



中文字符编码

刘秉权

智能技术与自然语言处理研究室

新技术楼612室

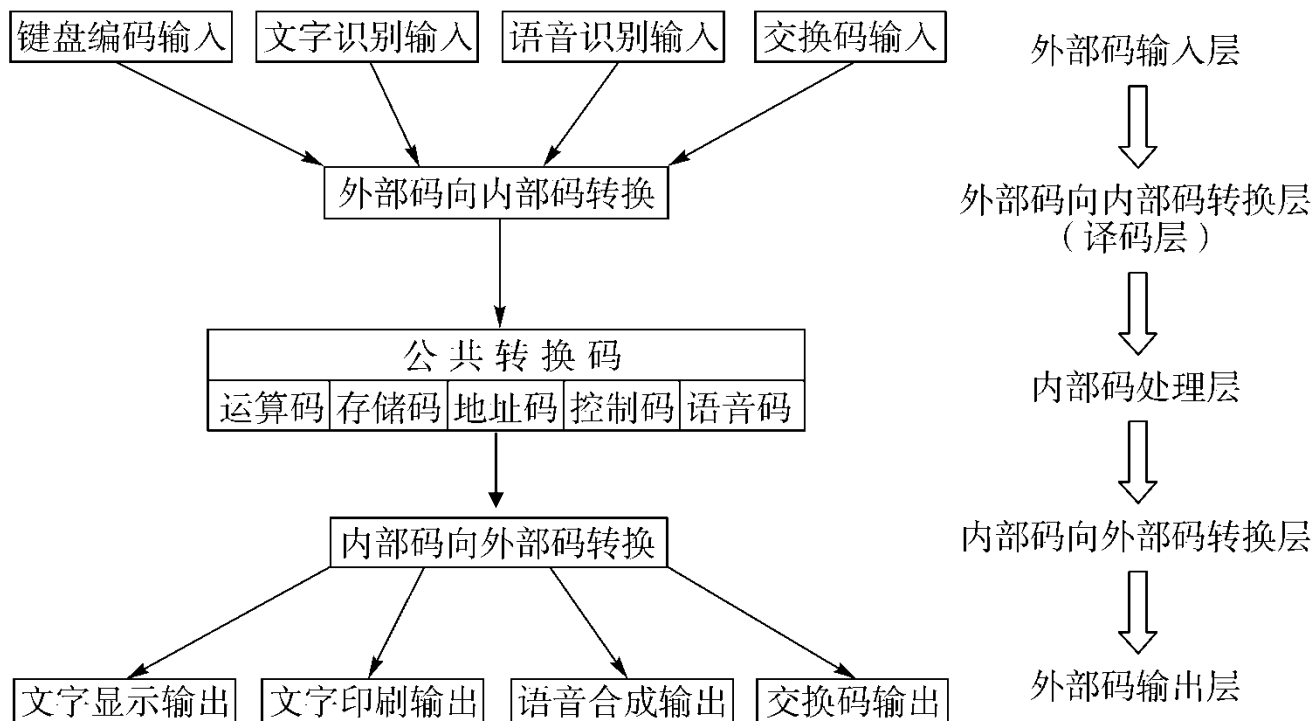
Email: liubq@hit.edu.cn



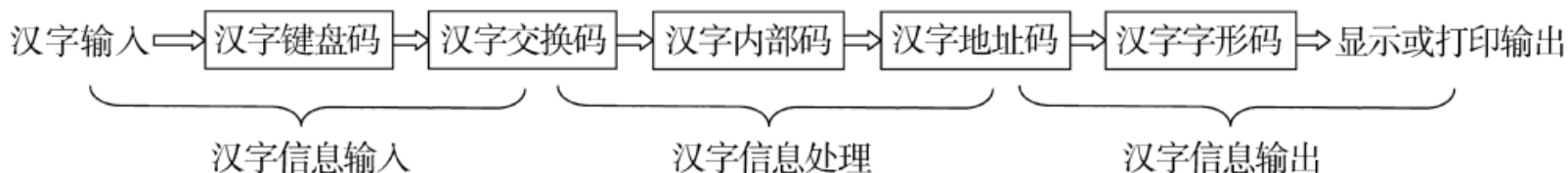
主要内容

- 汉字编码的含义
- 汉字编码的历史（现状）及其根源
- 主要汉字（文字）编码标准与规范
- ASCII码简介
- GB汉字编码体系
- BIG5码
- ISO/IEC 10646,Unicode
- 内码自动识别与转换

中文信息处理系统结构



中文信息处理过程中 汉字代码的变换流程





本章汉字编码的含义

- 汉字交换码（国家标准、国际标准）
- 汉字内部码（内码）



中英文兼容技术

- 出发点是完全保留并利用原来英文计算机系统的一切硬、软件功能,使系统能方便地处理中、英文混合信息流。
- 汉字的代码(即汉字信息交换码)要遵守英文、数字系统字符代码体系的数据格式。同时,要利用计算机原有的系统软件兼容中、英文两种代码,并要求系统能明确地区分两种代码。
- 用二个ASCII交叉组合成汉字信息交换码,汉字信息进入系统后,应对汉字代码添加相应的标识信息。
- 自Microsoft Windows 95版以后,开始使用Unicode作为统一的英文数字字符和汉字字符的编码,跨上了一个全新的台阶。



汉字编码的历史（现状）及其根源

- 多种编码方案共存，新旧标准同台使用，需相互转换，不利于交流和共享
- 中、日、韩、新等多国同时使用汉字
- 简繁体汉字并存
- 地区、国家间的文化、政治差异增加了汉字统一编码的难度
- 统一标准已经形成，尚需完善



汉字的几种通行名称

- 汉字
- Chinese character, Hanzi, Hantsu
- Ideographic character, 表意字符, 中文字符
- Kanji- 日文中的叫法
- Hanja- 朝鲜文中的叫法
- CJK- 中日韩通用字符集
- Unihan



相关用语

- simplified Hanzi, simplified Chinese character
- traditional (unsimplified) Chinese character
- Mandarin, Cantonese



主要汉字(文字)编码标准与规范

- 英文编码
 - ASCII码
- GB汉字编码体系（中国大陆、新加坡）
 - GB2312
 - GBK
 - GB13000
 - GB18030
- 繁体汉字编码（台湾、香港）
 - BIG5
- 其他语种编码体系
 - Shift_JIS
- 国际标准编码体系
 - ISO/IEC 10646
 - Unicode



ASCII码简介

- 美国信息交换标准编码(“美标”， American Standard Code for Information Interchange)
- 起始于50年代后期，在1967年定案
- 用从0到127的128个数字来代表信息的规范编码
- 包括33个控制码，一个空格码，和94个形象码
- 形象码中包括了英文大小写字母，阿拉伯数字，标点符号等
- 国际上大部分电脑的通用编码

ASCII码表

NUL	空字符	VT	垂直制表	SYN	空转同步
SOH	标题开始	FF	走纸控制	ETB	信息组传送结束
STX	正文开始	CR	回车	CAN	作废
ETX	正文结束	SO	移位输出	EM	纸尽
EOY	传输结束	SI	移位输入	SUB	换置
ENQ	询问字符	DLE	空格	ESC	换码
ACK	承认	DC1	设备控制 1	FS	文字分隔符
BEL	报警	DC2	设备控制 2	GS	组分分隔符
BS	退一格	DC3	设备控制 3	RS	记录分隔符
HT	横向列表	DC4	设备控制 4	US	单元分隔符
LF	换行	NAK	否定	DEL	删除

ASCII 值	控制字符	ASCII 值	控制字符	ASCII 值	控制字符	ASCII 值	控制字符
0	NUL	32	(space)	64	@	96	,
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	,	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	X	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

扩展ASCII码表

高四位 低四位		扩充ASCII码字符集															
		1000		1001		1010		1011		1100		1101		1110		1111	
		8		9		A/10		B/16		C/32		D/48		E/64		F/80	
		+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符
0000	0	128	Ç	144	É	160	á	176	⌘	192	Ł	208	⌚	224	α	240	≡
0001	1	129	Ü	145	æ	161	í	177	⌘	193	ł	209	⌚	225	β	241	±
0010	2	130	é	146	Æ	162	ó	178	⌘	194	ł	210	⌚	226	Γ	242	≥
0011	3	131	â	147	ô	163	ú	179	⌘	195	ł	211	⌚	227	π	243	≤
0100	4	132	ä	148	ö	164	ñ	180	⌘	196	ł	212	⌚	228	Σ	244	∫
0101	5	133	à	149	ò	165	Ñ	181	⌘	197	ł	213	⌚	229	σ	245	∫
0110	6	134	å	150	û	166	ä	182	⌘	198	ł	214	⌚	230	μ	246	÷
0111	7	135	ç	151	ù	167	ö	183	⌘	199	ł	215	⌚	231	τ	247	≈
1000	8	136	ê	152	ÿ	168	¿	184	⌘	200	ł	216	⌚	232	Φ	248	°
1001	9	137	ë	153	Ö	169	⌘	185	⌘	201	ł	217	⌚	233	Θ	249	•
1010	A	138	è	154	Ü	170	⌘	186	⌘	202	ł	218	⌚	234	Ω	250	·
1011	B	139	ï	155	ç	171	½	187	⌘	203	ł	219	⌚	235	δ	251	√
1100	C	140	î	156	£	172	¼	188	⌘	204	ł	220	⌚	236	∞	252	n
1101	D	141	ì	157	¥	173	¡	189	⌘	205	ł	221	⌚	237	φ	253	²
1110	E	142	Ä	158	℞	174	«	190	⌘	206	ł	222	⌚	238	ε	254	■
1111	F	143	Å	159	f	175	»	191	⌘	207	ł	223	⌚	239	∩	255	BLANK FF

