中文信息处理
实验分析（一）

苏州大学计算机科学与技术学院

主要内容

- 本课程实验作用
- 实验一分析
- 实验二分析
- 实验三分析
- 实验四分析

实验作用

- 加深理解课程知识点
- 为国际化软件开发创造基础
  - 简体版
  - 繁体版
  - 英文版
  - 轻松跨越网页设计“乱码”问题
- 将《程序设计》和《数据结构》等课程融会贯通、综合应用
- ……

实验一

- 汉字在计算机内的存储及文件操作
  - 西文
    - ASCII
    - ISO-8859
  - 中文
    - 中国大陆地区 GB2312-80 6763汉字
    - 中国台湾香港 BIG-5
  - GBK
  - Unicode
    - BMP平面

GB2312-80 例子

01区	1	2	3	4	5	6	7	8	9
0		、	。	•	ˉ	ˇ	¨	〃	々
1	—	～	‖	…	‘	’	“	”	〔 〕
2	〈 〉	《 》	「 」	『 』	〖 〗				
3	【 】	±	×	÷	∶	∧	∨	∑	∏
4	∪	∩	∈	∷	√	⊥	//	∠	⌒	⊙
5	∫	∮	≡	≌	≈	∽	∝	≠	≮	≯
6	≤	≥	∞	∵	∴	♂	♀	°	′	″
7	℃	$	¤	¢	£	‰	§	№	☆	★
8	○	●	◎	◇	◆	□	■	△	▲	※
9	→	←	↑	↓	〓				

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	俺
2	按	暗	岸	胺	案	肮	昂	盎	凹	敖
3	熬	翱	袄	傲	奥	懊	澳	芭	捌	扒
4	叭	吧	笆	八	疤	巴	拔	跋	靶	把
5	耙	坝	霸	罢	爸	白	柏	百	摆	佰
6	败	拜	稗	斑	班	搬	扳	般	颁	板
7	版	扮	拌	伴	瓣	半	办	绊	邦	帮
8	梆	榜	膀	绑	棒	磅	蚌	镑	傍	谤
9	苞	胞	包	褒	剥					

- 同一字符在不同字符编码集中可能编码不同
- 同一编码在不同字符编码集中可能对应不同字符
- 不同字符集的字符集合大小不一
- 全球统一将大大有利于交互
- 文件
  - 文本文件
  - 二进制文件
  - 文本文件也可以用二进制的方式打开

- 程序变量名、数据库字段名尽量少用中文

网页字符编码

苏州大学

搜索

校内登录 网关登录 通知发文 图书馆 [繁體]

首页 | 苏大新闻 | 学校概况 | 院部设置 | 师资队伍 | 教育教学 | 科学研究 | 招生就业 | 合作交流 | 苏大经纬

2013年7月17日 星期三 多云转晴 29℃~37℃

我校本一批次录取 投档线文科342分、理科353分

7月14日，江苏省高招本一批次院校开始录取。今年我校本一投档线为文科342分、理科353分，两门选测科目等级为AB，文理科投档线比省控线分别高出了14分和15分，比去年有所提高。

作为江苏省普通类本一批考生招生人数最多的高校，今年我校在招生人数方面，本一批次在江苏计划招生文科676名，理科1872名。从目前的招生情况来看，今年最为热门的专业依然是医学类和财经类专业。由于今年我校本一批次录取生源质量普遍较好，为了确保考生的入学机会，对凡是填报苏大并服从志愿的进档考生，我校将通过增加计划的方式录取进校。

