

Sistemas de Software Livre

Introdução

Prof. Jacson Rodrigues Correia da Silva
Departamento de Computação – UFES

Este documento foi criado com textos já existentes na Internet.
Referências no fim do documento.

Sumário

1. Breve Histórico.....	1
2. Software Livre.....	3
3. Copyleft.....	8
Referências.....	12

1. Breve Histórico

Da década de 1950 até o início dos anos de 1970, era normal para os usuários de computador ter as liberdades de software associadas com software gratuito. De forma comum, os softwares eram compartilhados por indivíduos que usavam computadores e pelos fabricantes de hardware que saudavam o fato de que as pessoas estavam fazendo software que fazia seu hardware ser útil. Organizações de usuários e fornecedores, por exemplo, SHARE, foram criados para facilitar o intercâmbio de softwares.

Porém, no início da década de 1970 o quadro mudou: os custos de software foram aumentando drasticamente, uma indústria de software que crescia estava competindo com os softwares embarcados criados pelos fabricantes de hardware (que eram grátis, pois seu custo era incluído no custo do hardware), máquinas alugadas necessitavam de suporte de software, mas não tinham rendimentos com software, e alguns clientes capazes de entender melhor suas próprias necessidades não desejavam que o custo do software fosse acoplado ao hardware.

Em 17 de janeiro de 1969, o governo dos Estados Unidos da América definiu que o software embarcado era anticompetitivo. Enquanto alguns softwares continuavam grátis, a quantidade de software produzido principalmente para a venda era crescente. Entre a década de 1970 e início de 1980, a indústria de software começava a usar medidas técnicas (como somente distribuir cópias binárias de programas de computador) para evitar que os usuários de computador fossem capazes de estudar ou adaptar o software como quisessem. Em 1980, a lei de copyright foi estendida a programas de computador.

Em 1983, Richard Stallman, um dos autores originais do programa popular Emacs e um membro de longa data da comunidade hacker no Laboratório de Inteligência Artificial do MIT, anunciou o projeto GNU, cujo objetivo era produzir um sistema operacional compatível com Unix completamente não-proprietário, dizendo que ele havia se tornado frustrado com a mudança do clima que cercou o mundo da informática e seus usuários.

Na sua declaração inicial sobre o projeto e sua finalidade, ele citou especificamente e de forma motivada a sua oposição ao ser perguntado sobre concordar com os acordos de não divulgação de código fonte e de licenças restritivas que proibiam seu livre compartilhamento, considerados lucrativos pelas empresas e por ele, como uma proibição diretamente contrária a ética hacker tradicional. O desenvolvimento do sistema operacional GNU foi iniciado em janeiro de 1984 e a Free Software Foundation (FSF) foi fundada em outubro de 1985. Ele desenvolveu uma

definição de software livre e do conceito de "copyleft", destinado a garantir a liberdade do software para todos.

Em meados de 1990, outro software importante na história do software livre foi iniciado por Linus Torvalds: o kernel Linux. Seu código fonte foi lançado em 1991, podendo ser livremente modificado. Sua primeira licença foi de software proprietário. Porém, a versão 0.12, em fevereiro de 1992, foi re-licenciada sob a GNU GPL. Muito parecido com o Unix, o kernel de Torvalds atraiu a atenção de programadores voluntários. Também nessa época surgiram outros projetos importantes, como o OpenBSD, em 1995, sob licença BSD, e o Apache HTTP Server, comumente referido como Apache, que foi lançado sob a licença Apache 1.0.

2. Software Livre

A definição de software livre (*free software*) apresenta os critérios utilizados para definir se um programa de computador em particular se qualifica como software livre ou não. O termo “free software” foi criado por Richard Stallman, em 1985, quando lançou o projeto GNU¹ e a Free Software Foundation. De tempos em tempos essa definição é revisada, buscando deixá-la sem ambiguidades e/ou resolvendo questões mais sutis.

Por software livre devemos entender aquele software que respeita a liberdade e o senso da comunidade dos usuários. A grosso modo, os usuários possuem a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o software e por isso, software livre é uma questão de liberdade, não de preço. Para entender o conceito, pense em liberdade de expressão, não em cerveja grátis.

