中文信息处理
实验分析（二）

苏州大学计算机科学与技术学院

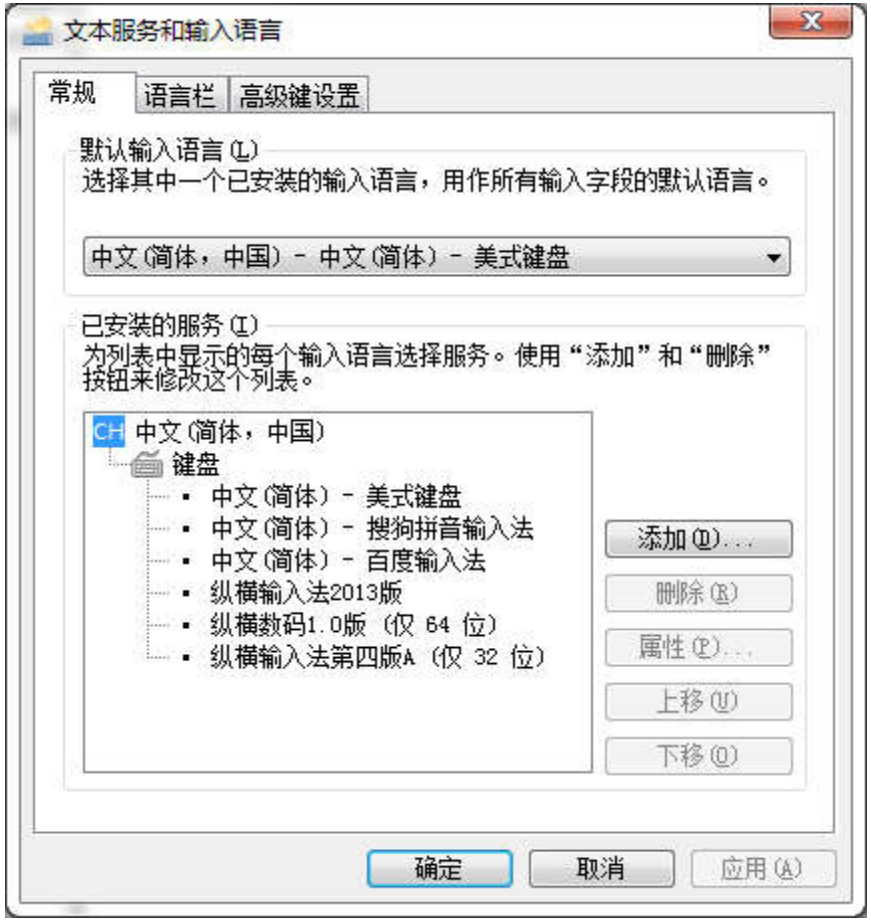
主要内容

- 实验五分析
- 实验六分析
- 实验七分析
- 实验八分析

实验五

- Windows 汉字输入法实现
- 预备知识
  - IMM-IME
    - Input Method Manager
    - Input Method Editor
    - 全拼输入法的 IME 文件是winpy.ime
    - IME实质是DLL
  - Windows 8 metro
    - TSF

IMM界面


文本服务和输入语言
常规
语言栏
高级键设置
默认输入语言(L)
选择其中一个已安装的输入语言，用作所有输入字段的默认语言。
中文(简体，中国) - 中文(简体) - 美式键盘
已安装的服务(I)
为列表中显示的每个输入语言选择服务。使用“添加”和“删除”按钮来修改这个列表。
中文(简体，中国)
键盘
中文(简体) - 美式键盘
中文(简体) - 搜狗拼音输入法
中文(简体) - 百度输入法
纵横输入法2013版
纵横数码1.0版 (仅 64 位)
纵横输入法第四版A (仅 32 位)
添加(D)...
删除(R)
属性(P)...
上移(U)
下移(O)
确定
取消
应用(A)

三种应用程序

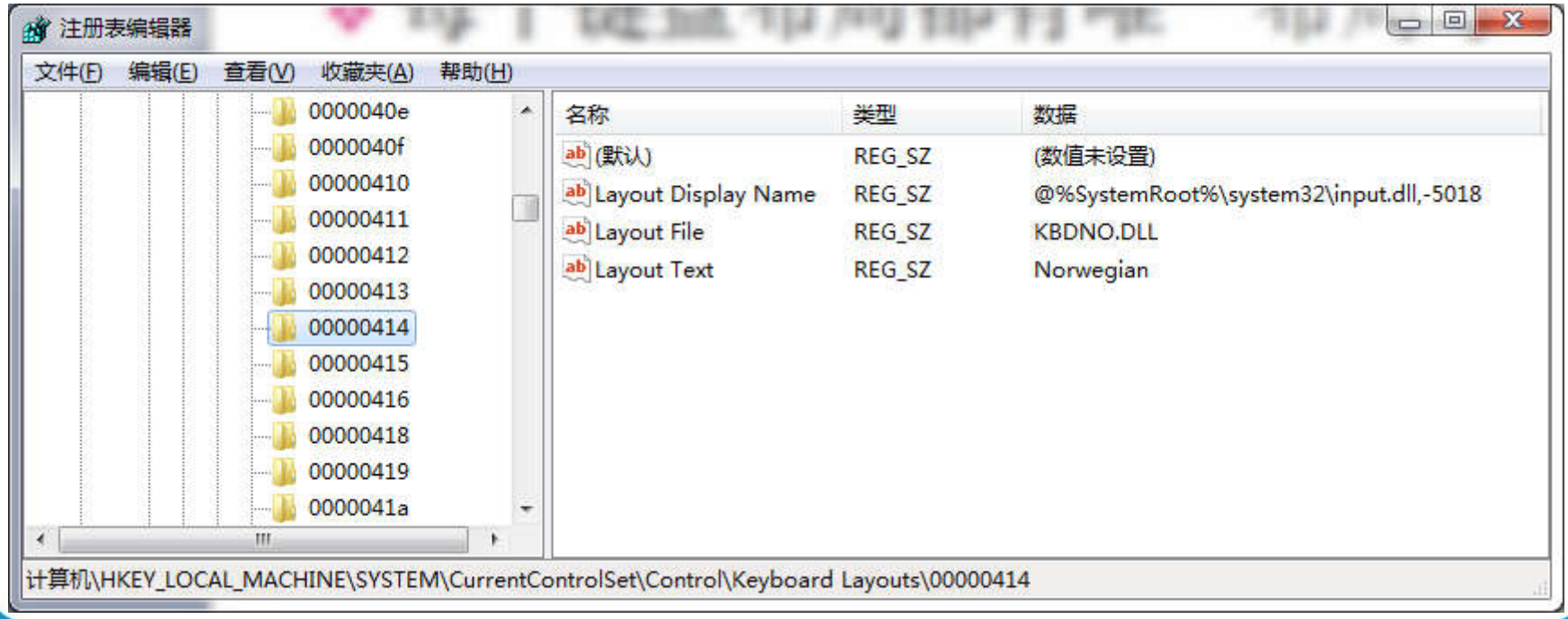
- 不识别IME 的应用程序
  - 不感知用户当前的输入法
- 部分识别 IME 应用程序
  - 感知用户当前的输入法
- 完全识别 IME 的应用程序
  - 自定义IME窗口
  - IME只是负责输入码到字词列表的转换