（新闻中心）

网页字符编码（续）


搜索
首页
苏大新闻
学校概况
院部设置
师资队伍
教育教学
科学研究
招生就业
合作交流
苏大经纬
居苺掛玲蠶棒翹　茫紫盅悖褪342煦〉燴褪353煦
7堎14　ざ蔬劼吽詢桸掛玲蠶棒垽苺羲実翹　（踏爛唐苺掛玲茫紫盅峈悖褪342煦〉燴褪353煦ざ謗藷恁聆褪醴脹撰峈ABざ悖燴褪茫紫盅掀吽諷盅煦梗詢堤賸14煦審15煦ざ掀　爛岫垀桓詢（
釬峈蔬劼吽　籿濬掛玲蠶蕉汜桸汜　杼鄋嗣腔詢苺ざ踏爛唐苺斐桸汜　杼源醱ざ掛玲蠶棒斐蔬劼數赫桸汜悖褪676靡ざ燴褪1872靡（植醴　腔桸汜　錶懂艘ざ踏爛鄋峈　藷腔蚳琯牲　岖颱悝濬審笙冪濬蚳琯（蚤衾踏爛唐苺掛玲蠶棒翹　汜埭窒講　梢誕疑ざ峈賸　悵蕉汜腔　悝儂頗ざ勤歇岖沓惆劼湮甜督植袾堋腔輛紫蕉汜ざ唐苺蔚籿徹峭樓數赫腔源宅翹　輛苺（
(陔恬箕陋)

实验一程序分析

- 编写程序让用户输入GB2312汉字，计算该汉字的区位码
- 分析
  - 可以用两个字节分别存储汉字的区号和位号
  - 为了保证不与ASCII的控制符号冲突，将区号和位号都加上32，得到的编码是国标码
  - 区位码和国标码都不能直接作为机内码，因为一个汉字可能会被误解成两个西文
  - 将国标码的两个字节的首位强行置成1，就可以避免和西文混淆，该编码是GB2312机内码

转换关系

- 国标码= 区位码两字节分别加上20H
- 机内码= 国标码两字节分别加上80H
- 机内码= 区位码两字节分别加上A0H

- 区位码通常用10进制表示
- 国标码和机内码通常用16进制表示

实验二

- Unicode和GBK汉字机内码的转换
- Unicode->GBK
- GBK ->Unicode

- 数据结构的设计
  - 节约存储
  - 检索快速
- 具体的实现方法

ISO10646/Unicode内码

- 汉字内码范围：4E00H-9FFFH
- 在此内码范围中的码位共有:
- （9f-4e+1)*100H=20992个
- 其中9FA6H-9FFFH,共90个是空位
- 所以，20902个汉字的内码范围是：
- 4E00H-9FA5H

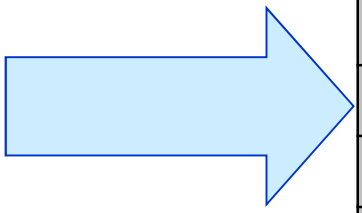
Unicode文本文件

- 文件内容显示为：“一啊”
- 文件内码为：
- FF FE 00 4E 4A 55 0D 00 0A 00
- 其中，“一”的Unicode内码是4E00
- “啊”的Unicode内码是554A
- 所以，Unicode是以word为单位的汉字编码
- Unicode的文本文件是以FF FE开头的

汉字序号对应关系

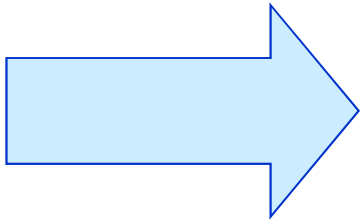
- 假设在Unicode集合中，所有的汉字按照内码排序进行存放的话，那么可以在汉字内码和序号间建立一一对应的关系
- 如汉字内码为i,则它的序号为：i-4e00h
  - 4e00h是第0个
  - 9FA5是第20901个

序号	Unicode内码
0	4E00
1	4E01
2	4E02
3	4E03
4	4E04
5	4E05
…	…
i	4E00+i
…	…
20898	9FA2
20899	9FA3
20900	9FA4
20901	9FA5

