

Compreendendo o Software de Fonte Aberta

Mark H Webblink, Red Hat, Inc.

Tradução: Leonardo Santiago, Alexsandro Meireles

RESUMO: Mark Webblink, vice-presidente principal e consultor jurídico geral da Red Hat, Inc., apresenta uma visão geral altamente informativa e abrangente sobre o *software* de fonte aberta. Webblink identifica os diferentes tipos de licenças de fonte aberta e discorre sobre como as licenças de fonte aberta se comparam a licenças de outros *softwares* com base nos direitos protegidos pelos direitos autorais estendidos ao licenciado. Ele apresenta uma análise detalhada da *GNU General Public Licence* (Licença Pública Geral GNU), a mais extensiva licença para *software* de fonte aberta, e examina como o conceito de *copyright* de um “trabalho derivativo” se aplica ao *software* de fonte aberta. O artigo conclui desbancando mitos existentes sobre o *software* de fonte aberta e sugere algumas das melhores práticas para software.

PALAVRAS-CHAVE: Software Livre. Fonte Aberta. Licenças. GNU.

Mark Webblink entrou para a Red Hat, Inc., como seu primeiro consultor jurídico geral em maio de 2000. Em seguida, foi eleito secretário e vice-presidente principal da companhia. Antes de juntar-se à Red Hat, Webblink estava associado à Moore & van Allen, PLLC, na qual seu trabalho se direcionava para transações de propriedade intelectual, incluindo licenciamento de software e patentes. Webblink tem falado sobre licenciamento de fonte aberta e questões sobre patentes de softwares para a National Academy of Sciences, Practicing Law Institute’s Computer Law Institute, Georgetown University’s Advanced Computer and Internet Law Institute, a Licensing Executives Society, a Federal Trade Commission, o Departamento de Justiça e o Congresso dos Estados Unidos da América.

O Que é Software de Fonte Aberta?

A *Open Source Initiative* (“OSI”) define Fonte Aberta como um software que oferece os seguintes direitos e obrigações:

1. Nenhum *royalty* ou outra taxa cobrada nas distribuições;

2. Disponibilidade do código-fonte;
3. O direito de criar modificações e trabalhos derivados;
4. Pode requerer versões modificadas a serem distribuídas como versão original mais atualizações;
5. Nenhuma discriminação contra pessoas ou grupos;
6. Nenhuma discriminação contra especialidades (ou grupos específicos de trabalho);
7. Todos os direitos concedidos devem se estender para as versões redistribuídas;
8. A licença se aplica ao programa como um todo e a cada um de seus componentes;
9. A licença não deve restringir outro software, permitindo assim a distribuição de software de fonte aberta e de fonte fechada ao mesmo tempo.

Essa definição claramente dá margem a uma ampla variedade de licenças, e nós examinaremos rapidamente um número desses tipos de licenças. Embora seja a essa definição OSI de Fonte Aberta que o restante deste texto se refere, vale a pena também examinar a definição de *Software Livre*, pois muitas vezes as expressões *Software Livre* e *Fonte Aberta* são usadas indistintamente. Embora sejam similares, existem diferenças que merecem ser verificadas.

Quando falamos *Software Livre*, não estamos referindo a *freeware*, ou seja, software que é essencialmente de domínio público. Ao contrário, estamos falando de software que é licenciado sobre as regras da *Free Software Foundation* (FSF) e de seu carro-chefe *GNU General Public License*.

De acordo com a definição da FSF:

“*Software livre* é uma questão da liberdade do usuário de usar, copiar, distribuir, estudar, mudar e melhorar o *software*. Mais precisamente, refere-se a quatro tipos de liberdade para os usuários do *software*:

1. Liberdade para usar o programa, para qualquer fim (liberdade 0);
2. Liberdade para estudar como o programa funciona e adaptá-lo a suas necessidades (liberdade 1). O acesso ao código-fonte é um pré-requisito para isso;
3. Liberdade para redistribuir cópias, para que se possa ajudar seu vizinho (liberdade 2);
4. A liberdade de melhorar o programa e liberar esses aperfeiçoamentos ao público, de tal forma que toda a comunidade seja beneficiada (liberdade 3). O acesso ao código-fonte é um pré-requisito para isso.

Um programa é definido *software* livre se os seus usuários têm todas essas liberdades”. Comparando as definições de Fonte Aberta e de *Software* Livre, descobre-se que todo *Software* Livre é de Fonte Aberta, mas, da forma administrada pela *Free Software Foundation*, nem toda Fonte Aberta é *Software* Livre. A diferença aparece principalmente a partir da assim chamada compatibilidade de licença, mas em grande medida as diferenças são principalmente filosóficas, e não substanciais.

