

Universidade Federal do Paraná

Departamento de Informática

Roberto A Hexsel

Software Livre

Propostas de Ações de Governo para
Incentivar o Uso de Software Livre

Relatório Técnico
RT-DINF 004/2002

Curitiba, PR

2002

Sumário

1	Introdução	2
1.1	Definições	3
1.1.1	Definições preliminares	3
1.1.2	Todo o Espectro: do Livre ao Proprietário	4
1.1.3	Software Comercial	6
1.2	Desenvolvimento de Software Livre	6
1.2.1	Histórico	7
1.2.2	Modos de Organização para o Desenvolvimento de Software	9
1.2.3	Modos de Desenvolvimento de Software Livre	10
1.2.4	Modos de Financiamento	11
1.3	Utilização de Software Livre	12
1.3.1	Vantagens	12
1.3.2	Desvantagens	16
1.3.3	Desvantagens Imaginárias	18
1.4	Treinamento e Capacitação	22
1.5	Combate ao Monopólio da Informação	23
2	Incentivos à Adoção de Software Livre	25
2.1	Ações de Governo	25
2.1.1	Sistemas de Informação de Governo	25
2.1.2	Ações de Incentivo	27
A	Licenças de Software Livre	32
A.1	O Projeto GNU e a Fundação para o Software Livre	32
A.2	GPL – Licença Pública Geral GNU	34
A.2.1	Preâmbulo	34
A.2.2	Licença Pública Geral GNU	35

A.3 Open Source Initiative	39
A.4 Licença BSD	40
A.5 Licença X.Org	41
B Livros sobre Linux	43

Software Livre

O movimento de publicação de Software Livre ganhou notoriedade nos últimos anos. Este modo de produção de software tem resultado em produtos de excelente qualidade e grande penetração em alguns setores do mercado mundial de software. A característica mais importante do software livre é a liberdade de uso, cópia, modificações e redistribuição. Esta liberdade é conferida pelos autores do programa e é efetivada através da distribuição do código fonte dos programas, o que os transforma em bens públicos, disponíveis para utilização por toda a comunidade e da maneira que seja mais conveniente a cada indivíduo.

A liberdade para usar, copiar, modificar e redistribuir software livre lhe confere uma série enorme de vantagens sobre o software proprietário. A mais importante delas é disponibilidade do código fonte, porque isto evita que os usuários se tornem reféns de tecnologias proprietárias. Além desta, as vantagens técnicas são também consideráveis.

A comunidade de desenvolvimento de software livre está espalhada pelo mundo todo e seus participantes cooperam nos projetos através da Internet. Estima-se que participam desta comunidade mais de 100 mil programadores e projetistas, com a grande maioria deles trabalhando voluntariamente em um ou mais projetos. Estima-se também que existam mais de 10 milhões de usuários regulares de sistemas operacionais e aplicativos distribuídos como software livre. Recentemente, empresas como IBM e Hewlet-Packard passaram a investir no desenvolvimento de software a ser distribuído livremente, bem como em serviços para usuários de software livre.

Com números desta ordem de grandeza, e produtos de excelente qualidade, o modo de produção de software livre aparenta ser um novo e fundamental componente da economia moderna, que por ser moderna depende fortemente de software. A importância estratégica do software livre é evidenciada pela publicação de dois estudos sobre o assunto, um encomendado pela Comunidade Européia¹, e outro encomendado pela Presidência dos Estados Unidos². Os dois relatórios contém recomendações ao incentivo da ampliação do uso de software livre, e ao financiamento pelo governo, direta ou indiretamente, do desenvolvimento de sistemas baseados em software livre.

Software proprietário (não-livre) geralmente é produzido com a finalidade de obtenção de lucro e portanto está sujeito à três tipos principais de pressões de mercado: [1] inclusão de funcionalidades ‘imprescindíveis’ (e freqüentemente inúteis), [2] obsolescência programada para possibilitar a venda de novas versões, e [3] prazos de desenvolvimento e testes muito curtos para atender às pressões já mencionadas. Na tentativa de atender ao mercado, e especialmente ao departamento de marketing, produtos importantes de software são vendidos antes de terem passado por testes suficientes, e estarem portanto estáveis e livres da maioria dos erros de programação.

¹Free Software/Open Source: Information Society Opportunities for Europe?, <http://eu.conecta.it>

²PITAC's Panel on Open Source Software for High End Computing, <http://www.itrd.gov>

Quando se considera a dificuldade de obtenção de software confiável no modo tradicional de produção empregado na indústria, e que os sistemas empregados nas sociedades modernas são cada vez mais complexos, sofisticados e imprescindíveis, a existência de um modo alternativo de produção que produz sistemas de qualidade freqüentemente superior àquela do modo tradicional, adquire importância estratégica para o País. A difusão do uso de software livre no Brasil traria conseqüências importantes, destacando-se dentre elas um incremento qualitativo na capacidade nacional de produzir software sofisticado e de boa qualidade.

A qualidade técnica do sistema GNU/Linux advém do modo de produção do software livre, que envolve grandes números de desenvolvedores voluntários. Por sua vez, esta mesma qualidade técnica atrai novos usuários, vários dos quais passam a agir como testadores e desenvolvedores do sistema. Esta atuação produz melhorias na qualidade do sistema, o que acaba por atrair novos usuários. Este ciclo virtuoso deve-se à existência de massa crítica, em nível mundial, tanto de usuários como de desenvolvedores.

Quanto à utilização de software livre no Brasil, percebe-se não há ainda massa crítica de programadores e desenvolvedores, e nem de usuários. Isso deve-se em parte à pequena capilaridade da Internet no país, e à relativa falta de alguns aplicativos que são usados pela maioria dos usuários não especialistas. Considerando-se as inúmeras vantagens da adoção de software livre em larga escala no país, o Governo pode e deve criar as condições para que se estabeleça a massa crítica no Brasil, de forma a que a utilização de software livre produza os benefícios econômicos e sociais que lhe são característicos.

Os benefícios econômicos são muito maiores e mais importantes que a simples economia com o licenciamento de software. A robustez e confiabilidade do software livre provocam reduções significativas em custos operacionais. A disponibilidade do código fonte permite que os sistemas sejam adaptados às condições e necessidades dos usuários. Estas adaptações poderão ser efetuadas por profissionais brasileiros, que terão oportunidades de desenvolvimento muito distintas daquelas vigentes num mercado monopolístico. Além disso, a possibilidade de consulta ao código dos programas permite condições de estudo e aprendizado que são absolutamente inviáveis com software proprietário.

O código de um programa distribuído como software livre torna-se um bem público que está à disposição de toda a sociedade. Neste sentido, software assemelha-se ao conhecimento científico, que uma vez difundido pode ser livremente utilizado por todos, e que assim possibilita o próprio avanço da Ciência. Portanto, os benefícios sociais da publicação e do uso de software livre são a liberdade na utilização das ferramentas, e especialmente na disponibilidade do conhecimento envolvido na produção destas ferramentas, bem como de sua evolução.

Outro benefício social importante é a transparência na codificação das informações tratadas pelos programas. Os formatos empregados para armazenar e tratar as informações são abertos porque o código fonte dos programas pode ser livremente examinado, e não existe assim a possibilidade de que, por exemplo, dados usados no serviço público sejam mantidos em formatos de propriedade de uma entidade privada. O mesmo raciocínio se aplica aos protocolos de comunicação empregados para a transferência de informações entre computadores ou sistemas.

O Papel do Governo

Em vista do exposto, os três níveis de governo tem um papel fundamental na criação das condições necessárias ao estabelecimento de massa crítica de usuários e desenvolvedores de software livre no Brasil. Os governos devem incentivar a utilização de software livre em suas repartições e autarquias, desta forma propiciando o estabelecimento de massa crítica de usuários. Estes por sua vez provocarão a demanda por treinamento e manutenção dos sistemas em uso. O poder de compra dos três níveis de governo, e em especial o do Governo Federal, deve ser empregado para criar a demanda pelo desenvolvimento de aplicativos que atendam às suas necessidades nas áreas de gestão, tributação, educação e governo-eletrônico, para citar apenas áreas de maior visibilidade.

Para que se desfrute das vantagens da utilização de software livre e se alcance os objetivos já enumerados, os governos devem executar as ações listadas abaixo. As três primeiras ações estabelecerão um mercado fornecedor e consumidor de software livre, a quarta e quinta eliminarão a dependência do governo de práticas monopolistas, e as três últimas reforçarão os efeitos das outras ações.

1. Incentivo e recomendação ao uso de software livre em todas as situações em que seu uso não seja inviável;
2. uso do poder de compra para criar padrões de fato;
3. implantação de mecanismos de financiamento e incentivos fiscais ao uso e desenvolvimento;
4. adoção prioritária de protocolos abertos de comunicação;
5. retenção de direitos sobre o código fonte de todo o software adquirido pelo governo;
6. implantação de mecanismos de capacitação ao uso;
7. criação de agência para facilitar a adoção e desenvolvimento de software livre; e
8. avaliação do impacto econômico e social da produção e utilização de software livre no país.

Capítulo 1

Introdução

Desde 1997 a visibilidade do Software Livre junto ao grande público tem aumentado, com reportagens freqüentes na mídia não especializada. O fenômeno é apresentado como novidade, mas a prática da distribuição do código fonte de programas remonta aos primeiros sistemas comercializados na década de 60. Desde então, o desenvolvimento de programas para serem distribuídos como software livre têm sido contínuo, com períodos de maior produtividade em meados da década de 80 e desde 1992 até o corrente. A Seção 1.2.1 contém um breve histórico deste modo de produção de software.

A grosso modo, *software livre* é definido como o software cujo autor o distribui e outorga à todos a liberdade de uso, cópia, alteração e redistribuição de sua obra. A liberdade de uso e alteração somente é viabilizada pela distribuição dos programas na forma de texto legível por humanos, isto é, com seu *código fonte*, bem como no formato executável por um computador. Além do código fonte, o autor do programa outorga a liberdade para que outros programadores possam modificar o código original e redistribuir versões modificadas. O modelo de desenvolvimento de software livre possui uma série de características que são distintas dos modelos de desenvolvimento empregados na indústria. Estas características são discutidas na Seção 1.2.

Ao disponibilizar um programa, seus autores escolhem o grau de liberdade com que modificações e redistribuições podem ser efetuadas. Uma *licença* define estas liberdades de forma mais ou menos estrita. A Seção 1.1 define mais precisamente estas liberdades e discute as diferenças entre os tipos de licença. O Apêndice A contém cópias das licenças mais populares entre os autores de software livre.

Este documento discute software livre com ênfase no sistema GNU/Linux porque, dentre os sistemas operacionais distribuídos como software livre, é o mais popular entre os usuários destes sistemas. Além do sistema operacional, o sistema GNU/Linux inclui um grande número de aplicativos e utilitários, o que lhe confere superioridade tecnológica, e funcionalidade, em muitos aspectos superiores aos sistemas comerciais mais populares, que são o Microsoft Windows, Apple MacIntosh e Sun Solaris. A ênfase adotada aqui reflete o fato de que sem um sistema operacional é muito difícil de se usar um computador, e não há alternativa viável, como software livre, aos sistemas operacionais descendentes do Unix¹.

¹Neste documento, quando se trata de sistemas assemelhados ao Unix, o texto refere-se principalmente ao Sistema GNU/Linux por ser este o mais difundido, e porque sua distribuição é efetuada segundo uma

Existem várias referências à Microsoft e seus produtos neste texto. Estas referências devem-se ao fato de que esta empresa é a maior fornecedora de software para usuários individuais, e detém uma parcela significativa do mercado de servidores de pequeno porte. As referências servem para expor as diferenças entre os modelos de produção de software proprietário e software livre, e não se constituem em crítica à Microsoft ou a seus produtos.

1.1 Definições

Esta seção contém definições para termos empregados neste documento. As definições aqui contidas são baseadas naquelas publicadas pela Free Software Foundation (FSF), Open Source Initiative (OSI), Debian e Berkeley Software Distribution (BSD).

