How to ask Questions the smart way

RaymondEric

esr (at) thysus.com

도운이

정문식

ruseel (at) yahoo.co.kr

좀더 나은 질문을 하기 위한 방법

고친 과정

고침 2.4 2001-10-17고친이 esr
문제 해결 답글의 올바른 형식
고침 2.3 2002-09-25고친이 esr
독일 번역문 링크 추가
고침 2.2 2002-09-22고친이 esr
'미리 감사드립니다'절에 내용 추가
고침 2.1 2002-08-27고친이 esr
'어떻게 게시판을 선택할 것인가'에 내용 추가
고침 2.0 2002-08-04고친이 esr
첫 XML 버전. Evelyn Mitchell의 프레젠테이션으로 부터의 조언. 그녕의 '어떻게 좋은 답을 줄 것인가'를 뺐음. '답
고침 1.192002-06-20고친이 esr
Bass-O-Matic 질문 예제와 문제해결 단계 추가
고침 1.182002-06-11고친이 esr
답장요구 편지의 제목 변경
고침 1.172002-04-21고친이 esr
게시판 선택 방법에 내용 추가
고침 1.162002-03-15고친이 esr
게시판 선택 방법에 내용 추가
고침 1.152002-03-11고친이 esr
질문에 급함이라고 적지마시오. line-wrap하지 마시오.
고침 1.142002-02-04고친이 esr

관련 자료 링크 추가
고침 1.132002-02-01고친이 esr
두개의 새로운 번역 추가. 오타 수정.
고침 1.122002-01-01고친이 esr
추천되는, 버그 리포트에 쓰이는 '주제 - 설명' 포맷. 자기 메일 형식이 거지같지 않은지 확인하는 방법.
고침 1.112001-10-11고친이 esr
프로젝트 메일링리스트에 관해 내용추가
고침 1.102001-10-05고친이 esr
프로젝트 메일링리스트에 관해 내용추가
고침 1.9 2001-10-02고친이 esr
메일첨부파일은 OK. HTML을 끄는 방법. 숙제 질문.
고침 1.8 2001-09-28고친이 esr
질문을 명확하기 하기
고침 1.7 2001-09-21고친이 esr
이메일 에티켓에 대한 몇 가지 팁. '미리 감사드립니다'에 대한 반대의견
고침 1.6 2002-09-14고친이 esr
답을 얻지 못했을 때 어떻게 할 것인가에 내용추가
고침 1.5 2001-09-13고친이 esr
'양이 많다고 정확한 것은 아니다'와 '무례한 사람 상대하기' 추가
고침 1.4 2002-09-11고친이 esr
사소한 편집 변경
고침 1.3 2001-09-10고친이 esr
해커적 성향을 좋아 하지 않는다면 어떻게 할 것 인가에 주석 추가. 또 다른 전형적인 바보 질문 추가
고침 1.2 2001-09-09고친이 esr
Richard Gooch로 부터의 공헌
고침 1.1 2001-09-09고친이 esr
William Stearns로 부터의 공헌

1. 서문

해커들의 세계에서 기술적인 질문에 대한 답의 질은 '답하기의 어려움'만큼이나 '어떻게 질문하느냐'에 달려있다. 이 가이드는 만족할만한 답을 얻기 위해 어떻게 질문해야 하느냐을 설명한다.

먼저 알아두어야 할 것은, 해커들이 어려운 문제를 좋아하고 그 문제에 대한 좋은(생각을 자극하는) 질문을 좋아한다는 것이다. 해커들이 그렇지 않다면 지금 우리는 이자리에 있지 않을 것이다. 만약 곱씹을 만한 좋은 질문을 한다면 우리는 당신에게 친절할 것이다. 좋은 질문은 자극제이자 선물이다. 좋은 질문은, 우리가 생각하지 못했거나 알아채지 못한 문제를 발견할 수 있게 하고 우리가 이해하는데 도움을 준다. 해커들 사이에서, "좋은 질문!"이라는 표현은 진심에서 우러나온 칭찬이다.

이에 불구하고 해커들은 간단한 질문에 대해 적대적이거나 오만해 보인다는 이야기를 자주 듣고 있다. 해커들은 초보자나 잘 모르는 사람들에 대해 무례한 것 처럼 보이고 있는 것 같다. 이건 사실이 아니다.

우리는 다만 질문을 하기 전에 생각조차 하지 않거나 당연히 해야 할 일을 하지 않는 사람들에 대해 적대적일 뿐이다. 그런 사람들은 시간을 소비하게만 한다. 그들은 자기들이 받은 것을 다시 되돌려주지도 않으면서, 우리가 흥미있는 질문이나 대답할 가치가 있는 사람들에게 쓸 시간을 버리게 한다. 우리는 이런 사람들을 멍청이(losers)라고 부른다.(그리고 역사적인 이유로 가끔 lusers라고 적는다.)

세상에는 우리가 만든 소프트웨어의 사용방법을 배울 생각도 없이 그냥 '사용'만 하고 싶어하는 사람이 매우 많다는 것을 깨닫고 있다. 그들에게 컴퓨터는 그냥 결과를 만들어 내는 도구일 뿐이다. 그들은 살기위해 살아가고 있을 뿐이고, 더 중요한 할 일이 있다. 우리는 그것을 알았고 이런 사람들이 우리를 흥분시킨 기술적인 것에 대해 관심을 가질 것이라 기대하지는 않는다. 그렇다하더라도 우리의 답변 스타일은 기술적인 것에 관심을 가지고 있고 문제해결에 적극성을 띄는 사람들에게 좀 더 친절하다. 이 것은 바뀌지 않을 것이다. 그래서도 안된다. 그렇게 했다가는 우리가 지금 하고 있는 일에 효율적이 되지 못할 것이다.

우리는 대부분 자원봉사자다. 바쁜 삶 속에서도 질문에 답하기 위해 시간을 쓰고 있고, 대부분 질문은 너무 많다. 그래서 질문들을 가차없이 잘라낸다. '승리자'들의 질문에 답하기 위해 '멍청이'로 보이는 사람들로 보이는 사람들의 질문을 버려 버린다.

이런 태도가 비위가 상하고 거만하다고 느낀다면, 당신의 가설을 점검해 보라. 모든 사람이 우리에게 존경을 표하기를 원하는 것이 아니다. 사실 우리는, 그 사람이 노력 한다면, 모든 사람을 동등하게 대하기를 원하고 우리 세계에 들어 오는 것을 환영한다. 무지한 것은 괜찮지만 멍청하게 구는 것은 괜찮지 않다.

우리의 관심을 얻기위해 미리 기술적인 능력을 가져야 한다는 것은 아니지만, 솔루션을 개발하는데 기꺼이 참여하려고 하고, 사려깊고, 주의깊음을 나타낼 수 있는 태도를 보여주는 것은 필요하다.

우리를 도와주고 싶다면, '멍청이'들 중 하나가 되고 싶지는 않을 것이다. '멍청이'처럼 보이고 싶지도 않을 것이다. 빠르고 훌륭한 답변을 얻는 가장 좋은 밥법은 자기가 똑똑하고, 자신감있고, 우연히 특별한 문제를 만났지만 단서를 가지고 있는 사람처럼 보이게 하는 것이다.

2. 질문하기 전에

이메일이나 뉴스그룹, 웹사이트를 통해서 질문을 하기 전에 해야할 일은 다음과 같다.

