相信大部分老鸟当年都看过这篇经典的文章。

在这里在转一次，以帮助大家能更好地问问题，以便获得更好的回答。

先贴结论吧

最后，不管是谁，来这里回答问题都是凭一腔热忱，凭兴趣和心情，如果版面充斥让人没有兴趣回答的问题，我想，对大家都不是好消息。自力更生真的很重要，不管你水平如何遇到什么样的困难，能自己解决多少就解决多少，然后再来求助，说需要什么什么帮助，多做一些努力只有好处没有坏处，问一个好问题并不比问一个坏问题困难多少。

**提问之前**

在通过电邮、新闻组或者聊天室提出技术问题前，检查你有没有做到：

1.通读手册，试着自己找答案。  
2.在FAQ里找答案（一份维护得好的FAQ可以包罗万象:）。  
3.在网上搜索（个人推荐google~~~）。  
4.向你身边精于此道的朋友打听。

当你提出问题的时候，首先要说明在此之前你干了些什么；这将有助于树立你的形象：你不是一个妄图不劳而获的乞讨者，不愿浪费别人的时间。如果提问者能从答案中学到东西，我们更乐于回答他的问题。

周全的思考，准备好你的问题，草率的发问只能得到草率的回答，或者根本得不到任何答案。越表现出在寻求帮助前为解决问题付出的努力，你越能得到实质性的帮助。

小心别问错了问题。如果你的问题基于错误的假设，普通黑客（J. Random Hacker）通常会用无意义的字面解释来答复你，心里想着“蠢问题…”，希望着你会从问题的回答（而非你想得到的答案）中汲取教训。

决不要自以为够资格得到答案，你没这种资格。毕竟你没有为这种服务支付任何报酬。你要自己去“挣”回一个答案，靠提出一个有内涵的，有趣的，有思维激励作用的问题–一个对社区的经验有潜在贡献的问题，而不仅仅是被动的从他人处索要知识–去挣到这个答案。

另一方面，表明你愿意在找答案的过程中做点什么，是一个非常好的开端。“谁能给点提示？”、“我这个例子里缺了什么？”以及“我应该检查什么地方？”比“请把确切的过程贴出来”更容易得到答复。因为你显得只要有人指点正确的方向，你就有完成它的能力和决心。

**怎样提问**  
-谨慎选择论坛

小心选择提问的场合。如果象下面描述的那样，你很可能被忽略掉或者被看作失败者：

1.在风马牛不相及的论坛贴出你的问题  
2.在探讨高级技巧的论坛张贴非常初级的问题；反之亦然  
3.在太多的不同新闻组交叉张贴

-用辞贴切，语法正确，拼写无误

我们从经验中发现，粗心的写作者通常也是马虎的思考者（我敢打包票）。 回答粗心大意者的问题很不值得，我们宁愿把时间耗在别处。

正确的拼写，标点符号和大小写很重要。

更一般的说，如果你的提问写得象个半文盲，你很有可能被忽视。

如果你在使用非母语的论坛提问，你可以犯点拼写和语法上的小错–但决不能在思考上马虎（没错，我们能弄清两者的分别）。

-使用含义丰富，描述准确的标题

在邮件列表或者新闻组中，大约50字以内的主题标题是抓住资深专家注意力的黄金时机。别用喋喋不休的“帮帮忙”（更别说“救命啊！！！！！”这样让人反感的话）来浪费这个机会。不要妄想用你的痛苦程度来打动我们， 别用空格代替问题的描述，哪怕是极其简短的描述。

蠢问题： 救命啊！我的膝上机不能正常显示了！

聪明问题：XFree86 4.1下鼠标光标变形，Fooware MV1005的显示芯片。

如果你在回复中提出问题，记得要修改内容标题，表明里面有一个问题。一个看起来象“Re：测试”或者“Re：新bug”的问题很难引起足够重视。另外，引用并删减前文的内容，给新来的读者留下线索。

-精确描述，信息量大

1.谨慎明确的描述症状。  
2.提供问题发生的环境（机器配置、操作系统、应用程序以及别的什么）。  
3.说明你在提问前是怎样去研究和理解这个问题的。  
4.说明你在提问前采取了什么步骤去解决它。  
5.罗列最近做过什么可能有影响的硬件、软件变更。