1.1.1 Definições preliminares

pacote – um *pacote* consiste de um conjunto de arquivos distribuídos pelo autor do/s programa/s nele contido/s. Estes arquivos geralmente consistem do código fonte dos programas que compõem o pacote, sua documentação, e possivelmente versões executáveis do programa.

autor – a *autor* é a pessoa, ou grupo de pessoas, que produziu a versão original de um pacote.

versão original – a *versão original* de um programa é a versão originalmente distribuída pelo seu autor, sem nenhuma modificação introduzida por terceiros.

trabalho derivado – um *trabalho derivado* resulta de introdução de modificações na versão original de um programa, modificações que alteram a funcionalidade do programa, ao invés de apenas corrigir eventuais erros.

licença – a *licença* é o termo de outorga de direitos em que o autor define qual o grau de liberdade que terceiros possuem para modificar e/ou redistribuir um programa e/ou seus trabalhos derivados. Geralmente, a licença *restringe* a liberdade de uso.

código fonte – o *código fonte* de um programa é a versão daquele programa produzida diretamente pelo autor, e que descreve o comportamento, ou função, do programa. Código fonte é geralmente escrito em uma linguagem de alto nível como C, Java ou Perl.

código executável – o *código executável*, ou a versão em *'binário'*, de um programa é aquela que resulta do processo de tradução automática do código fonte para uma versão que será executada diretamente pelo computador.

distribuição – uma *distribuição* geralmente consiste de um conjunto de pacotes, devidamente organizados e armazenados em um meio físico adequado para o manuseio e instalação em computador.

licença que melhor protege os interesses da comunidade do que aquela dos sistemas da família BSD. Contudo, praticamente toda a discussão que se segue aplica-se igualmente às duas famílias de sistemas.

1.1.2 Todo o Espectro: do Livre ao Proprietário

Software Livre (*Free Software*) é o software disponível com a permissão para qualquer um usá-lo, copiá-lo, e distribuí-lo, seja na sua forma original ou com modificações, seja gratuitamente ou com custo. Em especial, a possibilidade de modificações implica em que o código fonte esteja disponível. Se um programa é livre, potencialmente ele pode ser incluído em um sistema operacional também livre. É importante não confundir *software livre* com software *grátis* porque a liberdade associada ao software livre de copiar, modificar e redistribuir, independe de gratuidade². Existem programas que podem ser obtidos gratuitamente mas que não podem ser modificados, nem redistribuídos. Por outro lado, existe a possibilidade de uso não-gratuito em todas as categorias listadas no que segue. Há uma cópia da definição de *software livre* pela Free Software Foundation na página 32. A Figura 1.1 mostra as intersecções entre as várias categorias de licenças de software³.

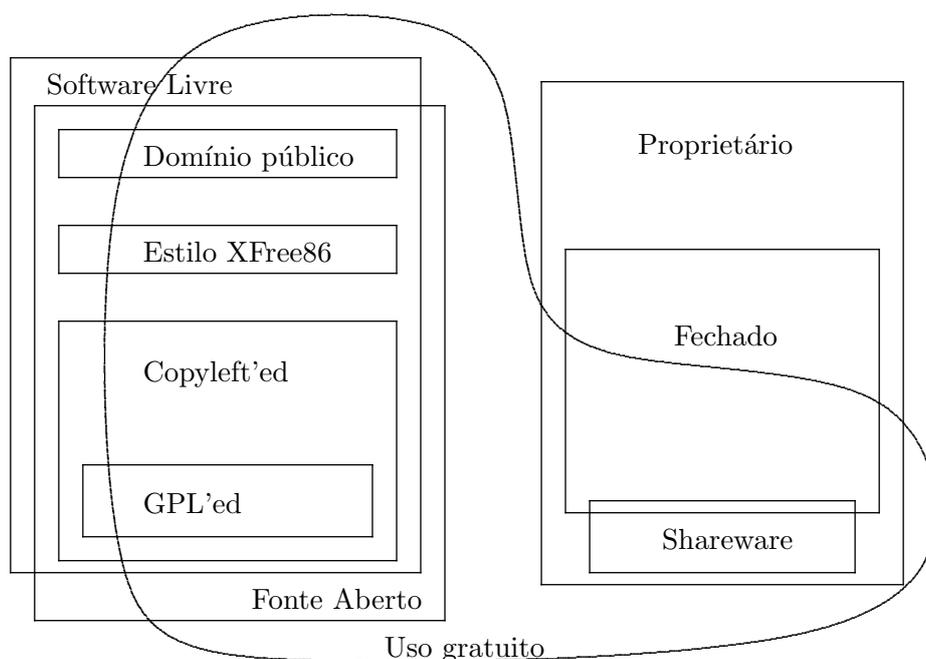


Figura 1.1: Categorias de licenças de software.

Copyleft

A maioria das licenças usadas na publicação de software livre permite que os programas sejam modificados e redistribuídos. Estas práticas são geralmente proibidas pela legislação internacional de *copyright*, que tenta justamente impedir que alterações e cópias sejam efetuadas sem a autorização do/s autor/es. As licenças que acompanham software livre fazem uso da legislação de *copyright* para impedir utilização não-autorizada, mas estas licenças definem clara e explicitamente as condições sob as quais cópias, modificações e redistribuições *podem* ser efetuadas, para garantir as liberdades de modificar e redistribuir o software assim licenciado. A esta versão de *copyright*, dá-se o nome de *copyleft*.

²Em inglês: *free software* versus *free beer*, onde *free* significa liberdade de expressão, e não gratuidade.

³Diagrama obtido de www.fsf.org/philosophy/categories.html, elaborado por Chao-Kuei.

Licenças de Software Livre

Existem vários estilos de licenças para a distribuição de software livre⁴, que se distinguem pelo grau de liberdade outorgado ao usuário. O Apêndice A.1 contém o texto integral das licenças GPL, Open Source, BSD e X.org.

GPL A Licença Pública Geral GNU (*GNU General Public License* – GPL) é a licença que acompanha os pacotes distribuídos pelo Projeto GNU, e mais uma grande variedade de software, incluindo o núcleo do sistema operacional Linux. A formulação da GPL é tal que ao invés de limitar a distribuição do software por ela protegido, ela de fato impede que este software seja integrado em software proprietário. A GPL é baseada na legislação internacional de *copyright*, o que deve garantir cobertura legal para o software licenciado com a GPL⁵. Texto integral na página 34.

Debian A licença Debian é parte do *contrato social* celebrado entre a Debian e a comunidade de usuários de software livre, e é chamada de *Debian Free Software Guidelines* (DFSG). Em essência, esta licença contém critérios para a distribuição que incluem, além da exigência da publicação do código fonte. Estes critérios são: (a) a redistribuição deve ser livre; (b) o código fonte deve ser incluído e deve poder ser redistribuído; (c) trabalhos derivados devem poder ser redistribuídos sob a mesma licença do original; (d) pode haver restrições quanto a redistribuição do código fonte, se o original foi modificado; (e) a licença não pode discriminar contra qualquer pessoa ou grupo de pessoas, nem quanto a formas de utilização do software; (f) os direitos outorgados não podem depender da distribuição onde o software se encontra; e (g) a licença não pode ‘contaminar’ outro software⁶.

Open Source A licença do Open Source Initiative é derivada da Licença Debian, com as menções à Debian removidas. Texto integral na página 39.

BSD A licença BSD cobre as distribuições de software da Berkeley Software Distribution, além de outros programas. Esta é uma licença considerada ‘permissiva’ porque impõe poucas restrições sobre a forma de uso, alterações e redistribuição do software licenciado. O software pode ser vendido e não há obrigações quanto a inclusão do código fonte, podendo o mesmo ser incluído em software proprietário. Esta licença garante o crédito aos autores do software mas não tenta garantir que trabalhos derivados permanecem como software livre. Texto integral na página 40.

X.org O Consórcio X distribui o X Window System sob uma licença que o faz software livre mas não adere ao *copyleft*. Existem distribuições sob a licença da X.org que são software livre, e outras distribuições não o são. Existem algumas versões não-livres do sistema de janelas X11 para estações de trabalho e certos dispositivos do IBM-PC que são as únicas funcionais disponíveis, sem similares distribuídos como software livre. Texto integral na página 41.

⁴www.fsf.org/philosophy/categories.html

⁵Até o momento, não há notícia de procedimentos legais iniciados por conta de violação da GPL.

⁶Ao contrário da GPL que o faz.

Software em Domínio Público

Software em domínio público é software sem *copyright*. Alguns tipos de cópia, ou versões modificadas, podem não ser livres porque o autor permite que restrições adicionais sejam impostas na redistribuição do original ou de trabalhos derivados.

Software Semi-livre

Software semi-livre é software que não é livre, mas é concedida a permissão para que indivíduos o usem, copiem, distribuam e modifiquem, incluindo a distribuição de versões modificadas, desde que o façam sem o propósito de auferir lucros.

Exemplos de software semi-livre são as primeiras versões do Internet Explorer da Microsoft, algumas versões dos *browsers* da Netscape, e o StarOffice.

Freeware

O termo *freeware* não possui uma definição amplamente aceita mas é usado com programas que permitem a redistribuição mas não a modificação, e seu código fonte não é disponibilizado. Estes programas *não são* software livre.

Shareware

Shareware é o software disponibilizado com a permissão para que seja redistribuído, mas a sua utilização implica no pagamento pela sua licença. Geralmente, o código fonte não é disponibilizado e portanto modificações são impossíveis.

Software Proprietário

Software proprietário é aquele cuja cópia, redistribuição ou modificação são em alguma medida proibidos pelo seu proprietário. Para usar, copiar ou redistribuir, deve-se solicitar permissão ao proprietário, ou pagar para poder fazê-lo. Há um exemplo de licença de software proprietário na Seção 1.3.3, página 18.

1.1.3 Software Comercial

Software comercial é o software desenvolvido por uma empresa com o objetivo de lucrar com sua utilização. Note que ‘comercial’ e ‘proprietário’ não são o mesmo. A maioria do software comercial é proprietário mas existe software livre que é comercial, e existe software não-livre não-comercial.

1.2 Desenvolvimento de Software Livre

A cultura de engenharia praticada pela comunidade que desenvolve os protocolos empregados na Internet se baseia na publicação de propostas de protocolos ou serviços, e na

avaliação destas propostas em listas de discussão via correio eletrônico. Geralmente, as propostas são acompanhadas de uma *implementação de referência*, cujo código fonte é disponibilizado a todos os interessados. De posse da especificação dos protocolos e de código que os implementa, a comunidade se lança à avaliação, correção de eventuais erros, e discussão e aperfeiçoamento da proposta ou implementação. Após o período de discussão aberta ao público, os protocolos são avaliados e sacramentados na Internet Engineering Task Force (IETF), que é a entidade responsável pela formalização e publicação dos protocolos⁷.

A disponibilidade das propostas de protocolos, através dos Request For Comments (RFCs), permite a discussão e a depuração dos protocolos por uma grande comunidade de especialistas. Este processo têm produzido protocolos extremamente confiáveis, robustos e escaláveis, especialmente quando se considera que Internet é heterogênea, multi-plataforma e geograficamente distribuída, e que opera há mais de trinta anos mantendo compatibilidade entre várias gerações de tecnologia [13, 19]. O modo de produção de software livre é similar à cultura de engenharia descrita acima e uma boa parte da infraestrutura de software da Internet é composta por programas e utilitários desenvolvidos e distribuídos como software livre, e sua confiabilidade, robustez e qualidade são excelentes.

A forma de operação da Internet, em especial no período anterior a comercialização de serviços, depende fundamentalmente de um princípio mutualista segundo o qual os participantes da ‘rede’ recebem dela mais do que nela colocam. Por exemplo, quando uma entidade se liga à Internet, suas máquinas passam a usufruir de rede para transmitir seu tráfego em troca da retransmissão de tráfego proveniente de todas as outras entidades ligadas à rede. Geralmente, os custos associados à retransmissão do tráfego externo são muito menores que aquele decorrente da montagem de infraestrutura de comunicação privativa.

A mutualidade é o mesmo princípio que anima os cientistas a publicarem os resultados de suas pesquisas. A publicação tem duas finalidades importantes: (1) os resultados ficam sujeitos ao escrutínio pela comunidade científica, o que garante sua qualidade e confiabilidade, e (2) estes resultados se incorporam ao corpo de conhecimento da área e ficam disponíveis para uso pelos outros pesquisadores, estudantes e demais interessados [20]. Este é o mesmo processo a que estão sujeitos os protocolos da Internet. Uma vez revisados e aprovados pela comunidade, os protocolos ficam disponíveis ao público para que sejam implementados e estudados, pela indústria, na academia, e pelas novas gerações de projetistas de protocolos. Da mesma forma, a distribuição do código fonte permite sua utilização por outros programadores em seus próprios projetos, bem como possibilita o aprendizado por outros programadores através do estudo do código dos programas que empregam diariamente. Sob esta ótica, o código fonte dos programas é uma forma de conhecimento científico.