1. 웹을 검색해서 답을 찾아보기
2. 매뉴얼을 읽어서 답을 찾아보기
3. FAQ를 보고 받을 찾아보기
4. 실험과 조사를 통해 답을 찾아보기
5. 경험을 가진 친구를 통해 답을 찾아보기
6. 프로그래머라면 소스코드를 보고 답을 찾아보기

질문을 할 때 이러한 일들을 먼저했다고 표현해라. 이렇게 하는 것이 당신이 게으른 스폰지처럼 사람들의 시간을 쓸 모 없이 소비하게만 만드는 사람도 아니라는 것을 믿게 하는데 도움

이 된다. 더 좋은 것은 당신이 이 일들을 하면서 무얼 배웠는지 적는 것이다. 우리는 답으로 부터 배울 수 있다고 증명한 사람들에게 답변하는 것을 좋아한다.

에러메세지를 구글에서 검색해보라.(웹페이지 뿐만 아니라 뉴스그룹도 같이 검색해보라). 당신의 질문에 답한 메일쓰레드나 버그픽스 문서를 바로 발견할 수 있을지도 모른다. 만약 답을 발견하지 못했다고 하더라도 "저는 다음 에러메세지를 검색해보았지만 쓸모있어 보이는 것을 발견하지 못했습니다."라고 적는 것은 도움을 청하는 이메일이나, 뉴스그룹에 포스팅하는데 좋은 습관이다.

질문하기 전에 준비해라. 주의깊게 생각해라. 경솔하게 들리는 질문은 성의없는 답변을 얻거나, 아무 답변도 듣지 못한다. 문제를 해결하는데 들인 노력을 보여줄 수록 답변을 얻을 가능성은 늘어난다.

잘못된 질문을 하는 것에 주의해라. 잘못된 가정에 근거한 질문을 누군가가 하게되면, 대부분의 해커는 바보같은 질문이라고 생각하면서 정말로 쓸모없는 답을 줄것이다. 그리고 '당신이 원하던 것'이 아니라 '당신을 가르쳐줄 것'을 해보기를 원할 것이다.

절대로 당신이 답을 얻어야 한다고 기대하지 말아라. 당신은 그렇지 않다. 무엇보다도 당신은 서비스를 위해 돈을 내지 않고 있다. 당신이 중요하고 흥미있고 생각을 자극하는 질문(단순히 다른 사람들로 부터 지식을 원하는 것이 아니라, 암묵적으로 공통체에 기여하는 질문)을 한다면 답을 얻을 수 있을 것이다.

또한 솔루션 개발에 기꺼히 참여할 만한 의사가 있다는 것을 명확히 하는 것도 좋은 방법이다. "제가 한 방법이 빠트리고 있는 것이 무엇인가요?", "누가 좋은 문서를 적은 것이 없나요?" 그리고 "어떤 사이트를 참조해 보는 것이 좋을까요?"라는 질문이 "제게 해결방법 것을 꼭 알려주세요"라는 것 보다 답변받을 가능성이 높다. 올바른 방향만 제시받는다면 당신이 정말로 문제를 해결할 수 있다는 것을 명확히 하고 있기 때문이다.

3. 질문할 때

3.1. 게시판 선택하기

어디에 질문을 할지 선택할 때 신중하라. 다음과 같은 경우 무시되거나 멍청이로 취급받을 수 있다.

1. 게시판의 주제에 벗어나는 질문을 올릴 때
2. 고급 기술 질문을 다루는 곳에서 기초적인 질문을 할 때, 혹은 반대되는 경우
3. 너무 많은 뉴스그룹에 동시투고 할 때
4. 당신을 알고 있지 못하고 문제를 해결해야 할 의무도 없는 사람에게 이메일을 보냈을 때

해커들은 공동체의 대화수단이 엉뚱한 글들로 가득채워지는 것을 방지하기 위해 당신의 글을 날려버릴 수 도 있다. 당신에게 이러한 일이 일어나기를 바라지는 않을 것이다.

그래서 첫 번째 단계는 적당한 게시판을 찾는 것이다. 역시 Google과 다른 웹서치엔진들이 도움이 될 수 있다. 문제와 관련된 하드웨어 또는 소프트웨어의 프로젝트 홈페이지를 찾는데

검색엔진을 이용하십시오. 보통 프로젝트 홈페이지는 FAQ(자주 묻는 질문과 답), 메일링리스트를 를 가지고 있습니다. 이 것들이 당신의 노력이 답을 찾지 못한다면 마지막으로 도움을 요청해야 할 곳 입니다.

당신이 익숙하지 않은 게시판이나 사람에게 질문을 하는 것은 위험합니다. 좋은 정보가 있는 사이트의 주인이 당신에게 꽁짜로 컨설팅을 해줄 것이라고 기대하지 마십시오. 당신의 질문이 환영받을 것이라고 어떤 긍적적인 가정도 하지 마십시오. 확실하지 않다면 다른 곳으로 보내거나 질문을 보내는 것을 그만두십시오.

뉴스그룹이나 메일링리스트를 선택할 때, 이름만 보고 그 내용을 판단하지 마십시오. FAQ를 보고 당신의 질문이 그곳의 주제와 관련된 것인지 확인해 보십시오. 먼저 이미 그 곳에 포스팅되어 있는 글을 읽어 보면 일이 어떻게 진행되고 있는지 감을 잡을 수 있을 겁니다. 이렇게 해서 답을 찾을 수 도 있습니다. 혹은 좀 더 나은 질문을 만들 수 있습니다.

당신의 주제가 뭔지 알고있어야 합니다. 전형적인 실수는 Unix, Window양쪽에서 돌아가는 라이브러리, 언어, 툴을 다루는 곳에서 Unix, Window 프로그래밍 인터페이스에 관해 묻는 것입니다. 이것이 왜 실수인지 알 수 없다면 알기전까지 어떤 질문도 하지 않는 것이 좋을 것 같습니다.

대개 잘 선택된, 그리고 공개된 곳에 하는 질문이 비공개 된 곳에 하는 질문보다 대답받을 확률이 높습니다. 여러가지 이유가 있을 수 있습니다. 하나는 답변할 수 있는 사람의 숫자가 더 많다는 것 이고. 다른 하나는 그 청중의 수가 많다라는 것입니다. 해커는 몇 명만 알 수 있는 것 보다 한 꺼번에 여러명을 가르칠 수 있는 곳에 답변하고 싶을 겁니다.

이해할 수 있겠지만, 실력있는 해커나 유명한 소프트웨어의 저작자 들은 이미 그들이 처리할 수 있는 범위 밖의, 잘 못 보내진 메일을 받고 있습니다. 이 메일 홍수 속에서 당신은 이미 낙타의 등을 부러트릴 지푸라기 한 올이 될 수도 있습니다. 벌써 몇 번이나, 유명한 프로젝트의 공헌자들이 쓸모없는 이메일들로 인해 개인적인 메일계정이 피해를 입어 사용자 지원을 중지한 경우가 있습니다.

3.2. 가능하다면 프로젝트 메일링리스트를 이용해라

프로젝트가 개발 메일링리스트를 가지고 있다면 메일링리스트에 질문을 올리십시오. 누가 가장 잘 답변해 줄 지 알때라도, 개인개발자에게 메일을 직접 보내지 마십시오. 메일링리스트 주소를 프로젝트의 참조문서나 홈페이지에서 확인해서 이용하십시오. 여기에는 몇 가지 이유가 있습니다.

