないという問題が発生した。そこで、アメリカの工業標準を定める ANSI がデータ交換用の標準的な文字コードとして定めたのが ASCII である。

この ASCII はその最後の「II」の部分が”Information Interchange”という正式名称の頭文字であることからもわかる通り、本来はデータ交換用途に限って規定されたものであるが、実際には多くのコンピュータメーカがデータ交換時に限らず、コンピュータ内部で文字を処理する際の文字コードとしても ASCII を採用したため、ASCII はあらゆる場面における文字コードの基本となった。

(b)　ASCII は 7 ビット単位の文字コード

英語の文字（アルファベット）と数字、それに「＋」や「＝」などの記号に対して 2 進数で固有の番号を割り振るとしたら、全部で何ビットが必要になるのであろうか。アルファベットが「A」～「Z」の大文字 26 個と「a」～「z」の小文字 26 個を合わせて 52 個、そして数字が「0」～「9」の 10 個、記号がおよそ 30 個とすると、計 92 個。これを 2 進数で表すと 6 ビット、すなわち、64 個（$=2^6$）では足りず、7 ビット、すなわち、128 個（$=2^7$）あれば、すべての文字に余裕を持って固有の番号を割り振ることができる計算になる。そこで、ASCII では 7 ビット単位で 1 つの文字を表現するしくみが採用されている。

パソコンの CPU は当初、8 ビット（=1 バイトとも言う）単位でデータを演算したり、メモリとやり取りしたりする構造が採用された。その後、パソコンの CPU が一度に処理できるデータサイズは「8 ビット→16 ビット→32 ビット→64 ビット」と 8 の倍数単位で進化してきたが、7 ビット単位で 1 つの文字を表現するしくみだけはずっと変わらなかった。そこで、英語のみを扱うアメリカ国内のコンピュータでは 1 バイトの最上位ビット=8 ビット目を無視したり、別の目的に利用したりして、残りの 7 ビットだけを用いて 1 つの文字を表現する方法が利用されてきた。インターネットメールでは「7 ビットの ASCII の英数字だけを送受信できる」決まりが基本となっているが、これもアメリカ国内におけるこのような 8 ビット目を使わない慣習が背景となって生まれた。

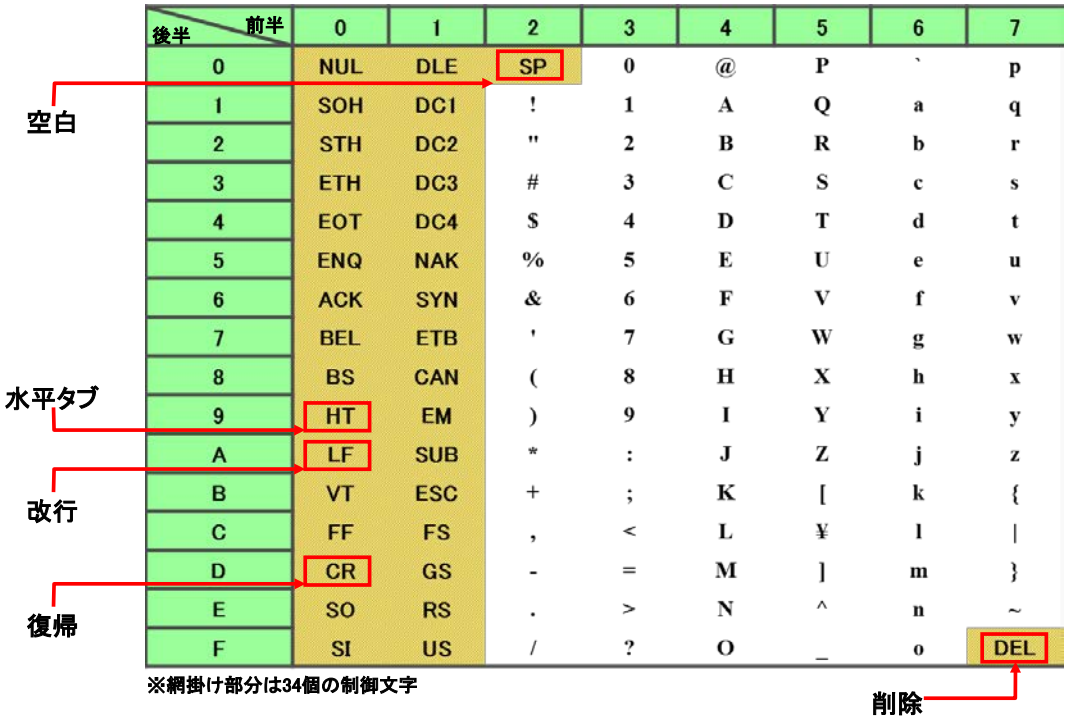
(c)　ASCII の文字集合

ASCII には、「A」～「Z」の英大文字、「a」～「z」の英小文字、「0」～「9」の数字、「+」や「$」などの記号が 32 文字収録されているほかに、空白を表す空白文字 1 個を含む 34 個の”制御文字（コントロールコード）”と呼ばれる特殊な文字が収録されている。

この制御文字は、ASCII が開発された当時に汎用コンピュータの端末として一般的だったテレタイプ装置を制御する働きを文字コードの一部に盛り込んだものである。当時のテレタイプ装置はライプライタに毛がはえた程度の原始的な機能しか持っていなかったため、文字コード中に定めた制御コードですべての機能を制御することができた。しかし、今日のように、通信手順も複雑になり、グラフィカルな表示装置や印刷装置が登場してくると、文字コードの一部として制御することはできなくなったので、

- 水平タブを表す「HT (Horizontal Tab)」
- 改行を表す「LF (Line Feed)」
- 復帰を表す「CR (Carriage Return)」
- 削除を表す「DEL (Delete)」
- 空白を表す「SP (SPACE)」

などを除くと今では制御文字の多くがその本来の働きを失っている。



後半＼前半	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STH	DC2	"	2	B	R	b	r
3	ETH	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	¥	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

図１　ASCII の文字集合

3-2 ヨーロッパ各国の文字コード

(a)　ISO 646

まず、第一ステップでは、7 ビット構造の ASCII をほぼそのまま採用したうえで、その中の一部の文字や記号を各国向けに入れ替えた文字コードがイギリスやドイツなどの各国独自の「国内規格」として開発された。世界の工業標準を定める ISO でも ISO 646 として規格化されて、さらに多くの国々で自国用バージョンが作られている。具体的には、ASCII の中の 23h・24h・40h・5Bh・5Ch・5Dh・5Eh・60h・7Bh・7Ch・7Dh・7E にある計 12 箇所のコード位置の文字は各国事情に応じて入れ替え可能とし、残りの箇所は ASCII の文字をそのまま採用する決まりとなっている。ASCII を元に、一部の文字を入れ替えただけの文字コードであるので、各国版規格に対応したフォントされ用意すれば、アメリカ産のハードウェアおよびソフトウェアを改造することなく、各国語の固有文字を表示できるようになった点が最大のメリットである。

しかし、ISO 646 では 7 ビットで表せるコード領域、すなわち、128 個のコード領域に元々あまり余裕がなかったため、各国語のすべての固有文字を表現することができなかったことに加えて、当然ながら、1 つの文書内で英語とフランス語、あるいは、フランス語とドイツ語など、「2 か国以上の文字を混在させることができない」という制約があった。そこで、より多くの固有文字を表現し、かつ、2 つ以上の国語の文字を混在させるために、ASCII が規定していた 7 ビットのコード領域を 8 ビットに拡張したうえで、その前半と後半に別々の ISO 646 各国版文字コードを割り振って、随時、この 2 つの文字コードを切り替えながら、同時に 2 つ（以上）の国語の文字を表現できるしくみが開発された。