1.2.1 Histórico

Na década de 1960, os fabricantes de sistemas comerciais como a IBM vendiam seus computadores e entregavam aos clientes o código fonte dos programas, permitindo-lhes alterar os programas e redistribuí-los livremente. Dez anos mais tarde, as aplicações e o

⁷O *modus operandi* do IETF pode ser resumido por “*We reject kings, presidents, and voting. We believe in rough consensus and running code*”, texto impresso em camisetas usadas por participantes de algumas reuniões daquela organização.

número de usuários aumentaram a tal ponto que passou a ser possível *vender* software ao invés de distribuí-lo gratuitamente. A situação se alterou de forma a que o software tornou-se relativamente mais importante que o hardware, e portanto os fabricantes passaram a vender os aplicativos, sem fornecer o código fonte, além de impor restrições à redistribuição dos programas.

No início da década de 80 estabeleceu-se o Projeto GNU, liderado por Richard Stallman e em reação à sua frustração com a crescente comercialização de software. Stallman trabalhava como programador no Massachusetts Institute of Technology (MIT) e demitiu-se para trabalhar no desenvolvimento de um sistema operacional completo a ser distribuído como software livre. Este sistema seria chamado de GNU⁸ e seria composto por um sistema operacional e uma série de aplicativos e utilitários. Em 1984, Stallman publicou o Manifesto GNU [24], onde definiu o que se entende por *software livre* e solicitou a participação de outros programadores na enorme tarefa e que se propunha. Como fruto deste esforço, foram produzidos o editor de textos **emacs**, o compilador **gcc**, e várias outras ferramentas e utilitários.

O ambiente de desenvolvimento no Projeto GNU era Unix porque este era o sistema tecnicamente mais avançado, estava disponível para várias plataformas, era distribuído mais livremente que os outros sistemas, sendo por estas razões o mais popular na academia e em setores da indústria de computação. Em pouco tempo as ferramentas produzidas pelo Projeto GNU atingiram tal qualidade que administradores de sistemas passaram a utilizá-las ao invés das ferramentas distribuídas com os sistemas proprietários. Estes programas são distribuídos sob os termos da GPL.

No final da década de 1960 iniciou-se na AT&T o desenvolvimento do sistema operacional Unix. Em 1976 a AT&T lançou no mercado a versão comercializada como Unix V6. No ano seguinte, a AT&T licenciou o Unix para a Universidade da Califórnia em Berkeley, cedendo-lhe o código fonte do sistema. Aquela universidade passou a desenvolver sua própria versão do Unix, conhecida como BSD-Unix (Berkeley Software Distribution). Em 1977 o Unix V7 foi lançado, com preços bastante diferentes para as Universidades (US\$100) e para empresas (US\$21.000), o que incentivou a difusão, e acelerou o desenvolvimento de uma versão do Unix no meio acadêmico. Em 1989 Berkeley distribuiu a versão Net/1 do Unix como sistema operacional livre, e em 1991 foi lançada a versão Net/2. Em 1993 a AT&T acusou a Universidade da Califórnia em Berkeley de distribuir a versão Net/2 com código pertencente ao AT&T-Unix. Como reação, a Universidade lançou a versão BSD4.4 em 1994, livre de código da AT&T. Hoje são disponíveis três herdeiros do BSD-Unix, FreeBSD, NetBSD e OpenBSD, distribuídos como software livre.

O Projeto GNU deu origem à Free Software Foundation, sob a liderança de Stallman. No início da década de 90 a maior parte das ferramentas de apoio já haviam sido criadas mas faltava o núcleo do sistema operacional. Esta deficiência seria suprida pelo trabalho de Linus Torvalds, que desenvolvera um núcleo de sistema operacional, batizado de Linux⁹. Em 1991 Torvalds disponibilizou o código fonte do Linux na Internet e solicitou a colaboração de outros programadores para que estes desenvolvessem as partes ainda faltantes. A resposta foi entusiástica e em menos de dois anos Linux já havia se tornado um sistema razoavelmente estável. Os esforços da FSF e da comunidade Linux foram conjugados e o sistema GNU/Linux passou a ser distribuído e desde então vem sendo continuamente

⁸GNU é uma referência circular: *GNU's not Unix*, e é isofônico à palavra Inglesa *new*.

⁹Torvalds era aluno de pós-graduação da Universidade de Helsinkí nesta época.

desenvolvido e aperfeiçoado. O núcleo do Linux é distribuído sob a GPL.

Outro projeto de software livre da década de 80 e digno de nota é o sistema de processamento de textos \TeX , projeto este iniciado por Donald Knuth e cada vez mais ativo. \TeX é um formador de textos, ou um tipógrafo automático, que permite produzir tipografia de altíssima qualidade, especialmente em textos de Matemática¹⁰.

Na década de 90 vários projetos obtiveram sucesso, especialmente o compilador `gcc`, o servidor de páginas WWW `apache`, a linguagem interpretada `perl`[5], e os ambientes de trabalho KDE¹¹ e GNOME¹². Estes dois últimos projetos são importantes porque eles devem resolver uma das questões mais difíceis associadas à difusão do uso do software livre, que é a possibilidade de uso por pessoal não-técnico, conforme discutido na Seção 1.3.2, na página 16.

1.2.2 Modos de Organização para o Desenvolvimento de Software

No seu artigo mais popular, *The Cathedral and the Bazaar* [21], Eric S Raymond descreve os dois modos de organização mais empregados no desenvolvimento de software, que são a *catedral* e o *bazar*. A discussão sobre estes modelos é aprofundada em outros dois artigos de Raymond [22, 23], além de comentário e crítica sobre os mesmos em [9, 6, 8].

O modelo tradicionalmente empregado na indústria é similar ao projeto de uma *catedral medieval*, no qual um restrito grupo de projetistas exerce controle férreo sobre o trabalho de um pequeno exército de operários. O modelo da catedral é empregado na maioria dos projetos de desenvolvimento de software proprietário, onde o modelo descreve o relacionamento entre a gerência de projeto (e o departamento de *marketing*) que estabelecem metodologias, tarefas e prazos, que devem ser cumpridos pelos programadores engajados no projeto.

O outro modo de organização, freqüentemente empregado pela comunidade de software livre se assemelha a um anárquico *bazar*, onde não há hierarquia entre os participantes e todos cooperam para que o bazar seja atrativo aos compradores, ao mesmo tempo em que competem pela atenção destes mesmos compradores. Na produção de software no bazar os projetos são informalmente organizados ao redor da proposta de desenvolvimento de algum aplicativo ‘interessante’, do qual os interessados participam voluntariamente, e o/a líder do projeto emerge por seus méritos como programador/a ou projetista.

Uma das conseqüências do desenvolvimento “na catedral” é a dificuldade para que se atinja massa crítica de usuários e desenvolvedores nas fases iniciais do desenvolvimento de um sistema. Na catedral, os projetistas definem as características técnicas e funcionais do produto e repassam o desenvolvimento aos programadores. Quando o produto atinge a fase de testes, um grupo restrito de testadores valida o produto, que é então liberado para uso e/ou comercialização. São conhecidos inúmeros exemplos de produtos que foram disponibilizados e vendidos antes de atingirem estabilidade ou maturidade simplesmente porque o prazo de ‘desenvolvimento’ se esgotou.

O desenvolvimento de software segundo o modelo da catedral não é exclusividade da indústria. A maioria dos projetos da FSF é desenvolvido neste modelo, onde um pequeno

¹⁰Este documento foi preparado com \LaTeX , sistema derivado do \TeX desenvolvido por Leslie Lamport.

¹¹www.kde.org

¹²www.gnome.org

grupo de projetistas e desenvolvedores trabalham em certo projeto antes que uma versão do mesmo seja disponibilizada. O grupo original geralmente detém o controle sobre as modificações a serem introduzidas no programa, após sua publicação como software livre. Isto faz com que o software desenvolvido pela FSF, em que pesem suas qualidades técnicas, geralmente seja empregado de forma restrita na academia e alguns setores da indústria. Os programas e ferramentas da FSF adquiriram maior popularidade com o advento do Linux porque então aqueles passaram a se beneficiar da massa crítica de usuários e desenvolvedores do Linux.

1.2.3 Modos de Desenvolvimento de Software Livre

O ciclo de vida típico de um projeto de software livre de sucesso passa pelas seguintes fases. O processo de desenvolvimento se inicia quando o autor escreve uma versão inicial do programa $\mathcal{P}v0$ e publica o código fonte desta versão incompleta. Se o programa é interessante ou útil, outros programadores o instalam e experimentam. Erros são descobertos e corrigidos, e melhorias são propostas ou introduzidas no programa. Estas correções e melhorias são submetidas ao autor, que as incorpora e publica a nova versão de seu programa, $\mathcal{P}v1$. A versão melhorada atrai mais usuários, que descobrem outros erros e introduzem novas melhorias, o que leva a uma nova versão. Para programas considerados úteis, e que portanto atraem muitos novos usuários, este ciclo se repete a cada poucos meses. Após algumas iterações, o programa \mathcal{P} atinge estabilidade e passa a contar com um grupo razoavelmente grande de usuários. Nestas condições, a comunidade de suporte ao programa atinge massa crítica, e isso garante a continuidade de seu desenvolvimento e suporte.

Em grande medida, a evolução da funcionalidade de um programa desenvolvido “no bazar” é orientada pelos usuários do programa, especialmente aqueles que são desenvolvedores ou programadores. Por outro lado, em uma ‘catedral’ ou na indústria, a orientação é definida pelos projetistas, pressões de mercado, e possivelmente dirigida por um departamento de *marketing*. No caso do software livre, muitos dos usuários das primeiras versões são programadores, que introduzem melhorias na versão original, o que atrai mais usuários, até que se estabeleça massa crítica. Estes sugerem acréscimos e outras melhorias, bem como reclamam de possíveis erros ou falhas, que são corrigidas. Não há prazo para que o ‘produto’ seja entregue porque a evolução se dá na medida da disponibilidade dos programadores participantes e da quantidade das melhorias sugeridas e erros descobertos. Este tipo de sinergia entre desenvolvedores e usuários simplesmente não é possível no estilo catedral de desenvolvimento por causa das pressões impostas pelos prazos de lançamento dos produtos, e pelas características funcionais que são “exigidas pelo mercado”.

A alta qualidade dos programas desenvolvidos no ‘bazar’ se deve ao número de desenvolvedores e usuários envolvidos no processo, especialmente após o estabelecimento de massa crítica, e para isso a publicação do código fonte dos programas é um fator essencial. Uma vez publicado, o código pode ser esmiuçado por quem se interessar, o que acelera e intensifica o processo de depuração, descoberta e correção de erros. Dependendo da popularidade ou importância de um programa, a comunidade de usuários e desenvolvedores rapidamente atinge massa crítica, e centenas de programadores e milhares de usuários se atiram à tarefa de testá-lo e eventualmente corrigi-lo. Este método de teste baseado em grandes populações de testadores simplesmente não é viável no modelo da catedral.

Quando um programa não atrai uma comunidade suficientemente grande de usuários e programadores dispostos a suportá-lo, geralmente seu desenvolvimento pára e o projeto tende a estagnar. Também é possível que o autor, ou os principais desenvolvedores, percam a disponibilidade ou interesse no projeto, o que tende a diminuir a qualidade do suporte e da evolução do programa. Note-se que isto não é necessariamente ruim, porque pode existir algum outro programa com funcionalidade similar mas com melhor qualidade.

O modelo *catedral*×*bazar* contempla aspectos importantes do desenvolvimento de software livre. Outros modelos e estudos são apresentados e discutidos em [4, 11, 26].

1.2.4 Modos de Financiamento

Os sistemas de software livre são financiados indiretamente pelos vários níveis de governo e academia e segmentos da indústria nos modos listados abaixo, e em diferentes combinações destes [5, 1, 11].

Voluntários individuais Há um grande número de desenvolvedores de software livre que trabalham em universidades e institutos de pesquisa. O desenvolvimento é parcialmente financiado pela instituição, como por exemplo o desenvolvimento do BSD-Unix na Universidade de Berkeley, o protocolo HTTP no Centro Europeu de Pesquisas Nucleares (CERN), e o browser Mosaic no Centro de Supercomputação da Univ. do Illinois, dentre vários outros.

Suporte indireto Em empresas médias e grandes, funcionários sub-ocupados podem devotar-se ao desenvolvimento de projetos fora do escopo de interesse da empresa, como a primeira versão do Unix e da linguagem C no Bell Labs, embora estes sistemas não tenham sido publicados como software livre.