- 답변받을 만큼 좋은 질문은 전체그룹에 공유될 가치가 있습니다. 반대로 질문이 너무 바보같다면 개인개발자를 성가시게 한 것은 용서받을 수 없습니다.
- 메일링리스트에 질문하는 것은 개발자들간의 일을 나누어 줍니다. 개인개발자 특히 리더는 질문에 답하기에는 너무 바쁠 수 있습니다.
- 대부분의 메일링리스트는 아카이브에 저장되고 아카이브는 서치엔진에 의해 인덱스됩니다. 나중에 다른 사람이, 리스트에 같은 질문을 올리는 것 대신에, 당신의 질문을 찾고 답을 볼 수 있을 지도 모릅니다.

- 어떤 질문이 자주 리스트에 보인다면, 개발자는 그 정보를 문서를 고치거나 소프트웨어를 좀 덜 헷갈리도록 고치는 데 사용할 수 있습니다. 그러나 개인적으로 질문이 된다면 아무도 어떤 질문이 자주 되는지 모를 것 입니다.

만약에 프로젝트 메일링리스트를 확인해 볼 수 없고 프로젝트관리자의 메일 주소만 보인다면 그 쪽으로 메일을 보내십시오. 그러나 이 경우에도 메일링리스트가 없을 거라고 가정하지 마십시오. 메일에 메일링리스트를 찾으려고 했지만 실패했다고 적으십시오. 그리고 다른 사람에게 메세지가 포워드되어도 괜찮다고 언급하십시오. (많은 사람들이, 메일 안에 비밀이 없더라도, 개인적인 e-mail은 개인적으로 남아야 한다고 믿습니다. 당신의 메일이 포워드가능하도록 허락해서 메일을 받는 사람이 메일처리방법을 선택하도록 할 수 있습니다.)

3.3. 답변하기 쉽게 질문하기

질문을 "xxx@xxx.com으로 답변해주세요"라고 끝내는 것은 오히려 더 답변을 얻기 힘들게 합니다. 메일프로그램에서 Reply-To 헤더를 설정하는 데 몇 초를 쓸 수 없다면, 우리도 문제를 생각하는데 몇 초라도 쓸 수 없습니다. 메일프로그램이 Reply-To 헤더 설정을 지원하지 않는다면 메일프로그램을 바꾸십시오. OS가 이 기능을 사용할 수 있는 어떤 프로그램도 지원하지 않는다면 OS를 바꾸십시오.

3.4. 명확하고, 문법적으로 올바른 질문하기

우리는 경험으로 주의가 부족한 조잡한 문장을 쓰는 사람들이 생각과 코딩도 마찬가지라는 것을 알고있다. (내기를 걸어도 될 많큼 충분히 많다.) 주의부족인 조잡한 생각을 하는 사람들을 위해 답변하는 것은 쓸 모 없는 짓이다. 차라리 다른 곳에 시간을 사용할 것 이다.

그래서 질문을 표현할 때는 명확히 하는 것이 중요하다. 그렇게 하는데 신경을 쓸 수 없다면 우리도 신경쓸 수 없다. 당신의 언어를 갈고 닦는데 노력해라. 질문이 형식을 차릴 필요는 없다. 사실 해커들은 비형식적이고, 자유롭고 유머있는 언어가 정확하게 쓰여진 것을 좋아한다. 다만 질문에 당신이 생각하고 신경썼다는 것을 알릴만한 표시가 있어야 한다.

문법적으로 올바로 적어라. "its"를 "it's"와, "loose"를 "lose"와 "discrete"를 "discreet"와 헷갈리지 말아라. 모든 글자를 대문자로 적지 말아라. 이 것은 소리지르는 것처럼 보이고 무례한 것으로 받아 들여 진다. (모든 글자가 소문자인 것은 좀 덜 화나지만 읽기가 힘들다. Alan Cox는 그 럴 수 있지만, 당신은 그렇지 않다.)

만약 적당히 교육된 얼간이 처럼 쓴다면 대부분 무시당할 것이다. 'l33t script kiddie haxOr'과 같이 쓰는 것은 죽음에의 키스이고 당신이 무거운 침묵이외에는 받을 것이 아무것도 없다는 것을 보장한다. (또는 경멸과 비꼼이 가득찬 도움을 받을 것이다.)

모국언어가 쓰이지 않는 곳에서 질문을 하다면, 아마도 약간의 철자와 문법에러는 용서 받을 것이다. 그리나 게으름 때문에 생긴 잘못은 용서할 수 없다.(그렇다. 우리는 차이점을 알 수 있다.) 그리고 상대방이 어떤 언어를 사용할지 알 수 없다면 영어를 사용하라. 바쁜 해커들은 알 수 없는 언어로 쓰여진 질문들은 무시하는 경향이 있다. 그리고 영어는 인터넷에서 주로 사용하는 언어이다. 영어를 사용함으로써 질문이 읽어지지 않는 경우를 최소화할 수 있다.

3.5. 이해하기 쉬운 형식으로 질문하기

질문이 읽기가 힘들다면, 다음 중에 하나를 만족하고 있지 않은지 확인해 보라.

- HTML이 아니라 Text형식으로 보내
- MIME 파일첨부는 보통 괜찮다. 메일프로그램에 의해 자동생성된 것이 아니라 내용이 있는 것(source파일, patch)이라면...
- 전체 달락이 한 줄로 되어 있는 메일을 보내지 말라. (메세지의 일부분만을 읽기 힘들게 만든다.) 상대방이 80컴럼을 가지고 있다고 가정하고 80보다 작은 값으로 line wrap을 설정하라.
- 그러나 로그파일 덤프나 session스크립트 같은 data 는 고정된 컬럼크기로 wrap하지 마라. data는 그대로 포함되어야 한다. 상대방이 당신이 본 것과 똑 같은 것을 보고 있다고 믿을 수 있어야 한다.
- MIME Queoted-Printable 인코딩으로 된 메일은 영어를 쓰는 곳으로 보내지 마라. 이 인코딩은 ASCII코드가 표현할 수 없는 언어를 쓰는 곳에서만 필요가 있다. 그러나 많은 메일프로그램이 이 인코딩을 지원하지 않아서, '=20'과 같은 글자가 글자안에 보기 싫게 흩어져 보일 수 있다.
- 절대로 해커가 Miscrosoft 워드같은 비공개된 파일 포맷을 읽을 수 있을 거라고 생각하지 말라.
- 윈도우에서 메일을 보내고 있다면, Microsoft의 "Smart Quotes" 기능을 꺼라. 이 기능을 끄면 메일안에서 번쩍이는 쓰레기 글자를 보지 않아도 된다.

만약에 GUI 메일 클라이언트를 이용하고 있다면 (MS Outlook, Netscape Messenger, etc) 이 프로그램들을 디폴트 설정으로 이용하는 것은 위의 룰들을 깰 수 있다는 것을 명심하라. 대 부분, 이런 메일 클라이언트들은 "View Source" 메뉴를 가지고 있다. 이 메뉴를 "보낸 메일"폴더에서 사용해 보고 불필요한 찌꺼기가 첨부로 붙지 않고 plain text인 메일을 보내고 있는지 확인해라.

