

# 提问的智慧

---

艾瑞克.史蒂文.雷蒙德 (Eric Steven Raymond)

Thyrsus Enterprises

[esr@thyrsus.com](mailto:esr@thyrsus.com)

瑞克.莫恩 (Rick Moen)

[respond-auto@linuxmafia.com](mailto:respond-auto@linuxmafia.com)

版权©2001, 2006 Eric S. Raymond, Rick Moen

翻译: 王刚 <yafrank at 126 dot com >

## 目录

- [弃权申明](#)
- [引言](#)
- [提问前](#)
- [提问时](#)
  - [仔细挑选论坛](#)
  - [面向新手的论坛和互联网中继聊天 \(IRC\) 通常响应最快](#)
  - [第二步, 使用项目的邮件列表](#)
  - [使用有意义且明确的主题](#)
  - [使问题容易回复](#)
  - [用清晰、语法、拼写正确的语句书写](#)
  - [使用易于读取且标准的文件格式发送问题](#)
  - [描述问题应准确且有内容](#)
  - [量不在多, 精炼则灵](#)
  - [别急于宣称找到臭虫](#)
  - [低声下气代替不了做自己的家庭作业](#)
  - [描述问题症状而不是猜测](#)
  - [按时间先后罗列问题症状](#)
  - [描述目标而不是过程](#)
  - [别要求私下回复电邮](#)
  - [提问应明确](#)
  - [关于代码的问题](#)
  - [别张贴家庭作业式问题](#)
  - [删除无意义的要求](#)
  - [不要把问题标记为“紧急”, 即使对你而言的确如此](#)
  - [礼貌总是有益的](#)
  - [问题解决后追加一条简要说明](#)
- [如何解读回答](#)
  - [“读读该死的手册” \(RTFM\) 和“搜搜该死的网络” \(STFW\) : 如何明白你已完全搞砸](#)
  - [如果还不明白.....](#)

- [对待无礼](#)
- [别象失败者那样反应](#)
- [提问禁忌](#)
- [好问题与坏问题](#)
- [如果得不到回答](#)
- [如何更好地回答](#)
- [相关资源](#)
- [鸣谢](#)

译文：印尼语 白俄罗斯语 巴西葡萄牙语 简体中文 荷兰语 法语 乔治亚语 德语 希腊语 希伯来语 日语 波兰语 葡萄牙语 罗马尼亚语 俄语 西班牙语 泰语 如果你想复制、镜像、翻译或引用本文，请参阅我的 [复制协议](#)。

## 弃权申明

许多项目的网站在如何取得帮助的部分链接了本文，这没有关系，也正是我们想要的。但如果你是该项目生成此链接的网管，请在链接附近显著位置注明：我们不提供该项目的服务支持！

我们已经领教了没有此说明带来的痛苦，我们将不停地被一些白痴纠缠，他们认为既然我们发布了本文，那么我们就有责任解决世上所有的技术问题。

如果你是因为需要帮助正在阅读本文，然后就带着可以直接从作者那取得帮助的印象离开，那么 你 就不幸成了我们所说的白痴之一。别向 我们 提问，我们不会理睬的。我们只是在这教你如何从那些真正懂得你软硬件问题的人那里取得帮助，但 99.9% 的时间我们不会是那些人。除非你非常地 确定 本文的作者是 你遇到问题方面的专家，请不要打搅，这样大家都更开心一点。

## 引言

在 黑客 的世界里，你所提技术问题的解答很大程度上取决于你提问的方式与解决此问题的难度，本文将教你如何提问才更有可能得到满意的答复。

开源程序的应用已经很广，你通常可以从其他更有经验的用户而不是黑客那里得到解答。这是好事，他们一般对新手常有的毛病更容忍一点。然尔，使用我们推荐的方法，象对待黑客那样对待这些有经验的用户，通常能最有效地得到问题的解答。

第一件需要明白的事是黑客喜欢难题和激发思考的好问题。假如不是这样，我们也不会写本文了。如果你能提出一个有趣的问题让我们咀嚼玩味，我们会感激你。好问题是种激励与礼物，帮助我们发展认知，揭示没有注意或想到的问题。在黑客中，“好问题！”是非常热烈而真挚的赞许。

此外，黑客还有遇到简单问题就表现出敌视或傲慢的名声。有时，我们看起来还对新手和愚蠢的家伙有条件反射式的无礼，但事情并不真是这样。

我们只是毫无歉意地敌视那些提问前不愿思考、不做自己家庭作业的人。这种人就象时间无底洞——他们只知道索取，不愿意付出，他们浪费了时间，这些时间本可用于其它更有趣的问题或更值得回答的人。我们将这种人叫做“失败者 (loser)”（由于历史原因，我们有时将“loser”拼写为“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，计算机只是种工具，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，我们承认这点，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。不过，我们回答问题的风格是为了适应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，我们就会在自己能做得最好的事情上不再那么犀利。

我们（大多数）是自愿者，从自己繁忙的生活中抽时间来回答问题，有时会力不从心。因此，我们会毫不留情地滤除问题，特别是那些看起来象是失败者提的，以便更有效地把回答问题的时间留给那些胜利者。

如果你认为这种态度令人反感、以施惠者自居或傲慢自大，请检查你的假设，我们并未要求你屈服——事实上，假如你做了该做的努力，我们中的大多数将非常乐意平等地与你交流，并欢迎你接纳我们的文化。试图去帮助那些不愿自救的人对我们简直没有效率。不懂没有关系，但愚蠢地做事不行。

