Penguin 拼音标注软件

设计书

指导老师：	薛　伟	
组　　长：	尚壬鹏	10125802
组　　员：	庞　博	10125793
	秦子童	10125797
	任　望	10125799
	王长海	10125806
	王　栋	10125808
	肖　洒	10125824
	邹永平	10125854

版本历史：

版本	日期	作者	说明
0.1	2010.12.04	尚壬鹏	创建
0.2	2010.12.22	邹永平	修改
0.3	2010-12-24	尚壬鹏	修改
1.0	2010-01-09	尚壬鹏	完善

目录

1　引言 5
1.1　编写目的 5
1.2　项目背景 5
1.3　定义 5
2　任务概述 6
2.1　目标 6
2.2　用户特点 6
2.3　假定和约束 6
2.4　注音处理方法 6
3　需求规定 7
3.1　功能规定 7
3.2　性能规定 7
3.2.1　精度 7
3.2.2　时间特性要求 7
3.2.3　灵活性 7
3.3　输入输出要求 8
3.4　数据管理能力要求 8
3.5　故障处理要求 8
3.6　其他专门要求 8
3.7　内部接口 8
3.7.1　定义 defination.h 8
3.7.2　统计接口 accounting.c 9
3.7.3　文件处理接口 pyFile.c 9
3.7.4　图形接口 pyGraphic.c 9
3.7.5　分词接口 particle.c 9
4　运行环境规定 11

4.1 设备.....11
4.2 运行环境.....11
4.3 外部接口.....11
4.4 控制.....11
5 系统整体流程图.....12
6 测试计划.....14
7 版本发布计划.....15

1 引言

1.1 编写目的

本详细设计书的编写目的，主要在于规范“Penguin 拼音标注”软件的编写。它说明了本软件的各项功能和性能需求，明确标识各功能的实现过程，提供一个度量和遵循的基准。另一方面，本说明书也是为了便于对开发过程的控制与管理，并作为工作成果的原始依据保存下来。

本文档面向的读者主要是项目发起人，项目管理员、项目设计人员和开发人员，希望能使开发工作更加具体。

1.2 项目背景

为了解决软件项目实训的任务，实现大部分项目组成员的目标，将设计开发一款软件实现汉字标注拼音的功能。

1、软件名称： Penguin 拼音标注

2、委托人：薛伟

3、开发单位：北京交通大学软件学院 10 级尚壬鹏组

4、开发团队：

分词	王栋，王长海
GTK	尚壬鹏、庞博、秦子童
文档及测试	肖洒、邹永平
QA	任望

5、与其他子系统： 本系统独立于其它软件，仅仅依赖字库文件

1.3 定义

1、汉字拼音标注：基本行为是对一篇文章的全部汉字标注拼音。

2、需求：用户解决问题或达到目标所需的条件或功能，系统或系统部件要满足合同、标准、规范或其它正式规定文档所需具有的条件或功能。

3、需求分析：包括提炼、分析和仔细审查已收集到的需求，以确保所有的风险承担者都明其含义并找出其中的错误、遗憾或其它不足的地方。

2 任务概述

2.1 目标

本汉字拼音标注系统是为项目实训任务所设计，目的是研究 Linux 系统下 C 程序团队设计开发软件，提高团队项目能力，学习自然语言文本处理方法。

2.2 用户特点

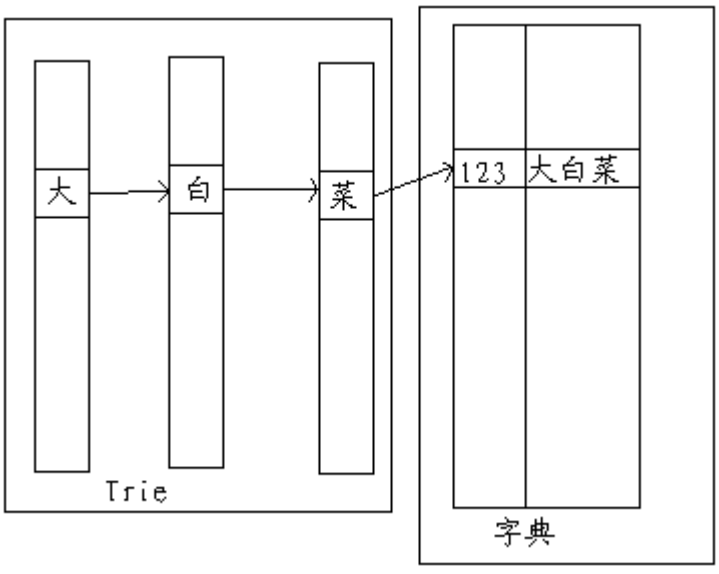
本软件用户是本组的成员以及以后可能用来研究的有 Linux 系统基础的学生，且因为软件的实验性质，估计最终用户较少，预期使用频率较低，因此使用方法可以比较难，界面可以简陋些，主要研究方向在于拼音标注的实现上。

