ประวัติ Character Set

นำเสนอ

ดร. ครรชิต มาลัยวงศ์

สารบัญ

1 ประวัติของ Character Set

Character set เป็นแนวคิดที่เริ่มมาจากการการใช้รหัสมอส (Morse Code) ในการทดลองส่งข่าวสารระหว่างกันเมื่อปี ค.ศ. 1838 ซึ่งมีลักษณะที่คล้ายการส่งข้อมูลในระบบคอมพิวเตอร์สมัยใหม่ ซึ่งรหัสมอส ประกอบไปด้วย จุด (Dot) และ ขีด (Dash) แต่ยังคงแตกต่างจากรหัสตัวอักษรที่ระบบคอมพิวเตอร์สมัยใหม่โดยจะมีการนำค่าทั้ง dot และ dash มาผสมผสานเพื่อสร้างตัวอักษรขึ้นมาใหม่เพื่อใช้แทนรหัสมอสที่มีความยาวที่ต่างกันในหลายลักษณะ โดยหลักการของรหัสมอสคือใช้รูปแบบของตัวอักษรที่ใช้บ่อยสุดมาลดรูปแบบให้สั้นที่สุด ซึ่งใช้ลดความยาวของข้อความ ยกตัวอย่างเช่น ตัวอักษรภาษาอังกฤษที่ใช้บ่อยสุดคือ “E” ถูกแทนด้วย dot และตัวอักษรที่ใช้บ่อยรองลงมาคือ “T” ถูกแทนด้วย dash การที่ Morse ค้นพบรูปแบบความถี่ของตัวอักษรไม่ได้มาจากการศึกษา text แต่ได้มาจากการเฝ้าสังเกตุการนับการพิมพ์ในแต่ละครั้งของเครื่องพิมพ์กล่อง ผลที่ได้คือรหัสที่มีประสิทธิภาพ ซึ่งยังคงมีการนำมาประยุกต์ใช้จนทุกวันนี้ รหัสมอสได้พัฒนาไปหลายเวอร์ชันในแต่ละช่วงเวลา โดยเริ่มแรกถูกพัฒนาเป็น American Morse Code จากนั้นถูกพัฒนาเป็น International Morse Code

หลังจากมีการนำรหัสมอสไปใช้กันอย่างแพร่หลาย มีการนำไปใช้ในการพัฒนาเครื่องส่งโทรสารเพื่อส่งสัญญาณให้ไกลกว่าเดิมด้วยการใช้ไฟฟ้า นอกจากนี้ยังทำการพัฒนาให้เกิดประโยชน์มากขึ้น เช่น สามารถสื่อสารกันแบบ 2 ทางในเวลาเดียวกันทิศทางเดียวกัน (Diplexing) สามารถสื่อสารแบบสี่ข้อความ (Quadiplexing) ยิ่งไปกว่านั้นยังมีการพัฒนาช่องทางการสื่อสารให้สามารถส่งสัญญาณได้มากขึ้น ก้าวต่อมาของการพัฒนาเครื่องส่งโทรสารคือ เครื่องส่งโทรเลขการพิมพ์ ซึ่งรวมไปถึงการสร้างรหัสตัวอักษรใหม่ๆ ขึ้นคือ 5-bit Baudot Code ซึ่งเป็นรหัสตัวอักษรไบนารี่แรกของโลกที่ใช้ประมวลข้อมูลในรูปแบบของตัวอักษร แต่การใช้ 5-Bit Code นั้นมีเนื้อที่สำหรับเก็บส่วนประกอบแค่ 32 ส่วน คือ (2^5 = 32 ตำแหน่ง) ไม่พอที่จะรองรับพยัญชนะภาษาลาติน (Latin Alphabet) รวมทั้งตัวเลขอารบิก (Arabic Numeral) และเครื่องหมายวรรคตอน (Punctuation mark) จึงใช้วิธีการ Locking Shift Scheme มาแก้ปัญหาโดยเปลี่ยนแนวระนาบของ 32 ส่วนประกอบ ภายหลังมีการพัฒนา Baudotให้ม่านประกอบถึง 55 ส่วน ซึ่งมีสถาบัน CCITT (Consultative Committee on International Telephone and Telegraph) เป็นผู้กำหนดมาตรฐาน

ในปี ค.ศ. 1890 เมื่อประเทศสหรัฐอเมริกาได้มีการสำรวจสำมะโนครัสประชากร โดยมีการบันทึกผลด้วย Punch Card ใช้ Hollerith Code ซึ่งเป็นรหัสตัวอักษรสำหรับข้อมูล Alphanumeric ในการเข้ารหัสใน Punch Card ประกอบด้วย 12 แถว และ80 คอลัมน์ ในคอลัมน์แต่ละอันจะแสดงด้วยตัวอักษร หรือสัญลักษณ์ ถูกอ่านและแปลความหมายด้วย Tabulating Machine ในระบบรหัสของ Hollerith ตัวอักษร Alpahnumeric 1 ชุดจะถูกบันทึกใน 12 แถวของ Punch Card ซึ่งมีลักษณะเหมือนรหัสตัวอักษร 12-bit แต่รหัสตัวอักษร 12-bit จะมี 4096 elements แต่ Hollerith code มีเพียง 69 elements อย่างไรก็ตามสำหรับข้อมูลที่สนใจขณะนั้น รหัส Hollerith ก็ยังเหมาะสมกับข้อมูลที่ใช้บน Punch Card ซึ่งในต่อมามีการนำ

Punch Card และ Tabulating Machine มาใช้ในการคำนวณทางธุรกิจ และถูกผสมผสานเข้ากับเครื่องคอมพิวเตอร์ทำให้มีการใช้กันอย่างแพร่หลาย

2 ความสำคัญของ Character Set

ผลจากการพัฒนา การแผ่ขยายของการคมนาคม และเทคโนโลยีการประมวลผลข้อมูลในประเทศสหรัฐอเมริกาช่วงครึ่งแรกของศตวรรษที่ 20 มีความต้องการที่จะใช้รหัสตัวอักษรที่เป็นมาตรฐานสำหรับแลกเปลี่ยนข้อมูลระหว่างกันเพื่อสนับสนุนกลุ่มตัวอักษรภาษาอังกฤษ สมาคมมาตรฐานของสหรัฐอเมริกา (The American Standard Association : ASA) ซึ่งต่อมาเปลี่ยนชื่อเป็น American National Standard Institute (ANSI) ได้เริ่มทำการศึกษาปัญหานี้ และในปี 1963ได้กำหนดรหัสมาตรฐานแห่งสหรัฐอเมริกาสำหรับแลกเปลี่ยนข้อมูลข่าวสาร (American Standard Code for Information Interchange : ASCII) ซึ่งเป็นรหัส 7-bit ขึ้นมาใช้ และในปี 1968 ได้กำหนดมาตรฐานของ ASCII ของ 32 Control characters และ 96 Printing characters ต่อมาได้ถูกขยายเป็น 8-bit Control characters และ 190 Printing characters เนื่องจาก ASCII ถูกใช้เพียงแค่ประเทศเดียว เมื่อต้องการปรับให้ใช้กับ Latin Alphabet ที่ใช้กันมากโดยเฉพาะประเทศทางยุโรปตะวันตก หน่วยงาน International Organization for Standardization (ISO) จึงเขามาดำเนินการในส่วนนี้ โดยกำหนด 10 bit ทางซ้ายสำหรับ National variants โดยจะกำหนดในเวอร์ชันเอกสารอ้างอิงระหว่างประเทศ (IRV)

