提问的智慧

How To Ask Questions The Smart Way

原文网址:http://www.catb.org/~esr/faqs/smart-questions.html

本中文指南是基于原文 3.10 版以及 2010 年由 Gasolin 所翻译版本的最新翻译；

协助指出翻译问题，请发**Issue**，或直接发**Pull Request**给我。

本文另有繁体中文版: https://github.com/ryanhanwu/How-To-Ask-Questions-The-Smart-Way

原文版本历史

目录

声明

许多项目在他们的使用协助/说明网页中链接了本指南，这么做很好，我们也鼓励大家都这么做。但如果你是负责管理这个项目网页的人，请在超链接附近的显着位置上注明：

本指南不提供此项目的实际支持服务！

我们已经深刻领教到少了上述声明所带来的痛苦。因为少了这点声明，我们不停地被一些白痴纠缠。这些白痴认为既然我们发布了这本指南，那么我们就有责任解决世上所有的技术问题。

如果你是因为需要某些协助而正在阅读这本指南，并且最后离开是因为发现从本指南作者们身上得不到直接的协助，那么你就是我们所说的那些白痴之一。别问我们问题，我们只会忽略你。我们在这本指南中是教你如何从那些真正懂得你所遇到软件或硬件问题的人取得协助，而99%的情况下那不会是我们。除非你确定本指南的作者之一刚好是你所遇到的问题领域的专家，否则请不要打扰我们，这样大家都会开心一点。

简介

在黑客的世界里，当你抛出一个技术问题时，最终是否能得到有用的回答，往往取决于你所提问和追问的方式。本指南将教你如何正确的提问以获得你满意的答案。

不只是黑客，现在开放源代码（Open Source）软件已经相当盛行，你常常也可以由其他有经验的使用者身上得到好答案，这是件*好事*；使用者比起黑客来，往往对那些新手常遇到的问题更宽容一些。然而，将有经验的使用者视为黑客，并采用本指南所提的方法与他们沟通，同样也是能从他们身上得到满意回答的最有效方式。

首先你应该明白，黑客们喜爱有挑战性的问题，或者能激发我们思维的好问题。如果我们并非如此，那我们也不会成为你想询问的对象。如果你给了我们一个值得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。好问题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，"好问题！"是诚挚的大力称赞。

尽管如此，黑客们有着蔑视或傲慢面对简单问题的坏名声，这有时让我们看起来对新手、无知者似乎较有敌意，但其实不是那样的。

我们不讳言我们对那些不愿思考、或者在发问前不做他们该做的事的人的蔑视。那些人是时间杀手 -- 他们只想索取，从不付出，消耗我们可用在更有趣的问题或更值得回答的人身上的时间。我们称这样的人为 `失败者（撸瑟）`（由于历史原因，我们有时把它拼作 `lusers`）。

我们意识到许多人只是想使用我们写的软件，他们对学习技术细节没有兴趣。对大多数人而言，电脑只是种工具，是种达到目的的手段而已。他们有自己的生活并且有更要紧的事要做。我们了解这点，也从不指望每个人都对这些让我们着迷的技术问题感兴趣。尽管如此，我们回答问题的风格是指向那些真正对此有兴趣并愿意主动参与解决问题的人，这一点不会变，也不该变。如果连这都变了，我们就是在降低做自己最擅长的事情上的效率。

我们（在很大程度上）是自愿的，从繁忙的生活中抽出时间来解答疑惑，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像失败者的家伙，以便更高效的利用时间来回答 `赢家（winner）` 的问题。

如果你厌恶我们的态度，高高在上，或过于傲慢，不妨也设身处地想想。我们并没有要求你向我们屈服 -- 事实上，我们大多数人非常乐意与你平等地交流，只要你付出小小努力来满足基本要求，我们就会欢迎你加入我们的文化。但让我们帮助那些不愿意帮助自己的人是没有效率的。无知没有关系，但装白痴就是不行。

所以，你不必在技术上很在行才能吸引我们的注意，但你必须表现出能引导你变得在行的特质 -- 机敏、有想法、善于观察、乐于主动参与解决问题。如果你做不到这些使你与众不同的事情，我们建议你花点钱找家商业公司签个技术支持服务合同，而不是要求黑客个人无偿地帮助你。

如果你决定向我们求助，当然你也不希望被视为失败者，更不愿成为失败者中的一员。能立刻得到快速并有效答案的最好方法，就是像赢家那样提问 -- 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。你可以 email 你的建议至 esr@thyrsus.com 或 respond-auto@linuxmafia.com。然而请注意，本文并非网络礼节的通用指南，而我们通常会拒绝无助于在技术论坛得到有用答案的建议。）

在提问之前

在你准备要通过电子邮件、新闻群组或者聊天室提出技术问题前，请先做到以下事情：

1. 尝试在你准备提问的论坛的旧文章中搜索答案。
2. 尝试上网搜索以找到答案。
3. 尝试阅读手册以找到答案。
4. 尝试阅读常见问题文件（FAQ）以找到答案。
5. 尝试自己检查或试验以找到答案
6. 向你身边的强者朋友打听以找到答案。
7. 如果你是程序开发者，请尝试阅读源代码以找到答案

当你提出问题的时候，请先表明你已经做了上述的努力；这将有助于树立你并不是一个不劳而获且浪费别人的时间的提问者。如果你能一并表达在做了上述努力的过程中所*学到*的东西会更好，因为我们更乐于回答那些表现出能从答案中学习的人的问题。

运用某些策略，比如先用Google搜索你所遇到的各种错误信息（既搜索Google论坛，也搜索网页），这样很可能直接就找到了能解决问题的文件或邮件列表线索。即使没有结果，在邮件列表或新闻组寻求帮助时加上一句 `我在Google中搜过下列句子但没有找到什么有用的东西` 也是件好事，即使它只是表明了搜索引擎不能提供哪些帮助。这么做（加上搜索过的字串）也让遇到相似问题的其他人能被搜索引擎引导到你的提问来。

