



链滴

《提问的智慧》精读注解版

作者: 88250

原文链接: <https://ld246.com/article/1536377163156>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>大学期间，我看过《大教堂与集市》、《提问的智慧》以及影片《操作系统的革命》，它们对我的职业生涯产生了重要的影响。其中《提问的智慧》是篇长文，几十分钟就能看完，此文作者 Eric Steven Raymond (ESR) 也是《大教堂集市》的作者，著名的黑客，开源先驱。他在 2001 年第一次发布《提》后一直对此文进行着更新维护，以保证它尽量适用于最新的情况，比如与时俱进地加入新章节、修复 URL 地址、更新翻译版链接。

<p>今天我又读了一遍《提》，作为一个开源项目的维护者，感慨万千。下文是我对《提》的 部分 注解，如果你时间有限，可以通过该注解版进行有限的了解。但还是强烈建议你精读一《提问的智慧》，英语好的同学可直接看原文 How To Ask Questions The Smart Way。</p>

<p>下面按照原文章节进行注解，标题即原文章节标题。再次说明，我仅仅是注解了《提》中的部分内容，有时间的话请一定要看原文。</p>

<h2 id="声明">声明</h2>

<p>该小节 ESR 建议项目维护者在用户指南文档的显著位置标注：</p>

<blockquote>

<p>本指南不提供此项目的实际支持服务！</p>

<p>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。</p>

</blockquote>

<p>虽然看上去言语有点过激，但其实现实就是这样的。特别是在项目的用户社区里，情况会更糟：一个“伸手党”可以浪费成千上万人的时间。</p>

<h2 id="简介">简介</h2>

<p>读懂该小节需要理解什么是“黑客”，我想能看到我写的文字的人应该都明白黑客是什么。当然如果这也算一个问题的话可按《提》的方法获得答案。</p>

<blockquote>

<p>黑客们喜爱有挑战性的问题，或者能激发他们思维的好问题。……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……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。</p>

</blockquote>

<p>归纳并提出一个好问题黑客们会赏心悦目；提问前不思考、不动手的人会被蔑视。</p>

<blockquote>

<p>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，我们就在降低做自己最擅长的事情上的效率。</p>

</blockquote>

<p>时间对于黑客和用户都是非常宝贵的，用户对技术细节不感兴趣很正常，但是对于没有对软件真产生兴趣或者不主动参与解决问题的人，他们永远也别想得到解答。</p>

<blockquote>

<p>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，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。</p>