แม้ว่า ASCII จะถูกใช้ในอุตสาหกรรมทางด้านMicrocomputer, workstations และ Personal Computer ในประเทศสหรัฐอเมริกาอย่างแพร่หลาย แต่ทางด้าน Mainframe computer IBM ได้คิดรหัสตัวอักษร 8-bit ที่เรียกว่า EBCDIC (Extend Binary Coded Decimal Interchange Code) ซึ่งเป็นการพัฒนาต่อมาจาก BCD (Binary Coded Decimal) ที่เป็นรหัส 6-bit มาใช้เฉพาะผลิตภัณฑ์ของทาง IBM เอง ซึ่งทาง IBM ก็ได้ทำการตลาดทางด้าน Mainframe Computer จึงได้จำกัดเฉพาะรหัสตัวอักษร EBCDIC ซึ่งมีการพัฒนาให้ใช้ได้ถึง 57 ประเทศ(อยู่บนพื้นฐานของ ISO 646) ถึงแม้ว่า ISO 646 และ ASCII จะมีความเข้ากันได้ แต่รหัสตัวอักษร EBCDIC ไม่ได้บรรจุรหัสตัวอักษร ASCII ลงไป จึงทำให้การแลกเปลี่ยนของระบบที่ใช้พื้นฐานรหัสตัวอักษร EBCDIC กับ ระบบที่ใช้พื้นฐานของ ASCII และรหัสอื่นไม่มีประสิทธิภาพ ISO จึงมีการสร้างมาตรฐาน ISO 8859-1 (Latin-1) ซึ่งเป็นส่วนขยายของ ASCII และ ISO 646 ใช้ในการแลกเปลี่ยนข้อมูลข่าวสารทางอินเตอร์เนตมากทางยุโรปตะวันตก และยังมีอีกหลายมาตรฐานเพื่อรองรับภาษาลาตินในแต่ละประเทศ

นอกจากนี้ยังมีรหัสตัวอักษรที่รองรับหลายภาษา (Multilanguage Character Set) ที่ถูกใช้แลกเปลี่ยนข่าวสารที่เป็นภาษาประจำชาติของแต่ละประเทศ โดย Xerox และ IBM มีการพัฒนาขึ้นใช้ในผลิตภัณฑ์ของตัวเอง ทั้งสองผลิตภัณฑ์สามารถใช้กับภาษาทางเอเชีย และลาตินสคริปต์ได้หลายภาษา แต่ไม่แพร่หลายเนื่องจากมีราคาสูง จึงทำให้ต้องมีการพัฒนา Unicode ขึ้น เพื่อให้เป็นรหัสตัวอักษรที่ใช้ได้หลายภาษา รองรับการทำงานกับทุกระบบ จึงมีการใช้กันในปัจจุบันอย่างแพร่หลาย

3 มาตรฐานของ Character Set

3.1 ASCII Standard

3.1.1 ASCII – 1963

มาตรฐานแรกของ ASCII คือ ASCII – 1963 ซึ่งเป็นมาตรฐานในปี 1963 โดย ASA (American Standards Association) 2 แถวสุดท้ายในตารางยังไม่ได้ตัดสินว่าจะเป็นอะไร แต่ช่องว่างระหว่าง ACK และ ESC ได้ถูกจองไว้สำหรับเพิ่มสัญลักษณ์อื่นๆ

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOM	EOA	EOM	EOT	WRU	RU	BEL	FEO	HT	LF	VT	FF	CR	SO	SI
020	DC0	DC1	DC2	DC3	DC4	ERR	SYN	LEM	S0	S1	S2	S3	S4	S5	S6	S7
040		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
060	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
120	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	↑	←
140																
160													ACK		ESC	DEL

ตารางที่ 1 ตัวอักษรตามมาตรฐาน ASCII-1963

3.1.2 ASCII – 1965

มาตรฐานที่สองของ ASCII มาจากปี 1965 ปุ่ม control (ยกเว้น DEL) ถูกย้ายจากแถวสุดท้าย บางปุ่มถูกเปลี่ยนชื่อและสองแถวสุดท้ายก็ถูกเติมให้เต็ม ลูกศรและ backslash หายไปและถูกแทนที่ด้วยสัญลักษณ์ตัวอื่น

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SS	ESC	FS	GS	RS	US
040		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
060	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	`	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
120	P	Q	R	S	T	U	V	W	X	Y	Z	[	~	]	^	_
140	@	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
160	p	q	r	s	t	u	v	w	x	y	z	{	¬	}	|	DEL

ตารางที่ 2 ตัวอักษรตามมาตรฐาน ASCII-1965

3.1.3 ASCII – 1967

มาตรฐานสุดท้ายมาจากปี 1967 ซึ่งมาการเปลี่ยนแปลงสัญลักษณ์ @ และสัญลักษณ์บางตัวเพื่อความเหมาะสม และสามารถเห็นสัญลักษณ์ที่ใช้ในยุโรปบนรูปนี้ สัญลักษณ์ vertical bar ถูกแทนที่ด้วยสัญลักษณ์ broken bar ซึ่งทำให้เกิดการโต้แย้งกันเป็นอย่างมากในอเมริกา และได้ถูกตัดสินให้สัญลักษณ์ exclamation สามารถถูกเลือกให้ใช้ได้เหมือนสัญลักษณ์ vertical bar ความแตกต่างนี้จึงได้หมดไป และในหลายๆที่ broken bar ได้ถูกใช้แทน vertical line มาตรฐานนี้ได้ถูกรับเอามาในปี 1967 โดย ISO เหมือนกับเป็นการอ้างอิงถึงมาตรฐานต่างประเทศ

รหัส ASCII ถูกปรับปรุงโดยผู้ผลิตคอมพิวเตอร์ในสหรัฐยกเว้น IBM ซึ่งได้ทำการพัฒนารหัสของตัวเองสำหรับใช้กับเครื่องคอมพิวเตอร์เมนเฟรม เพราะผู้ผลิตคอมพิวเตอร์ในสหรัฐเป็นผู้ผลิตคอมพิวเตอร์รายใหญ่ที่สุดในโลกในขณะนั้น รหัส ASCII จึงได้กลายเป็น รหัสมาตรฐานในระดับนานาชาติอย่างรวดเร็ว

รหัส ASCII ยังได้ถูกใช้เป็นพื้นฐานในการสร้างรหัส 7 – bit สำหรับภาษาที่ไม่ได้สร้างมาจากภาษาลาติน เช่น Arabic และ Greek อีกด้วย

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
040		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
060	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
120	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
140	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
160	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	DEL

ตารางที่ 3ตัวอักษรตามมาตรฐาน ASCII-1967

3.2 EBCDIC Standard

รหัส EBCDIC หรือ Extended Binary Code Decimal Interchange Code ได้รับการพัฒนาขึ้นโดยบริษัท IBM ในช่วงยุคที่ 3 ของคอมพิวเตอร์ (ประมาณปี ค.ศ. 1964-1965) พร้อมกับการเปิดตัวของ IBM System/360 mainframe computer โดยได้ทำการพัฒนาขึ้นมาจากรหัส BCD รหัส EBCDIC มีการเปลี่ยนแปลงจาก รหัส BCD คือมีการเพิ่มการเข้ารหัสตัวอักษรจาก 6 bit เป็น 1 byte มาเป็น 8 bit เป็น 1 byte ซึ่งจะทำให้สามารถสร้างรหัสขึ้นมาได้ถึง 256 รหัส (2^8 = 256) โดยมีการกำหนด bit ออกเป็น 2 ส่วน คือ

- bit แรก (bit ที่ 0 ถึง bit ที่ 3) เรียกว่า Numeric bit โดย 4 bit นี้ จะใส่ค่าลำดับของตัวอักษรตามค่าในเลขฐานสองคือ 8 4 2 1