别着急，不要指望几秒钟的Google搜索就能解决一个复杂的问题。在向专家求助之前，再阅读一下常见问题文件（FAQ）、放轻松、坐舒服一些，再花点时间思考一下这个问题。相信我们，他们能从你的提问看出你做了多少阅读与思考，如果你是有备而来，将更有可能得到解答。不要将所有问题一股脑抛出，只因你的第一次搜索没有找到答案（或者找到太多答案）。

准备好你的问题，再将问题仔细的思考过一遍，因为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。越是能表现出在寻求帮助前你为解决问题所付出的努力，你越有可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，某个普通黑客（J. Random Hacker）多半会一边在心里想着`蠢问题...`，一边用无意义的字面解释来答复你，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。

绝不要自以为*够格*得到答案，你没有；你并没有。毕竟你没有为这种服务支付任何报酬。你将会是自己去*挣到*一个答案，靠提出有内涵的、有趣的、有思维激励作用的问题 --一个有潜力能贡献社区经验的问题，而不仅仅是被动的从他人处索取知识。

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？`、`我的这个例子里缺了什么？`以及`我应该检查什么地方`比`请把我需要的确切的过程贴出来`更容易得到答复。因为你表现出只要有人能指个正确方向，你就有完成它的能力和决心。

当你提问时

慎选提问的论坛

小心选择你要提问的场合。如果你做了下述的事情，你很可能被忽略掉或者被看作失败者：

- 在与主题不合的论坛上贴出你的问题
- 在探讨进阶技术问题的论坛张贴非常初级的问题；反之亦然
- 在太多的不同新闻群组上重复转贴同样的问题（cross-post）
- 向既非熟人也没有义务解决你问题的人发送私人电邮

黑客会剔除掉那些搞错场合的问题，以保护他们沟通的渠道不被无关的东西淹没。你不会想让这种事发生在自己身上的。

因此，第一步是找到对的论坛。再说一次，Google和其它搜索引擎还是你的朋友，用它们来找到与你遭遇到困难的软硬件问题最相关的网站。通常那儿都有常见问题（FAQ）、邮件列表及相关说明文件的链接。如果你的努力（包括*阅读*FAQ）都没有结果，网站上也许还有报告Bug（Bug-reporting）的流程或链接，如果是这样，连过去看看。

向陌生的人或论坛发送邮件最可能是风险最大的事情。举例来说，别假设一个提供丰富内容的网页的作者会想充当你的免费顾问。不要对你的问题是否会受到欢迎做太乐观的估计 -- 如果你不确定，那就向别处发送，或者压根别发。

在选择论坛、新闻群组或邮件列表时，别太相信名字，先看看FAQ或者许可书以弄清楚你的问题是否切题。发文前先翻翻已有的话题，这样可以让你感受一下那里的文化。事实上，事先在新闻组或邮件列表的历史记录中搜索与你问题相关的关键词是个极好的主意，也许这样就找到答案了。即使没有，也能帮助你归纳出更好的问题。

别像机关枪似的一次"扫射"所有的帮助渠道，这就像大喊大叫一样会使人不快。要一个一个地来。

搞清楚你的主题！最典型的错误之一是在某种致力于跨平台可移植的语言、套件或工具的论坛中提关于Unix或Windows操作系统程序界面的问题。如果你不明白为什么这是大错，最好在搞清楚这之间差异之前什么也别问。

一般来说，在仔细挑选的公共论坛中提问，会比在私有论坛中提同样的问题更容易得到有用的回答。有几个理由可以支持这点，一是看潜在的回复者有多少，二是看观众有多少。黑客较愿意回答那些能帮助到许多人的问题。

可以理解的是，老练的黑客和一些热门软件的作者正在接受过多的错发信息。就像那根最后压垮骆驼背的稻草一样，你的加入也有可能使情况走向极端 -- 已经好几次了，一些热门软件的作者从自己软件的支持中抽身出来，因为伴随而来涌入其私人邮箱的无用邮件变得无法忍受。

Stack Overflow

搜索，*然后*在 Stack Exchange 问。

近年来，Stack Exchange community 社区已经成为回答技术及其他问题的主要渠道，尤其是那些开放源码的项目。

因为 Google 索引是即时的，在看 Stack Exchange 之前先在 Google 搜索。有很高的机率某人已经问了一个类似的问题，而且 Stack Exchange 网站们往往会是搜索结果中最前面几个。如果你在 Google 上没有找到任何答案，你再到特定相关主题的网站去找。用标签（Tag）搜索能让你更缩小你的搜索结果。

Stack Exchange 已经成长到超过一百个网站，以下是最常用的几个站：

- Super User 是问一些通用的电脑问题，如果你的问题跟代码或是写程序无关，只是一些网络连线之类的，请到这里。
- Stack Overflow 是问写程序有关的问题。
- Server Fault 是问服务器和网管相关的问题。

网站和IRC论坛

本地的使用者群组（user group），或者你所用的 Linux 发行版本也许正在宣传他们的网页论坛或 IRC 频道，并提供新手帮助（在一些非英语国家，新手论坛很可能还是邮件列表）， 这些地方是开始提问的好首选，特别是当你觉得遇到的也许只是相对简单或者很普通的问题时。经过宣传的 IRC 频道是公开欢迎提问的地方，通常可以即时得到回应。

事实上，如果程序出的问题只发生在特定 Linux 发行版提供的版本（这很常见），最好先去该发行版的论坛或邮件列表中提问，再到程序本身的论坛或邮件列表提问。（否则）该项目的黑客可能仅仅回复 "用*我们的*版本"。

在任何论坛发文以前，先确认一下有没有搜索功能。如果有，就试着搜索一下问题的几个关键词，也许这会有帮助。如果在此之前你已做过通用的网页搜索（你也该这样做），还是再搜索一下论坛，搜索引擎有可能没来得及索引此论坛的全部内容。

通过论坛或 IRC 频道来提供使用者支持服务有增长的趋势，电子邮件则大多为项目开发者间的交流而保留。所以最好先在论坛或 IRC 中寻求与该项目相关的协助。

第二步，使用项目邮件列表

当某个项目提供开发者邮件列表时，要向列表而不是其中的个别成员提问，即使你确信他能最好地回答你的问题。查一查项目的文件和首页，找到项目的邮件列表并使用它。有几个很好的理由支持我们采用这种办法：