</blockquote>

<p>黑客和用户是平等的，黑客们会无视“伸手党”的一切问题，并且默认提问者就是“伸手党”。</p>

<h2 id="在提问之前">在提问之前</h2>

<p>请注意该章节是“在提问之前”：</p>

```
<blockquote>
<ol>
<li>尝试在你准备提问的论坛的旧文章中搜索答案。 </li>
<li>尝试上网搜索以找到答案。 </li>
<li>尝试阅读手册以找到答案。 </li>
<li>尝试阅读常见问题文件 (FAQ) 以找到答案。 </li>
<li>尝试自己检查或试验以找到答案。 </li>
<li>向你身边的强者朋友打听以找到答案。 </li>
<li>如果你是程序开发者, 请尝试阅读源代码以找到答案。 </li>
</ol>
<p>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, 因为我更乐于回答那些表现出能从答案中学习的人的问题。 </p>
</blockquote>
<p>如果你“不假思索”地提问, 即使有人回答你, 那多半也是“脸上笑嘻嘻, 心里 MMP”, 你基得不到你想要的答案; 即使你这次侥幸得到了正确解答, 你肯定还会有更多的问题, 而这些问题基本会有人理你了。 </p>
<h2 id="当你提问时">当你提问时</h2>
<h3 id="慎选提问的论坛">慎选提问的论坛</h3>
<blockquote>
<ul>
<li>在与主题不合的论坛上贴出你的问题。 </li>
<li>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。 </li>
<li>在太多的不同新闻群组上重复转贴同样的问题 (cross-post) 。 </li>
<li>向既非熟人也没有义务解决你问题的人发送私人电邮。 </li>
</ul>
</blockquote>
<p>找对的地方提问; 不确定问题是否会受欢迎就不要提, 因为你的提问对有能力解答你问题的人来本身就是一系列问题: </p>
<ol>
<li>要不要回答</li>
<li>要怎么回答你才能避免你“误入歧途”, 因为有的时候一个答案会引发更多的提问</li>
<li>要怎么回答你才能既帮助到你也帮助到潜在的其他用户</li>
</ol>
<p>回答者考虑的远比你想象的要多, 所以尽量避免提问, 要提就提一个<strong>好问题</strong>。 </p>
<blockquote>
<p>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</p>
</blockquote>
<p>论坛发帖、提 issue、群里问、发邮件、发 IM 选其一。如果你没和维护者面过基, 千万不要把问通过邮箱或者 IM 发给他, 这对他是极大的骚扰和困扰! </p>
<p>对于 B3log 开源的项目而言, 我个人更倾向于解决在<a href="https://ld246.com/forward?goto=https%3A%2F%2Fhacpai.com" target="_blank" rel="nofollow ugc">黑客派</a>上的提问帖这样能帮助到更多人的概率更大。除非你是我的付费客户, 否则请不要发邮件或者 IM 向我提问, 我忽略此类信息。 </p>
<h3 id="Stack-Overflow">Stack Overflow</h3>
<blockquote>
<p>搜索, 然后在 Stack Exchange 问。 </p>
</blockquote>
<p>Stack Exchange 是一个站群, 按大分类分了<a href="https://ld246.com/forward?goto=http%3A%2F%2Fstackexchange.com%2Fsites" target="_blank" rel="nofollow ugc">很多子站点</a>, 其中 Stack Overflow 是编程相关问答的子站。ESR 之所以推荐它, 除了它是世界上目前最有效问答站外, 我觉得很可能还因为它们是非常<strong>开放</strong>的, 所有上面的内容都使用<a h
```

ef="https://ld246.com/forward?goto=https%3A%2F%2Fcreativecommons.org" target="_blank" rel="nofollow ugc">共创协议发布，鼓励带原作链接进行分享演绎，并且可商用。</p><p>除了内容协议非常宽松开放以外，SE 还提供了非常实用的 API，让开发者可以更好地通过他们数据来发展应用生态，这和很多类似的问答站点是截然不同的。就国内的问答站而言，基本都是强调护作者版权、禁止商用转载，甚至是禁止转载，这对作者看似是保护，实则是在破坏内容创作的生态导致恶性循环。</p><h3 id="网站和-IRC-论坛">网站和 IRC 论坛</h3><blockquote><p>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，索引引擎有可能没来得及索引此论坛的全部内容。</p></blockquote><p>搜索之后再搜索，实在搜不到答案了再想其他办法。IRC 我基本没用过，就不展开了，但如果你有 Linux 方面解决不了的问题，可以一试。</p><h3 id="第二步-使用项目邮件列表">第二步，使用项目邮件列表</h3><blockquote><p>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回答你的问题。</p><p>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，那些人多半会将的提问视为干扰他们开发的噪音。</p></blockquote><p>用户邮件列表和开发者邮件列表是两个邮件列表，开发者邮件列表是开发者之间沟通用的，不是外解决问题的地方。ESR 也提到了如果有自信，并且问题在用户邮件列表中没得到解决，那也可以在暗中观察（学习开发者之间的沟通交流方式和文化）后向开发者邮件列表求助。</p><p>自从有了 GitHub 后，邮件列表也用得很少了。尤其是 GitHub 提供了 @Team后，和开发团队交流就变得更有效率了。ESR 对于邮件列表的建议同样适用于 GitHub 上艾特团队也就是要分清楚开发团队和支持团队的区别，或者是分清楚核心团队和 XX 项目团队的区别，作为提者，请不要艾特个人开发者。</p><h3 id="使用有意义且描述明确的标题">使用有意义且描述明确的标题</h3><blockquote><p>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，而应该是在这点空间中使极简单扼要的描述方式来提出问题。</p></blockquote><p>再多的跪求、在线等、标点符号、Emoji 或者特殊符号也解决不了你碰到的问题，反而会被黑客“条件反射式地忽略”，简明扼要的标题是一个好问题的开始。</p></blockquote><p>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。</p><p><code>X.org</code> 6.8.1 的鼠标光标，在某牌显卡 MV1005 芯片组环境下 - 会变形。</p></blockquote><p>另外，这一小节对在回复中提出新问题也给出了建议：无论是邮件列表还是网页论坛，尽量不要回复中提出新问题，新问题就新发邮件或者新开帖子。</p><h3 id="使问题容易回复">使问题容易回复</h3><blockquote><p>以“请将你的回复发送到.....”来结束你的问题多半会使你得不到回答。
在论坛，要求通过电子邮件回复是非常无礼的，除非你认为回复的信息可能比较敏感。</p></blockquote><p>这点很好理解，用什么渠道发布的提问，回答者就回复到该渠道上。除了可能涉及敏感信息外，问者这样的请求是无理并且无礼的。</p><h3 id="用清晰-正确-精准并语法正确的语句">用清晰、正确、精准并语法正确的语句</h3>