- bit ถัดมา(bit ที่ 4 ถึง bit ที่ 7)เรียกว่า Zone bit โดย Zone bit นี้ก็จะมีค่าประจำหลักเป็น 8 4 2 1 เช่นกัน

Zone Bit				Numeric Bit			
8	4	2	1	8	4	2	1

อักขระของรหัส EBCDIC นี้จะใช้ตัวเลขฐานสองทั้งหมด 8 ตัว หรือ อีกนัยหนึ่ง คือ สามารถเขียนด้วยเลขฐาน 16 แทนได้ (ใช้ 2 ตัว สำหรับ Zone bit 1 ตัว และ Numeric bit อีก 1 ตัว)

3.2.1 การเข้ารหัสของ EBCDIC

การเข้ารหัสของ EBCDIC จะแบ่งข้อมูลออกเป็น 2 ประเภท คือ ข้อมูลตัวเลข และข้อมูลตัวอักษร ซึ่งมีรูปแบบการเข้ารหัส ดังนี้

ข้อมูลตัวเลข

- **Zone bit**: จะเป็น 1111_2
- **Numeric bit**: จะเป็นค่าของตัวเลขนั้น ๆ ในระบบเลขฐานสอง เช่น

  $3 = 11110011_2$ หรือ $F3_{16}$

  $7 = 11110111_2$ หรือ $F7_{16}$

  $9 = 11111001_2$ หรือ $F9_{16}$

ข้อมูลตัวอักษร

- **Zone bit**:

  ข้อมูล A - I จะเป็น 1100_2 หรือ C_{16}

  ข้อมูล J - R จะเป็น 1101_2 หรือ D_{16}

  ข้อมูล S - Z จะเป็น 1110_2 หรือ E_{16}

  ข้อมูล a - i จะเป็น 1000_2 หรือ 8_{16}

  ข้อมูล j - r จะเป็น 1001_2 หรือ 9_{16}

  ข้อมูล s - z จะเป็น 1010_2 หรือ A_{16}

- **Numeric bit**: จะเป็นค่าของตัวเลขนั้น ๆ ในระบบเลขฐานสอง เช่น

  B = $1100\ 0010_2$ หรือ $C2_{16}$ (B อยู่ในข้อมูลชุดที่ 1 ลำดับที่ 2)

  M = $1101\ 0100_2$ หรือ $D4_{16}$ (M อยู่ในข้อมูลชุดที่ 2 ลำดับที่ 4)

  X = $1110\ 0110_2$ หรือ $E6_{16}$ (X อยู่ในข้อมูลชุดที่ 3 ลำดับที่ 6)

หลังจากที่ IBM ได้ประกาศใช้ EBCDIC ในเครื่อง Mainframe ไปแล้ว standard EBCDIC ก็ได้มีการแตกย่อยเพิ่มขึ้นไปอีก เนื่องจากต้องการสนับสนุนการใช้ภาษาอื่น นอกจากภาษาอังกฤษใน EBCDIC character set โดย EBCDIC standard ทั้งหมดจะแบ่งออกได้เป็น

- Original EBCDIC
- Cyrillic EBCDIC
- Japanese EBCDIC & Revised Japanese EBCDIC
- Augmented EBCDIC
- Revised Augmented EBCDIC
- TIS 620-EBCDIC
- EEC System 4

3.2.2 EBCDIC Table

Dec	Hex	Code	Dec	Hex	Code	Dec	Hex	Code	Dec	Hex	Code
0	00	NUL	32	20		64	40	space	96	60	-
1	01	SOH	33	21		65	41		97	61	/
2	02	STX	34	22		66	42		98	62	
3	03	ETX	35	23		67	43		99	63	
4	04		36	24		68	44		100	64	
5	05	HT	37	25	LF	69	45		101	65	
6	06		38	26	ETB	70	46		102	66	
7	07	DEL	39	27	ESC	71	47		103	67	
8	08		40	28		72	48		104	68	
9	09		41	29		73	49		105	69	
10	0A		42	2A		74	4A	[	106	6A	|
11	0B	VT	43	2B		75	4B	.	107	6B	,
12	0C	FF	44	2C		76	4C	<	108	6C	%
13	0D	CR	45	2D	ENQ	77	4D	(	109	6D	_
14	0E	SO	46	2E	ACK	78	4E	+	110	6E	>
15	0F	SI	47	2F	BEL	79	4F	| !	111	6F	?
16	10	DLE	48	30		80	50	&	112	70	
17	11		49	31		81	51		113	71	
18	12		50	32	SYN	82	52		114	72	
19	13		51	33		83	53		115	73	
20	14		52	34		84	54		116	74	
21	15		53	35		85	55		117	75	

Dec	Hex	Code	Dec	Hex	Code	Dec	Hex	Code	Dec	Hex	Code
22	16	BS	54	36		86	56		118	76	
23	17		55	37	EOT	87	57		119	77	
24	18	CAN	56	38		88	58		120	78	
25	19	EM	57	39		89	59		121	79	‘
26	1A		58	3A		90	5A	!]	122	7A	:
27	1B		59	3B		91	5B	$	123	7B	#
28	1C	IFS	60	3C		92	5C	*	124	7C	@
29	1D	IGS	61	3D	NAK	93	5D	)	125	7D	‘
30	1E	IRS	62	3E		94	5E	;	126	7E	=
31	1F	IUS	63	3F	SUB	95	5F	^	127	7F	"

ตารางที่ 4 ตัวอักษรตามมาตรฐาน EBCDIC

3.2.3 Original EBCDIC

Original EBCDIC เป็นมาตรฐาน EBCDIC เวอร์ชั่นแรกที่ประกาศใช้โดย IBM ในปีค.ศ. 1964 ซึ่งจะมีการกำหนด ส่วนของตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ ตัวพิมพ์เล็ก เครื่องหมายต่าง ๆ ดังตาราง

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	PF	HT	LC	DEL	GE	RLF	SMM	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	TM	RES	NL	BS	IL	CAN	EM	CC	CU1	IFS	IGS	IRS	IUS
040	DS	SOS	FS		BYP	LF	ETB	ESC			SM	CU2		ENQ	ACK	BEL
060			SYN		PN	RS	UC	EOT				CU3	DC4	NAK		SUB
100											¢	.	<	(	+	|
120	&										!	$	*	)	;	¬
140	-	/									¦	,	%	_	>	?
160										‘	:	#	@	'	=	"
200		a	b	c	d	e	f	g	h	i						
220		j	k	l	m	n	o	p	q	r						
240		~	s	t	u	v	w	x	y	z						
260																
300	{	A	B	C	D	E	F	G	H	I			∫		Ψ	
320	}	J	K	L	M	N	O	P	Q	R						
340	\		S	T	U	V	W	X	Y	Z			⊦			
360	0	1	2	3	4	5	6	7	8	9	|					EO

