

Como fazer perguntas sobre programação na internet? (e aumentar 5x as chances de ter uma resposta)



Texto original em [inglês](#)

Nota sobre o texto: O autor original é um pouco ‘rude’ com alguns termos, tente ignorar esse fato e se atente aos ensinamentos principais (que são extremamente úteis), boa leitura!

Introdução

No mundo dos *programadores*, o tipo de respostas que você obtém para suas questões técnicas depende bastante da forma que você faz as perguntas bem como da dificuldade de desenvolver a resposta. Este guia irá ensinar você como fazer perguntas de forma a obter uma resposta satisfatória.

A primeira coisa a entender é que os *programadores* gostam de problemas difíceis e questões boas e provocantes sobre eles. Se não fosse assim, não estaríamos aqui. Se você nos der uma questão interessante para mastigar seremos gratos; boas questões são um estímulo e um presente. Boas perguntas nos ajudam a desenvolver nosso entendimento, e geralmente revelam problemas que poderíamos não ter percebido ou pensado de outra forma. Entre os *programadores*, “Boa pergunta!” é um forte e sincero cumprimento.

Apesar disto, os *programadores* tem uma reputação de abordar questões simples com o que se parece com hostilidade ou arrogância. Às vezes parece que somos hostis aos novatos e ignorantes. Mas isto não é verdade.

O que nós somos, sem qualquer desculpa, é hostis a pessoas que parecem não estar querendo pensar ou fazer seu próprio trabalho de casa antes de fazer perguntas. Pessoas assim são como precipícios — eles pegam sem dar nada em troca, eles desperdiçam um tempo que poderíamos ter usado em outra questão mais interessante e com outra pessoa que valha mais a pena responder. Chamamos este tipo de pessoa de ”*losers*” (e por razões históricas às vezes escrevemos ”*lusers*”).

Somos, em grande parte, voluntários. Tomamos tempo de nossas vidas ocupadas, e às vezes somos sobrecarregados por ela. Por isto filtramos as perguntas de forma rude. Em particular, jogamos fora questões de pessoas que pareçam ser *losers* para gastar nosso tempo de responder perguntas de forma mais eficiente, com vencedores.

Você não quer ser um dos *losers*. Você não quer parecer com um, também. A melhor forma de conseguir respostas rápidas é fazer as perguntas como um vencedor — perguntar como uma pessoa esperta, confiante, e com dicas, que parece precisar de ajuda apenas em um problema particular.

Antes de Perguntar

Antes de fazer uma pergunta técnica por email, ou em um *newsgroup*, ou em um *chat-board* de um *website*, faça o seguinte:

1. Tente encontrar uma resposta lendo o manual
2. Tente encontrar uma resposta lendo o FAQ
3. Tente encontrar uma resposta pesquisando a Web
4. Tente encontrar uma resposta perguntando a algum amigo mais preparado

Quando fizer sua pergunta, mostre que você fez estas coisas primeiro, isto ajuda a estabelecer que você não está sendo uma esponja preguiçosa e desperdiçando o tempo de outras pessoas. Melhor ainda, mostre que você aprendeu alguma coisa tendo feito estas coisas. Gostamos de responder perguntas de pessoas que tenham demonstrado que podem aprender das respostas.

Prepare sua questão. Pense adiante. Perguntas que pareçam precipitadas recebem respostas precipitadas, ou nenhuma resposta. Quanto mais você demonstrar que dedicou pensamentos e

esforços em resolver seu problema antes de pedir por ajuda, mais provável é que você vá conseguir ajuda.

Cuidado para não fazer a pergunta errada. Se você fizer perguntas baseadas em pressupostos falsos, *J. Random* provavelmente vai responder com uma resposta literalmente inútil ao mesmo tempo que pensa “Pergunta estúpida...”, e com a esperança de que a experiência de ganhar o que você pediu em vez do que você precisa vá lhe ensinar uma lição.

Por outro lado, tornar claro que você está apto e pronto a ajudar no processo de desenvolver a solução é um bom início. “Alguém pode me providenciar uma dica?”, “O que falta no meu exemplo?” e “Existe algum site que eu deveria ter procurado?” são perguntas que provavelmente vão ter respostas, em vez de “Por favor poste o procedimento que devo usar.” por que neste caso você está tornando claro que você não está disposto a completar o processo se alguém puder simplesmente apontar a direção certa para você.