Com essa liberdade, os usuários (tanto individualmente quanto coletivamente) controlam o programa e o que ele faz por eles. Quando os usuários não controlam o programa, o programa controla os usuários, pois quem tem que se adequar é o usuário e não o programa. Dessa forma, o desenvolvedor controla o programa e, por meio dele, controla os usuários. Esse programa não-livre e proprietário é, portanto, um instrumento de poder injusto.

Um programa é software livre se os usuários possuem as quatro liberdades essenciais:

0. A liberdade de executar o programa como você desejar, para qualquer propósito;
1. A liberdade de estudar como o programa funciona, e adaptá-lo às suas necessidades; Para tanto, acesso ao código-fonte é um pré-requisito.
2. A liberdade de redistribuir cópias de modo que você possa ajudar ao próximo;
3. A liberdade de distribuir cópias de suas versões modificadas a outros; Desta forma, você pode dar a toda comunidade a chance de beneficiar de suas mudanças. Para tanto, acesso ao código-fonte também é um pré-requisito.

Um programa é software livre se os usuários possuem todas essas liberdades. Portanto, você deve ser livre para redistribuir cópias, modificadas ou não, gratuitamente ou cobrando uma taxa pela distribuição ou pelo seu trabalho, a qualquer um, em qualquer lugar. Ser livre para fazer tudo isso significa (entre outras coisas) que você não deve ter que pedir ou pagar pela permissão para fazê-lo.

¹ Projeto lançado em 27 de Setembro de 1983 por Richard Stallman e atualmente patrocinado pela *Free Software Foundation*. Seu objetivo é fornecer um conjunto de softwares sob licenças que garantam a liberdade do usuário.

Você também deve ter a liberdade de fazer modificações e usá-las privativamente ou em seu trabalho ou lazer sem sequer mencionar que elas existem. Se publicar suas modificações, você não deve ser obrigado a avisar ninguém em particular, ou de qualquer modo em particular.

A liberdade de executar o programa significa que qualquer tipo de pessoa ou organização é livre para usá-lo em qualquer tipo de sistema computacional, ou para qualquer tipo de trabalho e propósito, sem que seja necessário comunicar ao desenvolvedor ou qualquer outra entidade específica. Nessa liberdade é o propósito do usuário que importa, não aquele do desenvolvedor. Você, como usuário, é livre para rodar o programa para seus propósitos e caso você o distribua a outra pessoa, ela também será livre para executá-lo com os propósitos dela, mas você não é intitulado a impor seus propósitos sobre ela.

A liberdade de redistribuir cópias deve incluir formas executáveis ou binárias do programa, bem como o código-fonte, tanto da versão modificada quanto da inalterada. A distribuição de programas em formato executável é necessária para alguns sistemas operacionais livres e convenientemente instaláveis, além de permitir que o usuário não se dê ao trabalho/conhecimento da/sobre processos de compilação. Não há problemas se não for possível produzir uma forma binária ou executável (pois algumas linguagens de programação não suportam este recurso), mas deve ser concedida a liberdade de se redistribuir nessas formas caso seja desenvolvido um meio de criá-las.

Para que as liberdades 1 e 3 (a liberdade de modificar e a liberdade de publicar versões modificadas) façam sentido, você deve ter acesso ao código-fonte do programa. Consequentemente, acesso ao código-fonte é uma condição necessária para o software livre. Código-fonte obscurecido, escondido, não é código-fonte real e não conta como código-fonte.

A liberdade 1 inclui a liberdade de usar sua versão modificada em lugar da original. Assim, se um programa é entregue como um produto projetado para rodar para outra pessoa, mas se recusa a rodar para você (prática conhecida como tivoização², travamento ou ainda como boot seguro), a liberdade 1 se torna ficção teórica ao invés de liberdade prática. Isso não é suficiente. Em outras palavras, esses binários não são softwares livres, mesmo que o código-fonte a partir do qual foram compilados seja livre.

Uma maneira importante de modificar um programa é agregar a ele módulos e sub-rotinas livres. Se a licença do programa diz que você não pode agregar a ele um módulo com uma licença adequada, por exemplo, se ele requer que você seja o detentor dos direitos autorais de qualquer

² Tivoização vem do inglês *tivoization*: prática utilizada pela marca de gravadores de vídeo digital TiVo que, embora utilizasse software livre, não permitia que usuários executassem versões modificadas do software em seu hardware.

código que adicionar, então essa licença é muito restritiva para ser qualificada como livre.

