Table of Contents

Introduction 1.1

提問的智慧

How To Ask Questions The Smart Way

原文網址:http://www.catb.org/~esr/faqs/smart-questions.html

本中文指南是基於原文 3.10 版以及 2010 年由 Gasolin 所翻譯版本的最新翻譯；

協助指出翻譯問題，請發**Issue**，或直接發**Pull Request**給我。

本文另有: 简体中文版

原文版本歷史

目錄

- 聲明
- 簡介
- 在提問之前
- 當你提問時
  - 慎選提問的論壇
  - Stack Overflow
  - 網站和IRC論壇
  - 第二步，使用專案郵件列表
  - 使用有意義且描述明確的標題
  - 使問題容易回覆
  - 用清晰、正確、精準並合法文法的語句
  - 使用易於讀取且標準的文件格式發送問題
  - 精確的描述問題並言之有物
  - 話不在多而在精
  - 別動輒聲稱找到Bug
  - 可以低聲下氣，但還是要先做功課
  - 描述問題症狀而非猜測

    - 按發生時間先後列出問題症狀
    - 描述目標而不是過程
    - 別要求使用私人電郵回覆
    - 清楚明確的表達你的問題以及需求
    - 詢問有關程式碼的問題時
    - 別把自己家庭作業的問題貼上來
    - 去掉無意義的提問句
    - 即使你很急也不要在標題寫緊急
    - 禮多人不怪，而且有時還很有幫助
    - 問題解決後，加個簡短的補充說明
- 如何解讀答案
    - RTFM和STFW：如何知道你已完全搞砸了
    - 如果還是搞不懂
    - 處理無禮的回應
- 如何避免扮演失敗者
- 不該問的問題
- 好問題與蠢問題
- 如果得不到回答
- 如何更好地回答問題
- 相關資源
- 鳴謝

聲明

許多專案在他們的使用協助/說明網頁中連結了本指南，這麼做很好，我們也鼓勵大家都這麼做。但如果你是負責管理這個專案網頁的人，請在超連結附近的顯著位置上註明：

本指南不提供此專案的實際支援服務！

我們已經深刻領教到少了上述聲明所帶來的痛苦。因爲少了這點聲明，我們不停地被一些白痴糾纏。這些白痴認爲既然我們發布了這本指南，那麼我們就有責任解決世上所有的技術問題。

如果你是因爲需要某些協助而正在閱讀這本指南，並且最後離開是因爲發現從本指南作者們身上得不到直接的協助，那麼你就是我們所說的那些白痴之一。別問我們問題，我們只會忽略你。我們在這本指南中是教你如何從那些眞正懂得你所遇到軟體或硬體問題的人取得協助，而99%的情況下那不會是我們。除非你確定本指南的作者之一剛好是你所遇到的問題領域的專家，否則請不要打擾我們，這樣大家都會開心一點。

簡介

在黑客的世界裡，當你拋出一個技術問題時，最終是否能得到有用的回答，往往取決於你所提問和追問的方式。本指南將教你如何正確的提問以獲得你滿意的答案。

不只是黑客，現在開放原始碼（Open Source）軟體已經相當盛行，你常常也可以由其他有經驗的使用者身上得到好答案，這是件好事；使用者比起黑客來，往往對那些新手常遇到的問題更寬容一些。然而，將有經驗的使用者視爲黑客，並採用本指南所提的方法與他們溝通，同樣也是能從他們身上得到滿意回答的最有效方式。

首先你應該明白，黑客們喜愛有挑戰性的問題，或者能激發他們思維的好問題。如果我們並非如此，那我們也不會成爲你想詢問的對象。如果你給了我們一個值得反覆咀嚼玩味的好問題，我們自會對你感激不盡。好問題是激勵，是厚禮。好問題可以提高我們的理解力，而且通常會暴露我們以前從沒意識到或者思考過的問題。對黑客而言，"好問題！"是誠摯的大力稱讚。

儘管如此，黑客們有著蔑視或傲慢面對簡單問題的壞名聲，這有時讓我們看起來對新手、無知者似乎較有敵意，但其實不是那樣的。

我們不諱言我們對那些不願思考、或者在發問前不做他們該做的事的人的蔑視。那些人是時間殺手 — 他們只想索取，從不付出，消耗我們可用在更有趣的問題或更值得回答的人身上的時間。我們稱這樣的人爲 `失敗者（魯蛇）` （由於歷史原因，我們有時把它拼作 `lusers` ）。

我們意識到許多人只是想使用我們寫的軟體，他們對學習技術細節沒有興趣。對大多數人而言，電腦只是種工具，是種達到目的的手段而已。他們有自己的生活並且有更要緊的事要做。我們了解這點，也從不指望每個人都對這些讓我們著迷的技術問題感興趣。儘管如此，我們回答問題的風格是指向那些眞正對此有興趣並願意主動參與解決問題的人，這一點不會變，也不該變。如果連這都變了，我們就是在降低做自己最擅長的事情上的效率。

我們（在很大程度上）是自願的，從繁忙的生活中抽出時間來解答疑惑，而且時常被提問淹沒。所以我們無情的濾掉一些話題，特別是拋棄那些看起來像失敗者的傢伙，以便更高效的利用時間來回答 `贏家（溫拿）` 的問題。

如果你厭惡我們的態度，高高在上，或過於傲慢，不妨也設身處地想想。我們並沒有要求你向我們屈服 -- 事實上，我們大多數人非常樂意與你平等地交流，只要你付出小小努力來滿足基本要求，我們就會歡迎你加入我們的文化。但讓我們幫助那些不願意幫助自己的人是沒有效率的。無知沒有關係，但裝白痴就是不行。

所以，你不必在技術上很在行才能吸引我們的注意，但你必須表現出能引導你變得在行的特質 -- 機敏、有想法、善於觀察、樂於主動參與解決問題。如果你做不到這些使你與眾不同的事情，我們建議你花點錢找家商業公司簽個技術支援服務合同，而不是要求黑客個人無償地幫助你。

如果你決定向我們求助，當然你也不希望被視爲失敗者，更不願成爲失敗者中的一員。能立刻得到快速並有效答案的最好方法，就是像贏家那樣提問 -- 聰明、自信、有解決問題的思路，只是偶爾在特定的問題上需要獲得一點幫助。

（歡迎對本指南提出改進意見。你可以 email 你的建議至 esr@thyrsus.com 或 respond-auto@linuxmafia.com。然而請注意，本文並非網路禮節的通用指南，而我們通常會拒絕無助於在技術論壇得到有用答案的建議。）

在提問之前

在你準備要通過電子郵件、新聞群組或者聊天室提出技術問題前，請先做到以下事情：

1. 嘗試在你準備提問的論壇的舊文章中搜尋答案。
2. 嘗試上網搜尋來找到答案。
3. 嘗試閱讀手冊來找到答案。
4. 嘗試閱讀常見問題文件（FAQ）來找到答案。
5. 嘗試自己檢查或試驗來找到答案
6. 向你身邊的強者朋友打聽來找到答案。
7. 如果你是程式開發者，請嘗試閱讀原始碼來找到答案

當你提出問題的時候，請先表明你已經做了上述的努力；這將有助於樹立你並不是一個不勞而獲且浪費別人的時間的提問者。如果你能一併表達在做了上述努力的過程中所學到的東西會更好，因爲我們更樂於回答那些表現出能從答案中學習的人的問題。