Nunca assuma que você tem direito a uma resposta. Você não tem. Você vai conquistar uma resposta se puder conquistar a mesma, fazendo uma pergunta que seja substancial, interessante, estimulante — uma que implicitamente contribua à experiência da comunidade em vez de meramente exigir passivamente por conhecimentos dos outros.

Quando Perguntar

Escolha seu fórum cuidadosamente

Seja perceptivo ao escolher onde fazer sua pergunta. Você tem mais chances de ser ignorado, ou chamado de *loser*, se:

- postar sua pergunta em um fórum no qual ela seja *off topic*
- postar uma pergunta muito elementar onde sejam esperadas questões técnicas avançadas, ou vice-versa
- fazer *cross-post* para muitos *newsgroups* diferentes

Os *programadores* detonam com perguntas que sejam feitas inapropriadamente para tentar proteger seus canais de comunicação de serem afogados em irrelevâncias. Você não vai querer que isto aconteça com você.

Escreva em linguagem clara, gramática e ortograficamente correta

Sabemos por experiência que pessoas que são escritores descuidados e superficiais são também geralmente pensadores descuidados e superficiais (com experiência suficiente para apostar nisto). Responder perguntas para pensadores descuidados e superficiais não traz nenhuma compensação; nós preferimos gastar nosso tempo com outras coisas.

Portanto, expressar sua dúvida de forma clara e correta é importante. Se você não pode se incomodar com isto, nós não podemos nos incomodar em prestar atenção. Gaste algum esforço extra para polir sua linguagem. Você não precisa ser cerimonioso ou formal — ao contrário, a cultura dos *programadores* valoriza a linguagem informal, cheia de gírias e humor, usada com precisão. Mas ela precisa ser precisa; tem que haver algum indício que você está pensando e prestando atenção.

Escreva corretamente. Não confunda *”its”* com *”it’s”*, ou *”loose”* com *”lose”*. Não escreva TUDO EM MAIÚSCULAS, isto é lido como um grito e considerado rude. Se você escreve como um bobo semi-analfabeto, provavelmente você será ignorado. Escrever como um *l33t script kiddie hax0r* é o beijo da morte e garante que você não receberá nada a não ser um silêncio pétreo (ou, na melhor das hipóteses, ajuda carregada de desprezo e sarcasmo).

Se você está fazendo perguntas em um fórum que não usa sua linguagem nativa, você vai ter uma quantia limitada de tolerância a erros ortográficos e gramaticais — mas nenhuma tolerância a preguiça mental. Além disso, a menos que você saiba a linguagem de quem vai te responder, escreva em inglês. *programadores* ocupados tendem a simplesmente ignorar questões em linguagens que eles não entendem, e o inglês é a linguagem de trabalho na Internet. Escrevendo as perguntas em inglês você minimiza as chances de que as mesmas sejam descartadas sem ser lidas.

Envie questões em formatos que sejam fáceis de entender

Se você faz questões artificialmente difíceis de ler, provavelmente será ignorado em favor de alguém que não faz isto. Portanto:

- Envie email em texto puro, não HTML
- Não envie emails em que os parágrafos inteiros sejam linhas únicas (isto torna mais difícil responder a somente uma parte da mensagem)
- Nunca, jamais espere que os *programadores* possam ler formatos de documentos proprietários fechados como do *Microsoft Word*. A maioria dos *programadores* reagem a estes formatos da

mesma forma que você reagiria a um monte fedorento de esterco de porco jogado na tua porta.

- Se você estiver enviando email em uma máquina Windows, retire a funcionalidade estúpida de “*Smart Quotes*” da Microsoft. Fazendo isto você evita pingar caracteres inúteis em seu email

Utilize linhas de assunto com significado e específicas

Em listas de correio ou grupos de notícias, a linha de assunto é sua oportunidade de ouro para atrair a atenção de especialistas qualificados em cerca de 50 caracteres ou menos. Não desperdice esta oportunidade com murmúrios e balbúcias, como “Por favor me ajudem” (ou então “POR FAVOR ME AJUDEM!!!!”). Não tente nos impressionar com a profundidade de tua angústia; ao invés disto, use o espaço para uma descrição super-concisa do problema.

Estúpida:

- SOCORRO! O vídeo não funciona no meu laptop!