IME是一种插件

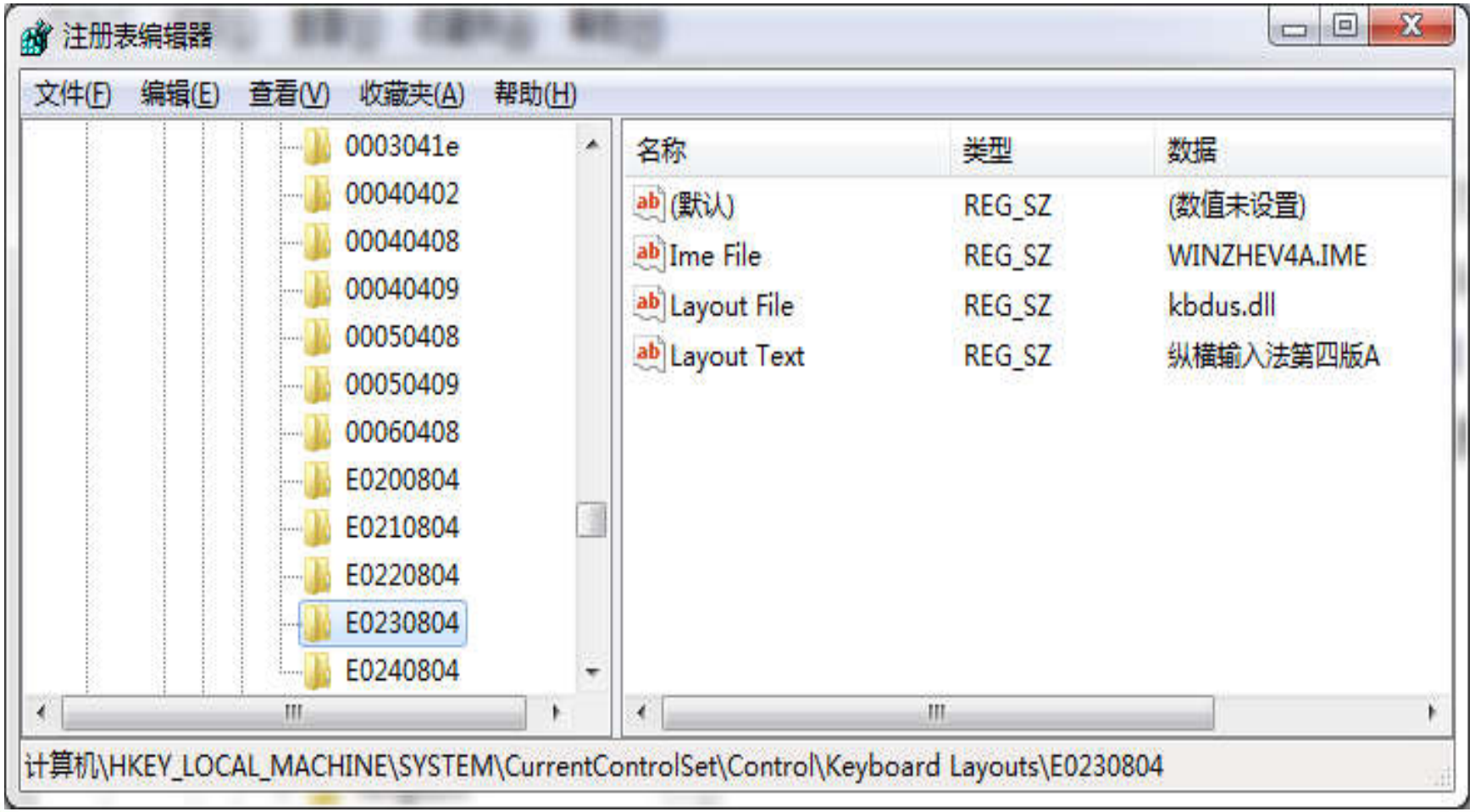
- 输入法属于系统软件
- 用户可以根据需要添加删除
  - 输入法需要被系统枚举
    - 注册输入法
  - 输入法需要和系统交互
    - 接口
    - CALLBACK

