

```

1 // 文件: errlog.c
2 // 内容: 错误处理模块( C 语言版 )
3 // 作者: WXQ#2018
4
5 /* 该模块包括:
6  *   报错错误的操作
7  *   将显示到 stdout 上的信息写入指定文件
8 */
9
10 #include <stdio.h>
11 #include <stdlib.h> // exit()
12 #include <stdarg.h> // for va_*
13
14 #include "errlog.h"
15
16 #include "ui.h"
17
18 static FILE* fpLog =NULL;
19 static FILE* fpErr =NULL;
20
21
22 void CloseError()
23 {
24     if(NULL != fpLog) fclose(fpLog); fpLog = NULL;
25     if(NULL != fpErr) fclose(fpErr); fpErr = NULL;
26 }
27
28 void InitError()
29 {
30     fpLog = fopen(LOGFILE_PATH, "w");
31     fpErr = fopen(ERRFILE_PATH, "w");
32     atexit( CloseError ); // CloseError() 将被自动调用
33 }
34
35 void error_msg(int line, const char *descrip, const char *string)
36 {
37     FILE* fp = (fpErr == NULL) ? stderr : fpErr;
38     char msg[1025];
39     snprintf(msg, sizeof(msg), "Error in line %d:%s %s\n", line, descrip ,string);
40
41     if(fp != stderr) fprintf(fp,msg);
42
43     ShowMessage(1, msg);
44 }
45
46 void logPrint(const char* format, ...)
47 {
48     va_list args;
49     FILE* fp = (fpLog == NULL) ? stdout : fpLog;
50
51     va_start(args, format);
52     vfprintf(fp, format, args);
53     va_end(args);
54 }
55
56

```