<blockquote>

<p>正确的拼写、标点符号和大小写是很重要的。通常来说，如果你觉得这样做很麻烦，不想在乎这，那我们也觉得麻烦，不想在乎你的提问。</p>

</blockquote>

<p>别以为别人不在乎拼写、标点符号和大小写，黑客们甚至在乎空格的使用！比如中文和英文、数、符号之间必须要有空格，这个空格被称作“盘古之白”/a>。</p>

<p>关于大小写引发的“悲剧”也不少，甚至可能就发生在你身上。比如你简历是不是写过“iphone/os 开发”，“熟练使用 JAVA”，“熟悉 Mysql”？这样“不拘小节”的人恐怕不知道大多数编程语是区分大小写的吧....</p>

<blockquote>

<p>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。</p>

</blockquote>

<p>B3log 开源项目提问无论是在 GitHub 上还是在黑客派上，均可优先使用中文。</p>

<h3 id="使用易于读取且标准的文件格式发送问题">使用易于读取且标准的文件格式发送问题</h3>

<blockquote>

对一些特殊的文件不要设置固定宽度（譬如日志档案拷贝或会话记录）。数据应该原样包含，让回复者有信心他们看到的是和你看到的一样的东西。

</blockquote>

<p>帖日志要保证是纯文本格式，特别是服务器端日志，千万不要截图（特别是那种截图还“勾画重”的行为应该立刻停止）；对于浏览器端的监控调试输出可以截图，但截图时要考虑信息是否完整，不要包含无用画面。</p>

<blockquote>

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，除非你只是对性而不是对答案感兴趣。

</blockquote>

<p>可能你碰到问题后心情很复杂，需要大量表情符号、红绿色、超大标题或者 gif 动图来表达你心的困惑，但是这样做除了表现出幼稚、干扰阅读外没有任何用。</p>

<p>这一小节中 ESR 还建议了很多条关于发送邮件时的细节，现在邮件在问答时用得比较少了，所略过。</p>

<h3 id="精确地描述问题并言之有物">精确地描述问题并言之有物</h3>

<blockquote>

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可控环境的方法。

</blockquote>

<p>项目维护者通常会提供问题报告模板来引导用户报告问题（比如这里），如果你在报告问题时没有看到模板，那就照上述建议进行问题描述。</p>

<h3 id="话不在多而在精">话不在多而在精</h3>

<blockquote>

<p>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。</p>

<p>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。</p>

</blockquote>

<p>动手简化问题是非常重要的，也许你在简化过程中就可以获得解决方案。</p>

<h3 id="别动辄声称找到-Bug">别动辄声称找到 Bug</h3>

<blockquote>

<p>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，也有可能冒犯到其中某部分人。当你在标题中嚷嚷着有 <code>Bug</code>，这尤其严重。</p>

</blockquote>

<p>不要自以为是的认为你真的发现了 <code>Bug</code>，更不要直接说出来。大家都是为了件的将来更好，礼貌明智的表达更能有效促进软件发展。</p>

<blockquote>

<p>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 Bug，你会在回复中看到这点。这样做的话，如果有 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。</p>