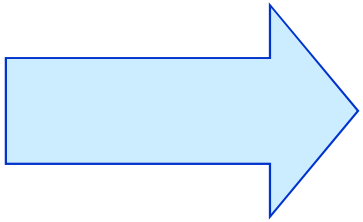


序号	Unicode内码	GBK内码
0	4E00	D2BB
1	4E01	B6A1
2	4E02	8140
3	4E03	C6DF
4	4E04	8141
5	4E05	8142
…	…	….
i	4E00+i	….
…	…	….
20898	9FA2	FD98
20899	9FA3	FD99
20900	9FA4	FD9A
20901	9FA5	FD9B

Unicode->GBK



序号	GBK内码
0	D2BB
1	B6A1
2	8140
3	C6DF
4	8141
5	8142
…	….
i	….
…	….
20898	FD98
20899	FD99
20900	FD9A
20901	FD9B



GBK内码
D2BB
B6A1
8140
C6DF
8141
8142
….
….
….
FD98
FD99
FD9A
FD9B

GBK内码

- 内码范围从8140H-FEFEH
  - 除了xx7F一条线
- 汉字内码为GBK/2、GBK/3和GBK/4，共21008个码位，其中有5个空的码位：
- D7 FA / D7 FB /D7 FC /D7 FD /D7 FE

GBK分布

类别	简称	范围	码位数	字符数	字符名	备注
符号标准区	GBK/1	A1A1-A9FE	846	717	图形符号	GB2312为主
	GBK/5	A840-A9A0	192	166	图形符号	BIG5和结构符
	小计		1038	883	图形符号	
汉字标准区	GBK/2	B0A1-F7FE	6763	6763	汉字	GB2312
	GBK/3	8140-A0FE	6080	6080	汉字	GB13000
	GBK/4	AA40-FEA0	8160	8160	汉字	GB13000等
	小计		21008	21003	汉字	
用户自定义区	1区	AAA1-AFFF	564			
	2区	F8A1-FEFE	658			
	3区	A140-A7A0	672			限制使用
	小计		1894			
总计			23940	21886		