この 8 ビットに拡張した文字コード領域の中で複数の文字コードを切り換えることによって扱える文字を拡張するしくみは、ISO 2022 として国際的に規格化され、日本においても JIS X 2022 として規格化されている。

(b)　ISO 8859

ヨーロッパ各国における文字コード現地化の第二ステップでは、ISO 2022 で文字コード領域が 8 ビットに拡張されたことを受けて、8 ビットのコード領域の前半は ASCII のままとし、後半のコード領域に各国語固有の文字・記号を収録した ISO 8859 が開発された。この ISO 8859 は、今日、ヨーロッパ諸国でもっとも広く利用されている文字コードとなっている。ISO 8859 の中では、最初に制定された西ヨーロッパ諸国のフランス語やドイツ語などのラテン文字をまとめて収録した ISO 8859-1 が有名であるが、地域別（言語別）に多くのバージョンが開発されている。

ちなみに、7 ビットの ASCII を基本として運用されていたインターネットメール世界で、ASCII

の英数字に加えて、日本語のひらがなや漢字を扱うために開発された ISO-2022-JP はこの ISO 2022 のサブセットと言える。この ISO 8859 によって、余裕をもって各国語固有文字を表現できるようになっただけでなく、たとえば、”Latin-1”という呼び名のある ISO 8859-1 には西ヨーロッパ諸国の固有文字がまとめて収録されているので、とくに文字コードを意識することなく、1 つの文書内で英語とフランス語とドイツ語といった西ヨーロッパ諸国の複数国語の文字を簡単に混在させることができるようになった。

図 2　ISO 646 の各国版

アメリカや日本に対抗しうる強力な経済圏となることを目指してきた EU 諸国にとって、経済活動においてすでに必要不可欠の存在となっているコンピュータ上で面倒な文字コードの切り替え操作を介することなく各国の文字を簡単に同時に扱えるようになることは、時代的な要請だったと言える。ちなみに、Windows や Mac OS に添付しているほとんどの欧文フォントには、ISO 8859-1 で規定されている文字がすべて収録されている。日本語版 Windows では、添付の「文字コード表」アプレットを用いて、フォントの種類に《Arial》や《Times New Roman》などの欧文フォントを指定すると、西ヨーロッパ諸国の文字を収録した ISO 8859-1 の文字集合を確認することができる。

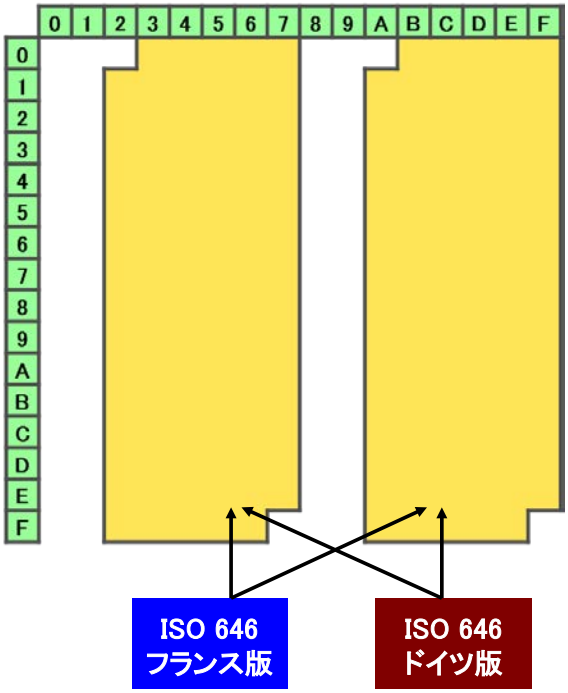


図 3　ISO 2022 での文字切り替え

ISO 8859	対象地域：含まれているおもな言語
ISO 8859-1 (Latin1)	West European: Albanian, Basque, Catalan, Danish, Dutch, Faroese, Finnish, French, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Rhaeto-Romanic, Scottish, Spanish, Swedish
ISO 8859-2 (Latin2)	East European: Croatian, Czech, Hungarian, Polish, Romanian, Slovak, Slovenian, Sorbian
ISO 8859-3 (Latin3)	South Europian: Esperanto, Maltese, Turkish (before Latin5/1988)
ISO 8859-4 (Latin4)	North European: Estonian, Latvian, Lithuanian Nordic: Greenlandic, Lappish (Latin6 も参照)
ISO 8859-5 (Latin/Cyrillic)	Cyrillic: Bulgarian, Byelorussian, Macedonian, Russian, Serbian, pre-1990, Ukrainian
ISO 8859-6 (Latin/Arabic)	Arabic: Arabic
ISO 8859-7 (Latin/Greek)	Greek: (modern monotonic) Greek
ISO 8859-8 (Latin/Hebrew)	Hebrew: Hebrew, Yiddish
ISO 8859-9 (Latin5)	Turkish (Latin1 の改訂): Turkish Latin-1のIcelandicを削除し、トルコ語固有文字を収録
ISO 8859-10 (Latin6)	Nordic (Latin4 の改訂): Latin4 + Inuit (Greenlandic Eskimo) + non-Skolt Sami (Lappish) + Icelandic
ISO 8859-11 (Latin/Thai)	Thai: Thai = TIS620
ISO 8859-12	reserved for ISCII Indian
ISO 8859-13 (Latin7)	Baltic Rim: Latin-6に収録されていなかったLatvianを収録
ISO 8859-14 (Latin8)	Celtic (Latin1 の改訂): Gaelic, Welsh
ISO 8859-15 (Latin9)	West European (Latin1 の改訂): ユーロ記号とフランス語に欠けていた3文字を収録
ISO 8859-16 (Latin10)	East European (Latin2 の改訂): Croatian, Hungarian, Polish, Romanian, Slovenian; Latin2 から Czech , Slovak, Sorbian を除き、Romanianに完全収録

図 4　ISO 8859 のバージョン

ISO 8859-1 (Latin-1)

	A	B	C	D	E	F
0		°	À	Đ	à	ð
1	¡	±	Á	Ñ	á	ñ
2	¢	²	Â	Ò	â	ò
3	£	³	Ã	Ó	ã	ó
4	¤	´	Ä	Ô	ä	ô
5	¥	µ	Å	Õ	å	õ
6	¦	¶	Æ	Ö	æ	ö
7	§	·	Ç	×	ç	÷
8	¨	¸	È	Ø	è	ø
9	©	¹	É	Ù	é	ù
A	ª	º	Ê	Ú	ê	ú
B	«	»	Ë	Û	ë	û
C	¬	¼	Ì	Ü	ì	ü
D		½	Í	Ý	í	ý
E	®	¾	Î	Þ	î	þ
F	¯	¿	Ï	ß	ï	ÿ

ISO 8859-2 (中央ヨーロッパ各国の文字)

	A	B	C	D	E	F
0		°	Ŕ	Đ	ŕ	đ
1	Ą	ą	Á	Ń	á	ń
2	˘	¸	Â	Ň	â	ň
3	Ł	ł	Ă	Ó	ă	ó
4	¤	´	Ä	Ô	ä	ô
5	Ľ	ľ	Ĺ	Ő	Í	ő
6	Ś	ś	Ć	Ö	ć	ö
7	§	ˇ	Ç	×	ç	÷
8	¨	¸	Č	Ř	č	ř
9	Š	š	É	Ů	é	ů
A	Ş	ş	Ę	Ú	ę	ú
B	Ť	ť	Ë	Ű	ë	ű
C	Ź	ź	Ě	Ü	ě	ü
D	—	˝	Í	Ý	í	ý
E	Ž	ž	Î	Ţ	î	ţ
F	Ż	ż	Ď	ß	ď	˙