Fundamentos da Lei de Direitos Autorais

Para melhor apreciação do *software* de Fonte Aberta, necessitamos de uma compreensão básica da Lei dos Direitos Autorais¹. Para avaliar os direitos concedidos sob as licenças de Fonte Aberta, é necessário primeiro estar acostumado com o pacote básico de direitos concedidos ao detentor de um direito autoral. Pela lei de direitos autorais dos E.U.A., esses direitos são:

1. Direito exclusivo de cópia do trabalho;
2. Direito exclusivo de fazer trabalhos derivados;
3. Direito exclusivo de distribuir o trabalho;
4. Direito exclusivo de executar o trabalho; e
5. Direito exclusivo de propagar o trabalho².

Esses direitos, por outro lado, estão sujeitos a certas limitações, tais como direitos de “uso honesto”. O uso honesto inclui o uso de uma obra para fins de crítica, comentário, reportagem, ensino, bolsa de estudo ou pesquisa, e não constitui violação da obra. Se um uso específico é “uso honesto” é determinado por um número de fatores, incluindo: (1) O propósito e caráter do uso, inclusive se tal uso é de natureza comercial ou se é para fins educacionais sem fins lucrativos; (2) A natureza do código protegido por direitos autorais; (3) A quantidade e substancialidade da porção usada em relação ao código protegido como um todo; e (4) o efeito do uso sobre o mercado potencial ou o valor do código protegido³.

Obras como um *software* podem ser postas em domínio público e existir fora do âmbito da lei de direitos autorais⁴. No entanto, com as mudanças na lei de direitos autorais nos anos 70 e 80, incluindo a aplicação automática dos direitos autorais nos termos da Convenção de Berna, não é mais uma tarefa fácil colocar um *software* sob domínio público⁵. Um *software* (ou qualquer outro conjunto de obras) que está no domínio público não pode, por definição, impor restrições quanto a quem ou como ele pode ser usado, modificado ou distribuído (embora outras leis, tais como

controles de exportação, podem ainda restringir o uso ou distribuição de algum *software*). Se o *software* de fonte aberta estivesse no domínio público (ou seja, não sujeito ao direito autoral, uma vez que o autor renunciara a esse direito para a obra), qualquer negócio ou pessoa poderia usar o *software* por qualquer propósito sem restrição alguma de direitos autorais, e não haveria condições para revisão legal sobre e além de uma garantia de obediência a outros estatutos (os quais se aplicam igualmente a todos os outros *softwares*, no domínio público ou não). Em razão de o *software* de fonte aberta não estar sob domínio público, mas sim protegido pela lei de direitos autorais e licenciado para uso sob certos termos, talvez incomuns, esses termos precisam ser compreendidos.

Uma licença válida de direitos autorais se aplica a um conjunto de obras e precisa impor pelo menos uma restrição. Uma licença de direitos autorais que não coloque qualquer restrição, implicitamente concede todos os direitos, incluindo direitos de usar, modificar, distribuir etc. A maioria das licenças de *softwares* proprietários impõe restrições ao uso (incluindo definições de “uso honesto”, que, de acordo com tais licenças, comumente não incluem descompilação, engenharia reversa ou outras afins), à cópia (comumente apenas para fins de backup) e redistribuição (comumente apenas quando agindo como agente autorizado para o proprietário dos direitos autorais).

Tipos de Licenças de Fonte Aberta

As licenças de fonte aberta podem ser de um modo geral categorizadas nos seguintes tipos: (1) Aquelas que não impõem qualquer restrição sobre a distribuição de obras derivadas (nós a chamaremos de Licenças Não Protegidas, porque elas não protegem o código de ser usado em aplicativos que não sejam de fonte aberta); e (2) aqueles que de fato aplicam tais restrições (as quais chamaremos de Licenças Protegidas, uma vez que asseguram que o código permanecerá sempre aberto/livre).

Para melhor avaliarmos a natureza dessas licenças, é proveitoso retratar as licenças de *softwares* numa seqüência contínua baseada nos direitos constantes no *copyright* estendidos ao licenciado. Veja o diagrama 1 ao final deste artigo.

O *software* que tenha sido colocado no domínio público está livre de todas as restrições, estando todos os direitos previstos no *copyright* concedidos ao público como um todo. Aqueles que concedem licenças não protegidas de fonte aberta mantêm seu *copyright*, mas concedem todos os

direitos do *copyright* ao licenciado. Aqueles que concedem licenças protegidas de fonte aberta mantêm seus direitos autorais, concedem todos os direitos do *copyright* ao licenciado, mas aplicam pelo menos uma restrição, geralmente que a redistribuição do *software*, quer seja modificado ou não modificado, deve estar sob a mesma licença. Aqueles que concedem licenças de propriedade mantêm seus direitos autorais e apenas concedem alguns poucos direitos sob *copyright*, geralmente apenas os direitos de usar e exibir. A tabela seguinte, onde a licença BSD é usada como exemplo de uma licença não protegida e da Licença Pública Geral GNU como exemplo de uma licença protegida de fonte aberta, mostra esses contrastes – veja o Diagrama 2 na conclusão deste artigo.

