

## Código-fonte, Linux e Supercomputadores: a relação entre Direito e Tecnologia\*

## Source Code, Linux and Supercomputers: the relationship between Law and Technology

*Marcos Vinício Chein Feres(1); Jordan Vinícius de Oliveira(2); Lorena Abbas da Silva(3)*

1 Pesquisador PQ2/CNPq. Mestre e Doutor em Direito Econômico pela UFMG. Professor Associado da UFJF.  
E-mail: [mvchein@gmail.com](mailto:mvchein@gmail.com)

2 Bolsista DS/CAPES. Bacharel em Direito, Mestre em Direito e Inovação pela UFJF e  
Doutorando em Direito pela UERJ.  
E-mail: [jordanoliveira@tutanota.com](mailto:jordanoliveira@tutanota.com)

3 Bolsista PROBIC/FAPEMIG. Graduanda em Direito pela UFJF. Mestranda em Direito e Inovação pela UFJF  
E-mail: [lorenaabbas@tutanota.com](mailto:lorenaabbas@tutanota.com)

**Revista Brasileira de Direito**, Passo Fundo, vol. 14, n. 1, p. 398-421, Jan.-Abr., 2018 - ISSN 2238-0604

[Recebido: Jul. 01, 2017; Aceito: Dez. 08, 2017]

DOI: <https://doi.org/10.18256/2238-0604.2018.v14i1.2021>

Como citar este artigo / How to cite item: [clique aqui!/click here!](#)

## Resumo

O presente artigo visa analisar, por meio do estudo de caso do núcleo Linux, o marco legislativo brasileiro que regula os programas de computador. O problema de pesquisa investiga se existe uma relação necessária entre a proteção jurídica do programa de computador e o fechamento da tecnologia contida em seu código-fonte. A partir da teoria do viver plenamente a lei e das regras de inferência, desenvolveu-se uma pesquisa empírica e uma análise qualitativa. O principal resultado é de que o fechamento do código-fonte deriva de um sistema proprietário e legalista de concepção do software, o que contradiz a moralidade da aspiração dos direitos de propriedade intelectual.

**Palavras-chave:** Código-fonte. Copyleft. Copyright. Licença Pública Geral. Linux Kernel.

## Abstract

This article aims to analyze the current Brazilian legislation that regulates the software industry, stemming from the case study of the Linux kernel. The research problem consists of investigating if there is a necessary correlation between the legal protection of software and the closure of its source code technology. Empirical research and qualitative analysis are executed in accordance with the living lawfully theory and the rules of inference. The main result of this research is that the restriction of source code derives from a proprietary and legalistic approach of software rights, which contradicts the morality of aspiration of intellectual property rights.

**Keywords:** Copyleft. Copyright. General Public License. Linux Kernel. Source Code.

## 1 Introdução

Este artigo se dedica à relação entre direito e novas tecnologias, especialmente a interação entre as normas de propriedade intelectual e o programa de computador. A análise pretendida foge à dogmática clássica para adentrar nas implicações fáticas dos programas em seu caráter dinâmico e complexo.

O Direito pretende intervir em praticamente todas as instâncias da vida humana estabelecendo diretrizes sobre assuntos diversos, a exemplo dos programas de computador. Para estabelecer uma regulação jurídica eficiente e adaptável aos novos contextos tecnológicos, faz-se necessária a compreensão das peculiaridades desses programas.

Entre essas complexidades, destaca-se o chamado código-fonte. Como afirmam Marengo e Vezzoso (2006, p. 12), esse código é essencial para a compreensão humana de um software<sup>1</sup>, pois revela como ele foi feito e como pode ser editado ou adaptado. De forma análoga, é como se esse código fosse a receita ou a partitura do programa.

A grande peculiaridade do código-fonte é que ele pode ou não ser revelado segundo a vontade de seu criador, sendo essa escolha a que determina o regime jurídico cabível. É o que explica Lemos (2005, p. 72), ao apresentar os regimes de copyright e *copyleft*. Conforme o autor, se o código-fonte for fechado, o regime aplicável é o de copyright, em que todos os direitos sobre o programa, incluindo o acesso ao código, estão reservados ao seu criador. Já o *copyleft* tem como premissa a abertura do código-fonte, o que permite a distribuição e adaptação do programa sem prejuízo da atribuição de créditos ao autor.

Dessa forma, este estudo visa a compreender esses regimes jurídicos a fim de investigar o caso do *kernel*<sup>2</sup> do Linux. O GNU/Linux é um software do tipo sistema operacional<sup>3</sup> de código-fonte aberto, lançado em 1991 a partir da união do núcleo de sistema Linux com o sistema operacional GNU, desenvolvido pelo programador Richard Stallman (FREE SOFTWARE FOUNDATION, 2017).

O plano teórico adotado é o do viver plenamente a lei de Zenon Bankowski (2008), a partir das noções de dever e de aspiração da lei. A técnica metodológica

- 1 Para os fins deste estudo, os termos software e programa de computador estão empregados como sinônimos, sob o fundamento do artigo 1º, inciso IV do *Draft Treaty for the Protection of Computer Software* (WIPO, 1983).
- 2 De acordo com a Linux Foundation, o *kernel* pode ser compreendido como o núcleo do sistema Linux, responsável por gerenciar os componentes físicos do computador, executar os programas do usuário e manter a segurança e a integridade de todo o sistema. Foi lançado em 1991 por Linus Torvalds, que abriu seu código-fonte e, desde então, o projeto reuniu inúmeros colaboradores (LINUX FOUNDATION, 2016, p. 03). Para facilitar a compreensão, o *kernel* seria uma espécie de motor e o sistema operacional corresponderia a todo o automóvel.
- 3 O sistema operacional é um tipo de programa que gerencia os recursos físicos (*hardwares*) de uma máquina (computador, telefone, entre outros) para oferecer os serviços básicos de software (PRESSMAN, 2007, p. 248). Exemplos são o Microsoft Windows, o GNU/Linux ou o Apple Mac OS X.

empregada consiste no uso dos conceitos de regras de inferência, implicações observáveis e hipótese rival, apresentados por Epstein e King (2013).

A escolha deste referencial teórico se justifica devido à sua capacidade de articulação dos elementos jurídicos à complexidade encontrada no mundo real. A técnica metodológica empregada, por sua vez, tem o papel de dar transparência à forma de análise, tornando-a replicável.

O objetivo da pesquisa é o de analisar, a partir do caso do sistema operacional Linux, o marco legislativo referente aos programas de computador tendo por foco o processo de restrição do código fonte.

Cabe, ainda, apresentar o problema que guiou esta pesquisa. Sabe-se que uma abordagem legislativa do tipo “todos os direitos reservados”, como a brasileira, resulta no máximo controle do autor sobre a obra e automático fechamento do código-fonte<sup>4</sup>. Assim, pergunta-se: a restrição do código-fonte constitui elemento necessário para a realização da aspiração do sistema de propriedade intelectual?

A partir das noções de dever e da aspiração da norma (BANKOWSKI, 2008), afirma-se que a restrição do código-fonte deriva de um sistema de direito autoral proprietário e legalista que está em dissonância com a aspiração geral das normas de propriedade intelectual, que é a de incentivo à inovação.

As etapas de pesquisa serão as seguintes: no próximo tópico serão elucidados a técnica metodológica, o referencial teórico e a forma de coleta e utilização dos dados. No item 3, será realizada uma breve revisão bibliográfica sobre softwares, os regimes jurídicos de copyright e de *copyleft* a eles aplicáveis e alguns aspectos da legislação brasileira. No item 4 discutem-se os resultados, as implicações observáveis e a hipótese rival.

## 2 Abordagem metodológica e teórica

A análise de conteúdo aqui empreendida é do tipo qualitativa e utiliza elementos empíricos para estabelecer uma teoria. O conteúdo extraído para a pesquisa é tanto de natureza indireta, como artigos e livros sobre propriedade intelectual e programas de computador, quanto de natureza direta, como relatórios, licenças e estatísticas ligadas ao caso do núcleo do sistema Linux.

Sobre o período de coleta dos dados analisados, os conteúdos extraídos diretamente junto a relatórios e bases de dados foram obtidos e sistematizados no período de 20 de abril a 29 de maio de 2017. Com relação aos conteúdos indiretos, como referenciais bibliográficos, à exceção de textos metodológicos, teóricos e artigos previamente utilizados em estágios anteriores desta pesquisa, a revisão de literatura específica para a elaboração deste artigo foi consultada ao longo do primeiro semestre de 2017, com a data limite de 01 de junho.