運用某些策略，比如先用 Google 搜尋你所遇到的各種錯誤訊息（既搜尋 Google論壇，也搜尋網頁），這樣很可能直接就找到了能解決問題的文件或郵件列表線索。即使沒有結果，在郵件列表或新聞組尋求幫助時加上一句 `我在 Google 中搜過下列句子但沒有找到什麼有用的東西` 也是件好事，即使它只是表明了搜尋引擎不能提供哪些幫助。這麼做（加上搜尋過的字串）也讓遇到相似問題的其他人能被搜尋引擎引導到你的提問來。

別著急，不要指望幾秒鐘的 Google 搜尋就能解決一個複雜的問題。在向專家求助之前，再閱讀一下常見問題文件（FAQ）、放輕鬆、坐舒服一些，再花點時間思考一下這個問題。相信我們，他們能從你的提問看出你做了多少閱讀與思考，如果你是有備而來，將更有可能得到解答。不要將所有問題一股腦拋出，只因你的第一次搜尋沒有找到答案（或者找到太多答案）。

準備好你的問題，再將問題仔細的思考過一遍，因爲草率的發問只能得到草率的回答，或者根本得不到任何答案。越是能表現出在尋求幫助前你爲解決問題所付出的努力，你越有可能得到實質性的幫助。

小心別問錯了問題。如果你的問題基於錯誤的假設，某個普通黑客（J. Random Hacker）多半會一邊在心裏想著 `蠢問題…`，一邊用無意義的字面解釋來答覆你，希望著你會從問題的回答（而非你想得到的答案）中汲取教訓。

絕不要自以爲夠格得到答案，你沒有；你並沒有。畢竟你沒有爲這種服務支付任何報酬。你將會是自己去掙到一個答案，靠提出有內涵的、有趣的、有思維激勵作用的問題 -- 一個有潛力能貢獻社群經驗的問題，而不僅僅是被動的從他人處索取知識。

另一方面，表明你願意在找答案的過程中做點什麼是一個非常好的開始。`誰能給點提示？`、`我的這個例子裏缺了什麼？`以及`我應該檢查什麼地方`比`請把我需要的確切的過程貼出來`更容易得到答覆。因爲你表現出只要有人能指出一個正確方向，你就有完成它的能力和決心。

當你提問時

慎選提問的論壇

小心選擇你要提問的場合。如果你做了下述的事情，你很可能被忽略掉或者被看作失敗者：

- 在與主題不合的論壇上貼出你的問題
- 在探討進階技術問題的論壇張貼非常初級的問題；反之亦然
- 在太多的不同新聞群組上重複轉貼同樣的問題（cross-post）
- 向既非熟人也沒有義務解決你問題的人發送私人電郵

黑客會剔除掉那些搞錯場合的問題，以保護他們溝通的管道不被無關的東西淹沒。你不會想讓這種事發生在自己身上的。

因此，第一步是找到對的論壇。再說一次，Google 和其它搜尋引擎還是你最好的朋友，用它們來找到與你遭遇到困難的軟硬體問題最相關的網站。通常在那裡都有常見問題（FAQ）、郵件列表及相關說明文件的連結。如果你的努力（包括閱讀 FAQ）都沒有結果，網站上也許還有回報臭蟲（Bug-reporting）的流程或連結，如果是這樣，連過去看看。

向陌生的人或論壇發送郵件最可能是風險最大的事情。舉例來說，別假設一個提供豐富內容的網頁的作者會想充當你的免費顧問。不要對你的問題是否會受到歡迎做太樂觀的估計 -- 如果你不確定，那就向別處發送，或者根本別發。

在選擇論壇、新聞群組或郵件列表時，別太相信名字，先看看 FAQ 或者許可書以弄清楚你的問題是否切題。發文前先翻翻已有的話題，這樣可以讓你感受一下那裡的文化。事實上，事先在新聞組或郵件列表的歷史記錄中搜尋與你問題相關的關鍵詞是個極好的主意，也許這樣就找到答案了。即使沒有，也能幫助你歸納出更好的問題。

別像機關槍似的一次"掃射"所有的幫助管道，這就像大喊大叫一樣會使人不愉快。要一個一個地來。

搞清楚你的主題！最典型的錯誤之一是在某種致力於跨平台可移植的語言、套件或工具的論壇中提關於 Unix 或 Windows 作業系統程序介面的問題。如果你不明白爲什麼這是大錯，最好在搞清楚這之間差異之前什麼也別問。

一般來說，在仔細挑選的公共論壇中提問，會比在私有論壇中提同樣的問題更容易得到有用的回答。有幾個理由可以支持這點，一是看潛在的回覆者有多少，二是看觀眾有多少。黑客較願意回答那些能幫助到許多人的問題。

可以理解的是，老練的黑客和一些熱門軟體的作者正在接受過多的錯發訊息。就像那根最後壓垮駱駝背的稻草一樣，你的加入也有可能使情況走向極端 -- 已經好幾次了，一些熱門軟體的作者從自己軟體的支援中抽身出來，因爲伴隨而來湧入其私人郵箱的無用郵件變得無法忍受。

Stack Overflow

搜尋，然後 才在 Stack Exchange 發問。

近年來，Stack Exchange community 社群已經成爲回答技術及其他問題的主要管道，尤其是那些開放源碼的專案。

因爲 Google 索引是即時的，在看 Stack Exchange 之前先在 Google 搜尋。有很高的機率某人已經問了一個類似的問題，而且 Stack Exchange 網站們往往會是搜尋結果中最前面幾個。如果你在 Google 上沒有找到任何答案，你再到特定相關主題的網站去找。用標籤（Tag）搜尋能讓你更縮小你的搜尋結果。

Stack Exchange 已經成長到超過一百個網站，以下是最常用的幾個站：

- Super User 是問一些通用的電腦問題，如果你的問題跟程式碼或是寫程式無關，只是一些網路連線之類的，請到這裡。
- Stack Overflow 是問寫程式有關的問題。
- Server Fault 是問伺服器和網管相關的問題。

網站和IRC論壇

在地的使用者群組（user group），或者你所用的 Linux 發行版本也許正在宣傳他們的網頁論壇或 IRC 頻道，並提供新手幫助（在一些非英語國家，新手論壇很可能還是郵件列表），這些地方是開始提問的好首選，特別是當你覺得遇到的也許只是相對簡單或者很普通的問題時。有廣告贊助的 IRC 頻道是公開歡迎提問的地方，通常可以即時得到回應。

事實上，如果程式出的問題只發生在特定 Linux 發行版提供的版本（這很常見），最好先去該發行版的論壇或郵件列表中提問，再到程式本身的論壇或郵件列表提問。（否則）該項目的黑客可能僅僅回覆 "用我們的版本"。

在任何論壇發文以前，先確認一下有沒有搜尋功能。如果有，就試著搜尋一下問題的幾個關鍵詞，也許這會有幫助。如果在此之前你已做過通用的網頁搜尋（你也該這樣做），還是再搜尋一下論壇，搜尋引擎有可能沒來得及索引此論壇的全部內容。