As licenças não protegidas de fonte aberta incluem: Academic Free License v.1.2; Apache Software License v.1.1; Artistic; Attribution Assurance license; BSD License; Eiffel Forum License; Intel Open Source License for CDSA/CSSM Implementation; MIT License; Open Group Test Suite License; Q Public License v.1.0; Sleepycat License; Sun Industry Standards Source License; University of Illinois/NCSA Open Source License; Vovida Software License v.1.0; W3C Software Notice and License; X.Net, Inc. License; zlib/libpng License; e Zope Public License v.2.0.

As licenças protegidas de fonte aberta incluem: Apple Public Source License v.1.2; Artistic License; Common Public License v.1.0; GNU General Public License v.2.0; GNU Lesser General Public License v.2.1; IBM Public License v.1.0; Jabber Open Source License v.1.0; MITRE Collaborative Virtual Workspace License; Motosoto Open Source License v.0.9.1; Mozilla Public License v.1.0 and v.1.1; Nethack General Public License; Noika Open Source License v.1.0a; OCLC Research Public License v.1.0; Open Software License v.1.1; Python License; Python Software Foundation License v.2.1.1; Ricoh Source Code Public License v.1.0; e Sun Public License v.1.0.

Todas essas licenças e novas licenças adicionais podem ser encontradas no *website Open Source Initiative* (www.opensource.org).

Algumas licenças de fonte aberta de ambos os tipos incluem outras condições, como restrições ao uso de marcas registradas, concessão expressa de licença com respeito a patentes aplicáveis, renúncias a garantias, indenização de detentores de direitos autorais em distribuições comerciais e renúncias de obrigações exigíveis. Contudo, nenhuma dessas condições é tão importante como as obrigações/restrições impostas sobre a redistribuição de direitos sob as licenças protegidas de fonte aberta, e são essas restrições sobre a redistribuição que enfocaremos a seguir.

A Licença Pública Geral GNU

A partir deste texto, a Licença Pública Geral GNU (GPL) é a licença mais difundida de *software* de fonte aberta. De todos os *softwares* aos quais esse título tenha sido aplicado, nenhum é mais conhecido que o Linux® kernel. De fato, a GPL tem sido aplicada à maioria daqueles módulos de *softwares* incluídos nas distribuições mais conhecidas do Linux®, como a Red Hat® Linux®. Seu amplo apelo no meio da comunidade de fonte aberta deriva do fato de que ela (GPL) cai naquela categoria de licenças de fonte aberta que obrigam as partes que querem redistribuir tal *software*, seja em forma original ou modificada (derivada), a fazê-lo nos termos do acordo da licença sob a qual tal *software* foi recebido (todos os quais nós referimos como licenças protegidas). Ou seja, tendo recebido o direito de usar, modificar e redistribuir o *software* sob a GPL, esta exige que você estenda os mesmos privilégios sob os mesmos termos a outros que recebem o *software* de você. Esta é a linha comum que comanda as licenças protegidas e, por essa razão, focalizaremos a GPL como padrão de licenças protegidas.

A GPL oferece certos direitos a qualquer um que receba uma licença de um *software* que esteja sob seu comando. Ao mesmo tempo, ela impõe muito poucas obrigações, a não ser àqueles que desejam redistribuir o *software*. Esses direitos e deveres são:

1. Direito de copiar e redistribuir desde que você inclua uma nota de *copyright* e uma renúncia de garantias. Você pode cobrar o custo de distribuição e pode oferecer proteção de garantia por uma taxa.
2. Direito de criar obras derivadas para seu próprio uso.
3. Direito de distribuir obras derivadas, desde que você:
 - a) Identifique a obra como modificada;
 - b) A licença sob a GPL; e
 - c) Conceda interativamente a informação de licença, se o programa normalmente funcionar interativamente.
4. Você pode distribuir a obra apenas de forma executável, desde que o código-fonte seja:
 - a) Distribuído com o código do objeto;
 - b) Oferecido por escrito, válido por um período de pelo menos três anos, e tornar o código-fonte disponível por não mais que o custo de distribuição; e
 - c) Para distribuições não-comerciais, acompanhadas da oferta à parte redistribuidora quanto à disponibilidade do código-fonte.