IME注册机制

- 一个输入法是一个键盘布局
- 每个键盘布局都有唯一布局号
  - 存储在注册表

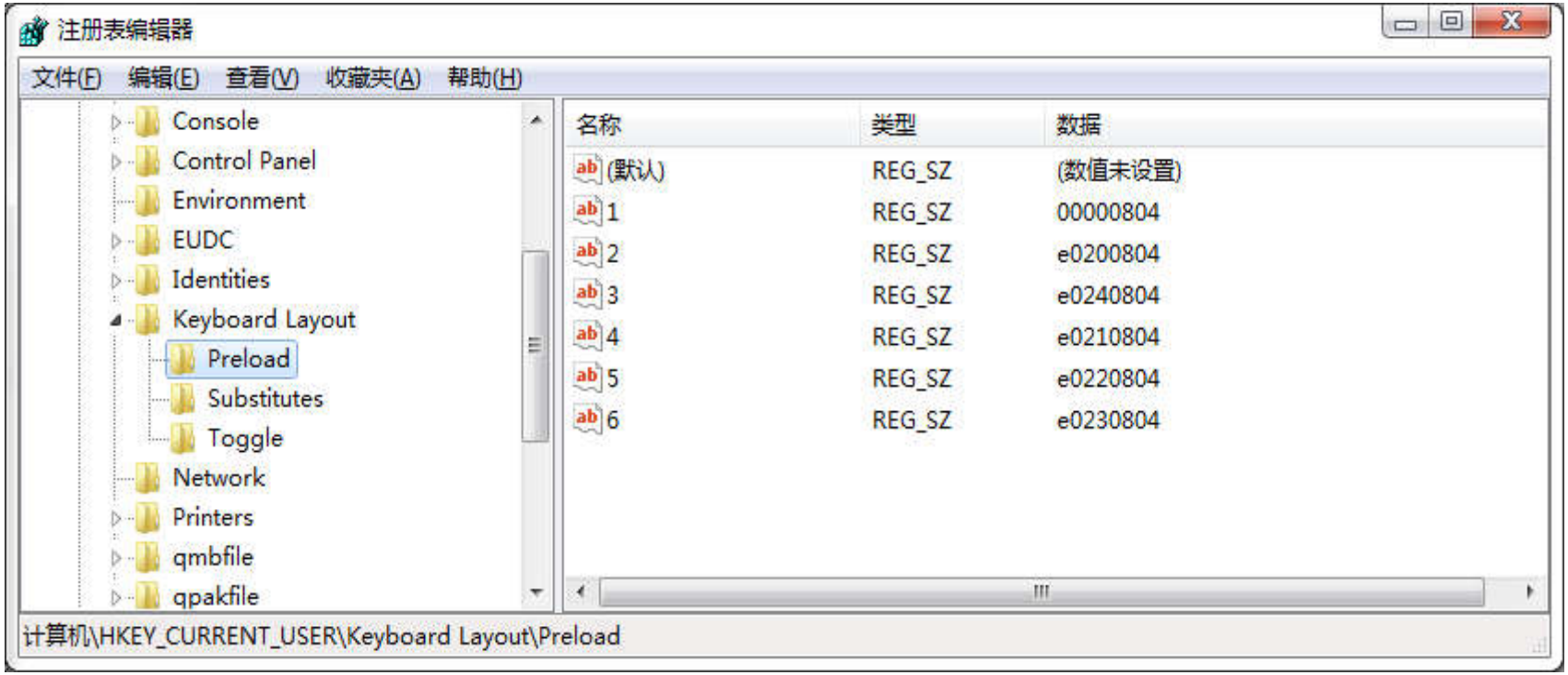


纵横输入法


注册表编辑器
文件(F) 编辑(E) 查看(V) 收藏夹(A) 帮助(H)
0003041e
00040402
00040408
00040409
00050408
00050409
00060408
E0200804
E0210804
E0220804
E0230804
E0240804
名称 类型 数据
(默认) REG_SZ (数值未设置)
Ime File REG_SZ WINZHEV4A.IME
Layout File REG_SZ kbdus.dll
Layout Text REG_SZ 纵横输入法第四版A
计算机\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Keyboard Layouts\E0230804

当前使用的输入法

- 注册表
  - \HKEY_CURRENT_USER\Keyboard_Layout\Preload



注册输入法

- 注册一个键盘布局
  - 1. IME 文件复制到 Windows 的系统目录
  - 2. 调用 API函数 ImmInstallIME

  - HKL ImmInstallIME(
    LPCTSTR lpszIMEFileName , / / IME 的路径
    LPCTSTR lpszLayout Text / / IME 的名称)

IME函数接口

- 1)ImeInquire
  - 此接口函数在用户选择输入法时最先由IMM调用，以获得该输入法的有关信息，函数应返回IME的初始化信息，设置当前输入法的各项属性，以及当前输入法的用户界面窗口类名称等。
- 2)ImeConfigure
  - 此接口函数将在用户通过控制面板或系统图标设置输入法属性时被IMM调用，在此函数中可以显示属性设置对话框，为用户配置输入法提供配置窗口界面。
- **3)ImeProcessKey**
  - 此接口函数由输入法在处理键盘事件时调用，判断输入法是否需要处理一个用户键入的键值，如果不需要则返回FALSE，该键将直接发给应用程序，否则返回TRUE，IMM将会调用ImeToAsciiEx进行处理。
- **4)ImeToAsciiEx**
  - 该函数的工作是对输入的按键进行处理，只有当ImeProcessKey返回TRUE时才会调用这个接口函数，该函数主要根据当前的状态进行处理，可以调用转换引擎，或者进行标点、符号的转换。

IME函数接口（续1）

- 5)**ImeSelect**
  - 当输入法被选择或者选出时，系统将调用本函数。
- 6)ImeSetActiveContext
  - 当输入法被激活或者搁置的时候，系统将调用本函数。
- 7)NotifyIME
  - 系统或者应用程序太过本函数根据参数的值改变输入法的当前状态。例如：显示或隐藏候选字词窗口，选定了一个候选字词，修改输入码串的内容。
- 8)ImeDestroy
  - 当输入法被卸载时，将调用本函数。

IME函数接口（续2）

- 9)ImeConversionList
  - 本函数用于将一个指定的输入码转为为结果串，通常用于被部分识别IME的应用程序和完全识别IME的应用程序调用，在大多数输入法中该函数可以无需实现而直接返回。
- 10)ImeEscape
  - 应用程序可以通过调用本函数来直接访问一个输入法的特定功能，一般而言这些功能应该是无法通过其他的IMM 函数调用来实现。这么做的主要目的是为了支持特定语种的函数或者IME 的私有函数。
- 11)ImeSetCompsitionString
  - 本函数可以直接设置输入法的输入码，通常用于被部分识别IME的应用程序和完全识别IME的应用程序调用，在大多数输入法中该函数可以无需实现而直接返回。