2.3 假定和约束

软件开发没有经费。开发时间是从 12 月 1 日至 1 月 7 日，每周开发时间为周日和周三两天。

2.4 注音处理方法

先对文章根据非中文字符断句，断句后根据类似 trie 树的结构查找结果，如果找到结果，则读出拼音的编号，根据编号在字典中查询拼音，并使用 char 类型返回拼音。最终将拼音返回至主函数调用显示。



检索过程

3 需求规定

3.1 功能规定

输入	处理	输出
两个文本文件的路径	将第一个文本的汉字注音	把注音结果保存到第二个文本文件
在输入文本框中输入文字	点击“注音”按钮，对文本文字注音	把注音结果显示在输出文本框
点击打开按钮	在计算机中查找一个文本文件	把查找到的文本文件的内容显示在输入文本框
点击保存按钮	把拼音标注的结果保存到一个文本文件	含拼音标注结果的文本文件

3.2 性能规定

3.2.1精度

对所有的汉字注音，原文件不能改动，输出文件是只有拼音和空格的文本文件。例如输入为：拼音标注，则输出的注音应该是：

pin1yin1biao1zhu4

注：1 代表一声，2 代表二声，3 代表三声，4 代表四声，5 代表轻声。

除单音字外，该软件对于多音字也应有一定的辨别能力。

目标是实现 50%的多音字注音准确率，启动占用内存最多 20MB（字典 7MB）。

3.2.2时间特性要求

处理文件大小 100kB 的汉字所用时间小于 5s。

3.2.3灵活性

提供两种注音方式，一是采用命令行的方式，直接读取输入文本，将拼音标注的结果保存至输出文本，然后从输出文本中查看拼音标注的结果；二是采用图形界面的方式，在输入文本框中读取或书写文本，点击“注音”按钮将注音结果显示在输出文本框中，点击保存时保存。

3.3 输入输出要求

输入为混合汉字、英文、数字、标点还有其它符号的文本，输出为仅含拼音的文本。

3.4 数据管理能力要求

一次处理一个文本，文本大小不超过 100KB。对于文本大小其运算时间和存储空间呈线性增长。

3.5 故障处理要求

对于用户引起的各种错误（输入错误、文件不存在）进行提示。命令行模式则在提示后终止。

3.6 其他专门要求

代码规范易读，便于维护。

3.7 内部接口

3.7.1定义 defination.h

```
typedef wchar_t pychar;
typedef long int intl;
typedef int BOOL;

struct hanzi
{
   wchar_t first_zi;
   struct cizu *second_index;
}
struct cizu
{
   short ci_num;
   wchar_t other_zi;
   struct cizu *same_index;
   struct cizu *different_index;
```

}

char **pinyin;

3.7.2统计接口 accounting.c

它的下属 aseetion.c 将作为**动态链接库**进行加载，该模块将不出现在 0.1 版本以后的版本中。

intl countChinese(pychar *strFrom,intl n);//统计 strFrom 中的中文个数。长度为 n

intl countEnglish(pychar *strFrom,intl n);//统计 strFrom 中的英文个数。长度为 n

intl countNumber(pychar *strFrom,intl n);//统计 strFrom 中的数字个数。长度为 n

intl countSign(pychar *strFrom,intl n);//统计 strFrom 中的标点符号个数。长度为 n

intl countOthers(pychar *strFrom,intl n);//统计 strFrom 中的其他符号个数。长度为 n

3.7.3文件处理接口 pyFile.c

pychar *changeFormat(char * strFrom, intl n); //转换文本格式，从多字符变宽字符，strTo 指针为空，要分配。

pychar *readfile(char *strFrom,intl n);//读入文件。读入失败时返回 NULL。成功时返回文本指针。

BOOL writefile(pychar *strFrom,intl m, char *strTo,intl n);//将 strFrom 指向的长度为 n 的文本保存到 strTo（长度 m）指向的文本文件中去。失败时返回 FALSE,否则返回 TRUE。