注：表中的ASCII字符可以用：ALT + “小键盘上的数字键” 输入



文本文件与二进制文件

- 字符大都是用八个二进制数字表示，美标只规定了**128**个编码，剩下的另外**128**个数码没有规范，美标中的**33**个控制码，各厂家用法也不尽一致
- 文本文件(**ASCII Text Files**)：美标形象码或空格码组成，通常可在不同电脑系统间直接交换
- 二进制文件(**Binary Files**)：含有控制码或非美标码的文件，通常不能在不同电脑系统间直接交换



GB汉字编码体系

- GB2312
- GBK
- GB13000
- GB18030



GB2312（国标、区位、机内码）

- **国标**：中华人民共和国国家标准信息交换用汉字编码
- **国标(GB2312-80)**表（基本表）把七千余汉字、以及标点符号、外文字母等，排成一个94行、94列的方阵
- 每一横行叫一个“区”，每个区有九十四个“位”
- 一个汉字在方阵中的坐标，称为该字的“**区位码**”
- 例如“中”字在方阵中处于第 5 4 区第 4 8 位，它的区位码就是5448



区位码表

- 区位码来源于信息交换用汉字编码字符集（基本集）国家标准（GB2312-80），该标准收汉字6763个，第一级3755个，位于16至55区，55区的最后5个字符没有定义；第二级3008个，位于56至87区
- 第一级汉字按照汉语拼音字母顺序排列，同音字以笔形顺序横（一）、直（丨）、撇（丿）、点（丶）、折（乙）为序。起笔相同按第二笔，依次类推。
- 第二级汉字按部首排序，本标准采用的部首与一般字典用的部首基本相同，略有改并。部首次序及同部首字按笔划数排列，同笔划数的字以笔形顺序横（一）、直（丨）、撇（丿）、点（丶）、折（乙）为序。起笔相同按第二笔，依次类推。
- 查表时先查区号，再查行、列，例如：“、”是0102，“蔼”是1610。

例

01	区	1	2	3	4	5	6	7	8	9
0		,	。	•	-	˘	¨	”	々	
1	—	~		...	‘	’	“	”	{	}
2	<	>	《	》	「	」	『	』	[]
3	【	】	±	×	÷	:	∧	∨	Σ	Π
4	U	∩	∈	::	√	⊥	//	∠	∩	⊙
5	∫	ℳ	≡	≅	≈	∞	∞	≠	≠	≠
6	≤	≥	∞	∴	∴	♂	♀	°	'	"
7	℃	\$	⊗	⊗	£	%	§	N ₂	☆	★
8	○	●	◎	◇	◆	□	■	△	▲	※
9	→	←	↑	↓	≡					

02	区	1	2	3	4	5	6	7	8	9
0		i	ii	iii	iv	v	vi	vii	viii	ix
1	x								1.	2.
2	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.
3	14.	15.	16.	17.	18.	19.	20.	(1)	(2)	(3)
4	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
5	(14)	(15)	(16)	(17)	(18)	(19)	(20)	①	②	③
6	④	⑤	⑥	⑦	⑧	⑨	⑩	€	(-)	
7	(二)	(三)	(四)	(五)	(六)	(七)	(八)	(九)	(十)	
8	I	II	III	IV	V	VI	VII	VIII	IX	
9	X	XI	XII							

例

09 区	1	2	3	4	5	6	7	8	9
0				—	—			---	---
1	⋮	⋮	---	---	⋮	⋮	┐	┐	┐
2	┐	┐	┐	┐	┐	┐	┐	┐	┐
3	┐	┐	┐	┐	┐	┐	┐	┐	┐
4	┐	┐	┐	┐	┐	┐	┐	┐	┐
5	┐	┐	┐	┐	┐	┐	┐	┐	┐
6	┐	┐	┐	┐	┐	┐	┐	┐	┐
7	┐	┐	┐	┐	┐	┐	┐	┐	┐
8									
9									



例

16 区	1	2	3	4	5	6	7	8	9
0	啊	阿	埃	挨	哎	唉	哀	皑	癌
1	蔼	矮	艾	碍	爱	隘	鞍	氨	安
2	按	暗	岸	胺	案	肮	昂	盎	凹
3	熬	翱	袄	傲	奥	懊	澳	芭	捌
4	叭	吧	笆	八	疤	巴	拔	跋	靶
5	耙	坝	霸	罢	爸	白	柏	百	摆
6	败	拜	稗	斑	班	搬	扳	般	颁
7	版	扮	拌	伴	瓣	半	办	绊	邦
8	梆	榜	膀	绑	棒	磅	蚌	镑	傍
9	苞	胞	包	褒	剥				

17 区	1	2	3	4	5	6	7	8	9
0	薄	雹	保	堡	饱	宝	抱	报	暴
1	豹	鲍	爆	杯	碑	悲	卑	北	辈
2	贝	钡	倍	狈	备	惫	焙	被	奔
3	本	笨	崩	绷	甬	泵	蹦	迸	逼
4	比	鄙	笔	彼	碧	蓖	蔽	毕	毙
5	币	庇	痹	闭	敝	弊	必	辟	臂
6	避	陛	鞭	边	编	贬	扁	便	变
7	辨	辩	辫	遍	标	彪	膘	表	鳖
8	别	瘰	彬	斌	濒	滨	宾	摈	兵
9	柄	丙	秉	饼	炳				



例

54 区	1	2	3	4	5	6	7	8	9
0	幀	症	郑	证	芝	枝	支	吱	蚰
1	知	肢	脂	汁	之	织	职	直	植
2	执	值	侄	址	指	止	趾	只	旨
3	志	摯	掷	至	致	置	帜	峙	制
4	秩	稚	质	炙	痔	滞	治	室	盅
5	忠	钟	衷	终	种	肿	重	仲	众
6	周	州	洲	诒	粥	轴	肘	帚	咒
7	宙	昼	骤	珠	株	蛛	朱	猪	诸
8	逐	竹	烛	煮	拄	瞩	嘱	主	著
9	助	蛀	贮	铸	筑				

55 区	1	2	3	4	5	6	7	8	9
0	住	注	祝	驻	抓	爪	拽	专	砖
1	转	撰	赚	篆	桩	庄	装	妆	撞
2	状	椎	锥	追	赘	坠	缀	谆	准
3	拙	卓	桌	琢	茁	酌	啄	着	灼
4	兹	咨	资	姿	滋	淄	孜	紫	籽
5	滓	子	自	渍	字	髭	棕	踪	宗
6	总	纵	邹	走	奏	揍	租	足	卒
7	祖	诅	阻	组	钻	纂	嘴	醉	最
8	尊	遵	昨	左	佐	柞	做	作	坐
9									

例

56 区	1	2	3	4	5	6	7	8	9
0	亍	丌	兀	丐	廿	卅	丕	亘	丞
1	鬲	舜	噩	丨	禺	丿	乚	乇	夭
2	卮	氏	凶	胤	馗	毓	皐	𠂔	亟
3	鼎	乚	乚	亅	𠂔	𠂔	𠂔	𠂔	𠂔
4	厝	厝	厝	厝	厝	厝	厝	厝	厝
5	匾	𠂔	卦	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔
6	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔
7	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔
8	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔
9	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔

57 区	1	2	3	4	5	6	7	8	9
0	佟	佗	佻	伽	佶	佻	侑	侑	侃
1	侏	侑	佻	侑	侑	侑	侑	侑	侑
2	侏	侑	佻	侑	侑	侑	侑	侑	侑
3	倬	倬	倬	倬	倬	倬	倬	倬	倬
4	偃	偃	偃	偃	偃	偃	偃	偃	偃
5	僖	僖	僖	僖	僖	僖	僖	僖	僖
6	余	余	余	余	余	余	余	余	余
7	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔
8	兗	毫	衰	衰	衰	衰	衰	衰	衰
9	羸	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔

例

86 区	1	2	3	4	5	6	7	8	9
0	觥	觥	觥	觥	觥	觥	觥	觥	觥
1	霰	霰	霰	霰	霰	霰	霰	霰	霰
2	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩
3	𪚪	𪚪	𪚪	𪚪	𪚪	𪚪	𪚪	𪚪	𪚪
4	𪚫	𪚫	𪚫	𪚫	𪚫	𪚫	𪚫	𪚫	𪚫
5	𪚬	𪚬	𪚬	𪚬	𪚬	𪚬	𪚬	𪚬	𪚬
6	𪚭	𪚭	𪚭	𪚭	𪚭	𪚭	𪚭	𪚭	𪚭
7	𪚮	𪚮	𪚮	𪚮	𪚮	𪚮	𪚮	𪚮	𪚮
8	𪚯	𪚯	𪚯	𪚯	𪚯	𪚯	𪚯	𪚯	𪚯
9	𪚰	𪚰	𪚰	𪚰	𪚰	𪚰	𪚰	𪚰	𪚰

87 区	1	2	3	4	5	6	7	8	9
0	𪚱	𪚱	𪚱	𪚱	𪚱	𪚱	𪚱	𪚱	𪚱
1	𪚲	𪚲	𪚲	𪚲	𪚲	𪚲	𪚲	𪚲	𪚲
2	𪚳	𪚳	𪚳	𪚳	𪚳	𪚳	𪚳	𪚳	𪚳
3	𪚴	𪚴	𪚴	𪚴	𪚴	𪚴	𪚴	𪚴	𪚴
4	𪚵	𪚵	𪚵	𪚵	𪚵	𪚵	𪚵	𪚵	𪚵
5	𪚶	𪚶	𪚶	𪚶	𪚶	𪚶	𪚶	𪚶	𪚶
6	𪚷	𪚷	𪚷	𪚷	𪚷	𪚷	𪚷	𪚷	𪚷
7	𪚸	𪚸	𪚸	𪚸	𪚸	𪚸	𪚸	𪚸	𪚸
8	𪚹	𪚹	𪚹	𪚹	𪚹	𪚹	𪚹	𪚹	𪚹
9	𪚺	𪚺	𪚺	𪚺	𪚺	𪚺	𪚺	𪚺	𪚺