</blockquote>

<p>即使你真的确定是 Bug 了，也不要直接说出来，维护者会明白你的意思的，并且向你道歉。不认为维护者是玻璃心，其实是他们只对他们认为有价值的问题上心。</p>

<h3 id="低声下气不能代替你的功课">低声下气不能代替你的功课</h3>

<blockquote>

<p>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，一个撻瑟，但...”这既使人困扰，也没有用，尤其是伴随着与实际题含糊不清的描述时更令人反感。</p>

</blockquote>

<p>时刻记住大家的时间都很宝贵，描述问题是关键。</p>

<h3 id="描述问题症状而非你的猜测">描述问题症状而非你的猜测</h3>

<blockquote>

<p>告诉黑客们你认为问题是怎样造成的并没有什么帮助。（如果你的推断如此有效，还用向别人求助？），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。</p>

</blockquote>

<p>往往简单的问题早都被解决了，你碰到的通常都是复杂的问题。复杂的问题可能有 100 种产生可能性，你说的只是一种可能性，而且很可能是错误的，即使你提供了理论解释。</p>

<h3 id="按发生时间先后列出问题症状">按发生时间先后列出问题症状</h3>

<blockquote>

<p>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操作步骤，以及机器和软件的反应，直到问题发生。</p>

</blockquote>

<p>描述问题发生的步骤至关重要，如果你按照你给的步骤不能重现问题，那就不要提问。另外，可已尝试手动调整日志级别来观察输出情况。</p>

<blockquote>

<p>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。</p>

</blockquote>

<h3 id="描述目标而不是过程">描述目标而不是过程</h3>

<blockquote>

<p>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，然后才陈述重现所卡住的特定步骤。</p>

</blockquote>

<p>你别笑，我真遇到过这样报告问题的：“我发现了个 bug，你的软件不能做 XXX。”我回复：这不是 bug，是个 feature。”并关闭。</p>

<p>软件所有的功能通常都在界面或者参数上提供出来，文档上也会有相应描述。如果某个功能你找到入口，只有两种情况：</p>

这个功能是个彩蛋

没这功能

<h3 id="别要求使用私人电邮回复">别要求使用私人电邮回复</h3>

<blockquote>

<p>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处，最初的回复才能够、也应该被纠正。同时，作为提供帮助者可以得到一些奖励，奖励就是他的能和学识被其他同行看到。</p>

</blockquote>

<p>秉承开源的理念：开放、共享、协作来进行提问和解决问题很有效也很有价值。</p>

<p>对于 B3log 开源项目而言，优先到论坛发帖提问，请勿私信我，私信我的话我只会回你：“请论坛发问答帖。”或者直接忽略你的提问。</p>

<h3 id="清楚明确的表达你的问题以及需求">清楚明确的表达你的问题以及需求</h3>

<blockquote>

<p>漫无边际的提问是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（们忙是因为要亲自完成大部分工作）。..... 因此，问“我想更好的理解 X，可否指点一下哪有好一点明？”通常比问“你能解释一下 X 吗？”更好。</p>

</blockquote>

<p>如果你对可能的答案的边界不清楚，最好不要问。</p>

<h3 id="询问有关代码的问题时">询问有关代码的问题时</h3>

<p>这一节 ESR 建议在反馈代码问题时可以通过最小化测试用例（前面“话不在多而在精”一节有到过）进行描述。</p>

<blockquote>

<p>一般而言，要得到一段相当精简的测试用例并不太容易，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。</p>

</blockquote>

<p>如果是 code review：</p>

<blockquote>

<p>一定要提到你认为哪一部分特别需要关注以及为什么。</p>

</blockquote>

<h3 id="别把自己家庭作业的问题贴上来">别把自己家庭作业的问题贴上来</h3>

<blockquote>

<p>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。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。</p>