通過論壇或 IRC 頻道來提供使用者支援服務有增長的趨勢，電子郵件則大多爲專案開發者間的交流而保留。所以最好先在論壇或 IRC 中尋求與該專案相關的協助。

第二步，使用專案郵件列表

當某個專案提供開發者郵件列表時，要向列表而不是其中的個別成員提問，即使你確信他能最好地回答你的問題。查一查專案的文件和首頁，找到專案的郵件列表並使用它。有幾個很好的理由支持我們採用這種辦法：

- 任何好到需要向個別開發者提出的問題，也將對整個專案群組有益。反之，如果你認爲自己的問題對整個專案群組來說太愚蠢，也不能成爲騷擾個別開發者的理由。
- 向列表提問可以分散開發者的負擔，個別開發者（尤其是專案領導人）也許太忙以至於沒法回答你的問題。
- 大多數郵件列表都會被封存，那些被封存的內容將被搜尋引擎索引。如果你向列表提問並得到解答，將來其它人可以通過網頁搜尋找到你的問題和答案，也就不用再次發問了。
- 如果某些問題經常被問到，開發者可以利用此資訊來改進說明文件或軟體本身，以使其更清楚。如果只是私下提問，就沒有人能看到最常見問題的完整場景。

如果一個項目既有"使用者" 也有"開發者"（或"黑客"）郵件列表或論壇，而你又不會動到那些原始碼，那麼就向"使用者"列表或論壇提問。不要假設自己會在開發者列表中受到歡迎，那些人多半會將你的提問視爲干擾他們開發的噪音。

然而，如果你確信你的問題很特別，而且在"使用者" 列表或論壇中幾天都沒有回覆，可以試試前往"開發者"列表或論壇發問。建議你在張貼前最好先暗地裡觀察幾天以了解那裡的行事方式（事實上這是參與任何私有或半私有列表的好主意）

如果你找不到一個專案的郵件列表，而只能查到專案維護者的電子郵件地址，儘管向他發信。即使是在這種情況下，也別假設（專案）郵件列表不存在。在你的電子郵件中，請陳述你已經試過但沒有找到合適的郵件列表，也提及你不反對將自己的郵件轉發給他人（許多人認爲，即使沒什麼秘密，私人電子郵件也不應該被公開。通過允許將你的電子郵件轉發他人，你給了相應人員處置你郵件的選擇）。

使用有意義且描述明確的標題

在郵件列表、新聞群組或論壇中，大約50字以內的標題是抓住資深專家注意力的好機會。別用喋喋不休的 `幫幫忙` 、 `跪求` 、 `急` （更別說 `救命啊！！！！` 這樣讓人反感的話，用這種標題會被條件反射式地忽略）來浪費這個機會。不要妄想用你的痛苦程度來打動我們，而是在這點空間中使用極簡單扼要的描述方式來提出問題。

一個好標題範例是 `目標 -- 差異` 式的描述，許多技術支持組織就是這樣做的。在 `目標` 部分指出是哪一個或哪一組東西有問題，在 `差異` 部分則描述與期望的行爲不一致的地方。

> 蠢問題：救命啊！我的筆電不能正常顯示了！
>
> 聰明問題：X.org 6.8.1的滑鼠游標會變形，某牌顯示卡 MV1005 晶片組。
>
> 更聰明問題：X.org 6.8.1的滑鼠游標，在某牌顯示卡 MV1005 晶片組環境下 - 會變形。

編寫 `目標 -- 差異` 式描述的過程有助於你組織對問題的細緻思考。是什麼被影響了？ 僅僅是滑鼠游標或者還有其它圖形？只在 X.org 的 X 版中出現？或只是出現在6.8.1版中？ 是針對某牌顯示卡晶片組？或者只是其中的 MV1005 型號？ 一個黑客只需瞄一眼就能夠立即明白你的環境和你遇到的問題。

總而言之，請想像一下你正在一個只顯示標題的封存討論串（Thread）索引中查尋。讓你的標題更好地反映問題，可使下一個搜尋類似問題的人能夠關注這個討論串，而不用再次提問相同的問題。

如果你想在回覆中提出問題，記得要修改內容標題，以表明你是在問一個問題， 一個看起來像 `Re: 測試` 或者 `Re: 新bug` 的標題很難引起足夠重視。另外，在不影響連貫性之下，適當引用並刪減前文的內容，能給新來的讀者留下線索。

對於討論串，不要直接點擊回覆來開始一個全新的討論串，這將限縮你的觀眾。因爲有些郵件閱讀程序，比如 mutt ，允許使用者按討論串排序並通過折疊討論串來隱藏消息，這樣做的人永遠看不到你發的消息。

僅僅改變標題還不夠。mutt 和其它一些郵件閱讀程式還會檢查郵件標題以外的其它信息，以便爲其指定討論串。所以寧可發一個全新的郵件。

在網頁論壇上，好的提問方式稍有不同，因爲討論串與特定的訊息緊密結合，並且通常在討論串外就看不到裡面的內容，故通過回覆提問，而非改變標題是可接受的。不是所有論壇都允許在回覆中出現分離的標題，而且這樣做了基本上沒有人會去看。不過，通過回覆提問，這本身就是曖昧的做法，因爲它們只會被正在查看該標題的人讀到。所以，除非你只想在該討論串當前活躍的人群中提問，不然還是另起爐灶比較好。

使問題容易回覆

以 `請將你的回覆寄到……` 來結束你的問題多半會使你得不到回答。如果你覺得花幾秒鐘在郵件客户端設置一下回覆地址都麻煩，我們也覺得花幾秒鐘思考你的問題更麻煩。如果你的郵件程式不支持這樣做，換個好點的；如果是作業系統不支持這種郵件程式，也換個好點的。

在論壇，要求通過電子郵件回覆是非常無禮的，除非你相信回覆的信息可能比較敏感（而且有人會爲了某些未知的原因，只讓你而不是整個論壇知道答案）。如果你只是想在有人回覆討論串時得到電子郵件提醒，可以要求網頁論壇發送給你。幾乎所有論壇都支持諸如 `追蹤此討論串` 、 `有回覆時發送郵件提醒` 等功能。

用清晰、正確、精準並合法文法的語句

我們從經驗中發現，粗心的提問者通常也會粗心的寫程式與思考（我敢打包票）。回答粗心大意者的問題很不值得，我們寧願把時間耗在別處。

正確的拼字、標點符號和大小寫是很重要的。一般來說，如果你覺得這樣做很麻煩，不想在乎這些，那我們也覺得麻煩，不想在乎你的提問。花點額外的精力斟酌一下字句，用不著太僵硬與正式 -- 事實上，黑客文化很看重能準確地使用非正式、俚語和幽默的語句。但它必須很準確，而且有跡象表明你是在思考和關注問題。

正確地拼寫、使用標點和大小寫，不要將 `its` 混淆爲 `it's` ， `loose` 搞成 `lose` 或者將 `discrete` 弄成 `discreet` 。不要全部用大寫，這會被視爲無禮的大聲嚷嚷（全部小寫也好不到哪去，因爲不易閱讀。Alan Cox也許可以這樣做，但你不行。）

更白話的說，如果你寫得像是個半文盲[譯註：小白]），那多半得不到理睬。也不要使用即時通訊中的簡寫或火星文，如將 `的` 簡化爲 `勺` 會使你看起來像一個爲了少打幾個鍵而省字的小白。更糟的是，如果像個小孩似地鬼畫符那絕對是在找死，可以肯定沒人會理你（或者最多是給你一大堆指責與挖苦）。