ISO 8859-5 (ロシアのキリール文字)

	A	B	C	D	E	F
0		А	Р	а	р	№
1	Ё	Б	С	б	с	ё
2	Ђ	В	Т	в	т	ђ
3	Ѓ	Г	У	г	у	ѓ
4	Є	Д	Ф	д	ф	є
5	S	Е	Х	е	х	s
6	І	Ж	Ц	ж	ц	і
7	Ї	З	Ч	з	ч	ї
8	J	И	Ш	и	ш	j
9	Љ	Й	Щ	й	щ	љ
A	Њ	К	Ъ	к	ъ	њ
B	Ћ	Л	Ы	л	ы	ћ
C	Ќ	М	Ь	м	ь	ќ
D	—	Н	Э	н	э	§
E	Ў	О	Ю	о	ю	ў
F	Џ	П	Я	п	я	џ

図 5　ISO 8859 のバージョン

3-3 日本の文字コード

(a)　JIS C 6220

日本語の場合、ヨーロッパ諸国語とは異なり、「ひらがな」あるいはカタカナだけでも 50 文字近くあるため、ヨーロッパ各国が ISO 646 の各国語版で対処した方法、すなわち、ASCII の一部の記号だけを入れ替える方法ではすべての「ひらがな」およびすべてのカタカナを扱うことさえもできなかった。そこで、コンピュータ上で日本語を扱うための第一ステップとして、独自の方法でカタカナだけを規定した JIS C 6220 が 1976 年に開発された。この JIS C 6220 ではなるべく広い分野や用途でこの文字コードが使われることを目指して、1 つの規格書の中に以下の 2 種類の文字コードが規定されている。

①ASCII と同様に 7 ビット構造のまま、ASCII のアルファベット部分をすべてカタカナに入れ替えた文字コード……ASCII とほとんど同じ文字・記号を収録したローマ字の文字集合も同時に規定され、「SO(Shift-Out: 0Eh)」と「SI(Shift-In: 0Fh)」という 2 つの制御文字を用いて、ローマ字とカタカナを切り替えながら利用する。

②ASCII を 8 ビット構造に拡張して、前半部分はほぼ ASCII のままとし、拡張した後半部分にすべてのカタカナを収録した文字コード……「SI」と「SO」による切り替えは不要。

JIS C 6220 のビット構造の文字コードは、ヨーロッパ各国で開発された ISO 8859 と基本的に同じ構造をしているので、日本語版 8859 と言うこともできる。

(b)　JIS C 6220 と ASCII の違い

JIS C 6220（JIS X 0201）で規定されているローマ字部分は ASCII とほとんど同じ文字・記号の集合であるが、日本国内での文字の利用頻度を考慮して、2 箇所の変更が施されている。まず、1 箇所目。ASCII では「＼」（バックスラッシュ）が割り振られている 5Ch の位置に JIS C 6220 では「¥」（円通貨記号）が割り振られている。たとえば、Windows 環境において、C ドライブのルートの真下にある「My Documents」フォルダを指図するのに、アメリカ版 Windows では「C:＼My Documents」と入力・表示するが、日本語版 Windows では「C:¥My Documents」と入力・表示する。米国のマイクロソフト社では、Windows においてパスを区切る記号として 5Ch に位置する「＼」記号を使用することにしたところ、日本語版 Windows の 1 バイト文字は（ASCII ではなく）JIS C

6220 に基づいていたため、OS 内部では同じ 5Ch という文字コードで表現される文字が日本語版の Windows 環境では「¥」記号で表示される結果となったものである。日頃、「＼」記号を見かけることはほとんどないが、日本語対応になっていない英語版のソフトウェアをセットアップしようとすると、セットアップ先の表示が「C:＼」となっていて戸惑うことがある。この「C:＼」は「C:¥」とみなして、そのままセットアップ作業を行うことができる。

JIS C 6220 ではもう 1 箇所、ASCII では「‾」（チルダ）が割り振られている 7Eh の箇所も「‾」（オーバーライン）に置き換えられている。DOS/V パソコンの日本語キーボードの該当するキーの刻印が「‾」（オーバーライン）になっているのは、この JIS 規格の規定に準拠しているためである。ただし、日本語版 Windows に添付している欧文フォントでは、この 7Eh に相当する記号は「‾」（オーバーライン）ではなく「‾」（チルダ）のまま収録されている。

後半＼前半	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	SP	0	@	P	`	p				ー	タ	ミ		
1	SOH	DC1	!	1	A	Q	a	q			。	ア	チ	ム		
2	STH	DC2	"	2	B	R	b	r			「	イ	ツ	メ		
3	ETH	DC3	#	3	C	S	c	s			」	ウ	テ	モ		
4	EOT	DC4	$	4	D	T	d	t			、	エ	ト	ヤ		
5	ENQ	NAK	%	5	E	U	e	u			・	オ	ナ	ユ		
6	ACK	SYN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7	BEL	ETB	'	7	G	W	g	w			ァ	キ	ヌ	ラ		
8	BS	CAN	(	8	H	X	h	x			ィ	ク	ネ	リ		
9	HT	EM	)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A	LF	SUB	*	:	J	Z	j	z			ェ	コ	ハ	レ		
B	VT	ESC	+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C	FF	FS	,	<	L	¥	l	|			ャ	シ	フ	ワ		
D	CR	GS	-	=	M	]	m	}			ュ	ス	ヘ	ン		
E	SO	RS	.	>	N	^	n	~			ョ	セ	ホ	゛		
F	SI	US	/	?	O	_	o	DEL			ッ	ソ	マ	゜		

図 6　JIS C 6220

(c)　JIS C 6266（JIS X 0208）

せいぜい数十個のアルファベットを規定すればよい欧米諸言語とは異なり、日本語には小学校で教わる教育用漢字だけでも 1,000 字以上、日常生活に必要な漢字を網羅したとされる常用漢字で 2,000 字弱、そして、高校時代などでお世話になった漢和辞典には 1 万字前後の漢字が掲載されている。そんな膨大な文字数を有する漢字を文字コードに規定するには従来の 7 ビットあるいは 8 ビットを基本とした 1 バイトの文字コードではまったく足らない。そのため世界ではじめて 2 バイトで 1 文字を表すしくみ、すなわち、2 バイト文字コードが開発された。

JIS C 6266 には、9,000 個弱のコード領域に、6,355 字の漢字を含めて、ひらがな・カタカナ・ローマ字・ギリシャ文字・記号など、全部で 6,869 字の文字・記号が規定されている。

さて、2 バイトを単純に 8 ビット X 2 = 16 ビットと考えると、2 の 16 乗となり、65,536 種類もの文字を収録できる計算となるが、JIS X 0208 では 9,000 個弱のコード領域を用意して、その中に文字を割り振る構造が採られている。わざわざ収録できる文字数を減らす構造が採用されているのは少し不思議に感じられるが、これはその前に開発された JIS C 6220 で 8 ビットをフルに使った文字コードのほかに、8 ビットの内の 7 ビットだけを用いた 7 ビット構造の文字コードが規定されていたことを踏まえたものである。

JIS C 6220 が規定している 2 つの文字コードの内、8 ビット構造の文字コードだけを前提として、8 ビットを 2 つ組み合わせれば、2 バイト文字コードとしてより広いコード領域を確保することができるが、それでは JIS C 6220 の 7 ビット構造の 1 バイト文字コードとの共存が困難になる。

