|  |
| --- |
| 《JSON库程序》 |
| C语言项目 说明文档 |

施笑晨

1. 提交内容

本程序共提交三个文件：

JSON.h：包含项目要求实现的接口以及JSON结构的定义

JSON.c：JSON.h中接口的实现

supportfunc.h：一些与JSON数据结构无关的函数，用来处理字符串。

1. 数据结构

本程序采用的JSON结构体包含如下成员：

int type;

struct JSON\* next;

struct JSON\* child;

char\* valuestring;

int valueint;

double valuedouble;

char\* key;

**type**:用来表示这个JSON的类型，与宏定义相配套。

**Next**:若本JSON的父JSON是JSONArray或JSONObject，则储存指向本JSON后面的JSON的指针。

**Child**:若本JSON是JSONArray或JSONObject，则储存指向本JSON的第一个子JSON的指针。

**valuestring**:若本JSON的值是字符串，则储存字符串内容。

**valueint**:若本JSON的值是布尔值，则储存布尔值内容。

**valuedouble**:若本JSON的值是数字，则储存数字的值。

**key**:若本JSON的父JSON是JSONObject，则储存本JSON的key。

1. 开发过程

在开发的过程中，注意到不算空格、换行和缩进的话JSON文件的格式个JSON字符串的格式是相同的，就确定了先编写解析JSON字符串的函数，然后通过在文件中读取JSON字符串再进行解析。JSON的结构有嵌套，于是决定使用递归的方法来编写解析函数，具体方法是：将JSON分为三种，’[‘开头的JSONArray和’{‘开头的JSONObject还有’\”’开头的字符串（最小元素），在解析字符串的过程中遇到’\”’就当做字符串解析，分情况解析出Bool、Number等类型的JSON，遇到’[‘和’{‘字符时就调用自己编写的arrayParse和objectParse函数，然后在这两个函数中再重复上面的过程。

输出的时候也是相同的思想，Bool、Number等基本类型就通过printf直接输出，遇到JSONArray和JSONObject就编写递归函数printArray和printObject来输出，在这两个函数中，先递归输出自己的子JSON，然后再递归输出自己的兄弟JSON，输出结果与项目要求相符合。

编写按格式输出到文件的函数时候曾经遇到困难，不知缩进该如何控制，后来想到使用静态变量，于是引入了static int depth这一变量，成功完成了项目要求。

1. 收获

在这次项目编写中，一度遇到很大的困哪，不知递归调用应该如何进行，后来通过对学长学姐和同学的询问和讨论，终于掌握了递归调用的方法。

这次写了很多自定义的函数，体会到了将相同代码段抽象成函数来提高代码可读性的好处，以后会继续采用这种方法。

初次使用内存申请的时候，单纯的使用了malloc而导致指针中存储着垃圾数据而出错，后来在newJSON()函数中将各个变量都赋予了初值，从而保证了不会再出类似的错误。

1. 额外文件说明

额外提供的supportfunc.h文件中包含着一些处理字符串的函数，将之包含在JSON.c中即可使用。