3.6. 의미있고 분명한 제목을 사용하라**.**

제목은, 메일링리스트나 뉴스그룹에서, 50개 정도의 글자로 검증받은 숙련된 전문가들의 관심을 끌 수 있는 황금같은 기회이다. "제발 도와주세요"와 같은 쓸모없는 말을 쓰지 말아라. ("제발 도와주세요" 같은 제목을 보면 무시해버리세요. 이런 제목을 가진 글은 반사적으로 무시될 겁니다.) 당신의 얼마나 급한 사정인지로 우리를 감동시키려고 하지 마십시오. 공간을 당신의 문제를 설명하는데 사용하십시오.

많은 사용자지원 기관에서 사용되는, 제목작성에 사용되는 관례는 '주제 - 이상한 점' 입니다. '주제' 부분에는 문제를 격고 있는 것을 적고 '이상한 점'에는 예상했던 것과 다른 행동을 보이는 것을 적습니다.

바보: 도와주세요. 제 노트북에서 비디오가 나오질 않아요!

똑똑함: XFree86 4.1 misshapen mouse cusor, Fooware MV1005 vid. chipset

더 똑똑함: XFree86 4.1 mouse cursor on Fooware MV1005 vid. chipset - is misshapen

"주제 - 이상한점"으로 설명을 적으면 생각을 좀 더 다듬는데 도움이 될 겁니다. 뭐가 이상하냐고요? 마우스 커서 뿐인가요, 아니면 다른 그래픽도 그런가요? XFree86만 이상한가요? 버전 4.1에서만 이상한가요? Fooware 비디오 칩셋에서만 그런가요? 모델 MV1005에서만? 이 제목을 보는 사람은 한 눈에 당신이 어떤 것과 문제를 겪고 있는지 어떤 문제를 겪고 있는지 알 수 있을 겁니다.

답변하기 버튼을 눌러 질문을 하고 있다면 질문하고 있다는 것을 나타내도록 제목을 바꿔라. 다음과 같은 제목, "Re : test" 또는 "Re: new bug"는 충분한 주의를 끌지 못 할 것이다. 그리고 새로 이 글을 읽는 사람들이 단서를 얻을 수 있는 최소한의 부분을 제외하고 이전 메세지를 인용한 부분을 잘라내십시오.

3.7. 명확하지만 자세하게 질문하라

- 문제나 버그의 증상을 주의깊게 명확하게 설명하라.
- 문제가 일어난 환경을 설명하라.(machine, OS, application, whatever)
- 질문하기 전에 문제를 해결하거나 이해하기 위해 해보았던 것를 설명하라.
- 질문하기 전에 문제의 범위를 줄이기 위해 시도해 본, 진단에 도움이 되는 절차를 설명해라.
- 관련이 있을 만한, 컴퓨터나 소프트웨어에 최근에 일어난 변화를 설명하라.

당신이 도움을 요청할 때 해커가 할 만한 질문을 예상하고, 거기에 미리 답해 놓기 위해 최선을 다하라.

Simon Tatham은 효율적으로 버그 리포트하는 방법 (http://www.chiark.greenend.org.uk/~sgtatham/bugs.html)이라는 훌륭한 에세이를 썼다. 한 번 읽어 볼 것을 추천한다.

3.8. 양이 많다고 명확한 것은 아니다

질문은 명확하지만 자세해야 한다. 이것은 한 무더기의 code와 data를 적는다고 해서 되는 것이 아니다. 만약에 버그를 일으키는 크고 복잡한 테스트 케이스를 가지고 있다면 될 수 있는 대로 잘라내고 최대한 작게 만들어야 한다.

이 것은 적어도 세가지 점에서 유용하다. 첫 째, 질문을 단순화 시키려고 노력함으로써 답변받을 가능성을 높일 수 있다. 둘 째, 질문을 단순화 시키는 것은 유용한 답변을 얻을 가능성을 높인다. 셋 째, 버그 리포트를 만들려고 노력하면서 fix를 개발하거나 workaround를 만들 수 있다.

3.9. 추측이 아니라 증상을 설명하라.

뭐가 문제를 일으키는 것 같다는 추측을 해커에게 말하는 것은 도움이 되지 않는다. (당신의 진단이론이 그렇게 훌륭하다면 왜 다른 사람에게 도움을 바라나요.) 당신의 판단이나 이론을 말하는 것이 아니라 뭐가 잘못되었지 가공되지 않은 증상을 말하도록 해라. 해커가 판단하고 해결하게 하라.

바보 : 커널 컴파일 도중에 SIG11 에러가 때때로 일어 나고 있습니다. 마더보드에 미세한 균열이 일어난 것이 아닐까 싶습니다. 이 문제인지 체크하려면 뭐가 가장 좋은 방법일까요?

똑똑함: 집에서 조립한 FIC-PA2007마더보드(VIA Apollo VP2 chipset) K6/233에 장착된 Corsair PC133 SDRAM이, 파워를 켠 후 20분쯤 후 커널 컴파일 도중에 SIG11에러가 발생하고 있습니다. 첫 20분간은 에러가 발생하지 않습니다. 리부팅을 하면 시계가 다시 시작하지 않고, 밤 새도록 전원을 꺼둔다음 리부팅하면 그렇습니다. 모든 RAM을 바꿔 보았지만 소용없고요. 컴파일 세션에서 관련된 부분은 다음과 같습니다.

3.10. 문제의 증상을 시간 순서대로 설명하라

뭐가 잘 못되 었는지 알아내는데 쓰이는 가장 유용한 단서는 문제가 일어나기 바로 전에 일어난 사건에 있는 경우가 많다. 그래서 문제의 설명은 정확히 "당신이 무엇을 했고, 컴퓨터가 무엇을 해서 문제가 일어났다"라고 적어야 한다. 커맨드라인 명령의 경우 session log를 남겨서 관련이 있는 20줄 정도를 인용하는 것은 매우 유용하다.

만약에 프로그램이 -v 같은 진단 옵션을 가지고 있다면, 어떤 옵션이 디버깅에 유용한 정보를 줄 지 잘 판단해서 실행해 본다.

만약이 이야기가 길다면 (4단락 보다 길다면) 문제를 제일 위에 적고 , 그 아래에 시간순서대로 적는 것이 도움이 된다. 이렇게 하면 해커가 이야기를 읽으면서 뭘 주의해야하는지 알 것이다.

3.11. 개인메일로 답장을 달라고 요구하지 않기

해커들은 문제 해결이 공개적이고 투명한 과정 - 불완전하거나 잘못된 첫 번째 해결책을 보다 더 잘 아는 사람이 수정할 수 있고 그래야 하는 -을 통해 이루어 지기를 바란다. 그리고 그들의 동료들에게 능력있는 사람이라고 인정받는 것으로 보상받기를 원한다.

개인메일로 답장을 받기를 원하는 것은, 공개된 과정과 보상 둘 다를 방해하는 것이다. 개인메일로 답장을 요구하지 말라. 개인적으로 답장메일을 보내는 것은 상대의 선택일 뿐이다. 만약에 상대방이 답장을 준다면 보통 질문이 너무 형식에 맞지 않거나 다른 사람들에게도 분명히 흥미있을 거라고 생각해서 일 것이다.

여기에 한 가지 예외가 있을 수 있다. 질문이 너무 비슷한 종류의 해결책을 많이 받을 것 같다고 생각되는 경우, "저에게 메일을 보내시면 정리해서 뉴스그룹에 올리겠습니다"라고 적어서 보내라. 뉴스그룹과 메일링리스트를 내용상으로 같은 질문으로 채워지는 것을 막으려는 노력이다. 그러나 정리하겠다는 약속을 지켜야한다.