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在行的姿态——机敏、有想法、善于观察、乐于主动参与问题的解决。如果你做不到这些使你与众不同的事情，我们建议你付钱跟别人签商业服务合同，而不是要求黑客无偿帮助。

如果你决定向我们求助，你不会想成为一名失败者，你也不想被看成一个失败者。得到快速有效回答的最好方法是使提问者看起来象个聪明、自信和有想法的人，并且暗示只是碰巧在某一特别问题上需要帮助。

（欢迎对本文指正，可以将建议发至 [esr@thyrsus.com](mailto:esr@thyrsus.com) 或 [respond-auto@linuxmafia.com](mailto:respond-auto@linuxmafia.com)。请注意，本文不想成为一般性的网络礼仪指南，我一般会拒绝那些与引出技术论坛中有用的回答不特别相关的建议。）

## 提问前

在通过电邮、新闻组或论坛提技术问题以前，做以下事情：

- 尝试在你准备提问论坛的历史文档中搜索答案
- 尝试搜索互联网以找到答案
- 尝试阅读手册以找到答案
- 尝试阅读“常见问题文档”（FAQ）以找到答案
- 尝试自己检查或试验以找到答案
- 尝试请教懂行的朋友以找到答案
- 如果你是程序员，尝试阅读源代码以找到答案

提问时，请先表明你已做了上述事情，这将有助于建立你不是寄生虫与浪费别人时间的印象。最好再表述你从中学到的东西，我们喜欢回答那些表现出能从答案中学习的人。

运用某些策略，比如用谷歌（Google）搜索你遇到的各种错误提示（既搜索谷歌论坛，也搜索网页），这样很可能直接就找到了解决问题的文档或邮件列表线索。即使没有结果，在邮件列表或新闻组寻求帮助时提一句“我在谷歌中搜过下列句子但没有找到什么有用的东西”也是件好事，至少它表明了搜索引擎不能提供哪些帮助。将搜索关键词与你的问题及可能的解决方案联系起来，还有助于引导其他有类似问题的人。

别着急，不要指望几秒钟的谷歌搜索就能解决一个复杂的问题。读一下常见问题文档。在向专家提问之前，先向后靠靠放松一下，再思考一下问题。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结果（或者结果太多）。

认真地思考，准备好你的问题。轻率的提问只能得到轻率的回答，或者压根没有。在提问时，你越是表现出在此前做过思考与努力去解决自己的问题，你越有可能得到真正的帮助。

注意别提错问题。如果提问基于错误的假设，某黑客多半会一边想“愚蠢的问题……”，一边按将错就错的答案回复你，并且希望这种只是得到你自己“问的问题”而非真正所需的解答，给你一个教训。

永远不要假设你有资格得到解答。你没有这种资格，毕竟你没有为此服务付费。如果你能够提出有内容、有趣和激励思考的问题——那种毫无疑问能够向社区贡献经验，而不仅仅是消极地要求从别人那获取知识的问题，你将“挣到”答案。

另一方面，表明你有能力也乐意参与问题的解决是个很好的开端。“有没有人能指个方向？”，我这还差点什么？”，“我应该查哪个网站？”，通常要比“请给出我可以用的完整步骤”更容易得到回复，因为你表明了只要有人能指个方向，你就很乐意完成剩下的过程。

## 提问时

### 仔细挑选论坛

要对在哪提问留心，如果你做了下述事情，多半会被一笔勾销或被看成“失败者”：

- 张贴与论坛主题无关的问题
- 在面向高级技术问题的论坛上张贴肤浅的问题，或者反之。
- 在太多不同的新闻组同时张贴
- 给既非熟人也没有义务解决你问题的人发送你私人的电邮

为保护通信的渠道不被无关的东西淹没，黑客会除掉那些没有找对地方的问题，你不会想让这种事落到自己头上的。

因此，第一步是找对论坛。谷歌和其它搜索引擎还是你的朋友，可以用它们搜索你遇到困难的软硬件问题最相关的项目网站。那里通常都有项目的常见问题（FAQ）、邮件列表及文档的链接。如果你的努力（包括阅读 FAQ）都没有结果，这些邮件列表就是最后能取得帮助的地方。项目的网站也许还有报告臭虫的流程或链接，如果是这样，去看看。

向陌生的人或论坛发送邮件极有可能是在冒险。譬如，不要假设一个内容丰富的网页的作者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，向别处发或者压根别发。

在选择论坛、新闻组或邮件列表时，别太相信名字，先看看 FAQ 或者许可书以明确你的问题是否切题。发贴前先翻翻已有的帖子，这样可以让你感受一下那里行事的方式。事实上，张贴前在新闻组或邮件列表的历史文档中搜索与你问题相关的关键词是个极好的主意，也许就找到答案了。即使没有，也能帮助你归纳出更好的问题。

别象机关枪似的一次性“扫射”所有的帮助渠道，这就象大喊大叫一样会令人不快，温柔地一个一个来。

弄懂主题！最典型的错误之一是在某种致立于跨平台可移植的语言、库或工具的论坛中提关于 Unix 或 Windows 操作系统程序接口的问题。如果你不明白为什么这是大错，最好在搞清楚概念前什么也别问。

一般来说，在仔细挑选的公共论坛中提问比在私有论坛中提同样的问题更容易得到有用的回答。有几个道理支持这点，一是看潜在的回复者有多少，二是看论坛的参与者有多少，黑客更愿回答能启发多数人的问题。

可以理解，老练的黑客和一些流行软件的作者正在承受过多的不当消息。就象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，一些流行软件的作者退出了对自己软件的支持，因为伴随而来的涌入其私人邮箱的垃圾邮件变得无法忍受。