Inteligente:

- XFree86 4.1 mostra cursor errado, chipset de vídeo Fooware MV1005

Seja preciso e informativo sobre o seu problema

- Descreva os sintomas do seu problema ou *bug* cuidadosa e claramente.
- Descreva o ambiente em que o mesmo ocorre (máquina, SO, aplicação, o que for)
- Descreva a pesquisa que você fez para experimenta resolver e entender o problema antes de ter feito a pergunta
- Descreva os passos de diagnóstico que você tomou para resolver o problema por conta própria antes de ter feito a pergunta
- Descreva qualquer alteração recente no seu computador ou configuração de software que possa ser relevante

Faça o melhor que puder para antecipar as perguntas que um *programador* possa fazer, e para responder as mesmas previamente no seu pedido de ajuda.

Simon Tatham escreveu um excelente documento chamado *How to Report Bugs Effectively*. Eu recomendo que você leia o mesmo.

Descreva os Sintomas, não suas suspeitas

Não há utilidade em dizer aos *programadores* o que você pensa que está causando seu problema (se suas teorias de diagnósticos fossem tão boas, você estaria pedindo ajuda?). Assim, certifique-se de informar os sintomas exatos do que está dando errado, em vez de suas interpretações e teorias. Deixe-os fazer a interpretação e diagnóstico.

Estúpida:

- Estou tendo erros SIG11 quando o kernel compila, e suspeito que uma trilha da motherboard está interrompida. Qual a melhor forma de verificar isto?

Inteligente:

- Meu computador feito em casa K6/233 com mobo FIC-PA2007 (chipset VIA Apollo VP2) com 256MB SDRAM Corsair PC133 começa a dar erros SIG11 frequentes cerca de 20 minutos após ser ligada, durante compilações de kernel, mas nunca nos primeiros 20 minutos. Reinicializar não reinicia o relógio, mas desligar à noite sim. Trocar toda a RAM não ajudou. O trecho relevante de um log de uma sessão de compilação típica segue abaixo.

Descreva os sintomas dos problemas em ordem cronológica

As pistas mais úteis para descobrir o que está errado geralmente estão no que ocorreu um pouco antes. Por isto, descreva precisamente o que você fez, e o que a máquina fez, até a ocorrência do problema. No caso de processos de linha de comando, ter um log da sessão (por exemplo usando o utilitário script) e copiando as vinte e tantas linhas relevantes é bastante útil.

Se o programa que está dando problemas possui opções de diagnóstico (tipo uma opção -v para modo verboso), tente pensar com carinho em usar algumas opções que possam acrescentar alguma informação de depuração úteis ao texto transcrito.

Se sua mensagem for muito longa (mais que cerca de quatro parágrafos), pode ser útil descrever o problema de forma sucinta no início, e então seguir com a história cronológica. Desta forma, os *programadores* saberão o que procurar quando estiverem lendo sua crônica.

Não peça para ser respondido via email privado

Os *programadores* acreditam que a solução de problemas deve ser um processo público e transparente, durante o qual uma primeira tentativa de resposta pode e deve ser corrigida por alguém com mais conhecimentos que tenha percebido que a mesma está incompleta ou incorreta. Da mesma forma, eles obtêm alguma recompensa por ter respondido por serem vistos como competentes e conhecedores por seus iguais.

Quando você pede uma resposta em private, você está interrompendo tanto o processo quanto o prêmio. Não faça isto. É escolha de quem responde se a resposta deve seguir em *private* — e se isto acontece, usualmente é por que ele pensa que a questão é muito óbvia ou mal-formada para interessar a outros.

Há uma exceção limitada a esta regra. Se você pensa que a questão é tal que você provavelmente vai receber muitas respostas que são muito semelhantes, então as palavras mágicas são “mandem por email que eu resumo as respostas para o grupo”. É cortês tentar proteger a lista de mail e o *newsgroup* de uma inundação de *posts* idênticos — mas você deve cumprir a promessa de mandar o resumo.

Corte fora perguntas inúteis

Resista a tentação de fechar seu pedido de ajuda com questões semanticamente nulas como “Alguém pode me ajudar?” ou “Há alguma resposta?” Primeiro, se você fez sua parte da descrição do problema de forma competente, este tipo de pergunta é, na melhor hipótese, supérflua. Em segundo, pelo que a precede, esta questão beira a perturbação, convidando a respostas inúteis mas logicamente impecáveis como “Sim” ou “Não”. Cortesia não machuca, e, às vezes, ajuda

Seja cortês. Use “Por favor”, e “Agradeço antecipadamente”. Torne claro que você dá valor ao tempo que as pessoas gastam tentando te ajudar gratuitamente.