5. Você não pode impor restrições a qualquer desses direitos.

Essa é uma abordagem simples, porém elegante. Basicamente, o licenciador está permitindo a qualquer licenciado exercer virtualmente todos os direitos disponíveis sob *copyright*, ou seja, o direito de fazer trabalhos derivados, o direito de distribuir, o direito de usar, o direito de exibir. A única obrigação imposta é: se o licenciado, por sua vez, deseja distribuir o *software* a outras partes, eles concordarão em fazê-lo apenas sob a GPL. O propósito único dessas restrições é preservar a integridade da oferta original de liberdade através de qualquer meio de distribuição e impossibilitar qualquer pessoa de criar uma versão do *software* que ofereça a qualquer um que o receba, menos liberdade do que a versão original ofereceu. Parafraseando, a GPL reafirma “uma vez livre, sempre livre”.

Note que a GPL não tem qualquer relevância para o caso em que uma parte licencia o *software* e escolhe não a redistribuir. Isso é verdadeiro, não importa se a parte é um indivíduo, uma corporação, um conglomerado corporativo ou o governo. Como foi notado pela FSF, quando a GPL menciona “You” (“Você”) no contexto de uma corporação, refere-se à empresa matriz e a todas as subsidiárias controladas por essa matriz. Da mesma forma, quando “You” dirige-se a uma unidade governamental, isso se refere àquela unidade e a todas as subdivisões daquela unidade que estão sob seu controle. Nesse contexto, “You” pode tranquilamente se referir a todo o governo federal dos Estados Unidos da América ou a qualquer estado ou governo de uma nação, inclusive às agências daquele estado ou governo de uma nação. A GPL não exige que um licenciado que não tenha distribuído o *software* a outrem forneça cópias daquele *software* a qualquer parte que assim o exija.

As restrições da GPL se aplicam apenas no caso de quando um *software* sob a GPL está sendo fornecido à outra parte e a GPL diz respeito apenas à preservação do seu original e a nenhum outro propósito.

Baseado no que foi dito acima, podemos dividir os tipos de uso de Fonte Aberta em categorias e analisar as implicações legais da GPL para cada categoria. As categorias de interesse são:

1. Usuários que usam apenas GPL binárias assim como usariam qualquer outro programa similar;
2. Usuários que modificam fontes GPL para tratar de problemas de configuração local ou para atender a exigências internas, e não para distribuir a outros; e
3. Usuários que modificam fontes GPL e as redistribuem por diversão ou para fins lucrativos.

No caso (1), a GPL absolutamente não afeta esses usuários; o uso do editor de textos de Fonte Aberta GNU Emacs TM não implica em que a ação de salvar um arquivo mude a propriedade do arquivo para a FSF, nem a compilação de um arquivo pela Fonte Aberta GNU C Compiler faz com que o código do objeto resultante pertença à FSF, nem criando um ponto de travamento num executável faz com que esse executável subitamente se torne propriedade da FSF. Assim, o uso normal de um *software* GPL (ou seja, o uso como o que alguém faria de um *software* comercial) num ambiente comercial não propõe qualquer problema legal extraordinário. A larga distribuição de *softwares* para o sistema operacional Linux nos últimos anos para uso em comercial na Internet ou para *enterprise servers* é uma ampla evidência de que não há razão legal para não se usar um *software* de fonte aberta se acontecer de você pensar que será melhor que as alternativas proprietárias.

No caso (2), o *software* modificado localmente, por definição, confere a seus usuários acesso às fontes modificadas localmente. Não há qualquer exigência dentro da GPL que tais modificações locais sejam reveladas a qualquer outra parte.

No caso (3), chegamos ao grupo de usuários para quem a GPL foi realmente escrita. Usuários que redistribuem versões modificadas ou não de *software* de Fonte Aberta devem obedecer à “Regra de Ouro” da GPL, de licenciar o *software* distribuído sob a GPL e não adicionar qualquer restrição criada por eles mesmos. Na medida em que alguém deseja obter lucros com um *software* GPL usando restrições tradicionais de licença de *software* proprietário, ficará demonstrado ser difícil, se não impossível, aplicar essas restrições. Note, no entanto, que obter lucro usando a GPL é legal e encorajado.

O que é uma obra derivada?

O *U.S. Copyright Act* define uma obra derivada como:

uma obra baseada em um ou mais obras pré-existentes, como uma tradução, um arranjo musical, uma dramatização, uma ficcionalização, uma versão de desenho animado, gravação de som, reprodução de arte, um resumo, uma condensação ou qualquer outra forma a partir da qual uma obra possa ser reformada, transformada ou adaptada. Uma obra que consiste de revisões editoriais, anotações, elaborações ou outras modificações, as quais, como um todo, representem uma obra de autoria original, é uma “obra derivada.