3.12. 문제에 대해서 명확하게 말하기

한계가 없는 질문은 시간을 소비하게만 하는 것으로 보일수 있다. 가장 훌륭한 답을 줄 수 있는 사람은 아마 가장 바쁜 사람들 중의 한 명일 것이다. (스스로 생계를 꾸려가고 있는 사람들을 겁니다.) 그런 사람들은 시간만 많이 소모하게 하는 명확하지 않은 질문을 싫어한다.

당신이 뭘 원하는지 명확할 수록 유용한 답을 얻을 수 있을 것이다. (코드를 보내고, 프로그램의 버전을 확인하고, 뭐든지..) 이렇게 하는 것은 답변하는 사람이 노력을 집중하게 하고, 그들이 들여야할 시간과 에너지의 최대치를 암묵적으로 제한해 줍니다. 좋은 일입니다.

전문가들의 세계를 이해하려면, 전문가들은 풍부하지만 그들의 시간은 희소한 자원으로 생각해 보십시오. 당신이 더 적은 시간을 암묵적으로 요구할 수 록 정말 훌륭하고 바쁜 사람들로 부터 답변얻을 가능성이 높습니다.

그래서 전문가가 처리하기 위해 들일 시간을 최소화하기 위해 질문에 가장자리를 치는 것은 유용합니다. 질문에 가장자리를 치는 것은 질문을 단순화하는 것과는 틀립니다. 예를 들어 "X윈도우에 관한 문서가 있는 곳을 알려주시겠습니까?"라는 것은 "X윈도우에 관해 알려주세요, 제발"이라는 질문보다 똑똑한 질문입니다. 만약에 동작하지 않는 코드를 가지고 있다면, 다른 사람들에게 고쳐달라고 하는 것 보다 뭐가 잘못된 건지 알려달라는 것이 보다 나은 방법입니다.

3.13. 학교숙제 질문하지 않기

해커들은 어떤 질문이 학교숙제인지 쉽게 알 수 있습니다. 대부분 그 들도 숙제를 했었기 때문입니다. 이 숙제들은 당신이 직접 해서 경험을 얻을 수 있도록 하기 위한 것입니다. 힌트를 달라고 하는 것은 괜찮지만 전체 소스를 요구해서는 안됩니다.

3.14. 의미 없는 말 빼기

질문을, 문법적으로 아무 의미없는 말 예를들어 "도와주실 수 있을까요", "여기 답이 있습니까" 와 같은 말로 끝내고 싶은 유혹에 넘어가지 마십시오. 첫 째, 당신이 질문을 잘 적었더라도 이런 쓸모없는 말은 넘칠 뿐입니다. 둘 째, 해커들은 이런 문장에 화를 내며 논리적으로 아무 문제없지만 경멸하는 말투의 답 "네, 당신은 도움 받을 수 있습니다."나 "아니요, 당신을 위해 도움을 줄 사람은 아무도 없습니다."와 같이 쓸 수 있습니다.

3.15. "긴급" 이라고 적지 마세요

이건 당신 문제 입니다. 우리 문제가 아닙니다. 급하다고 주장하는 것은 매우 비생산적입니다. 대부분의 해커가 그런 메세지를 무례하고 빠른 응답을 얻기위한 이기적인 시도로 여기고 그냥 지워버릴 겁니다.

3.16. 예의바르게 질문하기

예의바르게 질문하십시오. "부탁드립니다."나 "미리 감사드립니다."와 같은 말을 사용하십시오. 당신을 무료로 도와주는 사람들의 시간에 대해 감사하고 있다는 것을 표현하십시오.

솔직히 말하면 이 건 문법적으로 정확하고 명확한, 질문을 하는 것보다 중요하지 않습니다.(대체할 수도 없습니다.) 해커들은 대부분 좀 무뚝뚝하지만 기술적으로 날카로운 버그 리포트를 예의바른 애매함보다 좋아 합니다. (이게 이해가 잘 안된다면 우리가 뭔가 가르쳐 줄 질문을 보다 중요하게 생각한다는 것을 기억하십시오.)

그러나 당신이 여러가지 질문을 가지고 있다면 예의바른 것이 당신이 답변받을 확률을 높여 줄 겁니다.

3.17. 답변에 간단한 노트를 붙이십시오.

해결된 후에 도움을 준 사람들에게 노트를 보내십시오. 어떻게 해결되었고 그들에게 다시 한번 감사한다고 말하십시오. 만약에 문제가 메일링리스트나 뉴스그룹의 일반적인 관심을 끌만한 것이면 거기에 Followup을 올리십시오.

가장좋은 방법은, 원래의 질문이 있는 글타레에 'FIXED'나 'RESOLVED'라는 제목을 붙여서 올리는 것입니다. 빠르게 돌아가는 메일링리스트에서 사람들이 "Problem X - FIXED"라고 붙은 글을 보면 원래 질문을 읽어 보는 시간 조차도 아낄 수 있습니다. 그리고 그 시간을 다른 질문들을 해결하는데 쓸 수 있을 겁니다.

Followup이 길거나 자세할 필요는 없습니다. 간단히 "안녕하세요. 네트워크 케이블이 문제였습니다. 모두 감사드립니다. Bill로부터 "라고 적는 것이 아무 것도 없는 것 보다 좋습니다. 사실 짧고 같단한 요약이 기술적인 깊이가 없는 것이라면 논문보다 낳습니다. 문제를 해결한 행동을 적으십시오. 문제해결과정을 모두 적을 필요는 없습니다.

좀 깊이가 있는 문제였다면 문제해결 과정을 요약한 것을 올려두는 것이 좋습니다. 마지막으로 문제를 설명하고, 뭐가 해결책이었는지 적으십시오. 그리고 실수할 만한 것들을 적으십시오. 도움을 준사람들의 이름을 적으십시오. 이런 식으로 친구를 만들어 갈 수 있습니다.

당신이 해결한 방법을 정확히 알려주는 것이 예의바른 행동이고 정보를 널리 알리는 행동이자 다른사람이 메일링리스트를 찾거나 뉴스그룹을 검색하는 것을 도와줄 수 있습니다.

마지막으로 이런 Followup은 도와준 모든 사람들이 완결되었다는 느낌을 얻도록 도와줍니다. 당신이 개발자거나 해커가 아니라면 이런 느낌이 당신에게 도움을 준 guru나 전문가에게 매우 중요하다는 것을 믿어 주십시오. 해결되지 않는 공허함은 당황스런 것입니다. 해커들은 해결되는 것을 매우 보고 싶어 합니다. 이런 가려운 곳을 긁어 주는 것은 좋은 업을 쌓는 것이고 다음 번에 질문을 올렸을 때 매우 도움이 될 겁니다.

다른 사람이 똑같은 문제를 겪는 것을 어떻게 방지할지 생각해 보십시오. 스스로 문서나 FAQ에 패치를 하는 것이 도움이 될지 물어 보십시오. 답이 예라면 메인테이너에게 patch를 보내십시오.

해커들이게 이런 행동은 관례적인 예의바름보다 보다 더 중요합니다. 이렇게 함으로써 다른 사람과 잘 지낼 수 있다는, 매우 중요한 자산인 평판을 얻을 수 있습니다.

4. 답변을 어떻게 해석할 것인가?

