



提問的智慧

Eric S. Raymond 著

how to ask questions the smart way

提問的智慧

How To Ask Questions The Smart Way

Eric S. Raymond 著

D. H. Grand 译

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>

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。

前言

这是一本好书——如果认为它是一本书的话。

提问很简单吗？不简单吗？很简单吗？不简单吗？很简单吗？……这个不能不认真考虑。

在群里¹混了好久，总能见到好的问题——当然也经常见到很糟糕的问题，糟糕得了让人觉得人生前途暗淡。想起高中时代的同桌，那老兄英语不好，总是求我给讲讲，每次我问，讲什么？考卷还是语法？那仁兄总是来一句，你就讲吧。也许我不该问，拿起什么来讲就是了。群里的的问题呢，也许我也不该多事，自会有人回答的，由他去问罢了。不过，每次看到有人没头没尾地抛出一句话求翻译或者没有任何语境地问一个自认为有水准的问题，都会觉得难以忍受，久而久之，那些问问题的人成了 S，我却仿佛真成了 M，虽然我在群里的名字是老 M。后来花花的一番话警醒了我，真正的 S 和 M 是同时获得愉悦的感的，而我在看或回答那些糟糕问题的时候没有任何愉悦感。

于是释然。恰好花花今天在老肖的论坛上找到这部小册子，我便自告奋勇将其拷贝下来，制作成一本像书一样的指南，就按花花说的，叫《群新人指南》得了。希望大家，也包括我自己，认真研读，一边读一边想，以后再也不要出现糟糕的问题了，或者以后群里的糟糕问题的糟糕程度能较此前低一些。别人管不了，至少自己得做到。是为前言，与群里诸位（即群众）众共勉。

老 M

2011年秋（恰逢双十日）

¹ 语言学研究群：108438890；语言学硕博士群：4097145。本书为入群者必读之书。

目录

简介

提问之前

怎样提问

谨慎选择论坛

尽量使用邮件列表

用辞贴切，语法正确，拼写无误

用易读格式发送问题

使用含义丰富，描述准确的标题

精确描述，信息量大

话不在多

只说症状，不说猜想

按时间顺序列出症状

别要求私下答复

明白你想问什么

别问应该自己解决的问题

去除无意义的疑问

谦逊绝没有害处，而且常帮大忙

问题解决后，加个简短说明

如何理解答案

RTFM 和 STFW: 别烦我啦

还是不懂:(

面对无礼

决不要象个失败者

三思而后问

好问题，坏问题

找不到答案怎么办

=====

简介

=====

在黑客世界里，当提出一个技术问题时，你能得到怎样的回答？这取决于挖出答案的难度，同样取决于你提问的方法。本指南旨在帮助你提高发问技巧，以获取你最想要的答案。

首先你必须明白，黑客们只偏爱艰巨的任务，或者能激发他们思维的好问题。如若不然，我们还来干吗？如果你有值得我们反复咀嚼玩味的好问题，我们自会对你感激不尽。好问题是激励，是厚礼，可以提高我们的理解力，而且通常会暴露我们以前从没意识到或者思考过的问题。对黑客而言，“问得好！”是发自内心的大力称赞。

尽管黑客们有蔑视简单问题和不友善的坏名声，有时看起来似乎我们对新手，对知识贫乏者怀有敌意，但其实不是那样的。

我们不想掩饰对这样一些人的蔑视--他们不愿思考，或者在发问前不去完成他们应该做的事。这种人只会谋杀时间--他们只愿索取，从不付出，无端消耗我们的时间，而我们本可以把时间用在更有趣的问题或者更值得回答的人身上。我们称这样的人为“失败者”（由于历史原因，我们有时把它拼作“lusers”）。

我们也知道，很多人只想使用我们编写的软件，对技术细节没什么兴趣。对多数人们而言，计算机不过是一个工具，一种达到目的的手段；他们有更重要的事情要做，有更重要的生活要过。我们明白这点，也并不奢望每个人都对另我们痴狂的技术问题有兴致。然而，我们回答问题的风格是针对这样一群人--他们有兴趣，并且愿意积极参与问题的解决。这点不会改变，也不应该改变；如果变了，我们将失去我们引以为傲的效率。