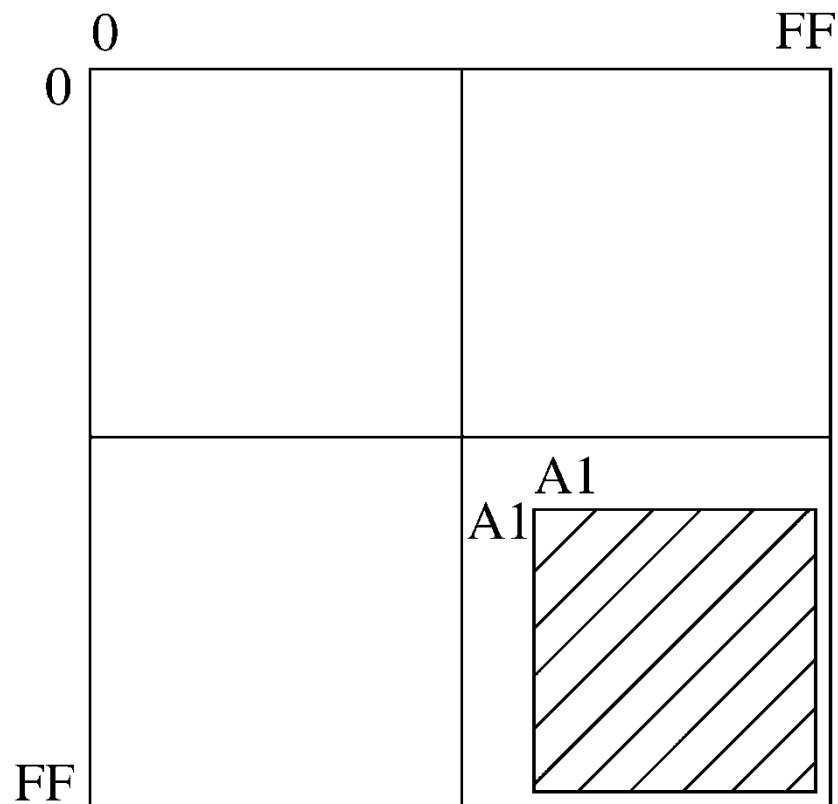
国标码、区位码、机内码

- 94: 美标中形象码的总数, 33--126
- 汉字区、位码各加上32, 就会与美标形象码的范围重合, 称为该字的“**国标码**”, 与其相对应的两个美标符号, 为该字的“**国标符**”
- 如何区分国标符与美标符: 国标码的两个数字各加上128, 称“**准国标**”或“**机内码**”
- 机内码 = (区位码) H + 2020H + 8080H

GB2312代码分布图

		C0				C1					
		00	20	40	60	80	A0	C0	E0	FF	
C0	00										
	20		GB2312								
	40										
	60										
	7F										
C1	80										
	A0		(1, 0)					(1, 1)			
	C0										
	E0										
	FF										

CCDOS汉字内码





GBK

- 汉字内码扩展规范，Rules/Specifications defining the extensions of internal codes for Chinese ideograms
- 为了推进Unicode的实施，同时也是为了向下兼容,由电子部与国家技术监督局联合颁布
- 在保持GB2312原貌的基础上，将其字汇扩充与ISO 10646中的CJK等量，同时也包容了台湾的工业标准Big5码汉字，此外还为用户留了1894个码位的自定义区



GB18030-2000

- 信息技术-信息交换用汉字编码字符集-基本集的扩充, Information technology-Chinese ideograms coded character set for information interchange-Extension for the basic set
- GBK的替代、超集



GB18030-2000

- 完全包含CJK(Unihan) Extension A
- 与GBK完全兼容(code- and character-compatible)的同时，为所有其它Unicode码点提供了空间
- 定义了4字节编码机制



GB18030-2000码位范围分配表

字节数	码位空间			
双字节	第一字节		第二字节	
	0x81—0xFE		0x40—0x7E, 0x80—0xFE	
四字节	第一字节	第二字节	第三字节	第四字节
	0x81—0xFE	0x30—0x39	0x81—0xFE	0x30—0x39

- 2字节编码共**23940**个码位
- 4字节编码共超过**150万**个码位



BIG5码

- 针对繁体汉字的编码，在台湾、香港的电脑系统中得到普遍应用

非汉字	第一字节	第二字节
	A1~A2	40~7E/A1~FE
	A3	40~7E/A1~E0
	C6	A1~FE
	C7~C8	40~7E/A1~FE
一级汉字	A4~C5	40~7E/A1~FE
	C6	40~7E
二级汉字	C9~F8	40~7E/A1~FE
	81~A0	40~7E/A1~D5



“标准” 历程

1983年	“通用汉字标准交换码”试用版发布，包括13,053个字与441个符号，分为二个字面，先笔画数，后部首序排列。 12月：Big-5大五码，包括13,053个字与441个符号，字集与字序与交换码试用版完全相同，仅字码定义不同。
1984年	3月：台湾资策会与其国内13家厂商签定“五大中文套装软件”开发计划，而“大五码”即是为“五大中文套装软件”所设计之中文内码。
1986年	国家标准“CNS 11643通用汉字标准交换码”正式版发布，包括13,051个字（删除2个重复字，调整20个字顺序）与441个符号，其余均与试用版相同。
1988年	公布“通用汉字标准交换码”—用户加字区交换码，即第十四字面，共6,148字。
1989年	再公布用户加字区交换码（增编）157字。
1992年	国家标准“CNS 11643中文标准交换码”新版发布，扩充第3-7字面并更换名称，总共包括48,027个字与684个符号（增加部首和数字符号）。
1997年	Big-5E大五码扩充，同1984年版，包括13,053个字与441个符号，另于造字区定义3,954个较常使用的造字。
2002年	国际标准ISO 10646 / Unicode的中文版“CNS 14649广用多八比特编码字符集”，包括中、日、韩、越等20,902个汉字（现已扩充至十万字左右），及全球使用的字符。
2008年	国家标准“CNS 11643中文标准交换码”扩充版发布，增加了户政用字与异体字等。



分布结构

0x8140-0xA0FE	保留给用户自定义字符（造字区）
0xA140-0xA3BF	标点符号、希腊字母及特殊符号， 包括在0xA259-0xA261，安放了九个计量用汉字：尅尅尅尅尅尅尅尅。
0xA3C0-0xA3FE	保留。此区没有开放作造字区用。
0xA440-0xC67E	常用汉字，先按笔划再按部首排序。
0xC6A1-0xC8FE	保留给用户自定义字符（造字区）
0xC940-0xF9D5	次常用汉字，亦是先按笔划再按部首排序。
0xF9D6-0xFEFE	保留给用户自定义字符（造字区）

- “高位字节”使用了0x81-0xFE，“低位字节”使用了0x40-0x7E，及0xA1-0xFE

码表示例1

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
A140		,	\	。	.	·	;	:	?	!	:	,	.	.
A150	·	;	:	?	!		-		—		-	{	—	()	∩
A160	∩	{	}	∩	∩	{	}	∩	∩	【	】	—	—	《	》	≧
A170	≧	<	>	∩	∩	「	」	—	—	『	』	—	—	()	
A1A0	{	}	()	‘	’	“	”	”	”	’	/	#	&	*	
A1B0	※	§	”	○	●	△	▲	◎	☆	★	◇	◆	□	■	▽	▼
A1C0	⊕	%	—	—	—	—	—	—	—	—	—	—	—	#	&	* +
A1D0	—	×	÷	±	√	<	>	=	≡	≡	≠	∞	≡	≡	+	-
A1E0	<	>	=	~	∩	∪	⊥	∠	⊥	△	log	ln	∫	φ	∴	∴
A1F0	♀	♂	⊕	⊙	↑	↓	←	→	↖	↗	↘	↙	↘	//		/



码表示例2

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
A540	世	丕	且	丘	主	乍	乏	乎	以	付	仔	仕	他	仗	代	令
A550	仙	仞	充	兄	冉	冊	冬	凹	出	凸	刊	加	功	包	匆	北
A560	匣	仟	半	卉	卡	占	卯	卮	去	可	古	右	召	叮	叩	叨
A570	呀	司	叵	叫	另	只	史	叱	台	句	叭	叻	四	囚	外	
A5A0		央	失	奴	奶	孕	它	尼	巨	巧	左	市	布	平	幼	弁
A5B0	弘	弗	必	戊	打	扔	扒	扑	斥	旦	朮	本	未	末	札	正
A5C0	母	民	氏	永	汁	汀	汜	犯	玄	玉	瓜	瓦	甘	生	用	甩
A5D0	田	由	甲	申	疋	白	皮	皿	目	矛	矢	石	示	禾	穴	立
A5E0	丞	丟	乒	乓	乜	互	交	亦	亥	仿	伉	伙	伊	佚	伍	伐
A5F0	休	伏	仲	件	任	仰	仇	份	企	伋	光	兕	兆	先	全	