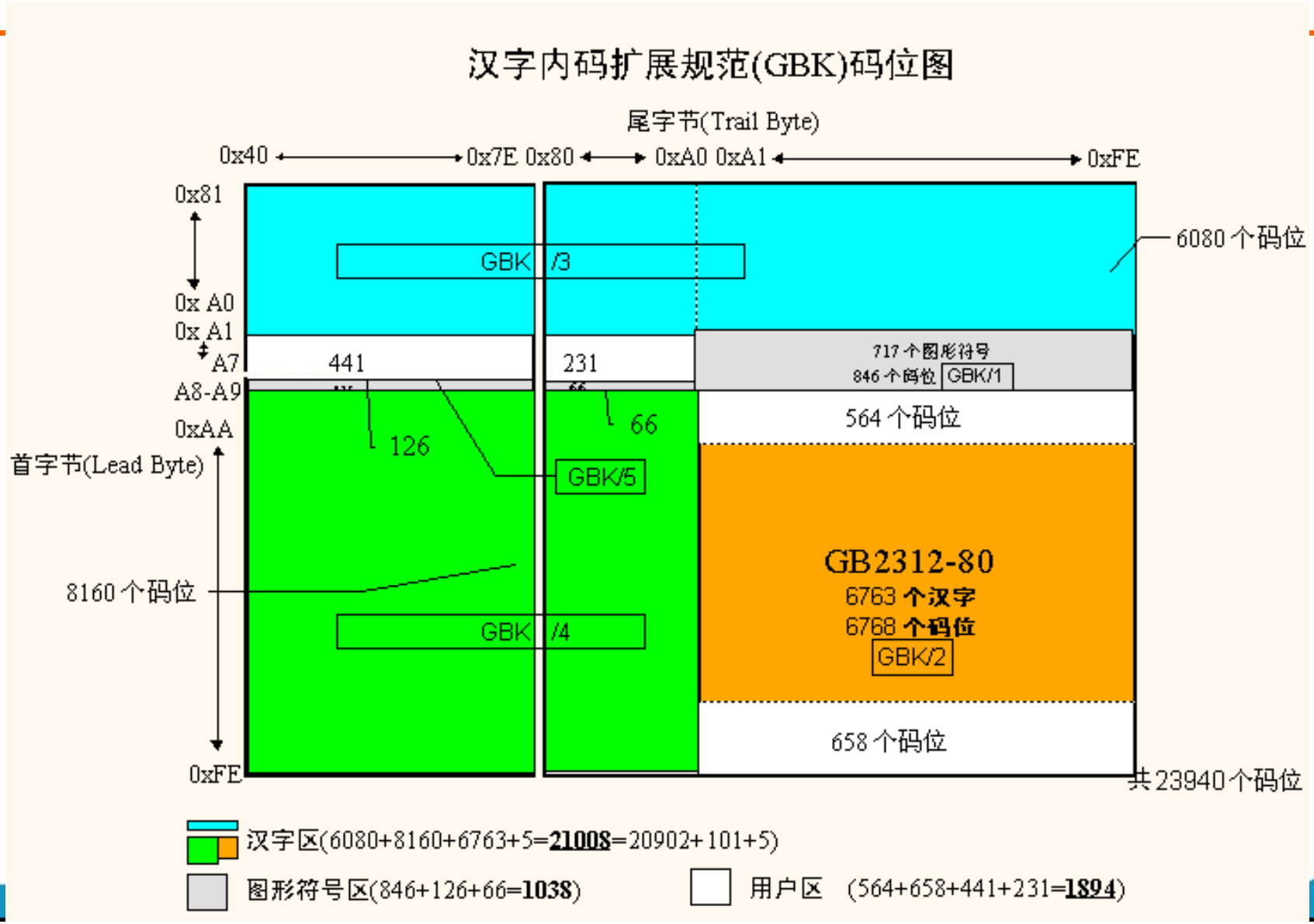
GBK文本文件

- 文件内容显示为：“一啊”
- 文件内码为：
- D2 BB B0 A1 0D 0A
- GBK内码是双字节内码，由高位字节和低位字节构成。
- 计算时需要高位和低位进行调换。
- 但GBK的计算方法和Unicode是不一样的，需要按照高位和低位字节来分布计算。

GBK的三个区间

- GBK/2为B0A1-F7FEH，表示高位内码从B0-A1H,低位内码位A1-FEH，不存在如B001H、80DEH等内码。
- GBK/3为8140-A0FEH，表示高位内码从81-A0H,低位内码位40-FEH。
- GBK/4为AA40-FEA0H，表示高位内码从AA-FEH,低位内码位40-A0H。

GBK码位图


汉字内码扩展规范(GBK)码位图
尾字节(Trail Byte)
0x40
0x7E 0x80
0xA0 0xA1
0xFE
首字节(Lead Byte)
0x81
0x A0
0x A1
A7
A8-A9
0xAA
0xFE
GBK /3
441
231
717个图形符号
846个码位
GBK/1
126
66
GBK/5
564个码位
6080个码位
8160个码位
GBK /4
GB2312-80
6763个汉字
6768个码位
GBK/2
658个码位
共23940个码位
汉字区(6080+8160+6763+5=21008=20902+101+5)
图形符号区(846+126+66=1038)
用户区 (564+658+441+231=1894)

GBK内码区间

- GBK的汉字内码分成三个区间，可以构成三张表，其序号和内码的计算公式如下：
- GBK/2：
- 序号＝（gbkh-0xb0）*(0xfe-0xa1+1)+gbkl-0xa1
- GBK/3：
- 序号＝（gbkh-0x81）*(0xfe-0x40+1)+gbkl-0x40
- GBK/4：
- 序号＝（gbkh-0xaa）*(0xa0-0x40+1)+gbkl-0x40
- 其中，gbkh为高位字节，gbkl为低位字节

序号公式

- 假设以H_1L_1-H_2L_2表示GBK/I(I=2,3,4)的内码范围，则以上公式可以统一为：
- 序号＝（gbkh-H_1）*(L_2-L_1+1)+gbkl-L_1
- (H_1<=gbkh<=H_2, L_1<=gbkl<=L_2)