---

4 O código-fonte e outros conceitos técnicos, serão melhor explicitados no item 3 deste estudo. Por enquanto, é necessário entender que ele é um elemento basilar para o desenvolvimento e a edição de um programa e a sua tecnologia pode ser ocultada conforme o interesse de seu proprietário.

Inicialmente, faz-se necessário explicar os conceitos de inferência, implicações observáveis e hipóteses rivais, bem como a maneira de extração e uso dos dados em cada um desses itens.

Segundo Epstein e King (2013, p. 36ss), a inferência descritiva consiste na elaboração de uma teoria a partir do emprego de fatos conhecidos para compreender um fenômeno inicialmente desconhecido. Portanto, para a sua realização é imprescindível distinguir qual é o fato conhecido da pesquisa (amostra) e como ele auxilia para a compreensão do fato desconhecido (população).

A partir da realização da inferência, amparada em elementos empíricos, tem-se uma resposta ao problema de pesquisa. Desse ponto em diante, estabelece-se quais seriam as implicações observáveis da teoria, o que significa identificar seus desdobramentos no mundo real (EPSTEIN; KING, 2013, p. 79).

Por fim, o último elemento trabalhado será o da hipótese rival, que de acordo com Epstein e King (2013, p. 96) corresponde às “explicações que não fecham com a teoria que [os pesquisadores] estão oferecendo”.

Dessa forma, pelo caminho acima traçado, serão analisados elementos empíricos com vistas a verificar o comportamento do objeto pesquisado na realidade e possíveis explicações alternativas ao fenômeno. Sobre os elementos empíricos, é preciso ainda ressaltar quais são eles e o porquê da utilização deles neste estudo.

Adotando-se como ponto de partida o estudo de caso do software aberto e livre<sup>5</sup> Linux, a hipótese do estudo é a de que a proteção jurídica do programa de computador não tem como condição necessária o fechamento de seu código-fonte. Diante da óptica teórica adotada, argumenta-se que a abertura de seu código-fonte simboliza a possibilidade de compartilhamento de conhecimento e de cooperação entre os desenvolvedores nesse mercado.

Para a realização da inferência, analisar-se-á a licença adotada pelo *kernel* Linux e dados de sua participação no mercado de software. Esses dados foram retirados do relatório sobre desenvolvimento do programa, publicado pela Linux Foundation<sup>6</sup>. A utilização desse relatório é fundamental para a compreensão de como esse programa integra desenvolvedores e empresas em plena sociedade de mercado. Ademais, serão analisados alguns casos pontuais de empresas que trabalham com o *kernel* Linux.

Esses dados, apresentados na tabela 1, retratam as principais empresas, por nome, que colaboraram para o desenvolvimento do *kernel* Linux. Assim, as categorias presentes no relatório original - “nenhuma” (quando o desenvolvedor não está vinculado a qualquer empresa), “desconhecida” (quando não é possível determinar

5 De acordo com Stallman (FREE SOFTWARE FOUNDATION, 2016), existem diferenças práticas entre o software de código aberto e o software livre. Enquanto o primeiro assegura primariamente, mas não apenas, que o código-fonte de um programa seja disponibilizado, o segundo revela uma abordagem filosófica de manter o programa e versões modificadas sempre livres. Logo, um software livre sempre será aberto, mas um software aberto pode não ser livre. O Linux, objeto deste trabalho, é um software livre.

6 Organização fundada em 2000, responsável pelo software operacional Linux, um dos maiores projetos de desenvolvimento de software da história. Disponível em: <<https://www.linuxfoundation.org/about>>.

se o desenvolvedor faz parte de alguma companhia) e “consultores” (quando o desenvolvedor é contratado de maneira esporádica e para um projeto específico) - foram retiradas da tabela por não estarem vinculadas definitivamente a nenhuma empresa (LINUX FOUNDATION, 2016).

Uma vez realizada e discutida a inferência, o próximo passo será o da exploração das implicações observáveis da teoria nela afirmada. Essas implicações estão agrupadas em três planos: jurídico, mercadológico e social. No plano jurídico são contrapostas as oito licenças mais populares entre as comunidades de softwares abertos e livres, segundo dados da *Open Source Initiative*<sup>7</sup> e da *OSS Watch*<sup>8</sup>.

O uso das informações das organizações *Open Source Initiative* e *OSS Watch* se justifica por demonstrarem a variedade das licenças de *copyleft*. A primeira organização é a responsável pela padronização global dessas licenças. Já a segunda oferece uma importante ferramenta para a comparação dessas licenças, chamada de *License Differentiator*, voltada para o universo dos programadores e usuários de softwares de código-fonte aberto e livre.

De acordo com a *Open Source Initiative* (2017a), existem 75 licenças de código aberto catalogadas, sem computar as que já foram substituídas, voluntariamente aposentadas ou em processo de análise. Dessas, são oito as licenças mais populares ou com uma comunidade mais ativa de usuários, listadas na tabela 2.

A partir da leitura completa dessas licenças, a análise teve como foco as características sobre o tipo de *copyleft* e sua relação com o código-fonte. Para tanto, o serviço *License Differentiator* foi utilizado subsidiariamente por padronizar os principais aspectos de cada licença. O acesso à ferramenta *License Differentiator* deu-se da seguinte maneira: uma vez selecionado o link<sup>9</sup> do serviço, no campo “*Choice One*”, a opção “*Only include licenses that are popular and widely used or with strong communities*” foi escolhida. Com essa seleção, constam nos resultados apenas as licenças amplamente utilizadas nas comunidades de softwares livres e abertos, segundo a *Open Source Initiative*. Assim, foram obtidas as oito licenças<sup>10</sup> mais populares da *Open Source Initiative* (OSS WATCH, 2014, tradução livre).

7 Corporação sem fins lucrativos voltada para o universo do software de código-fonte aberto. Entre suas principais funções está a de padronizar as licenças de softwares abertos ao redor do mundo. Disponível em: <<https://opensource.org/>>.

8 Site formado em 2003, por especialistas em softwares livre e de código aberto, vinculados à Universidade de Oxford. Visa a fornecer auxílio para o uso, desenvolvimento e licenciamento de programas de código aberto, de maneira imparcial e revisada (OSS WATCH, 2013). Disponível em: <<http://oss-watch.ac.uk>>.

9 Disponível em: <<http://oss-watch.ac.uk/apps/licdiff/>>.

10 Segundo a Free Software Foundation, são nove as licenças mais populares, mas considerando-se as duas versões da licença BSD como diferentes. O relatório da OSS Watch as mescla em uma única categoria, por isso o número de oito licenças. A licença BSD, ou *Berkley Software Distribution*, é uma licença de *copyleft* que possui duas versões: a de duas cláusulas (*BSD 2-Clause*) e a de três cláusulas (*BSD 3-Clause*). A grande diferença entre ambas é que a de três cláusulas tem uma adição à de duas cláusulas para vetar o uso da marca ou distinção do distribuidor ou do grupo responsável pela licença original (OPEN SOURCE INITIATIVE, 2017b; 2017c).

O segundo e último passo da coleta sobre as licenças ocorreu no campo “*Choice Two*”, “a” e “b”, que oferece uma classificação sobre o manuseio do código-fonte a partir da instrução da licença. Os cinco indicadores restantes – “*choice three*” a “*choice seven*” – trazem informações variadas, como a indicação de jurisdição competente para resolução de conflitos ou a possibilidade de uso do programa para patentes (OSS WATCH, 2014, tradução livre), que não se encaixam no escopo deste estudo.

No plano mercadológico, serão analisados alguns elementos adicionais do relatório da Linux Foundation, não empregados diretamente na inferência. Esses dados, contidos na tabela 3, revelam desdobramentos sobre como se dá a cooperação no projeto Linux e porque ela é importante para os paradigmas da propriedade intelectual.

No plano da sociedade em geral, será estudado o caso dos chamados supercomputadores e a sua conexão com o projeto Linux. Neste ponto, os dados analisados, contidos no gráfico 1, foram retirados diretamente do site Top500<sup>11</sup> e são importantes para a compreensão sobre como o Linux, a partir de seu código-fonte aberto, gera benefícios simultâneos ao programador e à sociedade.

Uma vez apresentadas as fontes das implicações observáveis, vale elucidar como se deu a escolha dos estudos que compõem a hipótese rival. Os quatro estudos escolhidos apresentam uma teoria concorrente à defendida neste trabalho. A seleção dos textos guiou-se pelos tipos de abordagens utilizadas, de natureza tanto jurídica quanto econômica para o fechamento do código-fonte.