</blockquote>

<p>这里“家庭作业式的问题”主要就是指那种 显然 是要通过自己学习来解决的问题并且通过学习肯定能解决的问题，总之，你学了就知道了。</p>

<h3 id="去掉无意义的提问句">去掉无意义的提问句</h3>

<blockquote>

<p>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？”。</p>

</blockquote>

<p>这和问你“在吗？”一样，大部分时候我都很想回复“不在。”你要知道，在一些代码实现中，们都是尽自己的最大努力来做简化，看到你这样问，我们会条件反射地把它优化掉。</p>

<h3 id="即使你很急也不要再在标题写">即使你很急也不要再在标题写 <code>紧急</code></h3>

<blockquote>

<p>这是你的问题，不是我们的。..... 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

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，人们也许有兴趣回答快一点。

过分强调往往适得其反。

礼多人不怪-而且有时还很有帮助

彬彬有礼，多用 `请` 和 `感谢您的关注`，或 `谢谢你的照`。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、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）。

礼多人不怪，但别颠倒主次，精准描述问题永远是第一位。

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。..... 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赢得声誉的方式，这是非常有价值的资产。

问题即使是自己解决的，也请在解决后补充下简短的说明，这对于积攒声誉很有帮助。其他遇到同样问题的人会感激你的，黑客们也会对你有始有终的行动表示赞赏，即使这个问题对他们来说不是个问题。

如何解读答案

RTFM-和-STFW-如何知道你已完全搞砸了

如果你收到 RTFM（Read The Fucking Manual）的回应，回答者认为你 `应该去读他的手册`。当然，基本上他是对的，你应该去读一读。..... 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`。那人多半也是对的，搜索一下吧。..... 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。**

鸟哥语录.jpg。

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了点什么。

只要你有搞不懂的东西，永远优先尝试自己解决。

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。..... 另一方面，你偶尔真的会碰到无和无聊的言行。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受的。然而，在行事之前一定要非常非常的有根据。

这一点让我想起了曾仕强的“只讲妥当话、不讲实在话”。至少在黑客的圈子里，这样说“妥当”完全就是在浪费大家时间。

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。

做个键盘侠实在是太容易了。

如何避免扮演失败者

如果你搞砸了，并且被人在公开场合告知你的愚蠢行为时：

<p>熬过去，这很正常。..... 当有人评论你的一个说法有误或者提出不同看法时，坚持声称受到个人击也毫无益处，这些都是失败者的态度。..... 夸张的讲法是：你要的是友善（上述方式）还是有用？两个里面挑一个。</p>

</blockquote>

<p>这一节 ESR 建议大家不要卷入毫无意义的口水战。有些“自以为是专家的不中用家伙”他们其是“测试你是否真会搞砸的心理专家”，这些人就靠引战和表演来满足他们的内心。</p>

<blockquote>

<p>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，这你不用操心。</p>

</blockquote>

<h2 id="不该问的问题">不该问的问题</h2>

<p>这一节是一些常见的不该发问的示例，建议看原文。</p>

<h2 id="好问题与蠢问题">好问题与蠢问题</h2>

<p>这一节是一些常见的好问题和蠢问题示例，建议看原文。</p>

<blockquote>

<p>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，如果我像个乞讨者样提问，不论我是谁，一定会惹恼某些人或者被他们忽视。</p>

</blockquote>

<h2 id="如果得不到回答">如果得不到回答</h2>

<blockquote>

<p>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。..... 另外，你可以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！..... 就算软件没费你一分钱，你也不能强求技术支持总是免费的。</p>

</blockquote>

<p>总之，正确的提问但没有得到回答时不要沮丧，因为这两者之间没有必然的因果关系。付费咨询是解决问题的一条路径，就像你选择使用免费软件一样。</p>

<h2 id="如何更好地回答问题">如何更好地回答问题</h2>

<p>这一节是对回答者的建议，保持友善、归纳问答、提高效率，授人以鱼不如授人以渔。</p>

<h2 id="相关资源">相关资源</h2>

<blockquote>

<p>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。</p>

<p>当你发布软件或补丁时，试着按软件发布实践操作。</p>

</blockquote>

<h2 id="鸣谢">鸣谢</h2>

<blockquote>

<p>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 R mendik 贡献了一些特别有价值的建议和改进。</p>

</blockquote>