ตารางที่ 5ตัวอักษรตามมาตรฐาน Original EBCDIC

3.2.4 Cyrillic EBCDIC

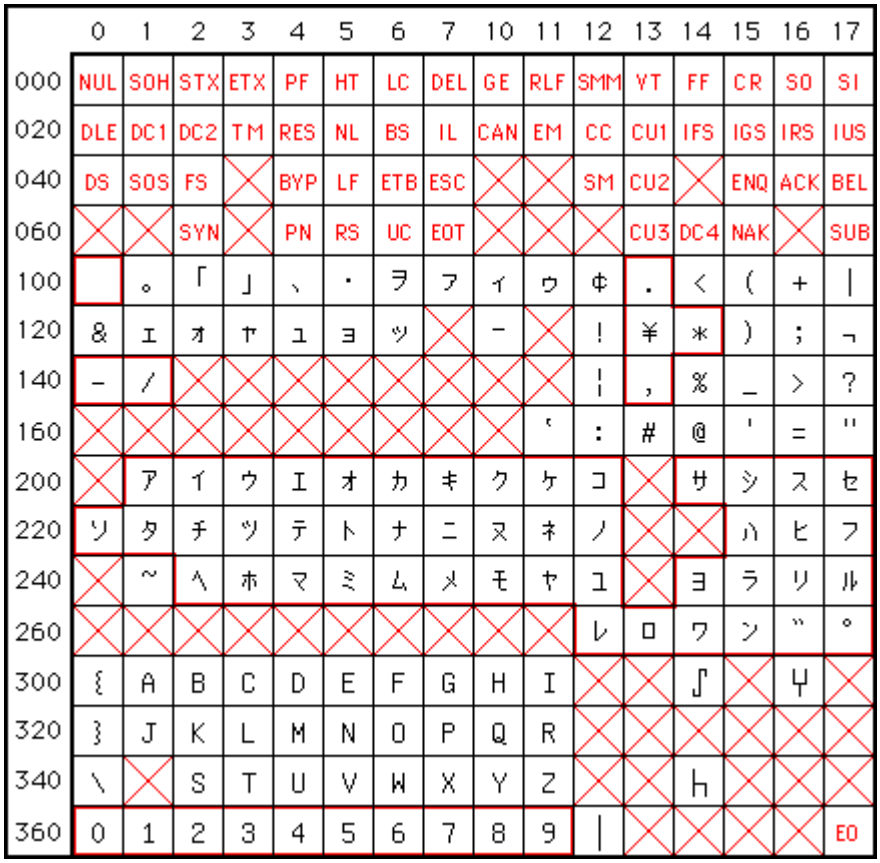
Cyrillic EBCDIC เป็นมาตรฐานของ EBCDIC ที่เพิ่มส่วนของตัวอักษร Cyrillic ของรัสเซียเข้าไป โดยจะทำการตัดตัวอักษรภาษาอังกฤษตัวพิมพ์เล็ก และเครื่องหมายบางส่วนออก แล้วเอาตัวอักษร Cyrillic ใส่เข้าไปแทน

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	PF	HT	LC	DEL	GE	RLF	SMM	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	TM	RES	NL	BS	IL	CAN	EM	CC	CU1	IFS	IGS	IRS	IUS
040	DS	SOS	FS		BYP	LF	ETB	ESC			SM	CU2		ENQ	ACK	BEL
060			SYN		PN	RS	UC	EOT				CU3	DC4	NAK		SUB
100											¢	.	<	(	+	|
120	&										!	¤	*	)	;	¬
140	-	/									¦	,	%	_	>	?
160										`	:	#	@	'	=	"
200		Ю	А	Б	Ц	Д	Е	Ф	Г	Х			И		Й	
220	К	Л	М	Н	О	П	Я	Р	С	Т						
240		~	У	Ж	В	Ь	Ы	З	Ш	Э			Щ			
260												Ч	Ъ			
300	{	A	B	C	D	E	F	G	H	I			ſ		Ч	
320	}	J	K	L	M	N	O	P	Q	R						
340	\		S	T	U	V	W	X	Y	Z			Ь			
360	0	1	2	3	4	5	6	7	8	9	|					EO

ตารางที่ 6 ตัวอักษรตามมาตรฐาน Cyrillic EBCDIC

3.2.5 Japanese EBCDIC

Japanese EBCDIC จะเป็นมาตรฐาน EBCDIC ที่มีการเพิ่มตัวอักษรคาตาคานะของญี่ปุ่นบางตัวเข้าไป โดยในช่วงต้นจะทำการเพิ่มเข้าไปแทนที่ตัวอักษรภาษาอังกฤษตัวพิมพ์เล็กที่ถูกตัดทิ้งไป แต่ต่อมาได้มีการออกมาตรฐานใหม่เป็น Revised Japanese EBCDIC ซึ่งจะยังคงมีตัวอักษร ภาษาอังกฤษตัวพิมพ์เล็กอยู่เหมือนเดิม แต่ตัวอักษรคาตาคานะที่เพิ่มเข้าไป จะไปแทรกอยู่ตามส่วนที่ยังว่างอยู่แทน



	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	PF	HT	LC	DEL	GE	RLF	SMM	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	TM	RES	NL	BS	IL	CAN	EM	CC	CU1	IFS	IGS	IRS	IUS
040	DS	SOS	FS		BYP	LF	ETB	ESC			SM	CU2		ENQ	ACK	BEL
060			SYN		PN	RS	UC	EOT				CU3	DC4	NAK		SUB
100		。	「	」	、	・	ヲ	ァ	ィ	ゥ	¢	.	<	(	+	|
120	&	ェ	ォ	ャ	ュ	ョ	ッ		−		!	¥	*	)	;	¬
140	−	/									¦	,	%	_	>	?
160										‘	:	#	@	'	=	"
200		ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ		サ	シ	ス	セ
220	ソ	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ			ハ	ヒ	フ
240		~	ヘ	ホ	マ	ミ	ム	メ	モ	ヤ	ユ		ヨ	ラ	リ	ル
260											レ	ロ	ワ	ン	゛	°
300	{	A	B	C	D	E	F	G	H	I			ʃ		Ψ	
320	}	J	K	L	M	N	O	P	Q	R						
340	\		S	T	U	V	W	X	Y	Z			┣			
360	0	1	2	3	4	5	6	7	8	9	|					EO

ตารางที่ 7ตัวอักษรตามมาตรฐาน Japanese EBCDIC

3.2.6 Augmented EBCDIC

หลังจากที่ ISO ได้กำหนดมาตรฐาน ISO 8859-1 ขึ้นมา ก็ได้มีการนำเอาสัญลักษณ์บางอย่างที่มีในมาตรฐาน ISO 8859-1 มาใช้ในมาตรฐาน EBCDIC โดยสัญลักษณ์ที่นำเข้ามาจะไปอยู่ตามส่วนที่ยังว่างอยู่ในตารางตัวอักษร ถัดมาได้มีการนำเอาสัญลักษณ์บางส่วนจากมาตรฐาน ANSI มาใส่ใช้ในมาตรฐาน EBCDIC และเปลี่ยนชื่อเป็นมาตรฐาน Revised Augmented EBCDIC

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	PF	HT	LC	DEL	GE	RLF	SMM	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	TM	RES	NL	BS	IL	CAN	EM	CC	CU1	IFS	IGS	IRS	IUS
040	DS	SOS	FS		BYP	LF	ETB	ESC			SM	CU2		ENQ	ACK	BEL
060			SYN		PN	RS	UC	EOT				CU3	DC4	NAK		SUB
100			]	¡	[	£	¤	¥	§	¨	¢	.	<	(	+	|
120	&	©	ª	«	^	-	®	¯	°	±	!	$	*	)	;	¬
140	-	/	²	³	´	µ	¶	·	¸	¹	¦	,	%	_	>	?
160	º	»	¼	½	¾	¿	À	Á	Â	'	:	#	@	'	=	"
200	Ã	a	b	c	d	e	f	g	h	i	Ä	Å	Æ	Ç	È	É
220	Ê	j	k	l	m	n	o	p	q	r	Ë	Ì	Í	Î	Ï	Ð
240	Ñ	~	s	t	u	v	w	x	y	z	Ò	Ó	Ô	Õ	Ö	×
260	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á	â	ã	ä	å	æ	ç
300	{	A	B	C	D	E	F	G	H	I	è	é	ê	ë	ì	í
320	}	J	K	L	M	N	O	P	Q	R	î	ï	ð	ñ	ò	ó
340	\		S	T	U	V	W	X	Y	Z	ô	õ	ö	÷	ø	ù
360	0	1	2	3	4	5	6	7	8	9	ú	û	ü	ý	þ	ÿ