GBK三段

GBK/2

序号	GBK内码
0	B0A1
1	B0A2
…	…
93	B0FE
94	B1A1
…	…
6755	F7FC
6766	F7FD
6767	F7FE

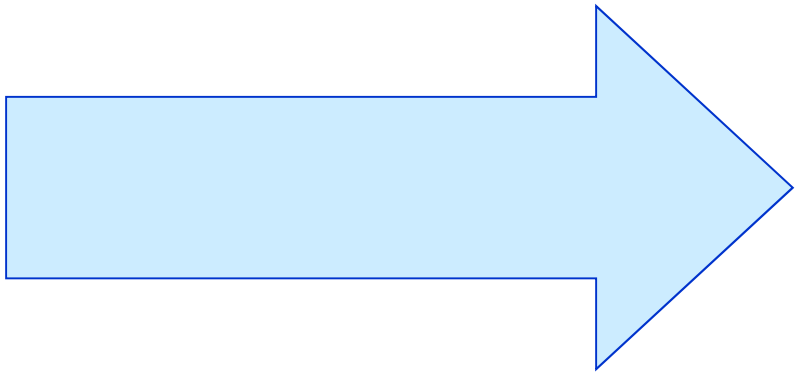
GBK/3

序号	GBK内码
0	8140
1	8141
…	…
190	81FE
191	8240
…	…
6077	A0FC
6078	A0FD
6079	A0FE

GBK/4

序号	GBK内码
0	AA40
1	AA41
…	…
96	AAFE
97	AB40
…	…
8157	A09E
8158	FE9F
8159	FEA0

- 三个表合并，构成一个表
- 重新修改分段函数



GBK/2

序号	GBK内码
0	B0A1
1	B0A2
…	…
93	B0FE
94	B1A1
…	…
6755	F7FC
6766	F7FD
6767	F7FE

GBK/3

序号	GBK内码
0＋6768	8140
1＋6768	8141
…	…
190 ＋6768	81FE
191 ＋6768	8240
…	…
6077 ＋6768	A0FC
6078 ＋6768	A0FD
6079 ＋6768	A0FE

GBK/4

序号	GBK内码
0＋12848	AA40
1 ＋12848	AA41
…	…
96 ＋12848	AAFE
97 ＋12848	AB40
…	…
8157 ＋12848	A09E
8158 ＋12848	FE9F
8159 ＋12848	FEA0

GBK/2

GBK/3

GBK/4

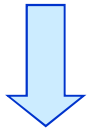
序号	Unicode内码
0	
1	
…	
93	
94	
…	
6755	
6766	
6767	

序号	Unicode内码
0＋6768	
1＋6768	
…	
190 ＋6768	
191 ＋6768	
…	
6077 ＋6768	
6078 ＋6768	
6079 ＋6768	

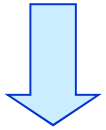
序号	Unicode内码
0＋12848	
1 ＋12848	
…	
96 ＋12848	
97 ＋12848	
…	
8157 ＋12848	
8158 ＋12848	
8159 ＋12848	

构造内码对照表

- Gbk->iso
- ANSI Code(GBK)

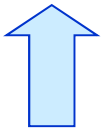


- MultiByteToWideChar



Unicode

构造内码对照表

- ISO->GBK
- ANSI Code(GBK)
- 
- WideCharToMultiByte
- 
- Unicode

实验二程序分析

- 编写程序实验一个文件级的Unicode到GBK的转换
  - 将内码转换表加载到内存
  - 二进制方式读取文件，跳过文件头
  - 读取2字节，计算序号，查表转换
  - 写结果文件
  - 注意非中文符号的处理

实验三

- 汉字机内码与交换码转换
- 在互联网发展早期，互联网上设备与软件主要传递的是ASCII字符，尤其是Email只能传递ASCII字符
- ASCII字符的最高位通常是0，而汉字机内码每个字节最高位往往可能是1
  - 字节的最高位可能被强行置成0
  - 也有可能被认为是错误的数据而丢弃