如果在使用非母語的論壇提問，你可以犯點拼寫和語法上的小錯，但決不能在思考上馬虎（沒錯，我們通常能弄清兩者的分別）。同時，除非你知道回覆者使用的語言，否則請使用英語書寫。繁忙的黑客一般會直接刪除用他們看不懂語言寫的消息。在網路上英語是通用語言，用英語書寫可以將你的問題在尚未被閱讀就被直接刪除的可能性降到最低。

如果英文是你的第一外語（Second language），提示潛在回覆者你有潛在的語言困難是很好的： [譯註：以下附上原文以供使用]

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

- 英文不是我的母語，請原諒我的錯字或文法

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

- 如果你說某語言，請寄信/私訊給我；我需要有人協助我翻譯我的問題

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

- 我對技術名詞很熟悉，但對於俗語或是特別用法比較不甚了解。

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

- 我把我的問題用某語言和英文寫出來，如果你只用一種語言回答，我會樂意將其翻譯成另一種。

使用易於讀取且標準的文件格式發送問題

如果你人爲地將問題搞得難以閱讀，它多半會被忽略，人們更願讀易懂的問題，所以：

- 使用純文字而不是 HTML (關閉 HTML 並不難)

- 使用 MIME 附件通常是可以的，前提是眞正有內容（譬如附帶的原始碼或patch），而不僅僅是郵件程式生成的模板（譬如只是信件內容的拷貝）。
- 不要發送一段文字只是單行句子但多次斷行的郵件（這使得回覆部分內容非常困難）。設想你的讀者是在80個字符寬的終端機上閱讀郵件，最好設置你的斷行點小於80字。
- 但是，也不要用任何固定斷行資料（譬如日誌檔案拷貝或會話記錄）。檔案應該原樣包含，讓回覆者有信心他們看到的是和你看到的一樣的東西。
- 在英語論壇中，不要使用 `Quoted-Printable` MIME編碼發送消息。這種編碼對於張貼非 ASCII 語言可能是必須的，但很多郵件程式並不支援這種編碼。當它們分斷時，那些文本中四處散佈的 `=20` 符號既難看也分散注意力，甚至有可能破壞內容的語意。
- 絕對，永遠不要指望黑客們閱讀使用封閉格式編寫的文檔，像是微軟公司的 Word 或 Excel 文件等。大多數黑客對此的反應就像有人將還在冒熱氣的豬糞倒在你門口階梯上時你的反應一樣。即便他們能夠處理，他們也很厭惡這麼做。
- 如果你從使用 Windows 的電腦發送電子郵件，關閉微軟愚蠢的 `智慧引號` 功能（從[選項] > [校訂] > [自動校正選項], 按掉 `智慧引號` 核取方塊），以免在你的郵件中到處散佈垃圾字符。
- 在論壇，勿濫用 `表情符號` 和 `HTML` 功能（當它們提供時）。一兩個表情符號通常沒有問題，但花哨的彩色文本傾向於使人認爲你是個無能之輩。過濫地使用表情符號、色彩和字體會使你看來像個傻笑的小姑娘。這通常不是個好主意，除非你只是對性而不是有用的回覆更有興趣。

如果你使用圖形用户界面的郵件程式（如微軟公司的 Outlook 或者其它類似的），注意它們的預設配置不一定滿足這些要求。大多數這類程式有基於選單的 `查看原始碼` 命令，用它來檢查發送文件夾中的消息，以確保發送的是沒有多餘雜質的純文本文件。

精確的描述問題並言之有物

- 仔細、清楚地描述你的問題或臭蟲的症狀。
- 描述問題發生的環境（機器配置、作業系統、應用程式、以及相關的資訊），提供經銷商的發行版和版本號（如：`Fedora Core 4` 、 `Slackware 9.1` 等）。
- 描述在提問前你是怎樣去研究和理解這個問題的。
- 描述在提問前爲確定問題而採取的診斷步驟。
- 描述最近做過什麼可能相關的硬體或軟體變更。
- 盡可能的提供一個可以 `重製這個問題的既定環境` 的方法

儘量去揣測一個黑客會怎樣反問你，在他提問的時候預先給他答案。

以上幾點中，當你回報的是你認爲可能在程式碼中的問題時，給黑客一個可以重製你的問題的環境尤其重要。當你這麼做時，你得到有效的回答的機會和速度都會大大的提升。

Simon Tatham 寫過一篇名爲《如何有效的回報Bug》的出色文章。強力推薦你也讀一讀。

話不在多而在精

你需要提供精確有內容的資訊。這並不是要求你簡單的把成堆的出錯程式碼或者資料完全轉錄到你的提問中。如果你有龐大而複雜的測試樣例能重現程式當掉的情境，儘量將它剪裁得越小越好。

這樣做的用處至少有三點。 第一，表現出你爲簡化問題付出了努力，這可以使你得到回答的機會增加； 第二，簡化問題使你更有可能得到有用的答案； 第三，在精鍊你的bug報告的過程中，你很可能就自己找到了解決方法或權宜之計。

別動輒聲稱找到Bug

當你在使用軟體中遇到問題，除非你非常、非常的有根據，不要動輒聲稱找到了Bug。提示：除非你能提供解決問題的原始碼補丁，或者對前一版本的回歸測試表現出不正確的行爲，否則你都多半不夠完全確信。這同樣適用在網頁和文件，如果你（聲稱）發現了文件的 `Bug` ，你應該能提供相應位置的修正或替代文件。

請記得，還有許多其它使用者沒遇到你發現的問題，否則你在閱讀文件或搜尋網頁時就應該發現了（你在抱怨前已經做了這些，是吧？）。這也意味著很有可能是你弄錯了而不是軟體本身有問題。

編寫軟體的人總是非常辛苦地使它盡可能完美。如果你聲稱找到了Bug，也就是在質疑他們的能力，即使你是對的，也有可能會冒犯到其中某部分人。這尤其嚴重當你在標題中嚷嚷著有 `Bug` 。

提問時，即使你私下非常確信已經發現一個眞正的臭蟲，最好寫得像是你做錯了什麼。如果眞的有臭蟲，你會在回覆中看到這點。這樣做的話，如果眞有臭蟲，維護者就會向你道歉，這總比你惹惱別人然後欠別人一個道歉要好一點。

別用低聲下氣取代你眞正該做的事

有些人明白他們不該粗魯或傲慢的提問並要求得到答覆，但他們選擇另一個極端 -- 低聲下氣： `我知道我只是個可悲的新手，一個魯蛇，但...` 。這既使人困擾，也沒有用，尤其是伴隨著與實際問題含糊不清的描述時更令人反感。

別用原始靈長類動物的把戲來浪費你我的時間。取而代之的是，盡可能清楚地描述背景條件和你的問題情況。這比低聲下氣更好地定位了你的位置。

有時網頁論壇會設有專爲新手提問的版面，如果你眞的認爲遇到了初學者的問題，到那去就是了，但一樣別那麼低聲下氣。

描述問題症狀而非猜測

告訴黑客們你認爲問題是怎樣造成的並沒什麼幫助。（如果你的推斷如此有效，還用向別人求助嗎？），因此要確信你原原本本告訴了他們問題的症狀，而不是你的解釋和理論；讓黑客們來推測和診斷。如果你認爲陳述自己的猜測很重要，清楚地說明這只是你的猜測，並描述爲什麼它們不起作用。