Explicados os passos metodológicos, passa-se à explicação do referencial teórico adotado. A teoria utilizada para a análise é a de viver plenamente a lei, de Zenon Bankowski (2008, p. 71-9), a partir dos conceitos de dever e de aspiração das normas. Como afirma o autor, o dever que uma regra expressa não pode ser analisado somente por sua literalidade, mas em consonância com a sua motivação original.

Dessa forma, quando os componentes da norma, dever e aspiração, estão em equilíbrio, tem-se que ela cumpre seu papel social, já quando eles sofrem de dissonância, verifica-se uma necessidade de renovação da lei com fins de que seus pressupostos sejam cumpridos (BANKOWSKI, 2008, p. 81).

Esse referencial teórico é importante, pois possibilita o questionamento do papel das normas de propriedade intelectual. Eles permitem questionar quais as aspirações do direito autoral aplicado ao software e indagar sobre a capacidade de renovação da lei para se adaptar aos novos contextos tecnológicos em que se vive.

### **3 Gestão do conhecimento: software, copyright e *copyleft***

A análise do contexto legal que abarca o software no Brasil desenvolvida neste tópico não é sistemática e também não pretende esgotar aspectos legislativos.

11 Projeto lançado em 1993, responsável por reunir estatísticas sobre os 500 computadores mais potentes do mundo. Website oficial: <<https://www.top500.org/project/>>.

O objetivo é o de apontar apenas a forma de proteção jurídica e licenciamento e as implicações que essa proteção gera para o desenvolvimento do programa de computador. Dessa maneira, destacam-se trechos do artigo 7º da Lei de Direitos Autorais (Lei nº 9.610/98) e dos artigos 2º, 6º da Lei de Programas de Computador (Lei nº 9.609/98), que regulam o software.

A Lei nº 9.609/98 elege como regime de proteção aplicável ao software o mesmo das obras literárias, o copyright, ressalvadas algumas particularidades<sup>12</sup>. Prevê, ainda, um rol limitado de faculdades ao uso do programa, o que traduz a lógica de todos os direitos reservados<sup>13</sup> (BRASIL, 1998a). A Lei nº 9.610/98, por sua vez, é aplicável de maneira residual, para os casos afins não previstos pela Lei de Programas de Computador<sup>14</sup> (BRASIL, 1998a; 1998b).

Ocorre, entretanto, que o emprego da lógica de todos os direitos reservados para o universo criativo e tecnológico do software acarreta inúmeras dificuldades. Marengo e Vezzoso (2006, p. 12) expõem talvez o maior desses problemas, o da engenharia reversa. De acordo com os autores, a engenharia reversa é o processo para obtenção do chamado *know-how* de um dado bem: como este foi desenvolvido, quais as funcionalidades dele, os problemas técnicos que enfrenta, entre outras questões. A importância da engenharia reversa reside na possibilidade de propagação do saber, a partir da compreensão de como algo foi feito e de como seria possível aprimorá-lo ou construir novos itens a partir dele.

---

12 Art. 2º O regime de proteção à propriedade intelectual de programa de computador é o conferido às obras literárias pela legislação de direitos autorais e conexos vigentes no País, observado o disposto nesta Lei.  
§ 1º Não se aplicam ao programa de computador as disposições relativas aos direitos morais, ressalvado, a qualquer tempo, o direito do autor de reivindicar a paternidade do programa de computador e o direito do autor de opor-se a alterações não-autorizadas, quando estas impliquem deformação, mutilação ou outra modificação do programa de computador, que prejudiquem a sua honra ou a sua reputação.

13 Art. 6 – Não constituem ofensa aos direitos do titular de programa de computador:  
I- a reprodução, em um só exemplar, de cópia legitimamente adquirida, desde que se destine a cópia de salvaguarda ou armazenamento eletrônico, hipótese em que o exemplar original servirá de salvaguarda;  
II- a citação parcial do programa de computador, para fins didáticos, desde que identificados o programa e o titular dos direitos respectivos;  
III- a ocorrência de semelhança de programa e outro, preexistente, quando se der por força das características funcionais de sua aplicação, da observância de preceitos normativos e técnicos, ou de limitação de forma alternativa para a sua expressão;  
IV- a integração de um programa, mantendo-se suas características essenciais, a um sistema aplicativo ou operacional, tecnicamente indispensável às necessidades do usuário, desde que para o uso exclusivo de quem a promoveu (BRASIL, 1998b).

14 Art. 7º São obras intelectuais protegidas as criações do espírito, expressas por qualquer meio ou fixadas em qualquer suporte, tangível ou intangível, conhecido ou que se invente no futuro, tais como:  
[...] XII - os programas de computador; [...]  
§ 1º Os programas de computador são objeto de legislação específica, observadas as disposições desta Lei que lhes sejam aplicáveis (BRASIL, 1998a).



Os autores apontam que o problema da aplicação do copyright ao software é que ele inclui a proteção ou fechamento do código-fonte, onde fica a linguagem de programação fundamental para compreensão do programa. Uma vez fechado o código, nenhum outro programador poderá entender sua elaboração, realizar modificações ou mesmo explorá-lo sem restrições, ainda que para fins educacionais (MARENGO; VEZZOSO, 2006, p. 12).

Marengo e Vezzoso (2006, p. 12-5) apontam que a grande contradição da aplicação do direito autoral ao software é que, embora se utilize do mesmo licenciamento das obras literárias e artísticas, o seu *know-how* pode ser ocultado. Desse modo, a compreensão das técnicas narrativas, do material e da estrutura textual de uma obra literária ou as notas, instrumentos e técnicas de execução de uma música podem ser compreendidas pela engenharia reversa. Já o aprendizado das técnicas de programação, a criação de soluções específicas para problemas técnicos enfrentados ou a customização e a modificação do programa de computador não são viabilizados pelo copyright.

Como o setor de programas de computador depende de inovação e de fluxo de conhecimento e a legislação autoral em muitos casos os desestimula, as licenças de *copyleft* surgiram a partir de uma adaptação do antigo sistema de copyright. Como explica Lemos (2005, p. 72-3), o *copyleft* nasceu como uma resposta à inflexibilidade do copyright, possibilitando o acesso ao código-fonte do software e viabilizando a liberdade de uso, modificação, criação, cópia e distribuição do programa, sem que os direitos do autor fossem violados.

Ademais, é necessário frisar que o simples fato de um software ser aberto ou de código livre não inibe a sua exploração comercial. Consoante Lemos (2005, p. 76), alguns dos principais modelos, dentro de uma crescente gama de possibilidades envolvendo a exploração econômica dos softwares livres, são: a) a propagação do software gratuitamente, seguida de assessoramento remunerado, com suporte ou customização às especificidades do cliente; b) a projeção e a conquista em nível de mercado acompanhada da venda de produtos relacionados ao software; c) a venda de um hardware já contando com o software instalado, diminuindo os custos totais ao cliente e; d) a possibilidade de venda de produtos que acompanhem o software, como cursos ou ofertas de aperfeiçoamento.

É importante ressaltar que as licenças *copyleft* não negam o copyright, pois o utilizam como premissa: somente após assegurados todos os direitos sobre o programa é que o *copyleft* surge como uma forma de licenciamento flexível, para oferecer o direito de edição, distribuição e adaptação do software (DUSOLLIER, 2003, p. 286).

As licenças de *copyleft*, por toda sua complexidade, suscitam alguns debates sobre o papel da propriedade intelectual aplicada ao software. No contexto norte-americano, Brow (2010, p. 763-73) lista alguns problemas do uso dessas licenças sem o estabelecimento de uma reforma legislativa em prol do *copyleft*. Entre os elementos

por ele citados, destacam-se, especialmente ao contexto brasileiro, as questões da jurisprudência, do consentimento e da compatibilidade entre licenças.

Sobre a jurisprudência, Brow (2010) argumenta que os tribunais estão acostumados a lidar com problemas relacionados à propriedade intelectual dos softwares pelas legislações ora vigentes, que por sua vez possuem uma lógica proprietária. Dessa forma, eventuais litígios seriam tratados pelos paradigmas do copyright, o que poderia gerar erros e desentendimentos (BROW, 2010, p. 763). No Brasil, um equívoco de fundamento na jurisprudência do Superior Tribunal de Justiça<sup>15</sup> já demonstra que o argumento do autor merece atenção.

