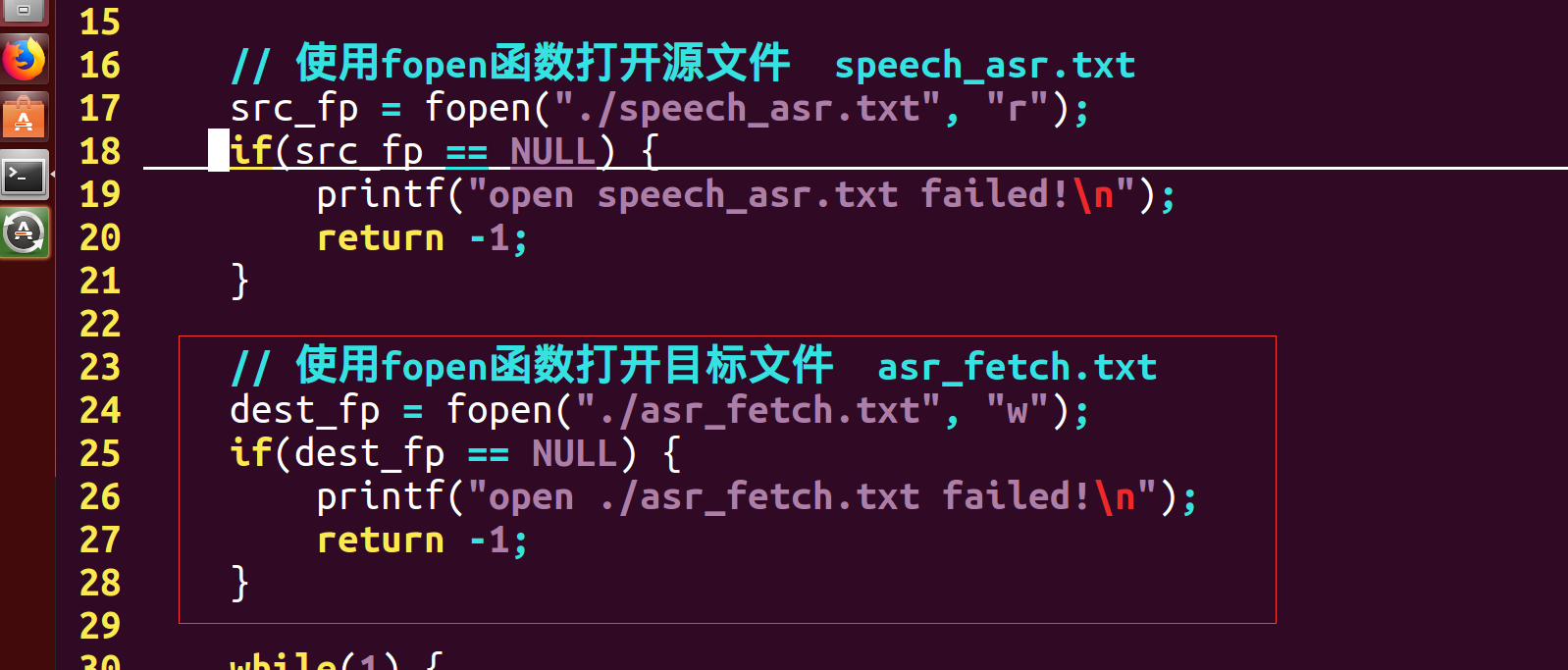
1. README.txt文件

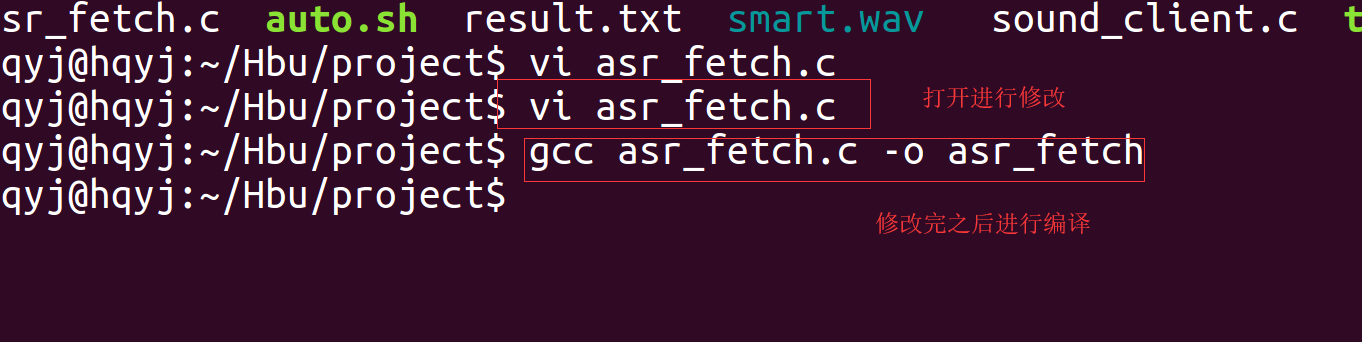


1. 修改语音识别提取关键字的代码

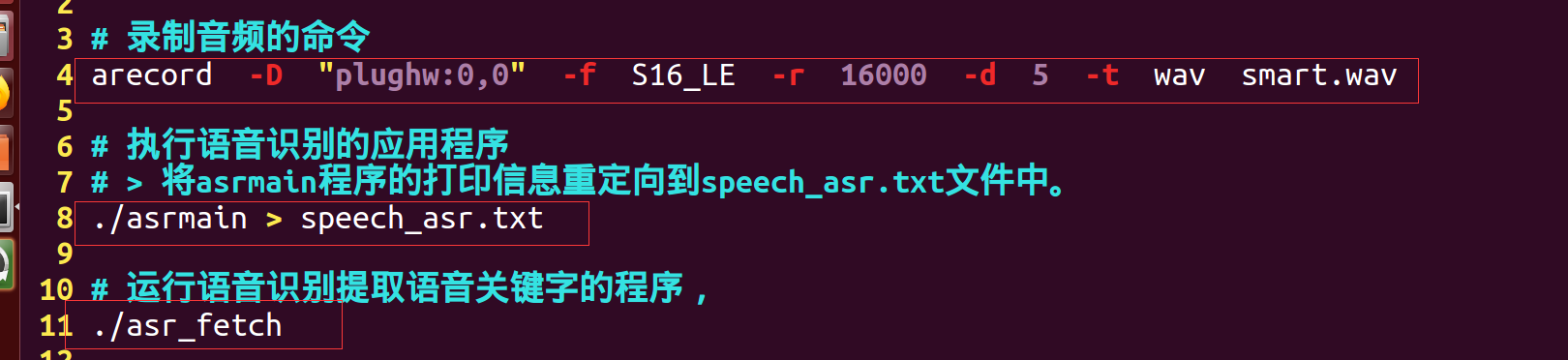
1》asr\_fetch.c,修改文件中的以下内容



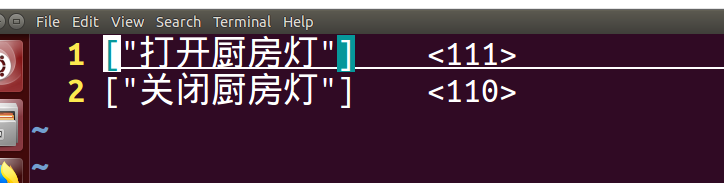
2》重新进行编译



3》可以进行单步测试，查看结果



1. 手动去创建command.txt控制语句及相关命令的文件，填充内容的格式如下：



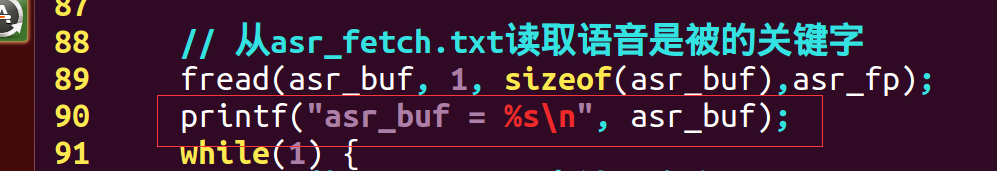
根据我的《命令格式定义.xlsx》去添加即可。

1. 修改sound\_client.c客户端的代码，
2. #include <stdio.h>
3. #include <sys/types.h>
4. #include <sys/socket.h>
5. #include <netinet/in.h>
6. #include <arpa/inet.h>
7. #include <stdlib.h>
8. #include <unistd.h>
9. #include <string.h>
10. *// 接收socket函数的返回值，*
11. int sockfd **=** 0**;**
12. *// 定义客户端链接服务器地址的结构体变量*
13. struct sockaddr\_in server\_addr**;**
14. *// 发送数据缓冲区*
15. char send\_buf**[**4**]** **=** ""**;**
16. *// 接收数据的缓冲区*
17. char recv\_buf**[**128**]** **=** **{**0**};**
18. // 语音识别的缓冲区
19. char asr\_buf[128] = {0};
20. // 命令码的缓冲区
21. char cmd\_buf[128] = {0};
22. int retval;
23. char \*retval\_p = NULL;
24. char \*str\_start = NULL, \*str\_end = NULL;
25. int main(int argc, const char \*argv[])
26. {
27. FILE \*asr\_fp = NULL;
28. FILE \*cmd\_fp = NULL;
29. FILE \*res\_fp = NULL;
30. char buffer[128] = {0};
31. /\*
32. \* ip地址：