码表示例3

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
BB40	罰	翠	翡	翟	聞	聚	肇	腐	膀	膏	膈	膊	腿	膂	臧	臺
BB50	與	舐	舞	舩	蓉	蒿	蓆	蓄	蒙	蒞	蒲	蒜	蓋	蒸	蓀	蓓
BB60	蒐	蒼	蓑	蓊	蜿	蜜	蜻	蜢	蜥	蜴	蚰	蝕	蝮	蝟	裳	褂
BB70	裴	褰	裸	製	裨	褚	裯	誦	誌	語	誣	認	誠	誓	誤	
BBA0	說	誥	誨	誘	誑	誚	誦	豪	貌	貌	賓	賑	賒	赫	趙	
BBB0	趕	跼	輔	輒	輕	輓	辣	遠	遘	遜	遣	遙	遞	遑	遑	
BBC0	鄙	鄺	鄺	醇	酸	酷	醅	鉸	銀	銅	銘	銖	銘	銓	銜	鉸
BBD0	鉸	銑	閏	閏	閏	閏	閏	隙	障	際	雌	雒	需	鞞	鞞	
BBE0	韶	頗	領	颯	颯	餃	餅	餌	餉	駁	骯	骯	髦	魁	魂	鳴
BBF0	鳶	鳳	麼	鼻	齊	億	儀	僻	僵	價	儂	儂	儉	儉	儉	凜

码表示例4

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
DC40	軹	軺	軻	軼	軽	軾	軼	軼	軼	軼	軼	軼	軼	軼	軼	軼
DC50	鄆	鄆	鄆	鄆	鄆	鄆	鄆	鄆	鄆	鄆	鄆	鄆	鄆	鄆	鄆	鄆
DC60	酖	酖	酖	酖	酖	酖	酖	酖	酖	酖	酖	酖	酖	酖	酖	酖
DC70	鉦	鉦	鉦	鉦	鉦	鉦	鉦	鉦	鉦	鉦	鉦	鉦	鉦	鉦	鉦	鉦
DCA0	隄	隄	隄	隄	隄	隄	隄	隄	隄	隄	隄	隄	隄	隄	隄	隄
DCB0	觜	觜	觜	觜	觜	觜	觜	觜	觜	觜	觜	觜	觜	觜	觜	觜
DCC0	從	從	從	從	從	從	從	從	從	從	從	從	從	從	從	從
DCD0	喝	喝	喝	喝	喝	喝	喝	喝	喝	喝	喝	喝	喝	喝	喝	喝
DCE0	啞	啞	啞	啞	啞	啞	啞	啞	啞	啞	啞	啞	啞	啞	啞	啞
DCF0	望	望	望	望	望	望	望	望	望	望	望	望	望	望	望	望

码表示例5

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
F940	纘	纘	纘	纘	纘	纘	纘	纘	纘	纘	纘	纘	纘	纘	纘	纘
F950	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡
F960	鷄	鷄	鷄	鷄	鷄	鷄	鷄	鷄	鷄	鷄	鷄	鷄	鷄	鷄	鷄	鷄
F970	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡	躡
F9A0		𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩
F9B0	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩
F9C0	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩
F9D0	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩
F9E0	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩
F9F0	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩	𪚩



ISO/IEC 10646

- 一个国际标准编号, 国际标准化组织 (ISO) 1993年正式颁布
- 英文全称: Information technology – Universal Multiple – Octet Coded Character Set, 简称UCS
- 中文全称: 信息技术—通用多八位编码字符集, 亦称大字符集
- 宗旨: 全球所有文种统一编码



Unicode

- 英文Universal Code的缩略语
- 统一编码
- 是对国际标准ISO/IEC 10646编码的一种称谓
- 是一个企业联盟集团的名称, 由美国的HP、Microsoft、IBM、Apple等几家知名的大型计算机企业所组成, 成立该集团的宗旨就是要推进多文种的统一编码
- 就内容而言, Unicode和ISO/IEC 10646是一致的, 并行的



背景

- Apple开创Unicode项目，力求改进Macintosh微机处理多语种文本的体系结构。
- Unicode提供一种统一的字符标识方法，比Macintosh文字体系更有效通用，还为所有文本的显示和编辑减少对文字（即字型）专用软件的依赖性。
- 消除为处理多种字符编码使用的专用系统和应用软件代码，从而加速本土化的进程，并减少了对应用软件和系统软件的测试工作。
- 提供更多的符合排版行业及办公室刊印质量要求的字符。
- 完整性——应具有足以包罗一般性文本交换可能用到的全部字符的字位。
- 效率——由一系列定长字符组成的普通文本易于从语法上分析，即易于确定字符，软件不必保持其状态（maintain state），也不必寻找特殊的换码序列或前后搜索文本。
- 首次发布的Unicode包含世界上所有主要文字的近25,000字符，这对现代通讯是绰绰有余的，其中有由中国、日本、朝鲜及台湾工业标准规定的大约18,000个独特汉字，包括许多经典语言，如希腊、希伯莱、拉丁、巴利、梵文。根据需要还将增加楔形文字、北欧古字、甲骨文之类的古老文字，以及在专门性研究中使用的附加汉字。



CJK-中日韩统一汉字

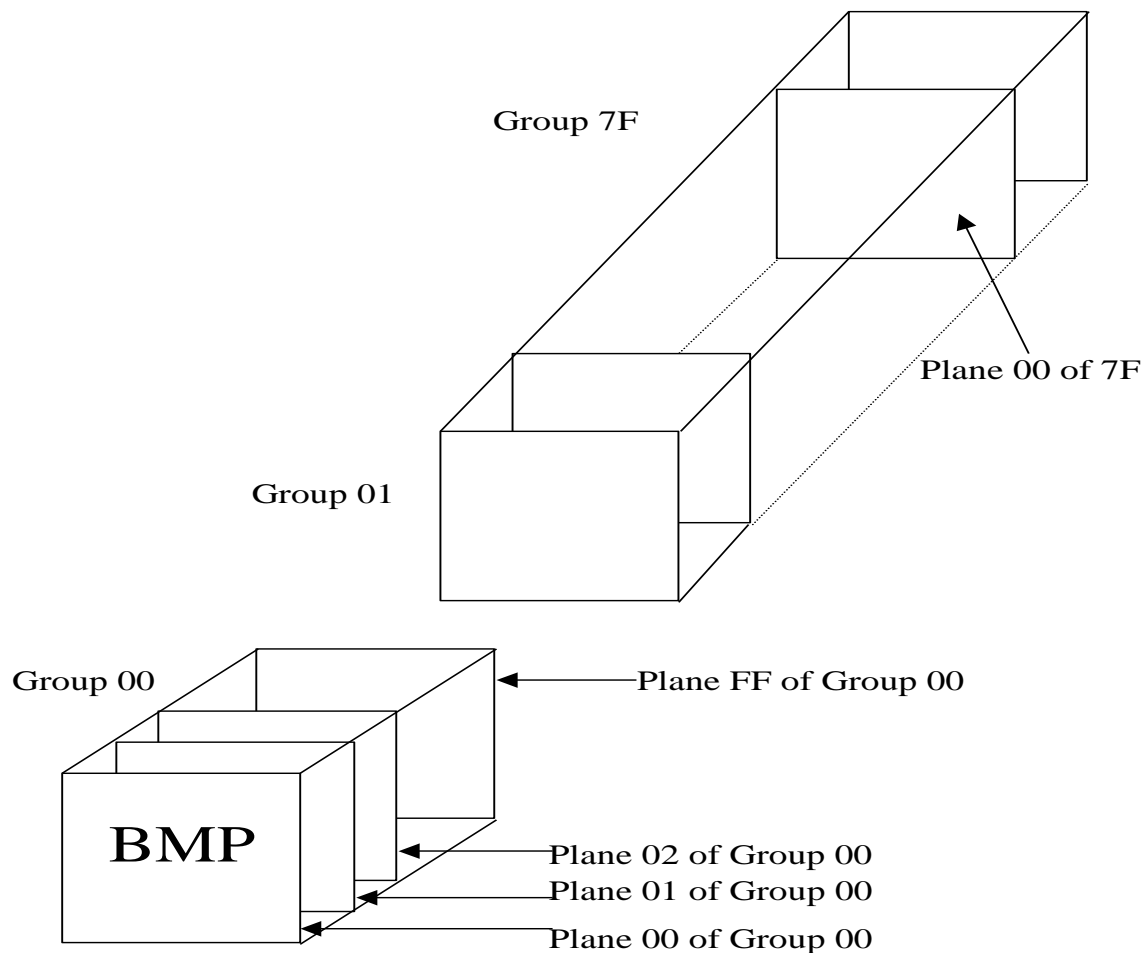
- 把中国、日本与韩国的英文称谓的首字母用于ISO/IEC 10646中的中、日、韩统一编码汉字的简称
- Unihan
- CJKV或许更准确，V代表越南



ISO/IEC 10646 的体系结构

- 四维的编码空间
- 总体上分为128个三维组 (group)，group的值范围是从00到7F
- 每一组包含256个平面(plane)，每一个平面包含256行(row)，每一行包含256个字位(cell)，又称为“列”，plane、row、cell的值范围都是从00到FF全编码
- 整个编码字符集的每个字符都是由4个八位序列表示，(按照组八位、面八位、行八位、列八位的顺序)
- 可编码空间为： $128 \times 256 \times 256 \times 256 = 32K \times 64K$

ISO/IEC 10646体系结构图





基本多文种平面

- 第一个平面（00组中的00平面）称作 Basic Multilingual Plane (基本多文种平面)，简称BMP
- 在其上规定了双八位形式，它可以作为双八位编码字符集使用，即在此平面上仅用行、列两个八位就可以表示一个编码字符