### 面向新手的论坛和互联网中继聊天（IRC）通常响应最快

本地的用户组织或者你所用的 Linux 发行版也许正在宣传新手取得帮助的论坛或 IRC 通道（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去处，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 通道是公开邀请提问的地方，通常可以得到实时的回复。

事实上，如果出问题的程序来自某发行版（这很常见），最好先去该发行版的论坛或邮件列表中提问，再到程序本身的项目论坛或邮件列表，（否则）该项目的黑客可能仅仅回复“用我们的代码”。

在任何论坛发帖以前，先看看有没有搜索功能。如果有，就试着用问题的几个关键词搜索一下，也许就有帮助。如果在此之前你已做过全面的网页搜索（你应该这样做），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 通道提供项目的用户支持有增长的趋势，电子邮件交流则更多地为项目开发者保留。所以先在论坛或 IRC 中寻求与该项目相关的帮助。

## 第二步，使用项目的邮件列表

当某个项目存在开发者邮件列表时，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。查一查项目的文档和主页，找到项目的邮件列表并使用它。采用这种办法有几个很好的理由：

- 向个别开发者提的问题（如果）足够好，也将对整个项目组有益。相反，如果你认为自己的问题对整个项目组来说太愚蠢，这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）也许太忙以至于没法回答你的问题。
- 大多数邮件列表都要存档，那些存档将被搜索引擎索引，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，也就不需要再次发问了。
- 如果某些问题经常被问到，开发者可以利用此信息改进文档或软件本身，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“用户”也有“开发者”（或“黑客”）邮件列表或论坛，而你又不摆弄那些代码，向“用户”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，那些人多半会遭受你的噪音干扰。

然尔，如果你确信你的问题不一般，而且在“用户”列表或论坛中几天都没有回复，可以试试“开发者”列表或论坛。建议你在张贴前最好先暗暗地观察几天，至少看看最近几天保存的帖子，以了解那的行事方式（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地址，只管向其发信。即便在这种情况下，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）。

## 使用有意义且明确的主题

在邮件列表、新闻组或论坛中，主题是你五十个或更少的字以内吸引有资格专家注意的黄金机会，不要用诸如“请帮我”（更别提大写的“请帮我！！！”），这种主题的消息会被条件反射式地删掉）之类的唠叨浪费机会。不要用你痛苦的深度来打动我们，相反，要在这点空间中使用超级简明扼要的问题描述。

使用主题的好惯例是“对象——偏差”（式的描述），许多技术支持组织就是这样做的。在“对象”部分指明是哪一个或哪一组东西有问题，在“偏差”部分则描述与期望的行为不一致的地方。

愚蠢：

救命啊！我的笔记本视频工作不正常！

明智：

X.org 6.8.1 扭曲鼠标光标，MV1005 型号的某显卡芯片组

更明智：

使用 MV1005 型号的某显卡芯片组在 X.org 6.8.1 的鼠标光标被扭曲



编写“对象——偏差”式描述的过程有助于你组织对问题的细致思考。是什么被影响了？仅仅是鼠标光标或者还有其它图形？只在 X.org 中出现？或只是在其 6.8.1 版中？是针对某显卡芯片组？或者只是其中的 MV1005 型号？一个黑客只需描一眼就能够立即明白什么是你遇到的问题，什么是你自己的问题。

更一般地，想象一下在一个只显示主题的文档索引中查找。让你的主题更好地反映问题，可以使下一个搜索类似问题的人能够在文档中直接就找到答案的线索，而不用再次发帖提问。

如果你想在回复中提问，确保改变主题以表明你是在问一个问题，一个主题象 `Re: 测试` 或者 `Re: 新臭虫` 的消息不太可能引起足够的注意。同时，将回复中与新主题不甚相关的引用内容尽量删除。

对于列表消息，不要直接点击回复（按钮）来开始一个全新的线索，这将限制你的观众。有些邮件阅读程序，比如 mutt，允许用户按线索排序并通过折叠线索来隐藏消息，这样做的人永远看不到你发的消息。

仅仅改变主题还不够。mutt 和其它一些邮件阅读程序还要检查邮件头主题以外的其它信息，以便为其指定线索，所以宁可发一个全新的邮件。

在论坛，因为消息与特定的线索紧密结合，并且通常在线索之外不可见，好的提问方式略有不同，通过回复提问并不要紧。不是所有论坛都允许在回复中出现分离的主题，而且这样做了基本上没有人会去看。不过，通过回复提问本身就是令人怀疑的做法，因为它们只会被正在查看该线索的人读到。所以，除非你只想在该线索当前活跃的人群中提问，还是另起炉灶比较好。

## 使问题容易回复

以 `请向.....回复` 来结束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考虑你的问题更麻烦。如果你的邮件客户端程序不支持这样做，换个好点的；如果是操作系统不支持所有这种邮件客户端程序，也换个好点的。

在论坛，要求通过电子邮件回复是完全无礼的，除非你确信回复的信息也许是敏感的（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线索时得到电子邮件提醒，可以要求论坛发送。几乎所有论坛都支持诸如 `留意本线索`、`有回复发送邮件` 等功能。

## 用清晰、语法、拼写正确的语句书写

经验告诉我们，粗心与草率的作者通常也粗心与草率地思考和编程（我敢打赌）。为这些粗心与草率的思考者回答问题没有什么好处，我们宁可将时间花在其它地方。

清楚、良好地表达你的问题非常重要。如果你觉得这样做麻烦，我们也觉得注意（你的问题）麻烦。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 [注：著名黑客，Linux 内核的重要参与者] 也许可以这样做，但你不行。）