33. \* 使用wifi上网查看wlan的IP地址
34. \* 使用有线网上网查看以太网的IP地址
35. \* 端口号：8888
36. \* \*/
37. if (argc != 3) {
38. printf("commond : %s <windows ip addr> <port number>\n", argv[0]);
39. return -1;
40. }
41. // 打开语音识别提取关键字的文件
42. asr\_fp = fopen("./asr\_fetch.txt", "r");
43. if (asr\_fp == NULL) {
44. perror("asr\_fetch.txt");
45. return -1;
46. }
47. // 打开命令码的文件
48. cmd\_fp = fopen("./command.txt", "r");
49. if(cmd\_fp == NULL) {
50. perror("commond.txt");
51. return -1;
52. }
53. // 打开存放客户端从接收服务器数据的文件
54. res\_fp = fopen("./result.txt", "w");
55. if (res\_fp == NULL) {
56. perror("result.txt");
57. return -1;
58. }
60. // 使用socket函数创建网络套接字
61. sockfd = socket(AF\_INET, SOCK\_STREAM, 0);
62. if (sockfd == 0) {
63. perror("socket");
64. return -1;
65. }
66. // 对server\_addr变量初始化
67. memset(&server\_addr, 0, sizeof(server\_addr));
68. server\_addr.sin\_family = AF\_INET;
69. server\_addr.sin\_port = htons(atoi(argv[2]));
70. server\_addr.sin\_addr.s\_addr = inet\_addr(argv[1]);
71. // 连接服务器端
72. retval = connect(sockfd, (struct sockaddr \*)&server\_addr, sizeof(server\_addr));
73. if (retval == -1) {
74. perror("connect");
75. return -1;
76. }
77. // 从asr\_fetch.txt读取语音是被的关键字
78. fread(asr\_buf, 1, sizeof(asr\_buf),asr\_fp);
80. while(1) {