蠢問題

> 我在編譯內核時接連遇到 SIG11 錯誤， 我懷疑某條飛線搭在主板的走線上了，這種情況應該怎樣檢查最好？

聰明問題

> 我的組裝電腦是 FIC-PA2007 主機板搭載 AMD K6/233 CPU（威盛 Apollo VP2晶片組）， 256MB Corsair PC133 SDRAM記憶體，在編譯內核時，從開機20分鐘以後就頻頻產生 SIG11 錯誤， 但是在頭20分鐘內從沒發生過相同的問題。重新啓動也沒有用，但是關機一晚上就又能工作20分鐘。 所有記憶體都換過了，沒有效果。相關部分的標準編譯記錄如下...。

由於以上這點似乎讓許多人覺得難以配合，這裡有句話可以提醒你：`所有的診斷專家都來自密蘇里州。` 美國國務院的官方座右銘則是：`讓我看看`（出自國會議員 Willard D. Vandiver 在1899年時的講話：`我來自一個出產玉米，棉花，牛勞和民主黨人的國家，滔滔雄辯既不能說服我，也不會讓我滿意。我來自密蘇里州，你必須讓我看看。`） 針對診斷者而言，這並不是一種懷疑，而只是一種眞實而有用的需求，以便讓他們看到的是與你看到的原始證據盡可能一致的東西，而不是你的猜測與歸納的結論。所以，大方的展示給我們看吧！

按發生時間先後列出問題症狀

問題發生前的一系列操作，往往就是對找出問題最有幫助的線索。因此，你的說明裡應該包含你的操作步驟，以及機器和軟體的反應，直到問題發生。在命令列處理的情況下，提供一段操作記錄（例如運行腳本工具所生成的），並引用相關的若干行（如20行）記錄會非常有幫助。

如果當掉的程式有診斷選項（如 -v 的詳述開關），試著選擇這些能在記錄中增加除錯資訊的選項。記住，`多` 不等於 `好`。試著選取適當的除錯級別以便提供有用的信息而不是讓讀者淹沒在垃圾中。

如果你的說明很長（如超過四個段落），在開頭簡述問題，接下來再按時間順序詳述會有所幫助。這樣黑客們在讀你的記錄時就知道該注意哪些內容了。

描述目標而不是過程

如果你想弄清楚如何做某事（而不是報告一個Bug），在開頭就描述你的目標，然後才陳述重現你所卡住的特定步驟。

經常尋求技術幫助的人在心中有個更高層次的目標，而他們在自以爲能達到目標的特定道路上被卡住了，然後跑來問該怎麼走，但沒有意識到這條路本身就有問題。結果要費很大的勁才能搞定。

蠢問題

> 我怎樣才能從某繪圖程式的顏色選擇器中取得十六進制的的 RGB 值？

聰明問題

> 我正試著用替換一幅圖片的色碼成自己選定的色碼，我現在知道的唯一方法是編輯每個色碼區塊，但卻無法從某繪圖程式的顏色選擇器取得十六進制的的 RGB 值。

第二種提問法比較聰明，你可能得到像是 `建議採用另一個更適任的工具` 的回覆。

別要求使用私人電郵回覆

黑客們認爲問題的解決過程應該公開、透明，此過程中如果更有經驗的人注意到不完整或者不當之處，最初的回覆才能夠、也應該被糾正。同時，作爲提供幫助者也能因爲能力和學識被其它同行看到而得到某種獎勵。

當你要求私下回覆時，這個過程和獎勵都被中止。別這樣做，讓回覆者來決定是否私下回答 -- 如果他眞這麼做了，通常是因爲他認爲問題編寫太差或者太膚淺，以至於對其它人沒有興趣。

這條規則存在一條有但書的例外，如果你確信提問可能會引來大量雷同的回覆時，那麼這個神奇的提問句會是 `向我發電郵，我將爲論壇歸納這些回覆` 。試著將郵件列表或新聞群組從洪水般的雷同回覆中解救出來是非常有禮貌的 -- 但你必須信守諾言。

清楚明確的表達你的問題以及需求

漫無邊際的提問近乎無休無止的時間黑洞。最有可能給你有用答案的人通常也正是最忙的人（他們忙是因爲要親自完成大部分工作）。這樣的人對無節制的時間黑洞相當厭惡，所以他們也傾向於厭惡那些漫無邊際的提問。

如果你明確表述需要回答者做什麼（如提供指點、發送一段程式碼、檢查你的補丁、或是其他等等），就最有可能得到有用的答案。因爲這會定出一個時間和精力的上限，便於回答者能集中精力來幫你。這麼做很棒。

要理解專家們所處的世界，請把專業技能想像爲充裕的資源，而回覆的時間則是稀缺的資源。你要求他們奉獻的時間越少，你越有可能從眞正專業而且很忙的專家那裡得到解答。

所以，界定一下你的問題，使專家花在辨識你的問題和回答所需要付出的時間減到最少，這技巧對你有用答案相當有幫助 -- 但這技巧通常和簡化問題有所區別。因此，問 `我想更好的理解X，可否指點一下哪裡有好一點的說明？` 通常比問 `你能解釋一下X嗎？` 更好。如果你的程式碼不能運作，

通常請別人看看哪裡有問題，比要求別人替你改正要明智得多。

詢問有關程式碼的問題時

別要求他人幫你有問題的代碼除錯而不提示一下應該從何入手。張貼幾百行的代碼，然後說一聲：`它不會動`會讓你完全被忽略。只貼幾十行代碼，然後說一句：`在第七行以後，我期待它顯示 <x>，但實際出現的是 <y>`比較有可能讓你得到回應。

最有效描述程式問題的方法是提供最精簡的臭蟲展示測試示例（bug-demonstrating test case）。什麼是最精簡的測試示例? 那是問題的縮影；一小個程式片段能剛好展示出程式的異常行爲，而不包含其他令人分散注意力的內容。怎麼製作最精簡的測試示例？如果你知道哪一行或哪一段程式碼會造成異常的行爲，複製下來並加入足夠重現這個狀況的程式碼（例如，足以讓這段程式碼能被編譯/直譯/被應用程式處理）。如果你無法將問題縮減到一個特定區塊，就複製一份程式碼並移除不影響產生問題行爲的部分。總之，測試示例越小越好（查看話不在多而在精一節）。

一般而言，要得到一段相當精簡的測試示例並不太容易，但永遠先嘗試這樣做的是種好習慣。這種方式可以幫助你了解如何自行解決這個問題 —— 而且即使你的嘗試不成功，黑客們也會看到你在嘗試取得答案的過程中付出了努力，這可以讓他們更願意與你合作。

如果你只是想讓別人幫忙審（Review）一下代碼，在信的開頭就要說出來，並且一定要提到你認爲哪一部分特別需要關注以及爲什麼。

別把自己家庭作業的問題貼上來

黑客們很擅長分辨哪些問題是家庭作業式的問題；因爲我們中的大多數都曾自己解決這類問題。同樣，這些問題得由你來搞定，你會從中學到東西。你可以要求給點提示，但別要求得到完整的解決方案。

如果你懷疑自己碰到了一個家庭作業式的問題，但仍然無法解決，試試在使用者群組，論壇或（最後一招）在專案的使用者郵件列表或論壇中提問。儘管黑客們會看出來，但一些有經驗的使用者也許仍會給你一些提示。