O segundo problema é o do consentimento. Como atualmente as licenças *copyleft* não são uniformes e não possuem uma legislação especial que as regule, o consentimento é automático: ao descompactar o software para uso, navegar pela página ou clicar para download (*shrink-wrap; browse-wrap e click-wrap licenses*) o pressuposto é de que o usuário concorda com os termos da licença, sem a necessária ciência de suas cláusulas.

O terceiro problema listado pelo autor é o da compatibilidade entre as licenças. No caso de adaptação de duas obras para a criação de uma terceira, por exemplo, é possível a ocorrência da violação dos termos dispostos, caso essas obras estejam sujeitas a cláusulas contrárias (BROW, 2010, p. 772-3).

Assim, o autor defende a reforma legislativa para comportar o *copyleft* como medida mais eficiente para os fins de mudar as atuais estruturas do copyright. Ele aponta que uma reforma padronizaria a possibilidade de uso de alguns e não o de todos os direitos reservados ao programa (BROW, 2010, p. 779-83).

Estabelecido o contexto e a crítica às licenças *copyleft*, cumpre elaborar uma análise direta sobre sua aplicação no direito brasileiro. Esclarece-se que, embora a lei brasileira de programas de computador não trate especificamente sobre os softwares abertos e livres, ela estabelece em seu artigo 9º que o licenciamento é a forma de uso do

---

15 O caso tratava de dois programas *freeware* explorados indevidamente por uma empresa. Como a própria decisão explica (BRASIL, 2013), o *freeware* é um software colocado à disposição de forma gratuita, o que não significa que seu uso comercial por terceiros seja autorizado. A empresa violadora dos direitos colocou os programas, dois jogos, em uma mídia e os comercializou em conjunto com uma revista. Embora a decisão de condenar a empresa pareça ter sido acertada, uma vez que não havia autorização para a venda, há uma confusão entre os termos *freeware* e software livre e uma remissão equivocada a um dispositivo da lei de softwares. A utilização indiscriminada dos termos como sinônimos é problemática, pois *freeware* e software livre possuem regramentos diversos. O *freeware* é um programa gratuito, mas isso não significa que o código-fonte estará disponível. Já o software livre, possui o código-fonte aberto e seus termos podem conter a autorização para a exploração comercial. Sobre o segundo problema, a decisão cita o §5º do artigo 2º da Lei 9.609/98, que trata do aluguel comercial do programa. Ocorre, entretanto, que o fundamento jurídico é equivocado: os programas foram vendidos e não alugados pela revista. Na prática, o julgador aplicou as normas de aluguel da legislação, quando deveria ter observado a licença do programa. Para a decisão em questão, todavia, a confusão não trouxe maiores problemas.

programa em território nacional<sup>16</sup> (BRASIL, 1998a). Logo, embora essa lei não legisle sobre os softwares abertos, o teor do artigo supracitado endossa o licenciamento que, por sua vez, pode ser por *copyleft*.

Diante de todos esses pontos, pode-se aplicar a óptica teórica de Bankowski (2008) para analisar as normas de propriedade intelectual relativas ao programa de computador. Ao se analisar o copyright, percebe-se que o dever contido nessa norma é o de proteger uma criação de apropriações indevidas e que sua aspiração mais ampla é de incentivar os autores em sua capacidade inventiva, como estímulo à inovação.

Ocorre, contudo, que tais pressupostos sofreram desvirtuações e o dever de proteção se tornou excessivo, aplicando-se automaticamente a qualquer programa e impedindo o acesso de terceiros ao conhecimento. Dessa forma, as licenças de *copyleft* surgiram como uma busca de adequação do dever de proteção à aspiração ao estímulo inventivo, sem prejudicar o autor ou terceiros em busca do saber do programa.

#### 4 O caso *kernel linux*: construindo a inferência descritiva

Estabelecidos os aspectos concernentes à proteção jurídica do software, este tópico se dedica ao núcleo do sistema Linux. Aqui serão verificados certos traços de interação entre as empresas que apostam no código-fonte aberto. Antes de iniciar a análise dos dados da tabela 1, faz-se pertinente esclarecer o tipo de proteção jurídica desse software e seus reflexos.

A licença jurídica escolhida por Linus Torvalds para o seu programa *kernel* é a GPL, ou *General Public License*, na versão 2.0. O principal aspecto da licença GPL 2.0 é que ela assegura a liberdade do código-fonte ao mesmo tempo em que exige que obras dele derivadas se mantenham livres. Quem utiliza um software, como o *kernel* Linux, deve deixar livre o código-fonte das modificações realizadas a partir dele. Assim, o código original e os códigos de outros projetos que dele derivam estarão sempre disponíveis para consulta de quaisquer interessados (LINUX KERNEL ORGANIZATION, 1991). Tendo em vista que a licença GPL 2.0 demonstra a possibilidade de usar os direitos de propriedade intelectual em favor da disseminação do conhecimento, passa-se então para a análise das empresas que mais colaboraram com o processo de desenvolvimento do *kernel* Linux.

---

16 Art. 9º O uso de programa de computador no País será objeto de contrato de licença (BRASIL, 1998a).

**Tabela 1.** As 10 Empresas que mais Contribuíram ao Desenvolvimento das Versões 3.19 a 4.7 do Kernel do Linux\*

Empresa	Número de Modificações	Porcentagem de Modificações
Intel	14.384	12.9%
Red Hat	8.987	8.0%
Linaro	4.515	4.0%
Samsung	4.338	3.9%
SUSE	3.619	3.2%
IBM	2.995	2.7%
Renesas Electronics	2.239	2.0%
Google	2.203	2.0%
AMD	2.100	1.9%
Texas Instruments	1.917	1.7%

\* Retiradas as Categorias “nenhuma” (7.7%), “desconhecida” (6.8%) e “consultores” (2.6%).

**Fonte:** The Linux Foundation (2016, p. 12, adaptado e traduzido livremente).

A tabela acima serve de substrato para investigar a inserção de mercado do software Linux. O perfil das empresas citadas é basicamente do setor de tecnologia, variando desde fabricantes de hardware, como a Intel e a IBM, a desenvolvedoras voltadas para software de código-fonte aberto, como a Red Hat. Com um olhar mais atento, percebe-se que até companhias concorrentes, como as indústrias do setor de hardware Intel e AMD, contribuíram para os incrementos de um mesmo sistema.

Um dado que chama atenção na tabela é o da categoria “nenhuma” (citada ao rodapé da tabela), na qual os desenvolvedores não estão vinculados a nenhuma empresa e não recebem salários. Segundo o relatório da Linux Foundation, essa categoria já representou mais de 14% do desenvolvimento do núcleo Linux em 2012, mas no último ano atingiu apenas 7.7% de participação. As justificativas da organização para essa diminuição é o maior interesse das empresas em projetos abertos e, conseqüentemente, maior contratação de programadores com esse perfil (LINUX FOUNDATION, 2016, p. 12).

A partir da tabela 1, pode-se citar dois exemplos de empreendimentos expressivos de mercado, baseados no kernel Linux: o software Android e a empresa de sistemas Red Hat.

O Android é um projeto de código-fonte aberto, lançado pela Google. Ele é voltado para sistemas operacionais de aparelhos portáteis, como celulares e tablets, e utiliza como base o *kernel* Linux (ANDROID OPEN SOURCE PROJECT, 2017). O projeto pode ser considerado um sucesso pela sua ampla disseminação e uso. Segundo o StatCounter<sup>17</sup>, considerando o mercado mundial de *smartphones*, o

17 Serviço online de monitoramento e análise de estatísticas de uso presente em mais de 2 milhões e 500 páginas e que conta com processos de revisão de qualidade (STATCOUNTER, 2017a) Website oficial: <<http://gs.statcounter.com>>.

Android atingiu o índice de uso de 71,84% em maio de 2017. Esses números são expressivos se comparados com os seus três principais concorrentes: 19,87% de uso do sistema operacional IOS, 1,22% do sistema Nokia e 0,96% do sistema Windows (STATCOUNTER, 2017b<sup>18</sup>). O Android atraiu a empresa Intel, líder na colaboração das versões do *kernel* Linux, que dedicou um projeto inteiramente voltado para o desenvolvimento aberto de componentes físicos para aparelhos com esse sistema (INTEL CORPORATION, 2017).

Esse dado chama atenção, sobretudo pela constatação de Feres et al. (2017, p. 85), de que o mercado de computadores pessoais (desktops e notebooks) é dominado pelo sistema fechado Microsoft Windows. Assim, o caso do sistema Android demonstra que é possível conquistar usuários a partir do código aberto.

