

中科院声学所离线SOX

概要设计说明

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| V1.0 | 王超锋 |  | 2021-11-17 | 创建文件 |
|  |  |  |  |  |

目录

[1.概述 3](#_Toc18205)

[1.1简介 3](#_Toc30439)

[1.2 读者对象 3](#_Toc23757)

[1.3术语定义 4](#_Toc22715)

[2.总体设计 5](#_Toc29627)

[2.1 总体需求 5](#_Toc28495)

[2.2 运行环境 6](#_Toc27658)

[2.2.1 软件环境 6](#_Toc26599)

[2.2.2 硬件环境 6](#_Toc7376)

[2.3系统总体结构设计 7](#_Toc18566)

[3.模块设计 8](#_Toc23109)

[3.1 sox基本使用 8](#_Toc18897)

[3.2 sox转码流程 12](#_Toc8834)

[4.测试设计 19](#_Toc8550)

[4.1 配置文件设计 19](#_Toc19716)

[4.2 测试功能点 20](#_Toc18363)

[5.可靠性设计 21](#_Toc15255)

[6.扩展性设计 22](#_Toc24046)

[7.维护性设计 23](#_Toc16217)

[8.易用性设计 24](#_Toc15160)

# 1.概述

1

**1.1简介**

Sox项目是由Lance Norskog创立的，后来被众多的开发者逐步完善，现在已经能够支持很多种声音文件格式和声音处理效果。基本上常见的声音格式都能够支持。更加有用的是，Sox能够进行声音滤波、采样频率转换。

SoX是Sound eXchange的简称，当ffmpeg不支持某些语音格式的转码时，负责对这些特殊的语音进行转码。

SoX可以读写最流行格式的音频文件，并且可以对音频进行任意操作。它可以将多个输入源合成音频。在许多系统，它可以作为一个大众的音频播放器或者多轨录音机。它也可以进行少量的将输入文件切割成多个文件。

## 1.2 读者对象

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SOX的架构形式及数据流向。

本说明给出FFMPEG的设计说明，包括最终实现的系统必须满足的功能、性能、接口、附属测试工具程序及设计约束等。

目的在于：

* 为开发人员提供依据；
* 为代码修改、维护提供条件；

## 1.3术语定义

语音识别（Automatic Speech Recognition），简称（ASR）。

# 2.总体设计

2

## 2.1 总体需求

功能需求：对语音进行转码；

性能需求：转码耗时低；

接口需求：接口参数简单明了，接口中的每个参数都要有实际意义，保证接口调用流程清晰。

系统验证工具：提供系统完整性，功能正确性的验证工具；

## 2.2 运行环境

### 2.2.1 软件环境

|  |  |  |
| --- | --- | --- |
| 分类 | 名称 | 版本 |
| 操作系统 | Centos | 7.0以上 |
| 数据库 | Redis | 3.2.0 |
| 能力接口 | Tomcat | 9.0.35 |

### 2.2.2 硬件环境

|  |  |  |
| --- | --- | --- |
| 服务器 | 最低配置 | 推荐配置 |
| redis服务器 | CPU：1  内存：256M | CPU：1  内存：1G |
| 识别服务器 | CPU：4  内存：10G  磁盘：20G | CPU：10  内存：15G  磁盘：30G |

## 2.3系统总体结构设计

SoX 可以读取和写入常见格式的音频文件，并在此过程中选择性的加入一些声音效果。它可以组合多个输入源及合成音效，在许多系统上也可以作为音频播放器或多轨录音机使用。sox整个流程如下图所示。



# 3.模块设计

3

## 3.1 sox基本使用

1、libavformat获取音频文件的元数据：

soxi 或 sox --i 命令可以通过分析音频文件的文件头，获取其元数据（如通道数、采样率、编码等）。

soxi Faded.wav

Input File : 'Faded.wav'

