

第7章内容概要

- 程序设计语言
- 编码风格
- 软件测试基础
- 单元测试
- 集成测试
- 确认测试
- 白盒测试技术
- 黑盒测试技术
- ★ ■ 调试
- 软件可靠性

调试

- 软件调试是在进行了成功的测试之后才开始的工作。它与软件测试不同，调试的任务是进一步诊断和改正程序中潜在的错误。

调试

- 从技术角度来看，查找错误的难度在于：
 - 现象与原因所处的位置可能相距甚远。
 - 当其它错误得到纠正时，这一错误所表现出的现象可能会暂时消失，但并未实际排除。
 - 现象实际上是由一些非错误原因(例如，舍入不精确)引起的。
 - 现象可能是由于一些不容易发现的人为错误引起的。
 - 错误是由于时序问题引起的，与处理过程无关。
 - 现象是由于难于精确再现的输入状态（例如，实时应用中输入顺序不确定）引起。
 - 现象可能是周期出现的。在软、硬件结合的嵌入式系统中常常遇到。
 - 现象可能是由分布在许多任务中的原因引起的，这些任务运行在不同的处理机上。

几种主要的调试方法

■ 蛮干法

- 这种调试方法目前使用较多，效率较低。它不需要过多的思考，比较省脑筋。例如：
- 通过内存全部打印来调试
- 在程序特定部位设置打印语句

几种主要的调试方法

■ 回溯法调试

- 这是在小程序中常用的一种有效的调试方法。一旦发现了错误，人们先分析错误征兆，确定最先发现“症状”的位置。
- 然后，人工沿程序的控制流程，向回追踪源程序代码，直到找到错误根源或确定错误产生的范围。

几种主要的调试方法

■ 原因排除法

■ 对分查找法

- 如果已经知道每个变量在程序内若干个关键点的正确值，则可以用赋值语句或输入语句在程序中点附近“注入”这些变量的正确值，然后运行程序并检查所得到的输出。如果输出结果是正确的，则错误原因在程序的前半部分；反之，错误原因在程序的后半部分。对错误原因所在的那部分再重复使用这个方法，直到把出错范围缩小到容易诊断的程度为止。

几种主要的调试方法

■ 原因排除法

■ 归纳法

- 是从个别现象推断出一般性结论的思维方法。使用这种方法调试程序时，首先把和错误有关的数据组织起来进行分析，以便发现可能的错误原因。然后导出对错误原因的一个或多个假设，并利用已有的数据来证明或排除这些假设。当然，如果已有的数据尚不足以证明或排除这些假设，则需设计并执行一些新的测试用例，以获得更多的数据。

几种主要的调试方法

■ 原因排除法

■ 演绎法

- 是一种从一般原理或前提出发，经过排除和精化的过程来推导出结论的思考方法。演绎法排错是测试人员首先根据已有的测试用例，设想及枚举出所有可能出错的原因做为假设；然后再用原始测试数据或新的测试，从中逐个排除不可能正确的假设；最后，再用测试数据验证余下的假设确是出错的原因。

调试

- 在动手改正错误之前，软件工程师应该仔细考虑下述3个问题：
 - (1)是否同样的错误也在程序其他地方存在？
 - (2)将要进行的修改可能会引入的“下一个错误”是什么？
 - (3)为防止今后出现类似的错误，应该做什么？