一般而言，如果你写得象个半文盲似的傻子，多半得不到理睬。也不要使用即时通讯中的简写，如将 `you` 简化为 `u` 会使你看起来象一个为了节约二次击键的半文盲式的傻子。更糟的是，如果象个小孩似地鬼画桃符那绝对是在找死，可以肯定没人会理你（或者最多是给你一大堆指责与挖苦）。

如果在非母语论坛提问，你的拼写与语法错误会得到有限的宽容，但懒惰完全不会被容忍（是的，我们通常看得出其中的差别）。同时，除非你知道回复者使用的语言，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。在互联网上英语是工作语言，用英语书写可以将你的问题不被阅读就被直接删除的可能性降到最低。

如果你用英语书写但它是你的第二语言，最好提醒潜在的回复者语言上可能的困难以便绕过这个问题，比如：

- 英语不是我的母语，请谅解拼写错误。
- 如果您使用某某语言，请电邮/私聊我，也许我需要您的协助翻译我的问题。
- 对于这个技术术语本身我很熟悉，但对于它的一些俚语或习惯表达方式就不太明白了。
- 我已经同时用某某语及英语提问，如果您使用两者之一回复，我很乐意翻译。

## 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：

- 使用纯文本而不是 HTML（超文本标注语言）（关闭HTML 并不难）
- 使用 MIME（多用途互联网邮件扩展）附件通常没有问题，前提是真正有内容（譬如附带的源文件或补丁），而不仅仅是邮件客户端程序生成的模板（譬如只是消息内容的拷贝）。
- 不要发送整段只是单行句子但多次折回的邮件（这使得回复部分内容非常困难）。设想你的读者是在80个字符宽的文本终端阅读邮件，设置你的行折回点小于 80 列。
- 但是，也 不要 用任何固定列折回数据（譬如日志文件拷贝或会话记录）。数据应该原样包含，使回复者确信他们看到的是与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。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。
- 永远不要 指望黑客们阅读使用封闭的专用格式编写的文档，诸如微软公司的 Word 或 Excel 文件等。大多数黑客对此的反应就象有人将还在冒热气的猪粪倒在你门口时你的反应一样。即使他们能够处理，也很厌恶这么做。
- 如果你从使用视窗的电脑发送电子邮件，关闭问题颇多的微软“聪明引用”功能（在“工具”->“自动纠正选项”的“输入时自动格式化”下去掉聪明引用的选框），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象个傻笑的小姑娘。这通常不是个好主意，除非你只是对性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客户端程序(如网景公司的 Messenger、微软公司的 Outlook 或者其它类似的)，注意它们的缺省配置不一定满足这些要求。大多数这类程序有基于菜单的 [查看源码](#) 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。

## 描述问题应准确且有内容

- 仔细、清楚地描述问题的症状
- 描述问题发生的环境(主机、操作系统、应用程序，任何相关的)，提供销售商的发行版和版本号（如：“Fedora Core 7”、“Slackware 9.1”等）
- 描述提问前做过的研究及其理解。
- 描述提问前为确定问题而采取的诊断步骤。
- 描述最近对计算机或软件配置的任何相关改变。
- 如果可能，提供在可控环境下重现问题的方法。
- 尽最大努力预测黑客会提到的问题，并提前备好答案。

如果你认为是代码有问题，向黑客提供在可控环境下重现问题的方法尤其重要。当你这么做时，得到有用且及时回复的可能性将大大增加。

[西蒙·泰瑟姆 \(Simon Tatham\)](#) 写过一篇《[如何有效报告臭虫](#)》的文章，我强烈推荐各位阅读。

## 量不在多，精炼则灵

你应该（写得）精炼且有内容，简单地将一大堆代码或数据罗列在求助消息中达不到目的。如果你有一个很大且复杂的测试样例让程序崩溃，尝试将其裁剪得越小越好。

至少有三个理由支持这点。第一，让别人看到你在努力简化问题使你更有可能得到回复。第二，简化问题使你更有可能得到 **有用的** 回复。第三，在提纯臭虫报告的过程中，你可能自己就找到了解决办法或权宜之计。

## 别急于宣称找到臭虫

当你在一个软件中遇到问题，除非你 **非常、非常** 的有根据，不要动辄声称找到了臭虫。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。对于网页和文档也如此，如果你（声称）发现了文档的“臭虫”，你应该能提供相应位置的替代文本。

记住，还有许多其它用户并未经历你遇到的问题，否则你在阅读文档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。如果你声称找到了臭虫，也就质疑了他们的能力，即使你是对的，也有可能使其中的部分人感到不快。（此外，）在主题中嚷嚷“臭虫”也是特别不老练的。

提问时，即使你私下非常确信已经发现一个真正的臭虫，最好写得象是 **你** 做错了什么。如果真的有臭虫，你会在回复中看到这点。这样做的话，如果真有虫子，维护者就会向你道歉，这总比你弄砸了然后欠别人一个道歉要强。

## 低声下气代替不了做自己的家庭作业

有些人明白他们不应该粗鲁或傲慢地行事并要求得到答复，但他们退到相反的低声下气的极端：“我知道我只是个可怜的新丁，一个失败者，但.....”。这既使人困扰，也没有用，当伴随着对实际问题含糊的描述时还特别令人反感。