去掉無意義的提問句

避免用無意義的話結束提問，例如`有人能幫我嗎？`或者`這有答案嗎？`。

首先：如果你對問題的描述不是很好，這樣問更是畫蛇添足。

其次：由於這樣問是畫蛇添足，黑客們會很厭煩你 -- 而且通常會用邏輯上正確，但毫無意義的回答來表示他們的蔑視， 例如：`沒錯，有人能幫你`或者`不，沒答案`。

一般來說，避免用`是或否`、`對或錯`、`有或沒有`類型的問句，除非你想得到是或否類型的回答。

即使你很急也不要在標題寫 `緊急`

這是你的問題，不是我們的。宣稱 `緊急` 極有可能事與願違：大多數黑客會直接刪除無禮和自私地企圖即時引起關注的問題。更嚴重的是， `緊急` 這個字（或是其他企圖引起關注的標題）通常會被垃圾信過濾器過濾掉 -- 你的問題可能永遠將無法得到解答。

有半個例外的情況是，如果你是在一些很高調，會使黑客們興奮的地方，也許值得這樣去做。在這種情況下，如果你有時間壓力，也很有禮貌地提到這點，人們也許會有興趣回答快一點。

當然，這風險很大，因爲黑客們興奮的點多半與你的不同。譬如從 NASA 國際空間站（International Space Station）發這樣的標題沒有問題，但用自我感覺良好的慈善行爲或政治原因發肯定不行。事實上，張貼諸如 `緊急：幫我救救這個毛絨絨的小海豹！` 肯定讓你被黑客忽略或惹惱他們，即使他們認爲毛絨絨的小海豹很重要。

如果你覺得這點很不可思議，最好再把這份指南剩下的內容多讀幾遍，直到你弄懂了再發文。

禮多人不怪，而且有時還很有幫助

彬彬有禮，多用 `請` 和 `謝謝您的關注` ，或 `謝謝你的關照` 。讓大家都知道你對他們花時間免費提供幫助心存感激。

坦白說，這一點並沒有比清晰、正確、精準並合法文法和避免使用專用格式重要（也不能取而代之）。黑客們一般寧可讀有點唐突但技術上鮮明的臭蟲報告，而不是那種有禮但含糊的報告。（如果這點讓你不解，記住我們是按問題能教我們什麼來評價問題的價值的）

然而，如果你有一串的問題待解決，客氣一點肯定會增加你得到有用回應的機會。

（我們注意到，自從本指南發佈後，從資深黑客那裡得到的唯一嚴重缺陷反饋，就是對預先道謝這一條。一些黑客覺得 `先謝了` 意味著事後就不用再感謝任何人的暗示。我們的建議是要麼先說 `先謝了` ，然後事後再對回覆者表示感謝，或者換種方式表達感激，譬如用 `謝謝你的關注` 或 `謝謝你的關照` 。）

問題解決後，加個簡短的補充說明

問題解決後，向所有幫助過你的人發個說明，讓他們知道問題是怎樣解決的，並再一次向他們表示感謝。如果問題在新聞組或者郵件列表中引起了廣泛關注，應該在那裡貼一個說明比較恰當。

最理想的方式是向最初提問的話題回覆此消息，並在標題中包含 `已修正` ， `已解決` 或其它同等含義的明顯標記。在人來人往的郵件列表裡，一個看見討論串 `問題 X` 和 `問題的X - 已解決` 的潛在回覆者就明白不用再浪費時間了（除非他個人覺得 `問題 X` 的有趣），因此可以利用此時間去解決其它問題。

補充說明不必很長或是很深入；簡單的一句 `你好，原來是網路線出了問題！謝謝大家 - Bill` 比什麼也不說要來的好。事實上，除非結論眞的很有技術含量，否則簡短可愛的小結比長篇大論更好。說明問題是怎樣解決的，但大可不必將解決問題的過程複述一遍。

對於有深度的問題，張貼除錯記錄的摘要是有幫助的。描述問題的最終狀態，說明是什麼解決了問題，在此之後才指明可以避免的盲點。避免盲點的部分應放在正確的解決方案和其它總結材料之後，而不要將此訊息搞成偵探推理小說。列出那些幫助過你的名字，會讓你交到更多朋友。

除了有禮貌和有內涵以外，這種類型的補充也有助於他人在郵件列表/新聞群組/論壇中搜尋到眞正解決你問題的方案，讓他們也從中受益。

至少，這種補充有助於讓每位參與協助的人因問題的解決而從中得到滿足感。如果你自己不是技術專家或者黑客，那就相信我們，這種感覺對於那些你向他們求助的大師或者專家而言，是非常重要的。問題懸而未決會讓人灰心；黑客們渴望看到問題被解決。好人有好報，滿足他們的渴望，你會在下次提問時嘗到甜頭。

思考一下怎樣才能避免他人將來也遇到類似的問題，自問寫一份文件或加個常見問題（FAQ）會不會有幫助。如果是的話就將它們發給維護者。

在黑客中，這種良好的後繼行動實際上比傳統的禮節更爲重要，也是你如何透過善待他人而贏得聲譽的方式，這是非常有價值的資產。

如何解讀答案

RTFM和STFW：如何知道你已完全搞砸了

有一個古老而神聖的傳統：如果你收到 `RTFM （Read The Fucking Manual）` 的回應，回答者認爲你應該去讀那該死的手冊。當然，基本上他是對的，你應該去讀一讀。

RTFM 有一個年輕的親戚。如果你收到 `STFW（Search The Fucking Web）` 的回應，回答者認爲你應該到該死的網路上搜索過了。那人多半也是對的，去搜尋一下吧。（更溫和一點的說法是 **Google**是你的朋友！）

在論壇，你也可能被要求去爬爬論壇的舊文。事實上，有人甚至可能熱心地爲你提供以前解決此問題的討論串。但不要依賴這種關照，提問前應該先搜索一下舊文。

通常，用這兩句之一回答你的人會給你一份包含你需要內容的手冊或者一個網址，而且他們打這些字的時候也正在讀著。這些答覆意味著回答者認爲

- 你需要的資訊非常容易獲得；
- 你自己去搜尋這些資訊比灌給你能讓你學到更多。

你不應該因此不爽；依照黑客的標準，他已經表示了對你一定程度的關注，而沒有對你的要求視而不見。你應該對他祖母般的慈祥表示感謝。

如果還是搞不懂

如果你看不懂回應，別立刻要求對方解釋。像你以前試著自己解決問題時那樣（利用手冊，FAQ，網路，身邊的高手），先試著去搞懂他的回應。如果你眞的需要對方解釋，記得表現出你已經從中學到了點什麼。

比方說，如果我回答你：`看來似乎是 zentry 卡住了；你應該先清除它。`，然後，這是一個很糟的後續問題回應：`zentry是什麼？` 好的問法應該是這樣：`哦~~~我看過說明了但是只有 -z 和 -p 兩個參數中提到了 zentries，而且還都沒有清楚的解釋如何清除它。你是指這兩個中的哪一個嗎？還是我看漏了什麼？`