3.7.4图形接口 pyGraphic.c

void tuxing(int argc, char *argv[]);//调用图形界面

3.7.5分词接口 particle.c

char *matchText(char *inText,int Length); //将 wchar_t 型的字符文本切分后，返回相应的拼音。拼音所占内存将在函数中得到分配

4 运行环境规定

4.1 设备

普通个人电脑，内存 512MB 及以上，cpu300Mhz 及以上，拥有键盘鼠标。

4.2 运行环境

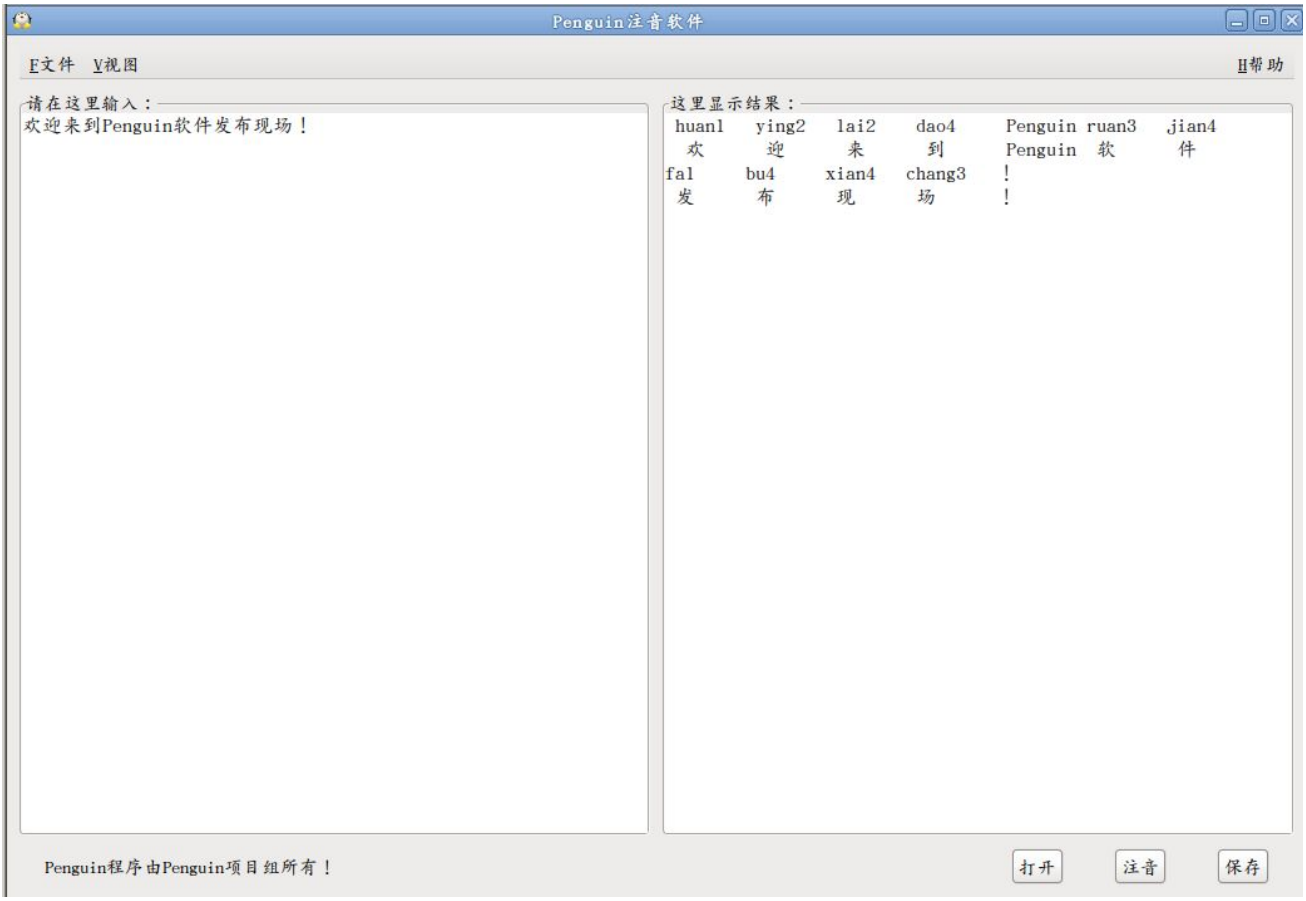
操作系统 Ubuntu10.10,图形用户界面 GTK2.0,编译程序 gcc4.4.4。