Suporte parcial Existem empresas que financiam o desenvolvimento de software livre como forma de promover a venda de seus equipamentos. Como exemplos pode-se citar a IBM que suporta o desenvolvimento do Apache, que é instalado nos computadores da série AS400, e a Silicon Graphics que suporta desenvolvimento do Samba¹³.

Outra forma de suporte parcial interessante é representada pelo consórcio *está* financiando a instalação e operação do Open Software Development Laboratory (OSDL)¹⁴, para suportar o desenvolvimento de aplicativos para sistemas de grande porte, especialmente sistemas empresariais com características de altas confiabilidade e disponibilidade. Destacam-se entre os participantes Hewlet-Packard e IBM.

Suporte direto Existem empresas que financiam o desenvolvimento de software livre para ganhar mercado ou competir diretamente no mercado, como a Netscape ao abrir o esforço de desenvolvimento do *browser* Mozilla à comunidade de software livre, ou empresas como Red Hat e Conectiva que vendem serviços e produtos baseados em software livre.

¹³Samba permite o compartilhamento de sistemas de arquivos de Microsoft, instalados em máquinas Unix, através da rede.

¹⁴www.osdlab.org

1.3 Utilização de Software Livre

Esta seção contém uma discussão sobre a utilização de software livre, e as vantagens e desvantagens associadas ao uso de sistemas e aplicativos desenvolvidos e distribuídos como software livre. No que concerne a desvantagens, são discutidas as desvantagens reais, assim como algumas características que são desvantagens imaginárias, decorrentes de diferenças de perspectiva ou de percepção.

1.3.1 Vantagens

Dentre as vantagens decorrentes da utilização de software livre salientam-se aquelas listadas abaixo, e discutidas nos próximos parágrafos.

- Custo social é baixo;
- não se fica refém de tecnologia proprietária;
- independência de fornecedor único;
- desembolso inicial próximo de zero;
- não obsolescência do hardware;
- robustez e segurança;
- possibilidade de adequar aplicativos e redistribuir versão alterada;
- suporte abundante e gratuito; e
- sistemas e aplicativos geralmente muito configuráveis.

Custo social é baixo O desenvolvimento de software proprietário é orientado para o benefício do fabricante enquanto que o do software livre é orientado para o benefício de seus usuários. Os lucros decorrentes das vendas de software proprietário são sempre privatizados, enquanto que os frutos da produção de software livre tornam-se disponíveis para toda a comunidade. Assim que disponibilizado como software livre, um aplicativo ou sistema torna-se um bem público, cuja utilização e evolução é determinada pela comunidade de usuários.

Não se fica refém de tecnologia proprietária Considere-se uma organização comercial ou governamental cujas operações dependam de sistemas de software produzidos por um único fornecedor. Se o fornecedor decidir, por razões que lhe são peculiares, descontinuar um produto, ou uma linha de produtos, para lançar uma ‘nova’ e ‘melhorada’ versão, os usuários não têm outra alternativa a não ser adotar esta nova versão e arcar com os custos da migração de seus sistemas. Existe também a possibilidade de o fornecedor interromper suas atividades. Há portanto um risco inerente na adoção de um plano de negócios que dependa de um único fornecedor de software e/ou sistemas. Aparentemente, este risco é com frequência ignorado ou considerado desprezível.

Independência de fornecedor único No caso de software livre, como não existe uma entidade que detenha os direitos de propriedade sobre o código fonte dos programas, não existe a possibilidade de que um determinado ‘produto’ seja descontinuado segundo a conveniência comercial do fornecedor do sistema. Da mesma forma, mesmo que alguma

das empresas que distribuem software livre seja extinta, existem várias outras provendo serviços e produtos similares, que poderiam facilmente substituir àquela que desapareceu. Além disso, estando o código fonte disponível na Internet, se houver um número razoável de usuários de determinado aplicativo e/ou sistema, a demanda por suporte gerada pelos usuários têm se mostrado suficiente para que suporte e manutenção sejam oferecidos por algum grupo de programadores através da Internet. Se a necessidade o justificar, sempre é possível contratar programadores para efetuarem manutenção nos programas pois seu código fonte está disponível. Esta opção simplesmente inexistente com software proprietário.

Desembolso inicial próximo de zero Existem distribuições que são vendidas por preço de custo (Debian), preço este que basicamente cobre os custos de produção da mídia de distribuição (conjunto de CDs e embalagem). Existem distribuições que, além da mídia, incluem farta documentação em papel, com manuais detalhados de instalação e administração dos sistemas (Conectiva, Red Hat). Nestes casos, o preço da distribuição cobre os custos de produção destes itens adicionais. De qualquer forma, o preço das distribuições é uma pequena fração do preço de produtos comerciais similares. Existe ainda a possibilidade da pura e simples cópia dos arquivos completos das instalações através da Internet. A menos do preço da conexão e do tempo necessário, é possível obter-se distribuições atualizadas a um custo próximo de zero.

No caso de sistemas proprietários, a manutenção após a aquisição têm custo elevado porque depende dos serviços monopolizados pelo fornecedor ou providos por outras empresas, ou consultores individuais. Estima-se que custos de manutenção sejam, no pior caso, similares para software livre quando comparados a sistemas proprietários [16]. Se o suporte necessário não é obtido gratuitamente (na Internet), alguém, ou alguma empresa, deverá ser contratado para prover o suporte necessário.

Não-obsolescência do hardware Uma conseqüência da utilização de software proprietário é a acelerada obsolescência do hardware. Tipicamente, quando o fornecedor decide publicar uma nova versão dos aplicativos de escritório, o equipamento que os executa deve também ser atualizado ou substituído. Isso é necessário porque as funcionalidades adicionais que sempre são introduzidas nas novas versões aumentam a complexidade e o tamanho dos aplicativos e portanto exigem processadores mais rápidos e maior capacidade de memória e disco. É freqüente o caso em que as ‘novas’ funcionalidades são apenas cosméticas e/ou utilizadas por uma parcela pequena dos usuários, mas produtos são vendidos com um conjunto enorme de funcionalidades com utilidade limitada para a grande maioria de seus usuários. Este fenômeno é conhecido como “inchaço do software”, ou *software bloat*.

Tal inchaço ocorre em escala muito menor com sistemas de software livre porque a pressão de marketing por novas funcionalidades é pequena ou inexistente. Além disso, os sistemas de software livre são concebidos e projetados *para serem usados pelos projetistas, no seu próprio equipamento*, de tal forma que funcionalidades cosméticas são sistematicamente excluídas em favor daquelas que são de fato úteis. Isso significa que máquinas que seriam consideradas obsoletas pelos padrões de mercado podem ser utilizadas plenamente com software livre.

Esta característica do software livre é extremamente favorável à administração pública [2]. Quando se utiliza software livre, o parque de máquinas instalado não necessita ser atu-

alizado com a mesma frequência que seria necessária no caso da utilização de software proprietário, porque este induz à aquisição de novas plataformas. Isso ocorre porque o desenvolvimento de aplicativos proprietários se dá na *plataforma modelo* de um dado fabricante de hardware. Esta plataforma modelo é uma das máquinas mais potentes que serão vendidas na *próxima geração* de produtos do fabricante. Assim, o hardware e o software da nova geração são desenvolvidos para operarem bem juntos, o que induz a aquisição da *nova* plataforma modelo para que o *novo* software opere em condições aceitáveis de desempenho.

Robustez e segurança Os sistemas desenvolvidos e distribuídos como software livre são reconhecidos por sua robustez e segurança. Estas qualidades são resultado do processo de desenvolvimento do software livre, que é similar ao sistema acadêmico de revisão de textos para publicação. Na academia, antes de ser considerado apto para publicação, um artigo é examinado por um ou mais revisores anônimos que atestam sua qualidade e que freqüentemente sugerem correções e ou adições. Este sistema, conhecido como *peer review*, garante alto nível de qualidade e integridade das publicações científicas.

Este mesmo processo de revisão ocorre com programas cujo código fonte é distribuído livremente. Assim que um programa é liberado para experimentação, outros programadores o instalam e usam e inicia-se o processo de depuração distribuída. Erros descobertos são reportados ao/s autor/es, freqüentemente já acompanhados da correção. Quanto maior a utilidade intrínseca do programa, maior é o número de usuários-testadores e mais curto o ciclo de depuração-correção. Isso também significa que quaisquer problemas associados à segurança são descobertos, resolvidos e as correções publicadas ampla e rapidamente.

O modelo de desenvolvimento de software proprietário inviabiliza este mecanismo de revisão e correções, e portanto seus produtos em geral não são tão robustos nem tão seguros quanto os similares desenvolvidos “no bazar”. São conhecidos casos onde versões instáveis de produtos são disponibilizadas comercialmente, e apesar de sua baixa qualidade estes produtos são adquiridos devido à bem-montada campanha de publicidade, e à existência de massa crítica de consumidores que acaba por reforçar a publicidade.

Possibilidade de adaptar aplicativos Uma das maiores vantagens advindas da distribuição do código fonte dos programas é a liberdade para adaptar um programa às necessidades de seus usuários. Evidentemente, programadores capacitados devem ser de alguma forma engajados para efetuar as adaptações necessárias. Em sistemas proprietários, esta possibilidade pode não existir, a não ser para clientes especiais e com custo elevado.

Estima-se que a compra do software e o pagamento pela licença de uso corresponda a cerca de 10 a 30% do custo total do sistema. A maior parcela do custo advém de adaptações e correções ao software que devem ser efetuadas ao longo de sua vida útil. As elevadas confiabilidade e segurança do software livre, aliadas à facilidade de adaptação, permitem reduções substanciais no custo total¹⁵ de sistemas baseados em software livre.

Suporte A revista Infoworld escolheu a comunidade Linux na Internet para receber o prêmio de “Melhor Suporte Técnico” nos anos de 1998 e 1999. Estes prêmios são em reconhecimento à qualidade do suporte técnico disponível gratuitamente através da

¹⁵ *Total Cost of Ownership*, ou TCO.

Internet, e fornecido pela comunidade de desenvolvedores e usuários do GNU/Linux e dos aplicativos normalmente distribuídos com o sistema operacional. De forma geral, o suporte aos usuários que estejam conectados à Internet é de boa qualidade e com tempos de resposta medidos em poucas horas, não importando o grau de dificuldade da consulta.

Uma dos preconceitos mais comuns sobre a aplicabilidade de software livre refere-se à falta de suporte. Esta reclamação carece de fundamento quando se considera a existência de uma comunidade internacional de usuários e programadores, acessível através de Internet, dispostos a colaborar na solução de eventuais problemas com os programas. Não é incomum de se obter auxílio do próprio projetista ou programador do aplicativo ou sistema para o qual se busca auxílio¹⁶.

No caso de instituições, provavelmente o que se deseja é suporte 24 horas por dia, 7 dias por semana, (regime 24x7), e/ou através de linha 0800. Este tipo de serviço já existe no Brasil, provido por Conectiva e IBM, dentre outros. Independentemente de o software ser livre ou proprietário, se a importância de um certo aplicativo para o funcionamento da instituição for suficientemente grande, é muito provável que se estabeleça a capacidade interna de prover suporte e manutenção àquele aplicativo. A alternativa à capacidade interna é a contratação de terceiros para efetuar tal serviço. O fato de empregar-se software livre não introduz nenhuma mudança fundamental nestes dois cenários.

No que concerne à capacitação interna ao suporte e manutenção de sistemas, o uso de software livre possibilita a obtenção de conhecimento muito além daquele possível de se obter em treinamento por empresas de software proprietário. Isto decorre de o código fonte poder ser livremente consultado pelos programadores. Este não é o caso com software proprietário, justamente porque não há acesso ao código fonte e o proprietário é quem decide o que pode ou não ser divulgado a respeito do funcionamento e operação de seus sistemas.

O suporte disponível ao software livre ocorre sob um paradigma diferente daquele aplicado ao software proprietário. No caso de software livre, o programador *pode* estender e adaptar um aplicativo porque se têm acesso ao código fonte. Geralmente, adaptar um programa não é necessário *mas é possível*. No caso de software proprietário, mesmo que seja necessário adaptar ou alterar um aplicativo, o pessoal de suporte fica limitado a solicitar a alteração ao fabricante, e a resignar-se a reinstalar o sistema em caso de pane.

Com a popularização do sistema GNU/Linux que vem ocorrendo desde 1998, há um razoável número de títulos em Português que trata especificamente da administração e uso de Linux¹⁷. A existência de bibliografia farta alivia mas não resolve completamente o problema da falta de suporte, especialmente para usuários iniciantes. Por exemplo, pode ser difícil de se obter suporte técnico e/ou *drivers* para hardware introduzido recentemente no mercado.