處理無禮的回應

很多黑客圈子中看似無禮的行爲並不是存心冒犯。相反，它是直接了當，一針見血式的交流風格，這種風格更注重解決問題，而不是使人感覺舒服而卻模模糊糊。

如果你覺得被冒犯了，試著平靜地反應。如果有人眞的做了出格的事，郵件列表、新聞群組或論壇中的前輩多半會招呼他。如果這沒有發生而你卻發火了，那麼你發火對象的言語可能在黑客社區中看起來是正常的，而你將被視爲有錯的一方，這將傷害到你獲取訊息或幫助的機會。

另一方面，你偶而眞的會碰到無禮和無聊的言行。與上述相反，對眞正的冒犯者狠狠地打擊，用犀利的語言將其駁得體無完膚都是可以接受的。然而，在行事之前一定要非常非常的有根據。糾正無禮的言論與開始一場毫無意義的口水戰僅一線之隔，黑客們自己莽撞地越線的情況並不鮮見。如果你是新手或外人，避開這種莽撞的機會並不高。如果你想得到的是信息而不是消磨時光，這時最好不要把手放在鍵盤上以免冒險。

（有些人斷言很多黑客都有輕度的自閉症或亞斯伯格綜合症，缺少用於潤滑人類社會正常交往所需的神經。這既可能是眞也可能是假的。如果你自己不是黑客，興許你認爲我們腦袋有問題還能幫助你應付我們的古怪行爲。只管這麼幹好了，我們不在乎。我們喜歡我們現在這個樣子，並且通常對病患標記都有站得住腳的懷疑。）

在下一節，我們會談到另一個問題，當你行爲不當時所會受到的`冒犯`。

如何避免扮演失敗者

在黑客社區的論壇中有那麼幾次你可能會搞砸 -- 以本指南所描述到的或類似的方式。而你會在公開場合中被告知你是如何搞砸的，也許攻擊的言語中還會帶點夾七夾八的顏色。

這種事發生以後，你能做的最糟糕的事莫過於哀嚎你的遭遇、宣稱被口頭攻擊、要求道歉、高聲尖叫、憋悶氣、威脅訴諸法律、向其雇主報怨、忘了關馬桶蓋等等。相反地，你該這麼做：

熬過去，這很正常。事實上，它是有益健康且合理的。

社區的標準不會自行維持，它們是通過參與者積極而公開地執行來維持的。不要哭嚎所有的批評都應該通過私下的郵件傳送，它不是這樣運作的。當有人評論你的一個說法有誤或者提出不同看法時，堅持聲稱受到個人攻擊也毫無益處，這些都是失敗者的態度。

也有其它的黑客論壇，受過高禮節要求的誤導，禁止參與者張貼任何對別人帖子挑毛病的消息，並聲稱 `如果你不想幫助用户就閉嘴。` 結果造成有想法的參與者紛紛離開，這麼做只會使它們淪爲毫無意義的嘮叨與無用的技術論壇。

誇張的講法是：你要的是友善（以上述方式）還是有用？兩個裡面挑一個。

記著：當黑客說你搞砸了，並且（無論多麼刺耳）告訴你別再這樣做時，他正在爲關心你和他的社群而行動。對他而言，不理你並將你從他的生活中濾掉更簡單。如果你無法做到感謝，至少要表現地有點尊嚴，別大聲哀嚎，也別因爲自己是個有戲劇性超級敏感的靈魂和自以爲有資格的新來者，就指望別人像對待脆弱的洋娃娃那樣對你。

有時候，即使你沒有搞砸（或者只是在他的想像中你搞砸了），有些人也會無緣無故地攻擊你本人。在這種情況下，抱怨倒是眞的會把問題搞砸。

這些來找麻煩的人要麼是毫無辦法但自以爲是專家的不中用傢伙，要麼就是測試你是否眞會搞砸的心理專家。其它讀者要麼不理睬，要麼用自己的方式對付他們。這些來找麻煩的人在給他們自己找麻煩，這點你不用操心。

也別讓自己捲入口水戰，最好不要理睬大多數的口水戰 -- 當然，是在你檢驗它們只是口水戰，而並未指出你有搞砸的地方，且也沒有巧妙地將問題眞正的答案藏於其後（這也是有可能的）。

不該問的問題

以下是幾個經典蠢問題，以及黑客沒回答時心中所想的：

問題：我能在哪找到 X 程式或 X 資源？

問題：我怎樣用 X 做 Y？

問題：如何設定我的 shell 提示？

問題：我可以用 Bass-o-matic 文件轉換工具將 AcmeCorp 檔案轉換爲 TeX 格式嗎？

問題：我的程式/設定/SQL語句沒有用

問題：我的 Windows 電腦有問題，你能幫我嗎？

問題：我的程式不會動了，我認爲系統工具 X 有問題

問題：我在安裝 Linux（或者 X）時有問題，你能幫我嗎？

問題：我怎麼才能破解 root 帳號/竊取 OP 特權/讀別人的郵件呢？

> 問題：我能在哪找到 X 程式或 X 資源？

回答：就在我找到它的地方啊，白痴 -- 搜尋引擎的那一頭。天哪！難道還有人不會用 Google 嗎？

> 問題：我怎樣用 X 做 Y？

回答：如果你想解決的是 Y，提問時別給出可能並不恰當的方法。這種問題說明提問者不但對 X 完全無知，也對 Y 要解決的問題糊塗，還被特定形勢禁錮了思維。最好忽略這種人，等他們把問題搞清楚了再說。

> 問題：如何設定我的 shell 提示？？

回答：如果你有足夠的智慧提這個問題，你也該有足夠的智慧去 RTFM，然後自己去找出來。

> 問題：我可以用 Bass-o-matic 文件轉換工具將 AcmeCorp 檔案轉換爲 TeX 格式嗎？

回答：試試看就知道了。如果你試過，你既知道了答案，就不用浪費我的時間了。

> 問題：我的程式/設定/SQL語句沒有用

回答：這不算是問題吧，我對要我問你二十個問題才找得出你眞正問題的問題沒興趣 -- 我有更有意思的事要做呢。在看到這類問題的時候，我的反應通常不外如下三種

- 你還有什麼要補充的嗎？
- 眞糟糕，希望你能搞定。
- 這關我有什麼屁事？

> 問題：我的 Windows 電腦有問題，你能幫我嗎？

回答：能啊，扔掉微軟的垃圾，換個像 Linux 或 BSD 的開放原始碼作業系統吧。

注意：如果程式有官方版 Windows 或者與 Windows 有互動（如Samba），你可以問與Windows相關的問題， 只是別對問題是由 Windows 作業系統而不是程式本身造成的回覆感到驚訝， 因爲 Windows 一般來說實在太爛，這種說法通常都是對的。

> 問題：我的程式不會動了，我認爲系統工具 X 有問題

回答：你完全有可能是第一個注意到被成千上萬用户反覆使用的系統呼叫與函式庫檔案有明顯缺陷的人，更有可能的是你完全沒有根據。不同凡響的說法需要不同凡響的證據，當你這樣聲稱時，你必須有清楚而詳盡的缺陷說明文件作後盾。

> 問題：我在安裝 Linux（或者 X）時有問題，你能幫我嗎？

回答：不能，我只有親自在你的電腦上動手才能找到毛病。還是去找你當地的 Linux 使用群組者尋求實際的指導吧（你能在這兒找到使用者群組的清單）。

注意：如果安裝問題與某 Linux 的發行版有關，在它的郵件列表、論壇或本地使用者群組中提問也許是恰當的。此時，應描述問題的準確細節。在此之前，先用 `Linux` 和所有被懷疑的硬體作關鍵詞仔細搜尋。