IME函数接口（续3）

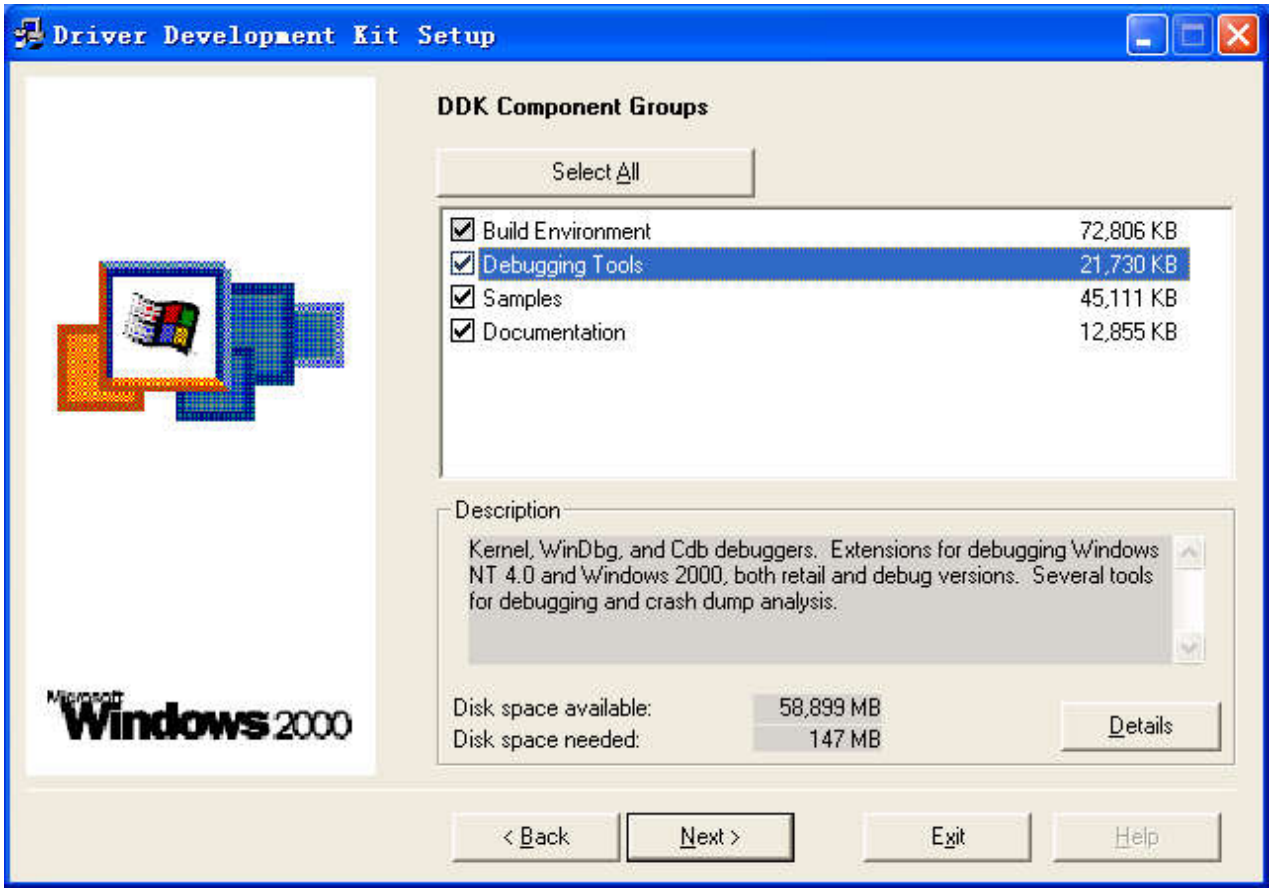
- 12)ImeRegisterWord
  - 本函数用于向输入法码本中添加一个字或词组，大多数输入法中该函数可以无需实现而直接返回。
- 13)ImeUnregisterWord
  - 本函数用于在输入法码本中删除一个字或词组，大多数输入法中该函数可以无需实现而直接返回。
- 14)ImeGetRegisterWordStyle
  - 本函数用于获取输入法码本中字或词组条目的形式，大多数输入法中该函数可以无需实现而直接返回。
- 15)ImeEnumRegisterWord
  - 本函数用于在输入法码本中枚举字或词条，大多数输入法中该函数可以无需实现而直接返回。

DDK

- Device Develop Kit
- 设备驱动程序开发包
- 用于开发Windows各种设备的驱动程序
- DDK 是操作系统相关的
  - Windows 98/ME/2000/XP/Vista/7/8
  - Windows Server 2000/2003/2008
  - 开发出的驱动程序，也只能运行在指定的系统
- DDK本身免费下载