BMP的最新概貌

- A-Zone(00至4D行)：拼音文字编码区,拉丁文、阿拉伯文、日文的平假名及片假名、数学符号等都在此区域编码
- CJK Unified Ideographs, Extension A(3400-4DB5)(6000多码位)
- CJK Unified Ideographs(4E00-9FA5)(20902个编码汉字)
- 韩文 (AC至D7这44行 ($44 \times 256 = 11264$))
- S-ZONE (D8至DF行)for UTF-16
- R-Zone(E0至FF行):限制使用区，一些兼容字符、字符的变形显现形式、特殊字符等均放在此区



ISO/IEC 10646空间分配现状

- 00平面:BMP, 被用于全球现已规范语种的基本文字编码, 编码空间已基本饱和
- 01平面:作为拼音文字辅助平面
- 02平面:作为汉字辅助平面, CJK Extension B即将放入该平面
- E0至FF平面:作为该标准的专用平面来使用
- 其它空间尚未分配



ISO/IEC 10646中CJK汉字组成

- CJK统一编码汉字（20902）
- CJK扩充集A(6585)
- CJK扩充集B(4万--)



ISO/IEC 10646中CJK汉字的来源

Primary Source		
Category	Standard	Number of Source Characters
G0	GB2312-80	6763
G1	GB12345-90	2352
G3	GB7589-87	4835
G5	GB7590-87	2842
G7	General Use Characters for Modern Chinese	42
G8	GB8565-88	290
T1	CNS11643-1986/plane 1	5401
T2	CNS11643-1986/plane 2	7650
Te	CNS11643-1986/plane 14	4198
Jo	JIS X 0208-1990	6356
J1	JIS X 0212-1990	5801
K0	KS C 5601-1987	4620
K1	KS C 5657-1991	2856
Secondary Source		
No.	Standard	Number of Source Characters
1	ANSI Z39.64-1989	13053
2	Big-5 (Taiwan)	13481
3	CCCII, level 1	4808
4	GB 12052-89 (Korean)	94
5	JEF (Fujitsu)	3149
6	PRC Telegraph Code	~8000
7	Taiwan Telegraph Code (CCDC)	9040
8	Xerox Chinese	9776



Unicode编码点的变形

- 编码点(code point)(或编码单元, code element):(1)表示待处理或交换的已编码文本单元的最小位组合(2)代码页或Unicode标准的索引
- 多种不同技术可以二进制格式表示每个Unicode编码点, 以此区分三种不同Unicode编码:UTF-8、UTF-16、UTF-32



什么是UTF?

- Unicode transformation format
- UCS transformation format
- 从Unicode码点到唯一字节序列的映射算法，一一映射，保证无损转换



UTF-16

- Unicode标准的**16位**编码形式
- 为每个字符指定一个**16位**的值
- 编码形式与**ISO/IEC 10646**中的定义形式相同
- 以一个**16位**的值来编码映射到不大于**65535**数值的字符；映射到大于**65535**的数值的字符则被编码成一组**16位**的值（代用对）



UTF-8

- 为满足面向字节、基于**ASCII**码系统的需要而制定
- 用最多达**4**个字节的序列来表示每个字符，为有效分析字符串，用第一个字节指明某个多字节序列中的字节数
- 通常用于数据交换、传输、互联网

Unicode 编码点和 UTF-8 编码字符之间的关系

Unicode 范围	UTF-8 编码的字节
0x00000000-0x0000007F	0xxxxxxx
0x00000080-0x000007FF	110xxxxx 10xxxxxx
0x00000800-0x0000FFFF	1110xxxx 10xxxxxx 10xxxxxx
0x00010000-0x001FFFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx



UTF-32

- 每个字符都表示成一个32位的整数
- 码长相等，便于某些特殊情况的处理
- Unix系统使用



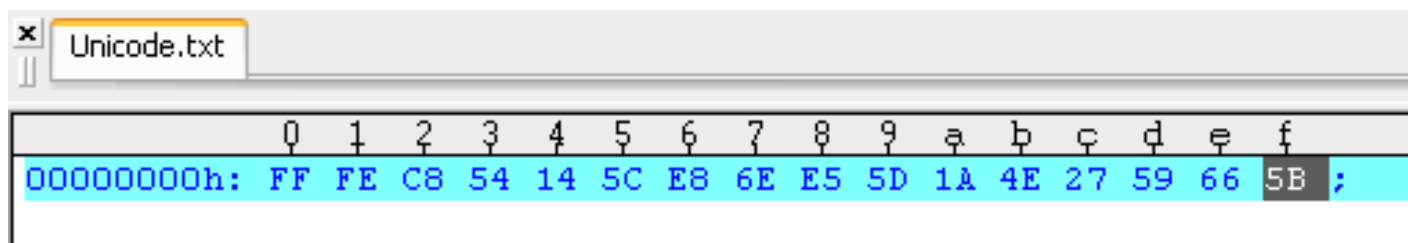
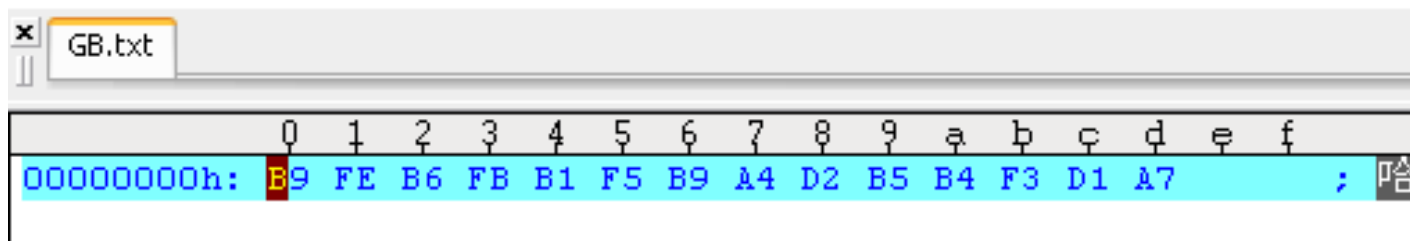
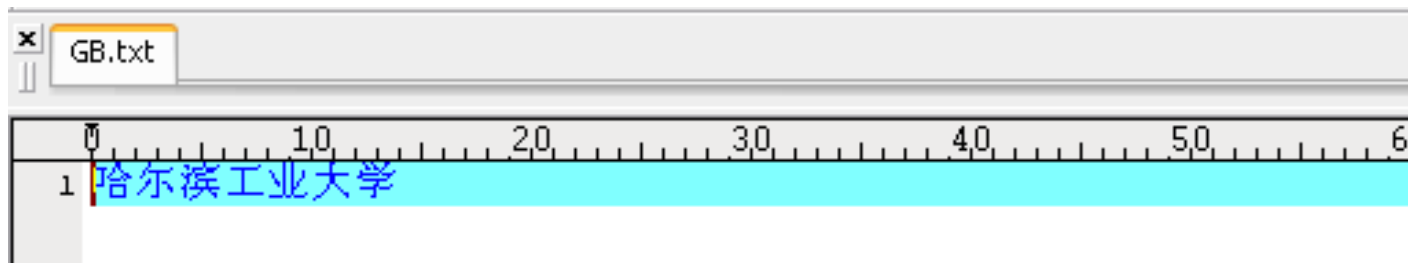
字节顺序标记(BOM)

- 指示处理器怎样把连续的文本放到一个字节序列中
- 权值最低的字节位于开头叫做“little-endian”，权值最高的字节位于开头叫做“big-endian”
- 可用作识别文本文件编码形式的依据

特定编码的字节顺序标记的十六进制表示

编码	编码后的 BOM
UTF-16 big-endian	FE FF
UTF-16 little-endian	FF FE
UTF-8	EF BB BF

字节标记示例





代理对(Surrogate pair)

- ISO/IEC 10646 在BMP定义了一个代理区 (Surrogate Zone) (D800至DFFF)
- 将这个区域平分为前后两个各容纳1024 (1K) 个编码的区域 (D800-DBFF及DC00-DFFF)，分别称作高半代理 (high surrogate) 及低半代理 (low surrogate) 区域
- 从这两个区域分别各取一个编码，分别称为高半代理键 (high surrogate key) 及低半代理键 (low surrogate key)，组合成一个4 bytes代理对 (surrogate pair) 来表示一个编码字符
- 由surrogate机制可对应到一百万个字符 (1024x1024)，分别对应到ISO 10646 中00组的00至0F这16个平面(plane) (其他平面如何处理？)



Windows对Unicode的支持

- Windows 3.1, Windows NT 4, Windows 2000, Windows XP支持Unicode.如果在这些操作系统上运行非Unicode编码程序，在处理之前，操作系统在其内部将应用程序的文本转化为Unicode编码的文本，在把信息传回应用程序之前，操作系统把Unicode编码的文本转化回所希望的代码页编码形式。
- Windows 95, Windows 98, Windows Me不是基于Unicode的，它们只提供了基于Windows NT的Windows版本所提供的Unicode支持的一个子集



创建Win32 Unicode应用程序

- WCHAR，一种16位的数据类型
用于8位(ANSI)和双字节字符：

```
typedef char CHAR;
```

```
typedef CHAR TCHAR;
```

用于Unicode(宽)字符：

```
typedef unsigned short WCHAR;
```

```
typedef WCHAR TCHAR;
```



创建Win32 Unicode应用程序

- Win32 API的W函数原型

```
//windows.h
```

```
#ifndef UNICODE
```

```
#define SetWindowText SetWindowTextW
```

```
#else
```

```
#define SetWindowText SetWindowTextA
```

```
#endif //UNICODE
```