4.1. RTFM과 STFM: 당신이 얼마나 얼간이 같은지 말하는 방법

이건 오래되고도 신성한 전통이다. 만약에 당신이 RTFM이라고 적혀진 답변을 받는 다면 그걸 적은 사람은 당신이 "Read The Fucking Manual"해야한다고 생각한다는 뜻이다. 그가 아마 맞을 것이다. 얼른 가서 매뉴얼을 읽어봐라.

RTFM은 좀 어린 친척이 있다. STFW라는 답변을 받으면 그걸 보낸사람은 당신이 "Search The Funcking Web"해야 한다고 생각한다는 것이다. 아마 맞을 것이고 Web을 찾아 봐라.

이런 답변을 적는 대부분의 사람들은 보통 당신이 필요로하는 정보가 들어있는 웹페이지나 매뉴얼을 가지고 있다. 아마도 이 사람들은 (a) 당신이 필요로 하는 정보가 찾기 쉽고 (b) 누군가가 직접 떠먹여 주는 것보다 스스로 찾아 보는 것이 더 많은 것을 배울 수 있을 거라고 믿고 있을 것이다.

이런 대답에 대해 공격적이 되서는 않된다. 해커들의 기준에서 보면 그는 당신을 그냥 무시하는 것이 아니라, 약간 퉁명스럽지만 당신을 존중하고 있는 것이다. 당신은 그의 할머니같은 친절함에 대해 감사해야 한다.

4.2. 이해할 수 없다면

답을 이해할 수 없으면, 바로 좀 더 알려달라고 요청하지 말아라. 답을 이해하기 위해 원래의 질문을 해결하기 위해 시도했던 것(매뉴얼, FAQ, 웹, 친구)를 이용해라. 그리고 아직도 좀 더 명확한 답이 필요하다면 당신이 배운 것이 뭔지 표현해라.

예를 들어, "zentry가 막혀 있다는 이야기 처럼 들리는 데요. 아마 그걸 지워야할 겁니다."라는 답을 이해할 수 없다면 나쁜 질문은 다음과 같다. "zentry가 뭔가요?" 좋은 질문은 다음과 같다. "예, man page를 찾아 보았는데 zentry라는 것이 -z와 -p 스위치에만 언급되어 있습니다. 그 중 어떤 것도 zentry를 clear하는 것에 대해 나와 있지 않네요. 이 옵션들이 당신이 말하는 것인가요? 아니면 제가 뭔가 빼먹은 것이 있나요?"

4.3. 무례함에 대처하기

해커들의 세계에서 무례하게 보이는 것의 대부분은 당신을 공격하기 위해서가 아니다. 당신을 공격하기 위해서 라기보다, 다른사람을 편안하게 해주는 것보다 문제를 해결하는데 더 관심이 많은, 직접적이고 목구멍에서 바로 나오는 화법의 부산물일 뿐이다.

무례하다고 느꼈다면, 진정하려고 해봐라. 다른 사람이 정말 무례하다면 메일링 리스트나 뉴스그룹의 경력자가 그/그녀를 꾸짖을 것이다. 만약이 이렇게 되지 않고 당신이 화를 낸다면, 당신이 화가난 그 사람은 해커 커뮤니티의 일반적인 행동규범에 어긋나지 않게 행동한 것이고, 당신이 잘못한 사람이라고 취급받고 있다는 것이다. 이렇게 되면 당신이 원하는 정보나 도움을 받는것이 많이 힘들어 질 것 이다.

반대로 당신이 자주 무례함이나 괜한 태도를 만난다면 위 단락의 반대로 그 사람을 아주 강하게 몰아부쳐도, 말발로 난도질해버리더라도 괜찮다. 그러나 이렇게 하기전에 서있는 곳을 확실하게 하기 바란다. 나쁜 버릇을 고치는 것과 요점없는 전쟁을 시작하는 것 사이의 차이는 매우 작다. 당신이 초보자이거나 아웃사이더라면 이런 실수를 피할 가능성은 거의 없다. 당신이 약간의 유희보다 정보를 원하고 있다면 이런 위험을 무릅쓰는 것 보다 키보드에서 손을 내려 놓는 것이 좋다.

어떤 사람들은 많은 해커들이 경미한 자폐증이나 Asperger's Syndrom이 있고, 분명히 다른 사람들과의 원활한 사회생활을 가능하게 해주는 뇌의 일부분이 없을 거라고 생각한다. 이건 사실일 수도 아닐 수도 있다. 당신이 해커가 아니라면 우리가 뇌가 손상된 사람이라고 생각하는 것이 우리의 기괴함을 이해하는데 도움이 될 지 도 모른다. 그렇게 생각해라. 우리는 상관하지 않는다. 우리는 뭐가 되던 우리 스스로를 좋아한다. 그리고 대부분 임상적으로 건강한 무신론을 가지고 있다.

다음 섹션에서 우리는 다른 주제에 대해 다룰 것이다. 당신이 잘 못 행동했을 때 보게 될 무례함에 대한 것이다.

5. 얼간이 처럼 행동하지 않기

당신은 아마 여기서 이야기하는 대로나 비슷하게 해커 커뮤니티에서 몇 번 실수를 저지를 것이다. 그리고 아마도 당신이 얼마나 엉뚱한 짓을 했는지 대중앞에서 성토당할 것이다.

만약에 이런 일이 일어나면 할 수 있는 가장 안좋은 방법은, 당신이 모욕당했다고 주장하고 사과를 요구하고, 소리치고, 고소하겠고 협박하고 그 사람의 상사에게 불명하고 화장실에서 물안내리고 나오고, 등등이다. 대신 여기에 당신이 해야할 일이 있다.

그냥 잊어라. 그게 정상이다. 사실 그건 건강하고 적당한 일이다.

공동체 규범은 스스로 유지되지는 않는다. 그건 대중앞에서 공개적으로 활발히 규범을 적용하는 사람들에 의해 유지된다. 개인메일로 비평이 되야 한다고 투덜거리지 마라. 그게 정상이다. 그리고 다른 사람이 자기주장이 틀렸다고 주장하거나 개인메일로 비평이 되야 한다고 투덜거리지 마라. 그게 정상이다. 그리고 당신의 주장이 틀렸다고 주장하거나 다른 의견을 가지고 있다는 사람에게 인격적으로 모욕당했다고 주장하는 것이 유용한 일은 아니다. 이건 패배자의 태도이다.

너무나도 예의만을 찾으려고 한, 잘못된 길로 들어선 해커게시판이 있었다. 그 곳에서는 "다른 사람을 돕지 않으려거든 아무 것도 말하지 말라"는 규칙이 있었고 다른 사람을 잘 못을 지적하는 것은 금지되었다. 이에 따라 도움이 될만한 구성원들이 다른 곳으로 떠나고 아무 의미없는 말장난 으로 전락해 버렸다. 그리고 기술 게시판으로 써 쓸모가 없어졌다.

과장되게 친절하기만 한 것과 쓸모있는 것중에 하나만 선택해라.

해커가 당신이 올바르지 못하다고 말 할 때, 다시는 그렇게 행동하지 말라고 할 때 그는 당신과 그의 공동체를 위해서 이야기하고 있는 것이다. 그의 입장에서는 당신을 그냥 무시하고 당신과 상관없이 생활하는 것이 훨씬 쉬울 것이다. 당신이 그에게 감사하는 마음이 들지 않는다고 하더라고 적어도 최소한의 자존심을 가지고 있다면 투덜거리지 마라. 당신이 과장되고 극

도록 예민한 영혼을 가지고 있고 뭐라도 된 듯한 망상을 가진 초보자라고 하더라도 깨지기 쉬운 인형처럼 취급받기를 바라지 마라.