Sistemas e aplicativos configuráveis A maioria dos aplicativos distribuídos como software livre podem ser configurados de muitas maneiras distintas pelos usuários. Isso é consequência de que os programadores que desenvolvem estes aplicativos estarem habituados a trabalhar com sistemas altamente configuráveis e portanto projetam seus sistemas

¹⁶A língua franca na Internet é o Inglês, e isto pode dificultar o acesso à esta forma de auxílio.

¹⁷Em 21 de julho de 2001, o autor encontrou sessenta e dois títulos contendo a palavra 'Linux' no catálogo eletrônico de uma grande livraria de Curitiba, dos quais haviam 14 nas prateleiras. No catálogo haviam dois títulos da série *for dummies*.

com estas mesmas características. Uma vez que os sistemas são distribuídos, novas opções são adicionadas por outros programadores e redistribuídas, ou são solicitadas ao autor e então introduzidas nas novas versões. O resultado são sistemas com grande número de opções que podem ser definidas pelos usuários.

Geralmente, os sistemas são distribuídos com um conjunto de opções pré-estabelecidas pelo autor e que satisfazem a um hipotético usuário ‘comum’ ou ‘mediano’. Caso estas não sejam adequadas, novas escolhas podem ser efetuadas individualmente pelos usuários. Para usuários habituados a sistemas comerciais, nos quais freqüentemente não existem alternativas àquelas do usuário ‘comum’, o universo de opções nos aplicativos de software livre resulta em aparentes complexidade e dificuldade de operação. Com freqüência maior que o desejável, a documentação não é suficientemente clara, ou não é escrita para usuários iniciantes, o que reforça a dificuldade no uso de certos aplicativos e/ou modos de configuração.

1.3.2 Desvantagens

Dentre as desvantagens associadas à utilização de software livre salientam-se aquelas listadas abaixo, e detalhadas nos parágrafos que seguem.

- Interface de usuário não é uniforme nos aplicativos;
- instalação e configuração pode ser difícil; e
- mão de obra escassa e/ou custosa para desenvolvimento e/ou suporte.

Interface de usuário inconsistente Não existe um ambiente integrado e com interfaces de usuário uniformes disponíveis em software livre. Esta situação está sendo remediada e existem projetos em estado adiantado de desenvolvimento com o objetivo de prover interfaces de usuário consistentes através de um conjunto completo de aplicativos, tais como editor, *browser*, planilha, calendário, etc. Os projetos KDE (K Desktop Environment), GNOME (GNU Object Model Environment) e OpenOffice¹⁸ têm por objetivo prover ambientes integrados e interfaces de usuário uniformes. Estes já estão disponíveis como plataforma estável e estima-se que atingirão plena maturidade até final de 2002.

Uma das conseqüências do desenvolvimento descentralizado é a inexistência de um estilo uniforme ou formatação bem-definidos para a interface de usuário entre os vários aplicativos, ao contrário do que ocorre com software *de um* proprietário. Esta é uma herança parcial do Unix, cujos aplicativos possuem interface de comandos muito heterogênea (como os editores *vi* e *emacs*, por exemplo), além da ênfase em interface de usuário baseada em caracteres. Não existe consistência de interface de usuário entre os vários aplicativos comumente distribuídos (editores e *browsers*) nem na forma de configuração destes aplicativos após sua instalação. Tais características dificultam o uso destes sistemas e tendem a afastar usuários iniciantes.

Instalação e configuração difíceis Por causa da própria natureza do software livre, e da maneira como este é desenvolvido, os primeiros usuários dos programas são programadores e/ou usuários avançados, que têm prática na instalação e na utilização de programas

¹⁸www.openoffice.org

que ainda se encontram em estágios finais de desenvolvimento. Estes são os usuários que inicialmente definem as características adicionais a serem acrescentadas aos programas, e de certa forma definem os parâmetros sob os quais os programas são usados por todos os usuários.

Portanto, em comparação com produtos de prateleira, sistemas como o Linux são mais difíceis de instalar e configurar, especialmente por usuários inexperientes. Esta situação têm evoluído rapidamente por causa do aumento da população de usuários (massa crítica) e das demandas por eles impostas.

Outro aspecto que tende a dificultar a instalação e configuração do ambiente de trabalho para usuários iniciantes é o grande número e variedade de aplicativos com funções similares ou equivalentes. Por exemplo, uma distribuição GNU/Linux contém vários editores de texto (`vi`, `emacs`, `xedit`, etc), vários programas para leitura de *e-mail* (`netscape`, `pine`, `elm`, `mutt`, etc), vários agentes de transmissão de *e-mail* (`sendmail`, `smail`, `qmail`, `exim`, etc). Enquanto que esta variedade contempla as preferências de diversas classes de usuário, a escolha e configuração destes pode ser assustadoramente complexa para um iniciante.

Finalmente, a administração de sistemas como o Linux não é tarefa simples porque a administração de um sistema operacional similar ao Unix é complexa. Algumas das características mais interessantes deste sistema operacional, como o sistema de permissões por exemplo, adiciona um nível de complexidade inexistente nas várias versões do sistema Windows. Há uma série de tarefas associadas à administração de sistemas Unix que demandam atenção continuada por pessoal com certo nível de conhecimento. Estas tarefas são devidamente escondidas dos usuários de Windows e MacOS, por exemplo. A comunidade de software livre também está se dedicando a minimizar a complexidade da administração de sistemas.

Considere-se a instalação do Windows 98, que é relativamente simples e rápida¹⁹ porque empresas como a Microsoft, cuja clientela é majoritariamente de usuários inexperientes, mantém departamentos inteiros de pesquisa e desenvolvimento, trabalhando justamente para desenvolver interfaces de usuário amigáveis e mecanismos de instalação que prescindam de intervenção. Enquanto relativamente fácil, o mecanismo de instalação e administração do Windows é absolutamente inflexível e a facilidade de administração é ilusória. O que ocorre é que os usuários *se habituaram a reinstalar todo o sistema a cada pane*.

Mão de obra escassa O sistema operacional Unix é muito popular na academia e em entidades de pesquisa e desenvolvimento. Estas instituições têm produzido pessoal capacitado em quantidade suficiente para administrar e operar redes de máquinas executando alguma(s) das várias versões de Unix.

Existem evidências de que em praticamente toda a grande empresa de informática no mundo há alguém trabalhando no desenvolvimento de software livre [4, 3]. Na medida em que o uso do sistema GNU/Linux se difunde, a demanda por pessoal capacitado para sua instalação, operação e administração, bem como por programadores, tende a aumentar. As universidades, principalmente as públicas, vêm formando pessoal qualificado no uso e administração de sistemas Unix, e mais recentemente em Linux por ser este gratuito e especialmente por seu código fonte estar disponível. Há indicações de que o número

¹⁹Exceto se ocorrer algum problema na detecção automática do hardware.

de técnicos qualificados é pequeno frente à demanda e portanto estes técnicos tornam-se mão de obra relativamente custosa, tanto para desenvolvimento como para suporte e administração de sistemas. O suporte aos sistemas de software livre não é intrinsecamente mais custoso, o que ocorre é que ainda não foi atingida a massa crítica em termos de usuários e conseqüentemente de pessoal de suporte. A massa crítica de usuários Windows, por exemplo, foi atingida há alguns anos, havendo portanto abundância de pessoal capaz de prestar auxílio. Cabe ressaltar que a qualidade do suporte varia muito entre as duas comunidades por causa das restrições impostas pelo proprietário do software.

As ferramentas de desenvolvimento distribuídas como software livre são menos amigáveis que as versões disponíveis comercialmente. Assim, em que pese sua qualidade, as ferramentas de desenvolvimento para Linux são mais primitivas e exigem um maior nível de sofisticação por parte do desenvolvedor, o que também aumenta o déficit de pessoal capacitado. Esta situação deve se amenizar com o recente lançamento pela Borland de ferramentas de desenvolvimento Kylix (Delphi), com suporte ao estilo 'visual' de programação.

1.3.3 Desvantagens Imaginárias

Dentre as características de software livre que são imaginadas como desvantagens salientam-se os aspectos listados abaixo e discutidos nos parágrafos seguintes.

- Ausência de proprietário ou responsável legal;
- suporte segundo o modelo tradicional é inexistente;
- instabilidade; e
- poucos aplicativos comerciais.

Ausência de proprietário e/ou responsável legal Do ponto de vista de uma empresa, um dos problemas mais sérios com a adoção de software livre é a inexistência de uma entidade com identidade jurídica claramente definida e que seja legalmente responsável pelos programas, aplicativos e sistemas. Em caso de prejuízos decorrentes de erros no software, não há nenhuma entidade que possa ser responsabilizada civil ou criminalmente por eventuais perdas e/ou danos.

O simples fato de existir um proprietário do software, e portanto legalmente imputável, não provê necessariamente garantia quanto a prejuízos decorrentes de erros ou falhas nos sistemas. Pelo contrário, freqüentemente o proprietário se exime de qualquer responsabilidade por danos ou prejuízos decorrentes da *utilização correta* de seus produtos. Abaixo encontra-se uma ilustração das práticas adotadas pelos produtores de software comercial, relacionadas à garantia de seus produtos de software. A garantia mostrada abaixo foi escolhida por ser representativa daquilo a que a maioria dos usuários está sujeita, e é uma transcrição da licença de software para o Windows 98 fornecido já instalado em um computador Compaq Presario 1200, adquirido em junho de 2000.

GARANTIA E DISPOSIÇÕES ESPECIAIS PARA O BRASIL

Garantia Limitada. O Fabricante garante que (a) o SOFTWARE desempenhará suas funções substancialmente em conformidade com a documentação escrita que o acompanha, por um período de 90 (noventa) dias a contar da data de entrega;

e (b) qualquer equipamento Microsoft fornecido juntamente com o SOFTWARE estará isento de defeitos com relação aos materiais e mão-de-obra empregados, por um período de 1 (um) ano, a contar da data de entrega, sob condições normais de uso e manutenção. Quaisquer garantias implícitas com relação ao SOFTWARE e aos equipamentos Microsoft estão limitadas a 90 (noventa) dias e 1 (um) ano, respectivamente.

Direitos do Cliente. A responsabilidade integral do Fabricante e de seus fornecedores e o único direito de V.Sa. será, a critério do Fabricante, (a) a devolução do preço pago, ou alternativamente, (b) o conserto ou substituição do SOFTWARE ou equipamento que não satisfaça os termos desta Garantia Limitada, sujeito à devolução do mesmo ao Fabricante juntamente com uma cópia do respectivo recibo. Esta Garantia Limitada ficará prejudicada e não gerará efeitos se a falha ou defeito do SOFTWARE ou do equipamento resultar de acidente, utilização abusiva ou inadequada. Qualquer SOFTWARE ou equipamento substituído será garantido pelo prazo remanescente da garantia original ou por 30 (trinta) dias, no caso deste último prazo ser mais extenso.

Garantias Únicas. Sujeito à legislação pertinente, o Fabricante e seus fornecedores não outorgam quaisquer outras garantias, sejam expressas ou implícitas, incluindo, sem limitação, garantias de comercialização ou adequação para determinada finalidade, com relação ao SOFTWARE, documentação correlata e qualquer equipamento que o acompanhe. Esta Garantia Limitada confere a V.Sa. direitos específicos.

Limitação de Responsabilidade (Danos Indiretos). Sujeito à legislação pertinente, em nenhuma hipótese o Fabricante ou seus fornecedores serão responsáveis por quaisquer danos (incluindo sem limitação danos diretos ou indiretos resultantes de lesão corporal, lucros cessantes, interrupção de negócios, perda de informações ou outros prejuízos pecuniários) decorrentes de uso ou da impossibilidade de usar este produto, ainda que o Fabricante tenha sido alertado quanto a possibilidade destes danos. Em qualquer caso, a responsabilidade integral do Fabricante e de seus fornecedores sob este Contrato limitar-se-á ao valor efetivamente pago por V.Sa. pelo SOFTWARE e/ou equipamento Microsoft.

Este Contrato é regido pelas leis da República Federativa do Brasil.

Como fica claro da leitura do texto desta licença, especificamente o item *Limitação de Responsabilidade (Danos Indiretos)*, o fornecedor se exime de qualquer responsabilidade quanto a compensações devidas por prejuízos causados pela utilização do produto *mesmo que correta de acordo com a documentação fornecida*. Note ainda que “o SOFTWARE desempenhará suas funções substancialmente em conformidade com a documentação escrita”, indicando que a documentação fornecida não é uma especificação confiável do produto. Em muitas aplicações, a simples perda de dados, ou perda de tempo por causa de sistemas que ‘congelam’ pode representar prejuízos bastante significativos, mas não há hipótese de ressarcimento.