Outro exemplo da viabilidade de abertura do código-fonte é o da empresa Red Hat<sup>19</sup>, segunda maior colaboradora do *kernel* Linux. A Red Hat trabalha com soluções tecnológicas baseadas em código aberto e um de seus principais projetos é o sistema operacional *Red Hat Enterprise Linux*. Atualmente a empresa presta serviços junto a mais de 90% das maiores empresas norte-americanas presentes na lista Fortune 500<sup>20</sup> (RED HAT INC, 2017).

O caso da Red Hat é tão expressivo que foi abordado por Boldrin e Levine (2007, p. 07). Os autores trazem uma importante evidência de que a flexibilização dos termos do copyright pelo *copyleft* não se desdobra em necessária perda de mercado.

Segundo os autores, em 2002 a empresa customizava e vendia pacotes de seu software de sistema operacional pelo valor de 59,95 dólares. Como ela trabalha com software aberto, seu código-fonte estava disponível para todos, inclusive para eventuais competidores. Assim, as empresas *Hcidesign* e *Linuxemporium*, “piratas legalizadas”, adaptaram suas versões do sistema oferecido pela Red Hat e ingressaram no mercado vendendo-os por cerca de um terço do preço original: apenas 16 dólares. Dessa maneira, como a Red Hat teria continuado no mercado?

A resposta dada pelos pesquisadores é a de que, além de vender mais pacotes de programa do que suas concorrentes, a Red Hat ganhou notoriedade de mercado por ser a criadora original do software adaptado em Linux e por possuir experiências em soluções voltadas para o seu produto. Ainda de acordo com Boldrin e Levine (2007, p. 07-8), em 2006, quatro anos após a primeira análise, a empresa continuou sólida e líder no mercado, posição não compartilhada pelas suas concorrentes diretas.

Atualizando o caso da Red Hat para o corrente ano, verifica-se que ela não somente permanece no mercado de programas de computador, como apresenta resultados econômicos expressivos. Conforme a NASDAQ (2017), a Red Hat saiu

18 Os dados informados foram consultados em 28 de maio de 2017. Dessa forma, os valores podem sofrer pequenas variações até o fechamento do mês.

19 Website oficial: <<https://www.redhat.com/pt-br>>.

20 Website oficial: <<http://beta.fortune.com/fortune500>>.

dos 278 mil e 330 dólares de receitas anuais, no ano fiscal de 2006, para a marca de 2 bilhões e 52 milhões de dólares no ano fiscal de 2016, com previsão de crescimento para 2 bilhões e 411 milhões em 2017.

Portanto, os dados acima demonstram que a abertura do código-fonte, do ponto de vista comercial, não é óbice à exploração mercadológica do software, colocando até companhias concorrentes em regime de colaboração indireta.

Diante disso, a partir dos conceitos de dever e de aspiração da norma (BANKOWSKI, 2008, p. 71-81), infere-se descritivamente (EPSTEIN; KING, 2013) que a restrição do código-fonte do software pelo copyright revela o seu caráter proprietário e puramente amparado na noção de dever dessa norma, mas que está em dissonância com a sua aspiração máxima de incentivo à inovação na sociedade tecnológica em que se vive.

Essa inferência é descritiva, pois se utilizam os dados do *kernel* Linux (fenômeno conhecido) para compreender, de maneira geral, os projetos baseados em licenças que incentivam o código aberto (fenômeno desconhecido).

Considerando-se a teoria de viver plenamente a lei a partir da aspiração da norma (BANKOWSKI, 2008), o modo de licenciamento por *copyleft* revela a capacidade de renovação do direito de tempos em tempos para comportar as motivações que o fizeram surgir. Se o dever de proteção do copyright surge para viabilizar a aspiração de incentivo ao criador para inovar, a sua aplicação atual se distancia dessa motivação por impossibilitar o acesso ao conhecimento contido no código-fonte.

Deve-se destacar como maior limitação da inferência descritiva o fato de não ser possível investigar todo o setor de softwares de código-fonte aberto (população do estudo). Todavia, isso não significa que os padrões do caso Linux, como o emprego de licenças *copyleft* e a exploração múltipla do programa por mais de um agente de mercado, não possam ser replicados em outros projetos de código aberto. Contudo, as dimensões desses projetos podem variar em relação ao número de programadores e eventuais empresas participantes.

#### 4.1 Implicações observáveis da teoria

Uma vez estabelecida a teoria de que, em plena sociedade de mercado, a restrição do código-fonte está em dissonância com as aspirações da propriedade intelectual, é relevante listar algumas implicações observáveis dessa afirmação. Como já explicado na metodologia, as implicações observáveis são elementos convergentes à hipótese estabelecida pelo pesquisador. Neste estudo, estão divididas em três planos: o jurídico, o mercadológico e o social.

No plano jurídico, partindo da premissa teórica atestada, devem ser observáveis licenças de código-fonte aberto que possuam a capacidade de interagir com empreendimentos de código fechado. No plano mercadológico, maiores nuances de uma parceria colaborativa entre empresas e sujeitos em nome da inovação também devem

ser verificáveis. Por fim, no plano social, é imprescindível constatar os benefícios para a sociedade em geral a partir da flexibilização do dever de fechamento do código-fonte.

Para começar com o âmbito jurídico, a tabela 2 demonstra as principais licenças de *copyleft* empregadas nas comunidades de softwares abertos e livres. Sua metodologia de elaboração já foi explicada no tópico 2.

**Tabela 2.** Tipo de Licenças dos Softwares de Código-fonte Aberto

Nome da Licença	Classificação ou Tipo da Licença
Licença Apache 2.0	Permissiva
Licenças BSD (2-Clause BSD e 3-Clause BSD)	Permissiva
Licença MIT	Permissiva
Licença de Desenvolvimento e Distribuição Comum (CDDL)	<i>Copyleft</i> Fraco
Licença Pública Eclipse	<i>Copyleft</i> Fraco
Licença Pública Mozilla	<i>Copyleft</i> Fraco
Licença Pública Geral Menor 2.1 (LGPL)	<i>Copyleft</i> Fraco
Licença Pública Geral 2.0 (GPL)	<i>Copyleft</i> Forte

Fonte: OSS Watch (2014).

Seguindo a classificação proposta pela OSS Watch (2014, tradução livre), existem três categorias de licenças de programas baseados em código aberto: a “permissiva”, a de “*copyleft* fraco” e a de “*copyleft* forte”.

Na licença permissiva, não existem normas específicas de licenciamento para a reutilização do código-fonte original. É como se o criador do programa não determinasse critérios de licenciamento para as criações derivadas de seu código, deixando-as livres ao discernimento de outros programadores (OSS WATCH, 2014).

Na licença de *copyleft* fraco, caso o programador tenha o intuito de distribuir seu programa a partir de uma modificação de um software aberto, apenas as modificações do código principal precisam estar disponíveis pelos mesmos termos da licença original. Já o novo código ou a compilação podem estar sujeitos a outra licença, sem que necessariamente sejam disponibilizados (OSS WATCH, 2014). Em termos práticos, isso significa que um software aberto com licença de *copyleft* fraco, como a LGPL, pode ser utilizado para um fim proprietário, em que o novo produto não precise ser revelado, mas somente as modificações do código original.

Por fim, na licença de *copyleft* forte como a GPL, já analisada, as partes original e derivada do código devem estar disponibilizadas pela mesma licença inicial, o que garante que elas sempre fiquem disponíveis (OSS WATCH, 2014).

Essas implicações demonstram que as variadas licenças não inviabilizam projetos derivados de código-fonte fechado de maneira que, quando se julgar necessário, essa ação poderá ser tomada. Elas revelam, portanto, o caráter adaptável do *copyleft* e capaz de comportar a aspiração maior de incentivo à inovação em variados contextos.

Passando para implicações adicionais da esfera mercadológica, a tabela 3 abaixo, demonstra alguns pontos importantes sobre o uso do código-fonte e a capacidade criativa e colaborativa do programa aberto.