别用低级灵长类动物的办法浪费你我的时间，相反，尽可能清楚地描述背景情况和你的问题，这比低声下气更好地摆正了你的位置。

有时，论坛设有单独的初学者提问版面，如果你真的认为遇到了肤浅的问题，到那去就是了，但一样别低声下气。

## 描述问题症状而不是猜测

告诉黑客是什么导致了问题是没用的（如果你的诊断理论是了不起的东西，你还会向别人咨询求助吗？）。所以，确保只是告诉他们问题的原始症状，而不是你的解释和理论，让他们来解释和诊断。如果你认为陈述自己的猜测很重要，应清楚地说明这只是你的猜测并描述为什么它们不起作用。

愚蠢：

我在编译内核时接连遇到 SIG11 错误，怀疑主板上的某根电路丝断了，找到它们的最好办法是什么？

明智：

我组装的电脑（K6/233 CPU、FIC-PA2007 主板[威盛 Apollo VP2 芯片组]、Corsair PC133 SDRAM 256Mb 内存）最近在开机 20 分钟左右、做内核编译时频繁地报 SIG11 错，但在头 20 分钟内从不出问题。重新启动不会复位时钟，但整夜关机会。更换所有内存未解决问题，相关的典型编译会话日志附后。

由于以上这点许多人似乎难以掌握，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”（出自国会议员威勒德·D·范迪弗 [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，这并不是怀疑，而只是一种真实而有用的需求，以便让他们看到与你看到的原始



证据尽可能一致的东西，而不是你的猜测与总结。（所以，）让我们看看。

## 按时间先后罗列问题症状

刚出问题之前发生的事情通常包含有解决问题最有效的线索。所以，记录中应准确地描述你、电脑和软件在崩溃前都做了什么。在命令行处理的情况下，有会话日志（如运行脚本工具生成的）并引用相关的若干（如20）行记录会非常有帮助。

如果崩溃的程序有诊断选项（如-v详述开关），试着选择这些能在记录中增加排错信息的选项。记住，“多”不等于“好”。试着选取适当的排错级别以便提供有用的信息而不是将阅读者淹没在垃圾中。

如果你的记录很长（如超过四段），在开头简述问题随后按时间先后罗列详细过程也许更有用。这样，黑客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（而不是报告一个臭虫），在开头就描述你的目标，然后才陈述遇到问题的特定步骤。

经常出现这种情况，寻求技术帮助的人在脑袋里有个更高层次的目标，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有问题，结果要费很大的劲才能通过。

愚蠢：

我怎样才能让某图形程序的颜色拾取器取得十六进制的 RGB 值？

明智：

我正试着用自己选定数值的颜色替换一幅图片的色表，我现在知道的唯一方法是编辑每个表槽，但却无法让某图形程序的颜色拾取器取得十六进制的 RGB 值。

第二种提法是明智的，它使得建议采用更合适的工具以完成任务的回复成为可能。

## 别要求私下回复电邮

黑客们认为问题的解决过程应该公开、透明，此过程中如果更有才能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，作为回复者也因为能力和学识被其它同行看到而得到某种回报。

当你要求私下回复时，此过程和回报都被中止。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，以至于对其它人毫无意义。

对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“向我发电邮，我将为论坛归纳这些回复”将是神奇的句子。试着将邮件列表或新闻组从洪水般雷同的回复中解救出来是非常有礼貌的——但你必须信守诺言。

## 提问应明确

漫无边际的问题通常也被视为没有明确限制的时间无底洞。最有可能给你有用答案的人通常也是最忙的人（假如只是因为他们是承担了太多工作的话），这些人对于没有止境的时间无底洞极其敏感，所以他们也倾向于讨厌那些漫无边际的问题。

如果你明确了想让回复者做的事（如指点方向、发送代码、检查补丁或其它），你更有可能得到有用的回复。（因为）这样可以让他们集中精力并间接地设定了他们为帮助你需要花费的时间和精力上限，这很好。

要想理解专家生活的世界，可以这样设想：那里有丰富的专长资源但稀缺的响应时间。你暗中要求他们奉献的时间越少，你越有可能从这些真正懂行也真正很忙的专家那里得到解答。

所以限定你的问题以使专家回答时需要付出的时间最少——这通常与简化问题还不太一样。举个例，“请问可否指点一下哪有好一点的 X 解释？”通常要比“请解释一下 X”明智。如果你的代码不运行了，通常请别人看看哪有问题比叫他们帮你改正更明智。

## 关于代码的问题

别要求他人给你出问题的代码排错而不提及应该从何入手。张贴几百行的代码，然后说一声“它不能运行”会让你得不到理睬。只贴几十行代码，然后说一句“在第七行以后，本应该显示，但实际出现的是”非常有可能让你得到回复。

最精确描述代码问题的方法是提供一个能展示问题的最小测试样例。什么是最小测试样例？它是对问题的展现，只需要刚好能够重现非预期行为的代码即可。如何生成一个最小测试样例？如果你知道哪一行或哪一段代码会产生问题，将其复制并提供刚好够用的外围支撑代码以构成一个完整的样例（够用是指源码刚好能被编译器、解释器或任何处理它的程序所接受）。如果你不能将问题缩小到特定的段落，复制源码并去除那些与问题无关的代码段。你能提供的最小测试样例越小越好（参见 量不在多，精炼则灵）。

生成一个非常小的最小测试样例并不总是可能，但尽力去做是很好的锻炼，这有可能帮助你找到需要自己解决的问题。即使你找不到，黑客们喜欢看到你努力过，这将使他们更合作。

如果你只是想让别人帮忙审一下代码，在最开头就要说出来，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。