> 問題：我怎麼才能破解 root 帳號/竊取 OP 特權/讀別人的郵件呢？

回答：想要這樣做，說明了你是個卑鄙小人；想找個黑客幫你，說明你是個白癡！

好問題與蠢問題

最後，我將透過舉一些例子，來說明怎樣聰明的提問；同一個問題的兩種問法被放在一起，一種是愚蠢的，另一種才是明智的。

蠢問題：

> 我可以在哪兒找到關於 Foonly Flurbamatic 的資料？

這種問法無非想得到 STFW 這樣的回答。

聰明問題：

> 我用Google 搜尋過 "Foonly Flurbamatic 2600"，但是沒找到有用的結果。誰知道上哪兒去找對這種設備編程的資料？

這個問題已經 STFW 過了，看起來他眞的遇到了麻煩。

蠢問題

> 我從 foo 項目找來的源碼沒法編譯。它怎麼這麼爛？

他覺得都是別人的錯，這個傲慢自大的提問者

聰明問題

> foo 專案代碼在 Nulix 6.2 版下無法編譯通過。我讀過了 FAQ，但裏面沒有提到跟 Nulix 有關的問題。這是我編譯過程的記錄，我有什麼做的不對的地方嗎？

提問者已經指明了環境，也讀過了FAQ，還列出了錯誤，並且他沒有把問題的責任推到別人頭上，他的問題值得被關注。

蠢問題

> 我的主機板有問題了，誰來幫我？

某黑客對這類問題的回答通常是：`好的，還要幫你拍拍背和換尿布嗎？`，然後按下刪除鍵。

聰明問題

> 我在 S2464 主機板上試過了 X 、 Y 和 Z ，但沒什麼作用，我又試了 A 、 B 和 C 。請注意當我嘗試 C 時的奇怪現象。顯然 florbish 正在 grommicking，但結果出人意料。通常在 Athlon MP 主機板上引起 grommicking 的原因是什麼？有誰知道接下來我該做些什麼測試才能找出問題？

這個傢伙，從另一個角度來看，值得去回答他。他表現出了解決問題的能力，而不是坐等天上掉答案。

在最後一個問題中，注意 `告訴我答案` 和 `給我啓示，指出我還應該做什麼診斷工作` 之間微妙而又重要的區別。

事實上，後一個問題源自於 2001 年 8 月在 Linux 內核郵件列表（lkml）上的一個眞實的提問。我（Eric）就是那個提出問題的人。我在 Tyan S2464 主板上觀察到了這種無法解釋的鎖定現象，列表成員們提供了解決這一問題的重要資訊。

通過我的提問方法，我給了別人可以咀嚼玩味的東西；我設法讓人們很容易參與並且被吸引進來。我顯示了自己具備和他們同等的能力，並邀請他們與我共同探討。通過告訴他們我所走過的彎路，以避免他們再浪費時間，我也表明了對他們寶貴時間的尊重。

事後，當我向每個人表示感謝，並且讚賞這次良好的討論經歷的時候， 一個 Linux 內核郵件列表的成員表示，他覺得我的問題得到解決並非由於我是這個列表中的名人，而是因爲我用了正確的方式來提問。

黑客從某種角度來說是擁有豐富知識但缺乏人情味的傢伙；我相信他是對的，如果我像個乞討者那樣提問，不論我是誰，一定會惹惱某些人或者被他們忽視。他建議我記下這件事，這直接導致了本指南的出現。

如果得不到回答

如果仍得不到回答，請不要以爲我們覺得無法幫助你。有時只是看到你問題的人不知道答案罷了。沒有回應不代表你被忽視，雖然不可否認這種差別很難區分。

總的來說，簡單的重複張貼問題是個很糟的點子。這將被視爲無意義的喧鬧。有點耐心，知道你問題答案的人可能生活在不同的時區，可能正在睡覺，也有可能你的問題一開始就沒有組織好。

你可以通過其他管道獲得幫助，這些管道通常更適合初學者的需要。

有許多網上的以及本地的使用者群組，由熱情的軟體愛好者（即使他們可能從沒親自寫過任何軟體）組成。通常人們組建這樣的團體來互相幫助並幫助新手。

另外，你可以向很多商業公司尋求幫助，不論公司大還是小。別爲要付費才能獲得幫助而感到沮喪！畢竟，假使你的汽車發動機汽缸密封圈爆掉了-- 完全可能如此 --你還得把它送到修車鋪，並且爲維修付費。就算軟體沒花費你一分錢，你也不能強求技術支援總是免費的。

對像是 Linux 這種大眾化的軟體，每個開發者至少會對應到上萬名使用者。根本不可能由一個人來處理來自上萬名使用者的求助電話。要知道，即使你要爲這些協助付費，和你所購買的同類軟體相比，你所付出的也是微不足道的（通常封閉原始碼軟體的技術支援費用比開放原始碼軟體的要高得多，且內容也沒那麼豐富）。

如何更好地回答問題

態度和善一點。問題帶來的壓力常使人顯得無禮或愚蠢，其實並不是這樣。

對初犯者私下回覆。對那些坦誠犯錯之人沒有必要當眾羞辱，一個眞正的新手也許連怎麼搜尋或在哪找常見問題都不知道。

如果你不確定，一定要說出來！一個聽起來權威的錯誤回覆比沒有還要糟，別因爲聽起來像個專家很好玩，就給別人亂指路。要謙虛和誠實，給提問者與同行都樹個好榜樣。

如果幫不了忙，也別妨礙他。不要在實際步驟上開玩笑，那樣也許會毀了使用者的配置 --有些可憐的呆瓜會把它當成眞的指令。

試探性的反問以引出更多的細節。如果你做得好，提問者可以學到點東西 --你也可以。試試將蠢問題轉變成好問題，別忘了我們都曾是新手。

儘管對那些懶蟲抱怨一聲 RTFM 是正當的，能指出文件的位置（即使只是建議個 Google 搜尋關鍵詞）會更好。

如果你決定回答，就請給出好的答案。當別人正在用錯誤的工具或方法時別建議笨拙的權宜之計（workaround），應推薦更好的工具，重新界定問題。

正面的回答問題！如果這個提問者已經很深入的研究而且也表明已經試過 X 、 Y 、 Z 、 A 、 B 、 C 但沒得到結果，回答 `試試看 A 或是 B` 或者 `試試X 、 Y 、 Z 、 A 、 B 、 C` 並附上一個連結一點用都沒有。

幫助你的社群從問題中學習。當回覆一個好問題時，問問自己 `如何修改相關文件或常見問題文件以免再次解答同樣的問題？` ，接著再向文件維護者發一份補丁。

如果你是在研究一番後才做出的回答，展現你的技巧而不是直接端出結果。畢竟 `給人吃魚不如教他釣魚` 。

相關資源

如果你需要個人電腦、Unix 系統和網路如何運作的基礎知識，參閱Unix系統和網路基本原理。

當你發布軟體或補丁時，試著按軟體發布實踐操作。

鳴謝

Evelyn Mitchel 貢獻了一些愚蠢問題例子並啓發了編寫 `如何更好地回答問題` 這一節， Mikhail Ramendik 貢獻了一些特別有價值的建議和改進。