我们在很大程度上属于志愿者，从繁忙的生活中抽出时间来解惑答疑，而且时常被提问淹没。所以我们无情的滤掉一些话题，特别是抛弃那些看起来象失败者的家伙，以便更高效的利用时间来回答胜利者的问题。

如果你觉得我们过于傲慢的态度让你不爽，让你委屈，不妨设身处地想想。我们并没有要求你向我们屈服--事实上，我们中的大多数人最喜欢公平交易不过了，只要你付出小小努力来满足最起码的要求，我们就会欢迎你加入到我们的文化中来。但让我们帮助那些不愿意帮助自己的人是没有意义的。如果你不能接受这种“歧视”，我们建议你花点钱找家商业公司签个技术支持协议得了，别向黑客乞求帮助。

如果你决定向我们求助，当然不希望被视为失败者，更不愿成为失败者中的一员。立刻得到有效答案的最好方法，就是象胜利者那样提问--聪明、自信、有解决问题的思路，只是偶尔在特定的问题上需要获得一点帮助。

(欢迎对本指南提出改进意见。任何建议请 E-mail 至 esr@thyrsus.com, 然而请注意, 本文并非网络礼节的通用指南, 我通常会拒绝无助于在技术论坛得到有用答案的建议。)

(当然, 如果你写中文, 最好还是寄到 DHGrand@hotmail.com;-)

=====

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。能说明你从这些操作中学到了什么就更好了。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

黑客们通常砍掉问错地方的问题，以保护自己的社区不被大量无关帖子淹没。你不会希望自己的帖子被这样砍掉吧。

总的说来，问题发到精心挑选的公众论坛，比发到封闭的小圈子更容易得到有用的答案。

这一现象有多种原因，其中之一是公众论坛有更多潜在的问题回答者；另一个原因是公众

论坛有更多的听众。黑客们更愿意让尽量多的人--而非有限的一两个--从回答中受益。

尽量使用邮件列表

如果某项目有自己的开发邮件列表，要把问题发到这个邮件列表而不是某个开发者，即使你很清楚谁最能回答你的问题。仔细查看项目文档和项目主页，找到这个项目的邮件列表地址，这样做的理由有四：

1. 任何值得问某位开发者的好问题，都值得向整个开发团体提出。反之，若你认为这个

问题不值得在邮件列表中提起，就没有理由用它来骚扰任何一位开发者。

2. 在邮件列表提问可以分担开发者的工作量。某位开发者（尤其当他是项目负责人的情况下），可能忙得没时间回答你的问题。

3. 大多数邮件列表都有历史存档，而且都能在搜索引擎中检索到。人们可以从中找到你的问题和答案，不用一遍又一遍在列表中发问。

4. 如果某个问题经常被提出，开发者可以据此改进文档或改进软件，以减少用户的困惑。而如果问题总在私下提出，就不会有人对此有整体上的把握了。

如果你找不到项目的邮件列表地址，只能看到项目维护者的，那就写给维护者吧。在这种情况下，也别以为邮件列表并不存在。在你的信中写明你已尽力寻找，仍无法找到邮件列表。另外表明你不介意将此消息转给他人。（大多数人认为私信就应该是私下的，即使并没有什么可保密的内容。允许你的消息被转寄给他人，给了收信者一种处理你邮件的选择。）

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。

因此，明确充分表述你的问题非常重要。如果你嫌这样做麻烦，我们也会懒得搭理你。注意推敲你的用辞，不一定要用呆板正式的语言--事实上，黑客文化的价值观是不拘小节。

准确的运用俚语和富有幽默感的语言，但别乱用；一定要能表明你在思考，在关注。

正确的拼写，标点符号和大小写很重要。别把“its”和“it's”或者“loose”和“lose”搞混淆了。别用全部大写的形式，这被视为粗鲁的大声叫嚷

（全都用小写也好不到哪儿去，因为这会给阅读带来困难。Alan Cox 可以用全部小写，但你不行）。

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。如果写得象一个窥客（破解爱好者）或者灰客（只会用现成工具的捣乱者）绝对是自己找死，保证你除了无情的抵制什么也得不到（或者，最好的结局是得到一大堆挖苦嘲笑的“帮助”）。