- 任何好到需要向个别开发者提出的问题，也将对整个项目群组有益。反之，如果你认为自己的问题对整个项目群组来说太愚蠢，也不能成为骚扰个别开发者的理由。
- 向列表提问可以分散开发者的负担，个别开发者（尤其是项目领导人）也许太忙以至于没法回答你的问题。
- 大多数邮件列表都会被存档，那些被存档的内容将被搜索引擎索引。如果你向列表提问并得到解答，将来其它人可以通过网页搜索找到你的问题和答案，也就不用再次发问了。
- 如果某些问题经常被问到，开发者可以利用此信息来改进说明文件或软件本身，以使其更清楚。如果只是私下提问，就没有人能看到最常见问题的完整场景。

如果一个项目既有"使用者" 也有"开发者"（或"黑客"）邮件列表或论坛，而你又不会动到那些源代码，那么就向"使用者"列表或论坛提问。不要假设自己会在开发者列表中受到欢迎，那些人多半会将你的提问视为干扰他们开发的噪音。

然而，如果你*确信*你的问题很特别，而且在"使用者" 列表或论坛中几天都没有回复，可以试试前往"开发者"列表或论坛发问。建议你在张贴前最好先暗地里观察几天以了解那里的行事方式（事实上这是参与任何私有或半私有列表的好主意）

如果你找不到一个项目的邮件列表，而只能查到项目维护者的电子邮件地址，尽管向他发信。即使是在这种情况下，也别假设（项目）邮件列表不存在。在你的电子邮件中，请陈述你已经试过但没有找到合适的邮件列表，也提及你不反对将自己的邮件转发给他人（许多人认为，即使没什么秘密，私人电子邮件也不应该被公开。通过允许将你的电子邮件转发他人，你给了相应人员处置你邮件的选择）。

使用有意义且描述明确的标题

在邮件列表、新闻群组或论坛中，大约50字以内的标题是抓住资深专家注意力的好机会。别用喋喋不休的`帮帮忙`、`跪求`、`急`（更别说`救命啊！！！！`这样让人反感的话，用这种标题会被条件反射式地忽略）来浪费这个机会。不要妄想用你的痛苦程度来打动我们，而是在这点空间中使用极简单扼要的描述方式来提出问题。

一个好标题范例是`目标 -- 差异`式的描述，许多技术支持组织就是这样做的。在`目标`部分指出是哪一个或哪一组东西有问题，在`差异`部分则描述与期望的行为不一致的地方。

> 蠢问题：救命啊！我的笔电不能正常显示了！

> 聪明问题：X.org 6.8.1的鼠标游标会变形，某牌显卡 MV1005 芯片组。

> 更聪明问题：X.org 6.8.1的鼠标游标，在某牌显卡 MV1005 芯片组环境下 - 会变形。

编写`目标 -- 差异`式描述的过程有助于你组织对问题的细緻思考。是什么被影响了？ 仅仅是鼠标游标或者还有其它图形？只在 X.org 的 X 版中出现？或只是出现在6.8.1版中？ 是针对某牌显卡芯片组？或者只是其中的 MV1005 型号？ 一个黑客只需瞄一眼就能够立即明白你的环境*和*你遇到的问题。

总而言之，请想像一下你正在一个只显示标题的存档讨论串（Thread）索引中查寻。让你的标题更好地反映问题，可使下一个搜索类似问题的人能够关注这个讨论串，而不用再次提问相同的问题。

如果你想在回复中提出问题，记得要修改内容标题，以表明你是在问一个问题， 一个看起来像`Re: 测试`或者`Re: 新bug`的标题很难引起足够重视。另外，在不影响连贯性之下，适当引用并删减前文的内容，能给新来的读者留下线索。

对于讨论串，不要直接点击回复来开始一个全新的讨论串，这将限制你的观众。因为有些邮件阅读程序，比如 mutt，允许使用者按讨论串排序并通过折叠讨论串来隐藏消息，这样做的人永远看不到你发的消息。

仅仅改变标题还不够。mutt 和其它一些邮件阅读程序还会检查邮件标题以外的其它信息，以便为其指定讨论串。所以宁可发一个全新的邮件。

在网页论坛上，好的提问方式稍有不同，因为讨论串与特定的信息紧密结合，并且通常在讨论串外就看不到里面的内容，故通过回复提问，而非改变标题是可接受的。不是所有论坛都允许在回复中出现分离的标题，而且这样做了基本上没有人会去看。不过，通过回复提问，这本身就是暧昧的做法，因为它们只会被正在查看该标题的人读到。所以，除非你*只想*在该讨论串当前活跃的人群中提问，不然还是另起炉灶比较好。

使问题容易回复

以`请将你的回复寄到……`来结束你的问题多半会使你得不到回答。如果你觉得花几秒钟在邮件客户端设置一下回复地址都麻烦，我们也觉得花几秒钟思考你的问题更麻烦。如果你的邮件程序不支持这样做，换个好点的；如果是操作系统不支持这种邮件程序，也换个好点的。

在论坛，要求通过电子邮件回复是非常无礼的，除非你相信回复的信息可能比较敏感（而且有人会为了某些未知的原因，只让你而不是整个论坛知道答案）。如果你只是想在有人回复讨论串时得到电子邮件提醒，可以要求网页论坛发送给你。几乎所有论坛都支持诸如`追踪此讨论串`、`有回复时发送邮件提醒`等功能。

用清晰、正确、精准并语法正确的语句

我们从经验中发现，粗心的提问者通常也会粗心的写程序与思考（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。

正确的拼字、标点符号和大小写是很重要的。一般来说，如果你觉得这样做很麻烦，不想在乎这些，那我们也觉得麻烦，不想在乎你的提问。花点额外的精力斟酌一下字句，用不着太僵硬与正式 -- 事实上，黑客文化很看重能准确地使用非正式、俚语和幽默的语句。但它*必须很*准确，而且有迹象表明你是在思考和关注问题。