**Tabela 3.** Os Números do Desenvolvimento das Versões 3.19 a 4.7 do Kernel do Linux

Versão	Data	Dias	Modificações	Linhas de Código	Correções de Bugs	Programadores	Empresas
3.19	08/02/15	63	12.461	19.134.418	4.833	1.451	230
4.0	12/04/15	63	10.346	19.316.184	873	1.458	214
4.1	21/06/15	70	11.916	19.516.298	757	1.539	238
4.2	30/08/15	70	13.694	20.315.522	1.643	1.591	251
4.3	01/11/15	63	11.894	20.625.248	903	1.625	211
4.4	10/01/16	70	13.071	20.865.919	618	1.575	220
4.5	13/03/16	63	12.080	21.158.348	1.892	1.537	231
4.6	15/05/16	63	13.517	21.426.491	973	1.678	243
4.7	24/07/16	70	12.283	21.724.632	550	1.582	221

**Fonte:** The Linux Foundation (2016, p. 05-9, adaptado e traduzido livremente).

Os dados iniciais revelam um projeto coletivo de desenvolvimento, com múltiplas modificações e correções no núcleo do sistema. Verifica-se uma média de colaboração superior a 1.550 programadores distribuída pelas nove versões abarcadas, que duraram de 6 a 7 semanas. Dois aspectos devem ser ressaltados: o perfil desses programadores do *kernel* Linux e o modo como o projeto lida com a correção de erros (*bugs*).

De acordo com Xia (2009, p. 06), o desenvolvimento do *kernel* do Linux opera a partir de uma rede tríplice interconectada: formada por uma rede de comunicação, uma rede humana e uma rede de conhecimento. A rede de comunicação é basicamente a internet, que viabiliza a comunicação, colaboração e a difusão da informação. A rede humana consiste nos próprios programadores que, ao mesmo tempo em que produzem o “produto” *kernel*, também geram conhecimento sobre o processo de desenvolvimento em si. Esse processo fica disponível a todos, inclusive aos que não participaram do projeto e se integram posteriormente. Já a rede de conhecimento constitui no saber agregado no processo de desenvolvimento, angariando resoluções para problemas de computação variados.

Lee e Cole (2003, p. 642-4) afirmam que o caso de desenvolvimento do núcleo do Linux é um dos expoentes do modelo coletivo de criação do conhecimento, e também observam três características imanentes ao projeto. A primeira característica se refere à forma das normas de criação do conhecimento. Elas são mais flexíveis, desde os direitos de propriedade intelectual até a maneira de promover a confiança, o compartilhamento de informações e a correção de erros. A segunda característica trata da estrutura de relacionamento entre os programadores, não hierárquica, mas formada



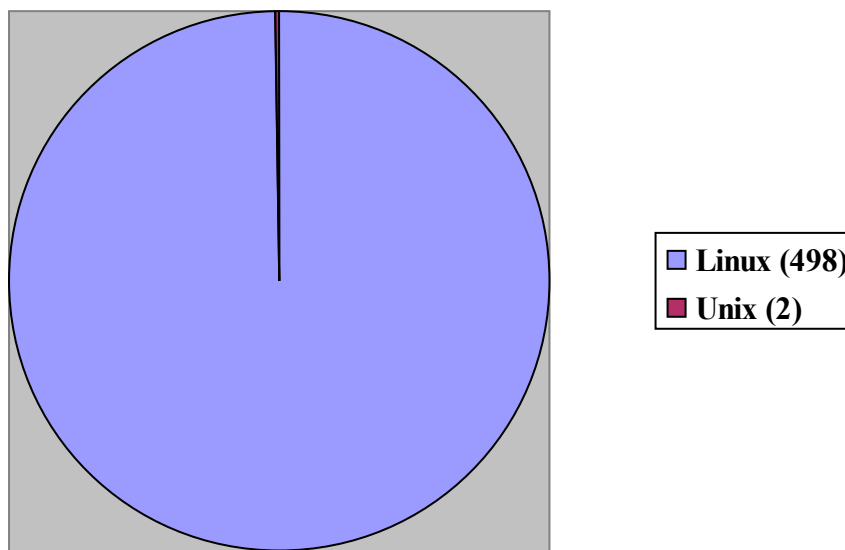
em uma estrutura de dois níveis. Há um núcleo principal (do qual fazem parte o líder do projeto e seus mantenedores mais próximos) e um núcleo periférico (do qual estão os times de desenvolvimento e de relatório de erros). O relacionamento harmônico entre essas camadas é essencial para o desenvolvimento do programa. Por fim, segundo os autores, o terceiro aspecto é o da tecnologia: o uso da internet é utilizado para transpassar os limites físicos da organização, que conta com a cooperação advinda de várias partes do globo.

Em conjunto, essas implicações jurídicas e econômicas demonstram uma capacidade de organização coletiva que maximiza a cooperação em busca da inovação. A estruturação coletiva do projeto *kernel* Linux demonstra uma predisposição ao fluxo do conhecimento, que é essencial para a capacidade criativa do ser humano. Uma última implicação observável a partir do caso Linux trata dos incrementos que o programa aberto pode trazer à sociedade em geral. O exemplo escolhido é o dos chamados supercomputadores.

Como explicam Meuer e Gietl (2013, p. 03; 10-6), os supercomputadores são máquinas com capacidade de processamento muito superior à dos computadores tradicionais e cuja tecnologia está sempre em rápida evolução. Assim, um supercomputador dos anos 80 se tornou o computador de hoje, de maneira que os atuais supercomputadores representam o futuro. Os autores destacam que esses computadores são essenciais para inúmeras áreas, a saber, aplicações na indústria de óleo e gás, previsão do tempo, modelagem de dados, simulações numéricas e pesquisa e desenvolvimento em saúde e outras ciências.

Mas qual seria a relação desses supercomputadores com a abertura do código-fonte e, mais especificamente, com o *kernel* do Linux? A resposta está contida no gráfico 1.

**Gráfico1.** Top500 Supercomputadores por Família de Sistema Operacional



**Fonte:** Top500 (2016): Top500 Release: November 2016; Category: Operating System Family.

O gráfico acima foi elaborado a partir dos dados fornecidos pelo site Top500, que desde 1993 publica estatísticas semestrais sobre os respectivos 500 supercomputadores mais poderosos do mundo. A metodologia atual empregada pelo site baseia-se no teste comparativo *Linpack*, que avalia os computadores a partir de seu rendimento para a resolução de equações lineares (TOP500, 2017b).

Sistemas derivados do Linux são utilizados em 498 das 500 máquinas mais potentes do último relatório disponível, datado de novembro de 2016. De acordo com a Linux Foundation (2016, p. 16), a explicação desse dado está no caráter adaptável do *kernel*, em que as modificações e incrementos são sempre feitas em consideração aos interesses de todos os desenvolvedores. Essa versatilidade mantém o núcleo em constante renovação e, ao mesmo tempo, compacto para ser utilizado por pequenos sistemas ou, mesmo, por supercomputadores.

O caráter de impacto social dos supercomputadores é atestado pela análise dos segmentos atrelados a essas principais máquinas do mundo: elas acolhem empreendimentos de variadas naturezas, de âmbitos como indústria, governo e pesquisa. Exemplo desses impactos sociais é o do supercomputador brasileiro Santos Dumont, presente na lista com dois de seus módulos (TOP500, 2017). Ele acolhe atualmente 56 pesquisas em andamento, vinculadas a várias instituições nacionais, referentes a propostas de projetos de pesquisa de diferentes áreas do conhecimento, como Engenharia, Física, Química, Biologia, Computação, Meteorologia, Geociências e Ciências da Saúde (BRASIL, 2017).

Em suma, esses dados significam que, uma vez desenvolvido e tornado acessível a todos, o conhecimento empregado em um programa pode ser utilizado para múltiplos propósitos. Licenças de *copyleft*, como a GPL, tornam possível que o saber informatizado possa ser sempre incrementado, propulsionando a inovação tecnológica. Essas licenças superam o legalismo da restrição do conhecimento como maneira de beneficiar não somente o autor, mas toda a comunidade no seu entorno.

## 4.2 Hipóteses rivais à teoria

Diante de tudo isso, cabe levantar uma hipótese rival ao que aqui se propõe. Esta consiste em que a proteção ao código-fonte por meio do copyright possui o papel de garantir ao criador do programa o retorno exclusivo dos seus investimentos, para estimular a criação e evitar a pirataria. Nesse sentido, faz-se uma breve menção às conclusões opostas à inferência, contidas em Qiu (2001) e Suh e Oh (2015), no âmbito econômico, e em Ginsburg (1994) e Samuelson et al. (1994), no âmbito jurídico.