ตารางที่ 8 ตัวอักษรตามมาตรฐาน Augmented EBCDIC

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	ST	HT	SSA	DEL	EPA	RI	SS2	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	DC3	OSC	NL	BS	ESA	CAN	EM	PU2	SS3	IFS	IGS	IRS	IUS
040	PAD	HOP	BPH	NBH	IND	LF	ETB	ESC	HTS	HTJ	VTS	PLD	PLU	ENQ	ACK	BEL
060	DCS	PU1	SYN	STS	CCH	MW	SPA	EOT	SOS	SGCI	SCI	CSI	DC4	NAK	PM	SUB
100		NBS	â	ä	à	á	ã	å	ç	ñ	[	.	<	(	+	!
120	&	é	ê	ë	è	í	î	ï	ì	ß	]	$	*	)	;	^
140	-	/	Â	Ä	À	Á	Ã	Å	Ç	Ñ	¦	,	%	_	>	?
160	ø	É	Ê	Ë	È	Í	Î	Ï	Ì	'	:	#	@	'	=	"
200	Ø	a	b	c	d	e	f	g	h	i	«	»	ð	ý	þ	±
220	°	j	k	l	m	n	o	p	q	r	ª	º	æ	¸	Æ	¤
240	µ	~	s	t	u	v	w	x	y	z	¡	¿	Ð	Ý	Þ	®
260	¢	£	¥	·	©	§	¶	¼	½	¾	¬	|	¯	¨	´	×
300	{	A	B	C	D	E	F	G	H	I	-	ô	ö	ò	ó	õ
320	}	J	K	L	M	N	O	P	Q	R	¹	û	ü	ù	ú	ÿ
340	\	÷	S	T	U	V	W	X	Y	Z	²	Ô	Ö	Ò	Ó	Õ
360	0	1	2	3	4	5	6	7	8	9	³	Û	Ü	Ù	Ú	APC

ตารางที่ 9 ตัวอักษรตามมาตรฐาน Revised Augmented EBCDIC

3.2.7 TIS 620-EBCDIC

TIS 620-EBCDIC เป็นมาตรฐาน EBCDIC เวอร์ชั่นภาษาไทย โดยจะมีการเพิ่มส่วนของภาษาไทยเข้าไปตรงส่วนที่ยังว่างอยู่ โดยยังคงมีตัวอักษรภาษาอังกฤษตัวพิมพ์เล็กอยู่เหมือนเดิม

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	ST	HT	SSA	DEL	EPA	RI	SS2	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	DC3	OSC	NL	BS	ESA	CAN	EM	PU2	SS3	IFS	IGS	IRS	IUS
040	PAD	HOP	BPH	NBH	IND	LF	ETB	ESC	HTS	HTJ	VTS	PLD	PLU	ENQ	ACK	BEL
060	DCS	PU1	SYN	STS	CCH	MW	SPA	EOT	SOS	SGCI	SCI	CSI	DC4	NAK	PM	SUB
100		NBS	ก	ข	ฃ	ค	ฅ	ฆ	ง		[	.	<	(	+	!
120	&		จ	ฉ	ช	ซ	ฌ	ญ	ฎ		]	$	*	)	;	^
140	-	/	ฏ	ฐ	ฑ	ฒ	ณ	ด	ต		¦	,	%	_	>	?
160	฿	๎	ถ	ท	ธ	น	บ	ป	ผ		:	#	@	'	=	"
200	๐	a	b	c	d	e	f	g	h	i	ฝ	พ	ฟ	ภ	ม	ย
220	ๆ	j	k	l	m	n	o	p	q	r	ร	ฤ	ล	ฦ	ว	ศ
240	๏	~	s	t	u	v	w	x	y	z	ษ	ส	ห	ฬ	อ	ฮ
260	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	ๅ	ะ	ั	า	ำ	็
300	{	A	B	C	D	E	F	G	H	I	-	ิ	ี	ึ	่	๋
320	}	J	K	L	M	N	O	P	Q	R	ุ	ู	ฺ	฿	๊	๏
340	\		S	T	U	V	W	X	Y	Z	๑	๒	๓	'	๚	๛
360	0	1	2	3	4	5	6	7	8	9	๔	๕	๖			APC

ตารางที่ 10 ตัวอักษรตามมาตรฐาน TIS-620 EBCDIC

3.3 ISO 8859 Standard

3.4 Unicode Standard

โดยพื้นฐานแล้ว คอมพิวเตอร์จะเกี่ยวข้องกับเรื่องของตัวเลข คอมพิวเตอร์จัดเก็บตัวอักษรและอักขระอื่นๆ โดยการกำหนดหมายเลขให้สำหรับแต่ละตัว ก่อนหน้าที่ Unicode จะถูกสร้างขึ้น, ได้มีระบบ encoding อยู่หลายร้อยระบบสำหรับการกำหนดหมายเลขเหล่านี้ ไม่มี encoding ใดที่มีจำนวนตัวอักขระมากเพียงพอ ยกตัวอย่างเช่น, เฉพาะในกลุ่มสหภาพยุโรปเพียงแห่งเดียว ก็ต้องการหลาย encoding ในการครอบคลุมทุกภาษาในกลุ่ม หรือแม้แต่ในภาษาเดี่ยว เช่น ภาษาอังกฤษ ก็ไม่มี encoding ใดที่เพียงพอสำหรับทุกตัวอักษร, เครื่องหมายวรรคตอน และสัญลักษณ์ทางเทคนิคที่ใช้กันอยู่ทั่วไป

ระบบ encoding เหล่านี้ยังขัดแย้งซึ่งกันและกัน. นั่นก็คือ, ในสอง encoding สามารถใช้หมายเลขเดียวกันสำหรับตัวอักขระสองตัวที่*แตกต่างกัน*,หรือใช้หมายเลขต่างกันสำหรับอักขระ*ตัวเดียวกัน*. ใน

ระบบคอมพิวเตอร์ (โดยเฉพาะเซิร์ฟเวอร์) ต้องมีการสนับสนุนหลาย encoding; และเมื่อข้อมูลที่ผ่านไปมาระหว่างการเข้ารหัสหรือแพล็ตฟอร์มที่ต่างกัน, ข้อมูลนั้นจะเสี่ยงต่อการผิดพลาดเสียหาย