81. // 从commond.txt中读取命令码
82. retval\_p = fgets(cmd\_buf, sizeof(cmd\_buf), cmd\_fp);
83. if (retval\_p == NULL) {
84. break;
85. }
86. // 提取命令字符串
87. str\_start = strstr(cmd\_buf, "[\"");
88. str\_end = strstr(cmd\_buf, "\"]");
89. if (str\_start != NULL && str\_end != NULL) {
90. // 字符串的拷贝
91. strncpy (buffer, str\_start+2, str\_end-(str\_start+2));
92. // 字符串的比较
93. retval = strcmp(asr\_buf, buffer);
94. if(retval == 0) {
95. // 提取命令字符串对应的命令码
96. str\_start = strstr(cmd\_buf, "<");
97. str\_end = strstr(cmd\_buf, ">");
98. if(str\_start != NULL && str\_end != NULL) {
99. // 拷贝命令码到send\_buf
100. strncpy(send\_buf, str\_start+1, str\_end-(str\_start+1));
101. // 客户端给服务器端发送数据
102. retval = send(sockfd, send\_buf, 3, 0);
103. if (retval == -1) {
104. perror("send");
105. return -1;
106. } else {
107. memset(send\_buf, 0, sizeof(send\_buf));
108. }
109. // 客户端从服务器端接收数据
110. retval = recv(sockfd, recv\_buf, sizeof(recv\_buf), 0);
111. if(retval == -1) {
112. perror("recv");
113. return -1;
114. } else {
115. fwrite(recv\_buf, 1 , sizeof(recv\_buf), res\_fp);
116. memset(recv\_buf, 0, sizeof(recv\_buf));
117. }
118. }
119. }
121. }
123. }
124. // 关闭客户端
125. close(sockfd);
127. return 0;
128. }
129. 修改auto.sh