Channels : 2

Sample Rate : 44100

Precision : 16-bit

Duration : 00:03:32.63 = 9376836 samples = 15947 CDDA sectors

File Size : 37.5M

Bit Rate : 1.41M

Sample Encoding: 16-bit Signed Integer PCM

1. 获取音频的统计信息：

可以使用 sox <inputfile> -n stat 命令获取某音频文件的统计信息。示例如下：

sox Faded.wav -n stat

Samples read: 18753672

Length (seconds): 212.626667

Scaled by: 2147483647.0

Maximum amplitude: 0.977417

Minimum amplitude: -0.977478

Midline amplitude: -0.000031

Mean norm: 0.229415

Mean amplitude: -0.000006

RMS amplitude: 0.302594

Maximum delta: 1.765564

Minimum delta: 0.000000

Mean delta: 0.202369

RMS delta: 0.273320

Rough frequency: 6339

Volume adjustment: 1.023

1. 播放与录制：

play 和 rec 命令提供了最基本的播放和录制功能。  
 播放：$ play existing-file.wav  
 录制：$ rec new-file.wav

上述命令等同于 sox 命令的如下形式：  
 $ sox existing-file.wav −d（播放）和 sox −d new-file.wav（录制）  
 其中 -d 选项用于指定播放或录制时使用的音频设备，不指定时则表示使用默认设备。

1. 音频格式转换：

文件格式类型

对于音频数据格式的描述，主要通过以下 4 种属性：

采样率（sample rate）：指声音由模拟信号转换成数字信号的过程中，每秒从连续信号中提取的用于组成离散信号的样本个数。

音频CD所用的采样率为 44100 Hz，数字音频磁带和许多计算机系统使用 48000 Hz，专业级音频系统通常使用 96000 Hz。

采样大小（sample size 或 Precision）：音频采样时用于存储每个样本的数据位数（bits）。

编码格式（data encoding）：即每个音频样本的表示（即“编码”）方式。常用的编码类型包括 floating-point、μ-law、ADPCM、singed-integer PCM、MP3 和 FLAC 等。

通道（channel）：即文件中包含的音频通道的数量。其中单声道（mono）和双声道（stereo）是最常见的两种，“环绕声”音频（Surround sound）通常包含六个或更多声道。

此外，音频文件还使用比特率（Bit Rate）表示一个单位时间内编码音频信号占用的存储空间大小， 它的数值一般取决于所有的上述四个参数。

MP3 编码的立体声音乐通常具有 128-196kbps 的比特率， FLAC 编码的立体声音乐通常具有 550-760kbps 的比特率。

格式转换

形式最简单的 sox 命令即使用两个文件名作为参数，如：

sox Faded.wav Faded.mp3 ：将 Faded.wav 文件的格式由 wav 转为 mp3

上述命令执行时，SoX 会先从 Faded.wav 文件中读取音频数据，再将其输出到 Faded.mp3 文件中。而 SoX 程序会根据参数中文件名的后缀推断出相应的格式，并在复制音频数据的过程中自动进行转码。

SoX 可以处理 self-describing 和 raw 格式的音频文件。

self-describing 格式（如 WAV、FLAC、MP3）的文件包含一个用于描述信号和编码属性的文件头，而 raw 或 headless 格式的音频则不包含这些信息。