创建Win32 Unicode应用程序

- Unicode文本宏

```
LPWSTR str = L"This is a Unicode string";
```

```
.....
```

```
#ifdef UNICODE
```

```
#define TEXT(string) L#string
```

```
#else
```

```
#define TEXT(string) string
```

```
#endif //UNICODE
```

```
.....
```

```
LPWSTR str = TEXT("This is a Unicode string");
```



创建Win32 Unicode应用程序

- C运行库扩展

处理字符串的C运行库函数举例

通用CRT

8位字符集

Unicode

_tcscpy

strcpy

wcscpy

_tcscmp

strcmp

wcscmp

等价的Win32 API函数

通用Win32

8位字符集

Unicode

lstrcpy

lstrcpyA

lstrcpyB

lstrcmp

lstrcmpA

lstrcmpB



ISO 10646/Unicode的实现及其重要意义

- 在全球范围内建立起实时、无障碍的信息交换模式
- 推动了汉字典籍的数字化
- 为数字化图书馆的建立铺平了道路
- 为弘扬汉字文化提供了舞台
- Single Binary技术的诞生：同一套基本程序用于多个语言环境的技术
- 使汉字关联活起来：正-异关联、中-日关联、繁-简关联，正-讹关联以及古今、新旧字形关联



关联实例

■ 正-异关联

- 獠 VS. 龙(杂毛狗)
- 稃(指所有草本植物包括水稻、玉米、小麦、草籽的皮屑) VS. 麸(仅指小麦的皮屑)

■ 中-日关联

B. 与汉语在字形上大同小异

例如：惡魔 压迫 衣類 黑板 差別 加減 過密
吸收 等等。



内码自动识别与转换

- 内码自动识别
- 内码相互转换
- 简繁转换



问题

- 如何识别内码？
- 哪些资源可以利用？



内码自动识别

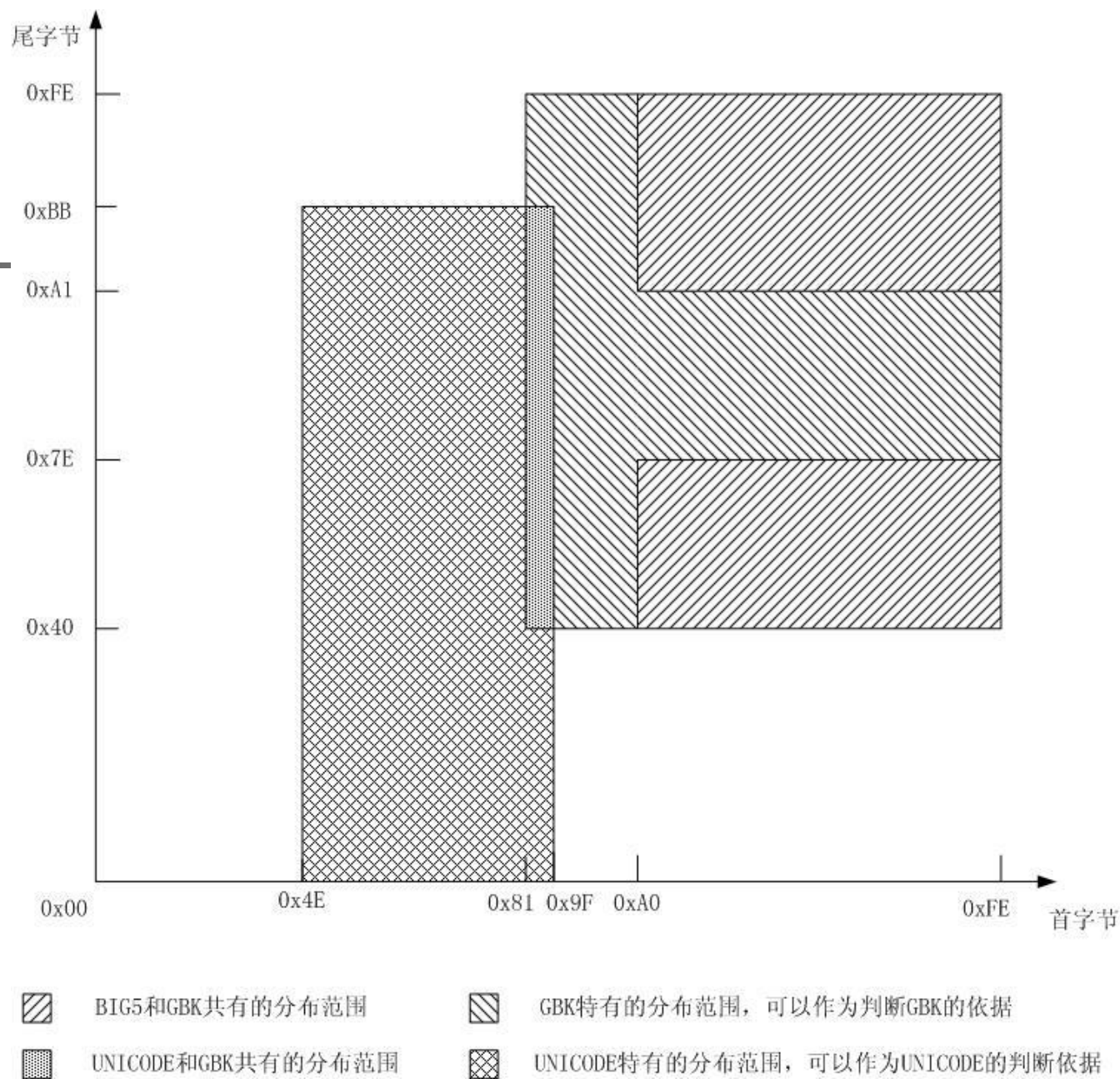
- 基于内码分布特征的识别算法
- 基于常用字内码分布特征的识别算法
- 基于标点符号特征的识别算法
- 基于字频特征的识别算法
- 基于语义特征的识别算法
- 几种算法的比较



基于内码分布特征的识别算法

- 几种常见的双字节中文内码，其分布范围不完全相同，所以在一段文本中，所有字符的分布范围可以作为识别内码的依据。
- 1. **BIG5**分布范围最小，完全被**GBK**包括，所以出现**BIG5**分布范围以外字符的时候可以判定该文本不是**BIG5**内码的；
- 2. 其次首字节在**0x4E**到**0x80**，第二字节在**0x00**到**0x39**的字符均为**UNICODE**特有的字符，可以作为**UNICODE**的判定标准。
- 此算法适于作为内码识别的初步筛选。

三种内码分布范围





算法实现

首先设置一个位置计数器 i ，用来记录当前正在处理的文本的位置。再设置四个计数器 C_g ， C_b ， C_u ， C_a ，分别用来统计文本中 GBK、BIG5、Unicode、ASCII 字符出现的次数， $Length$ 为整个文本的长度， $Word$ 为一双字节临时变量， B_i 表示文本中第 i 个字节。算法如下：

- (1) 设置初值：位置计数器 $i=0$ 。计数器 C_g ， C_b ， C_u ， C_a 均为 0；
- (2) 如果 $i+1>Length$ ，说明剩下的文本不够两个字节，结束，否则 $Word=B_iB_{i+1}$ ；
- (3) 如果 $Word \in \text{GBK}$ ，则 C_g 加 1，如果 $Word \in \text{BIG5}$ ，则 C_b 加 1，如果 $Word \in \text{Unicode}$ ，则 C_u 加 1。然后 i 加 2，跳转到(2)；
- (4) 如果 $B_i \in \text{ASCII}$ ，则 C_a 加 1， i 加 1，跳转到(2)。

通过上面的算法对字符串进行扫描后得到三个计数器的值 C_g ， C_b ， C_u 。如果其中只有一个的值等于 $(Length-C_a)/2$ ，则该内码即为文本的内码，否则文本的内码无法识别。



基于常用字内码分布特征的识别算法

- 虽然几种内码的编码范围有重合，但是在某些范围的编码在一种内码中是常用汉字，而在另外一种内码中却没有定义或者不是常用汉字。
- 统计一段文本中这些特定编码范围中的字符出现的频率也可以作为区别不同内码的依据。



基于常用字内码分布特征的识别算法

- 首字节**0xA4**到**0xA9**的字符在**GBK**中为日文假名、希腊字母、俄文字母和制表符，正常文本中很少出现，
- 此外首字节为**0xAA**到**0xAF**，第二字节为**0xA1**到**0xFE**的编码位则根本没有定义，但这个范围却是**BIG5**码的常用汉字
- 所以如果文本中频繁出现这个范围（特别是后一种情况）的字符，就可以认为是**BIG5**码。



基于常用字内码分布特征的识别算法

- 首字节在**0xC6**到**0xD7**，第二字节在**0xA1**到**0xFE**的字符在**GBK**中属于一级字库，是常用汉字，
- 而在**BIG5**中，首字节在**0xC6**到**0xC7**间的编码位没有明确定义，但通常用来放日文假名和序号，
- 首字节在**0xC8**到**0xD7**的字符则属于罕用汉字区。
- 所以如果这个范围的字符出现较多，则可以判定为**GBK码**。



算法实现

设置两个计数器 C_1 , C_2 , 分别记录两个编码范围中字符出现的次数, i 为当前位置, $Length$ 为源文本的长度, B_i 表示文本中第 i 个字节。