## 别张贴家庭作业式问题

黑客们善于发现“家庭作业”式的问题。我们中的大多数人已经做了自己的家庭作业，那是该你做的，以便从中学到东西。问一下提示没有关系，但不是要求完整的解决方案。

如果你怀疑自己碰到了一个问题，但仍然无法解决，试试在用户组、论坛或（作为最后一招）在项目的“用户”邮件列表或论坛中提问。尽管黑客们会看出来，一些老用户也许仍会给你提示。

## 删除无意义的要求

抵制这种诱惑，即在求助消息末尾加上诸如“有人能帮我吗？”或“有没有答案？”之类在语义上毫无意义的东西。第一，如果问题描述还不完整，这些附加的东西最多也只能是多余的。第二，因为它们是多余的，黑客们会认为这些东西烦人——就很有可能用逻辑上无误但打发人的回复，诸如“是的，你可以得到帮助”和“不，没有给你的帮助”。

一般来说，避免提“是或否”类型的问题，除非你想得到“是或否”类型的回答。

## 不要把问题标记为“紧急”，即使对你而言的确如此

这是你的问题，不要我们的。宣称“紧急”极有可能事与愿违：大多数黑客会直接删除这种消息，他们认为这是无礼和自私地企图得到即时与特殊的关照。而且“紧急”或其它有类似含义的主题有可能触发垃圾过滤规则，潜在的回复者可能永远看不到你的问题！

有一点点局部的例外，如果你是在一些知名度很高、会使黑客们激动的地方使用程序，也许值得这样去做。在这种情况下，如果你有期限压力，也很有礼貌地提到这点，人们也许会有足够的兴趣快一点回答。

当然，这是非常冒险的，因为黑客们对什么是令人激动的标准多半与你的不同。譬如从国际空间站这样张贴没有问题，但代表感觉良好的慈善或政治原因这样做几乎肯定不行。事实上，张贴诸如“紧急：帮我救救这个毛绒绒的小海豹！”肯定会被黑客回避或光火，即使他们认为毛绒绒的小海豹很重要。

如果你觉得这不可思议，再把剩下的内容多读几遍，直到弄懂了再发贴也不迟。

## 礼貌总是有益的

礼貌一点，使用 `请` 和 `谢谢你的关注` 或者 `谢谢你的关照`，让别人明白你感谢他们无偿花时间帮助你。

坦率地讲，这一点没有语法正确、文字清晰、准确、有内容和避免使用专用格式重要（同时也不能替代它们）。黑客们一般宁可读有点唐突但技术鲜明的臭虫报告，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它的）

然尔，如果你已经谈清楚了技术问题，客气一点肯定会增加你得到有用回复的机会。

（我们必须指出，本文唯一受到一些老黑客认真反对的地方是以前曾经推荐过的“提前谢了”，一些黑客认为这隐含着事后不用再感谢任何人的暗示。我们的建议是要么先说 `提前谢了`，事后 再 对回复者表示感谢，要么换种方式表达，譬如用 `谢谢你的关注` 或 `谢谢你的关照`）。

## 问题解决后追加一条简要说明

问题解决后向所有帮助过的人追加一条消息，让他们知道问题是如何解决的并再次感谢。如果问题在邮件列表或新闻组中受到广泛关注，在那里追加此消息比较恰当。

最理想的方式是向最初提问的线索回复此消息，并在主题中包含 `已解决`、`已搞定` 或其它同等含义的明显标记。在人来人往的邮件列表里，一个看见线索 `问题 x` 和 `问题 x-已解决` 的潜在回复者就明白不用再浪费时间了（除非个人觉得“问题 X”有趣），因此可以利用此时间去解决其它问题。

追加的消息用不着太长或太复杂，一句简单的“你好——是网线坏了！谢谢大家——比尔”就比什么都没有要强。事实上，除非解决问题的技术真正高深，一条简短而亲切的总结比长篇大论要好。说明是什么行动解决了问题，用不着重演整个排错的故事。

对于有深度的问题，张贴排错历史的摘要要是恰当的。描述问题的最终状态，说明是什么解决了问题，在此之后 才指明可以避免的弯路。应避免的弯路部分应放在正确的解决方案和其它总结材料之后，而不要将此消息搞成侦探推理小说。列出那些帮助过你的名字，那样你会交到朋友的。

除了有礼貌、有内容以外，这种类型的追帖将帮助其他人在邮件列表、新闻组或论坛文档中搜索到真正解决你问题的方案，从而也让他们受益。

最后，此类追帖还让每位参与协助的人因问题的解决而产生一种满足感。如果你自己不是技术专家或黑客，相信我们，这种感觉对于你寻求帮助的老手和专家是非常重要的。问题叙述到最后不知所终总是令人沮丧的，黑客们痒痒地渴望它们被解决。`挠痒痒` 为你挣到的信誉将对你下次再次张贴提问非常非常的有帮助。

考虑一下怎样才能避免他人将来也遇到类似的问题，问问自己编一份文档或 FAQ 补丁会不会有帮助，如果是的话就将补丁发给维护者。

在黑客中，这种良好的后继行动实际上比传统的礼貌更重要，也是你善待他人而赢得声誉的方式，这是非常有价值的财富。

## 如何解读回答

### “读读该死的手册”（RTFM）和“搜搜该死的网络”（STFW）：如何明白你已完全搞砸

有一个古老而神圣的传统：如果你收到 `读读该死的手册`（RTFM）的回复，发信人认为你应该去“读读该死的手册”。他或她多半是对的，去读一下吧。

“读读该死的手册”（RTFM）有个年轻一点的亲戚，如果你收到“搜搜该死的网络”（STFW）的回复，发信人认为你应该“搜搜该死的网络”。那人多半也是对的，去搜一下吧。（更温和一点的说法是“[谷歌是你的朋友！](#)”）