4.3 外部接口

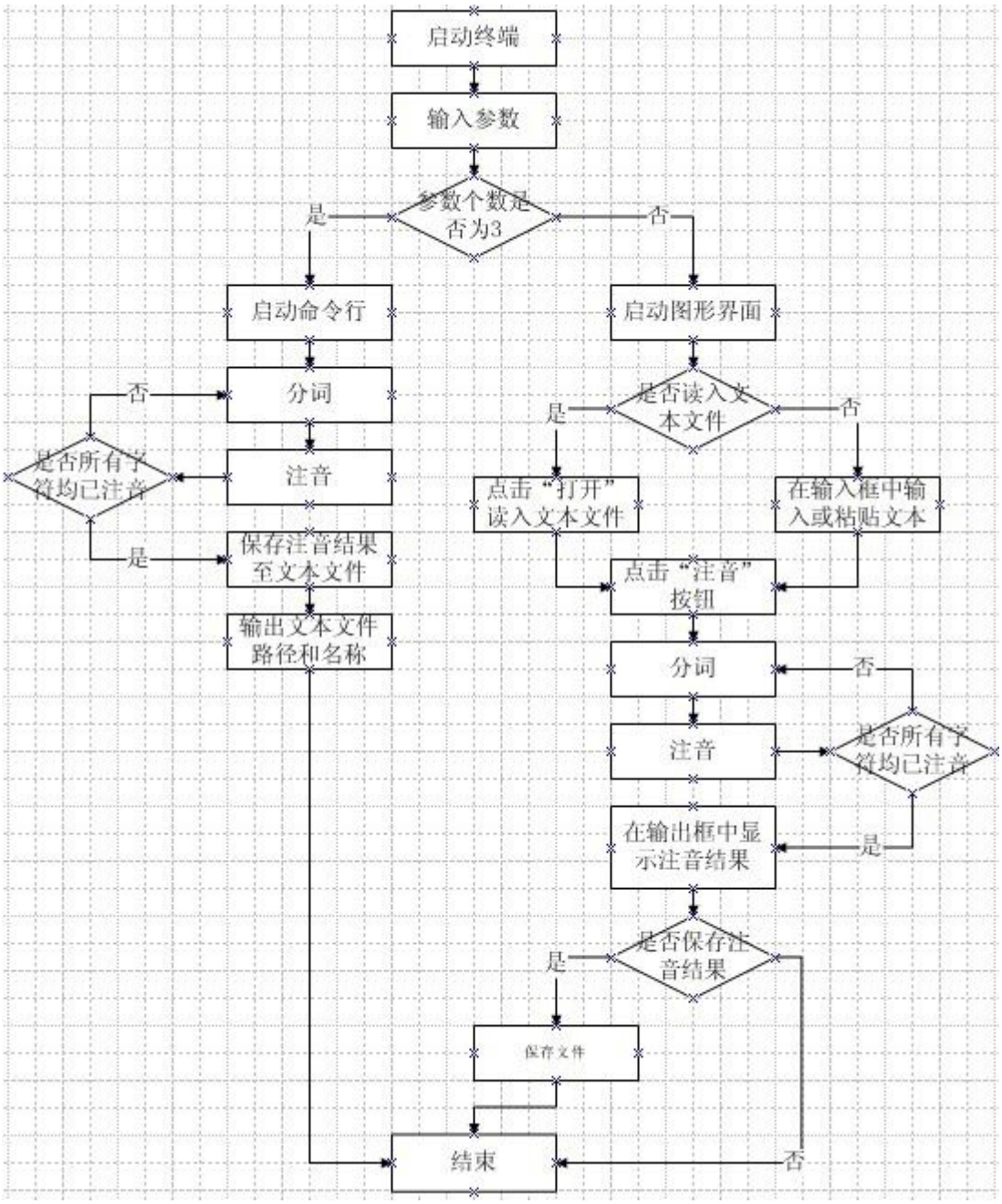
命令行启动接口,以及图形界面使用方式。

4.4 控制

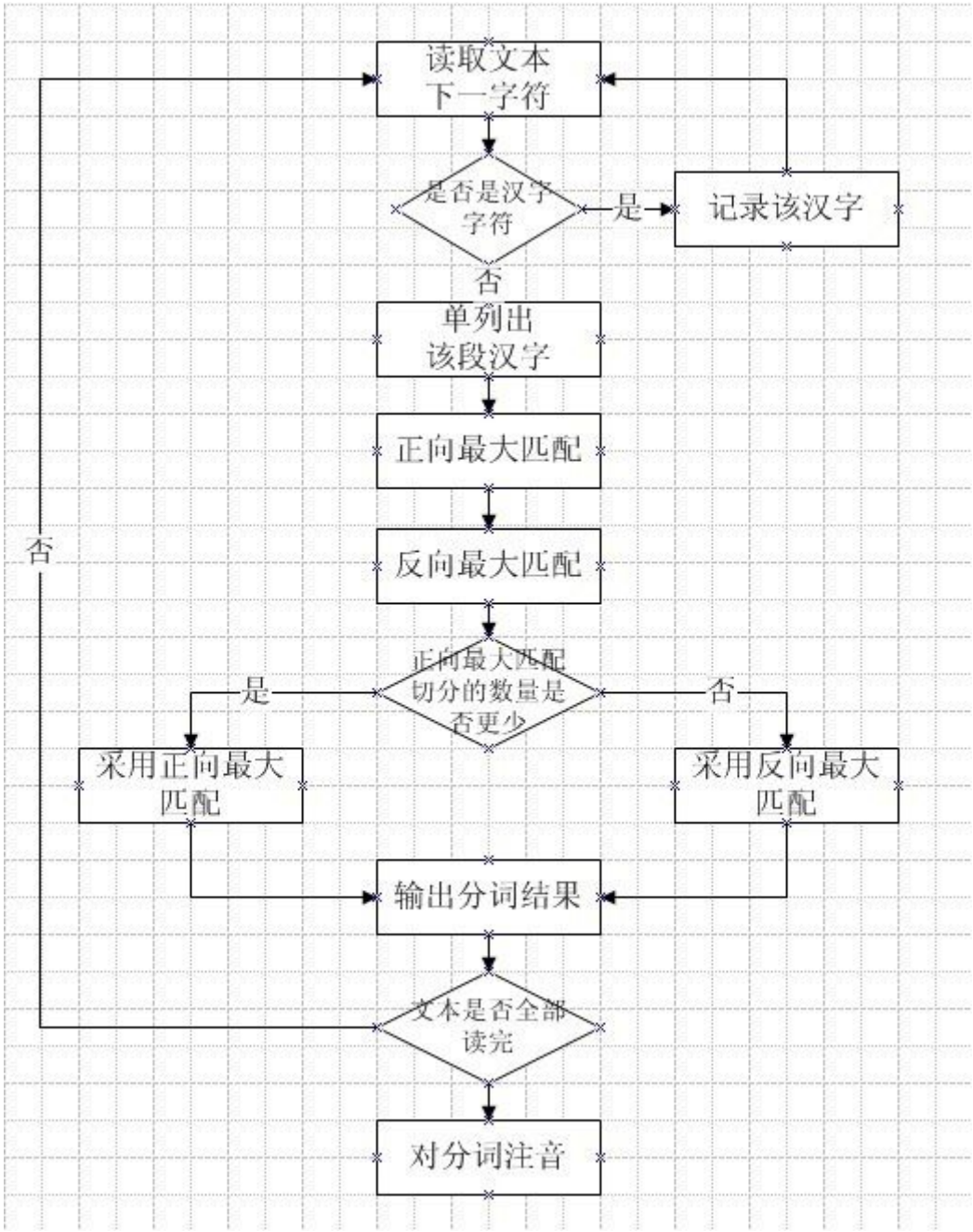
通过命令行启动，设置三个参数，输入参数后自动运行，若输入参数的个数不为 3 则启动图形界面。下图是实际结果图。



5 系统整体流程图



分词设计流程图


读取文本
下一字符
是否是汉字
字符
是
记录该汉字
否
单列出
该段汉字
正向最大匹配
反向最大匹配
正向最大匹配
切分的数量是
否更少
是
否
采用正向最大
匹配
采用反向最大
匹配
输出分词结果
文本是否全部
读完
否
对分词注音

6 测试计划

预计整体开发时间为三周，预计测试时间为四周。每次测试用例将于开发前由 QA 发起讨论编写，编写后作为测试专用文件留存，并记录版本信息。第四周将用于专门处理集成测试。

7 版本发布计划

每周都将提供可执行的小型里程碑版本。

第一周 0.1，将实现文本的统计。

第二周 0.2，将实现软件的核心功能。

第三周 0.3，将实现软件的完整逻辑。

第四周 1.0，将实现软件的集成测试。

第五周 1.1，将实现项目的收尾工作，makefile 制作等。