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）。另外，除非你确切知道你的回答者会使用什么语言，否则请用英文。匆匆忙忙的黑客往往简单的跳过他们看不懂的问题，而英文是网络上的工作语言。用英文可以降低你的问题未被阅读即遭抛弃的风险。

用易读格式发送问题

如果人为造成你的提问难以阅读和理解，将会更容易被人忽略。因此你要：

1. 使用纯文本邮件，不要使用 HTML（关掉 HTML 并不难）。
2. 通常可以附加 MIME 附件，但一定要有真正的内容（例如附加的源文件或者补丁），而

不仅仅是你的邮件客户端产生的文件模板（例如你邮件的一份拷贝）。

3. 不要把所有问题放在不停换行的一整段中。（这将让答复的人难于回答其中一部分问

题，即使能回答所有问题，我也更希望条理清楚的一个一个来:）。很可能收件人只能在

80 个字符宽度的文本显示器上读信，因此要相应的把行环绕模式设在 80 字符以内。

4. 不要在英文论坛使用 MIME Quoted-Printable 编码发送；这种编码格式对 ASCII 码不能表达的语言来说是非常必要的，但很多邮件代理不支持它，这时，满篇的“=20”符号把文字分割开，既难看，又分散注意力。

5. 永远不要指望黑客会乐于阅读封闭所有权的文件格式，例如微软的 Word 格式。多数黑客对此的反应就象你在门口的阶梯上堆满热烘烘的猪粪（意即谁也不会踏进你的门--译者注）。

6. 如果你通过一台安装 Windows 的电脑发送邮件，关闭微软愚蠢的“智能引用”功能。这能使你免于在邮件中夹带垃圾字符。

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。强烈推荐你也读一读。

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。在命令行操作的情况下，保存一个操作记录（例如使用脚本工具），并且引用相关的大约 20 条命令会大有帮助。

如果崩溃的程序有诊断选项（例如用 -v 转到详尽模式），试着仔细考虑选择选项以在操作记录中增加有用的调试信息。

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。

别要求私下答复

黑客们认为解决问题应该有公开、透明的流程。只要任何更有见地的人注意到答案的不完善或者不正确，这个最初的答案就可以和应该得到纠正。同时，通过能力和知识被大家注意，被大家接受，回答问题者得到了应有的奖励。

如果你要求对方私下回答你，这既破坏了整个流程，也破坏了奖励制度。别提这要求，这是回答者的权利，由他来选择是否私下答复--如果他选择这样做，通常是因为他认为这个答案过于显而易见或者有不良的公开影响，别人不会感兴趣。

只有一种有限的例外：如果你预计将收到大量雷同的答复，你可以说：“把答案寄给我，由我来汇总吧。”将邮件列表或者新闻组从大量重复的帖子中打救出来是很有君子之风的--但请记住，履行自己关于汇总的承诺。

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上限，便于回答者集中精力来帮你，这很凑效。

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。

实话实说，虽然这不象合乎语法、清楚准确的描述，避免私有格式等等那么重要（也不能用来替代它们）；黑客一般更喜欢直接了当然而技术上敏锐的 bug 报告，而不是彬彬有礼的废话（如果这让你迷惑不解，请记住，我们衡量一个问题价值的标准是：它能让我们学会多少）。

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。

=====

如何理解答案

=====

RTFM 和 STFW: 别烦我啦

有一个古老而神圣的传统：如果你收到“RTFM（Read The Fucking Manual）”的回复，回答者认为你应该去读 TMD 手册。当然，基本上他是对的，你应该读一读。

RTFM 有一个年轻的亲戚。如果答案是“STFW（Search The Fucking Web）”，回答者认为你应该到 TMD 的网上去搜索。基本上，他也是对的，你就去找吧。