正确地拼写、使用标点和大小写，不要将`its`混淆为`it's`，`loose`搞成`lose`或者将`discrete`弄成`discreet`。不要**全部用大写**，这会被视为无礼的大声嚷嚷（全部小写也好不到哪去，因为不易阅读。Alan Cox也许可以这样做，但你不行。）

更白话的说，如果你写得像是个半文盲[译注：小白]），那多半得不到理睬。也不要使用即时通讯中的简写或火星文，如将`的`简化为`ㄉ`会使你看起来像一个为了少打几个键而省字的小白。更糟的是，如果像个小孩似地鬼画符那绝对是在找死，可以肯定没人会理你（或者最多是给你一大堆指责与挖苦）。

如果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，我们通常能弄清两者的分别）。同时，除非你知道回复者使用的语言，否则请使用英语书写。繁忙的黑客一般会直接删除用他们看不懂语言写的消息。在网络上英语是通用语言，用英语书写可以将你的问题在尚未被阅读就被直接删除的可能性降到最低。

如果英文是你的外语（Second language），提示潜在回复者你有潜在的语言困难是很好的：
[译注：以下附上原文以供使用]

> English is not my native language; please excuse typing errors.

- 英文不是我的母语，请原谅我的错字或语法

> If you speak $LANGUAGE, please email/PM me;
> I may need assistance translating my question.

- 如果你说**某语言**，请寄信/私讯给我；我需要有人协助我翻译我的问题

> I am familiar with the technical terms,
> but some slang expressions and idioms are difficult for me.

- 我对技术名词很熟悉，但对于俗语或是特别用法比较不甚了解。

> I've posted my question in $LANGUAGE and English.
> I'll be glad to translate responses, if you only use one or the other.

- 我把我的问题用**某语言**和英文写出来，如果你只用一种语言回答，我会乐意将其翻译成另一种。

使用易于读取且标准的文件格式发送问题

如果你人为地将问题搞得难以阅读，它多半会被忽略，人们更愿读易懂的问题，所以：

- 使用纯文字而不是HTML (关闭HTML并不难）
- 使用MIME附件通常是可以的，前提是真正有内容（譬如附带的源代码或patch），而不仅仅是邮件程序生成的模板（譬如只是信件内容的拷贝）。
- 不要发送一段文字只是单行句子但多次断行的邮件（这使得回复部分内容非常困难）。设想你的读者是在80个字符宽的终端机上阅读邮件，最好设置你的断行点小于80字。
- 但是，也*不要*用任何固定断行资料（譬如日志档案拷贝或会话记录）。档案应该原样包含，让回复者有信心他们看到的是和你看到的一样的东西。
- 在英语论坛中，不要使用`Quoted-Printable` MIME编码发送消息。这种编码对于张贴非ASCII语言可能是必须的，但很多邮件程序并不支持这种编码。当它们分断时，那些文本中四处散布的`=20`符号既难看也分散注意

力，甚至有可能破坏内容的语意。
- 绝对，*永远*不要指望黑客们阅读使用封闭格式编写的文档，像是微软公司的Word或Excel文件等。大多数黑客对此的反应就像有人将还在冒热气的猪粪倒在你门口阶梯上时你的反应一样。即便他们能够处理，他们也很厌恶这么做。
- 如果你从使用Windows的电脑发送电子邮件，关闭微软愚蠢的`智能引号`功能 （从[选项] > [校订] > [自动校正选项], 按掉`智能引号`单选框），以免在你的邮件中到处散布垃圾字符。
- 在论坛，勿滥用`表情符号`和`HTML`功能（当它们提供时）。一两个表情符号通常没有问题，但花哨的彩色文本倾向于使人认为你是个无能之辈。过滥地使用表情符号、色彩和字体会使你看来像个傻笑的小姑娘。这通常不是个好主意，除非你只是对sex而不是有用的回复更有兴趣。

如果你使用图形用户界面的邮件程序（如微软公司的Outlook或者其它类似的），注意它们的默认设置不一定满足这些要求。大多数这类程序有基于选单的`查看源代码`命令，用它来检查发送文件夹中的消息，以确保发送的是没有多余杂质的纯文本文件。

精确的描述问题并言之有物

- 仔细、清楚地描述你的问题或Bug的症状。
- 描述问题发生的环境（机器配置、操作系统、应用程序、以及相关的信息），提供经销商的发行版和版本号（如：`Fedora Core 4`、`Slackware 9.1`等）。
- 描述在提问前你是怎样去研究和理解这个问题的。
- 描述在提问前为确定问题而采取的诊断步骤。
- 描述最近做过什么可能相关的硬件或软件变更。
- 尽可能的提供一个可以`重现这个问题的既定环境`的方法

尽量去揣测一个黑客会怎样反问你，在他提问的时候预先给他答案。

以上几点中，当你报告的是你认为可能在代码中的问题时，给黑客一个可以重现你的问题的环境尤其重要。当你这么做时，你得到有效的回答的机会和速度都会大大的提升。

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

话不在多而在精

你需要提供精确有内容的信息。这并不是要求你简单的把成堆的出错代码或者资料完全转录到你的提问中。如果你有庞大而复杂的测试样例能重现程序挂掉的情境，尽量将它剪裁得越小越好。

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，简化问题使你更有可能得到*有用*的答案；
第三，在精炼你的bug报告的过程中，你很可能就自己找到了解决方法或权宜之计。

别动辄声称找到Bug

当你在使用软件中遇到问题，除非你非常、*非常*的有根据，不要动辄声称找到了Bug。提示：除非你能提供解决问题的源代码补丁，或者对前一版本的回归测试表现出不正确的行为，否则你都多半不够完全确信。这同样适用在网页和文件，如果你（声称）发现了文件的`Bug`，你应该能提供相应位置的修正或替代文件。

请记得，还有许多其它使用者没遇到你发现的问题，否则你在阅读文件或搜索网页时就应该发现了（你在抱怨前已经做了这些，是吧？）。这也意味着很有可能是你弄错了而不是软件本身有问题。

编写软件的人总是非常辛苦地使它尽可能完美。如果你声称找到了Bug，也就是在质疑他们的能力，即使你是对的，也有可能会冒犯到其中某部分人。这尤其严重当你在标题中嚷嚷着有`Bug`。