Este tipo de prática comercial só é possível porque produtos de software têm status legal similar a trabalhos artísticos, não existindo nenhuma garantia de que executarão funções úteis (mesmo que anunciadas na propaganda), nem de que foram projetados ou produzidos com a tecnologia mais adequada e à disposição do produtor na época em que o produto

foi comercializado. Isso também significa que, a menos de disposições específicas²⁰, os produtores do software não são obrigados a corrigir os erros detectados em seus produtos. A título de exemplo, é mais fácil de se obter judicialmente compensação de um mecânico de automóvel por executar seu trabalho de forma inadequada, do da Microsoft porque seus produtos não operam de acordo com o anunciado, ou por quaisquer perdas ou danos causados por erros no, ou falhas do, software vendido por aquela empresa.

Quando o fabricante publica correções para os erros em seus produtos, a responsabilidade pela aplicação dos remendos fica inteiramente por conta do comprador, que deve dispende o tempo e a energia para aplicar as ‘correções’ a um produto pelo qual já pagou. No caso da Microsoft, as correções são disponibilizadas gratuitamente na Internet, mas o comprador deve conectar-se à rede para obter uma cópia das correções e então aplicá-las. Periodicamente, uma *nova* versão do produto que inclui uma série de correções é publicada, mas geralmente esta ‘nova’ versão não é gratuita.

Suporte inexistente pelo modelo tradicional O suporte no modelo tradicional, em regime 24x7, existe para sistemas baseados em software proprietário com custos que variam de acordo com o produto suportado. Como mencionado na Seção 1.3.1, este tipo de serviço já está disponível no Brasil para sistemas baseados em software livre, especialmente Linux.

Quanto ao suporte telefônico gratuito através de linhas 0800, na maioria dos casos este serviço está longe de ser satisfatório. Portanto, a reclamação sobre a falta de suporte telefônico adequado para software livre se aplica a um serviço que é pouco satisfatório mesmo para software proprietário.

Por outro lado, o suporte a software livre disponível através da Internet é considerado ágil, eficiente e eficaz. Existe um sistema de suporte através da Internet, provido pela comunidade Linux brasileira, que ameniza este problema. A lista de discussão *dicas-l*, em www.dicas-l.unicamp.br, é um meio de difusão de notícias e contatos técnicos para usuários de software livre, além de outros assuntos. O sistema *rau-tu*²¹ contém um repositório dinâmico de informações sobre sistemas computacionais, sendo que a seção sobre Linux é uma das mais ativas. Existem empresas que prestam serviços de treinamento e de assistência técnica a sistemas baseados em software livre, sendo a mais notável delas a Conectiva, embora não seja a única. Existem empresas pequenas e profissionais autônomos que prestam estes mesmos serviços.

Instabilidade Existe a falsa crença de que *software ‘grátis’ não pode ter boa qualidade*. Há confusão e alguma falta de informação reunidos nesta crença. O software livre não é ‘grátis’, embora freqüentemente seja distribuído sem custos. Quanto à falta de informação, cumpre lembrar que há uma fração bastante significativa da infraestrutura da Internet que é composta por software livre. A própria Internet tornou-se viável por causa da disponibilidade de código fonte de implementações da pilha de protocolos TCP/IP²². De forma similar, a linguagem C tornou-se um padrão de fato na indústria porque o compilador *gcc* estava disponível sob a licença GPL, o que permitiu seu transporte para praticamente todas as plataformas populares.

²⁰Disponíveis apenas para clientes especiais.

²¹Do Inglês *how-to*, em www.rau-tu.unicamp.br

²²O código dos protocolos foi disponibilizado pelo Computer Science Research Group da Universidade de Berkeley no BSD Unix.

Os sistemas listados abaixo são distribuídos sob as licenças mencionadas na Seção 1.1 e são empregados há vários anos em funções críticas para a Internet, em especial **bind**, **sendmail** e **apache**.

bind – o serviço de nomes na Internet que faz a tradução de uma url para o número IP que lhe corresponde (de www.socinfo.org.br para 200.130.37.2), é definido pelo protocolo Domain Name Service (DNS), e depende quase que exclusivamente de **bind**, ou Berkeley Internet Name Daemon²³.

sendmail – o agente de transferência de correio eletrônico **sendmail**²⁴ é distribuído como software livre e é largamente empregado na Internet.

apache – o servidor de conteúdo que é usado em pelo menos 40% dos sítios na Internet é o servidor **apache**²⁵ (ou *a patchy server*), distribuído como fonte aberto (*open source*), e como parte de produtos da IBM. A pesquisa da Netcraft indica que o **apache** detém mais de 40% do mercado de servidores de conteúdo desde 1997²⁶.

perl – o interpretador **perl**, ou Practical Extraction and Report Language²⁷ (ou ainda *Pathologically Eclectic Rubbish Lister*), é usado para criar formulários e páginas ativas e é um pacote de software livre extremamente popular.

gcc, **emacs**, **gdb** – os editores (**emacs**), compiladores (**gcc**, **g++**, **egcs**), e depuradores (**gdb**, **xxgdb**) desenvolvidos pela Free Software Foundation²⁸ são amplamente usados na indústria e academia.

Poucos aplicativos comerciais Correntemente existem vários sistemas de aplicação comercial disponíveis como software livre, ou sistemas proprietários que foram portados para o sistema GNU/Linux. Por exemplo, na classe de aplicativos de escritório existem os pacotes da Applix, StarOffice, Lotus, Corel Word Perfect, além dos sistemas Gnome, KDE e OpenOffice, citados anteriormente. A Oracle comercializa uma versão de seu banco de dados e aplicativos auxiliares para o sistema GNU/Linux. A Borland²⁹ comercializa o sistema Kylix [25] para desenvolvimento de aplicativos, também para o sistema GNU/Linux. Além destes, IBM e SAP estão disponibilizando versões de aplicativos empresariais para computadores de grande porte para GNU/Linux.

O número e a variedade de aplicativos como folha de pagamento, contas a pagar, e similares é de fato pequeno. Isso deve-se à pequena demanda por versões destes sistemas que executem em plataformas livres, o que por sua vez, deve-se à percepção pelos empresários do software livre como algo marginal, usado por *hackers*³⁰, e de caráter acadêmico. Esta situação decorre da ‘novidade’ do software livre como plataforma para aplicações comerciais, e do conservadorismo de uma fração significativa do empresariado tradicional, que por diversas razões, tendem a ser ou últimos a adotar tecnologias tidas como avançadas.

²³<http://isc.org/bind.html>

²⁴www.sendmail.org

²⁵www.apache.org

²⁶Detalhes em <http://www.netcraft.co.uk/Survey>

²⁷www.perl.com

²⁸www.gnu.org e www.fsf.org

²⁹www.borland.com

³⁰Infelizmente, esta palavra adquiriu uma conotação pejorativa e é frequentemente associada a atividades ilegais. Na sua conotação original *hacker* significa “programador (muito) competente”.

1.4 Treinamento e Capacitação

Os autores do estudo intitulado *Being Fluent with Information Technology* concluem que um programa de capacitação em tecnologias da informação e comunicações (TICs) deve incluir um currículo que forneça aos alunos fluência no uso do computador, definida como “uma firme compreensão do que é necessário para usar efetivamente a tecnologia da informação em uma gama de aplicações” [15]. Além de prover fluência na utilização de software, os indivíduos capacitados devem ser capazes de aplicar a tecnologia da informação em novos problemas, com a compreensão das conseqüências desta atividade. Evidentemente, “estas capacidades transcendem aplicações particulares de hardware e software”.

Uma das dificuldades a serem enfrentadas na adoção de sistemas de software livre em larga escala é a necessidade de maior capacitação dos operadores destes sistemas. Por operadores entenda-se os técnicos que instalam, configuram e mantêm os sistemas em operação. Dada a natureza própria do software livre, especialmente seu modo de desenvolvimento, a instalação e configuração destes sistemas exige maior nível de conhecimento técnico do que simplesmente “inserir o CD e re-instalar o sistema”. A necessidade de maior capacitação é a contrapartida à flexibilidade e configurabilidade do software livre.

A comunidade de desenvolvedores tem trabalhado e obtido sucesso em simplificar os mecanismos de instalação e configuração de sistemas. Por exemplo, a instalação da versão doméstica das distribuições do sistema GNU/Linux Debian, Red Hat, Conectiva ou Mandrake, possuem um grau de dificuldade comparável ao da instalação de sistemas comerciais. No caso de sistemas mais complexos e baseados em redes locais, servidores de arquivos e acesso à Internet, o grau de complexidade é evidentemente maior. A grande diferença reside em que os sistemas de software livre são fartamente documentados e podem ser adaptados e adequados às necessidades de cada organização. A robustez e confiabilidade destes sistemas libera os técnicos para executar atividades mais interessantes do que a constante re-instalação de sistemas proprietários instáveis e inseguros.

No que concerne ao usuário não-técnico, os sistemas de escritório disponíveis como software livre possuem graus de usabilidade comparáveis ao de sistemas proprietários equivalentes. Os sistemas de escritório, especialmente Gnome, KDE, StarOffice e OpenOffice, possuem funcionalidades similares aos sistemas da Microsoft, e a estabilidade destes sistemas é comparável à de sistemas proprietários.

Certificação Considerando-se o nível de capacitação necessário aos técnicos encarregados de instalar e manter sistemas de software livre, é interessante que sejam criados mecanismos de certificação de técnicos. Esta certificação poderia contemplar diferentes níveis de capacitação, como por exemplo a certificação de (i) *usuário* que é dotado de fluência no uso de sistemas de escritório e Internet, (ii) *instalador* que é capaz de instalar e configurar sistemas relativamente complexos, e de (iii) *desenvolvedor* que é capaz de projetar e desenvolver aplicativos e sua documentação, de acordo com os padrões de distribuição de software livre. A agência facilitadora, ou a entidade homologadora, poderia coordenar, preparar e aplicar os exames de certificação nos vários níveis de capacitação³¹

³¹Algumas das iniciativas neste sentido são Linux Certified System Engineer Project www.okcforum.org/~mark/mark/lcse Linux Professional Institute (LPI) www.LPI.org, Linux Certification www.linuxcertification.com, Red Hat Training www.redhat.com/services/training/training.html.

1.5 Combate ao Monopólio da Informação

Um dos maiores problemas associados ao software proprietário é a utilização de formatos proprietários para a codificação da informação manipulada pelos aplicativos. Este problema é especialmente sério no caso dos conjuntos de aplicativos para escritório, por causa da popularidade destes aplicativos e pelo virtual monopólio dos produtos de um único fabricante. A discussão que segue exemplifica o problema com o editor de textos, mas a argumentação vale para as planilhas de cálculo, *browsers* e outros aplicativos populares.

Considere-se, a título de exemplo, o uso difundido em uma organização de um editor de textos que emprega um formato proprietário, que chamaremos de formato X_1 . Após um período inicial de acomodação de toda a organização ao novo editor, uma parcela significativa de toda a documentação produzida passa a ser gerada e armazenada com a formatação daquele editor, no formato X_1 .

Devido às pressões de mercado, o fornecedor do editor lança uma nova versão do produto, com uma série de características adicionais, *com um formato X_2 que é ligeiramente incompatível com o formato X_1* . Em geral, as novas versões são compatíveis com as (imediatamente) anteriores, exceto pelas novas características. Assim que alguma das novidades passe a ser usada regularmente por parte relevante da organização (central de pagamentos, por exemplo), as cópias do editor devem ser atualizadas para a última versão, para que todos os usuários possam se beneficiar da ‘novidade’. Se a ‘novidade’ em questão é uma melhoria realmente útil, o preço da atualização de todas as cópias pode ter um custo aceitável. Contudo, se a ‘novidade’ for apenas cosmética, paga-se um preço talvez elevado por algo de utilidade discutível. A prática corrente no mercado é o lançamento de uma nova versão a cada 18 ou 24 meses, de forma que os custos advindos da situação aqui descrita são recorrentes.

Outro problema está na incompatibilidade do formato proprietário com os formatos de produtos similares. Como a maior parte dos documentos da organização está no formato X_1 , qualquer proposta de adoção de outro produto encontrará sérias resistências por causa da cultura já estabelecida, bem como por causa dos documentos que já estão no formato X_1 e que deverão ser convertidos para o novo formato, e dos custos associados a conversão dos documentos no formato X_1 para o formato Y . A possibilidade do uso concorrente de dois formatos incompatíveis não é atrativa.