Para ser honesto, isto não é tão importante quanto (e não é substituto de) ser gramaticalmente correto, claro, preciso e descritivo, evitando formatos proprietários, etc. Os *programadores* em geral preferem um relatório de bugs um tanto brusco mas tecnicamente correto, que uma mensagem polidamente vaga (se isto te surpreende, lembre que damos valor a uma questão pelo que ela nos ensina).

Entretanto, se você tem os fatos técnicos citados coerentemente, ser polido aumenta as chances de conseguir uma resposta útil.

Poste uma mensagem curta apontando a solução encontrada

Envie uma nota após o problema ter sido solucionado para todos que te ajudaram; deixe-os saberem como ficou e agradeça novamente pela ajuda. Se o problema atraiu o interesse geral na lista de email ou *newsgroup*, é apropriado postar um retorno no mesmo.

O *follow-up* não precisa ser longo e detalhado, um simples “Feito - era um cabo de rede com problemas! Obrigado a todos - Bill” é melhor que nada. De fato, um resumo curto e simples é melhor que uma longa dissertação a menos que a resposta tenha um real valor técnico.

Além de ser cortês e informativo, este tipo de *follow-up* ajuda todo mundo que deu assistência a ter uma sensação de fechamento do problema. Se você não é um *techie* ou *programador*, acredite em nós que este sentimento é muito importante para os gurus e especialistas que você pediu ajuda. Narrativas de problemas que terminam em um vazio não resolvido são coisas frustrantes; os *programadores* sentem uma comichão por ver os mesmos resolvidos. O bom karma que você ganha coçando esta comichão é que será muito mais útil postar uma pergunta na próxima vez que você tiver algum problema.

Como Interpretar as Respostas

RTFM e STFW: Como Saber que Você “se Ferrou”

Há uma tradição antiga e respeitada: se você recebe uma resposta escrita “RTFM”, a pessoa que enviou pensa que você deve Ler O Diabo do Manual (“*Read The Fucking Manual*” - a expressão é mais “colorida” no original). E ele provavelmente está certo sobre isto. Vá ler o manual.

RTFM tem um parente mais jovem. Se você receber uma resposta “STFW”, a pessoa que mandou ela pensa que você deveria Procurar no Diabo da Web (“*Search The Fucking Web*”). E ele provavelmente está certo sobre isto. Vá fazer uma pesquisa.

Geralmente, a pessoa que envia um destas respostas tem o manual ou a página com a informação que você precisa aberta, e está olhando para ela enquanto escreve. Estas respostas significam que (a) a informação a ler é fácil de encontrar, e (b) você vai aprender mais se procurar por conta própria do que se você ganhar a mesma de mão beijada.

Você não deve se sentir ofendido por estas respostas - pelos padrões dos *programadores*, ele está te mostrando um pouco de respeito simplesmente por não ter te ignorado. Você deveria, ao invés, ficar agradecido pela sua bondade de vovózinha.

Se você não entendeu...

Se você não entendeu a resposta, não devolva imediatamente um pedido de esclarecimento. Use as mesmas ferramentas que você usou para encontrar uma resposta para sua pergunta original (manuais, FAQ, a Web, um amigo que saiba mais) para entender a resposta. Se você precisar mais esclarecimentos, mostre o que você aprendeu.

Por exemplo, suponha que eu tenha dito: “Parece que você tem uma zentry presa, você precisa limpar ela.” Então:

Eis uma pergunta ruim: “O que é uma zentry?”

Eis uma pergunta boa: “OK, eu li a página man e as zentries são mencionadas somente sob as opções -z e -p. Nenhuma delas esclarece nada sobre zentries. É só isto ou eu estou deixando passar alguma coisa aqui?”

Não Reaja como um Perdedor

Eventualmente você vai se dar mal algumas vezes em fóruns da comunidade de *programadores* — em formas detalhadas neste artigo, ou parecidas. E vai ser dito para você exatamente onde você se deu mal, possivelmente com termos coloridos. Em público.

Quando isto acontecer, a pior coisa que você pode fazer é se queixar da experiência, alegar ter sido verbalmente atacado, exigir que se desculpem, gritar, prender a respiração, ameaçar com processos legais, reclamar com os chefes das pessoas, deixar a tampa do vaso levantada, etc. Ao invés disso, aqui está o que você deve fazer:

- Supere isto. É normal. De fato, é saudável e apropriado.