そこで、既存の 1 バイト文字コードとの共存性を考慮して、JIS C 6220 の 7 ビット構造を基本としたわけである。

具体的には、JIS C 6220 の 7 ビット構造の文字コード領域の中で特殊な機能を持っている制御文字 34 個の領域を除いた 94 個（= 128－34）のコード領域（21h～7Eh の領域）を 2 つ組み合わせることによって、94 X 94 = 8,836 個のコード領域を確保し、そのコード領域内に漢字や「ひらがな」などの文字を割り振る構造が採用された。

文字数	文字の種類	文字の例
137	各種記号	、。・：；〃々〇+＜♂℃＄％ §☆◆△※〒→＝⊇∀≡√♪
62	数字および大小ローマ字 （アルファベット）	1234789ABCDVWXYZ abcdevwxyz
83	ひらがな	あいろわがざぱいっゅゐゑん
86	カタカナ	アイロワガザパィッュヰヱヴ
48	大小ギリシャ文字	Α Β Γ Ε Ζ Η α β γ δ ε ζ
66	大小キリール文字	А Б В Г Д Е а б в г д е
32	罫線素片	｜ ┬ ┌ ┼ ┴
2,965	第一水準文字	亜唖娃阿哀愛･･･藁蕨椀湾碗腕
3,390	第二水準文字	弌丐丕个丱丶･･･堯槇遙瑤凜熙

図 7　JIS X208-1997

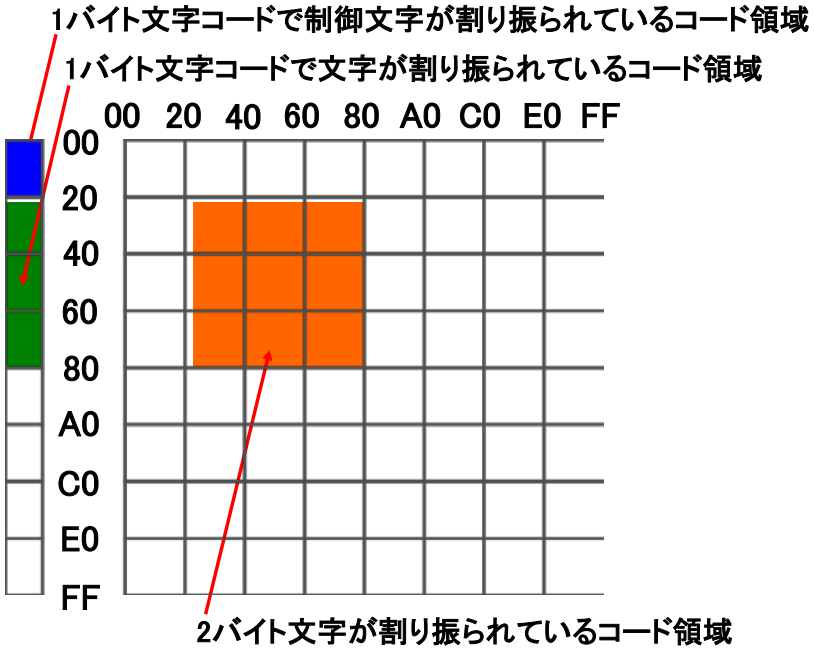


図 8　JIS 漢字コードの構造

(d)　1 バイト文字コードと 2 バイト文字コードを共存させるしくみ

JIS C 6266（JIS X 0208）の 2 バイト文字コードの各バイトの値は JIS C 6220（JIS X 0201）で規定されている 1 バイト文字コードの値の範囲と重複しているため、文字コードだけを単純に羅列させると、各バイトが 1 バイト文字を表すのか、それとも、2 バイト文字の一部を表すのか、

区別することができない。たとえば、JIS C 6266（JIS X 0208）では「漢」は 34h・41h という 2 バイトで表されるが、この 34h・41h というバイト列を 1 バイト文字とみなすと、それぞれ「4」と「A」を表す文字になる。

そこで、データ文字列の途中に、制御文字「ESC」とそれぞれの文字コードを表す規定の文字列を挿入することによって、それ以降の文字コードが 1 バイトのアルファベットやカタカナ、あるいは 2 バイトの漢字やひらがなに切り替わる方法が採用された。この制御文字をエスケープシーケンスと呼び、JIS X 0202 として規格化されている。

文字コードの種類	エスケープ・シーケンス	
	16進数	16進数を英字で表現すると
JISローマ字	1Bh 28h 4Ah	<esc> (J
ASCII	1Bh 28h 42h	<esc> (B
半角カタカナ	1Bh 28h 49h	<esc> (I
JIS C 6226-1978	1Bh 24h 40h	<esc> $ @
JIS X 0208-1983	1Bh 24h 42h	<esc> $ B
JIS X 0208-1990	1Bh 26h 40h 1Bh 24h 42h	<esc>& @ <ESC> $ B
JIS X 0208-1997		
JIS X 0212-1990	1Bh 24h 28h 44h	<esc> $ (D

図 9　文字コードの切り替え

（e）　パソコンの内部処理用文字コードとしてのシフト JIS

JIS C 6266（JIS X 0208）では 2 バイト文字の各バイトが、1 バイト文字の英字などを表すコードと重複していて、バイトの並びを見ただけでは 1 バイト文字なのか、それとも、2 バイト文字の一部なのかを区別することができないため、エスケープシーケンスで 1 バイト文字コードと 2 バイト文字コードを切り換える方法が開発された。しかし、これをそのまま OS の内部処理用の文字コードとして採用すると、2 バイト文字の削除・検索・行端での改行などを行う際に、絶えず前方にあるエスケープシーケンスにさかのぼって、現在のバイト並びが 1 バイト文字なのか、それとも、2 バイト文字の一部なのかというモードを把握していないといけないので、ソフトウェアの開発が面倒になる。JIS 漢字コードがその本来の目的としているデータ交換用途においては、データの先頭から順番に送受信するのが基本的な使い方なので、エスケープシーケンスによる切り替えは有効な処理方法であったが、ランダムにデータをアクセスすることの多い OS 内部の処理方法としてはあまり適しているとはいえないのである。

（f）　シフト JIS コードのしくみ

MS-DOS の日本語化作業を行っていたアスキーマイクロソフト社では、MS-DOS の内部処理用コードとして、ひらがなや漢字などの 2 バイト文字の先頭バイトを 1 バイト文字と重複しないコード領域に巧妙にシフトさせた方法を開発した。これがシフト JIS コードである。

このシフト JIS コードでは、モードの切り替えを意識することなく、バイト並びを見ただけでそのバイトが 1 バイト文字を表すのか、2 バイト文字の先頭バイトを表すのかがわかるため、当時の非力なパソコンでも効率的に日本語を扱えるようになった。その後、シフト JIS コードは CP/M や Macintosh 用の OS である漢字 TALK（後の Mac OS）に採用されたのに続いて、Windows にも採用されて、パソコンにおける日本語文字コードのデファクトスタンダードとなった。最近では、インターネット対応の携帯電話として爆発的にヒットしている NTT ドコモの iMODE も、内部処理用の文字コードとしてシフト JIS コードを採用している。

ただし、シフト JIS コードはもともとアスキーマイクロソフト社という私企業によって開発されたものであり、JIS 漢字コードのような公的な規格ではない。そのため、同じシフト JIS コードと言っても、Windows と Mac OS とではあらかじめ OS が外字として収録している“システム外

字”の数やコード領域が異なっていて、情報交換用文字コードとしては問題点を抱えていた。そこで、1997 年の JIS X 0208 改定の際にデファクトスタンダードを追認する形で、JIS 漢字コードを実際にコンピュータ等に実装する方法の一つとして“シフト符号化表現”が規定された。