所以当 raw 格式的音频作为输入文件时，需要在 sox 命令的格式选项里指定其信号和编码属性。

常用的音频格式选项：

选项 描述

-b, --bits BITS 每个编码样本占用的数据位数

-c, --channels CHANNELS 音频文件包含的通道数

-e, --encoding ENCODING 音频文件的编码类型

-r, --rate RATE 音频文件的采样率

-t, --type FILE-TYPE 音频文件的文件类型

上述选项适用于输入或输出文件，主要用于说明 raw（或 headless）文件作为输入时的格式信息，或格式转换时指定输出文件的具体参数。

sox −r 48k −e float −b 32 −c 2 input.raw output.wav

将某个特定的 raw 格式的音频文件转换为 wav 格式。

sox Faded.wav Faded.raw

将音频文件 Faded.wav 转为 raw 格式。

play -r 44800 -b 16 -e signed-integer -c 2 Faded.raw

播放 raw 格式的音频文件。

sox Faded.wav -c 1 Faded-mono.wav

将 Faded.wav 文件转换成单声道（-c 1）后输出。

## 3.2 sox转码流程

Sox具体处理流程如下：



关键函数说明：

sox\_init()：初始化全局参数

sox\_open\_mem\_read：打开文件；

sox\_open\_men\_write：打开输出文件；

sox\_create\_effects\_chain：构建一个效果器链；

sox\_create\_effect：创建一个简单的效果输入文件；

sox\_add\_effect：添加效果器到效果器链；

sox\_flow\_effects：效果器运行；

sox\_delete\_effects\_chain：释放资源；

sox\_close：关闭文件；

sox\_quit：退出；

主要函数介绍：

函数原型：

int LSX\_API sox\_init(void);

参数说明：

返回值：

int类型。取值及说明如下：

0 ：成功

非0 ：失败

功能：

初始化全局参数。

函数原型：

LSX\_RETURN\_OPT sox\_format\_t \* LSX\_API sox\_open\_mem\_read(

LSX\_PARAM\_IN\_BYTECOUNT(buffer\_size) void \* buffer,

size\_t buffer\_size,

LSX\_PARAM\_IN\_OPT sox\_signalinfo\_t const \* signal,

LSX\_PARAM\_IN\_OPT sox\_encodinginfo\_t const \* encoding,

LSX\_PARAM\_IN\_OPT\_Z char const \* filetype

);

参数说明：

LSX\_PARAM\_IN\_BYTECOUNT(buffer\_size) void \* buffer 数据指针

size\_t buffer\_size, 数据大小

LSX\_PARAM\_IN\_OPT sox\_signalinfo\_t const \* signal 信号量

LSX\_PARAM\_IN\_OPT sox\_encodinginfo\_t const \* encoding 编码信息

LSX\_PARAM\_IN\_OPT\_Z char const \* filetype 文件类型

返回值：sox\_format\_t指针

功能：

打开输入文件。

函数原型：

LSX\_RETURN\_OPT sox\_format\_t \* LSX\_API sox\_open\_mem\_write(

LSX\_PARAM\_OUT\_BYTECAP(buffer\_size) void \* buffer,

LSX\_PARAM\_IN size\_t buffer\_size,

LSX\_PARAM\_IN sox\_signalinfo\_t const \* signal,

LSX\_PARAM\_IN\_OPT sox\_encodinginfo\_t const \* encoding,

LSX\_PARAM\_IN\_OPT\_Z char const \* filetype

LSX\_PARAM\_IN\_OPT sox\_oob\_t const \* oob

);

参数说明：

LSX\_PARAM\_OUT\_BYTECAP(buffer\_size) void \* buffer 数据指针

size\_t buffer\_size, 数据大小

LSX\_PARAM\_IN\_OPT sox\_signalinfo\_t const \* signal 信号量

LSX\_PARAM\_IN\_OPT sox\_encodinginfo\_t const \* encoding 编码信息

LSX\_PARAM\_IN\_OPT\_Z char const \* filetype 文件类型

LSX\_PARAM\_IN\_OPT sox\_oob\_t const \* oob 输出数据指针

返回值：sox\_format\_t指针

功能：

打开输出文件。

函数原型：

LSX\_RETURN\_OPT sox\_effects\_chain\_t \* LSX\_API sox\_create\_effects\_chain(

LSX\_PARAM\_IN sox\_encodinginfo\_t const \* in\_enc,

LSX\_PARAM\_IN sox\_encodinginfo\_t const \* out\_enc

);

参数说明：

LSX\_PARAM\_IN sox\_encodinginfo\_t const \* in\_enc 效果器指针

LSX\_PARAM\_IN sox\_encodinginfo\_t const \* out\_enc效果器结果指针

返回值：sox\_effects\_chain\_t指针

功能：

构建效果器链。

函数原型：

LSX\_RETURN\_OPT sox\_effect\_t \* LSX\_API sox\_create\_effect(

LSX\_PARAM\_IN sox\_effect\_handler\_t const \* eh

);