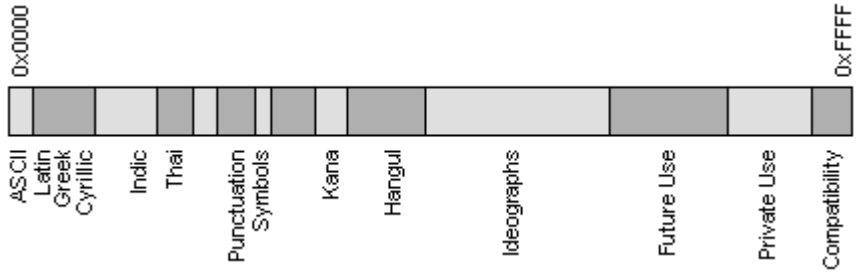
3.4.1 Unicode benefits

Unicode กำหนดหมายเลขเฉพาะสำหรับแต่ละอักขระ, โดยไม่สนใจว่าเป็นแพล็ตฟอร์มใด, ไม่ขึ้นกับว่าจะเป็นโปรแกรมใดและไม่ว่าจะเป็นภาษาใด. มาตรฐาน Unicode ได้ถูกนำไปใช้โดยผู้นำในอุตสาหกรรม เช่น Apple, HP, IBM, JustSystem, Microsoft, Oracle, SAP, Sun, Sybase, Unisys และอื่นๆ อีกมาก. Unicode เป็นสิ่งที่จำเป็นสำหรับมาตรฐานใหม่ๆ เช่น XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML ฯลฯ., และเป็นแนวทางอย่างเป็นทางการในการทำ ISO/IEC 10646. Unicode ได้รับการสนับสนุนในระบบปฏิบัติการจำนวนมาก, บราวเซอร์ใหม่ๆ ทกตัว, และผลิตภัณฑ์อื่นๆ อีกมาก. การเกิดขึ้นของ Unicode Standard และทูลส์ต่างๆ ที่มีในการสนับสนุน Unicode, เป็นหนึ่งในแนวโน้มทางเทคโนโลยีซอฟต์แวร์ระดับโลกที่มีความสำคัญที่สุด

3.4.2 What is Unicode?

Unicode เป็นการเข้ารหัสตัวอักษรแบบ 16 bit (ในช่วงเริ่มต้น) ซึ่งจะรวมทุก ๆ ตัวอักษรที่มีใช้โดยทั่วไปในการประมวลผล ทำให้สามารถมีตัวอักษรได้ทั้งหมด 65535 ตัวอักษร และในขณะนี้ก็ยังมีที่ว่างเหลืออีกประมาณ 1 ใน 3 ที่สามารถนำตัวอักษรใหม่ ๆ มาใส่ได้

Unicode ถูกสร้างขึ้นมาเพื่อแก้ปัญหาเกี่ยวกับการทำงานที่มีการส่งผ่านข้อมูลข้ามกันในหลาย ๆ ภาษา โดยมาตรฐานนี้จะเป็นข้อตกลงในการจัดเก็บตัวอักษรซึ่งไปรับการยอมรับจากสมาชิกของ Unicode Consortium



ตารางที่ 11 ตำแหน่งการเก็บข้อมูลตัวอักษรในแต่ละภาษา

Unicode Consortium ได้รับการก่อตั้งขึ้นมา ก่อนที่ ISO/IEC จะกำหนดมาตรฐาน ISO/IEC 10646 นี้ขึ้นสำหรับเป็นมาตรฐานสากล ในปัจจุบัน Unicode Consortium เป็นคณะกรรมการ และเป็นผู้ลงคะแนนร่วมกับผู้แทนจากหลาย ๆ ประเทศสมาชิก ในการร่างและกำหนดมาตรฐาน เกี่ยวกับ Unicode ด้วย

หลักการเบื้องต้นของ Unicode ก็คือแต่ละภาษาจะเป็นอิสระต่อกัน ตัวอักษรหนึ่ง ๆ จะไม่สามารถระบุภาษาที่ใช้ได้ เช่นตัวอักษร “a” จะสามารถเป็นได้ทั้งภาษาฝรั่งเศส,เยอรมัน หรือแม้แต่ภาษาอังกฤษ

3.4.3 Forms of Unicode

ในช่วงเริ่มต้น Unicode ถูกออกแบบมาให้ใช้การ encode แบบ 16 bit คงที่เพื่อสนับสนุนตัวอักษรในทุก ๆ ภาษา ซึ่งเพียงพอสำหรับจำนวนตัวอักษรในขณะนั้น แต่เมื่อเวลาผ่านไปจำนวนตัวอักษรก็เพิ่มมากขึ้น รวมถึงต้องมีการเพิ่มตัวอักษรเพื่อให้สนับสนุนการทำงานร่วมกับชุดตัวอักษรแบบเก่า ดังนั้นเพียง 16 bit จึงไม่เพียงพอสำหรับการ encode ตัวอักษรทั้งหมดอีกต่อไป

จากปัญหาเรื่องจำนวนตัวอักษรที่เพิ่มขึ้น Unicode จึงได้ทำการกำหนดมาตรฐานเพิ่มเติมเพื่อให้สามารถรองรับจำนวนตัวอักษรที่เพิ่มขึ้นได้ โดยใช้ตัวอักษร Unicode เป็นคู่เพื่อแทนค่าตัวอักษร 1 ตัว เรียกว่า “surrogates” ซึ่งจะรองรับจำนวนตัวอักษรได้ถึง 1,000,000 ตัวอักษร

นอกจากนี้ ในระบบเก่าบางระบบยังไม่สามารถที่จะใช้ตัวอักษรขนาด 16 bit ในการประมวลผลได้ ดังนั้นจะต้องมีการกำหนดมาตรฐานของ Unicode ให้รองรับการ encode รหัสตัวอักษรแบบ 8 bit ด้วย

จากความต้องการทั้งหมดที่กล่าวมา Unicode จึงถูกกำหนดถึงมาในรูปแบบที่แตกต่างกันทั้งหมด 3 รูปแบบ คือ UTF-8, UTF-16 และ UTF-32 ซึ่งแต่ละรูปแบบจะเหมาะสม (UTF : UCS Transformation Format)

1. UTF-8

   UTF-8 เป็นการ encode ตัวอักษรโดยใช้ byte หลาย ๆ byte มาต่อกันขึ้นอยู่กับรหัสของตัวอักษาที่ต้องการจะ encode โดยตามทฤษฎีแล้วจะสามารถใช้ได้ตั้งแต่ 1-6 bytes เรียงต่อกัน โดยลักษณะสำคัญคือ ตัวอักษรทีมีรหัสระหว่าง U+0000 ถึง U+007F (ASCII character) จะถูก encode โดยใช้จำนวน byte เพียง 1 byte (0x00 ถึง 0x7F) ซึ่งจะตรงกับรหัสของการเข้ารหัสแบบ ASCII ตัวอักษรอื่น ๆ ที่มีรหัสมากกว่า U+007F จะถูก encode โดยการนำหลาย ๆ byte มาต่อกัน ซึ่งจะมีการกำหนด bit ที่เป็น most significant bit ดังนั้นค่ารหัสของตัวอักษร ASCII (0x00-0x7F) จะไม่ปรากฏอยู่บนส่วนใดเลยของตัวอักษรนั้น Byte แรกของแต่ละตัวอักษรที่ไม่ใช้ตัวอักษร ASCII จะอยู่ในช่วง 0xC0 ถึง 0xFD เสมอ ซึ่งจะใช้ในการระบุว่ามีอีกกี่ byte ที่ตามมาสำหรับตัวอักษรตัวนั้น และทุก ๆ byte ที่ตามมาจะต้องอยู่ในช่วง 0x80 ถึง 0xBF เท่านั้น ซึ่งจะทำให้ง่ายในการ resynchronization และการตรวจสอบ byte ที่อาจจะหายไประหว่าการส่งข้อมูล มีจำนวนตัวอักษรที่รองรับทั้งหมด 2^31 ตัวอักษร Byte 0xFE และ 0xFF จะไม่ถูกใช้ใน UTF-8 encoding

2. UTF-16

UTF-16 จะเป็นมาตรฐานที่ถูกกำหนดอยู่ใน Unicode standard version 3.0 โดยจะเป็นการ encode ตัวอักษรให้อยู่ในรูปของ 16-bit integer ต่อกัน 1-2 ตัว ขึ้นอยู่กับรหัสของตัวอักษรที่จะ encode โดยลักษณะของ UTF-16 คือ ตัวอักษรที่มีรหัสน้อยกว่า 0x10000 จะถูก encode อยู่ในรูปของ 16-bit integer 1 ตัว โดยค่าของ 16-bit integer นั้นจะเท่ากับค่าของรหัสตัวอักษรนั้น ตัวอักษรที่มีรหัสอยู่ระหว่าง 0x10000 ถึง 0x10FFFF จะถูก encode อยู่ในรูปของ 16-bit integer 1 ตัว ซึ่งมีค่าอยู่ระหว่าง 0xD800 ถึง 0xD8FF (เรียกว่า high-half zone หรือ high surrogate area) แล้วตามด้วย 16-bit integer อีก 1 ตัวที่มีค่าอยู่ระหว่าง 0xDC00 ถึง 0xDFFF (เรียกว่า low-half zone หรือ low surrogate area) ลักษณะของการ encode แบบนี้จะได้เป็น <high-half zone code> <low-half zone code>

ตัวอักษรที่มีค่ามากกว่า 0x10FFFF จะไม่สามารถ encode โดยใช้ UTF-16 ได้

3.