A liberdade 3 inclui a liberdade de publicar quaisquer versões modificadas como software livre. Uma licença livre também pode permitir outras maneiras de liberá-las; em outras palavras, ela não tem que ser uma licença copyleft. No entanto, a licença que requer que modificações sejam não-livres não se qualifica como uma licença livre.

Para que essas liberdades sejam reais, elas devem ser permanentes e irrevogáveis desde que você siga a licença do software. Se o desenvolvedor do software tiver o poder de revogar a licença, ou adicionar restrições retroativamente a seus termos sem restringi-la, o software não é livre.

Todavia, certas regras sobre a maneira de distribuir o software são aceitáveis quando elas não entram em conflito com as liberdades centrais. Por exemplo, o copyleft (apresentado de maneira muito simples) é a regra de que, quando redistribuindo um programa, você não pode adicionar restrições que neguem as liberdades centrais de outras pessoas. Essa regra não entra em conflito com as liberdades centrais, na verdade, ela as protege.

No projeto GNU, utilizam o termo copyleft para proteger legalmente essas liberdades para todos. Eles acreditam que existem razões importantes pelas quais é melhor usar o copyleft, mas se o seu programa é software livre que não faz uso do copyleft, ele ainda é basicamente ético.

Outro ponto importante é que software livre não significa “não comercial”. Um programa livre deve estar disponível para uso comercial, para o desenvolvimento comercial e para distribuição comercial. O desenvolvimento comercial de software livre já deixou de ser incomum. Tais softwares livres comerciais são muito importantes. Você pode ter pago dinheiro por suas cópias de software livre, ou você pode tê-las obtido a custo zero, mas independentemente de como você conseguiu suas cópias, você sempre deve ter a liberdade para copiar e mudar o software, ou mesmo para vender cópias.

Se uma modificação constitui ou não um aperfeiçoamento é uma questão subjetiva. Se o seu direito de modificar um programa é limitado, fundamentalmente, à mudanças que outra pessoa considere um aperfeiçoamento, então o programa não é livre.

No entanto, regras sobre como empacotar (fato relativo a organizar, homologar e as vezes compilar um código para fornecê-lo aos outros) uma versão modificada são aceitáveis, se elas não limitam substantivamente sua liberdade de liberar versões modificadas, ou sua liberdade de criar e usar modificações privadamente. Portanto, é aceitável que uma licença requira que você mude o nome do programa na versão modificada, remova uma logomarca ou identifique suas modificações

como suas. Desde que esses requerimentos não sejam tão penosos que eles efetivamente sejam um empecilho à distribuição de suas mudanças, eles são aceitáveis. Afinal, se você já está fazendo outras modificações ao programa, não terá muitos problemas em fazer algumas a mais.

Regras que dizem que “se você publicar sua versão de certa maneira, terá de publicá-la dessa outra também” podem ser também aceitáveis, na mesma condição. Um exemplo de regra aceitável é uma que diz que se você distribuiu uma versão modificada e um desenvolvedor anterior pedir por uma cópia, você deve enviar uma (note que tal regra ainda lhe deixa a possibilidade de distribuir ou não sua versão). Regras que requerem a liberação do código-fonte de versões que você fez públicas também são aceitáveis.

Uma questão especial surge quando uma licença requer a mudança do nome pelo qual o programa é invocado por outros programas. Isso efetivamente cria obstáculos à publicação de uma versão modificada que possa substituir a original quando invocada por aqueles outros programas. Esse tipo de requerimento é aceitável apenas quando existe uma maneira fácil que permite especificar o nome do programa original como um apelido ao nome da versão modificada.

Algumas vezes, as regulamentações de controle de exportação do governo e as sanções comerciais podem reprimir sua liberdade de distribuir cópias de programas internacionalmente. Desenvolvedores de software não têm o poder para eliminar ou passar por cima dessas restrições, mas o que eles podem e devem fazer é se recusar a impô-las como condições para o uso do programa. Dessa maneira, restrições não irão afetar as atividades e pessoas fora da jurisdição desses governos. Portanto, licenças de software livre não devem requerer a obediência a qualquer regulamentações de exportação como uma condição para qualquer das liberdades essenciais.

