

中科院声学所离线识别引擎

总控端接口设计

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| V1.0 | 王超锋 |  | 2021-11-02 | 创建文件 |
|  |  |  |  |  |

目录

[1.概述 3](#_Toc16623)

[1.1 编写目的 3](#_Toc12241)

[2.总体设计 4](#_Toc6776)

[2.1 总体需求 4](#_Toc5066)

[3.总控端接口设计 5](#_Toc7794)

[3.1任务获取模块 6](#_Toc29520)

[3.1.1 任务接收接口 7](#_Toc19494)

[3.2 任务发送模块 8](#_Toc22912)

[3.2.1 任务接收接口 8](#_Toc20307)

[3.3 结果接收模块 10](#_Toc9193)

[3.3.1 结果接收接口 10](#_Toc5351)

[3.4 结果处理模块 11](#_Toc730)

[3.4.1 结果处理接口 12](#_Toc3757)

[4.代码目录 14](#_Toc24796)

[4.1 代码目录结构 14](#_Toc25932)

[4.2 主要代码文件说明 15](#_Toc4916)

[4.2.1 数据库操作及转码模块 15](#_Toc4930)

[4.2.2redis操作模块 15](#_Toc882)

[4.2.3 流程控制模块 16](#_Toc30921)

# 1.概述

1

**1.1 编写目的**

本文档主要描述总控端内部接口设计及接口中参数的数据结构，本文档的读者对象为离线识别系统的开发人员、测试人员、系统维护人员及接入识别系统的第三方业务人员，通过本文档能够从总体上了解识别系统内部的数据流向及处理方式。

本说明给出离线识别系统的接口设计说明，包括接口安全、版本兼容性、数据格式、服务端的异常处理等。

目的在于：

* 为开发人员提供依据；
* 为代码修改、维护提供条件；

# 2.总体设计

2

## 2.1 总体需求

接口需求：

1，接口参数简单明了，参数有实际意义；

2，接口功能定义明确；

3，接口的数据特性明确；

4，接口的规格和技术要求明确；

5，接口程序的数据处理逻辑清晰；

# 3.总控端接口设计

3

总控端务包含任务获取、数据发送模块、处理数据模块和接收结果返回模块。

模块间关系如下：



## 3.1任务获取模块

总控端和能力接口之间通过redis通信，总控端通过阻塞的方式从redis里获取能力接口存入redis的任务信息；

处理流程如下：



### 3.1.1 任务接收接口

THREAD\_VOID GetTaskThread(THREAD\_PVOID pvoid)

接口功能：该函数主要实现阻塞的从redis里获取任务，任务信息入库；

例如：

{"audioName":"xiaoyun.wav","audioServerIP":"192.168.0.51","audioUrl":"/home/thinkit/tempvoice","channels":"0","postfix":".xml","sid":"xiaoyun","stereo\_on":"3"}

说明：

audioName：音频名。

audioServerIP：音频落地的机器ip。

audioUrl：音频的存放目录。

channels：音频的声道信息。

postfix：获取结果的类型。

sid：音频文件的唯一性标识。

stereo\_on：双声道参数。

## 3.2 任务发送模块

总控端和各服务端之间通过socket通信，任务发送模块负责轮询任务队列数据，并发任务信息发送到对应的服务端；

处理流程如下：



### 3.2.1 任务接收接口

THREAD\_VOID TaskSendThread(THREAD\_PVOID pvoid)

接口功能：该函数主要向服务端发送任务；

例如：

message TaskInfo

{

required string serialNum =20211102-01a28fa3-00;

required string audioname=20211102-01a28fa3-00.wav;

required string audiourl =/home/thinkit/tempvoice;

required int32 channels=0;

required string language=zhn;

optional string keywordlist = keyword.txt;

optional string resultfilePath = /home/thinkit/tempvoice\_Xml ;

optional string xmlfilePath=/home/thinkit/tempvoice\_Xml ;

}

说明：

serialNum：音频文件的唯一性标识。

audioname：音频文件的名称。

audiourl ：音频的存放目录。

channels：音频的声道信息。

language：音频语种信息。

keywordlist ：关键词列表的绝对路径。

resultfilePath ：结果文本文件存放的绝对目录。

xmlfilePath:最终的xml结果文件存放的绝对路径。

## 3.3 结果接收模块

总控端和各服务端之间通过socket通信，结果接收模块负责接收各服务端返回的任务处理信息，并把接收到的任务信息存入结果队列；

处理流程如下：



### 3.3.1 结果接收接口

THREAD\_VOID ResultReceiveThread(THREAD\_PVOID pvoid)

接口功能：该函数主要实现接收服务端返回任务处理信息存入结果队列；

例如：

message ResultInfo

{

required string serialNum = 20211102-01a28fa3-00;

required int32 errorCode = 0;

optional bytes audioname=20211102-01a28fa3-00.wav;

optional bytes xmlfilePath = /home/thinkit/tempvoice\_Xml;

}

说明：

serialNum：音频文件的唯一性标识。

errorCode：结果错误码。

audioname：音频文件的名称。

xmlfilePath: xml结果存放路径。

## 3.4 结果处理模块

从结果队列中取出服务端返回的处理信息任务，判断处理返回码。失败则根据配置删除原始语音及中间结果文件更新数据库状态。成功则判断是否走完所有服务端，若走完所有服务端，则根据配置删除原始语音及中间结果文件更新数据库状态，若未走完所有服务端，则把任务信息压入下一服务端处理。

流程图如下：



### 3.4.1 结果处理接口

THREAD\_VOID ResultProcThread(THREAD\_PVOID pvoid)

接口功能：处理各服务返回任务处理结果；

例如：

message ResultInfo

{

required string serialNum = 20211102-01a28fa3-00;

required int32 errorCode = 0;

optional bytes audioname=20211102-01a28fa3-00.wav;

optional bytes xmlfilePath = /home/thinkit/tempvoice\_Xml;

}

说明：

serialNum：音频文件的唯一性标识。

errorCode：结果错误码。

audioname：音频文件的名称。

xmlfilePath: xml结果存放路径。

# 4.代码目录

4

## 4.1 代码目录结构

OfflineSystemClientSrc

├── include //依赖头文件目录

├── lib //依赖库目录

├── src

│  ├── CPPSocket.cpp //socket通信

│  ├── CPPSocket.h

│  ├── CPPTcpSocket.cpp //socket通信

│  ├── CPPTcpSocket.h

│  ├── cfg.ini //配置文件

│  ├── configure\_code.cfg //转码配置文件

│  ├── client\_multiserver.cpp //主函数，任务获取、结果接收、处理、任务发送线程

│  ├── makefile //编译脚本

│  ├── MsgInfo.pb.cc //protobuf文件

│  ├── MsgInfo.pb.h

│  ├── porting.cpp

│  ├── porting.h

│  ├── myqueue\_template.cpp //队列操作相关函数

│  ├── myqueue\_template.h

│  ├── RedisClient.cpp //redis操作相关

│  ├── RedisClient.h

│  ├── tools.cpp //数据库函数，转码

│  ├── tools.h

## 4.2 主要代码文件说明

### 4.2.1 数据库操作及转码模块

tools.cpp该cpp文件中主要包含数据库操作相关函数的实现，转码函数及加载配置文件函数的实现。

### 4.2.2redis操作模块

RedisClient.cpp该cpp主要包含redis相关函数的实现。

### 4.2.3 流程控制模块

client\_multiserver.cpp该cpp功能为日志文件初始化，加载配置文件，任务获取、结果接收、结果处理、任务发送线程的起动及具体处理逻辑。