DDK的安装

- 必须和Visual C++结合使用



DDK例子程序

- DDK中附带了一个
  - 区位输入法的程序
  - 实现了所有的必要函数接口
- 修改
  - 添加输入法码本检索代码
  - 添加输入法逻辑处理代码

输入法界面发展

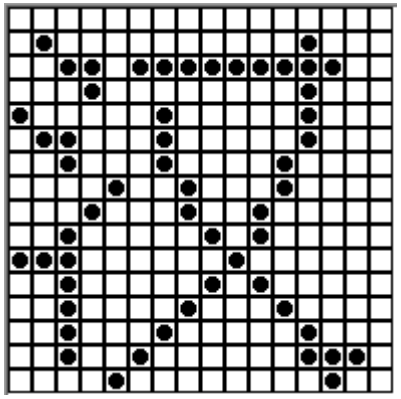
- 输入法状态窗口
- 输入码编辑窗口
- 候选字词窗口

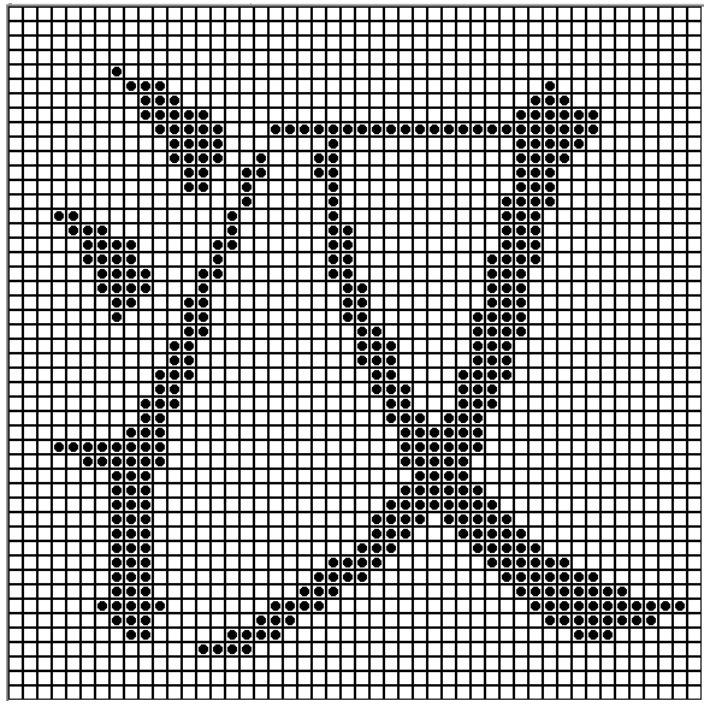
- 输入码编辑窗口与候选窗口合并
- 竖排->横排
- 候选字词变少

实验六

- 基于点阵字库的汉字显示
  - 点阵字库，最经典的汉字显示技术
  - 在小分辨率设备上效果明显

不同点阵的效果





- 纵向点阵
  - 按照纵向8位归并成字节
  - 传统的针式打印机是把汉字从左向右打印过去
- 横向点阵
  - 按照横向8位归并成字节
  - 传统的CRT显示器是采用逐行扫描的方式

点阵字库的原始文件

- HZK16
  - 16*16点阵
  - 包含了GB2312-80中的所有字符
  - GB2312从16区开始才是汉字
  - 包含前15区的字符点阵
- HZK48S
  - 48*48点阵 宋体
  - 不包含前15区的字符点阵

地址码

- 点阵字库文件是二进制文件
- 获取点阵的过程
  - 给定汉字HZ
  - 计算出HZ的序号Order
  - Order*Len(每字符点阵码所占的长度)得到Start
  - 定位到Start
  - 读Len个字节

地址码计算

- HZK16
  - unsigned int order
    = buff[1]-0xA1+94*(buff[0]-0xb0)+15*94;
  - unsigned int position=order*32；
- HZK48S
  - unsigned int order
    = buff[1]-0xA1+94*(buff[0]-0xb0);
  - unsigned int position=order*288；

汉字变形处理

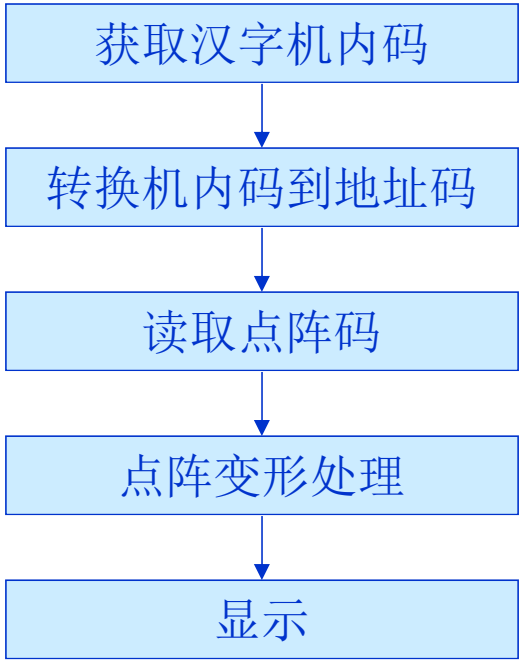
- 拉伸
- 加粗
  - 密集
- 反色
  - 阴文
- 倾斜
- 加下划线
- 加删除线

小字库的制作

- 嵌入式等存储受限领域
  - 只需要显示少量的汉字
  - 可以将字库中的部分汉字信息抽出
  - 存储成一个二进制文件
  - 自己维护汉字顺序关系
  - 例如：微波炉、冰箱面板