尽量想象一个黑客会怎样反问你，在提问的时候预先给他答案。

Simon Tatham写过一篇名为《如何有效的报告Bug》的出色短文。强力推荐你也读一读。

-话不在多

你需要提供精确有效的信息。这并不是要求你简单的把成吨的出错代码或者数据完全转储摘录到你的提问中。如果你有庞大而复杂的测试条件，尽量把它剪裁得越小越好。

这样做的用处至少有三点。第一，表现出你为简化问题付出了努力，这可以使你得到回答的机会增加；第二，简化问题使你得到有用答案的机会增加；第三，在提炼你的bug报告的过程中，也许你自己就能找出问题所在或作出更正。

-只说症状，不说猜想

告诉黑客们你认为问题是怎样引起的没什么帮助。（如果你的推断如此有效，还用向别人求助吗？），因此要确信你原原本本告诉了他们问题的症状，不要加进你自己的理解和推论。让黑客们来诊断吧。

蠢问题： 我在内核编译中一次又一次遇到SIG11错误，我怀疑某条飞线搭在主板的走线上了，这种情况应该怎样检查最好？

聪明问题： 我自制的一套K6/233系统，主板是FIC-PA2007（VIA Apollo VP2芯片组），256MB Corsair PC133 SDRAM，在内核编译中频频产生SIG11错误，从开机20分钟以后就有这种情况，开机前20分钟内从没发生过。重启也没有用，但是关机一晚上就又能工作20分钟。所有内存都换过了，没有效果。相关部分的典型编译记录如下…。

-按时间顺序列出症状

对找出问题最有帮助的线索，往往就是问题发生前的一系列操作，因此，你的说明应该包含操作步骤，以及电脑的反应，直到问题产生。

如果你的说明很长（超过四个段落），在开头简述问题会有所帮助，接下来按时间顺序详述。这样黑客们就知道该在你的说明中找什么。

-明白你想问什么

漫无边际的提问近乎无休无止的时间黑洞。最能给你有用答案的人也正是最忙的人（他们忙是因为要亲自完成大部分工作）。这样的人对无节制的时间黑洞不太感冒，因此也可以说他们对漫无边际的提问不大感冒。

如果你明确表述需要回答者做什么（提供建议，发送一段代码，检查你的补丁或是别的），就最有可能得到有用的答案。这会定出一个时间和精力的上限，便于回答者集中精力来帮你，这很奏效。要理解专家们生活的世界，要把专业技能想象为充裕的资源，而回复的时间则是贫乏的资源。解决你的问题需要的时间越少，越能从忙碌的专家口中掏出答案。

因此，优化问题的结构，尽量减少专家们解决它所需要的时间，会有很大的帮助–这通常和简化问题有所区别。因此，问“我想更好的理解X，能给点提示吗？”通常比问“你能解释一下X吗？”更好。如果你的代码不能工作，问问它有什么地方不对，比要求别人替你修改要明智得多。

-别问应该自己解决的问题

黑客们总是善于分辨哪些问题应该由你自己解决；因为我们中的大多数都曾自己解决这类问题。同样，这些问题得由你来搞定，你会从中学到东西。你可以要求给点提示，但别要求得到完整的解决方案。

-去除无意义的疑问

别用无意义的话结束提问，例如“有人能帮我吗？”或者“有答案吗？”。首先：如果你对问题的描述不很合适，这样问更是画蛇添足。其次：由于这样问是画蛇添足，黑客们会很厌烦你–而且通常会用逻辑上正确的回答来表示他们的蔑视，例如：“没错，有人能帮你”或者“不，没答案”。

-谦逊绝没有害处，而且常帮大忙

彬彬有礼，多用“请”和“先道个谢了”。让大家都知道你对他们花费时间义务提供帮助心存感激。然而，如果你有很多问题无法解决，礼貌将会增加你得到有用答案的机会。

（我们注意到，自从本指南发布后，从资深黑客处得到的唯一严重缺陷反馈，就是对预先道谢这一条。一些黑客觉得“先谢了”的言外之意是过后就不会再感谢任何人了。我们的建议是：都道谢。）