提问时，即使你私下非常确信已经发现一个真正的Bug，最好写得像是*你*做错了什么。如果真的有Bug，你会在回复中看到这点。这样做的话，如果真有Bug，维护者就会向你道歉，这总比你惹恼别人然后欠别人一个道歉要好一点。

可以低声下气，但还是要先做功课

有些人明白他们不该粗鲁或傲慢的提问并要求得到答复，但他们选择另一个极端 -- 低声下气：`我知道我只是个可悲的新手，一个撸瑟，但...`。这既使人困扰，也没有用，尤其是伴随着与实际问题含糊不清的描述时更令人反感。

别用原始灵长类动物的把戏来浪费你我的时间。取而代之的是，尽可能清楚地描述背景条件和你的问题情况。这比低声下气更好地定位了你的位置。

有时网页论坛会设有专为新手提问的版面，如果你真的认为遇到了初学者的问题，到那去就是了，但一样别那么低声下气。

描述问题症状而非猜测

告诉黑客们你认为问题是怎样造成的并没什么帮助。（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，而不是你的解释和理论；让黑客们来推测和诊断。如果你认为陈述自己的猜测很重要，清楚地说明这只是你的猜测，并描述为什么它们不起作用。

蠢问题

> 我在编译内核时接连遇到 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

> 我的组装电脑是 FIC-PA2007 主机板搭载 AMD K6/233 CPU（威盛 Apollo VP2芯片组），
> 256MB Corsair PC133 SDRAM内存，在编译内核时，从开机20分钟以后就频频产生 SIG11 错误，
> 但是在头20分钟内从没发生过相同的问题。重新启动也没有用，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。相关部分的标准编译记录如下...。

由于以上这点似乎让许多人觉得难以配合，这里有句话可以提醒你：`所有的诊断专家都来自密苏里州。` 美国国务院的官方座右铭则是：`让我看看`（出自国会议员 Willard D. Vandiver 在1899年时的讲话：`我来自一个出产玉米，棉花，牛蒡和民主党人的国家，滔滔雄辩既不能说服我，也不会让我满意。我来自密苏里州，你必须让我看看。`） 针对诊断者而言，这并不是一种怀疑，而只是一种真实而有用的需求，以便让他们看到的是与你看到的原始证据尽可能一致的东西，而不是你的猜测与归纳的结论。所以，大方的展示给我们看吧！

按发生时间先后列出问题症状

问题发生前的一系列操作，往往就是对找出问题最有帮助的线索。因此，你的说明里应该包含你的操作步骤，以及机器和软件的反应，直到问题发生。在命令行处理的情况下，提供一段操作记录（例如运行脚本工具所生成的），并引用相关的若干行（如20行）记录会非常有帮助。

如果挂掉的程序有诊断选项（如 -v 的详述开关），试着选择这些能在记录中增加调试信息的选项。记住，`多`不等于`好`。试着选取适当的调试级别以便提供有用的信息而不是让读者淹没在垃圾中。

如果你的说明很长（如超过四个段落），在开头简述问题，接下来再按时间顺序详述会有所帮助。这样黑客们在读你的记录时就知道该注意哪些内容了。

描述目标而不是过程

如果你想弄清楚如何做某事（而不是报告一个Bug），在开头就描述你的目标，然后才陈述重现你所卡住的特定步骤。

经常寻求技术帮助的人在心中有个更高层次的目标，而他们在自以为能达到目标的特定道路上被卡住了，然后跑来问该怎么走，但没有意识到这条路本身就有问题。结果要费很大的劲才能搞定。

蠢问题

> 我怎样才能从某绘图程序的颜色选择器中取得十六进制的的RGB值？

聪明问题

> 我正试着用替换一幅图片的色码成自己选定的色码，我现在知道的唯一方法是编辑每个色码区块，
> 但却无法从某绘图程序的颜色选择器取得十六进制的的RGB值。

第二种提问法比较聪明，你可能得到像是`建议采用另一个更合适的工具`的回复。

别要求使用私人电邮回复

黑客们认为问题的解决过程应该公开、透明，此过程中如果更有经验的人注意到不完整或者不当之处，最初的回复才能够、也应该被纠正。同时，作为提供帮助者也能因为能力和学识被其它同行看到而得到某种奖励。

当你要求私下回复时，这个过程和奖励都被中止。别这样做，让*回复者*来决定是否私下回答 -- 如果他真这么做了，通常是因为他认为问题编写太差或者太肤浅，以至于对其它人没有兴趣。

这条规则存在一条有限的例外，如果你确信提问可能会引来大量雷同的回复时，那么这个神奇的提问句会是`向我发电邮，我将为论坛归纳这些回复`。试着将邮件列表或新闻群组从洪水般的雷同回复中解救出来是非常有礼貌的 -- 但你必须信守诺言。

清楚明确的表达你的问题以及需求

漫无边际的提问近乎无休无止的时间黑洞。最有可能给你有用答案的人通常也正是最忙的人（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相当厌恶，所以他们也倾向于厌恶那些漫无边际的提问。

如果你明确表述需要回答者做什么（如提供指点、发送一段代码、检查你的补丁、或是其他等等），就最有可能得到有用的答案。因为这会定出一个时间和精力的上限，便于回答者能集中精力来帮你。这么做很棒。

要理解专家们所处的世界，请把专业技能想像为充裕的资源，而回复的时间则是稀缺的资源。你要求他们奉献的时间越少，你越有可能从真正专业而且很忙的专家那里得到解答。

所以，界定一下你的问题，使专家花在辨识你的问题和回答所需要付出的时间减到最少，这技巧对你有用答案相当有帮助 -- 但这技巧通常和简化问题有所区别。因此，问`我想更好的理解x，可否指点一下哪有好一点说明？`通常比问`你能解释一下x吗？`更好。如果你的代码不能运作，通常请别人看看哪里有问题，比要求别人替你改正要明智得多。

询问有关代码的问题时

别要求他人帮你有问题的代码调试而不提示一下应该从何入手。张贴几百行的代码，然后说一声：`它不会动`会让你完全被忽略。只贴几十行代码，然后说一句：`在第七行以后，我期待它显示 <x>，但实际出现的是 <y>`比较有可能让你得到回应。