A maioria das licenças de software livre são baseadas em copyright e existem limites para que tipo de requerimentos possam ser impostos por meio do copyright. Se uma licença baseada no copyright respeita a liberdade nas maneiras descritas acima, é improvável que ela possua algum outro tipo de problema nunca antes antecipado (embora isso ocorra ocasionalmente). No entanto, algumas licenças de software livre são baseadas em contratos, que podem impor uma lista muito maior de restrições possíveis. Isso significa que existem muitas maneiras nas quais tal licença pode ser inaceitavelmente restritiva e não-livre. Existem diversos outros casos, mas torna-se difícil listar tudo o que pode acontecer.

Quando falamos sobre software livre, é melhor evitar termos como “dado” ou “de graça”, porque estes termos implicam que a questão é o preço, não a liberdade. Alguns termos comuns como “pirataria” englobam opiniões que espera-se que você não defenda.

Finalmente, note que critérios, assim como expressos nessa definição de software livre, requerem cuidadosa meditação quanto à sua interpretação. Saiba que qualquer tipo de obra pode ser livre, pois a definição de software livre pode ser estendida para a definição de obras culturais livres, aplicável a qualquer tipo de obra.

Além disso, outro grupo começou a usar o termo “código aberto” (do inglês *open source*) com um significado parecido (mas não idêntico) ao de “software livre”. Nós preferimos o termo “software livre” porque, uma vez que você tenha ouvido que ele se refere à liberdade ao invés do preço, ele traz à mente a liberdade. A palavra “aberto” nunca se refere à liberdade.

3. Copyleft

O modo mais simples de tornar um programa livre é colocá-lo em domínio público sem copyright. Isto permite que as pessoas compartilhem o programa e suas melhorias, se elas estiverem dispostas a tal. Mas isto também permite que pessoas que não gostam de cooperação transformem o programa em software proprietário. Eles podem fazer modificações, poucas ou muitas, e distribuir o resultado como um produto proprietário. As pessoas que receberem esta forma modificada do programa não tem a liberdade que o autor original havia lhes dado, pois o intermediário eliminou estas liberdades.

Sendo o objetivo dar a todos os usuários a liberdade de redistribuir e modificar o software, se algum intermediário fosse capaz de retirar a liberdade, os demais usuários não teriam mais essa liberdade. Então, em vez de colocar o software em domínio público, ele pode ser colocado sob copyleft, denotando que qualquer um que distribuir o software, com ou sem modificações, terá que passar adiante a liberdade de copiar e modificar novamente o programa. O copyleft garante que todos os usuários tem liberdade e também fornece um incentivo para que outros programadores contribuam com o software livre. Programas livres importantes como o Compilador GNU C++ existem só por causa disto.

O copyleft também ajuda os programadores que desejam contribuir a obterem permissão de fazer isto. Esses programadores frequentemente trabalham para empresas ou universidades que fariam qualquer coisa para ganhar mais dinheiro. Um programador pode desejar fornecer suas modificações para a comunidade, mas seu empregador pode desejar transformar as mudanças em um produto de software proprietário.

Quando explicamos ao empregador que é ilegal distribuir a versão melhorada, exceto como software livre, o empregador geralmente decide liberá-lo como software livre em vez de jogá-lo fora.

Para colocar um programa sob copyleft, primeiro deve-se registrar seu copyright. Então, deve-se adicionar nos termos de distribuição, que são um instrumento legal que garante a qualquer pessoa os direitos de usar, as regras que permitirão modificar e redistribuir o programa ou derivações se e somente se os termos de distribuição não forem modificados. Desta forma, o programa e as liberdades se tornam legalmente inseparáveis.

Desenvolvedores de software proprietário usam o copyright para retirar a liberdade dos usuários. O copyleft utiliza o copyright para garantir a liberdade. É por isso que invertem o nome,

mudando-o de copyright para copyleft.

Copyleft é um conceito genérico e você não pode utilizá-lo diretamente. Você pode usar somente uma implementação específica desse conceito, ou seja, uma licença que forneça suas características. No Projeto GNU, por exemplo, os termos específicos da distribuição estão contidos na Licença Pública Geral GNU, frequentemente chamada de GNU GPL.

Para bibliotecas, como diversas bibliotecas do GNU, recomenda-se utilizar uma forma derivada da GNU GPL, a GNU LGPL – GNU Lesser General Public License. As bibliotecas com essa licença podem ser utilizadas em programas proprietários. Porém, as bibliotecas com a licença GNU GPL podem ser utilizadas somente em programas livres.