实验程序分析

- 二进制文件读
- 位运算

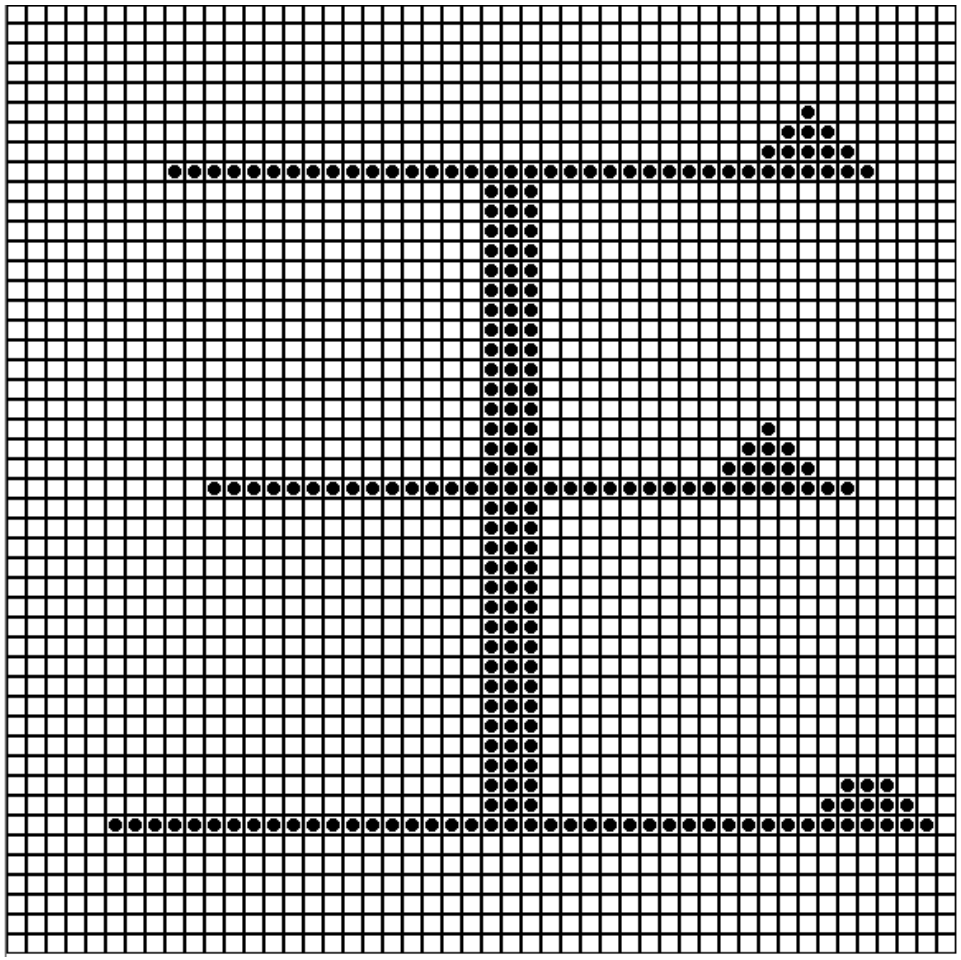


实验七

- 点阵字库的压缩与还原
- 书刊、报纸实际印刷中对汉字点阵的要求往往很高
  - 以512*512为例，
  - 一个汉字就需要512*512/8=32KB，
  - Unicode BMP平面
  - 宋体、楷体、仿宋体、黑体四种最常用的字体
  - 需要20902*4*32KB=2612.75MB的存储空间
    - 以前难以逾越的空间
    - 上世纪90年代早期，大多数个人计算机的存储容量500M

黑白段

- 上个世纪60年代
- 德国Hell公司
- 第三代阴极射线管照排机Digiset上采用
- 黑白段压缩的思想：
  - 有数据的部分称为黑段，无数据的部分称为白段
  - 通过扫描点阵信息交替记录下每行的黑段与白段
  - 并设计了“重复行数”来处理汉字中大量存在的连续多行点阵信息相同的情况。
- 黑白段的具体记录原始数据格式如下：
  - 【行标识】【重复行数】【白段】【黑段】……【白段】【黑段】



“王”的黑白段压缩码

行标记	重复行数	白段	黑段	白段	黑段	白段
*	5	48				
*	1	40	1	7		
*	1	39	3	6		
*	1	38	5	5		
*	1	8	36	4		
*	12	24	3	21		
*	1	24	3	11	1	9
*	1	24	3	10	3	8
*	1	24	3	9	5	7
*	1	10	33	5		
*	14	24	3	21		
*	1	24	3	15	3	3
*	1	24	3	14	5	2
*	1	5	42	1		
*	6	48				

黑白段的压缩效果

- 48*48点阵“王”字
- 点阵码
  - 48*48/8=288个字节
- 采用了黑白段压缩法压缩
  - 设行标记、重复行数、白段、黑段都占用一个字节
  - 81个字节
  - 由于每一行黑段和白段的和应该是48，在实际实现的时候每行最后一个黑白段信息可以通过48减去该行前面黑白段的和计算出来，因此每行最后一个黑白段信息可以省略，
  - 如果这样的话，48*48点阵的“王”字只需要用66个字节
- 显然对于点阵数越多的字形库，黑白段的压缩率越好

线性增量压缩法

- 黑白段的辅助方法
- 处理汉字中的大量斜线
- 思想是：在一行黑、白段记录信息的后面再注明线段的增量，这样下一行的黑、白段长度在上一行的基础上按增量的大小作相应的变化。
- 线性增量的具体记录原始数据格式如下：
  - 【行标识】【重复行数】【白段】【白段增量】【黑段】【黑段增量】……【白段】【白段增量】【黑段】【黑段增量】

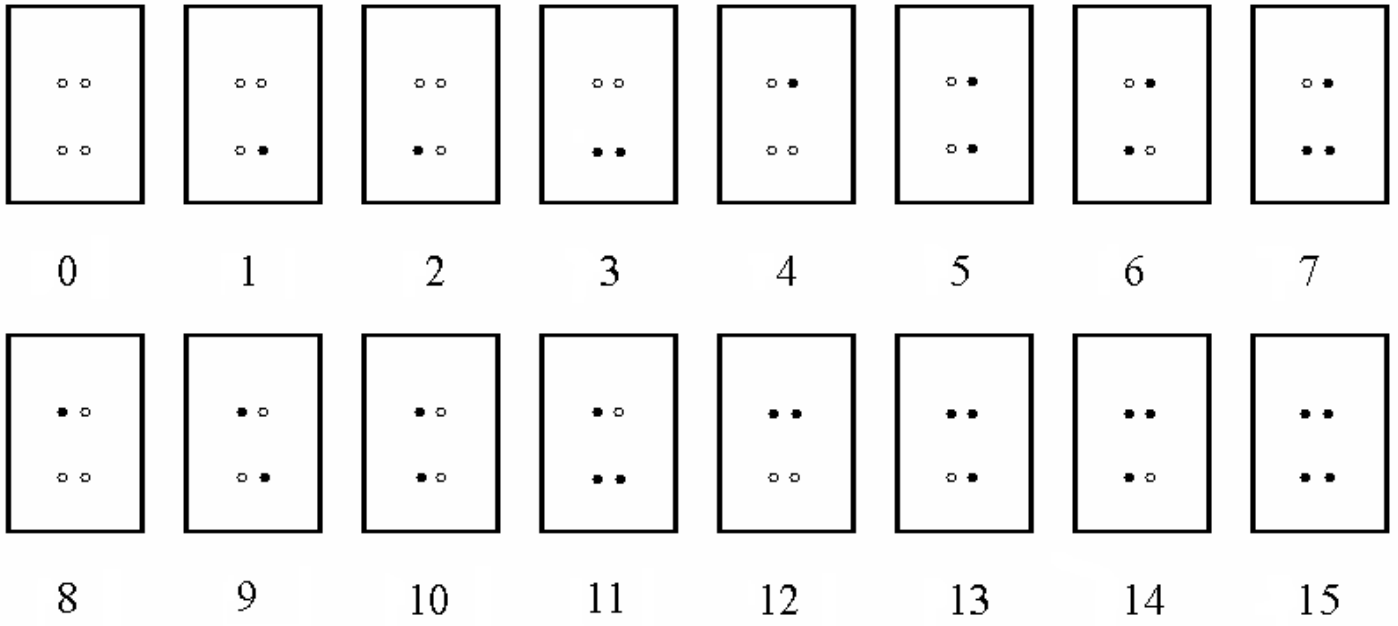
“王”