最有效描述程序问题的方法是提供最精简的Bug展示测试示例（bug-demonstrating test case）。什么是最精简的测试示例? 那是问题的缩影；一小个程序片段能**刚好**展示出程序的异常行为，而不包含其他令人分散注意力的内容。怎么制作最精简的测试示例？如果你知道哪一行或哪一段代码会造成异常的行为，复制下来并加入足够重现这个状况的代码（例如，足以让这段代码能被编译/直译/被应用程序处理）。如果你无法将问题缩减到一个特定区块，就复制一份代码并移除不影响产生问题行为的部分。总之，测试示例越小越好（查看话不在多而在精一节）。

一般而言，要得到一段相当精简的测试示例并不太容易，但永远先尝试这样做的是种好习惯。这种方式可以帮助你了解如何自行解决这个问题 —- 而且即使你的尝试不成功，黑客们也会看到你在尝试取得答案的过程中付出了努力，这可以让他们更愿意与你合作。

如果你只是想让别人帮忙审查（Review）一下代码，在信的开头就要说出来，并且一定要提到你认为哪一部分特别需要关注以及为什么。

别把自己家庭作业的问题贴上来

黑客们很擅长分辨哪些问题是家庭作业式的问题；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。

如果你怀疑自己碰到了一个家庭作业式的问题，但仍然无法解决，试试在使用者群组，论坛或（最后一招）在项目的使用者邮件列表或论坛中提问。尽管黑客们*会*看出来，但一些有经验的使用者也许仍会给你一些提示。

去掉无意义的提问句

避免用无意义的话结束提问，例如`有人能帮我吗？`或者`这有答案吗？`。

首先：如果你对问题的描述不是很好，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，但毫无意义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`。

一般来说，避免用`是或否`、`对或错`、`有或没有`类型的问句，除非你想得到是或否类型的回答。

即使你很急也不要在标题写`紧急`

这是你的问题，不是我们的。宣称`紧急`极有可能事与愿违：大多数黑客会直接删除无礼和自私地企图即时引起关注的问题。更严重的是，`紧急`这个字（或是其他企图引起关注的标题）通常会被垃圾信过滤器过滤掉 -- 你希望能看到你问题的人可能永远也看不到。

有半个例外的情况是，如果你是在一些很高调，会使黑客们兴奋的地方，也许值得这样去做。在这种情况下，如果你有时间压力，也很有礼貌地提到这点，人们也许会有兴趣回答快一点。

当然，这风险很大，因为黑客们兴奋的点多半与你的不同。譬如从 NASA 国际空间站（International Space Station）发这样的标题没有问题，但用自我感觉良好的慈善行为或政治原因发肯定不行。事实上，张贴诸如`紧急：帮我救救这个毛绒绒的小海豹！`肯定让你被黑客忽略或惹恼他们，即使他们认为毛绒绒的小海豹很重要。

如果你觉得这点很不可思议，最好再把这份指南剩下的内容多读几遍，直到你弄懂了再发文。

礼多人不怪，而且有时还很有帮助

彬彬有礼，多用`请`和`谢谢您的关注`，或`谢谢你的关照`。让大家都知道你对他们花时间免费提供帮助心存感激。

坦白说，这一点并没有比清晰、正确、精准并合法语法和避免使用专用格式重要（也不能取而代之）。黑客们一般宁可读有点唐突但技术上鲜明的Bug报告，而不是那种有礼但含糊的报告。（如果这点让你不解，记住我们是按问题能教我们什么来评价问题的价值的）

然而，如果你有一串的问题待解决，客气一点肯定会增加你得到有用回应的机会。

（我们注意到，自从本指南发布后，从资深黑客那里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`意味着事后就不用再感谢任何人的暗示。我们的建议是要么先说`先谢了`，*然后*事后再对回复者表示感谢，或者换种方式表达感激，譬如用`谢谢你的关注`或`谢谢你的关照`。）

问题解决后，加个简短的补充说明

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，应该在那里贴一个说明比较恰当。

最理想的方式是向最初提问的话题回复此消息，并在标题中包含`已修正`，`已解决`或其它同等含义的明显标记。在人来人往的邮件列表里，一个看见讨论串`问题 X`和`问题的X - 已解决`的潜在回复者就明白不用再浪费时间了（除非他个人觉得`问题 X`的有趣），因此可以利用此时间去解决其它问题。

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`比什么也不说要来的好。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大论更好。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。

对于有深度的问题，张贴调试记录的摘要是有帮助的。描述问题的最终状态，说明是什么解决了问题，在此*之后*才指明可以避免的盲点。避免盲点的部分应放在正确的解决方案和其它总结材料之后，而不要将此信息搞成侦探推理小说。列出那些帮助过你的名字，会让你交到更多朋友。

除了有礼貌和有内涵以外，这种类型的补充也有助于他人在邮件列表/新闻群组/论坛中搜索到真正解决你问题的方案，让他们也从中受益。

至少，这种补充有助于让每位参与协助的人因问题的解决而从中得到满足感。如果你自己不是技术专家或者黑客，那就相信我们，这种感觉对于那些你向他们求助的大师或者专家而言，是非常重要的。问题悬而未决会让人灰心；黑客们渴望看到问题被解决。好人有好报，满足他们的渴望，你会在下次提问时尝到甜头。

思考一下怎样才能避免他人将来也遇到类似的问题，自问写一份文件或加个常见问题（FAQ）会不会有帮助。如果是的话就将它们发给维护者。

在黑客中，这种良好的后继行动实际上比传统的礼节更为重要，也是你如何透过善待他人而赢得声誉的方式，这是非常有价值的资产。

如何解读答案

RTFM和STFW：如何知道你已完全搞砸了

有一个古老而神圣的传统：如果你收到`RTFM （Read The Fucking Manual）`的回应，回答者认为你**应该去读他妈的手册**。当然，基本上他是对的，你应该去读一读。