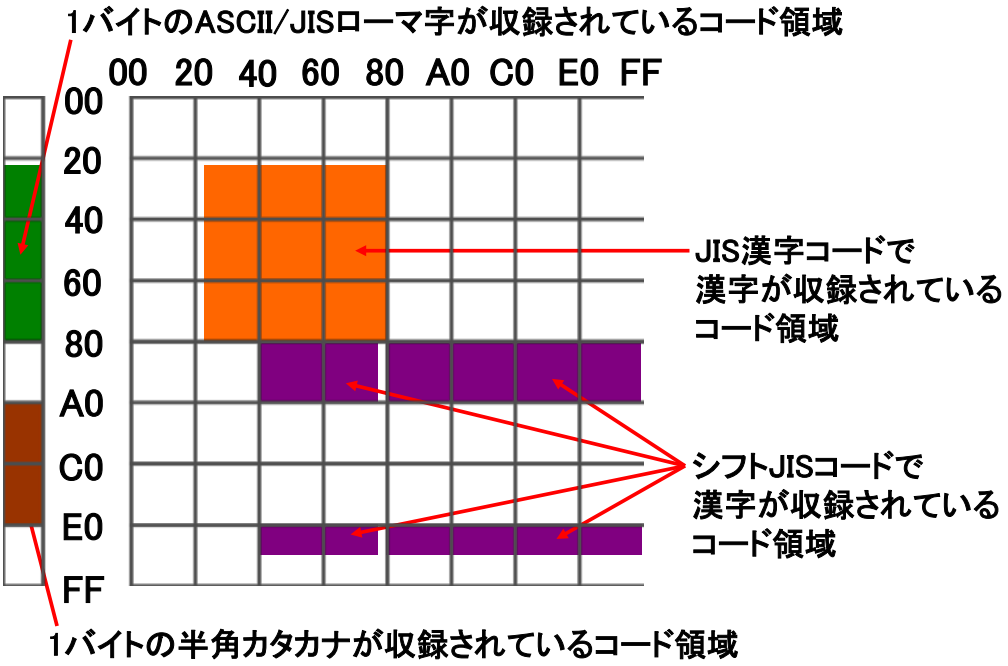


図 10　シフト JIS の構造

3-4 国際化文字コード

(a)　ISO 10646 と Unicode

ISO 2022 によって同時に 4 つまでの文字コードを切り換えて利用できるようになったが、1980 年代の半ばになると、コンピュータでより多くの言語をより手軽に扱えるしくみが必要になってきた。そこで、世界中のローカルな文字コードを 1 つにまとめて同時に利用できる、新たな文字コード ISO 10646 の検討が始められた。一方、ほぼ同じ頃、アメリカではアップルコンピュータや IBM、マイクロソフトなどのコンピュータ関連のメーカが中心となって、16 ビットの単一バイトで世界中の文字を表現することを目指した文字コード= Unicode を開発する動きが出てきた。

ISO 10646 と Unicode は当初、別々に開発が進められ、その内容も大きく異なっていたが、目的を同じにする 2 種類の異なる標準ができあがることに危惧を覚えた双方の関係者が協議して、両規格は統一されることとなり、1993 年には統一した内容に基づく ISO 10646-1:1993 と Unicode 1.1 が発行された。

以来、ISO/IEC と Unicode Consortium では共同して規格内容の改定作業にあたっている。ISO は世界各国の工業標準を定める団体（日本からは JISC）が国を代表して集まって、ISO 10646 = Unicode に収録する文字や理論的な枠組みなどを中心に検討しているのに対して、Unicode Consortium のほうはコンピュータやソフトウェアを開発している企業の技術者が集まって、その文字コードを実装する方法を中心に検討・開発するという、おおまかな作業分担になっているようである。

10646 は本来 32 ビットで 1 個の文字を表すのに対して、Unicode では 16 ビットで 1 個の文字を表すという基本的な設計思想の違いがある。そのため、10646 の 16 ビットを超えるコード領域に収録されている文字は Unicode では本来、利用することができないという制約があった。ただし、10646-1 と Unicode 1.1 で規定されている文字集合は完全に同じで、さらに、個々の文字に割り振られている文字コードもすべて同じである。また、今日までのところ、16 ビットで符号化できるコード領域にしか、文字が割り振られていないので、現実的には「10646-1 = Unicode 1.1」とみなされている。

年	おもなできごと
1984	ISO/IECに10646のワーキング・グループが設立される
1989	Unicodeとは別内容の10646の最初の案(DP)が発表される
1990	ISO総会で上記案が否決される
1991	Unicode 1.0が発行される
	ISOとUnicode Consortiumが10646とUnicodeの統一に合意する
1993	ISO 10646の公式規格書『10646-1:1993』が発行される
	10646-1と内容を一致させたUnicode 1.1が発行される
1996	Unicode 2.0が発行される
2000	Unicode 3.0が発行される
	ISO 10646-1の改訂版『10646-1:2000』が発行される

図 11　ISO 646 およびユニコードの開発歴史

	ISO 10646	Unicode
文字コードの性格	ISOによる 公的な標準	Unicode Consortium による私的な標準
実装されている システム	なし	Windows 98/2000 Mac OS (8.5以降)など
1個の文字を表す ビット数	32ビット	16ビット
文字集合	10646-1:1993=Unicode 1.1で、 完全に同じ文字集合	
個々の文字の 文字コード	10646-1:1993=Unicode 1.1で、 完全に同じ文字コード	

図 12　ISO 646 とユニコードとの違い

(b)　ISO 10646

Unicode が本来、16 ビットの単一バイトで世界中の文字を表現することを目指した文字コードであったのに対して、ISO 10646 のほうは４オクテット、すなわち、32 ビットで 1 文字を表すしくみが基本となっている。この基本的なしくみを”UCS (Universal Multiple-Octet Coded Character Set)”と呼んでいる。

個々の文字に固有の文字コードを割り振るために、4 オクテット =32 ビットで表現可能な莫大なコード領域を持っているので、ISO 10646 では最大で 21.5 億もの文字を収録することができるようになっている。これだけのコード領域が用意されていれば、その総数が 10 万字を超えるとも言われている漢字を含めて、世界中のすべての文字を収録することはもとより、例えば、古代中国の象形文字やマヤ文字などの今はすでに廃れてしまった歴史上の文字を収録することさえ余裕でできる。

(c)　10646 の構造

JIS 漢字コードが区・点の 2 次元で構成されているように、ISO 10646 の全体構造は UCS の 32 ビットのコード領域が群（Group）・面（Plane）・区（Row）・点（Cell）の 4 次元で構成されている。群 0～群 127 の 1 つの群はそれぞれが面 0～面 255 の 256 面から構成されていて、さらに、1 つの面は 256 区 X256 点のマトリックスで構成されているという 4 次元の構造と考えられる。

この膨大なコード領域の先頭の群（群 0）の中にさらに先頭の面（面 0）をとくに”基本多言語面（BMP: Basic Multilingual Plane）”と呼んでいる。BMP には、英語・フランス語・ドイツ語など、欧米の主要各国の言語で使われているラテン文字がすべて収録されているほかに、

- ギリシャ文字
- ロシアなどで使われているキリール文字
- アラビア文字
- 日本・中国・台湾・韓国などで使われている漢字
- 日本で使われているひらがなやカタカナ
- 韓国で使われているハングル
- 矢印や数学用記号などのさまざまな記号