交换码

- 交换码的主要思想
  - 是在数据通过网络传输之前对之进行编码，使之符合网络传递要求，在接收到数据后对之进行解码，从而恢复数据
  - UUENCODE
  - Quoted-Printable
  - Base64
  - HZ
  - ……

Base64

- MIME
  - Multipurpose Internet Mail Extensions
  - 多功能Internet 邮件扩充服务
  - RFC1341
- 它是被多媒体电子邮件和 WWW 超文本广泛使用的一种编码标准，用于传送诸如图形、声音和传真等非文本数据

MIME中使用Base64的例子

MIME-Version: 1.0
Content-Type: text/plain;
charset="gb2312"
Content-Transfer-Encoding: base64
X-MSMail-Priority: Normal
X-MimeOLE: Produced By Microsoft MimeOLE V5.00.2615.200
ztK5+rK7vfbKx8rAvefJz9futPO1xMuutb7J+rL6ufq6zc/7t9G5+qOsuPzKx7W+1/fA+sq31+7T
xr7DoaLLrrW+0sW0q9fK1LTX7rfhuLu1xLn6vNLWrtK7oaO/vLnF1qTD96Os1OfU2jcwMDC24MTq
x7CjrM7Sufq+zb+qyrzW1tayy661vqGjDQo=

Base64编码思想

- 实质：
  - 三字节变四字节
  - 3*8=4*6=24
- 但是为了避免与ASCII的控制符号冲突
  - 转换后的每个字节根据对应的值用一个符号加以代替，从而保证编码后肯定是可显示的符号
  - 因为2^6=64，因此编码后的结果每一位是64个符号之一

Base64编码表

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	/

编码例子

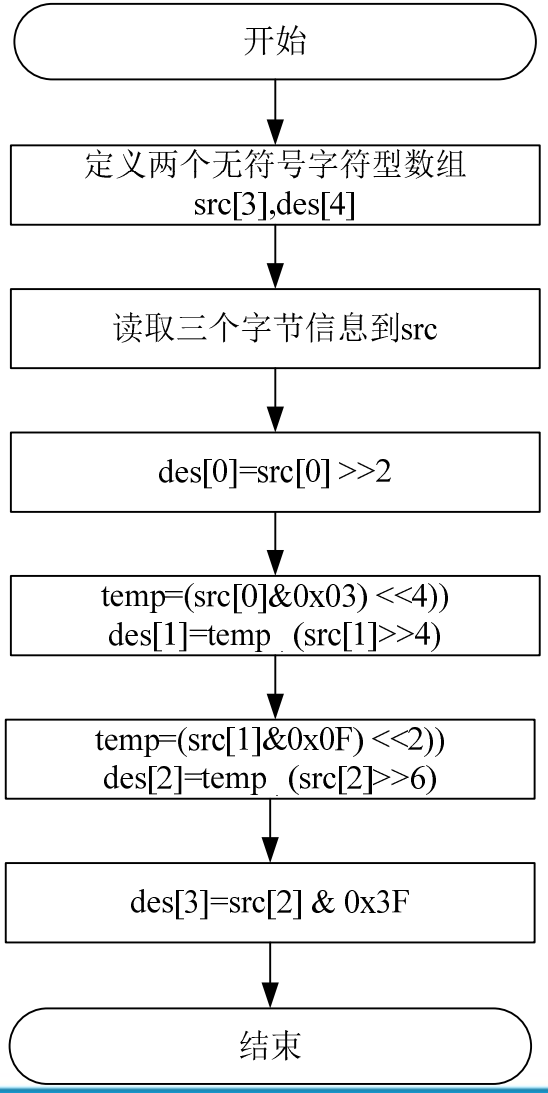
字符	A	B	1
字符ASCII码16进制	0x41	0x42	0x31
字符ASCII码2进制	01000001	01000010	00110001