Logo, uma obra baseada em uma ou mais obras pré-existentes constitui uma obra derivada, na medida em que o novo material adicionado constitua uma obra de autoria original. Esse novo

material pode incluir revisões editoriais, anotações, elaborações ou outras modificações. Obras derivadas podem modificar a obra original, como numa tradução de texto, por exemplo, inclusive traduzir um software produzido por uma determinada linguagem de programação para uma outra linguagem, ou podem combinar a obra original com outras obras, como numa compilação ao modo Red Hat® Linux®. A proteção do *copyright* numa obra derivada ou numa compilação se estende apenas ao material agregado pelo autor de tal obra, e não concede direitos em material pré-existente incluído na nova obra⁷.

Onde se aplica a lei sobre obras de *software* derivadas?

A lei sobre obras derivadas em *software* não está bem estabelecida. A “*U.S. Copyright Act*” não se refere especificamente a obras derivadas em *software*, e não existe nenhum caso em estudo na Suprema Corte dos E.U.A. Na maioria dos casos, a lei tem-se desenvolvido entre as várias Cortes de Apelação norte-americanas, mas mesmo nelas a lei varia entre uma e outra.

O “*Copyright Act*” fornece uma definição importante em adição àquela de “obras derivadas”, a de “programas de computação”, que são definidas como: “um conjunto de enunciados ou instruções a serem usados direta ou indiretamente num computador para chegar a um certo resultado.”⁹

Além disso, o “*Copyright Act*” limita o alcance do que é protegido por *copyright* ao excluir algum assunto. O parágrafo 102(b) do “*Act*” determina:

Em nenhuma hipótese a proteção de copyright para uma obra de autoria original se estenderá a qualquer idéia, procedimento, processo, sistema, método de operação, conceito, princípio ou descoberta, independente da forma na qual seja descrito, explicado, ilustrado ou incluído em tal obra.

Talvez o mais aceito dos testes para obras derivadas em *software* seja o de “abstração, filtração e comparação” (“AFC”), estabelecido pelo “*Second Circuit*”¹⁰. Sob o teste em três partes AFC, uma corte primeiramente determina (sumariza) as partes estruturais constituintes do programa original. A partir dessas partes estruturais, a Corte então filtra todas as porções não passíveis de proteção, incluindo aquelas matérias não protegíveis definidas no parágrafo 102(b) do “*Copyright Act*” e elementos que estão no domínio público. No passo final, a Corte compara qualquer código remanescente que contenha expressão criativa para a estrutura do segundo programa para determinar se o *software* em questão é suficientemente similar à obra pré-existente para justificar

uma conclusão de que o segundo programa é uma obra derivada do primeiro. Essa abordagem da AFC tem sido adotada por três outros circuitos: o Quinto¹¹, o Décimo¹² e o Undécimo¹³.

Das nove Cortes de Apelação norte-americanas restantes, apenas uma adotou um teste claro para obras derivadas em *software*. O teste do Nono Circuito está baseado num exame analítico profundo, que primeiramente considera se há similaridades substanciais nas idéias e expressões das duas obras e então utiliza tal dissecação analítica para determinar se há características similares que sejam protegidas por *copyright*¹⁴. Os elementos similares são categorizados pelo grau de proteção que eles devem receber. Uma proteção “estreita” é dada a fatos ou idéias não passíveis de *copyright* que demandem proteção apenas pela maneira como esses fatos ou idéias são alinhados e apresentados. Uma proteção “larga” é dada a uma expressão passível de *copyright*. A Corte usa esses padrões para fazer uma comparação subjetiva das obras para determinar se, como um todo, são suficientemente similares para justificar uma conclusão de que uma delas é uma obra derivada da outra.

Como esses testes se aplicam a obras derivadas em um *software* de Fonte Aberta?

Com relação a obras derivadas, a Fonte Aberta demanda uma consideração especial. Isso se deve principalmente ao fato de que um *software* de fonte aberta, por definição, permite a criação de obras derivadas. Sob uma licença Não-Protegida, as novas porções de tal obra derivada pode ser licenciada sob uma licença escolhida pelo autor, e há pequena probabilidade de uma disputa violadora de regras.

A situação é muito diferente com uma licença protegida, porque esta requer que as obras derivadas sejam licenciadas sob a mesma licença aplicada à obra original. Aqui, a questão se torna largamente um grau de copiar *versus* uma proibição adequada de derivação. Onde um *software* de fonte aberta licenciado sob uma licença protegida parece ter sido copiado, no todo ou em parte, e que seja então licenciado sob uma licença diferente que a da obra original, a questão de obra derivada e infração seria determinada pelas Cortes usando os testes mencionados acima. No entanto, não é o caso onde o autor da segunda obra mantém a licença protegida original com relação à obra original, mas licencia a nova obra sob uma licença diferente, pois aqui o autor subsequente não infringiu os direitos do autor original, exceto na medida em que a nova obra pode ser considerada uma obra derivada da original. Este último caso requer uma abordagem inteiramente diferente para considerar a derivação.

