

语音助手程序结构

May 7, 2013

目录

1	语音合成	3
2	语音识别	3
3	录音	3
4	命令解析	4
5	执行操作	4
5.1	窗口激活	4
5.2	发送按键	5
5.3	调用函数	5
5.4	执行命令	5
6	命令配置目录	5

语音合成与识别使用科大讯飞的语音库，录音使用的是 alsa 库，按键事件的发送使用的是 xlib 库，窗口激活使用的是 wmctrl 命令，进程是否存在使用 ps aux 进行判断。

1 语音合成

代码文件：

my_qtts.h – 头文件

my_qtts.cpp – 具体实现

语音合成接口：

```
1 int SpeechSynth(const char *text, const char *outFile)
2 text 为需要合成的文本信息
3 outFile 为合成输出的音频文件
```

2 语音识别

代码文件：

my_qisr.h – 头文件

my_qisr.cpp – 具体实现

语音合成接口：

```
1 int SpeechRecog(const char *inFile, const char *
    outFile)
2 inFile 为需要识别的音频文件
3 outFile 为识别输出的文本文件
```

3 录音

代码文件：

my_alsa.h – 头文件

my_alsa.cpp – 具体实现

语音合成接口：

```
1 int MySndRecord(int timenum, const char *output_file)
2 timenu 为录音的秒数
3 output_file 为录音输出的音频文件
```

4 命令解析

代码文件：

执行命令：

exec_cmd.h — 头文件

exec_cmd.cpp — 具体实现

解析语音：

parse_speech.h — 头文件

parse_speech.cpp — 头文件

解析思想：

struct process_info cur_process; 用来存储当前进程的信息，即进程名、标题、配置文件和命令类型，其初始化为空。

语音输入后，得到识别后的文本。开始检测 cur_process 是否设置。

1、若 cur_process 尚未设置，则执行下面的步骤：

(1)、搜索 index.conf 文件，若有匹配，则对 cur_process 进行设置。然后检测此进程是否打开，若打开则将其激活，否则搜索 open.conf 文件将其打开。若无匹配则返回。

2、若 cur_process 已经设置，则搜索此进程对应的配置文件，若有匹配则执行相关的命令，否则搜索 command.conf，若有匹配则执行相关的命令，否则执行上面的第(1)步。

5 执行操作

my_xlib.h 为 X11 相关操作头文件，即窗口激活与按键事件发送。

5.1 窗口激活

使用 wmctrl -lx 命令获得打开窗口信息，然后遍历匹配其结果，得到对应的窗口 ID，然后激活窗口。

详见 activate_win.cpp

其接口为：

```
1 void set_focus()
```

5.2 发送按键

详见 send_keys.cpp

其接口为：

```
1 void exec_command(int *keys)
2 keys    为解析得到按键数组
```

5.3 调用函数

通过调用应用程序提供的接口，来达到想要的效果。

详见 exec_func.h

5.4 执行命令

使用 system() 函数执行命令。

6 命令配置目录

默认索引文件：

index.conf

格式为：语音命令进程名窗口标题配置文件动作类型（如：快捷键、命令、函数）

程序打开命令文件：

open.conf

格式为：程序名命令

各应用程序配置文件：

firefox.conf

terminal.conf

deepin-music-player.conf

.....

格式为：语音命令对应的动作（如：快捷键、命令、函数）

其他命令文件：

command.conf

格式为：语音命令对应的动作（命令）