在论坛，你也可能被要求去搜索论坛的文档。事实上，有人甚至可能热心地为你提供以前解决此问题的线索。但不要依赖这种关照，提问前应该先搜索一下文档。

通常，叫你搜索的人已经打开了能解决你问题的手册或网页，正在一边看一边敲键盘。这些回复意味着他认为：

- 第一，你要的信息很容易找到。
- 第二，自己找要比别人喂到嘴里能学得更多。

你不应该觉得这样就被冒犯了，按黑客的标准，回复者没有不理你就是在向你表示某种尊敬，你反而应该感谢他热切地想帮助你。

## 如果还不明白.....

如果你看不懂回答，不要马上回复一个要求说明的消息，先试试那些最初提问时用过的相同工具（如手册、FAQ、网页、懂行的朋友等）试着搞懂回答。如果还是需要说明，展现你已经明白的。

譬如，假如我告诉你：“看起来象是某输入项有问题，你需要清除它”，接着是个不好的回帖：“什么是某输入项？”。而这是一个很好的跟帖：“是的，我读了手册，某某输入项只在 -z 和 -p 开关中被提到，但都没有涉及到如何清除它们，你指的是哪一个还是我弄错了什么？”

## 对待无礼

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，这种风格对于更关注解决问题而不是使别人感觉舒服而混乱的人是很自然的。

如果你觉得被冒犯了，试着平静地反应。如果有人真的做了过格的事，邮件列表、新闻组或论坛中的前辈多半会招呼他。如果这没有发生而你却光火了，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，你会偶而真的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。然尔，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。如果你是新手或外来者，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阿斯伯格综合症，缺少用于润滑人类社会“正常”交往所需的脑电路。这既可能是真也可能是假。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。我们喜欢现在这个样子，并且一般都对病号标记有站得住脚的怀疑。）

在下一节，我们会谈到另一个问题，当你行为不当时会受到的“冒犯”。

## 别象失败者那样反应

在黑客社区的论坛中有那么几次你可能会搞砸——以本文描述或类似的方式。你会被示众是如何搞砸的，也许言语中还会带点颜色。

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，你该这样做：

熬过去，这很正常。事实上，它是有益健康与恰当的。

社区的标准不会自己维持，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，这不是事情运作的方式。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”。有思路的参与者纷纷离开的结果只会使它们变成了毫无意义的唠叨与无用的技术论坛。

是夸张的“友谊”（以上述方式）还是有用？挑一个。

记着：当黑客说你搞砸了，并且(无论多么刺耳地)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除要容易得多。如果你无法做到感谢，至少要有点尊严，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，就指望别人象对待脆弱的洋娃娃那样对你。

有时候，即使你没有搞砸（或者只是别人想象你搞砸了），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。

这些找茬者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。这些找茬者在给自己找麻烦，这点你不用操心。

也别让自己卷入口水战，大多数口水战最好不要理睬——当然，是在你核实它们只是口水战、没有指出你搞砸的地方，而且没有巧妙地将问题真正的答案藏于其中之后（这也是可能的）。

## 提问禁忌

下面是些典型的愚蠢问题和黑客不回答它们时的想法。

问：我到哪可以找到某程序或 X 资源？

问：我怎样用 X 做 Y？

问：如何配置我的 shell 提示？

问：我可以用 Bass-o-matic 文件转换工具将 AcmeCorp 文档转为 TeX 格式吗？

问：我的{程序、配置、SQL 语句}不运行了

问：我的视窗电脑出问题了，你能帮忙吗？

问：我的程序不运行了，我认为系统工具X有问题

问：我安装 Linux 或 X 遇到困难，你能帮忙吗？

问：我如何才能破解超级用户口令/盗取通道操作员的特权/查看某人的电子邮件？

---

问：

我到哪可以找到某程序或 X 资源？

答：

在我找到它的同样地方，笨旦——在网页搜索引擎上。上帝啊，难道还有人不知道如何使用 谷歌 吗？

问：

我怎样用 X 做 Y？

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，也对要解决的 Y 问题糊涂，还被特定形势禁锢了思维。等他们把问题弄好再说。



问：

如何配置我的 shell 提示？

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“读读该死的手册”（RTFM），然后自己去找出来。

问：

我可以用 Bass-o-matic 文件转换工具将 AcmeCorp 文档转为 TeX 格式吗？

答：

试试就知道了。如果你试过，你既知道了答案，又不用浪费我的时间了。

问：

我的{程序、配置、SQL 语句}不运行了

答：

这不是一个问题，我也没有兴趣去猜你有什么问题——我有更要紧的事要做。看到这种东西，我的反应一般如下：

- 你还有什么补充吗？
- 噢，太糟了，希望你能搞定。
- 这跟我究竟有什么关系？

问：

我的视窗电脑出问题了，你能帮忙吗？

答：

是的，把视窗垃圾删了，装个象 Linux 或 BSD 的开源操作系统吧。

注意：如果程序有官方的视窗版或者与视窗有交互(如 Samba)，你可以问与视窗相关的问题，只是别对问题是由视窗操作系统而不是程序本身造成的回复感到惊讶，因为视窗一般来说太差，这种说法一般都成立。

问：

我的程序不运行了，我认为系统工具 X 有问题

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库文件有明显缺陷的人，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档作后盾。

问：

我安装 Linux 或 X 遇到困难，你能帮忙吗？

答：

不行，我需要亲手操作你的电脑才能帮你排错，去向当地的 Linux 用户组寻求方便的帮助（你可以在 [这里](#) 找到用户组列表）

注意：如果安装问题与某 Linux 发行版有关，在针对 它 的邮件列表、论坛或本地用户组织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：