Na situação em que a obra original continua licenciada sob uma licença protegida e a nova obra é licenciada sob uma licença alternativa, os fatores a seguir devem ser considerados ao se determinar se a nova obra é derivada da original:

1. A substancialidade da nova obra;
2. Se alguma parte da obra original foi modificada; e
3. De que maneira tal modificação foi efetuada.

Esta análise é consistente com a distinção delineada pela própria GPL. A cláusula 2 da GPL declara: “Logo, não é intenção desta seção reclamar direitos ou contestar os seus direitos de escrever uma obra inteiramente para você; pelo contrário, a intenção é exercitar o direito de controlar a distribuição de obras derivadas ou coletivas baseadas no Programa. Além disso, a simples agregação de outra obra não baseada no Programa com o Programa (ou com uma obra baseada no Programa) num volume de armazenamento ou meio de distribuição não põe a outra obra sob o alcance desta licença”.

Por exemplo, se a obra em questão é uma base de dados inteiramente escrita por você, e o Programa em questão é um sistema operacional com GPL (um dos muitos a que a base de dados pode estar ligada), a distribuição da base de dados com o sistema operacional num volume de armazenamento (como o disco rígido do sistema) não concederia a GPL do sistema operacional ao *software* de base de dados. Por outro lado, se forem feitas modificações no Programa (o sistema operacional) para se adaptar a obra (base de dados), então aquelas modificações, as quais são uma obra derivada do Programa, precisariam ser disponibilizadas sob a GPL. Nenhuma modificação da obra (a base de dados) precisa ser redistribuída nesse caso.

Em suma, as exigências legais da GPL são totalmente dirigidas para fornecedores de *software* comercial: se você deseja usar um modelo proprietário para obter lucros, mantenha suas obras (ou seja, o código) separadas das obras com GPL, mantenha independentes as modificações feitas a cada uma delas, e não encontrará problemas ao proteger suas obras originais. Ao mesmo tempo, quaisquer modificações que você fizer no *software* que já esteja coberto pela GPL estará sujeito à GPL.

Mitos sobre a Fonte Aberta

Antes de deixarmos esta discussão sobre licenciamento de Fonte Aberta, vale a pena tratar de alguns dos mitos ou equívocos que se levantaram à volta de Fonte Aberta.

Mito 1 – Fonte Aberta é como um vírus que mina os direitos de propriedade intelectual

Esse mito é particularmente rico. Primeiro, como já foi notado, *software* de Fonte Aberta é fundamentalmente fundamentado na lei dos direitos autorais. Como acontece com o detentor de qualquer *copyright*, o detentor de *copyright* do *software* de Fonte Aberta tem como escolher quais direitos quer conceder aos outros. Os autores de Fonte Aberta simplesmente escolhem conceder mais direitos que os fornecedores de obras de proprietários. O simples fato de que um autor de Fonte Aberta, usando uma licença protegida, insiste que obras derivadas distribuídas a outros sejam licenciadas sob a mesma licença, deve ser contrastado com as licenças de *softwares* de proprietários que simplesmente negam ao licenciado o direito de criar obras derivadas ou redistribuí-las. Cada um é uma maneira de exercitar os direitos de propriedade intelectual, e nenhum dos dois está errado.

Mito 2 – O *software* de Fonte Aberta é mais inclinado a acusações de infrações de propriedade intelectual

A sugestão do fornecedor de *softwares* de proprietários é que, pelo fato de o modelo de desenvolvimento de Fonte Aberta depender de uma vasta rede de desenvolvedores de Fonte Aberta que não estão necessariamente sob o controle do distribuidor, o código produzido é mais suscetível de ser exposto a acusações de infrações de propriedade intelectual. Os fatos simplesmente não sustentam essa alegação. Enquanto inegavelmente tenha havido tais acusações contra alguns projetos de desenvolvimento de Fonte Aberta e/ou contra seus distribuidores, as reclamações têm sido poucas e esparsas.

Mito 3 – Ao contrário de distribuidores de [*softwares* pertencentes aos seus donos], os distribuidores de Fonte Aberta não oferecem garantias ou indenização contra infração de propriedade intelectual

Isso é verdadeiro, mas não mais verdadeiro para Fonte Aberta que para distribuidores de *softwares* pertencentes aos seus donos. Por exemplo, a licença do Windows 98 expressamente recusa qualquer garantia de não infração.

Mito 4 – A GNU General Public License é arriscada porque nunca foi testada numa Corte

Verdadeiro novamente. Mas, o que é mais arriscado, licenciar práticas que estão constantemente sendo desafiadas, ou aquelas que, em sua simplicidade e efetividade, têm evitado desafios?