6. Question Not to Ask

여기에 전형적인 멍청한 질문들이 있다. 그리고 해커들이 이 질문들을 본다면 뭐라고 생각할지 적어 두 었다.

Q: X에 관한 자료나 프로그램을 어디서 찾아 볼 수 있나요? Q: Y를 하려면 어떻게 X를 이용해야 하나요? Q: 어떻게 shell prompt를 설정할 수 있나요? Q: AcmeCorp문서를 Bass-o-matic파일 컨버터를 이용해서 TeX으로 변경할 수 있나요? Q: 제 {프로그램, 설정, SQL문}이 동작하지 않아요. Q: 제 Windows머신이 동작하지 않습니다. 도와주실 수 있나요? Q: 제 프로그램이 동작하지 않습니다. 제가 보기에는 X와 관련된 시스템에 문제가 있는 것 같습니다. Q: Linux나 X를 설치하는데 문제가 있습니다. 도와주실 수 있나요? Q: 어떻게 하면 root를 크랙하고/channel-ops 권한을 훔치고/다른 사람의 메일을 볼 수 있나요?

Q: X에 관한 자료나 프로그램을 어디서 찾아 볼 수 있나요? A: 여기에다가 질문을 올리느니 그 동안 웹서치해서 찾아 보겠다. 바보야. 아직 모두들 어떻게 Google을 이용하는지 모르나? Q: Y를 하려면 어떻게 X를 이용해야 하나요? A: 당신이 하고 싶은 것이 Y라면 적절하지 않을지도 모르는 도구를 예상하지 말고 질문해야 한다. 이런 종류의 질문은 그 사람이 X에 관해 무지하고 자기가 풀고 있는 문제 Y에 해새서 헷갈리고 있거나 그들의 특정한 상황에 너무 매달리고 있다는 것을 알 수 있게 한다. 이런 종류의 질문은 그들이 문제를 좀 더 정확하게 정의할 때 까지 무시하는 것이 제일이다. Q: 어떻게 shell prompt를 설정할 수 있나요? A: 이 질문을 할 정도로 똑똑하다면, RTFM, 매뉴얼을 읽어서 스스로 찾을 수 도 있을 겁니다. Q: AcmeCorp 문서를 Bass-o-matic 파일 컨버터를 이용해서 TeX으로 변경할 수 있습니까? A: 한번 해보세요. 한 번 해보면 답을 할 수 있고, 내 시간을 소비할 필요도 없을 겁니다. Q: 제 {프로그램, 설정, SQL문}이 동작하지 않아요. A: 이건 질문이 아닙니다. 당신이 진짜로 하고 싶은 질문이 뭔지 알기위해 스무고개 놀이를 하고 싶지않습니다. 전 다른 할 일이 있습니다. 이런 질문을 보면 전 보통 다음과 같이 행동합니다.

- 뭔가 더 할 말은 없나요?
- 그거 안 됐네요. 잘 해결되길 빕니다.
-
- 근데 이거 나랑 무슨 상관입니까?

Q: 제 Windows머신이 동작하지 않습니다. 도와주실 수 있나요? A: 예, 그 마이크로소프트에서 만든 쓰레기를 버리고 Linux나 BSD 같은 open-source 운영체계를 사용하세요. Q: 제 프로그램이 동작하지 않습니다. 제가 보기에는 X와 관련된 시스템에 문제가 있는 것 같습니다. A: 당신이 다른 수많은 사용자들이 수없이 이용하고 있는 System Call이나 Library에 있는 문제를 처음 발견했을 수 도 있지만 당신이 잘 못 생각하고 있을 가능성이 훨씬 높습니다. 특별한 주장은 특별한 증거를 필요로 합니다. 당신이 이런 주장을 하려면 실패사례에 대한 철저한 문서가 뒷 받침되어야 합니다. Q: Linux나 X를 설치하는데 문제가 있습니다. 도와주실 수 있나요? A: 아니요. 이걸 고치려면 당신 머신을 직접 손으로 만질 수 있어야 합니다. 지역 Linux유저그

룹에 도움을 요청하십시오. Q: 어떻게 하면 root를 크랙하고/channel-ops 권한을 훔치고/다른 사람의 메일을 볼 수 있나요? A: 당신은 그런 것을 하기를 원하는 하류민이고 해커에게 그걸 물어보는 바보다.

7. 좋은 질문과 나쁜 질문

마지막으로 어떻게 하면 똑똑하게 질문할 수 있는지 예를 들어 설명하겠습니다. 같은 문제에 대한 질문을 멍정한 방법과 똑똑한 방법으로 한 번 씩 보여드리겠습니다. Stupid : Foonly Flurbamatic에 대해 어디에서 찾을 수 있나요? 이 질문은 답으로 "STFW"을 원하고 있는 것 같습니다. Smart: Foonly Flurbamatic 2600에 관해 찾으려고 Google해 보았지만 쓸 만한 링크를 얻지 못했습니다. 누가 이 기계의 프로그래밍 정보를 어디서 찾을 수 있는지 아시나요? 이 사람은 이미 STFW 해 보았습니다. 그리고 정말로 해결할 할 만한 문제를 가진 것으로 보입니다.

Stupid : Project foo의 코드를 컴파일 할 수 없습니다. 뭐가 잘못된 걸까요? 그는 다른 사람이 뭔가 잘 못 했을 거라고 가정하고 있습니다. 그냥 무시하세요.

Smart: Project foo의 코드가 Nulix 6.2에서 컴파일되지 않습니다. FAQ를 읽어 보았지만 Nulix 6.2와 관련한 문제로는 보이지 않습니다. 여기 제 컴파일 transscript가 있습니다. 제가 뭔가 잘못 한 건가요? 그는 환경을 말했고, FAQ를 읽어 봤고, error메세지를 보여 주고 있습니다. 그리고 문제가 다른 사람이 잘못이라고 가정하고 있지도 않습니다. 이 사람은 좀 주의를 기울여줄 필요가 있습니다.

Stupid: 제 마더 보드에 뭔가 문제가 있는 것 같습니다. 누가 도와줄 수 있나요? 어떤 해커의 응답은 아마도 다음과 같을 겁니다. "그렇죠. 당신은 누가 트림시켜주고 귀저기 채워주는 것도 필요하죠?" 그리고 delete키 펀치가 뒤 따를 겁니다.

Stupid : S2464 마더보드에서 X, Y, Z를 시도해 보았습니다. 그게 동작하지 않아서 A, B, C도 시도해 보았습니다. 그리고 C를 시도했을 때 이상한 현상을 발견했습니다. 분명히 동작하고 있기는 한데 (XXX) 결과가 기대한 것과 같지 않습니다. 어떤 것이 Athlon MP 마더보드에서 grommicking을 일으키는 주요 원인인가요? 누가 이 문제의 범위를 좀 더 좁히기 위해 해볼만한 테스트에 관해 알고 있습니다. 이 사람은 반대로 대답할 만한 가치가 있어 보입니다. 그는 누가 하늘에서 답을 알려주는 것이 아니라 문제를 풀 수 있는 지성을 보여 주고 있습니다.

