

使用C 异常来取代exit()函数[2] PDF转换可能丢失图片或格式  
， 建议阅读原文

[https://www.100test.com/kao\\_ti2020/246/2021\\_2022\\_\\_E4\\_BD\\_BF\\_E7\\_94\\_A8C\\_\\_\\_E5\\_c97\\_246000.htm](https://www.100test.com/kao_ti2020/246/2021_2022__E4_BD_BF_E7_94_A8C___E5_c97_246000.htm) 在原始代码中缺少恢复或日志记录的功能，如果发生了一个错误，程序就会"消失"不见了，让用户手足无措。下面是重新组织后的代码，注意，没有修改函数修饰符：

```
void main() { //初始化 ... try {  
ProcessMail(...). } catch (int ret) { switch (ret) { case  
E_INITIALIZATION_FAILURE: ... case E_IRRECOVERABLE: ...  
... } }}void ProcessMail(...) { //初始化 ... if ( initializationError ) {  
throw(E_INITIALIZATION_FAILURE). } while ( !shutdown ) { try  
{ ReadMail(...) } catch (int ret) { switch (ret) { case  
E_READ_ERROR: //记录错误信息 ... //试图恢复 ... if ( recovered  
) { continue. } else { throw(E_IRRECOVERABLE). } break. case ... }  
} //继续处理 ... } //throw()可以用来取代缺少的返回码 //但也要注意  
由此带来的性能损失 throw(S_OK).} // ProcessMail()void  
ReadMail(...) { ... //在此无须捕捉异常 nBytesAvailable =  
ReadBytes(...) ...}int ReadBytes(...) { //读取数据 if ( error ) {  
throw(E_READ_ERROR). } return nBytesRead.}
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问

[www.100test.com](http://www.100test.com)