↓ 3字节转4字节

每字节前补00	00 010000	00 010100	00 001000	00 110001
16进制	0x10	0x14	0x08	0x31
10进制	16	20	8	49
查base64表	Q	U	I	x

实验分析

- 3字节变4字节
  - 位运算



特别说明

- 如果待编码的数据长度不是3的倍数
- 则编码后不足4字节的部分补充“=”，保证编码后的结果一定是4的倍数

- 因此Base64编码最终结果肯定是4的倍数
- Base64编码的结果最多有65种符号

实验三程序分析

- 编写程序实现文件级的Base64编码
  - 二进制打开输入文件
  - 读取3字节
  - 调用编码函数
  - 写输出文件
  - 循环
- 输出文件可以文本方式打开
- 注意文件尾数不足三字节的情况

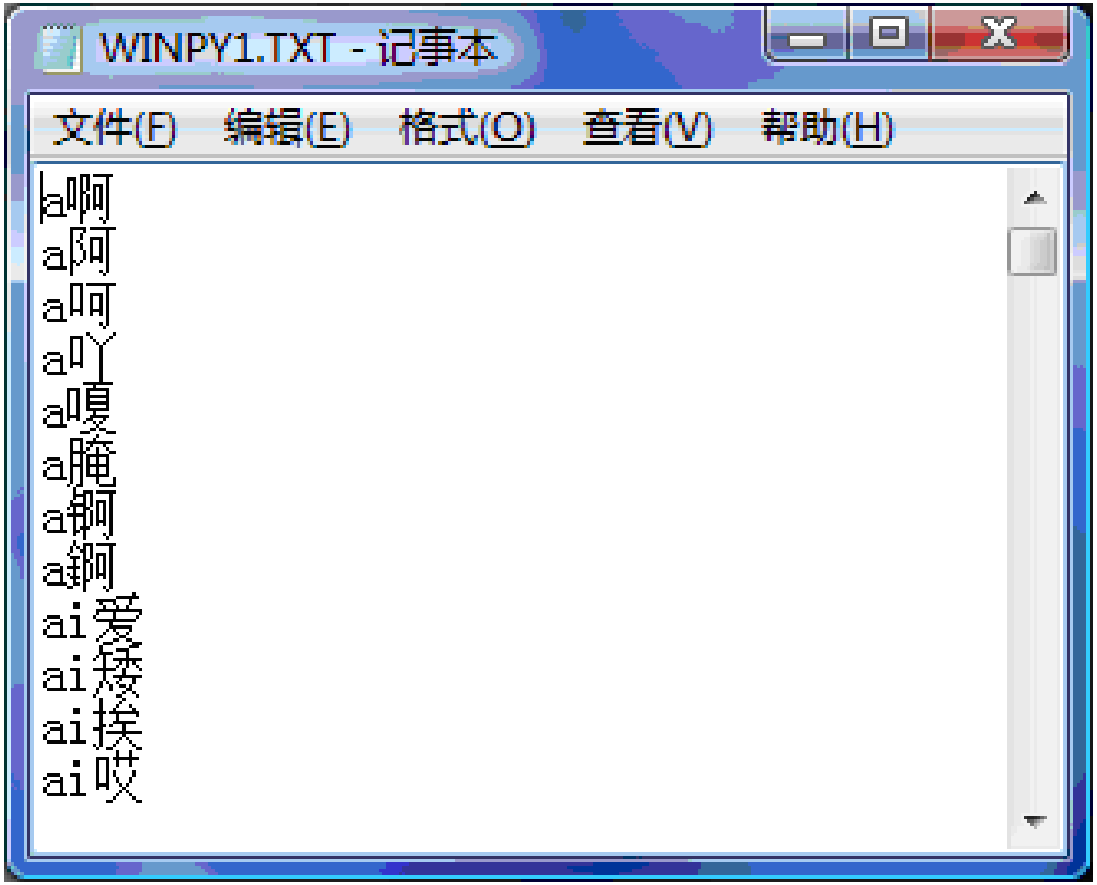
实验四

- 输入系统码本的构建与检索
  - 几乎每一个输入法都需要有自己的检索码本
    - 内码输入法可以不需要码本
    - 为什么？
  - 几乎输入法都未采用数据库
    - 用户计算机没有数据库引擎怎么办？
    - 用户计算机数据库引擎版本不对怎么办？
- 自己设计码本格式
  - 便于检索
  - 节约存储
  - 便于扩展