-问题解决后，加个简短说明

问题解决后，向所有帮助过你的人发个说明，让他们知道问题是怎样解决的，并再一次向他们表示感谢。如果问题在新闻组或者邮件列表中引起了广泛关注，应该在那里贴一个补充说明。补充说明不必很长或是很深入；简单的一句“你好，原来是网线出了问题！谢谢大家–Bill”比什么也不说要强。事实上，除非结论真的很有技术含量，否则简短可爱的小结比长篇学术论文更好。说明问题是怎样解决的，但大可不必将解决问题的过程复述一遍。除了表示礼貌和反馈信息以外，这种补充有助于他人在邮件列表/新闻组/论坛中搜索对你有过帮助的完整解决方案，这可能对他们也很有用。最后（至少？），这种补充有助于所有提供过帮助的人从中得到满足感。如果你自己不是老手或者黑客，那就相信我们，这种感觉对于那些你向他们求助的导师或者专家而言，是非常重要的。问题久拖未决会让人灰心；黑客们渴望看到问题被解决。好人有好报，满足他们的渴望，你会在下次贴出新问题时尝到甜头。

-还是不懂

如果你不是很理解答案，别立刻要求对方解释。象你以前试着自己解决问题时那样（利用手册，FAQ，网络，身边的高手），去理解它。如果你真的需要对方解释，记得表现出你已经学到了点什么。比方说，如果我回答你：“看来似乎是zEntry被阻塞了；你应该先清除它。”，然后：一个很糟的后续问题： “zEntry是什么？” 聪明的问法应该是这样：“哦~~~我看过帮助了但是只有-z和-p两个参数中提到了zEntry而且还都没有清楚的解释:<你是指这两个中的哪一个吗？还是我看漏了什么？”

**三思而后问**  
以下是几个经典蠢问题，以及黑客在拒绝回答时的心中所想：

问题：我能在哪找到X程序？  
问题：我的程序/配置/SQL申明没有用  
问题：我的Windows有问题，你能帮我吗？  
问题：我在安装Linux（或者X）时有问题，你能帮我吗？  
问题：我怎么才能破解root帐号/窃取OP特权/读别人的邮件呢？

提问：我能在哪找到X程序？  
回答：就在我找到它的地方啊蠢货–搜索引擎的那一头。天呐！还有人不会用Google吗？

提问：我的程序（配置、SQL申明）没有用  
回答：这不算是问题吧，我对找出你的真正问题没兴趣–如果要我问你二十个问题才找得出来的话–我有更有意思的事要做呢。

在看到这类问题的时候，我的反应通常不外如下三种：

1.你还有什么要补充的吗？  
2.真糟糕，希望你能搞定。  
3.这跟我有什么鸟相关？

提问：我的Windows有问题，你能帮我吗？  
回答：能啊，扔掉萎软的垃圾，换Linux吧。

提问：我在安装Linux（或者X）时有问题，你能帮我吗？  
回答：不能，我只有亲自在你的电脑上动手才能找到毛病。还是去找你当地的Linux用户组寻求手把手的指导吧（你能在这儿找到用户组的清单）。

提问：我怎么才能破解root帐号/窃取OP特权/读别人的邮件呢？  
回答：想要这样做，说明你是个卑鄙小人；想找个黑客帮你，说明你是个ΘΘΘΘ！

好问题，坏问题  
最后，我举一些例子来说明，怎样聪明的提问；同一个问题的两种问法被放在一起，一种是愚蠢的，另一种才是明智的。

蠢问题：我可以在哪儿找到关于Foonly Flurbamatic的资料？

//这种问法无非想得到“STFW”这样的回答。

聪明问题：我用Google搜索过“Foonly Flurbamatic 2600”，但是没找到有用的结果。谁知道上哪儿去找对这种设备编程的资料？

//这个问题已经STFW过了，看起来他真的遇到了麻烦。

蠢问题：我从FOO项目找来的源码没法编译。它怎么这么烂？

//他觉得都是别人的错，这个傲慢自大的家伙

聪明问题：FOO项目代码在Nulix 6.2版下无法编译通过。我读过了FAQ，但里面没有提到跟Nulix有关的问题。这是我编译过程的记录，我有什么做得不对的地方吗？