3.5 Others Standards

3.5.1 GOST Standard

GOST เป็นมาตรฐาน ที่ ถือกำเนิด จาก Soviet Union โดย Version แรก ที่ออกมาตรฐานนี้มา จะ ใช้แทน ตัวอักษร Cyrillic ตัวใหญ่ และ ต่อมา ก็เพิ่มภาษาอังกฤษ ด้วย มาตรฐานนี้ ได้กลาย มาใช้ พัฒนา ระบบมาตรฐานอื่น ๆ เช่น EBCDIC

GOST มีทั้งหมด 5 Version โดยมีดังนี้

- 10859
- 10859 Latin
- 13025
- 19768/74
- 19768/87

1. GOST 10859 เป็นมาตรฐาน Version แรก ที่ ทำขึ้น โดย มีการเริ่ม พัฒนานี้เมื่อ ค.ศ. 1964 ซึ่งเป็นการพัฒนา พร้อม ๆ กับ ฝ่าย ASA จาก US standard โดยใน version นี้ จะมีแต่ ตัวอักษร Cyrillic

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	0	1	2	3	4	5	6	7	8	9	+	-	/	,	.	
020	10	↑	(	)	×	=	;	[	]	*	'	'	≠	<	>	:
040	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
060	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь	Э	Ю	Я	D
100	F	G	I	J	L	N	Q	R	S	U	V	W	Z	¯	≤	≥
120	∨	∧	⊃	¬	÷	≡	%	◇	|	−	_	!	''	Ъ	°	'
140	→	←	?	↓	Ø	±	▽									
160																DEL

ตารางที่ 12 ตัวอักษรตามมาตรฐาน GOST 10859

การแทนค่า ตัวอักษร โดยใช้เป็นตัวกราฟิก โดย ใช้ Algol60 Program มาตรฐาน ข้อมูลที่ใช้คือ ข้อมูลแถวแรก แทนด้วย 4 bit

ข้อมูลสองแถวแรก แทนด้วย 5 bit

ข้อมูลสี่แถวแรก แทนด้วย 6 bit

2. GOST 10859 Latin

คือมาตรฐาน รุ่นต่อมาถัด จาก 10859 โดย เพิ่มส่วนของภาษาอังกฤษ เพิ่ม โดยใช้ข้อมูล 6 bit ใช้ในโปรแกรม Algol60

3. GOST 13025

Version นี้จะมีทั้ง ภาษา Cyrillic มีทั้งตัวใหญ่ และตัวเล็ก

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
040		!	''	#	ж	%	&	'	(	)	*	+	,	-	.	/
060	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
120	п	я	р	с	т	у	ж	в	ь	ы	з	ш	э	щ	ч	_
140	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
160	П	Я	Р	С	Т	У	Ж	В	Ь	Ы	З	Ш	Э	Щ	Ч	DEL

ตารางที่ 13 ตัวอักษรตามมาตรฐาน GOST 13025

4. GOST 19768/74

คือ Version ที่มีมาตรฐาน ใช้เป็นแบบอย่างให้ มาตรฐาน EBCDIC และ ใช้ ใน บัตรเจาะ ข้อมูล (punched card) ด้วย

5. GOST 19768/87

เป็น Version สุดท้าย ที่สมบูรณ์ ของ GOST โดยมีการแก้ไขให้สมบูรณ์ เมื่อ 1987 และมาตรฐานนี้ ได้กลายมาเป็น พื้นฐาน ของมาตรฐาน ISO 8859-5 และ EBCDI

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
040		!	"	#	¤	%	&	'	(	)	*	+	,	-	.	/
060	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
120	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
140	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
160	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
200																
220																
240	NBS	Ё	Ђ	Ѓ	Є	S	I	Ї	J	Љ	Њ	Ћ	Ќ	SHY	Ў	Џ
260	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
300	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
320	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
340	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
360	№	ё	ђ	ѓ	є	s	i	ї	j	љ	њ	ћ	ќ	§	ў	џ

ตารางที่ 14 ตัวอักษรตามมาตรฐาน GOST 19768/87

3.5.2 JISC II Standard

JSSC II คือ มาตรฐานที่ จัดทำโดย สมาคมอุตสาหกรรม ประเทศ ญี่ปุ่น (Japan Industrial Standards Committee) โดยมีการแก้ไขพัฒนาต่อมาจาก มาตรฐาน JIS X 0201-1976 โดย มี ตัวหนังสือ ทั้งหมด 4 แบบ คือ

- Hiragana
- Katakana
- Kanji (Chinese characters)
- Latin alphabet

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
020	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
040		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
060	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
120	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
140	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
160	p	q	r	s	t	u	v	w	x	y	z	{	|	}	¯	DEL
200																
220																
240		。	「	」	、	・	ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
260	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
300	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
320	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	゛	°
340																
360																

ตารางที่ 15 ตัวอักษรตามมาตรฐาน JISC II Standard

3.5.3 Korean Standard

ประเทศเกาหลีเป็นประเทศเดียวในเอเชียตะวันออกที่ตัวหนังสือแบบเขียนของชาติเป็นอักขระเรียงตามลำดับที่ถูกเรียกว่า hangul แม้ว่าตัวอักษรต่างๆ ของประเทศจีนจะถูกผสมเข้าด้วยกันกับ hangul ในการเขียน แนวโน้มในประเทศเกาหลีจะเปลี่ยนจากการใช้ตัวอักษรต่าง ๆ ของประเทศจีน ยกเว้นสำหรับที่ถูกใช้ในการแทนชื่อและชื่อของบุคคลต่าง ๆ ตัวหนังสือแบบเขียน hangul มีแค่ 24 ตัวอักษร (มีสระ 10 ตัวและพยัญชนะ 14 ตัว) ดังนั้นมันจึงดูเหมือนว่าการประมวลผลตัวอักษรบนคอมพิวเตอร์จะทำได้ง่ายเมื่อเทียบกับภาษาอื่น อย่างไรก็ตามในการเขียนตัวอักษรต่าง ๆ ของ hangul จะสร้างอยู่ในรูปของพยางค์ที่ซึ่งตัวอักษรต่าง ๆ ที่อยู่ในสแตกในส่วนบนสุดของแต่ละอันและสิ่งนี้ได้ถูกนำไปสู่การอภิปรายเกี่ยวกับตัวอักษรต่าง ๆ ของระบบการเขียนของประเทศเกาหลีที่แท้จริงควรจะถูกประมวลผลภายในระบบคอมพิวเตอร์อย่างไร