参数说明：

LSX\_PARAM\_IN sox\_effect\_handler\_t const \* eh 效果器信息指针

返回值：sox\_effect\_t指针

功能：

构建简单效果器。

函数原型：

int LSX\_API sox\_add\_effect(

LSX\_PARAM\_INOUT sox\_effects\_chain\_t \* chain,

LSX\_PARAM\_INOUT sox\_effect\_t \* effp,

LSX\_PARAM\_INOUT sox\_signalinfo\_t \* in,

LSX\_PARAM\_IN sox\_signalinfo\_t const \* out

);

参数说明：

LSX\_PARAM\_INOUT sox\_effects\_chain\_t \* chain 效果器链指针

LSX\_PARAM\_INOUT sox\_effect\_t \* effp 效果器指针

LSX\_PARAM\_INOUT sox\_signalinfo\_t \* in Input format

LSX\_PARAM\_IN sox\_signalinfo\_t const \* out Output format

返回值：

int类型。取值及说明如下：

0 ：成功

非0 ：失败

功能：

添加效果器到效果器链。

函数原型：

int LSX\_API sox\_flow\_effects(

LSX\_PARAM\_INOUT sox\_effects\_chain\_t \* chain,

LSX\_PARAM\_IN\_OPT sox\_flow\_effects\_callback callback,

LSX\_PARAM\_IN\_OPT void \* client\_data

);

参数说明：

LSX\_PARAM\_INOUT sox\_effects\_chain\_t \* chain 运行效果器链指针

LSX\_PARAM\_IN\_OPT sox\_flow\_effects\_callback callback 回调

LSX\_PARAM\_IN\_OPT void \* client\_data 回调结果数据

返回值：

int类型。取值及说明如下：

0 ：成功

非0 ：失败

功能：

运行效果器。

函数原型：

void LSX\_API sox\_delete\_effects\_chain(

LSX\_PARAM\_INOUT sox\_effects\_chain\_t \*ecp

);

参数说明：

LSX\_PARAM\_INOUT sox\_effects\_chain\_t \*ecp 效果器链指针

返回值：

功能：

释放效果器链资源。

函数原型：

int LSX\_API sox\_close(

LSX\_PARAM\_INOUT sox\_format\_t \* ft

);

参数说明：

LSX\_PARAM\_INOUT sox\_format\_t \* ft 效果器指针

返回值：

int类型。取值及说明如下：

0 ：成功

非0 ：失败

功能：

关闭效果器。

函数原型：

int LSX\_API sox\_quit(void);

参数说明：

返回值：

int类型。取值及说明如下：

0 ：成功

非0 ：失败

功能：

退出。

# 4.测试设计

4

## 4.1 配置文件设计

设置配置文件，方便使用不同控制逻辑，支持不同的功能及结果输出，配置参数实现如下目的：

* 通过参数控制支持特殊功能

配置主要参数如下表：

|  |  |  |
| --- | --- | --- |
| 参数 | 取值 | 说明 |
| inwav | 字符串 | 待转码语音 |
| -b |  | 每个编码样本占用的数据位数 |
| -c |  | 音频文件包含的通道数 |
| -r |  | 音频文件的采样率 |
| outwav | 字符串 | 转码后语音 |

## 4.2 测试功能点

根据设计需求，对服务必须满足的功能点进行验证测试。

|  |  |
| --- | --- |
| 测试功能 | 期望结果 |
| 使用不同的语音进行转码 | 正确配置参数，转码后语音正常。 |

# 5.可靠性设计

5

离线系统SOX设计考虑了实际应用场景的复杂性和输入的多样性的情况，内部加入了大部分实际情况中会出现的异常情况的处理方案，保证稳定性和可靠性。

# 6.扩展性设计

6

系统本身遵循模块化设计，可以随时添加定制化功能，满足不同应用场景的需要。

# 7.维护性设计

7

离线系统为了方便后期的引擎维护，采用标准c/c++函数同时在较低编译器上进行编译

# 8.易用性设计

8

系统安装部署时有一键安装部署脚本，只需要执行脚本即可安装,减少操作量。系统启停均有对应的脚本，实现自动启动、停止服务。