- (1) 设置初值: $i=0$, $C_1=0$, $C_2=0$ 。
- (2) 如果 $i+1>Length$, 结束;
- (3) 如果 $B_i \in ASCII$, 则 i 加 1, 跳转到(2);
- (4) 如果 $B_i > 0xA4$, 并且 $B_i < 0xA9$, 则 C_1 加 1, i 加 2; 如果 $B_i > 0xC6$, $B_i < 0xD7$, $B_{i+1} > 0xA1$, $B_{i+1} < 0xFE$, 则 C_2 加 1, i 加 2, 跳转到(2);

完成以上过程之后, 判断 $C_1/Length$, $C_2/Length$ 和预先设定好的阈值即可判断出该段文本是哪种内码的。



基于标点符号特征的识别算法

- 标点符号是中文的一个重要特征，一段文本通常都会包含标点符号，而且标点符号出现的频率也较高。
- 统计结果表明，单个中文字符出现频率中，“，”的出现频率最高，“的”排在第二，而“。”和“、”分别排在第三和第四位，此外，双引号排在第十二位，冒号排在第三十三位。
- 由于标点符号出现频率极高，而各种内码中这些标点的编码又不相同，所以文本达到一定长度的情况下，识别率可以达到很高。
- 据此，可以统计一段文本中各种内码的标点符号出现的频率来判断所使用的内码。



算法实现

首先构造三张表，分别为 GBK，BIG5 和 Unicode 的标点符号表，分别为 S_g ， S_b ， S_u ，由于这三种内码的标点符号分布没有交集，也为识别工作带来了便利。↵

设置三个计数器 C_g ， C_b ， C_u ，分别记录三种内码的标点符号出现的次数， i 为当前位置， $Length$ 为源文本的长度， B_i 表示文本中第 i 个字节， $Word$ 为一双字节临时变量。↵

- (1) 设置初值 $C_g=0$ ， $C_b=0$ ， $C_u=0$ ， $i=0$ ；↵
- (2) 如果 $i+1>Length$ ，结束，否则 $Word=B_iB_{i+1}$ ；↵
- (3) 如果 $B_i \in ASCII$ ， i 加 1，跳转到(2)；↵
- (4) 如果 $Word \in S_g$ ，则 C_g 加 1，如果 $Word \in S_b$ ，则 C_b 加 1，如果 $Word \in S_u$ ，则 C_u 加 1。然后 i 加 2，跳转到(2)。↵

经过上述过程之后， C_g ， C_b ， C_u ，中，值最大的，且和 $Length$ 的比值超过阈值的那种内码就是文本的内码。↵



基于字频特征的识别算法

- 汉字的字频指的是汉字的相对使用频度，是汉字使用情况的一种反映。
- 虽然几种内码中一级字库的编码范围有重叠，但出现频率较高的那些字符的编码并不相同。
- 所以通常情况下，一段文本经过处理后，哪种内码的常用字出现的频率越高，哪种内码就越有可能是该段文本的内码，而且文本越长，效果就越好。
- 根据这种特征来识别位置文本的内码是可行的。



算法实现

首先要构造三张表 F_g , F_b , F_u , 分别用 GBK, BIG5, Unicode 三种内码存储大约 1500 个最常用的汉字, 并且按照使用频度由高到低排序 (大陆, 香港, 台湾前 1500 个常用字至少 90% 是一致的, 所以此处均使用人民日报统计出的字频信息)。

设置三个计数器 C_g , C_b , C_u , 分别记录三种内码的最常用汉字出现的次数, i 为当前位置, $Length$ 为源文本的长度, B_i 表示文本中第 i 个字节, $Word$ 为一双字节临时变量。

- (1) 设置初值 $C_g=0$, $C_b=0$, $C_u=0$, $i=0$;
- (2) 如果 $i+1>Length$, 结束, 否则 $Word=B_iB_{i+1}$;
- (3) 如果 $B_i \in ASCII$, i 加 1, 跳转到(2);
- (4) 如果 $Word \in F_g$, 则 C_g 加 1, 如果 $Word \in F_b$, 则 C_b 加 1, 如果 $Word \in F_u$, 则 C_u 加 1。然后 i 加 2, 跳转到(2)。

经过上述过程以后, 可以得到输入的文本中, 三种内码的高频字出现的总次数, 可以比较三个计数器的值和总长度 $Length$ 的比值, 如果超过事先设定好的阈值并且是三个中最大的, 则该内码即为文本的内码。



基于语义特征的识别算法

- 以上四种识别算法都需要文本达到一定的长度才能得到较好结果，而对于很短且没有标点符号的文本则难以识别。
- 构成表意的段落、句子、词语的汉字之间是有一定的关联性的，这种关联性可以作为内码识别的依据。
- 但是要构造所有汉字之间的关联性不太实际，而且没有必要，在此仅需要语义的最低级别——词语，就能满足内码识别的要求。



基于语义特征的识别算法

- 由于各种内码中字符的分布不一致，一种内码中的词语在其他内码中就变成两个不相关的字（仅极少数可能还是词语）。
- 所以，根据事先存储好的词语表就能够完成内码的识别。
- 从大量文本资料中统计出常见的词语，并按照出现的次数从高到低的顺序，大约取前**4000**条词语，作成词语表。
- 统计出目标文本中相应内码的词语出现的数量，词语出现次数超过一定比例则可认为是该种内码。



算法实现

首先收集并整理出最常用的词语大约 4000 条，按照字母顺序由低到高排序（查找的时候使用折半查找，尽量提高效率），并且分别用 GBK, BIG5, Unicode 三种内码存储。表的结构为：第一列存储该词语的字数，后面跟着完整的词语，表名为 W_g , W_b , W_u 。

设置三个计数器 C_g , C_b , C_u ，分别记录三种内码的常用词语出现的次数， i 为当前位置， $Length$ 为源文本的长度， B_i 表示文本中第 i 个字节， $Word_1$, $Word_2$, $Word_3$, $Word_4$ 为四个双字节的临时变量。

- (1) 设置初值 $C_g=0$, $C_b=0$, $C_u=0$, $i=0$;
- (2) 如果 $i+3>Length$ ，说明剩下的不够两个字，即不够组成一个最短的词语，结束，否则 $Word_1=B_iB_{i+1}$, $Word_2=B_{i+2}B_{i+3}$;
- (3) 如果 $Word_1 \in W_g$ 的首字集，则根据 W_g 中词语的字数继续向后判断，如果不是词语，则 i 加 2，跳转到(2)。否则 C_g 加一， i 加词语字数乘以 2；
如果 $Word_1 \in W_b$ 的首字集，则根据 W_b 中词语的字数继续向后判断，如果不是词语，则 i 加 2，跳转到(2)。否则 C_b 加一， i 加词语字数乘以 2；
如果 $Word_1 \in W_u$ 的首字集，则根据 W_u 中词语的字数继续向后判断，如果不是词语，则 i 加 2，跳转到(2)。否则 C_u 加一， i 加词语字数乘以 2；跳转到(2)。

经过上述过程以后，得到输入的文本中三种内码常用词语出现的次数。比较三个计数器的值，最大的即为输入文本的内码。



几种算法的比较

- 基于内码分布特征、基于常见字内码分布特征的识别算法是根据中文内码本身的特点进行识别的，不需要考虑文本的内容，而仅仅需要统计各个字符出现的次数即可，不需要额外的数据。
- 基于标点符号特征、基于字频特征以及基于词频特征的识别算法比较复杂，不仅要统计各个字符出现的次数，还要涉及到文本的内容，需要维护额外的数据，而这些数据是识别成功的关键。识别过程中有大量的判断、查表以及处理操作，效率会比较低。



内码相互转换

- **UNICODE和UTF-8码间的转换通过计算进行**
- **其他内码转换的方法基本上都是基于查找对照表实现**



GBK, BIG5和UNICODE间的转换

- 主要是通过查找对应的码表来实现
- GBK和BIG5之间的转换以UNICODE为中转，即先转换为UNICODE，再转换为另外一种内码
- GBK和Unicode之间是一一对应的关系，直接使用Unicode官方网站提供的对应码表即可
- BIG5只包含繁体字符集，只对应UNICODE的一部分，在UNICODE转回BIG5的时候会出现无对应字符的情况，所以UNICODE转换成BIG5之前要先进行简繁转换，将大字符集转换为繁体字符集



UNICODE-GBK对应表示例

UxA47A	0x5E72	#CJK UNIFIED IDEOGRAPHH
0xA47B	0x5EFE	#CJK UNIFIED IDEOGRAPHH
0xA47C	0x5F0B	#CJK UNIFIED IDEOGRAPHH
0xA47D	0x5F13	#CJK UNIFIED IDEOGRAPHH
0xA47E	0x624D	#CJK UNIFIED IDEOGRAPHH
0xA4A1	0x4E11	#CJK UNIFIED IDEOGRAPHH
0xA4A2	0x4E10	#CJK UNIFIED IDEOGRAPHH
0xA4A3	0x4E0D	#CJK UNIFIED IDEOGRAPHH
0xA4A4	0x4E2D	#CJK UNIFIED IDEOGRAPHH
0xA4A5	0x4E30	#CJK UNIFIED IDEOGRAPHH
0xA4A6	0x4E39	#CJK UNIFIED IDEOGRAPHH
0xA4A7	0x4E4B	#CJK UNIFIED IDEOGRAPHH
0xA4A8	0x5C39	#CJK UNIFIED IDEOGRAPHH
0xA4A9	0x4E88	#CJK UNIFIED IDEOGRAPHH
0xA4AA	0x4E91	#CJK UNIFIED IDEOGRAPHH
0xA4AB	0x4E95	#CJK UNIFIED IDEOGRAPHH
0xA4AC	0x4E92	#CJK UNIFIED IDEOGRAPHH