Qiu (2001, p. 01-2; 28-9) aponta que especialmente o software de prateleira<sup>21</sup>, aquele de uso geral, possui custo zero de reprodução, o que facilita sua cópia por

21 Software de prateleira é todo o software que tem um propósito geral e não é customizado para as necessidades de um cliente em específico (QIU, 2001, p. 01). Exemplos são ferramentas de escritório, navegadores ou sistemas operacionais para os usuários comuns.

terceiros e leva à pirataria. O autor defende que legislações fortes de copyright são essenciais para proteger o criador do software e todo o seu investimento em desenvolver o programa. Assim, essa seria a explicação para países com leis de copyright consideradas fracas não se destacarem no desenvolvimento de software de prateleira, já que eles estariam facilmente sujeitos à pirataria.

Da mesma forma, Suh e Oh (2015, p. 152-3) analisam o caso da Coreia do Sul e apontam que regimes de proteção da propriedade intelectual, como o copyright e a patente, geram impacto positivo no desempenho inventivo das empresas de software. Em alguns casos, a proteção por copyright com o fechamento do código-fonte seria uma medida ainda mais eficiente do que a da patente, pois sua concessão é imediata, não demandando prazo de análise. Ainda, segundo os autores, a cópia ilegal do programa desestimula o criador do software e diminui os lucros da indústria responsável por seu desenvolvimento, de maneira que os direitos de propriedade intelectual têm o papel fundamental de evitar que isso ocorra.

No campo jurídico, destaca-se o manifesto defendido por Samuelson et al. (1994), que propõe um regime próprio (*sui generis*) da proteção do software, uma vez que o copyright seria insuficiente em alguns aspectos. Em um documento com 121 páginas, publicado em 1994 no *Columbia Law Journal*, os autores concordam com a proteção que o copyright confere ao código-fonte, mas defendem que ela deveria ser estendida. Como o código é um elemento literal, a proteção seria insuficiente por não acomodar outros elementos não literais do conhecimento de programação. Esses elementos, como o comportamento do software e o respectivo design industrial, demandam muito trabalho para serem desenvolvidos, mas por estarem fora do escopo do código-fonte, poderiam ser facilmente copiados. Assim, cópias com funcionalidades bem similares, ou clones, seriam capazes de “aniquilar” os incentivos para desenvolver programas de computador e uma reforma legislativa seria necessária para estender a proteção (SAMUELSON et al., 1994, p. 2429-31).

Em resposta aos autores, Ginsburg (1994, p. 2559; 2572) argumenta que tal reforma é plenamente desnecessária, pois o copyright já comporta as necessidades do mercado de programas de computador. Segundo o autor, a interpretação dada pelos tribunais e o reforço internacional das legislações por meio de acordos internacionais de direitos autorais, que incluem o software, já demonstram a plena eficiência e flexibilidade do regime de copyright.

Samuelson *et al.* (1994) advogam a necessidade de reforço da proteção do programa de computador. Embora coloquem ressalvas, como o prazo razoável do direito autoral, defendem algo como “todos os direitos reservados e mais um pouco” para o caso do software. Apesar de divergirem com Ginsburg (1994) em alguns aspectos, tais autores concordam que o copyright é essencial para proteger o código-fonte e evitar apropriações ilegais, que desestimulariam o desenvolvimento de software.

Em suma, a hipótese rival defendida pelos quatro autores parte do risco de eventuais apropriações indevidas, ou a pirataria dos programas, e os efeitos destas sobre os criadores de software. Sobre a pirataria de software, embora esse não seja o objeto do presente trabalho, cumpre salientar brevemente que outras abordagens críticas já identificaram que seus efeitos sobre as grandes empresas de software são, em verdade, mais benéficos do que prejudiciais. Nesse sentido, trabalhos como os de Feres et al. (2017), Katz (2005) e Tigre e Marques (2009), entre outros, apontam que a pirataria no mercado de software tem o papel de atrair novos usuários a custo zero para as grandes empresas e de gerar sobre eles uma dependência do programa, o que evita o ingresso de novos concorrentes.

Logo, a inferência descritiva desenvolvida não visa negar o copyright, até porque as licenças de *copyleft* são baseadas a partir desse regime jurídico. O que a teoria aqui defendida diverge das demais apresentadas está no fato de que a restritividade automática do código-fonte, estabelecida pelo copyright, torna-se uma regra inflexível da propriedade intelectual, incapaz de comportar a dinamicidade dos tempos informáticos em que se vive.

## 5 Considerações finais

Este artigo se propôs a verificar o fenômeno dos softwares de código aberto, buscando desmistificar conceitos técnicos, bem como analisar o seu atual escopo jurídico no contexto brasileiro. Metodologicamente, realizou-se uma análise qualitativa de conteúdo pautada em dados diretos e indiretos sobre programas de computador e o direito da propriedade intelectual. Sobre os conteúdos oriundos de fontes diretas, analisaram-se dados relativos ao desenvolvimento do núcleo de sistema GNU/Linux, bem como as licenças de *copyleft* empregadas nos softwares de código aberto e a participação deste programa na composição dos chamados supercomputadores.

Todos os componentes foram analisados a partir do plano teórico dos conceitos de dever e de aspiração da lei, propostos na noção de viver plenamente a lei por Zenon Bankowski. O emprego deste referencial teórico visou a atingir uma abordagem com o olhar mais reflexivo sobre o verdadeiro papel dos regimes jurídicos aplicáveis ao software, tendo em vista a interação entre o dever e a aspiração das normas sobre propriedade intelectual que regulam este objeto de pesquisa.

Constatou-se que a atual legislação brasileira de softwares, a Lei 9.609/98, e a forma de licenciamento autoral por ela privilegiada se apegam às estruturas tradicionais já estabelecidas. Tais estruturas, quando dominantes, negligenciam a disseminação de saber em tempos tecnológicos e integrados, tomando como falso pressuposto o dever intransigente de que a restrição do código-fonte é a única forma de proteção jurídica para atingir a aspiração de incentivo à criação.

Verificou-se, a partir do caso *kernel* Linux, que o modo de elaboração e licenciamento dos softwares livres demonstra-se como possibilidade viável e contemporânea, trazendo benefícios econômicos e sociais, sem que a privação do conhecimento seja estabelecida como premissa inquestionável.

### *Agradecimento*

\* Esta pesquisa conta com o fomento do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e da Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG).

## Referências

ANDROID OPEN SOURCE PROJECT. *About the Android Open Source Project*, 2017.

Disponível em: <<https://source.android.com/>>. Acesso em: 25 maio 2017.

BABBIE, Earl. *The Practice of Social Research*. Eleventh Edition. Belmont: Thomson Wadsworth, 2007.

BANKOWSKI, Zenon. *Vivendo Plenamente a Lei*. Rio de Janeiro: Elsevier Brasil, 2008. 289p.

BOLDRIN, Michele; LEVINE, David K. *Against Intellectual Monopoly*. Newer version, capítulo 2, 22 jan. 2007. Disponível em: <<http://levine.sscnet.ucla.edu/papers/ane02.pdf>>. Acesso em: 03 maio 2017.

BRASIL. *Lei no 9.609, de 19 de fevereiro de 1998*. Dispõe sobre a proteção da propriedade intelectual de programa de computador, sua comercialização no País, e dá outras providências. 1998a. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/leis/l9609.htm](http://www.planalto.gov.br/ccivil_03/leis/l9609.htm)>. Acesso em: 20 abril 2017.

\_\_\_\_\_. *Lei 9.610, de 19 de fevereiro de 1998*. Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências. 1998b. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/leis/l9609.htm](http://www.planalto.gov.br/ccivil_03/leis/l9609.htm). Acesso em: 20 abril 2017.

\_\_\_\_\_. Superior Tribunal de Justiça. *Agravo em Recurso Especial nº 152.661*. Relator: Min. Ricardo Villas Bôas Cueva. Data da publicação: 12 abril 2013. Disponível em: <<http://www.stj.jus.br/SCON/decisoes/toc.jsp?livre=freeware&&DTXT&thesaurus=JURIDICO&p=true>>. Acesso em: 01 jun. 2017.

\_\_\_\_\_. *Sdumont – Projetos em Andamento*. Ministério da Ciência, Tecnologia, Inovações e Comunicações, 2017. Disponível em: <[http://sdumont.lncc.br/projects\\_ongoing.php?ob=4](http://sdumont.lncc.br/projects_ongoing.php?ob=4)>. Acesso em: 01 jun. 2017.

BROW, Christopher F. Copyleft, the Disguised Copyright: Why Legislative Copyright Reform Is Superior To Copyleft Licenses. *University of Missouri Kansas City Law Review*, Vol. 78, p. 749, 2010. Disponível em: SSRN: <<https://ssrn.com/abstract=1688402>>. Acesso em: 10 fev. 2017.