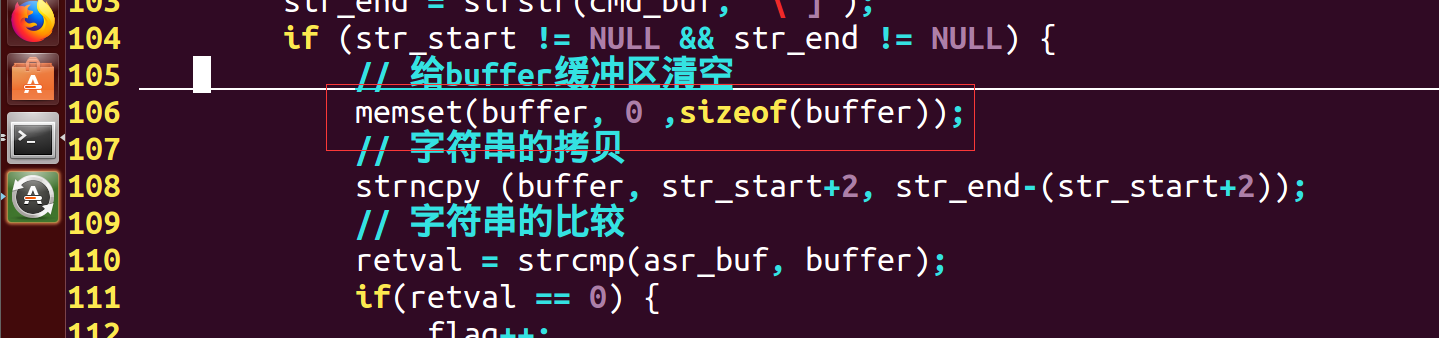


修正sound\_client.c

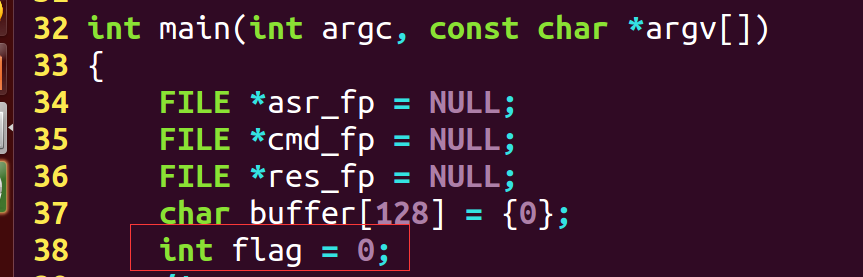
1. 打印识别的语音

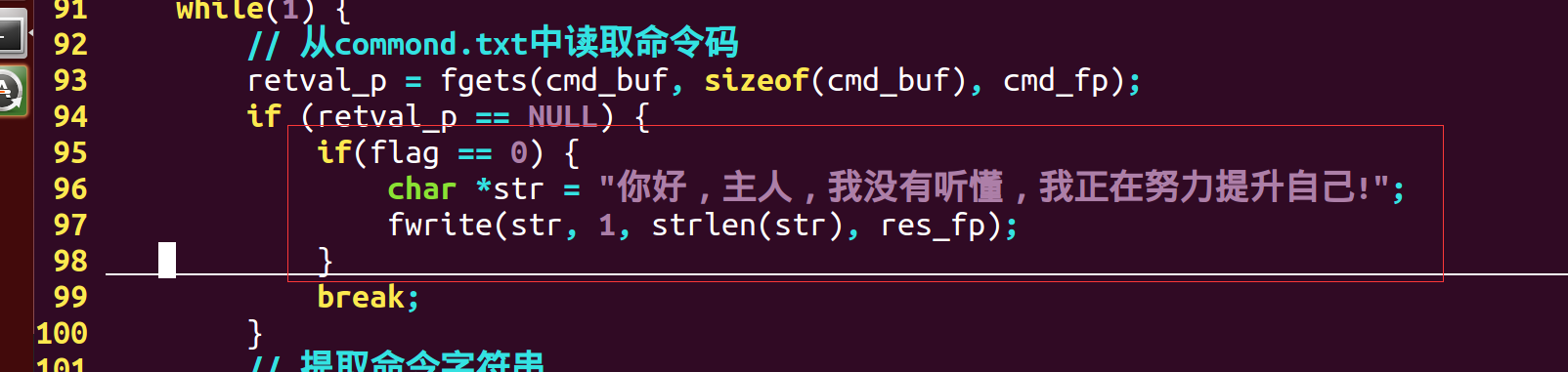


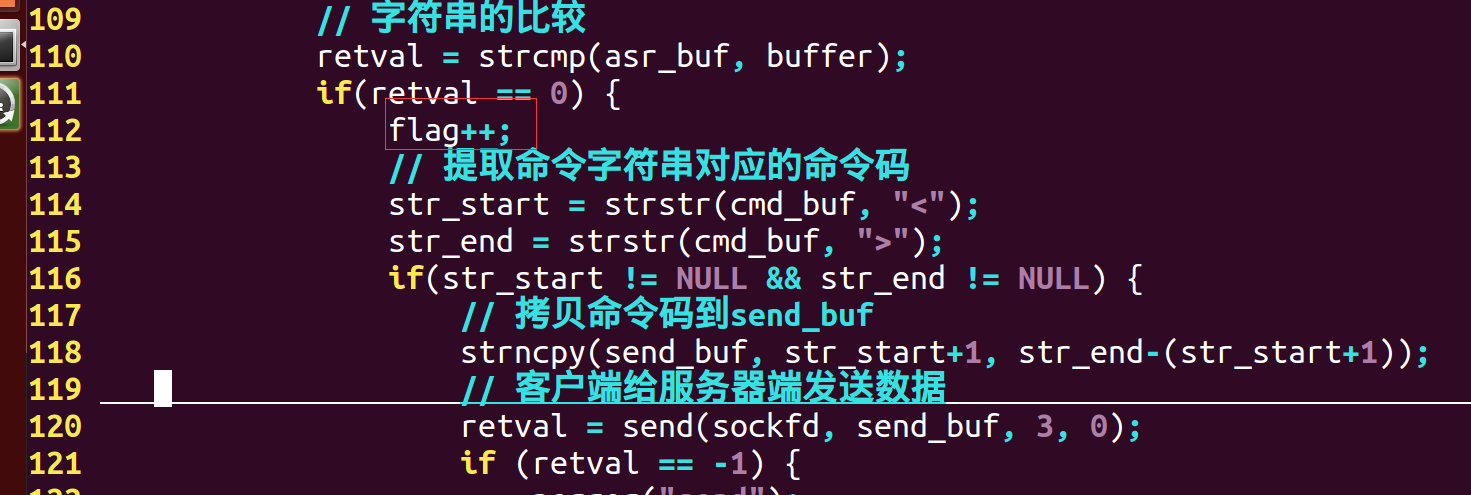
2. 清除buffer缓冲区



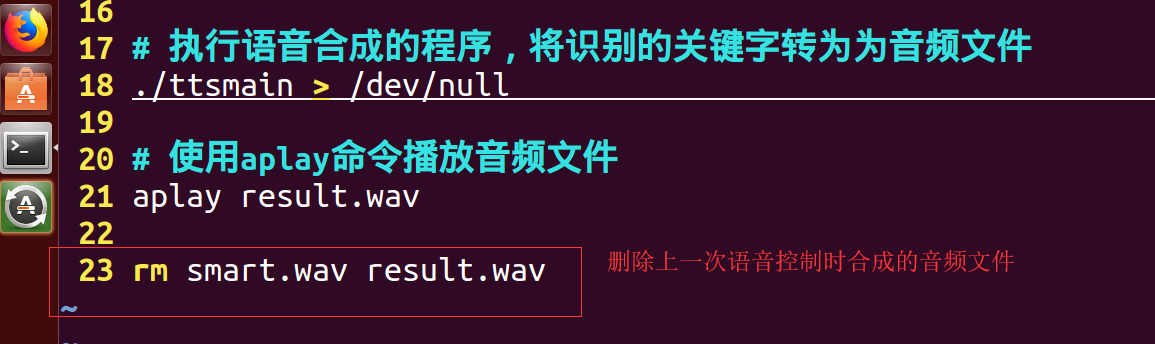
3. 语音识别错误，或者没有说语音的，播放的是上一次的语音文件







切记，修改完代码之后要重新编译。



2. 当执行auto.sh是先播放一段问候的语音，然后在说控制的命令。

1》可以使用arecord命令提前录制一个awaken.wav的音频文件

arecord -D "plughw:0,0" -f S16\_LE -r 16000 -d 5 -t wav awaken.wav

或者

使用ttsmain可执行程序将一段文字合成一段音频文件。

打开result.txt文件，将里边的内容清空，输入一段问候的语音，比如：“主人，你好，我在呢，请讲。” ，保存退出。然后执行./ttamain生成result.wav的音频文件，使用mv 命令对result.wav音频文件进行重命名，命名为awaken.wav。

2》修改auto.sh文件，添加以下信息。



3》修改完毕之后，重新执行auto.sh进行测试。