といった、比較的、使用人口の多い文字が網羅されている。ちなみに、1993 年に発行された ISO 10646-1 の規格書では、その正式名称が示す通り、UCS が群・面・区・点の 4 次元で構成されているという基本的な構造と先頭の BMP に収録される文字だけが規定されていて、BMP 以外の群および面に収録される文字は一文字も規定されていない。

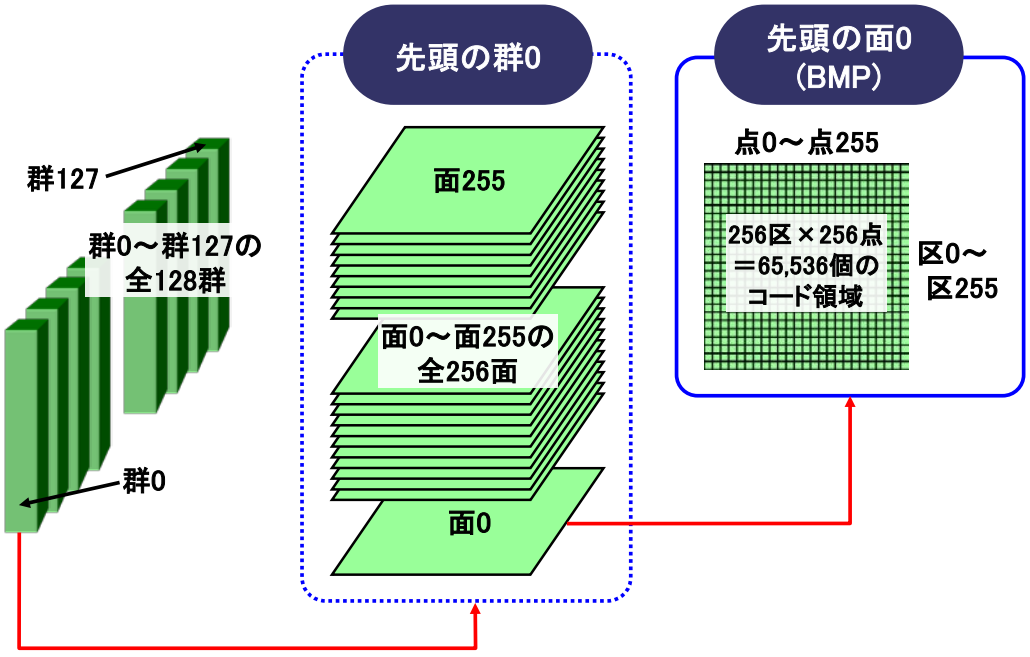


図 13　ISO 10646 の群・面・区・点の 4 次元構造

(d)　10646 の符号化方法

ISO 10646 は 4 オクテット、すなわち、32 ビットで 1 文字を表すしくみが基本となっているが、実際に文字を符号化する方法としては”UCS-4”と”UCS-2”の 2 種類が規定されている。前者の UCS-4 は 32 ビットで 1 文字を表す 10646 本来の符号化方法であるのに対して、後者の UCS-2 は Unicode との整合性を保つために 16 ビットで 1 文字を表す符号化方法である。UCS-4 で文字列を符号化すると、現時点では BMP 以外の文字がまったく規定されていないにもかかわらず、すべての文字に群 0・面 0 を意味する「0000」を付けて符号化しなければならないため、無駄が多くなる。一方の UCS-2 ではこの余分な「0000」を付加する必要がない分、BMP に収録されている文字をより効率よく符号化できるというメリットがある。

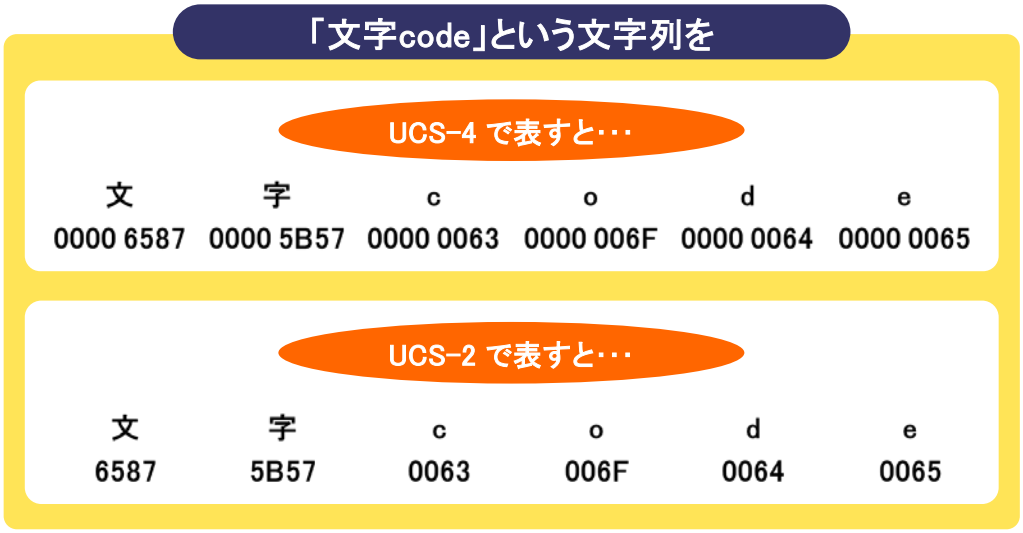


図 14　文字列を UCS4 と UCS2 で符号化

3-5 Unicode

(a)　Unicode の 10 の基本方針

Unicode の規格書には Unicode を開発するうえで基準とした 10 の基本方針が掲げられている。この 10 の基本方針については、ISO 10646 の（廃案となった）最初の案が既存のローカルな文字コードをかき集めた、いかにも国際機関らしい折衷的な仕様であったのに比べて、より高い理想を掲げて、究極の文字コードを開発しようとした意気込みはじゅうぶんに感じられる。しかし、実際のところは、UTF-8 形式で符号化する場合や BMP 以外の面に収録される文字を符号化する場合には固定長にはならない（後述）など、必ずしも、この方針通りに開発されているとは言えない。

また、”Unification（統合）”の方針に対しては、日本・中国・台湾・韓国の間で微妙に字形や意味が異なる漢字を 1 つにまとめていることに対して、一部文芸家や評論家などから「漢字文化の衰退」あるいは「黒船の来襲」といった表現で問題視する発言が相次いだため、一般マスコミにおいても大きく取り上げられて話題となった。

基本方針	説　　明
1. Sixteen-bit character codes	テキストの表示・編集・検索・ソートなどが容易にできるように、16ビット固定長の文字コードであること
2. Efficiency	エスケープ・シーケンスやシフトなどの方法を用いることなく、効率よく符号化できること
3. Character, not glyphs	「a」と「a」を区別せずに同じコードを割り振るように、字形ではなく、文字に対してコードを割り振ること
4. Semantics	明確に文字を識別できること
5. Plain text	フォント・サイズやフォント色などの情報を持たないプレイン・テキスト用の文字コードであること
6. Logical order	メモリ上での文字の並びは理論的な順番であること
7. Unification	日本・中国・台湾・韓国などで使われている漢字のように、言語間で重複している文字は統合すること
8. Dynamic composition	アクセント付きの文字の動的な統合を許すこと
9. Equivalent sequence	結合済み文字に対しては、それと同一文字を動的に結合する形式で表す場合の文字の並びを示すこと
10. Convertibility	広く利用されている既存の文字コードとの間で変換できることを保証すること