Todas essas licenças foram criadas de modo que você possa facilmente aplicá-los ao seu próprio trabalho, supondo que você é o detentor do copyright. Você não tem que modificar a licença para fazer isso, basta simplesmente incluir uma cópia da licença no trabalho e adicionar notas aos arquivos-fonte fazendo referências adequadas à licença.

Além dessa licença derivada, também existe a GNU Affero General Public License – GNU AGPL, que foi criada para programas designados para servidores e a GNU Free Documentation License – GNU FDL, que pode ser utilizada em livros, manuais, documentações e outros documentos. A GNU AGPL garante que os programas modificados utilizados no servidor terão seu código fonte expostos ao público e a GNU FDL garante a liberdade de cópia e distribuição, com ou sem modificação, comercialmente ou não.

Todas as decisões tomadas pelas pessoas têm origem nos seus valores e objetivos. As pessoas podem ter muitos objetivos e valores diferentes; fama, lucro, amor, sobrevivência, diversão e liberdade são apenas alguns dos valores que uma pessoa correta pode ter. Quando o objetivo é uma questão de princípio, nós chamamos a isso idealismo.

Meu trabalho com software livre é motivado por um objetivo idealista: espalhar a liberdade e cooperação. Eu quero encorajar a difusão do software livre, substituindo o software proprietário, que proíbe a cooperação, e assim tornando nossa sociedade melhor.



E esta é a razão principal pela qual a GNU GPL (Licença Pública Geral GNU) foi escrita como foi — como copyleft. Todo código-fonte adicionado a um programa coberto pela GPL tem que ser software livre, mesmo se colocado em um arquivo separado. Eu torno o meu código disponível para uso em software livre, e não em software proprietário, a fim de encorajar outras pessoas que programam a deixar seu código livre também. Acho que, já que desenvolvedores de software proprietário usam copyright a fim de nos impedir compartilhar código, nós colaboradores podemos usar copyright para dar uma vantagem a outros colaboradores: eles podem usar o nosso código.

Nem todos que usam a GNU GPL têm isso por objetivo. Há muitos anos atrás, pediram a um amigo meu que relançasse um programa seu, com copyleft, sob termos não-copyleft, e ele respondeu mais ou menos assim:

“Às vezes eu trabalho desenvolvendo software livre, e às vezes eu trabalho desenvolvendo software proprietário — mas quando eu desenvolvo software proprietário, eu espero ser pago”

Ele desejava compartilhar seu trabalho com uma comunidade que compartilha software, mas não via motivo para ajudar a desenvolver produtos de grandes empresas, o que estaria fora dos limites de nossa comunidade. Seu objetivo era diferente do meu, mas ele decidiu que a GNU GPL era útil para o seu objetivo também.

Se você quer realizar algo, idealismo não é o bastante — você precisa escolher um método que funcione de verdade para atingir o seu objetivo. Em outras palavras, você precisa ser “pragmático”. E a GPL, é pragmática? Vamos dar uma olhada em seus resultados.

Considere o GNU C++. Por que nós temos um compilador C++ livre? Somente porque a GNU GPL disse que ele teria que ser livre. O GNU C++ foi desenvolvido por um consórcio de empresas, o MCC, começando a partir do Compilador C GNU. O MCC normalmente lança seu trabalho como software proprietário, tão proprietário quanto este possa ser. Mas eles fizeram a interface para o C++ livre, porque a GNU GPL disse que esta era a única forma de fazê-lo. A interface para C++ incluiu muitos arquivos novos, mas já que eles seriam ligados ao GCC, a GPL se aplicava a eles. E o benefício para a nossa comunidade é evidente.

Considere o GNU Objective C. NeXT primeiramente quis fazer essa interface proprietária. Eles propuseram lançar isso como arquivos “.o”, e deixar que os usuários conectassem-os com o resto do GCC, pensando que isso seria uma forma de burlar os requisitos impostos pela GPL. Mas nosso advogado disse que isso não os isentaria desses requisitos, que isso não era permitido. Então eles fizeram a interface para o Objective C ser software livre.

Estes exemplos aconteceram há anos atrás, mas a GNU GPL continua nos trazendo mais software livre.