RTFM 有一个年轻的亲戚。如果你收到`STFW（Search The Fucking Web）`的回应，回答者认为你**应该到他妈的网上搜索**过了。那人多半也是对的，去搜索一下吧。（更温和一点的说法是 **Google是你的朋友**！）

在论坛，你也可能被要求去爬爬论坛的旧文。事实上，有人甚至可能热心地为你提供以前解决此问题的讨论串。但不要依赖这种关照，提问前应该先搜索一下旧文。

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也正在读着。这些答复意味着回答者认为

- 你需要的信息非常容易获得；
- 你自己去搜索这些信息比灌给你能让你学到更多。

你不应该因此不爽；**依照黑客的标准，他已经表示了对你一定程度的关注，而没有对你的要求视而不见**。你应该对他祖母般的慈祥表示感谢。

如果还是搞不懂

如果你看不懂回应，别立刻要求对方解释。像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），先试着去搞懂他的回应。如果你真的需要对方解释，记得表现出你已经从中学到了点什么。

比方说，如果我回答你：`看来似乎是 zentry 卡住了；你应该先清除它。`，然后，这是一个*很糟的*后续问题回应：`zentry是什么？` *好*的问法应该是这样：`哦~~~我看过说明了但是只有 -z 和 -p 两个参数中提到了 zentries，而且还都没有清楚的解释如何清除它。你是指这两个中的哪一个吗？还是我看漏了什么？`

处理无礼的回应

很多黑客圈子中看似无礼的行为并不是存心冒犯。相反，它是直接了当，一针见血式的交流风格，这种风格更注重解决问题，而不是使人感觉舒服而却模模糊糊。

如果你觉得被冒犯了，试着平静地反应。如果有人真的做了出格的事，邮件列表、新闻群组或论坛中的前辈多半会招呼他。如果这*没有*发生而你却发火了，那么你发火对象的言语可能在黑客社区中看起来是正常的，而*你*将被视为有错的一方，这将伤害到你获取信息或帮助的机会。

另一方面，你偶而真的会碰到无礼和无聊的言行。与上述相反，对真正的冒犯者狠狠地打击，用犀利的语言将其驳得体无完肤都是可以接受的。然而，在行事之前一定要非常非常的有根据。纠正无礼的言论与开始一场毫无意义的口水战仅一线之隔，黑客们自己莽撞地越线的情况并不鲜见。如果你是新手或外人，避开这种莽撞的机会并不高。如果你想得到的是信息而不是消磨时光，这时最好不要把手放在键盘上以免冒险。

（有些人断言很多黑客都有轻度的自闭症或亚斯伯格综合症，缺少用于润滑人类社会**正常**交往所需的神经。这既可能是真也可能是假的。如果你自己不是黑客，兴许你认为我们脑袋有问题还能帮助你应付我们的古怪行为。只管这么干好了，我们不在乎。我们*喜欢*我们现在这个样子，并且通常对病患标记都有站得住脚的怀疑。）

在下一节，我们会谈到另一个问题，当*你*行为不当时所会受到的`冒犯`。

如何避免扮演失败者

在黑客社区的论坛中有那么几次你可能会搞砸 -- 以本指南所描述到的或类似的方式。而你会在公开场合中被告知你是如何搞砸的，也许攻击的言语中还会带点夹七夹八的颜色。

这种事发生以后，你能做的最糟糕的事莫过于哀嚎你的遭遇、宣称被口头攻击、要求道歉、高声尖叫、憋闷气、威胁诉诸法律、向其雇主报怨、忘了关马桶盖等等。相反地，你该这么做：

熬过去，这很正常。事实上，它是有益健康且合理的。

社区的标准不会自行维持，它们是通过参与者积极而*公开地*执行来维持的。不要哭嚎所有的批评都应该通过私下的邮件传送，它不是这样运作的。当有人评论你的一个说法有误或者提出不同看法时，坚持声称受到个人攻击也毫无益处，这些都是失败者的态度。

也有其它的黑客论坛，受过高礼节要求的误导，禁止参与者张贴任何对别人帖子挑毛病的消息，并声称`如果你不想帮助用户就闭嘴。` 结果造成有想法的参与者纷纷离开，这么做只会使它们沦为毫无意义的唠叨与无用的技术论坛。

夸张的讲法是：你要的是**友善**（以上述方式）还是有用？两个里面挑一个。

记着：当黑客说你搞砸了，并且（无论多么刺耳）告诉你别再这样做时，他正在为关心**你**和**他的社区**而行动。对他而言，不理你并将你从他的生活中滤掉更简单。如果你无法做到感谢，至少要表现地有点尊严，别大声哀嚎，也别因为自己是个有戏剧性超级敏感的灵魂和自以为有资格的新来者，就指望别人像对待脆弱的洋娃娃那样对你。

有时候，即使你没有搞砸（或者只是在他的想像中你搞砸了），有些人也会无缘无故地攻击你本人。在这种情况下，抱怨倒是*真的*会把问题搞砸。

这些来找麻烦的人要么是毫无办法但自以为是专家的不中用家伙，要么就是测试你是否真会搞砸的心理专家。其它读者要么不理睬，要么用自己的方式对付他们。这些来找麻烦的人在给他们自己找麻烦，这点你不用操心。

也别让自己卷入口水战，最好不要理睬大多数的口水战 -- 当然，是在你检验它们只是口水战，而并未指出你有搞砸的地方，且也没有巧妙地将问题真正的答案藏于其后（这也是有可能的）。

不该问的问题

以下是几个经典蠢问题，以及黑客没回答时心中所想的：

问题：我能在哪找到 X 程序或 X 资源？

问题：我怎样用 X 做 Y？

问题：如何设定我的 shell 提示？

问题：我可以用 Bass-o-matic 文件转换工具将 AcmeCorp 档案转换为 TeX 格式吗？

问题：我的程序/设定/SQL语句没有用

问题：我的 Windows 电脑有问题，你能帮我吗？

问题：我的程序不会动了，我认为系统工具 X 有问题

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？

> 问题：我能在哪找到 X 程序或 X 资源？

回答：就在我找到它的地方啊，白痴 -- 搜索引擎的那一头。天哪！难道还有人不会用 Google 吗？

> 问题：我怎样用 X 做 Y？

回答：如果你想解决的是 Y，提问时别给出可能并不恰当的方法。这种问题说明提问者不但对 X 完全无知，也对 Y 要解决的问题糊涂，还被特定形势禁锢了思维。最好忽略这种人，等他们把问题搞清楚了再说。