A situação de dependência de um fornecedor de software proprietário é freqüente, quando se considera sistemas de bancos de dados ou outros aplicativos comerciais, por exemplo. Se, ao invés de uma única organização comercial privada, considerarmos a situação em termos de governo, a gravidade do problema toma dimensões críticas. Por exemplo, uma das aplicações governamentais de maior sucesso no país é o sistema de coleta de Declarações de Imposto de Renda pela Internet. O aplicativo distribuído pelo Ministério da Fazenda, através da Secretaria da Receita Federal, somente executa em sistemas operacionais da Microsoft. Se o contribuinte não possuir este um destes sistemas, ele é obrigado a preencher um formulário em papel, que será tratado somente após o processamento de todos os formulários entregues por via eletrônica. Desta forma, o próprio Governo Federal induz o cidadão a adquirir produtos de um único fornecedor.

Além do Imposto de Renda, várias outras instâncias de governo adotam postura similar. Na área de pesquisa e desenvolvimento, as duas maiores agências federais de fomento à pesquisa, exigem que as solicitações de recursos sejam enviadas através de formulários

eletrônicos que somente podem ser preenchidos em determinados produtos da Microsoft. As universidades públicas são obrigadas a dispende parte de seu orçamento na aquisição de software proprietário se desejarem obter recursos daquelas agências [2].

Do ponto de vista do país como um todo, a dominação do mercado de aplicativos de escritório por um único fornecedor estrangeiro, de tal forma que uma parcela grande da economia nacional depende destes sistemas, pode ter conseqüências no mínimo desagradáveis em caso de quebra daquele fornecedor, ou de distúrbios nas relações entre o Brasil e o país sede do fornecedor. Sendo este fornecedor estrangeiro, a virtual obrigatoriedade da aquisição destes produtos, como discutido acima, onera o balanço de pagamentos do país.

A adoção de Software Livre nestes casos reduziria a transferência de recursos para o exterior, que é inescapável por causa do monopólio. A migração de sistemas proprietários amplamente difundidos para software livre não ocorreria sem os custos associados à instalação dos aplicativos e ao treinamento dos usuários. Estas atividades possibilitariam a criação de novos empregos em pequenas e médias empresas de desenvolvimento, suporte e manutenção [3, 16]. Os incentivos à adoção de software livre são discutidos em mais detalhe no Capítulo 2.

Capítulo 2

Incentivos à Adoção de Software Livre

2.1 Ações de Governo

O Capítulo 1 define *software livre* e enumera as vantagens advindas de sua utilização. No Brasil, a utilização de software livre vem crescendo rapidamente nos últimos anos, conforme demonstra a pesquisa conduzida pelo Grupo Temático de Pesquisa e Desenvolvimento deste Programa [12]. Apesar do crescimento e da divulgação na mídia, ainda não se estabeleceu no país uma comunidade suficientemente grande para manter-se de forma auto-sustentada. Para que isso ocorra, são necessárias ações em nível de governo para incentivar a adoção de software livre no próprio governo e em suas agências. A demanda, em termos de serviços e desenvolvimento, gerada pelas aplicações governamentais criaria as condições para o estabelecimento de uma comunidade de usuários e desenvolvedores capaz de manter-se ativa indefinidamente.

A próxima seção contém uma breve descrição dos sistemas em uso nos três níveis de governo e indica as oportunidades para a adoção de soluções baseadas em software livre. A Seção 2.1.2 lista uma série de ações que devem ser executadas pelos governos para acelerar a adoção de software livre nas aplicações em que seu uso não seja inviável.

2.1.1 Sistemas de Informação de Governo

A grosso modo, os sistemas de informação de governo dividem-se em quatro classes: (i) sistemas de grande porte, (ii) sistemas de administração interna, (iii) sistemas de interação com o cidadão, e (iv) sistemas de escritório. Estas classes de sistemas representam diferentes desafios e oportunidades para a adoção de soluções baseadas em software livre.

Dentre os sistemas de grande porte enquadram-se aqueles que armazenam e processam as bases de dados com informações sobre os cidadãos adstritos à esfera administrativa da União, Estado ou Município. Por exemplo, as bases de dados do Departamento Nacional de Trânsito (DENATRAN) mantém registros sobre toda a frota automotiva do país, bem como de todos os cidadãos habilitados a conduzir veículos automotores. No nível estadual o DETRAN mantém um sub-conjunto das bases nacionais, e no nível municipal o DIRETRAN mantém um sub-conjunto das bases estaduais. Normalmente, estes sistemas são

implantados em plataformas de grande porte e os aplicativos evoluem lentamente.

Nos sistemas de administração interna estão incluídos os sistemas de armazenamento e processamento de informações administrativas do próprio governo, tais como controle de patrimônio e folha de pagamento. Normalmente, estes sistemas são implantados em plataformas que são “proporcionais” ao porte da unidade federativa ou administrativa. Estes sistemas também evoluem com relativa lentidão.

Dentre os sistemas de interação com o cidadão destacam-se os programas como os da Receita Federal para fins de declaração de rendimentos de pessoa física ou pessoa jurídica, das Receitas Estaduais para fins de arrecadação de tributos estaduais, e sistemas de informação ao cidadão.

Os sistemas de escritório incluem os aplicativos básicos tais como editores de texto, planilhas eletrônicas e sistemas de correio eletrônico. Estes programas são usados em tarefas administrativas rotineiras, e em alguns sistemas administrativos de menor porte ou abrangência.

Sistemas de Grande Porte

Os sistemas de grande porte de nível nacional, e nos estados mais populosos, possuem características de sistemas legados porque estão em uso há vários anos e vem sendo adaptados em função de alterações na legislação e nas práticas de administração pública. Os conteúdos das bases de dados geralmente estão codificados com formatos proprietários e contém milhões de registros, coletados ao longo de vários anos. A migração destes sistemas para plataformas livres pode ter custos elevados por conta da dificuldade na tradução do código do aplicativo para uma linguagem aberta, e na transcodificação dos dados para um formato não-proprietário. Apesar da possibilidade de os custos serem elevados, a transcodificação dos dados para formatos abertos e não-proprietários libera o Estado do relacionamento com características monopolistas com o fabricante do sistema.

Os mecanismos de acesso a estas bases de dados podem ser alterados de forma a que inteligência mantida nos programas usados para consultas (clientes) seja transferida para um programa instalado junto às bases de dados que realize as consultas (servidor), de maneira similar aos serviços bancários disponíveis na Internet. Assim, o usuário necessita usar apenas um *browser* para interagir com os aplicativos e com as bases de dados¹. É importante garantir que os programas no servidor obedeçam fielmente aos protocolos de comunicação abertos e a formatos de dados que sejam abertos e de acesso público. Caso contrário, grupos de usuários podem ser excluídos do acesso às informações simplesmente porque os *browsers* que usam não tem capacidade para exibir dados, ou executar programas, que necessitem de tecnologias proprietárias para funcionar.

Sistemas Administrativos

A disponibilidade de sistemas administrativos desenvolvidos como software livre pode criar as condições para que municípios pequenos venham a empregar sistemas de informação relativamente sofisticados, aumentando assim a eficiência e eficácia no uso dos recursos

¹Por exemplo, o Estado do Paraná disponibiliza vários serviços ao cidadão através da Internet segundo este modelo.

públicos. Isso traria economia em licenças, na utilização continuada de equipamentos pré-existentes, e nas eventuais adaptações e evolução dos aplicativos. Dependendo do tipo de sistema, a manutenção do software poderia ficar sob responsabilidade de entidades públicas (prefeituras ou estados com maior capacidade tecnológica), ou de entidades como o Instituto Brasileiro de Administração Municipal (IBAM), por exemplo.

Sistemas de Interação com o Cidadão

Quanto aos sistemas de interação com o cidadão, há um exemplo óbvio de cerceamento da liberdade do cidadão, que é o sistema de declaração eletrônica de rendimentos de pessoa física da Receita Federal. O sistema disponibilizado pela Receita, em que pesem suas várias qualidades técnicas e de usabilidade, somente pode ser executado em sistemas vendidos pela Microsoft. Quem não dispuser de alguma versão de Windows deve preencher um formulário em papel, formulário este que será processado pela Receita posteriormente a todas as declarações entregues em formulários eletrônicos. Esta é uma situação que deve e pode ser evitada porque ela limita o acesso do cidadão às melhorias de serviços possibilitadas pela tecnologia. Além disso, existe o aspecto ético da questão em que um órgão importante do Governo Federal age para reforçar a dominação monopolista no mercado brasileiro de software de uso pessoal ao prover um sistema extremamente útil, mas que só pode ser executado em sistemas vendidos por uma única empresa estrangeira.

Sistemas de Escritório

A utilização de software livre em programas de escritório é um desafio maior que os descritos até agora porque envolve o re-treinamento de grande número de usuários. Além disso, deverá ser efetuada a migração de grandes volumes de documentos e dados, que em sua maioria são correntemente armazenados em formatos proprietários, para formatos abertos. Estas duas atividades demandarão cuidadoso planejamento e execução. Existem várias iniciativas neste sentido, destacando-se entre elas os governos municipais de Belo Horizonte, Campinas, Porto Alegre e São Paulo, e do governo estadual do Rio Grande do Sul². Apesar dos custos envolvidos, a troca de sistemas proprietários para sistemas abertos, e a migração de dados para formatos abertos serão benéficos para os administradores públicos, pela redução nos gastos com compra, uso e manutenção de software, para o contribuinte e a sociedade, pelo melhor uso dos recursos públicos, oportunidades para desenvolvimento local de software, e principalmente o controle sobre os aplicativos e dados usados como insumos da gestão dos recursos públicos.

2.1.2 Ações de Incentivo

A discussão sobre as vantagens técnicas e sócio-econômicas inerentes ao modelo de produção e utilização de software livre na Seção 1.3, juntamente com as oportunidades de utilização apresentadas acima, demonstram a necessidade de ações em nível de governo para promover e incentivar a adoção de software livre em escala nacional.

Os três níveis de governo devem incentivar a adoção de software livre em suas próprias atividades. A adoção e recomendação de uso de software livre implicará em vários tipos

²www.softwarelivre.rs.gov.br

de demandas que, em curto e médio prazo, terão o efeito de criar as condições para o estabelecimento da massa crítica de usuários e desenvolvedores no país. Além disso, o incentivo governamental sinaliza ao mercado produtor de software sobre a necessidade de adaptação a este modo de produção, e sobre a necessidade de capacitação de desenvolvedores e usuários, além das atividades de suporte.

As atividades de promoção devem incluir as ações listadas abaixo e descritas em maior detalhe nos próximos parágrafos. Alguns tópicos desta lista foram adaptados de recomendações contidas em [17, 16, 18, 14].

- Incentivo ao uso de software livre;
- recomendação para adoção de software livre;
- uso do poder de compra;
- adoção prioritária de protocolos abertos;
- aquisição de direitos sobre o código fonte de todo o software;
- implantação de mecanismos de financiamento;
- criação de incentivos fiscais ao uso e desenvolvimento;
- implantação de mecanismos de capacitação ao uso;
- criação de uma agência facilitadora; e
- avaliação do impacto econômico e social.

Incentivo ao uso de software livre É necessário um esforço no sentido de esclarecer, e talvez até mesmo de convencer, a todos os níveis decisórios da Administração Pública quanto às vantagens decorrentes da adoção de software livre, para que soluções baseadas nestas tecnologias sejam consideradas como bons candidatos à adoção pelas instituições de governo.

Recomendação para adoção de software livre Os governos, além de incentivar o uso e a adoção de software livre, devem também recomendar a adoção de soluções baseadas em software livre nas instituições de governo, públicas e autarquias. Estas recomendações devem obrigar a que sistemas de software livre sejam considerados para aquisição sempre que tais sistemas ofereçam alternativas viáveis a sistemas proprietários.

Uso do poder de compra O poder de compra do Estado deve ser usado através de abertura para a participação de sistemas abertos baseados em software livre nas licitações e concorrências para aquisição de sistemas e serviços, sempre que isso for adequado e possível. As compras governamentais geralmente são de grande escala, e isso representa um atrativo comercial significativo. Além disso, os sistemas adotados pelo governo tendem a se tornar padrões de fato, se já não o são de direito.