図 15　ユニコードの 10 の基本方針

(b)　Unicode の符号化方法

Unicode は 16 ビットで 1 文字を表すことを基本とした文字コードであるが、実際にこの Unicode をパソコン上などに実装するにあたっては、インターネットメールで標準的に採用されている 7 ビット通信などの既存の 7 ビットあるいは 8 ビットのシステムと整合性を保つため、さまざまな符号化方法が開発されてきた。

①UTF-16

Unicode 本来の 16 ビットの符号化方法であるが、ISO 10646-1: 1993 で規定された UCS-2 とは異なり、BMP に収録されている文字だけでなく、BMP に続く 16 面（面 1～面 16）に収録されている文字も表現できるように工夫されている。 具体的には、BMP に続く 16 面を符号化するために、BMP 中の D800h～DBEFh と DC00h～DFFFh の 2 箇所のコード領域を用意しておき、それぞれの 2 バイトの値を所定の数式で計算することによって、BMP に続く 16 面に収録されている文字を符号化する”Surrogate”というしくみが導入された。BMP の中に Surrogate 用に予約されている 2 箇所のコード領域はそれぞれ 1,024 個の値を持っているので、この 2 箇所を組み合わせる(1,024 X 1,024 = 1,048,576)ことによって、BMP に続く 16 面に収録されているすべての文字（256 区 X256 点 X16 面 = 1,048,576 文字）を符号化するというきわめて巧妙なしくみである。 ただし、2001 年 3 月現在、BMP 以外の面にフォントを実装したシステムは実在しないので、現状では Surrogate というしくみも規格上で決まっているだけで、この Surrogate を実装しているシステムは存在しない。

②UTF-7

当初、インターネットメールで Unicode 文字を利用するための符号化方法として提案されたが、その後、より効率のよい UTF-8 が提案されて、現在は UTF-8 を標準とする方向に向かっている。

③UTF-8

16 ビット固定長の Unicode を 8 ビット単位の可変長の文字コードに変換する符号化方法で、今日、インターネットメールで Unicode 文字を扱う際の標準的な符号化方法とみなされている。 この UTF-8 では文字種によって 1 文字を表すのに最低 1 オクテット～最大 6 オクテットの可変長となり、英語やドイツ語などのラテン文字は従来通りの 1 オクテット = 1 バイトで 1 文字を転送できる一方で、漢字やハングルなどは 1 文字を転送するのに 3 オクテット = 3 バイト必要となるため、今まで 2 バイトで 1 文字を転送できていた日本や中国などの漢字文化圏では従来よりも効率が悪くなるという欠点がある。

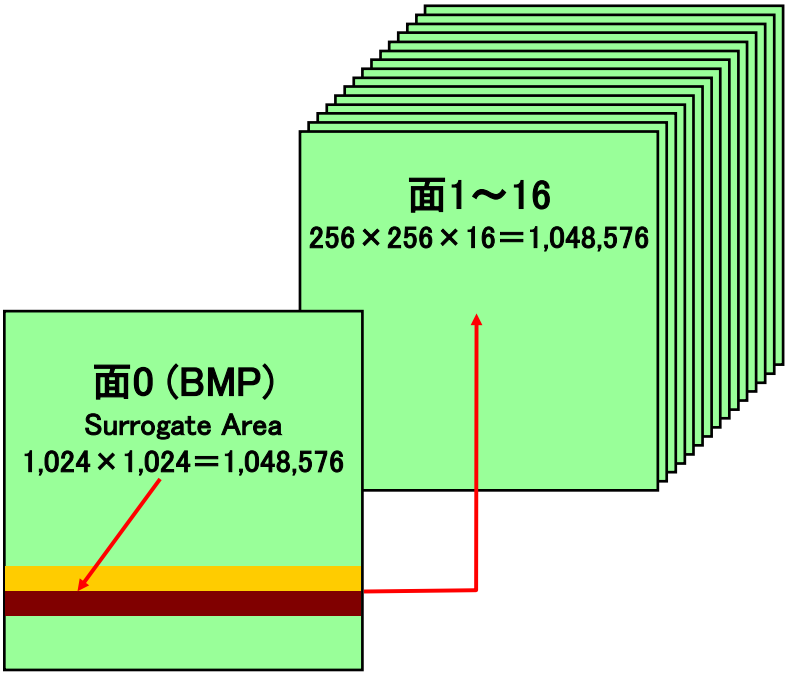


図 16　BMP 以外の面の文字を符号化するしくみ

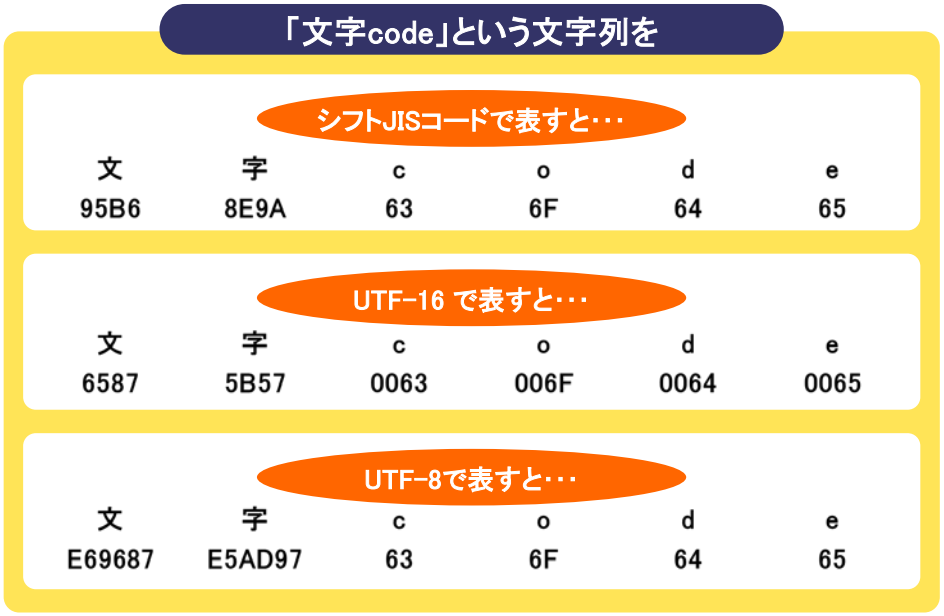


図 17　文字列を UTF16 と UTF8 で符号化

3-6 国際規格に収録されている漢字

(a)　10646/Unicode に収録されている漢字

10646/Unicode に収録する漢字を決めるにあたって、まず、日本・中国・台湾・韓国など、漢字を用いている国々の既存の文字コードが集められ、その延べ 11 万を超えるすべての漢字に対して字形が同じ漢字を 1 つにまとめる Unification（統合）という作業が行われた結果、20,902 字の漢字が”CJK 統合漢字（CJK Unified Ideographs）”として収録された。

日本からは JIS 漢字（JIS X 0208）と補助漢字（JIS X 0212）の 2 つの文字コードに規定されていた漢字がすべて収録されているほか、以下のように、やはり漢字を使用している中国・台湾・韓国の各国の国内規定で規定されていた漢字も幅広く収録されている。

その後、BMP 内の A 領域にあったハングルがすべて（保留領域とされていた）0 領域に移動した後のコード領域に 6,582 字の漢字が”CJK 統合漢字　拡張 A”として追加されており、さらに、BMP 以降の面 2 にも多くの漢字が追加される方向で検討されているが、2001 年 3 月現在、Word98/2000 や一太郎 9/10/11 などの Unicode 対応アプリケーションで実際に利用できる漢字は Unicode のバージョン 2.1 に規定されている 20,902 字がすべてである。