มาตรฐานปัจจุบันสำหรับการประมวลผลข้อมูลและการแลกเปลี่ยนข้อมูลข่าวสารคือ KS C 5601-1992 (KS คือมาตรฐานของประเทศเกาหลี) ได้จดทะเบียนพยางค์ที่ถูกใช้บ่อยที่สุดในแม่พิมพ์พยางค์ที่ถูกจัดเรียงเอาไว้ก่อนแล้ว มาตรฐาน KS C 5601-1992 ได้กำหนดขึ้นดังนี้

- สัญลักษณ์ 94 ตัว
- คำย่อและสัญลักษณ์ต่าง ๆ 69 ตัว

- ISO 646-KR ด้วยตัวอักษรเต็มความกว้าง 94 ตัว
- ส่วนประกอบต่าง ๆ ของ hangul 94 ตัว
- ตัวเลขโรมันและตัวอักษรที่เรียงตามลำดับของกรีก 68 ตัว
- ส่วนประกอบในการวาดเส้น 68 ตัว
- คำย่อต่าง ๆ 79 ตัว
- สัญลักษณ์ต่าง ๆ เกี่ยวกับเสียงของภาษา, ตัวอักษรต่าง ๆ ที่ถูกทำเป็นวงกลม และเศษส่วนต่าง ๆ 91 ตัว
- สัญลักษณ์ต่าง ๆ เกี่ยวกับเสียงของภาษา, ตัวอักษรต่าง ๆ ที่ถูกเสริม, ตัวห้อยต่าง ๆ และตัวยกต่าง ๆ 94 ตัว
- Hiragana 83 ตัว
- Katakana 86 ตัว
- ตัวอักขระเรียงตามลำดับของรัสเซีย 66 ตัว
- Hangul (พยางค์ที่ถูกเชื่อมเข้าด้วยกันแล้ว 2,350 ตัว)
- Hanja (ตัวอักษรต่าง ๆ ของประเทศจีนที่ถูกเรียงตามลำดับโดยการอ่านแบบ hangul และรากของคำ 4,888 ตัว)

ภาษาเกาหลีถูกเข้ารหัสด้วยวิธีต่าง ๆ ดังนี้ 7 บิท ISO 2022, ISO-2022-KR (การเข้ารหัสข้อความทางอีเมล์), EUC-KR และ Johab หรือรหัสที่เป็นการรวมกันของ 2 ไบท์ เป็นระบบที่ซึ่งแม่พิมพ์พยางค์ของ hangul ที่เป็นไปได้ทั้งหมด (11,172 ตัว) ถูกเข้ารหัส ถูกใช้จริงในปัจจุบันหรือไม่ก็ตาม ไมโครซอฟท์ได้ทำการสร้าง Unified Hangul Code (UHC) หรือที่เรียกว่า Extended Wansung สำหรับระบบปฏิบัติการ Windows 95 ที่เป็นเวอร์ชันภาษาเกาหลี (Win95K) ซึ่งดูแลรักษาความเข้ากันได้กับ KS C 5601-1992 ขณะที่มีการเพิ่มการสนับสนุนสำหรับระบบการเข้ารหัส Johab แบบเต็ม (พยางค์ต่าง ๆ ที่ถูกรวมเข้าไว้ด้วยกันก่อนแล้ว 8,822 ตัว ที่ไม่อยู่ใน KS C 5601-1992) นอกจากนั้นทางบริษัทไมโครซอฟท์กำลังวางแผนที่จะสนับสนุนระบบนี้บนระบบปฏิบัติการ Windows NT ภาษาเกาหลีอีกด้วย

3.5.4 Thai Standard

- TIS-620

TIS-620 หรือ มอก. 620 หรือที่เรียก กันทั่วไปว่า รหัส สมอ. เป็นมาตรฐานของรหัสตัวอักษร (Charset Code) ที่ใช้บนคอมพิวเตอร์ ซึ่งกำหนดโดยสำนักงานมาตรฐานอุตสาหกรรม หรือ สมอ. (Thai Industrial Standards Institute [TISI]). TIS-620 เป็นรหัสตัวอักษรที่ต่อเพิ่มจากรหัสตัวอักษรของ ISO-646 ซึ่งเป็น รหัสตัวอักษรแบบ 7 bit คล้ายกับ ASCII

มาตรฐาน TIS-620 ตัวแรกคือ TIS-620 2529 (1986) ซึ่งได้มีการแก้ไขเพิ่มเติมอีก ในปี 2533 เป็น TIS-620 2533 (1990) เพื่อเพิ่มเนื้อหาบางส่วนให้สอดคล้องกับ ISO/IEC 2022 แต่ตารางรหัสตัวอักษรทั้งหมดยังคงเหมือนเดิม

ปัจจุบัน GNU C library (GLIBC) ได้สนับสนุนมาตรฐาน TIS-620 ในการใช้งาน สำหรับกับท้องถิ่นประเทศไทยและภาษาไทย ภายใต้ชื่อ th_TH (th_TH.TIS-620)

- ISO8859-11

รหัสตัวอักษรแบบ 8 bit ของ TIS-620 คล้ายกับ กับรหัสตัวอักษรในระบบ ISO/IEC 8859 มาก เนื่องจาก สมอ. (TISI) นั้นไม่ประสบความสำเร็จมากนักในการกระตุ้นให้ TIS-620 เป็นมาตรฐาน จึงได้คิดจะ ใส่ไว้ในระบบ ISO/IEC 8859 แทน เพื่อให้ในระบบอุตสาหกรรมต่าง ๆ หันมาใช้ตารางรหัสภาษาไทยตามมาตรฐานมากขึ้น ตารางนี้ได้รับการใส่ไว้ในส่วนที่ 11 (Part 11) ของมาตรฐาน ISO/IEC 8859 ถึงแม้จะมีการปฏิเสธการใช้มาตรฐานนี้เนื่องจากภาษาไทยนั้นต่างจากภาษาแบบละตินตรงที่ต้องมีการประกอบตัวอักษรเข้าด้วยกัน แต่ในภายหลัง ก็มีการผลักดันให้ ISO/IEC 8859 Part 11 ผ่านในที่ประชุม ISO และประกาศเป็นทางการในปลายปี พ.ศ. 2544

- ISO-10646-1

โปรแกรมในปัจจุบันได้เริ่มออกแบบให้สามารถใช้ได้หลายภาษา (Multilingual) โดยใช้มาตรฐานของตัวอักษร ของ ISO/IEC 10646 (Universal Multi-octet Coded Character Set - UCS) ซึ่งเป็นระบบสำหรับเก็บข้อมูลตัวอักษรสากลในระบบ 8bit (หรือ byte) ซึ่งอาจอยู่ในรูป 8 bit หลาย ๆ ตัวต่อกัน และรู้จักกันดีในชื่อ Unicode UCS หรือ UTF-8

Unicode Consortium ได้รับการก่อตั้งขึ้นมา ก่อนที่ ISO/IEC จะกำหนดมาตรฐาน ISO/IEC 10646 นี้ขึ้นสำหรับเป็นมาตรฐานสากล ในปัจจุบัน Unicode Consortium เป็นคณะกรรมการ และเป็นผู้ลงคะแนนร่วมกับผู้แทนจากหลาย ๆ ประเทศสมาชิก ในการร่างและกำหนดมาตรฐาน เกี่ยวกับ Unicode ด้วย

4 Reference

http://www.microsoft.com/typography/unicode/cs.htm

http://linux.thai.net/~sfalpha/thai-howto/Thai-HOWTO-2.html

http://www.unicode.org/reports/tr19/tr19-9.html

http://encyclozine.com/UTF-32

http://www-106.ibm.com/developerworks/library/utfencodingforms/