实验的原始数据

- 拼音码本
- 格式：
  - <拼音码本>::=<拼音项、字 >[<拼音项、字 >]
  - <拼音项、字 >::=<拼音><汉字><0x0d0a>
  - <汉字>::=<2个字节的GBK机内码>
  - <拼音>::=[<声母>]<韵母>

示例文本


WINPY1.TXT - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
a啊
a阿
a呵
a吖
a嗄
a腌
a锕
a錒
ai爱
ai矮
ai挨
ai哎

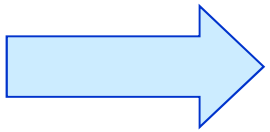
压缩拼音方案

- 拼音不等长，存储和查询麻烦
- 可以设计方案进行压缩
  - 节约存储
  - 处理方便

声母	b(1) p(2) m(3) f(4) d(5) t(6) n(7) l(8) g(9) k(10) h(11) j(12) q(13) x(14) zh(15) ch(16) sh(17) r(18) z(19) c(20) s(21)w(22)y(23) 无声母（0）
韵母	a(1) o(2) e(3) ai(4) ei(5) ao(6) ou(7) an(8) en(9) ang(10) eng(11) ong(12) i(13) ia(14) ie(15) iao(16) iou(17) ian(18) in(19) iang(20) ing(21) iong(22) u(23) ua(24) uo(25) uai(26) uei(27) uan(28) uen(29) uang(30) ueng(31) v(32) ve(33) van(34) vn(35)

压缩示例

❖ wo 我	16 02 CE D2
❖ yong用	17 0C D3 C3
❖ guang光	09 1E B9 E2
❖ zuo坐	13 19 D7 F8



定长结构

- 定长结构的表项为：
  - <2个字节压缩拼音码><2个字节汉字机内码>
- 如果是GB2312则一共6763项
- 按照拼音排序

定长结构例子

0	00 01	啊(b0 a1)
1	00 01	阿(b0 a2)
……	……	……
	01 01	八(b0 cb)
	01 01	巴(b0 cd)
	……	……
	13 19	怍(e2 f4)
6759	13 19	胙(eb d1)
6760	13 19	祚(ec f1)
6761	13 19	酢(f5 a1)
6762		

每个表项长度为4

定长结构分析

- 以GB2312字符集为例
- 对照表大小：
  - 6763×4＝27052
  - 因为多音字的缘故，略大于6763
- 检索算法：二分法
  - 二分的前提是有序
  - 平均检索次数：$\log_2 6763 - 1 \approx 12$

变长结构

- 定长结构中拼音被不断地重复
  - 数据冗余
- 在6763个汉字中，
  - 一共有399个不同的拼音
  - 则共有399个表项
  - 每个表项由
    - 压缩输入码（2字节）
    - 汉字集（不定长）构成

变长结构样例

a	阿吖嗄锕埃
ai	挨哎唉哀皑癌蔼矮艾碍爱隘捱嗳嗌嫒瑷暧砹锿霭
…	
…	
…	
zun	尊遵撙樽鳟
zuo	昨左佐柞做作坐阼 阼唑怍胙祚酢

变长结构分析

- 对照表大小：
  - 6763×2＋399×2＝14324
  - 要略大于该数字，因为有多音字
  - 还需要行结束符
- 检索算法：
  - 扫描法
  - 平均检索次数：（399＋1）/2≈200

程序分析

- 编写程序生成定长和变长结构的码本
  - 原始数据
    - 是以Unicode存储方便？
    - 还是以GBK编码存储方便？
  - 生成的码本
    - 是二进制文件格式好？
    - 还是文本文件好？