我如何才能破解超级用户口令/盗取通道操作员的特权/查看某人的电子邮件？

答：

想做这种事情说明你是个卑劣的家伙，想让黑客教你做这种事情说明你是个白痴。

## 好问题与坏问题

最后，我将通过举例来演示提问的智慧。同样的问题两种提法，一种愚蠢，另一种明智。

愚蠢：我在哪能找到关于 Foonly Flurbamatic 设备的东西？

这个问题在乞求得到“搜搜该死的网络”（STFW）式的回复。

明智：我用谷歌搜索过“Foonly Flurbamatic 2600”，但没有找到什么有用的，有谁知道在哪能找到这种设备的编程信息？

这个人已经搜索过网络了，而且听起来他可能真的遇到了问题。

愚蠢：我不能编译某项目的源代码，它为什么这么破？

提问者假设是别人搞砸了，太自大了。

明智：某项目的源代码不能在某 Linux 6.2 版下编译。我读了常见问题文档，但其中没有与某 Linux 相关的内容。这是编译时的记录，我做错了什么吗？

提问者已经指明了运行环境，读了常见问题文档（FAQ），列出了错误，也没有假设问题是别人的过错，这家伙值得注意。

愚蠢：我的主板有问题，谁能帮我？

某黑客对此的反应可能是：“是的，还需要帮你拍背和换尿布吗？”，然后是敲下删除键。

明智：我在 S2464 主板上试过 X、Y 和 Z，当它们都失败后，又试了 A、B 和 C。注意我试 C 时的奇怪症状，显然某某东西正在做某某事情，这不是期望的行为。通常在 Athlon MP 主板上导致某某事情的原因是什么？有谁知道我还能再试点什么以确定问题？

相反地，这个人看来值得回答。他或她展现了解决问题的能力而不是坐等天上掉馅饼。

在最后那个问题中，注意“给我一个回答”与“请帮我看看我还能再做什么测试以得到启发”之间细微但重要的差别。

事实上，最后那个问题基本上源于 2001 年 8 月 Linux 内核邮件列表（lkml）上的真实事件，是我（Eric）当时提了那个问题，我发现 Tyan S2462 主板有神秘的死机现象，邮件列表成员给我提供了解决此问题的关键信息。

通过这种提问方式，我给了别人可以咀嚼玩味的东西。我设法使之对参与者既轻松又有吸引力，也表明了对同行能力的尊敬并邀请他们与我一起协商。通过告诉他们我已经走过的弯路，我还表明了对他们宝贵时间的尊重。

事后，当我感谢大家并评论这次良好的经历时，一个 Linux 内核邮件列表的成员谈到，他认为我得到答案并不是因为我的名字挂在列表上，而只是因为我正确的提问方式。

黑客们在某种方面是非常不留情面的精英分子。我想在这事上他是对的，如果我表现得象个不劳而获的寄生虫，不管我是谁都会被忽略或斥责。他建议将整个事件作为对其它人提问的指导，这直接导致了本文的编写。

## 如果得不到回答

如果得不到回答，请不要认为我们不想帮你，有时只是因为被问到的小组成员的确不知道答案。没有回复不等于不被理睬，当然必须承认从外面很难看出两者的差别。

一般而言，直接将问题再张贴一次不好，这会被视为毫无意义的骚扰。耐心一点，知道你问题答案的人可能生活在不同的时区，有可能正在睡觉，也有可能你的问题一开始就没有组织好。

还有其它资源可以寻求帮助，通常是在一些面向新手的资源中。

有许多在线与本地的用户组织，虽然它们自己不编写任何软件，但是对软件很热心。这些用户组通常因互助和帮助新手而形成。

还有众多大小商业公司提供签约支持服务，别因为要付点钱才有支持就感到沮丧！毕竟，如果你车子的汽缸垫烧了，你多半还得花钱找个修理店把它弄好。即使软件没花你一分钱，你总不能指望服务支持都是免费的。

象 Linux 这样流行的软件，每个开发者至少有一万个以上的用户，一个人不可能应付这么多用户的服务要求。记住，即使你必须付费才能得到支持，也比你还得额外花钱买软件要少得多（而且对封闭源代码软件的服务支持与开源软件相比通常还要贵一点，也要差一点）。

## 如何更好地回答

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儿找 FAQ 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，别因为听起来象个专家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，别妨碍。不要在具体步骤上开玩笑，那样也许会毁了用户的安装——有些可怜的傻瓜会把它当成真的指令。

探索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。试试将很差的问题转变成好问题，别忘了我们都曾是新手。

尽管对那些懒虫报怨一声“读读该死的手册”（RTFM）是正当的，指出文档的位置（即使只是建议做个谷歌关键词搜索）会更好

如果你决意回答，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，应推荐更好的工具，重新组织问题。

请回答真正的问题！如果提问者已经做了自己该做的研究，并且说明尝试过 X, Y, Z, A, B 与 C 都没有得到想要的结果，那么回复 `试试 A 或 B` 或者给出一个内容为 `试一下 X, Y, Z, A, B 或 C` 的链接将极其无益！

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

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

如果需要个人电脑、Unix 和互联网如何工作的基础知识，参阅 [Unix 和互联网工作的基本原理](#)。

当你发布软件或补丁时，试着按 [软件发布实践](#) 操作。

## 鸣谢

伊夫林.米切尔（Evelyn Mitchell）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，米哈伊尔.罗门迪克（Mikhail Ramendik）贡献了一些特别有价值的建议和改进。