> 问题：如何设定我的 shell 提示？？

回答：如果你有足够的智慧提这个问题，你也该有足够的智慧去 RTFM，然后自己去找出来。

> 问题：我可以用 Bass-o-matic 文件转换工具将 AcmeCorp 档案转换为 TeX 格式吗？

回答：试试看就知道了。如果你试过，你既知道了答案，就不用浪费我的时间了。

> 问题：我的程序/设定/SQL语句没有用

回答：这不算是问题吧，我对要我问你二十个问题才找得出你真正问题的问题没兴趣 -- 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

- 你还有什么要补充的吗？
- 真糟糕，希望你能搞定。
- 这关我有什么屁事？

> 问题：我的 Windows 电脑有问题，你能帮我吗？

回答：能啊，扔掉萎软的垃圾，换个像 Linux 或 BSD 的开放源代码操作系统吧。

注意：如果程序有官方版 Windows 或者与 Windows 有互动（如Samba），你 *可以*问与Windows相关的问题， 只是别对问题是由 Windows 操作系统而不是程序本身造成的回复感到惊讶， 因为 Windows 一般来说实在太烂，这种说法通常都是对的。

> 问题：我的程序不会动了，我认为系统工具 X 有问题

回答：你完全有可能是第一个注意到被成千上万用户反复使用的系统调用与函数库档案有明显缺陷的人，更有可能的是你完全没有根据。不同凡响的说法需要不同凡响的证据，当你这样声称时，你必须有清楚而详尽的缺陷说明文件作后盾。

> 问题：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 使用群组者寻求实际的指导吧（你能在这儿找到使用者群组的清单）。

注意：如果安装问题与某 Linux 的发行版有关，在它的邮件列表、论坛或本地使用者群组中提问也许是恰当的。此时，应描述问题的准确细节。在此之前，先用 `Linux` 和*所有*被怀疑的硬件作关键词仔细搜索。

> 问题：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，说明了你是个卑鄙小人；想找个黑客帮你，说明你是个白痴！

好问题与蠢问题

最后，我将透过举一些例子，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，这个傲慢自大的提问者

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，我有什么做的不对的地方吗？

提问者已经指明了环境，也读过了FAQ，还列出了错误，并且他没有把问题的责任推到别人头上，他的问题值得被关注。

蠢问题

> 我的主机板有问题了，谁来帮我？

某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 主机板上试过了 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 florbish 正在 grommicking，但结果出人意料。通常在 Athlon MP 主机板上引起 grommicking 的原因是什么？有谁知道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（lkml）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这一问题的重要信息。

通过我的提问方法，我给了别人可以咀嚼玩味的东西；我设法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，并邀请他们与我共同探讨。通过告诉他们我所走过的弯路，以避免他们再浪费时间，我也表明了对他们宝贵时间的尊重。

事后，当我向每个人表示感谢，并且赞赏这次良好的讨论经历的时候， 一个 Linux 内核邮件列表的成员表示，他觉得我的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。

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*像*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，这直接导致了本指南的出现。

如果得不到回答

如果仍得不到回答，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。

总的来说，简单的重复张贴问题是个很糟的点子。这将被视为无意义的喧闹。有点耐心，知道你问题答案的人可能生活在不同的时区，可能正在睡觉，也有可能你的问题一开始就没有组织好。

你可以通过其他渠道获得帮助，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使用者群组，由热情的软件爱好者（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。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。

对像是 Linux 这种大众化的软件，每个开发者至少会对应到上万名使用者。根本不可能由一个人来处理来自上万名使用者的求助电话。要知道，即使你要为这些协助付费，和你所购买的同类软件相比，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的要高得多，且内容也没那么丰富）。

如何更好地回答问题

态度和善一点。问题带来的压力常使人显得无礼或愚蠢，其实并不是这样。

对初犯者私下回复。对那些坦诚犯错之人没有必要当众羞辱，一个真正的新手也许连怎么搜索或在哪找常见问题都不知道。

如果你不确定，一定要说出来！一个听起来权威的错误回复比没有还要糟，别因为听起来像个专家很好玩，就给别人乱指路。要谦虚和诚实，给提问者与同行都树个好榜样。

如果帮不了忙，也别妨碍他。不要在实际步骤上开玩笑，那样也许会毁了使用者的设置 --有些可怜的呆瓜会把它当成真的指令。

试探性的反问以引出更多的细节。如果你做得好，提问者可以学到点东西 --你也可以。试试将蠢问题转变成好问题，别忘了我们都曾是新手。

尽管对那些懒虫抱怨一声 RTFM 是正当的，能指出文件的位置（即使只是建议个 Google 搜索关键词）会更好。

如果你决定回答，就请给出好的答案。当别人正在用错误的工具或方法时别建议笨拙的权宜之计（wordaround），应推荐更好的工具，重新界定问题。

正面的回答问题！如果这个提问者已经很深入的研究而且也表明已经试过 X 、 Y 、 Z 、 A 、 B 、 C 但没得到结果，回答 `试试看 A 或是 B` 或者 `试试X 、 Y 、 Z 、 A 、 B 、 C` 并附上一个链接一点用都没有。

帮助你的社区从问题中学习。当回复一个好问题时，问问自己`如何修改相关文件或常见问题文件以免再次解答同样的问题？`，接着再向文件维护者发一份补丁。

如果你是在研究一番后才做出的回答，*展现你的技巧而不是直接端出结果*。毕竟`授人以鱼不如授人以渔`。

相关资源

如果你需要个人电脑、Unix 系统和网络如何运作的基础知识，参阅Unix系统和网络基本原理。

当你发布软件或补丁时，试着按软件发布实践操作。

鸣谢

Evelyn Mitchel贡献了一些愚蠢问题例子并启发了编写`如何更好地回答问题`这一节， Mikhail Ramendik贡献了一些特别有价值的建议和改进。