//他讲明了环境，也读过了FAQ，还指明了错误，并且他没有把问题的责任推到别人头上，这个家伙值得留意。

蠢问题：我的主板有问题了，谁来帮我？

//普通黑客对这类问题的回答通常是：“好的，还要帮你拍拍背和换尿布吗？” ，然后按下删除键。

聪明问题：我在S2464主板上试过了X、Y和Z，但没什么作用，我又试了A、B和C。请注意当我尝试C时的奇怪现象。显然边带传输中出现了收缩，但结果出人意料。在多处理器主板上引起边带泄漏的通常原因是什么？谁有好主意接下来我该做些什么测试才能找出问题？

//这个家伙，从另一个角度来看，值得去回答他。他表现出了解决问题的能力，而不是坐等天上掉答案。

在最后一个问题中，注意“告诉我答案”和“给我启示，指出我还应该做什么诊断工作”之间微妙而又重要的区别。事实上，后一个问题源自于2001年8月在Linux内核邮件列表上的一个真实的提问。我（Eric）就是那个提出问题的人。我在Tyan S2464主板上观察到了这种无法解释的锁定现象，列表成员们提供了解决那一问题的重要信息。

通过我的提问方法，我给了大家值得玩味的东西；我让人们很容易参与并且被吸引进来。我显示了自己具备和他们同等的能力，邀请他们与我共同探讨。我告诉他们我所走过的弯路，以避免他们再浪费时间，这是一种对他人时间价值的尊重。后来，当我向每个人表示感谢，并且赞赏这套程序（指邮件列表中的讨论 –译者注）运作得非常出色的时候，一个Linux内核邮件列（lkml）成员表示，问题得到解决并非由于我是这个列表中的“名人”，而是因为我用了正确的方式来提问。我们黑客从某种角度来说是拥有丰富知识但缺乏人情味的家伙；我相信他是对的，如果我象个乞讨者那样提问，不论我是谁，一定会惹恼某些人或者被他们忽视。他建议我记下这件事，给编写这个指南的人一些指导。

**找不到答案怎么办**  
如果仍得不到答案，请不要以为我们觉得无法帮助你。有时只是看到你问题的人不知道答案罢了。没有回应不代表你被忽视，虽然不可否认这种差别很难区分。

总的说来，简单的重复张贴问题是个很糟的想法。这将被视为无意义的喧闹。

你可以通过其它渠道获得帮助，这些渠道通常更适合初学者的需要。有许多网上的以及本地的用户组，由狂热的软件爱好者（即使他们可能从没亲自写过任何软件）组成。通常人们组建这样的团体来互相帮助并帮助新手。

另外，你可以向很多商业公司寻求帮助，不论公司大还是小（Red Hat和LinuxCare就是两个最常见的例子）。别为要付费才能获得帮助而感到沮丧！毕竟，假使你的汽车发动机汽缸密封圈爆掉了–完全可能如此– 你还得把它送到修车铺，并且为维修付费。就算软件没花费你一分钱，你也不能强求技术支持总是免费的。

对大众化的软件，就象Linux之类而言，每个开发者至少会有上万名用户。根本不可能由一个人来处理来自上万名用户的求助电话。要知道，即使你要为帮助付费，同你必须购买同类软件相比，你所付出的也是微不足道的（通常封闭源代码软件的技术支持费用比开放源代码软件要高得多，且内容也不那么丰富）。

如何有效地报告Bug  
引言

为公众写过软件的人，大概都收到过很拙劣的bug（计算机程序代码中的错误或程序运行时的瑕疵——译者注）报告，例如：

在报告中说“不好用”；  
所报告内容毫无意义；  
在报告中用户没有提供足够的信息；  
在报告中提供了虚假信息；  
所报告的问题是由于用户的过失而产生的；  
所报告的问题是由于其他程序的错误而产生的；  
所报告的问题是由于网络错误而产生的；