- 黑白段辅以线性增量

行标记	重复行数	白段		黑段		白段		黑段		白段	
*	5	48									
$	3	40	-1	1	2	7	-1				
*	1	8		36		4					
*	12	24		3		21					
$	3	24	0	3	0	11	-1	1	2	9	-1
*	1	10		33		5					
*	14	24		3		21					
$	2	24	0	3	0	15	-1	3	2	3	-1
*	1	5		42		1					
*	6	48									

哈夫曼压缩法

- 将汉字点阵的组合合并观察



不等长编码

- 对汉字点阵信息进行统计

状态编号	出现概率	编码	状态编号	出现概率	编码
0	0.438	1	8	0.011	011101
1	0.050	00000	9	0.024	000010
2	0.011	0111000	10	0.148	001
3	0.041	00010	11	0.023	000011
4	0.032	01100	12	0.025	01111
5	0.132	010	13	0.002	01110010
6	0.032	00011	14	0.015	011010
7	0.014	011011	15	0.001	01110011

实验程序分析

- 利用汉字内码获取点阵信息
- 读取点阵信息
- 生成黑白段压缩码
- 写压缩结果

- 注意黑白段重复行的处理

实验八

- 汉语自动分词
  - 是目前基于内容文本研究的首要基础
    - 信息检索、信息分类、信息抽取等
  - 直接应用
    - 自动校对
      - 抛妻别字 —— 抛妻别子
    - 多音字识别
    - 简繁转换
      - 後面，皇后 —— 后
      - 松树，鬆开 —— 松

发展

- 研究了几十年
- 目前正确率98%以上
- ACL
  - (Association for Computational Linguistics)
  - SIGHAN
  - 2003年开始
  - 组织了多次国际汉语自动分词竞赛，对自动分词的发展起到了很大的促进作用。