Os padrões da comunidade não se mantêm por si mesmos: eles são mantidos por pessoas que ativamente os aplicam, visivelmente, em público. Não se queixe que toda crítica deveria ir por email privado: não é assim que isto funciona. Nem é útil insistir que você foi insultado pessoalmente quanto alguém comentar que algumas de suas alegações estão erradas, ou que ele vê de forma diferente. Estas são atitudes de perdedores.

Tem havido forums de *programadores* onde, por conta de algum senso de hiper-cortesia mal orientado, os participantes são banidos por postar qualquer mensagem apontando erros de outros, e ouvem um “Não diga nada se você não está a fim de ajudar o usuário”. A resultantes partida de

participantes que sabiam alguma coisa para outros lugares causou a queda do grupo para uma balbúrdia sem sentido e os tornou inúteis como fóruns técnicos.

Exageradamente “amigável” (daquele jeito) ou útil: escolha.

Lembre-se: quando aquele *programador* te disser que você errou, e (não importa o quão rude) te diz para não fazer isto novamente, ele está agindo por preocupação (1) por ti e (2) pela comunidade. Seria mais fácil para ele ignorar você e filtrar você fora da vida dele. Se você não consegue ser grato, pelo menos tenha um pouco de dignidade, não chore, e não espere ser tratado como uma boneca frágil só por que você é um novato com uma alma teatralmente hipersensível e ilusões de merecimento.

Perguntas Que Não Se Faz

Aqui estão algumas perguntas estúpidas clássicas, e o que os *programadores* estão pensando quando não respondem às mesmas.

P: Onde eu encontro o programa X?

R: No mesmo lugar em que eu encontrei ele, bobão — na outra ponta de uma pesquisa na web. D’eus, ninguém sabe usar o Google ainda?

P: Estou tendo problemas com minha máquina Windows. Você pode me ajudar?

R: Sim. Jogue fora aquele lixo da Microsoft e instale o Linux.

P: Estou tendo problemas para instalar o Linux ou o X. Você pode me ajudar?

R: Não. Eu preciso ter acesso à tua máquina para resolver este problema. Vá pedir ajuda no teu grupo de usuários Linux local.

P: Como eu posso crackear o root/roubar privilégios de ops de canais/ler os emails de outra pessoa?

R: Você é uma forma de vida inferior por querer fazer este tipo de coisa e um debilóide por pedir que um *programador* te ajude.

Boas e Más Perguntas

Por fim, vou ilustrar como fazer perguntas de uma forma inteligente por exemplo. Pares de questões sobre o mesmo problema, uma pergunta feita de forma estúpida, e outra de forma inteligente.

Estúpida: Onde eu posso encontrar alguma informação sobre o Foonly Flubarmatic?

- Esta pergunta está implorando por um “STFW” como resposta.

Inteligente: Eu usei o Google para pesquisar por “Foonly Flubarmatic 2600” na Web, mas não consegui nenhum link útil. Alguém sabe onde eu posso encontrar alguma informação sobre a programação deste dispositivo?

- Este usuário já SFTW ou, e parece que ele tem um problema real.

Estúpida: Não consigo compilar o projeto foo. Por quê ele está com erros?

- Ele está assumindo que todo mundo está errado. Arrogância da parte dele.

Inteligente: O código do projeto foo não compila no Nulix versão 6.2. Eu li o FAQ, mas não tem nada sobre problemas relacionados ao Nulix. Aqui tem uma transcrição de minhas tentativas de compilar o mesmo, é algo que eu fiz?

- Ele especificou o ambiente, ele leu o FAQ, ele está mostrando o erro, e ele não está assumindo que seu problema é por causa do erro de outra pessoa. Este cara merece alguma atenção.

Estúpida: Estou tendo problemas com minha motherboard. Alguém pode me ajudar?

- A resposta de *J. Random* é provavelmente “Certo. Precisa arrotar e trocar as fraldas também?” seguido pelo pressionar da tecla delete.

Inteligente: Eu tentei X, Y e Z na motherboard S2464. Quando isto não funcionou, eu tentei A, B e C. Note o sintoma curioso quando tentei C. Obviamente o florbish está gromicando, mas os resultados não são os esperados. Quais são as causas usuais de gromicamento em motherboards MP? Alguém tem idéias de testes que eu possa fazer para descobrir o problema?