这便是为什么“技术支持”被认为是一件可怕的工作，因为有拙劣的bug报告需要处理。然而并不是所有的bug报告都令人生厌：我在业余时间维护自由软件，有时我会收到非常清晰、有帮助并且内容丰富的bug报告。

在这里我会尽力阐明如何写一个好的bug报告。我非常希望每一个人在报告bug之前都读一下这篇短文，当然我也希望用户在给我报告bug之前已经读过这篇文章。

简单地说，报告bug的目的是为了让程序员看到程序的错误。您可以亲自示范，也可以给出能导致程序出错的、详尽的操作步骤。如果程序出错了，程序员会收集额外的信息直到找到错误的原因；如果程序没有出错，那么他们会请您继续关注这个问题，收集相关的信息。

在bug报告里，要设法搞清什么是事实（例如：“我在电脑旁”和“XX出现了”）什么是推测（例如：“我想问题可能是出在……”）。如果愿意的话，您可以省去推测，但是千万别省略事实。

当您报告bug的时候（既然您已经这么做了），一定是希望bug得到及时修正。所以此时针对程序员的任何过激或亵渎的言语（甚至谩骂）都是与事无补的 ——因为这可能是程序员的错误，也有可能是您的错误，也许您有权对他们发火，但是如果您能多提供一些有用的信息（而不是激愤之词）或许bug会被更快的修正。除此以外，请记住：如果是免费软件，作者提供给我们已经是出于好心，所以要是太多的人对他们无礼，他们可能就要“收起”这份好心了。

“程序不好用”

程序员不是弱智：如果程序一点都不好用，他们不可能不知道。他们不知道一定是因为程序在他们看来工作得很正常。所以，或者是您作过一些与他们不同的操作，或者是您的环境与他们不同。他们需要信息，报告bug也是为了提供信息。信息总是越多越好。

许多程序，特别是自由软件，会公布一个“已知bug列表”。如果您找到的bug在列表里已经有了，那就不必再报告了，但是如果您认为自己掌握的信息比列表中的丰富，那无论如何也要与程序员联系。您提供的信息可能会使他们更简单地修复bug。

本文中提到的都是一些指导方针，没有哪一条是必须恪守的准则。不同的程序员会喜欢不同形式的bug报告。如果程序附带了一套报告bug的准则，一定要读。如果它与本文中提到的规则相抵触，那么请以它为准。

如果您不是报告bug，而是寻求帮助，您应该说明您曾经到哪里找过答案，（例如：我看了第四章和第五章的第二节，但我找不到解决的办法。）这会使程序员了解用户喜欢到哪里去找答案，从而使程序员把帮助文档做得更容易使用。

“演示给我看”

报告bug的最好的方法之一是“演示”给程序员看。让程序员站在电脑前，运行他们的程序，指出程序的错误。让他们看着您启动电脑、运行程序、如何进行操作以及程序对您的输入有何反应。

他们对自己写的软件了如指掌，他们知道哪些地方不会出问题，而哪些地方最可能出问题。他们本能地知道应该注意什么。在程序真的出错之前，他们可能已经注意到某些地方不对劲，这些都会给他们一些线索。他们会观察程序测试中的每一个细节，并且选出他们认为有用的信息。

这些可能还不够。也许他们觉得还需要更多的信息，会请您重复刚才的操作。他们可能在这期间需要与您交流一下，以便在他们需要的时候让bug重新出现。他们可能会改变一些操作，看看这个错误的产生是个别问题还是相关的一类问题。如果您不走运，他们可能需要坐下来，拿出一堆开发工具，花上几个小时研究。但是最重要的是在程序出错的时候让程序员在电脑旁。一旦他们看到了问题，他们通常会找到原因并开始试着修改。

“告诉我该怎么做”

如今是网络时代，是信息交流的时代。我可以点一下鼠标把自己的程序送到俄罗斯的某个朋友那里，当然他也可以用同样简单的方法给我一些建议。但是如果我的程序出了什么问题，我不可能在他旁边。“演示”是很好的办法，但是常常做不到。