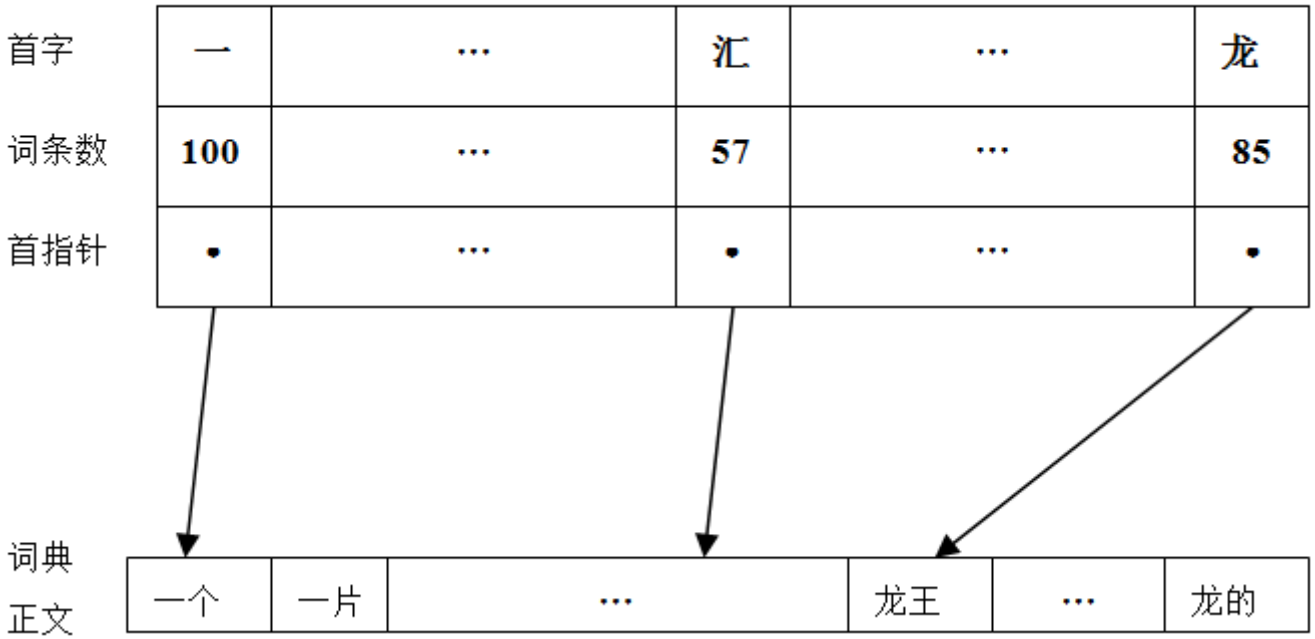
主要问题

- 歧义处理
  - 交集型歧义
  - 组合型歧义
- 未登录词识别
  - 新词层出不穷
- 内塔尼亚胡说

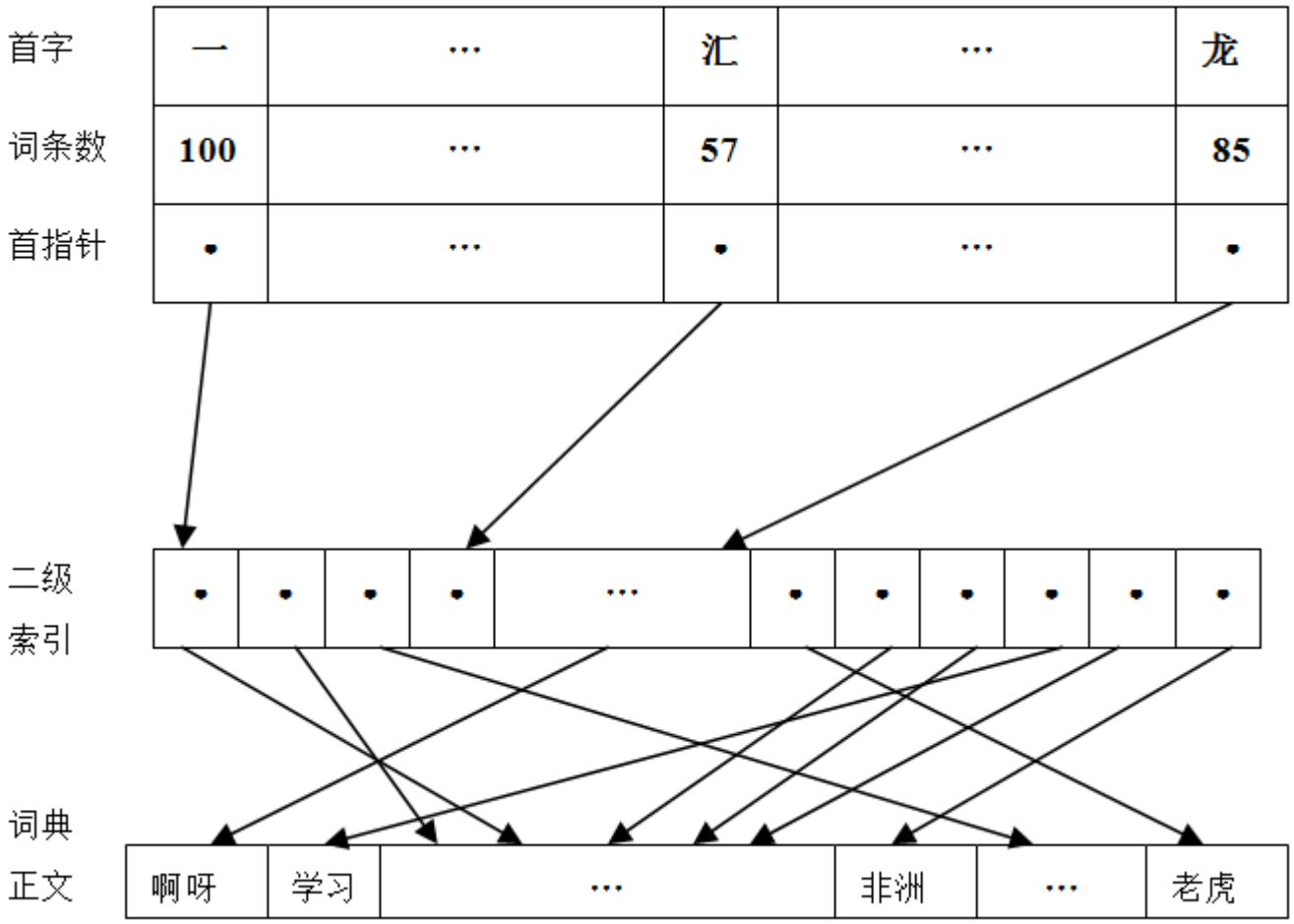
机械分词

- 正向机械分词
- 逆向机械分词
- 算法简单
  - 词多，如何保证高效？
  - 可以参考联想码本的设计技术来设计分词词典

基于首汉字索引的分词字典结构



基于首汉字二级索引的分词字典结构



第三方分词系统的使用

- ICTCLAS
  - Institute of Computing Technology, Chinese Lexical Analysis System
  - 中科院计算所
  - 开源
- 海量智能分词
  - 天津海量信息技术有限公司
- 斯坦福分词
  - 是一个开源的软件
  - 采用Java语言实现

海量分词接口分析

- 三类接口
  - 第一类是基本接口
    - 用于初始化和卸载分词运行环境
    - 例如加载分词字典、分配内存等
  - 第二类用于分词
  - 第三类接口用于获取分词的结果

基本接口

- 1）bool HLSplitInit (const char* lpszDataFilePath)
  - 功能描述：初始化海量分词系统，加载分词用数据。
  - 说　明：如果初始化失败，通常的原因是初始化路径不正确或数据文件遭到破坏。初始化成功后，可进行多次分词，在不再进行分词时，调用HLFreeSplit函数卸载分词系统。
- 2) void HLCloseSplit(HANDLE hHandle)
  - 功能描述：关闭分词结果句柄, 释放分词结果所占资源。
  - 说　明：调用此接口前请保证分词词典已加载成功；在每次调用HLOpenSplit后调用此函数释放分词所占资源。
- 3) void HLFreeSplit(void)
  - 功能描述：卸载海量自动中文分词系统，释放分词系统所占资源。
- 4) HANDLE HLOpenSplit()
  - 功能描述：创建自动中文分词结果句柄。

分词接口

- bool HLSplitWord (HANDLE hHandle, LPCSTR lpText, int iExtraCalcFlag = 0)
  - 功能描述：　　对指定字符串进行分词。
  - 参数说明：
    - hHandle　　[IN]　　分词结果句柄。
    - lpText　　[IN]　　要分词的字符串。
    - ExtraCalcFlag[IN]　　附加计算标志，默认为0
  - 返 回 值：　返回成功标志。
    - 成功返回true，否则返回false。
  - 说　明：　调用此接口前请保证分词词典已加载成功，并且分词结果句柄有效。

获取结果

- 1) int HLGetWordCnt(HANDLE hHandle)
  - 功能描述：　　得到分词结果中的词的个数。
  - 参数说明：　　hHandle　[IN]　　分词结果句柄。
  - 返 回 值：　分词结果的个数，小于 0 表示失败。
  - 说明：
    - 调用此接口前请保证分词词典已加载成功，并且分词结果句柄有效。
- 2) SHLSegWord* HLGetWordAt(HANDLE hHandle, int iIndex)
  - 功能描述：　　得到指定的分词结果。
  - 参数说明：　　hHandle　　[IN]　　分词结果句柄。
  - 　　iIndex　　[IN]　　分词结果下标。
  - 返回值：　分词结果，返回NULL表示失败。
  - 说明：　　调用此接口前请保证词典已加载成功，且分词结果句柄有效。

代码片段

```
if(!HLSplitInit ())    //海量分词初始化
 {
        cout<<"海量分词初始化失败！";
        return ;
}
        ……
HANDLE hHandle = HLOpenSplit();                    //创建分词句柄
if(hHandle == INVALID_HANDLE_VALUE)
 {
        cout<< "创建分词句柄失败！" ;
        HLFreeSplit() ;//卸载分词字典
        return ;
}
LPCSTR lpText= _T("待分词的文本数据");
bool bRet = HLSplitWord (hHandle , lpText , 0);
                                                   //对分词结果进行处理
        ……
HLCloseSplit(hHandle);                             //关闭分词句柄
        ……
HLFreeSplit();                                     //卸载海量分词
```

程序分析

- 初始化分词数据
- 打开文本文件
- 读取文本数据
- 调用分词
- 获取分词结果
- 关闭文本文件
- 关闭分词数据

- 注释待分词文件的编码（Ansi/Unicode/Utf-8)