链滴

Thinking——C 模拟 Exception

作者：salamander

原文链接：https://ld246.com/article/1530243015453

来源网站：链滴

异常处理

C中没有exception，所以我们可以用函数返回值来判断错误类型。但有时候又希望在顶层能统一处理误，让代码更简洁一点。其实第一个想到的可能是goto语句，但是goto不能跳转到另一个函数的某个abel，不过C提供了另外两个函数来完成这个任务：**setjmp** 和 **longjmp**。

函数原型

```
#include <setjmp.h>
int setjmp(jmp_buf env);
void longjmp(jmp_buf env, int val);
```

例子

第一次调用setjmp会返回0，并且将函数在此处的上下文保存在jmp_buf结构体里。调用longjmp函时，jmp_buf从setjmp函数保存的上下文恢复，然后程序跳转到setjmp处，setjmp再次返回，如果logjmp设置的val为0，则setjmp返回1，否则返回val。

```
#include <stdio.h>      /* printf, scanf */
#include <stdlib.h>     /* exit */
#include <setjmp.h>     /* jmp_buf, setjmp, longjmp */

main()
{
  jmp_buf env;
  int val;

  val = setjmp (env);
  if (val) {
    fprintf (stderr,"Error %d happened",val);
    exit (val);
  }

  /* code here */

  longjmp (env,101);   /* signaling an error */

  return 0;
}
```

C中的宏

宏（macro）就是用#define定义的一个符号，在**编译预处理**时，对程序中所有出现的“宏名”，都被宏定义时的字符串所替换。一般形式：

```
#define 宏名 字符串
```

宏可以分为两类：有参数和无参数。

```
// 无参数
#define PI 3.1415926
```

```
// 有参数
#define MULTIPLY(x, y) (x) * (y)
```

宏用的好，可以让C程序简洁合理很多。

实现Exception

```
#include <stdio.h>
#include <setjmp.h>

static jmp_buf ex_buf__;

#define TRY switch(setjmp(ex_buf__)) { case 0:
#define CATCH(x) break; case x:
#define ETRY break; }
#define THROW(x) longjmp(ex_buf__, x)

#define FOO_EXCEPTION (1)
#define BAR_EXCEPTION (2)
#define BAZ_EXCEPTION (3)

void dosomething()
{
    printf("I am doing something!\n");
    THROW( BAR_EXCEPTION );
    printf("I am not reachable\n");
}

int main(int argc, char** argv)
{
    TRY
    {
        dosomething();
    }
    CATCH( FOO_EXCEPTION )
    {
       printf("Got Foo!\n");
    }
    CATCH( BAR_EXCEPTION )
    {
       printf("Got Bar!\n");
    }
    CATCH( BAZ_EXCEPTION )
    {
       printf("Got Baz!\n");
    }
    ETRY;
    printf("everything ends\n");
    return 0;
}
```