Mito 5 – Tornar seu código-fonte visível a alguns usuários é equivalente à Fonte Aberta.

A Fonte Aberta valoriza seus clientes e usuários ao lhes dar total controle sobre seus ambientes de computação. O cliente tem a permissão de escolher se deve executar a versão padrão ou se deseja modificações. O cliente pode não apenas ver os *bugs*, ele pode repará-los. Tornar o código-fonte apenas visível para uns poucos usuários não os ajuda a compreender o código, não lhes permite reparar o código quando ele apresenta problemas. Esse acesso ao “partilhamento” do código-fonte equivale a entrar numa biblioteca pública apenas para descobrir que não há um fichário de catalogação e que todos os livros estão em armários de vidro trancados. Sim, você pode remexer e encontrar os títulos dos livros, mas não terá como obter conhecimento deles. O *software* proprietário maximiza seu valor ao assegurar que um monopólio não pode ser acessado.

Mito 6 – Os métodos de Fonte Aberta não produzem inovação

Isso é um mito. A comunidade de Fonte Aberta: (a) desenvolveu o servidor *web* Apache, que é usado para rodar a maioria dos servidores *web* no mundo hoje; (b) desenvolveu o *Sendmail*, o mais popular *software* de gerenciamento de e-mails; e (c) desenvolveu o BIND, a base para o uso de nomes de domínio, ao invés de endereços IP, para localizar *websites*. Claramente, a Fonte Aberta é capaz de promover avanços na arte do *software*.

Sem criticar esse ponto, voltemos às boas práticas que um escritório de advocacia corporativo deve manter com respeito a *software*, se de Fonte Aberta ou proprietário.

Boas práticas de um escritório de advocacia corporativo para *software*

Com respeito a qualquer forma de propriedade intelectual, há riscos associados com o licenciamento do uso de um *software*. Alguns dos riscos podem estar ligados especificamente à Fonte Aberta, porém muito freqüentemente acontecem seja qual for a forma de licença. A seguir colocamos uma série de boas práticas que todo escritório corporativo de advocacia deve implementar em sua companhia.

1. Não permitir a importação descontrolada de *software* nos computadores da companhia.

Não permitir que os empregados baixem *freewares*, *sharewares* ou programas de Fonte Aberta para os computadores da companhia sem antes conhecer com clareza os termos de licença com ajuda do departamento jurídico. Ao mesmo tempo, barrar o uso de *software* proprietário, a não ser na medida em que a companhia possa contar com as licenças permitidas. Em outras palavras, saber o que você está colocando em suas máquinas – agir de modo diferente expõe sua companhia a risco.

2. Tratar com distribuidores de *software* respeitáveis e com estabilidade financeira.

Um dos maiores riscos que uma companhia assume é adotar o uso de um *software* que não tenha futuro. Igualmente verdadeiro é licenciar um *software* de uma companhia sem os recursos financeiros necessários para manter e proteger aquele *software*. Conheça seus fornecedores. Conheça sua capacidade financeira, saiba sobre suas políticas de licenciamento, conheça sua rapidez de resposta e saiba que seu *software* é confiável.

3. Saiba como o *software* será usado.

Uma coisa é a Fonte Aberta ser usada como um sistema operacional ou um servidor BackOffice, e outra coisa completamente diferente é esse mesmo *software* de Fonte Aberta deve ser modificado e incluído num produto. A primeira opção não é problemática; a segunda pode ser. Ao mesmo tempo, assegure-se das restrições típicas do proprietário que proíbem engenharia reversa ou modificações. Enquanto alguns fornecedores de *softwares* proprietários possam permitir tais atividades sob uma licença especial para desenvolvimento ou uma licença comunitária de código-fonte, eles geralmente não permitem as atividades sob suas licenças comerciais gerais. Pode valer a pena categorizar cada item de *software* e os usos permitidos a cada um, por exemplo, aprovados para uso geral apenas em forma executável, ou aprovados para uso no plano do código-fonte em aplicações especializadas ou aplicações modificadas, e os não aprovados seja para que uso for. Finalmente, a natureza do uso é importante, isto é, saber se o *software* será distribuído fora da companhia, potencialmente suscitando restrições de licenciamento de Fonte Aberta.

4. Tenha meios para documentar qual *software* e qual versão desse *software* está em uso.

Possuindo essa informação e tendo pronto acesso a ela o ajudará a assegurar o cumprimento da licença e ao mesmo tempo permitirá aos administradores de TI a habilidade para gerir a arquitetura TI e seu avanço.