UNICODE和UTF-8的转换

■ UTF-8的编码规则

Unicode 编码(十六进制)	UTF-8 编码（二进制）
U00000000-U0000007F	0xxxxxxx
U00000080-U000007FF	110xxxxx10xxxxxx
U00000800-U0000FFFF	1110xxxx10xxxxxx10xxxxxx
U00010000-U001FFFFF	11110xxx10xxxxxx10xxxxxx10xxxxxx
U00200000-U03FFFFFF	111110xx10xxxxxx10xxxxxx10xxxxxx10xxxxxx
U04000000-U7FFFFFFF	1111110x10xxxxxx10xxxxxx10xxxxxx10xxxxxx10xxxxxx



UNICODE到UTF-8

- 首先要判断UNICODE字符值的范围
- 当UNICODE的值在0x0000到0x007F之间时，说明这是一个ASCII字符，目标UTF-8字符只包含一个字节，直接赋值即可；
- 当UNICODE值在0x0080到0x07FF之间时，目标UTF-8应该是两个字节，且第一个字节前两位为1，第三位为0，第二个字节第一位为1，第二位为0，然后将UNICODE的内容，按照从右到左的顺序，从UTF-8最右一位0开始一次填充进入空位，便构成了UTF-8字符；
- 当UNICODE值在0x0800到0xFFFF之间时，对应的UTF-8字符应该包含3个字节，且第一个字节的前三位为1，第四位为0，后两个字节的最高位为1，第二位为0，然后从右到左的顺序，从UTF-8最右一个空位开始依次填充进入空位，便构成了UTF-8字符。



UNICODE转换为UTF-8的规则

UNICODE 编码 (十六进制)	UTF-8 编码 (二进制)
0x0000-0x007F	0xxxxxxx
0x0080-0x07FF	110xxxxx 10xxxxxx
0x0800-0xFFFF	1110xxxx 10xxxxxx 10xxxxxx



UTF-8到UNICODE

- 首先要判断UTF-8字符的第一个字节的范围；
- 如果首字节小于0x7F，则直接赋值即可；
- 如果UTF-8首字节大于等于0xC0，小于等于0xDF，则说明该UTF-8包含两个字节，然后取出低字节的低6位和高字节的低2位，组成UNICODE的低字节，再取出高字节的4-6位作为UNICODE的高字节，组合成一个UNICODE字符即可；
- 如果UTF-8首字节大于等于0xE0，小于等于0xEF，则说明包含3个字节，取出低字节的低6位和中间字节的低2位组成UNICODE的低字节，再取出中间字节的3-6位和高字节的低4位，组成UNICODE的高字节，再组合成一个UNICODE字符即可。下表说明了转换规则。



UTF-8转换为UNICODE的规则

UTF-8 首字节编码	UNICODE 编码(二进制)
0x00-0x7F	直接赋值, 即 00000000 xxxxxxxx
0xC0-0xDF	设 UTF-8 为 110xxxxx 10yyyyyy, 则 UNICODE 为 00000xxx,xyyyyyyy
0xE0-0xEF	设 UTF-8 为 1110xxxx 10yyyyyy 10zzzzzz, 则 UNICODE 为 xxxxyyyy,yyzzzzzz



■ 简繁转换涉及哪些因素？



简繁转换

- 汉字简化知识
- 非对称简繁字的处理
- 转换结果
- 简繁字词汇差异



汉字简化知识

- 现行的简体中文以国家语言文字工作委员会于**1986**年公布的《简化字总表》（和**1964**年发布的相同）为标准，共分三表，第一表收录**352**个不做偏旁使用的简体字，第二表收录**132**个可做偏旁的简体字及**14**个简化偏旁，第三表收录应用第二表的简化字和简化偏旁作为偏旁得出的简化字共**1754**个。
- 新加坡**1976**年公布的《简体字总表》，和中国大陆的标准完全一致。
- 汉字简化的原则是“述而不作”，即尽量使用现有的民间广泛流行的简化字，而不创造新字。



简化的方法

- 采用笔画简单的古字。如“从〔從〕”、“众〔衆〕”、“礼〔禮〕”、“无〔無〕”、“尘〔塵〕”、“云〔雲〕”等等，这些字都见于《说文解字》。
- 草书楷化。如“专〔專〕”、“东〔東〕”、“汤〔湯〕”、“乐〔樂〕”、“当〔當〕”、“买〔買〕”、“农〔農〕”、“孙〔孫〕”、“为〔為〕”等。
- 用简单的符号代替复杂的偏旁。如“鸡〔鷄〕”、“观〔觀〕”、“戏〔戲〕”、“邓〔鄧〕”、“区〔區〕”、“岁〔歲〕”、“罗〔羅〕”、“刘〔劉〕”、“齐〔齊〕”等。



简化的方法-续

- 仅保留原字的有特征的部份。如“声〔聲〕”、“习〔習〕”、“县〔縣〕”、“医〔醫〕”、“务〔務〕”、“广〔廣〕”、“条〔條〕”、“凿〔鑿〕”等。
- 原来的形声字改换简单的声旁。如“辽〔遼〕”、“迁〔遷〕”、“邮〔郵〕”、“阶〔階〕”、“运〔運〕”、“远〔遠〕”、“扰〔擾〕”、“犹〔猶〕”、“惊〔驚〕”、“护〔護〕”等。
- 保留原字轮廓。比如“龟〔龜〕”、“虑〔慮〕”、“爱〔愛〕”等。



简化的方法-续

- 在不引起混淆的情况下，同音字合并为简单的那个字。比如“里程”的“里”和“裏面”的“裏”合并，“面孔”的“面”和“麵條”的“麵”合并，“皇后”的“后”和“以後”的“後”合并，“忧郁”的“郁”和“鬱鬱葱葱”的“鬱”合并。
- 第七种方法会导致简体和繁体不一一对应的现象。那么在简繁转换的过程中，遇到这样的情况该怎么处理呢？



非对称简繁字的处理

- 汉字在简化的过程中，将一些同音的繁体字简化为一个简体字，虽然在词语中并没有引起歧义，但是造成了一个简体字对应多个繁体字的状况，这些简体字和繁体字被称为非对称简繁字。
- 非对称简繁字需要专门处理。



非对称简繁字简介

- 同音兼并类：即将笔画繁的字的意义加载到与之同音的笔画简的字上，从而省掉一个繁体字。如“老閻”和“木板”。
- 非同音（调）兼并类：即将笔画繁的字的意义加载到与之不同音的笔画简的字上，从而省掉一个繁体字。如“告别”和“瞥扭”。
- 统一简化类：即将几个繁体统一简化为一个简体，从而用一个简体代替多个繁体。如“發财”和“头髮”。
- 部分简化类：即只将部分意义加载到某个笔画稍简的字上，该字及其他意义仍然保留。如“夥”在“伙计”时简化，而表示多的含义的时候不简化。
- 多头简化类：即在不同音义上作不同的简化，分别与不同简体对应。



礙航襖壩閘辦幫寶報幣斃標錶警當補緝竊燦層攙謾饒
碍航袂坝板办帮宝报币毙标表别卜补才蚕灿层挽讒饒



具体处理方法

- 首先根据《简化字总表》构造出简繁对照表，其中的汉字均使用**UNICODE**存储（由于整个系统内码转换均以**UNICODE**为中转，所以简繁转换也仅在**UNICODE**中实现）。
- 然后收集对应多个繁体的简体字和对应多个简体的繁体字，将简繁对照表中这些字的位置替换为非对称关系的标志。转换过程中遇到这样的标志，则转到非对称简繁表进行处理。
- 非对称简繁表中存储这些字（不对称的简体字和繁体字）构成的词语，以及在这个词语中应该对应哪个繁体字或简体字。每条记录的存储结构为（要处理的字，字在词中的位置，词语，应该对应的字），这张表也使用**UNICODE**存储。
- 这种方式的简繁转换，正确率很大程度上依赖于非对称简繁表中词语的数量，可以通过不断向表中添加新的词语来提高正确率。



转换结果

- 目前系统中的非对称简繁表中共收录词语**691**条，规则是根据商务印书馆的《现代汉语词典》。
- 经验证，对非对称现象能够较好地处理，如“台湾昨天刮台风，把我的台灯吹飞了”被转换为“臺灣昨天颶颶風，把我的檯燈吹飛了”，句中的三个“台”都被正确地转换为对应的繁体字。



简繁体词汇差异

简体和繁体间词汇的不同

简体用词	繁体用词
位	位元
字节	位元組
光盘	光碟
计算机	電腦
数据库	資料庫
文件	檔案
信息	資訊
因特网	網際網路
软件	軟體
星期	禮拜

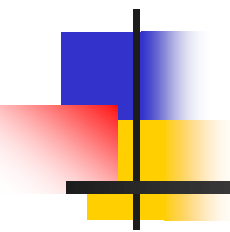
简体繁体间专有名词的不同

英语	简体	繁体
Berlin Wall	柏林墙	柏林圍牆
Chad	乍得	查德
Georgia	佐治亚	喬治亞
Kennedy	肯尼迪	甘迺迪
Wisconsin	威士康星	威士康辛



大陆台湾词汇差异最新例子

- 柬埔寨-高棉
- 老挝-寮国
- 嘎纳-成坎城
- 悉尼-雪梨
- 佛罗伦萨-翡冷翠
- 橙子-柳丁
- 菠萝-凤梨
- 番石榴-芭乐
- 花生-地豆
- 情报-情治
- 大陆妹-生菜
- 穷人-待富者
- 自助餐-吃到饱
- 创可贴-OK绑
- 抓狂-俩共
- 红心大战-伤心小栈
- 蛋卷冰激凌-吧噗
- 沙发-头香



谢谢！