如果您必须报告bug，而此时程序员又不在您身边，那么您就要想办法让bug重现在他们面前。当他们亲眼看到错误时，就能够进行处理了。

确切地告诉程序员您做了些什么。如果是一个图形界面程序，告诉他们您按了哪个按钮，依照什么顺序按的。如果是一个命令行程序，精确的告诉他们您键入了什么命令。您应该尽可能详细地提供您所键入的命令和程序的反应。

把您能想到的所有的输入方式都告诉程序员，如果程序要读取一个文件，您可能需要发一个文件的拷贝给他们。如果程序需要通过网络与另一台电脑通讯，您或许不能把那台电脑复制过去，但至少可以说一下电脑的类型和安装了哪些软件（如果可以的话）。

“哪儿出错了？在我看来一切正常哦！”

如果您给了程序员一长串输入和指令，他们执行以后没有出现错误，那是因为您没有给他们足够的信息，可能错误不是在每台计算机上都出现，您的系统可能和他们的在某些地方不一样。有时候程序的行为可能和您预想的不一样，这也许是误会，但是您会认为程序出错了，程序员却认为这是对的。

同样也要描述发生了什么。精确的描述您看到了什么。告诉他们为什么您觉得自己所看到的是错误的，最好再告诉他们，您认为自己应该看到什么。如果您只是说：“程序出错了”，那您很可能漏掉了非常重要的信息。

如果您看到了错误消息，一定要仔细、准确的告诉程序员，它们很重要。在这种情况下，程序员只要修正错误，而不用去找错误。他们需要知道是什么出问题了，系统所报的错误消息正好帮助了他们。如果您没有更好的方法记住这些消息，就把它们写下来。只报告“程序出了一个错”是毫无意义的，除非您把错误消息一块报上来。

特殊情况下，如果有错误消息号，一定要把这些号码告诉程序员。不要以为您看不出任何意义，它就没有意义。错误消息号包含了能被程序员读懂的各种信息，并且很有可能包含重要的线索。给错误消息编号是因为用语言描述计算机错误常常令人费解。用这种方式告诉您错误的所在是一个最好的办法。

在这种情形下，程序员的排错工作会十分高效。他们不知道发生了什么，也不可能到现场去观察，所以他们一直在搜寻有价值的线索。错误消息、错误消息号以及一些莫名其妙的延迟，都是很重要的线索，就像办案时的指纹一样重要，保存好。

如果您使用UNIX系统，程序可能会产生一个内核输出（core dump）。内核输出是特别有用的线索来源，别扔了它们。另一方面，大多数程序员不喜欢收到含有大量内核输出文件的EMAIL，所以在发邮件之前最好先问一下。还有一点要注意：内核输出文件记录了完整的程序状态，也就是说任何秘密（可能当时程序正在处理一些私人信息或秘密数据）都可  
能包含在内核输出文件里。

出了问题之后，我做了……”

当一个错误或bug发生的时候，您可能会做许多事情。但是大多数人会使事情变的更糟。我的一个朋友在学校里误删了她所有的Word文件，在找人帮忙之前她重装了Word，又运行了一遍碎片整理程序，这些操作对于恢复文件是毫无益处的，因为这些操作搞乱了磁盘的文件区块。恐怕在这个世界上没有一种反删除软件能恢复她的文件了。如果她不做任何操作，或许还有一线希望。

这种人仿佛一只被逼到墙角的鼬（黄鼠狼、紫貂一类的动物——译者注）：背靠墙壁，面对死亡的降临奋起反扑，疯狂攻击。他们认为做点什么总比什么都不做强。然而这些在处理计算机软件问题时并不适用。不要做鼬，做一只羚羊。当一只羚羊面对料想不到的情况或受到惊吓时，它会一动不动，是为了不吸引任何注意，与此同时也在思考解决问题的最好办法（如果羚羊有一条技术支持热线，此时占线。）。然后，一旦它找到了最安全的行动方案，它便去做。