- Esta pessoa, por outro lado, parece que vale a pena responder. Ela exibiu inteligência para resolver problemas ao invés de ficar esperando que uma solução caísse do céu. Na última questão, note a sutil mas importante diferença entre “Me dê uma resposta” e “por favor me ajude a descobrir que diagnósticos adicionais posso tentar para descobrir o problema”. De fato, a forma da última questão é baseada em um incidente real que aconteceu em agosto de 2001 na lista de correio linux-kernel. Eu era quem estava fazendo a pergunta aquela vez. Eu estava observando misteriosos travamentos em uma motherboard Tyan S2464. Os membros da lista forneceram as informações que eu precisava para resolver o problema. Fazendo a pergunta do jeito que eu fiz, eu dei algo para que as pessoas mastigassem, eu tornei fácil e atrativo o envolvimento. Eu demonstrei respeito pela habilidade de meus colegas e convidei eles a me consultarem como a um igual. Eu demonstrei respeito pelo tempo deles informando quais as ruas escuras que eu já havia tomado. Mais tarde, quando agradei a todos e destaquei como o processo funcionou bem, um membro da lkml apontou que ele pensava que o processo funcionou não por que eu era um “nome” naquela lista, mas por que eu fiz a pergunta de forma apropriada.

Nós, *programadores*, estamos, de certa forma, em uma rude meritocracia: estou certo que ele está correto, e se eu tivesse agido como uma esponja eu teria recebido *flames* ou seria ignorado, não importando quem eu era. Sua sugestão que eu escrevesse sobre o incidente como uma instrução para outros foi o que me inspirou a compor este guia.

Considerações finais

Agradecimentos: texto original foi escrito por *Por: Eric Raymond* em

inglês: <http://www.catb.org/esr/faqs/smart-questions.html> Tradução feita por: [César A. K. Grossmann](#)

Evelyn Mitchell contribuiu com alguns exemplos de perguntas estúpidas e inspirou a seção “Como responder perguntas de forma útil”. Mikhail Ramendik contribuiu com valiosas sugestões de aprimoramento deste guia.

Como responder perguntas de forma útil

Seja gentil. Stress relacionado a problemas podem fazer as pessoas parecerem rudes ou estúpidas mesmo que elas não sejam.

Responda à uma primeira falta offline. Não há necessidade de humilhar publicamente alguém que tenha honestamente cometido um erro. Um novato de verdade pode não saber como pesquisar o histórico da lista ou onde o FAQ está localizado.

Se você não tem certeza, deixe isto claro! Uma resposta errada mas cheia de autoridade é pior do que nenhuma resposta. Não indique a ninguém um caminho errado só porque é divertido falar como um expert. Seja humilde e honesto; dê um bom exemplo para os seus companheiros.

Se você não pode ajudar, não atrapalhe! Não faça piadas sobre procedimentos que podem bagunçar o sistema do usuário — o pobre coitado pode interpretar isto como instruções sérias.

Faça perguntas para levantar mais informações. Se você for bom nisso o usuário irá aprender alguma coisa — e talvez você também. Tente transformar uma pergunta ruim em uma pergunta boa; lembre-se que todos nós fomos novatos um dia.

Enquanto mandar um RTFM é algumas vezes justificável quando você está respondendo uma pergunta de um preguiçoso, uma indicação do local exato da documentação (ou a frase correta para pesquisar no Google) é bem melhor.

Se você for responder uma pergunta, responda direito. Não sugira gambiarras para fazer funcionar a ferramenta errada. Sugira a ferramenta certa. Refaça a pergunta.

Ajude sua comunidade a aprender com a pergunta. Quando você responder a uma pergunta pense: “Como a documentação ou o FAQ podem ser melhorados para que ninguém precise perguntar isso novamente?” E então envie um patch ao mantenedor da documentação.

Se você teve que pesquisar para encontrar uma resposta, demonstre como você encontrou a resposta ao invés de fazer parecer que você a tirou de dentro de uma cartola. Responder a uma boa pergunta é como dar um prato de comida a um faminto. Mas ensiná-lo a plantar o próprio alimento é lhe garantir comida para o resto da vida. == Fontes relacionadas ==

Se você precisa de instruções sobre o básico dos computadores, sistemas Unix e Internet consulte o HOWTO Fundamentos do Unix e da Internet. (em inglês)

Quando você distribuir um software ou escrever patches para um, tente seguir o guia HOWTO Software Release Practice. (em inglês)

@PAULOLUAN
REATIVA

CLIQUE AQUI E ME SIGA NO

INSTAGRAM



2021 [Reativa Tecnologia](#)