마지막 질문에서 "저에게 답을 주세요"와 "어떤 실험을 더 해보면 좋을지 알려주세요" 간의 미묘하지만 중요한 차이를 주목하십시오.

사실 마지만 질문은 리눅스 커널 메일링리스트에서 2002년 8월에 있었던 실제 상황과 유사합니다. 저(Eric)는 그 때 질문을 하던 사람이었습니다. 전 그 때 Tyan S2462마더보드에서 이상한 lockup현상을 겪고 있었습니다. 그 리스트에 멤버들이 제가 문제를 해결하는데 필요했던 중요한 정보를 제공해 주었습니다.

내가 했던 방법으로 질문을 함으로서 사람들에게 뭔가 가지고 놀 거리를 주었습니다. 전 질문을 사람들이 참여하기 쉽고 매력적으로 보이도록 만들었습니다. 전 동료들의 능력에 존경을 표시했고 이미 제가 지나간 골목길을 적어서 그들의 시간에도 존경을 표시했습니다.

문제가 해결된 다음, 도와준 모든 사람들에게 프로세스가 잘 작동한 것에 대해 감사를 표시했을 때 리눅스커널 메일링리스트의 한 사람은 'Eric이 list에서 명성을 가지고 있어서가 아니라 질문을 올바른 형식으로 물어서 잘 동작한 것 같다'고 말했습니다.

해커들은 어떤 면에서 매우 예의 없는 능력우선주의자들입니다. 난 그가 옳다고 생각합니다. 만약 제가 스폰지처럼 행동했다면 아마 내가 누구던 간에 질책받거나 무시당했을 겁니다. 이 모든 사건을 다른 사람을 위한 글로 써보라는 그의 권유가 이 가이드를 쓰게된 시초였습니다.

8. 답변을 얻을 수 없다면

답변을 얻을 수 없다면 그걸 아무 도움이 안될 개인적인 이유로 생각하지 마십시오. 가끔은 질문을 받은 그룹이 답을 모를 수 도 있습니다. 외견상으로는 차이가 없는 것처럼 보이지만 대답이 없는 것이 무시당하는 것과 동일한 것이라고 생각하지 마십시오.

일반적으로, 당신은 질문을 그냥 다시 올리는 것은 좋지않은 아이디어입니다. 요점없고, 화나게 만드는 일입니다.

초보자의 필요에 맞는 다른 곳이 있을 겁니다.

세상에는 스스로 소프트웨어를 만들어 본 적은 없지만 그 소프트웨어에 열광적인 온라인이나 지역 사용자 그룹이 있습니다. 이런 그룹들은 대개 다른 사람이나 새 사용자를 돕기 위해 만들어 집니다.

그리고 당신이 도움을 받기 위해 계약할 수 있는 많은 영리를 목적으로 하는, 크고 작은 회사들이 있습니다. (RedHat과 Linuxcare가 가장 유명한 회사들입니다. 그리고도 많습니다.) 당신이 약간의 도움을 받기위해 돈을 내야한다는 것이 놀라지 마십시오. 무엇보다 당신 차의 엔진이 고장나면 정비소로 가서 돈을 내고 수리할 것입니다. 소프트웨어가 톤을 내지 않은 것이라도 사용자지원도 역시 무료로 이루어 질 거라고 기대하지 마십시오.

리눅스같은 유명한 소프트웨어의 경우 1만 유저당 한 사람의 개발자가 있습니다. 한 사람이 만명의 유저로 부터 오는 요청을 모두 처리하는 것은 불가능합니다. 당신이 사용자지원을 받기위해 돈을 받아야 할 지라도 당신이 소프트웨어를 사는 것보다 훨씬 더 싸다는 것을 기억하십시오. (폐쇄소프트웨어를 위한 지원은 오픈소스보다 훨 씬 비쌀뿐 아니라 열악합니다.)

9. 도움을 줄 수 있는 답변하기

친절하십시오. 문제와 관련된 스트레스가 실제로는 그렇지 않다 할 지라도 사람들을 무례하고 멍청해보이도록 만들수 있습니다.

정확히 모르면 그렇다고 말하십시오. 틀렷지만 그럴 듯한 답은 아예 없는 것 보다 못합니다. 전문가 처럼 이야기하는 것이 재미있다고 해서 틀린 길을 가르쳐 주지는 마십시오. 겸손하고 정직해지십시오. 질문하는 사람에게나 당신의 동료들에게나 좋은 본보기가 되십시오.

도와줄 수 없거든 헷갈리게 하지 마십시오. 사용자의 설정을 망쳐버릴 수 도 있는 절차에 관해서 농담하지 마십시오. 어떤 불쌍한 사람은 농담을 진담으로 받아 들일 수 도 있습니다.

세부사항을 이끌어 내기위해 확인하는 질문을 하십시오. 당신이 이걸 잘 한다면 질문하는 사람은 뭔가 얻을 수 있을 겁니다. 잘못된 질문을 좋은 질문으로 바꿔 보려고 하십시오. 우리 모두 처음에는 초보였습니다.

답변을 하면서 그냥 "RTFM"이라고 말하는 것이 정말 게으른 사람한테 답할 때는 정당화될 수 있지만 문서를 찾을 수 있도록 해 주는 것(구글에서 사용할 수 있는 키워드 권유해 주는 것 뿐이라도)이 더 낳습니다.

답변을 할 생각이라면 뭔가를 주십시오. 다른 사람이 잘 못된 툴이나 접근법을 사용하고 있을 때 서투른 Workaround을 권유하지 마십시오. 좋은 툴을 추천하고 질문을 다시 정리하도록 도와주십시오.

공동체가 질문으로 부터 배울 수 있도록 도와주십시오. 좋은 질문을 발견했다면 "어떻게 하면 다른 사람이 같은 질문을 반복하지 않도록 관련된 문서나 FAQ를 업데이트 할 수 있을지 물어 보십시오." 그리고 문서 관리자에게 patch를 보내십시오.

질문에 답변하기 위해서 조사를 했다면, 당신 머리에서 바로 꺼내서 적은 것 처럼 행동하지 말고 어떻게 찾았는지 설명하십시오. 좋은 답변을 하나 하는 것은 배고픈 사람에게 한 끼 식사를 주는 것이 아니라 그들에게 평생동안의 양식을 키울 수 있도록 가르치는 것과 같습니다.

10. 관련 자료

어떻게 PC, Unix, Internet이 작동되는지 궁금하다면, The Unix and Internet Fundamentals HOWTO (http://en.tldp.org/HOWTO/Unix-and-Internet-Fundamentals-HOWTO/)를 참조하십시오.

당신이 소프트웨어를 배포하거나 소프트웨어의 패치를 만들고 있을 때 Software Release Pratice HOWTO (http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/index.html)에 있는 가이드라인을 따르도록 하십시오.

11. FAQ 리스트 관리자나 웹마스터를 위한 특별한 노트

많은 웹사이트, 뉴스그룹, 그리고 온라인 포럼에서 이 how-to를 초보자를 위한 가이드로서 링크시키고 있습니다. 작가는 당신이 이렇게 하는 것을 기쁘게 생각합니다. 그러나 이 링크의 바로 옆에 볼드체로 우리는 당신 프로젝트의 헬프데스크가 아니라고 추가 시켜 주십시오. 우리는 반대로 생각하는 유저들로 부터 너무나 많은 질문을 받고 있습니다.