DUSOLLIER, Severine. Open Source and Copyleft: Authorship Reconsidered? *Columbia Journal of Law & Arts*, 2003, vol. 26, pp. 281-296. Disponível em: SSRN: <[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2186190](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2186190)>. Acesso em: 21 mar. 2017.

EPSTEIN, Lee. KING, Gary. *Pesquisa empírica em direito* [livro eletrônico]: as regras de inferência. São Paulo: Direito GV, 2013 (Coleção acadêmica livre) 7 Mb; Título original: The rules of inference. – Vários tradutores.

FERES, Marcos Vinício Chein; OLIVEIRA, Jordan Vinícius de; GONÇALVES, Daniel Domingues. Robin Hood às avessas: software, pirataria e direito autoral. *Revista Direito GV*, vol. 13, n. 1, 2017, p. 69-94. Doi: <<https://doi.org/10.1590/2317-6172201704>>.

FREE SOFTWARE FOUNDATION. *Porque o Código Aberto não compartilha dos objetivos do Software Livre*. Por Richard Stallman, 2016. Disponível em: <<https://www.gnu.org/philosophy/open-source-misses-the-point.pt-br.html>>. Acesso em 28 maio 2017.

\_\_\_\_\_. *GNU/Linux FAQ by Richard Stallman*, 2017. Disponível em: <<https://www.gnu.org/gnu/gnu-linux-faq.html#why>>. Acesso em: 21 abr. 2017.

GINSBURG, Jane C. Four Reasons and a Paradox: the Manifest Superiority of Coyright over Sui Generis Protection of Computter Software. *Columbia Law Review*, vol. 94, nº 08, dez. 1994. p. 2259-2572.

INTEL CORPORATION. *Android on Intel Platforms*. Android Intel Architecture, 2017. Disponível em: <<https://01.org/android-ia/user-guides/android-ia-qa>>. Acesso em: 24 maio 2017.

KATZ, Ariel. A network effects perspective on software piracy. *University of Toronto Law Journal*, v. 55, p. 155, maio 2005. DOI: <<https://doi.org/10.2139/ssrn.386141>>.

LEE, Gwendolyn K.; COLE, Robert. From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development. *Organization Science*, vol. 14, n. 6, p. 633-649, 2003. Disponível em: <<https://ssrn.com/abstract=950262>>. Acesso em: 20 abr. 2017.

LEMOS, Ronaldo. *Direito, Tecnologia e Cultura*. Rio de Janeiro: Editora FGV, 2005. 211p.

LINUX FOUNDATION. *Linux Kernel Development, 25th Anniversary Edition*, 2016. Disponível em: <[http://go.linuxfoundation.org/linux-kernel-development-report-2016?utm\\_source=press-release&utm\\_medium=pr&utm\\_campaign=2016-linux-kernel-report](http://go.linuxfoundation.org/linux-kernel-development-report-2016?utm_source=press-release&utm_medium=pr&utm_campaign=2016-linux-kernel-report)>. Acesso em: 22 dez. 2016.

LINUX KERNEL ORGANIZATION. *Copying: Gnu General Public License version 2*, 1991. Disponível em: <<https://www.kernel.org/pub/linux/kernel/COPYING>>. Acesso em: 03 maio 2017.

MARENCO, Luigi. VEZZOSO, Simonetta. Dynamic Inefficiencies of Intellectual Property Rights from an Evolutionary/Problem- Solving Perspective: Some Insights on Computer Software and Reverse Engineering. . *Social Science Research Network*, jun. 2006. Disponível em: <<http://ssrn.com/abstract=1358920>>. Acesso em: 15 nov. 2013.

MEUER, Hans W.; GIETL, Horst. Supercomputers – Prestige Objects or Crucial Tools for Science and Industry?. *Pik - Praxis Der Informationsverarbeitung Und Kommunikation*, [s.l.], v. 36, n. 2, pp.1-21, jan. 2013. Walter de Gruyter GmbH. DOI: <<https://doi.org/10.1515/pik-2012-0054>>. Acesso em: 03 abril 2017.

NASDAQ. *Red Hat, Inc. Revenue & Earnings Per Share (EPS)*. NASDAQ Stock Exchange, 2017. Disponível em: <<http://www.nasdaq.com/symbol/rht/revenue-eps>>. Acesso em: 01 jun. 2017.

OPEN SOURCE INITIATIVE (2017a). *Open Source Licenses by Category: Popular Licenses*. . Disponível em: <<https://opensource.org/licenses/category>>. Acesso em: 22 mar. 2017.

\_\_\_\_\_. (2017b). *The 2-Clause BSD License*. Disponível em: <<https://opensource.org/licenses/BSD-2-Clause>>. Acesso em: 25 maio 2017.

\_\_\_\_\_. (2017c). *The 3-Clause BSD License*. Disponível em: <<https://opensource.org/licenses/BSD-3-Clause>>. Acesso em: 25 maio 2017.

OSS WATCH (2013). *Frequently Asked Questions*. University of Oxford. Disponível em: <<http://oss-watch.ac.uk/about/faq>>. Acesso em: 22 abril 2017.

\_\_\_\_\_. (2014). *Licence differentiator*. University of Oxford. Disponível em: <<http://oss-watch.ac.uk/apps/licdiff/>>. Acesso em: 22 abril 2017.

QIU, Larry D. Software Development and Trade: Role of Contract Enforcement and Copyright Protection. . *SSRN Eletronic Journal*, nov. 2001. DOI: <<http://dx.doi.org/10.2139/ssrn.298302>>. Acesso em: 20 abril 2017.

RED HAT INC. *About Red Hat - Company Information.*, 2017. Disponível em: <<https://www.redhat.com/en/about/company>>. Acesso em 30 abril 2017.

SAMUELSON, Pamela. DAVIS, Randall. KAPOR, Mitchell D. REICHMAN, J. H. A Manifesto Concerning the Legal Protection of Computer Programs. *Columbia Law Review*, vol. 94, 1994, p. 2308-2431.

STATCOUNTER (2017a). *Frequently Asked Questions*. StatCounter Global Stats. Disponível em: <<http://gs.statcounter.com/faq>>. Acesso em: 28 maio 2017.

\_\_\_\_\_. (2017b). *Mobile Operating System Market Share Worldwide, Apr 2016 to Apr 2017*. StatCounter Global Stats. Disponível em: <<http://gs.statcounter.com/os-market-share/mobile/worldwide>>. Acesso em: 29 maio 2017.

SU, Dukrok. OH, Dong-hyun. The role of software intellectual property rights in strengthening industry performance: Evidence from South Korea. *Technological Forecasting & Social Change*, vol. 92, 2015, p. 140-154. DOI: <<http://dx.doi.org/10.1016/j.techfore.2014.11.003>>. Acesso em: 20 abril 2017.

TIGRE, Paulo Bastos; MARQUES, Felipe Silveira. Apropriação tecnológica na economia do conhecimento: inovação e propriedade intelectual de software na América Latina. *Economia e Sociedade*, n. 18, v. 3, p. 547-566, 2009.

TOP500 (2016). *List Statistics*. Top500.org. Disponível em: <<https://www.top500.org/statistics/list/>>. Acesso em: 27 maio 2017.

\_\_\_\_\_. (2017a). *Sublist Generator*. Top500.org. Disponível em: <<https://www.top500.org/statistics/sublist>>. Acesso em 24 maio de 2017.

\_\_\_\_\_. (2017b). *About*. TOP500.org. Disponível em: <<https://www.top500.org/project/>>. Acesso em: 11 dez. 2017.

WIPO. *Draft Treaty for the Protection of Computer Software*. World Intellectual Property Organization, Committee Of Experts On The Legal Protection Of Computer Software Second Session Geneva, June 13 to 17, 1983. Disponível em: <[http://www.wipo.int/mdocsarchives/LPCS\\_II\\_83/LPCS\\_II\\_3\\_E.pdf](http://www.wipo.int/mdocsarchives/LPCS_II_83/LPCS_II_3_E.pdf)>. Acesso em: 01 jan. 2014.

XIA, Haoxiang. A Collective Intelligence View on the Linux Kernel Developer Community. *International Journal of Knowledge and Systems Science*, vol. 1, n. 4, p. 20-32, 2010. Disponível em: <<https://ssrn.com/abstract=1691604>>. Acesso em: 01 maio 2017.