当程序出毛病的时候，立刻停止正在做的任何操作。不要按任何按钮。仔细地看一下屏幕，注意那些不正常的地方，记住它或者写下来。然后慎重地点击 “确定” 或“取消”，选择一个最安全的。学着养成一种条件反射——一旦电脑出了问题，先不要动。要想摆脱这个问题，关掉受影响的程序或者重新启动计算机都不好，一个解决问题的好办法是让问题再次产生。程序员们喜欢可以被重现的问题，快乐的程序员可以更快而且更有效率的修复bug。

“我想粒子的跃迁与错误的极化有关”

并不只是非专业的用户才会写出拙劣的bug报告，我见过一些非常差的bug报告出自程序员之手，有些还是非常优秀的程序员。

有一次我与另一个程序员一起工作，他一直在找代码中的bug，他常常遇到一个bug，但是不会解决，于是就叫我帮忙。“出什么毛病了？”我问。而他的回答却总是一些关于bug的意见。如果他的观点正确，那的确是一件好事。这意味着他已经完成了工作的一半，并且我们可以一起完成另一半工作。这是有效率并有用的。

但事实上他常常是错的。这就会使我们花上半个小时在原本正确的代码里来回寻找错误，而实际上问题出在别的地方。我敢肯定他不会对医生这么做。“大夫，我得了Hydroyoyodyne（真是怪病——译者），给我开个方子”，人们知道不该对一位医生说这些。您描述一下症状，哪个地方不舒服，哪里疼、起皮疹、发烧……让医生诊断您得了什么病，应该怎样治疗。否则医生会把您当做疑心病或精神病患者打发了，这似乎没什么不对。

做程序员也是一样。即便您自己的“诊断”有时真的有帮助，也要只说“症状”。“诊断”是可说可不说的，但是“症状”一定要说。同样，在bug报告里面附上一份针对bug而做出修改的源代码是有用处的，但它并不能替代bug报告本身。

如果程序员向您询问额外的信息，千万别应付。曾经有一个人向我报告bug，我让他试一个命令，我知道这个命令不好用，但我是要看看程序会返回一个什么错误（这是很重要的线索）。但是这位老兄根本就没试，他在回复中说“那肯定不好用”，于是我又花了好些时间才说服他试了一下那个命令。

多动动脑筋对程序员是有帮助的。即使您的推断是错误的，程序员也应该感谢您，您的尝试使他们的工作变的更简单。不过千万别忘了报告“症状”，否则只会使事情变得更糟。

“真是奇怪，刚才还不好用，怎么现在又好了？”

“间歇性错误”着实让程序员发愁。相比之下，进行一系列简单的操作便能导致错误发生的问题是简单的。程序员可以在一个便于观察的条件下重复那些操作，观察每一个细节。太多的问题在这种情况下不能解决，例如：程序每星期出一次错，或者偶然出一次错，或者在程序员面前从不出错（程序员一离开就出错。——译者）。当然还有就是程序的截止日期到了，那肯定要出错。

大多数“间歇性错误”并不是真正的“间歇”。其中的大多数错误与某些地方是有联系的。有一些错误可能是内存泄漏产生的，有一些可能是别的程序在不恰当的时候修改某个重要文件造成的，还有一些可能发生在每一个小时的前半个小时中（我确实遇到过这种事情）。

同样，如果您能使bug重现，而程序员不能，那很有可能是他们的计算机和您的计算机在某些地方是不同的，这种不同引起了问题。我曾写过一个程序，它的窗口可以蜷缩成一个小球停在屏幕的左上角，它在别的计算机上只能在800×600解析度工作，但是在我的机器上却可以在1024×768工作。

程序员想要了解任何与您发现的问题相关的事情。有可能的话您到另一台机器上试试，多试几次，两次，三次，看看问题是不是经常发生。如果问题出现在您进行了一系列操作之后，不是您想让它出现它就会出现，这就有可能是长时间的运行或处理大文件所导致的错误。程序崩溃的时候，您要尽可能的记住您都做了些什么，并且如果您看到任何图形,也别忘了提一下。您提供的任何事情都是有帮助的。即使只是概括性的描述（例如：当后台有EMACS运行时，程序常常出错），这虽然不能提供导致问题的直接线索，但是可能帮助程序员重现问题。