Muitas bibliotecas do projeto GNU estão cobertas pela GNU LGPL (Licença Pública Geral Menor), mas não todas. Uma biblioteca GNU que é coberta pela GNU GPL é a Readline, que implementa edição pela linha de comando. Uma vez eu descobri um programa proprietário que foi projetado para usar a Readline, e avisei o desenvolvedor que isso não era permitido. Ele poderia ter tirado a edição via linha de comando de seu programa, mas o que acabou acontecendo foi ele relançá-lo sob a GNU GPL. Agora este programa é software livre.

Os programadores que desenvolvem melhorias para o GCC (ou para o Emacs, ou Bash, ou Linux, ou qualquer programa coberto pela GPL) são, muito frequentemente, funcionários de empresas ou universidades. Quando o programador quer devolver suas melhorias para a comunidade, e ver o seu código na próxima versão do programa, seu chefe pode dizer, “Espere aí — seu código pertence a nós! Não queremos compartilhá-lo; nós decidimos transformar sua versão melhorada do programa em software

proprietário.”

E aqui a GNU GPL vem ajudar! O programador mostra ao seu chefe que lançar este programa como software proprietário seria infringir direitos de propriedade autoral, e o chefe percebe que ele tem apenas duas opções: lançar o novo código como software livre ou simplesmente não lançá-lo. Quase sempre ele permite ao programador fazer o que este sempre quis, e o código é liberado na nova versão do programa.

Mas veja bem: a GNU GPL não é o Sr. Bonzinho. Ela diz não a algumas das coisas que as pessoas às vezes gostariam de fazer. Existem usuários que dizem que esta é uma coisa ruim — que a GPL “exclui” alguns desenvolvedores de software proprietário que “precisam ser trazidos à comunidade do software livre.”

Mas nós não estamos excluindo-os de nossa comunidade, eles estão escolhendo não entrar. A sua decisão de desenvolver software proprietário é a decisão de ficar de fora de nossa comunidade. Estar em nossa comunidade significa se juntar e cooperar conosco. Não podemos “trazê-los para nossa comunidade” se eles não querem entrar.

O que nós podemos fazer é convidá-los, induzi-los a entrar. A GNU GPL foi projetada para isso, através do nosso software já existente: “Se você for tornar o seu software livre, você pode usar esse código”. É claro, isso não ganhará a todos, mas ganha em algumas vezes.

O desenvolvimento de software proprietário não contribui com a nossa comunidade, mas seus desenvolvedores frequentemente querem a nossa ajuda. Usuários de software livre acabam motivando-nos através do nosso próprio ego — dando-nos reconhecimento e gratidão — mas pode ser muito tentador quando uma grande companhia lhe diz, “Apenas nos deixe colocar seu código em nosso programa proprietário, e seu programa será usado por muitos milhares de pessoas!”. A tentação pode ser poderosa, mas a longo prazo estaremos melhor se resistirmos a ela.

A tentação e pressão ficam mais difíceis de serem reconhecidas quando vêm indiretamente, através de organizações de software livre que adotaram a política de fornecerem também software proprietário. O X Consortium (e seu sucessor, o Open Group) nos dá um exemplo: fundado por empresas que desenvolvem software proprietário, eles lutaram por uma década para persuadir os programadores a não usarem copyleft. Agora que o Open Group fez do X11R6.4 software não-livre, aqueles de nós que resistiram a essa pressão estão felizes por tê-lo feito.

Em Setembro de 1998, alguns meses depois do X11R6.4 ter sido lançado sob termos não-livres para distribuição, o Open Group voltou atrás em sua decisão e o relançou sob a mesma licença de distribuição livre de software não-copyleft que era usada para o X11R6.3. Muito obrigado, Open Group — mas a decisão destes de voltarem atrás não invalida as conclusões que tiramos, do fato de que adicionar essas restrições era possível.

Pragmaticamente falando, pensar em objetivos de mais longo prazo nos fortalecerá para resistirmos a essa pressão. Se focarmos nossa mente na liberdade e na comunidade que podemos construir ao permanecermos firmes, encontraremos a força para fazermos isso. “Mantenha-se firme por alguma coisa, ou você cairá por qualquer coisa.”

E se cínicos ridicularizarem a liberdade, ridicularizarem a comunidade... se “realistas” disserem que lucro é o único objetivo... apenas os ignore, e use copyleft assim mesmo.

Referências

https://en.wikipedia.org/wiki/Free_software

<http://www.gnu.org/philosophy/free-sw.html>

<http://opensource.org>

<http://www.fsf.org>