通常，用这两句之一回答你的人会给你一份包含你需要内容的手册或者一个网址，而且他们打这些字的时候正在阅读着。这些答复意味着回答者认为(1). 你需要的信息非常容易获得；(2). 你自己去搜索这些信息比灌给你能让你学到更多。

别为这个而不爽；依照黑客的标准，他没有对你的要求视而不见，已经能大致能表示对你的关注。你应该对他祖母般的慈祥表示感谢。

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？”

面对无礼

黑客圈子里很多貌似粗鲁的言行并非有意冒犯。更恰当的说，这是直率、不说废话的沟通方式的产物，这种沟通方式源于人们关注问题的解决--多过让人感受温暖亲情然而却依旧糊里糊涂--的天性。

如果你觉得受到粗鲁的对待，请保持冷静。如果真有人表现粗野，通常会有列表/新闻组/论坛的长辈找他谈心，如果没有这样，而你又大发脾气，则很可能对方的言行是黑客社区行为规范许可内，而你被认为是有过错的。这会不利于你得到信息或者帮助。

另一方面，你偶尔也会无缘无故有粗野的言行和心态。上述现象的另一面是，人们允许狠狠打击真正的冒犯者，用尖刻的言语剖析他们的不当言行。如果你真决定这样做，先仔细又仔细的掂量一下你自己的分量。合理的粗鲁与发动一场无意义的论战之间只隔了一条细细的线，冒冒失失撞上去的黑客不在少数；如果你是新手或者门外汉，不犯这种错的机会是很渺茫的。如果你想得到信息而不是来胡闹，别冒险回复，最好把手从键盘上拿开。

（有些人声称多数黑客有孤僻症或者社交障碍综合征的轻度症状，而且确实缺少部分有助“常人”进行社交行为的脑组织结构。这也许是真的，也许不是。如果你自己不是黑客，那么，把我们想象成脑部有缺陷的人有助你面对我们的古怪。有话直说，我们无所谓；我们乐于按自己的想法生活，而且总是对医学概念持相当怀疑的态度。）

在下一节里，我们将谈论另一个话题；当你行差踏错时可能遇到的“无礼”。

=====

决不要象个失败者

=====

很有可能，你在黑客社区的论坛会受到很多公开的攻击--用本文提到的各种方式或类似的方法，而且很可能会有各式各样的旁敲侧击来告诉你你有多讨厌。

如果噩梦成真，你能做的最糟的事就是为此发牢骚，抱怨受到人身攻击，要求对方道歉，尖叫，屏住呼吸，威胁要控诉对方，向他老板告状，不掀起马桶座圈，等等等等。然而，你应该这样：

由它去吧，这没什么大不了的。实际上这么做是恰当的和有益的（主要是有利身心健康:）。

社区的规范不靠社区，而是靠积极推行它们的人们来维护，这种维护是公开的，显而易见的。别抱怨说一切批评都应该通过私信传送，它本来就不该那样。当别人指出你的话有错误，或者他有不同观点的时候，坚持认为他在羞辱你是没有用的。这些都是失败者的态度。

有那么一些黑客论坛，出于对高度自谦的误解，禁止参与者张贴专给人找茬的帖子，而且被告知“如果不愿帮助用户，那就闭嘴。”，他们认为，引开参与者的话题，只会使得他们陶醉在毫无意义的喋喋不休中，从而失去了技术论坛的意义。

夸张的“友善”（以那种方式）还是有用的帮助：你自己选择吧。

记住：当黑客说你很烦人，（无论用多么粗暴的语言）警告你别再那样做了，他的本意并非是针对(1)你，以及(2)他的社区。他本来可以轻易的忽略你，把你从他的视线中抹去。

如果你无法接受要向他表示感谢，至少应该表现出你的气度，别抱怨，别期望只因为你是新人，你有戏剧般的敏感脆弱的神经和自封的权利，而受到易碎玩偶般的特别对待。

=====

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：我怎样才能破解 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，说明你是个白痴！

=====

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 内核邮件列表（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，而且内容也不那么丰富）。