(b)　漢字の統合

日本・中国・台湾・韓国などの既存文字コードに収録されている漢字を集めて、字形が同じ漢字を統合して 1 つだけの文字コードを割り振る CJK 統合漢字は、前述した通り、Unicode 開発時の 10 の基本方針の一つで、Unicode の大きな特徴となっているが、同時に、とくに日本国内において「漢字を理解しないアメリカ人による横暴」といった論調で Unicode が批判される最大の理由ともなった。

日本・中国・台湾・韓国の漢字は起源を同じくするとはいえ、各国での千年以上にわたる歴史をふまえて、「平」や「骨」のように微妙に字形が異なっている文字がある。また、同じ字形の漢字でも、各国でその意味が異なる場合もある。たとえば、「机」という漢字は、日本と台湾では「つくえ」を表すが、中国では「機」の簡略字で、韓国の漢字コードには相当する字形の漢字は規定されていない。

このような微妙な字形や意味の違いを無視してまで強引に 1 つに統合してしまっている点が一部の文芸家などに反感を呼んだが、現実的には、日本語 Windows98/Me に添付している「MS 明朝」などの Unicode 対応の日本語フォントにおいては「平」や「骨」などの漢字には日本の JIS 漢字コード掲載の字体が採用されているので、大きな問題とはなっていない。これは、各国で字形が異なる漢字については、フォントに実装する際に各国の字形を優先させることが容認されているためである。

典拠規格	国	漢字数
JIS X 0208-1990(JIS漢字)	日本	6,355
JIS X 0212-1990(補助漢字)		5,801
GB 2312-80	中国	6,763
GB/T 12345-90		2,180
GB 7589-87		4,835
GB 7590-87		2,842
現代漢語通用字表		41
GB 8565.2-88		290
GB 12052-89		94
PRC Telegraph Code		8,000
Big Five (Big 5)	台湾	13,053
CCCII, Level 1		4,808
CNS 11643-1986 Planes 1		5,401
CNS 11643-1986 Planes 2		7,650
CNS 11643-1986 Planes 14		4,198
Taiwan Telegraph Code		9,040
KS X 1001:1992	韓国	4,620
KS X 1002:1991		2,856
ANSI Z39.64-1989 (EACC)	アメリカ	13,481
Xerox Chinese		9,776
TCVN 6056:1995	ベトナム	3,311

図 18　ISO 10646/ユニコード漢字の典拠規格

用語・略語

BCD	Binary Coded Decimal
ASCII	American Standard Code for Information Interchange
JIS	Japan Industrial Standards：日本工業規格
EBCDIC	Extended Binary Coded Decimal Interchange Code
BMP	Bit MaP
TIFF	Tag Image File Format
JPEG	Joint Photographic Experts Group
GIF	Graphics Interchange Format
PICT	Macintosh PICT format
MPEG	Motion Picture Experts Group
BCD	Binary Coded Decimal：2 進化 10 進数
CRC	Cyclic Redundancy Check：巡回冗長検査
CPU	Central Processing Unit
ALU	Arithmetic and Logic Unit
クロックパルス	動作を合わせるために用いる周期的なパルス
PSW	Program Status Word
NMI	Non Maskable Interrupt
IRQ	Interrupt ReQuest
RAM	Random Access Memory
SRAM	Static RAM
DRAM	Dynamic RAM
ROM	Read Only Memory
マスク ROM	Masked Read Only Memory
PROM	Programmable Read Only Memory
EPROM	Erasable-Programmable Read Only Memory
フラッシュメモリ	ディジタルカメラのメモリカード。携帯情報端末のメモリ素子などを中心に急速に普及が進んでいる
MO	Magneto Optical disk：光磁気ディスク
CD	Compact Disc
CD-R	CD-Recordable：追記型光ディスク
CD-ROM	CD-Read Only Memory
CD-RW	CD-ReWritable
DVD	Digital Versatile Disk
DVD-ROM	DVD-Read Only Memory
DVD-R	DVD-Recordable
DVD-RAM	DVD-Random Access Memory
RAID	Redundant Array of Inexpensive Disks
ECC	Error Correcting Code：誤り訂正符号
OCR	Optical Character Reader：光学式文字読取り装置
OMR	Optical Mark Reader：光学式記号読取り装置
CCD	Charge Coupled Device：電荷結合素子
GUI	Graphical User Interface
LCD	Liquid Crystal Display
GP-IB	General Purpose Interface Bus
IEEE	Institute of Electrical and Electronics Engineers：米国電気電子技術者協会
デイジーチェーン	装置から装置へ次々に、いもづる式に接続すること
SCSI	Small Computer System Interface：スカジー
DTE	Data Terminal Equipment
DCE	Data Circuit terminating Equipment：回線終端装置
EIA	Electronic Industries Association：米国電子工業会
IDE	Integrated Device Electronics

USB	Universal Serial Bus
Ir-DA	Infrared Data Association
OS	Operating System：Windows や UNIX などハードウェアの管理やユーザインタフェースの提供を行うソフトウェア
CASE	Computer Aided Software Engineering
CAD	Computer Aided Design：コンピュータ支援設計
CAM	Computer Aided Manufacturing：コンピュータ支援製造
CAI	Computer Aided Instruction：コンピュータ支援教育
CMI	Computer Management Instruction：コンピュータ支援教育管理
JCL	Job Control Language
クライアント/サーバ	厳密にはサービスを提供するプログラムがサーバで、サービスを要求するプログラムをクライアントと呼んだほうが正確であるが、現在では同義として扱われる場合が多い
スプーリング	spooling：アプリケーションを実行しながら、周辺機器とのデータのやり取りをする
データベースサーバ	セキュリティ上、検索のみに限定して使われる場合もある
GUI	Graphical User Interface
メインフレーム	汎用コンピュータのこと
DSU	Digital Service Unit
RPC	Remote Procedure Call：遠隔手続き呼出しともいう
シームレス	透過的
LPC	Local Procedure Call
ANSI	American National Standards Institute
ISO	International Organization for Standardization：国際標準化機構 国際的な規格の制定を行う国際機関
23h	“h”は hexa の略で、16 進数を表す
JIS C 6220	JIS C 6220 はその後、JIS 全体のグループ変えに伴い、JIS X 0201 という規格書番号に変更されている。JIS C 6220 ・JIS X 0201 では、そこで規定している 2 種類の文字コードを”7 単位符号”および”8 単位符号”と呼んでいる
JIS C 6226	JIS C 6226 はその後、JIS 全体のグループ変えに伴い、JIS X 0208 という規格書番号に変更されている
システム外字	ユーザが定義する一般的な外字に対して、OS があらかじめ用意している外字。機種依存文字とも呼ぶ
現地化（ローカライズ）	コンピュータの世界では、主にアメリカで開発された技術や規格などを自国向けに変更することを”L10N”と、また、個別のローカライズではなく、国際的に適用する汎用のしくみとして進化させることを”I18N”と呼ぶことがある
ISO 10646	国際化時代の新たなるスタンダードとなることを象徴した規格番号といえる
オクテット	パソコンの世界では 8 ビットを 1 つのまとまりとした単位を”バイト(Byte)”と呼ぶのが通例だが、このバイトは 8 ビット以外の値を定義する際にも使われるため、厳密に 8 ビットを単位として表したいときには、”オクテット(Octet)”という言葉を使う
ISO 10646-1	正式名称は、”ISO/IEC 10646-1: 1993 Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane”
CJK 漢字統合	その後、ベトナムで使われていた漢字の文字集合も対象となったので、現在では”CJKV 統合漢字”と呼ばれることもある

引用文献：「基本情報処理技術者 標準教科書」 オーム社
引用文献：「図解でわかる文字コードのすべて」 日本実業出版社