5. Exija a documentação de todos os projetos de desenvolvimento de *softwares* internos.

Isso inclui modificações de *softwares* de Fonte Aberta. Tal documentação deve indicar a fonte de qualquer *software* usado como base e que tenha sido modificado, todos os autores do *software* desenvolvido, os projetos anteriores no quais tais programadores trabalharam (tanto para uso interno quanto aqueles destinados a empregadores anteriores), e a identificação de toda propriedade intelectual relacionada já conhecida, particularmente patentes.

Essas são apenas algumas sugestões. Têm a intenção de tratar dessas questões mais comumente encontradas sobre *software*, inclusive de Fonte Aberta.

Para aqueles interessados em conhecer mais sobre Fonte Aberta, sugere-se ler os seguintes sites:

- Free Software Foundation - <<http://www.fsf.org>>
- Open Source Initiative - <<http://www.opensource.org>>
- FAQs técnicos sobre Linux, da IBM:

<<http://www-106.ibm.com/developerworks/linux/library/lfaq/?open&l=252,t=grl,p=LinuxFAQ>>

- Link para *whitepapers* sobre a legalidade da GPL:

<<http://www.newtechusa.com/Viewpoints/GPLLegalityLinks.asp>>

- Referência rápida para escolha de licença de *software* gratuito:

<http://zooko.com/license_quick_ref.html>

- Por que *software* de Fonte Aberta: <http://www.dwheeler.com/oss_fs_why.html>

- Segurança do Linux: <<http://www.linuxsecurity.com>>

Referência

<http://www.nswscl.org.au/journal/51/Mark_H_Webbink.html>

Notas de Rodapé

1. Quando eu falo sobre lei de direitos autorais neste texto, estou discutindo a lei de *copyright* dos E.U.A. como inserida no Título 17 do Código dos E.U.A. Os Estados Unidos são signatários da Convenção de Berna que trata de direitos autorais, e muito da lei de *copyright* dos E.U.A. é bastante similar àquela de outros países signatários de Berna. No entanto, há disposições na lei de direitos autorais nos E.U.A. que são exclusivas dos E.U.A., como o registro de *copyright*. Pessoas em outras pátrias que não os E.U.A. devem consultar o conselho jurídico local especializado em lei de direitos autorais.
2. §1-106, Título 17, Código dos E.U.A.
3. §1-107, Título 17, Código dos E.U.A.

4. 37 CFR 201.26 define *software* de domínio público como um *software* publicamente distribuído com uma renúncia explícita de proteção de *copyright* pelo detentor dos direitos autorais. Como declara a *Free Software Foundation*, *software* de domínio público significa *software* não protegido por *copyright*.
5. Sob o *Judicial Improvements Act* de 1990, que autorizou a criação de um registro nacional de *softwares*, os detentores de *copyright* de *software* podem doar seu *software* ao domínio público consignando-o junto ao Machine-Readable Collections Reading Room da Biblioteca do Congresso americano. 37 Code of Federal Regulations, Parte 201.26 (1991).
6. 17 U.S. Code §101.
7. 17 U.S. Code §103.
8. Para uma discussão mais aprofundada do estado da lei, veja “Software Derivative Work: A Circuit Dependent Determination”, Dan Ravicher, 31 de outubro de 2002, http://www.pbwt.com/Attorney/files/ravicher_1.pdf.
9. 17 U.S. Code §101.
10. Computer Associates Intl. VC. Altai, Inc., 982 F.2d 693 (2nd Cir. 1992).
11. Engineering Dynamics, Inc. v. Structural Software, inc., 26 F.3d 1335 (5th Cir. 1994);
12. Gates Rubber Co. v. Bando CHem. Indust. Ltd., 9 F.3d 823 (10th Cir. 1993); Mitel, Inc. v. Iqtel, Inc., 124 F.3d 1366 (10th Cir. 1997).
13. Bateman v. Mnemonics. Inc., 79 F.3d 1532 (11th Cir. 1996); Mitek Holdings, Inc. v. Arce Engineering Co., Inc., 89 F.3d 1548
14. Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994).

Diagram 1

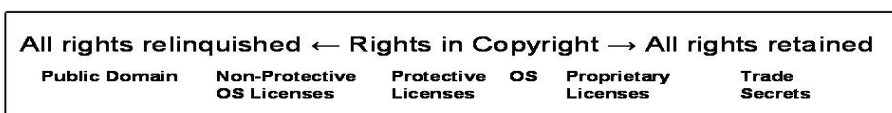


Diagram 2

Rights Granted	Public Domain	BSD	GPL	Windows 98
Copyright retained	No	Yes	Yes	Yes
Right to perform	Yes	Yes	Yes	Yes
Right to display	Yes	Yes	Yes	Yes
Right to copy	Yes	Yes	Yes	No
Right to modify	Yes	Yes	Yes	No
Right to distribute	Yes	Yes	Yes, under same license	No