最重要的是：程序员想要确定他们正在处理的是一个真正的“间歇性错误”呢，还是一个在另一类特定的计算机上才出现的错误。他们想知道有关您计算机的许多细节，以便了解您的机器与他们的有什么不同。有许多细节都依仗特定的程序，但是有一件东西您一定要提供——版本号。程序的版本、操作系统的版本以及与问题有关的程序的版本。

“我把磁盘装进了我的Windows……”

表意清楚在一份bug报告里是最基本的要求。如果程序员不知道您说的是什么意思，那您就跟没说一样。我收到的bug报告来自世界各地，有许多是来自非英语国家，他们通常为自己的英文不好而表示歉意。总的来说，这些用户发来的bug报告通常是清晰而且有用的。几乎所有不清晰的bug报告都是来自母语是英语的人，他们总是以为只要自己随便说说，程序员就能明白。

精确：

如果做相同的事情有两种方法，请说明您用的是哪一种。例如：“我选择了‘载入’”，可能意味着“我用鼠标点击‘载入’”或“我按下了‘ALT+L’”，说清楚您用了哪种方法，有时候这也有关系。

详细：

信息宁多毋少！如果您说了很多，程序员可以略去一部分，可是如果您说的太少，他们就不得不回过头再去问您一些问题。有一次我收到了一份bug报告只有一句话，每一次我问他更多事情时，他每次的回复都是一句话，于是我花了几个星期的时间才得到了有用的信息。

谨慎使用代词：

诸如“它”，“窗体”这些词，当它们指代不清晰的时候不要用。来看看这句话：“我运行了FooApp，它弹出一个警告窗口，我试着关掉它，它就崩溃了。”这种表述并不清晰，用户究竟关掉了哪个窗口？是警告窗口还是整个FooApp程序？您可以这样说，“我运行FooApp程序时弹出一个警告窗口，我试着关闭警告窗口，FooApp崩溃了。”这样虽然罗嗦点，但是很清晰不容易产生误解。

检查：

重新读一遍您写的bug报告，您觉得它是否清晰？如果您列出了一系列能导致程序出错的操作，那么照着做一遍，看看您是不是漏写了一步。

小结：

bug报告的首要目的是让程序员亲眼看到错误。如果您不能亲自做给他们看，给他们能使程序出错的详细的操作步骤。

如果首要目的不能达成，程序员不能看到程序出错。这就需要bug报告的第二个目的来描述程序的什么地方出毛病了。详细的描述每一件事情：您看到了什么，您想看到什么，把错误消息记下来，尤其是“错误消息号”。

当您的计算机做了什么您料想不到的事，不要动！在您平静下来之前什么都别做。不要做您认为不安全的事。

尽量试着自己“诊断”程序出错的原因（如果您认为自己可以的话）。即使做出了“诊断”，您仍然应该报告“症状”。

如果程序员需要，请准备好额外的信息。如果他们不需要，就不会问您要。他们不会故意为难自己。您手头上一定要有程序的版本号，它很可能是必需品。

表述清楚，确保您的意思不能被曲解。

总的来说，最重要的是要做到精确。程序员喜欢精确。

本文来源  
Copyright(C) 2001 by Eric S. Raymond  
中文版Copyleft 2001 by D.H.Grand(nOBODY/Ginux)  
英文版：[http://www.tuxedo.org/~esr/faqs/smart-questions.html](http://www.tuxedo.org/%7Eesr/faqs/smart-questions.html)  
感谢Eric的耐心指点和同意，本文才得以完成并发布  
本指南英文版版权为Eric Steven Raymond 所有

中文版版权由D.H.Grand[nOBODY/Ginux]所有

最后，不管是谁，来这里回答问题都是凭一腔热忱，凭兴趣和心情，如果版面充斥让人没有兴趣回答的问题，我想，对大家都不是好消息。自力更生真的很重要，不管你水平如何遇到什么样的困难，能自己解决多少就解决多少，然后再来求助，说需要什么什么帮助，多做一些努力只有好处没有坏处，问一个好问题并不比问一个坏问题困难多少。