Adoção prioritária de protocolos abertos A obrigatoriedade da adoção de protocolos abertos é uma salvaguarda imprescindível para evitar que o Estado fique refém de tecnologias proprietárias, como discutido na Seção 1.5. Mantendo-se as tendências atuais, e nada indica que haverá mudanças, grande parte da interação do cidadão com o Estado passará a se dar através da Internet. Se os protocolos de comunicação empregados em

aplicações governamentais não forem abertos, outras áreas importantes de atuação do Estado podem ficar comprometidas em situações similares àquela do Imposto de Renda, o que seria catastrófico para a cidadania.

A concepção do Sistema do Cartão Nacional de Saúde³ (SCNS) é pioneira na adoção de protocolos abertos em um sistema de governo de âmbito nacional [10, 7]. Todos os protocolos de comunicação em uso no SCNS são abertos e publicados pelo IETF ou W3C. Em iniciativa paralela à implantação do SCNS, o Ministério da Saúde publica na Internet a definição dos dicionários de dados (DTDs XML) que devem ser empregados na comunicação entre os diversos aplicativos da área de atuação daquele Ministério. Todo o software produzido pelos fornecedores do projeto SCNS é de propriedade do Ministério da Saúde.

Aquisição de direitos sobre o código fonte de todo o software Todos os sistemas adquiridos pelo Estado devem ser fornecidos com todo o código fonte dos programas, bem como de todos os métodos e mecanismos de adequação destes sistemas à sua utilização (configurações e/ou ajustes). Isso evitaria dois problemas sérios enfrentados pelos governos, que são a obsolescência dos formatos de armazenamento de dados, e a dificuldade de adaptação dos sistemas para novos usos e aplicações.

Implantação de mecanismos de financiamento Considerando-se os benefícios para a sociedade da produção e distribuição de software livre, as atividades associadas à produção de software livre deveria ser financiadas pelos governos, especialmente aqueles projetos de interesse público, ou aplicações que não despertam o interesse de produtores tradicionais de software. Tal financiamento poderia se dar de forma institucional, através do qual uma ou mais entidades são contratadas para desenvolver um conjunto de aplicativos. Dentre os possíveis atores institucionais destacam-se instituições de ensino, organizações não governamentais ou empresas públicas.

Incentivos fiscais ao uso e desenvolvimento O governo poderia criar incentivos fiscais às empresas que produzem e/ou distribuem software livre, ou que o utilizem de forma intensa. Estes mecanismos poderiam ser extensões da legislação vigente sobre a produção de software. Os incentivos fiscais se justificam plenamente pela economia de divisas advinda da utilização de software não-importado, pelo estímulo ao desenvolvimento auto-sustentado de software livre no Brasil, e por gerar demanda pelo trabalho de desenvolvedores brasileiros.

Implantação de mecanismos de capacitação ao uso O conjunto de instituições que provê treinamento em larga escala sobre produtos de software livre é relativamente pequeno. São portanto necessárias medidas de incentivo aos programas de treinamento, bem como auxílio financeiro a instituições que desejem treinar sua força de trabalho para a aplicação de tecnologias baseadas em software livre. Tais medidas permitiriam acelerar a utilização de software livre, especialmente em pequenas e médias empresas.

³www.saude.gov.br/cartao

Agência facilitadora Uma agência governamental devotada a promover a utilização de software livre se encarregaria do processo de avaliação, e do desenvolvimento e adoção de produtos e tecnologias baseados em software livre, tanto no governo como pela sociedade. Esta agência teria a responsabilidade de publicar diretivas quanto à qualidade dos produtos, preparação de material informativo sobre a tecnologia de software livre, e deveria cooperar na resolução de problemas associados a patentes de software. Além disso, a agência suportaria a formulação de políticas de governo para a adoção de software livre, e na preparação, e adoção, de critérios de financiamento a projetos de informatização que contemplem as peculiaridades do software livre.

Esta agência também deveria manter uma “bolsa de software livre” através da qual projetos de grande interesse social seriam distribuídos a candidatos ao desenvolvimento destes projetos, possivelmente mediante alguma forma de remuneração ou acesso à equipamentos e infra-estrutura. Uma tarefa importante da Agência seria o incentivo ao desenvolvimento de aplicativos de interesse exclusivo do Brasil ou da comunidade de Língua Portuguesa.

Avaliação do impacto econômico e social O impacto econômico e social da adoção de software livre é difícil de avaliar porque não existem indicadores sócio-econômicos apropriados. É portanto necessário trabalho de pesquisa para o desenvolvimento destes indicadores, pois seu uso permitiria alocação mais eficiente de esforços e recursos, bem como melhor utilização do potencial do software livre pela sociedade.

Homologação de Software Livre

É necessário estabelecer uma entidade que efetue a avaliação de pacotes de software livre e os homologue com relação a usabilidade, confiabilidade, segurança, documentação e conflitos de propriedade. Esta Entidade Homologadora publicaria os resultados de suas avaliações, de forma a que a informação quanto a qualidade geral dos pacotes/programas seja amplamente disponível. As funções da Entidade Homologadora, incluem as listadas abaixo.

- Repositórios de código homologado para acesso via Internet;
- repositórios de documentação e/ou FAQs⁴;
- mecanismo de assistência técnica estilo 0800;
- serviço de busca de soluções (existe pacote que resolva um certo problema?); e
- formação, capacitação e treinamento.

Quanto à organização e ao financiamento da entidade, ela poderia estar associada à Agência Facilitadora, ou preferencialmente poderia estar associada a órgãos não-governamentais, universidades ou institutos de pesquisa. Pela própria natureza do software livre, é importante que a Agência Facilitadora e a Entidade Homologadora sejam entidades de caráter público, para garantir que sejam mantidas as liberdades do software.

⁴*Frequently Asked Questions.*

Agradecimentos

Este trabalho foi parcialmente financiado pelo Programa das Nações Unidas para o Desenvolvimento, Projeto PNUD BRA/99/021, contrato 2001/003574, em consultoria contratada pelo Programa Sociedade da Informação–Brasil, do Ministério da Ciência e Tecnologia.

Várias pessoas colaboraram com o autor na preparação deste documento. O autor agradece a Eduardo Tadao Takahashi, Gorgonio Araújo, Hans Liesenberg, Hélio Pedrini, Marcos Castilho, Marcos Sunyé, Pablo J Madril, Rubens Queiroz de Almeida, Wagner Zola, e especialmente a Renato J S Carmo. Quaisquer erros ou imprecisões são de responsabilidade do autor.

Apêndice A

Licenças de Software Livre

Este apêndice contém vários exemplos de licenças empregadas na publicação de software livre. Sempre que possível, o texto contém a versão em Português da respectiva licença. Os textos das Seções A.1 e A.2 são versões das páginas disponíveis em www.fsf.org e foram editadas para remover material que não pertence às licenças propriamente ditas.

A.1 O Projeto GNU e a Fundação para o Software Livre

Segundo a Free Software Foundation *Software Livre* é uma questão de liberdade, não de preço. Para entender o conceito, você deve pensar em *liberdade de expressão*, não em *cerveja grátis*¹.

Software livre se refere à liberdade dos usuários executarem, copiarem, distribuírem, estudarem, modificarem e aperfeiçoarem o software. Mais precisamente, ele se refere a quatro tipos de liberdade, para os usuários do software:

- A liberdade de executar o programa, para qualquer propósito (liberdade no. 0).
- A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades (liberdade no. 1). Acesso ao código-fonte é um pré-requisito para esta liberdade.
- A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade no. 2).
- A liberdade de aperfeiçoar o programa, e publicar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade no. 3). Acesso ao código-fonte é um pré-requisito para esta liberdade.

Um programa é software livre se os usuários tem todas essas liberdades. Portanto, você deve ser livre para redistribuir cópias, seja com ou sem modificações, seja de graça ou cobrando uma taxa pela distribuição, para qualquer um em qualquer lugar. Ser livre para fazer essas coisas significa (entre outras coisas) que você não tem que pedir ou pagar pela permissão.

¹Em inglês: *free software* versus *free beer*.

Você deve também ter a liberdade de fazer modificações e usa-las privativamente no seu trabalho ou lazer, sem nem mesmo mencionar que elas existem. Se você publica as modificações, você não deve ser obrigado a avisar a ninguém em particular, ou de nenhum modo em especial.

Para que a liberdade de fazer modificações, e de publicar versões aperfeiçoadas, seja significativa, você deve ter acesso ao código-fonte do programa. Portanto, acesso ao código-fonte é uma condição necessária ao software livre.

Você pode ter pago em dinheiro para obter cópias do software GNU, ou você pode ter obtido cópias sem custo nenhum. Independentemente de como você obteve a sua cópia, você sempre tem a liberdade de copiar e modificar o software.

Para que essas liberdades sejam reais, elas tem que ser irrevogáveis desde que você não faça nada errado; caso o desenvolvedor do software tenha o poder de revogar a licença, mesmo que você não tenha dado motivo, o software não é livre.

Entretanto, certos tipos de regras sobre a maneira de distribuir software livre são aceitáveis, quando elas não entram em conflito com as liberdades principais. Por exemplo, *copyleft* (apresentado de forma bem simples) é a regra de que, quando redistribuindo um programa, você não pode adicionar restrições para negar para outras pessoas as liberdades principais. Esta regra não entra em conflito com as liberdades; na verdade, ela as protege.

Regras sobre como empacotar uma versão modificada são aceitáveis, se elas não acabam bloqueando a sua liberdade de liberar versões modificadas. Regras como “se você tornou o programa disponível deste modo, você também tem que torna-lo disponível deste outro modo” também podem ser aceitas, da mesma forma. Note que tal regra ainda deixa para você a escolha de tornar o programa disponível ou não.

No projeto GNU, nós usamos *copyleft* para proteger estas liberdades legalmente para todos. Também existe software livre que não é *copyleft*. Nós acreditamos que hajam razões importantes pelas quais é melhor usar o *copyleft*, mas se o seu programa é software-livre mas não é *copyleft*, nós ainda podemos utiliza-lo.

Às vezes regras de controle de exportação e sanções de comércio podem limitar a sua liberdade de distribuir cópias de programas inter-nacionalmente. Desenvolvedores de software não tem o poder para eliminar ou sobrepor estas restrições, mas o que eles podem e devem fazer é se recusar a impô-las como condições para o uso dos seus programas. Deste modo, as restrições não afetam as atividades e as pessoas fora da jurisdição destes governos.

Quando falando sobre o software livre, é melhor evitar o uso de termos como “dado” ou “de graça”, porque estes termos implicam que a questão é de preço, não de liberdade. Alguns termos comuns como “pirataria” englobam opiniões que nós esperamos você não endossaria.

Outro grupo (www.opensource.org) iniciou o uso do termo software aberto para significar algo próximo (mas não idêntico) a “software livre”.

Copyright © 1996, 1997, 1998, 1999, 2000 Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA.

A cópia fiel e a distribuição deste artigo completo é permitida em qualquer meio, desde que esta nota seja preservada.

21 Jan 2001 fsl

Traduzido por: Fernando Lozano fsl@centroin.com.br

Texto original disponível em www.fsf.org/philosophy/free-sw.pt.html

A.2 GPL – Licença Pública Geral GNU

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

This is an unofficial translation of the GNU General Public License into Portuguese. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL – only the original English text of the GNU GPL does that. However, we hope that this translation will help Portuguese speakers understand the GNU GPL better.

Copyright © 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA

É permitido a qualquer pessoa copiar e distribuir cópias tal desse documento de licença, sem a implementação de qualquer mudança.

A.2.1 Preâmbulo

As licenças de muitos softwares são desenvolvidas para cercear a liberdade de uso, compartilhamento e mudanças. A Licença Pública Geral GNU ao contrário, pretende garantir a liberdade de compartilhar e alterar softwares de livre distribuição - tornando-os de livre distribuição também para quaisquer usuários. A Licença Pública Geral aplica-se à maioria dos softwares da Free Software Foundation e a qualquer autor que esteja de acordo de utilizá-la (alguns softwares da FSF são cobertos pela GNU Library General Public License).

Quando nos referimos a softwares de livre distribuição, referimo-nos à liberdade e não ao preço. Nossa Licença Pública Geral foi criada para garantir a liberdade de distribuição de cópias de softwares de livre distribuição (e cobrar por isso caso seja do interesse do distribuidor), o qual recebeu os códigos fonte, o qual pode ser alterado ou utilizado em parte em novos programas.

Para assegurar os direitos dos desenvolvedores, algumas restrições são feitas, proibindo a todas as pessoas a negação desses direitos ou a solicitação de sua abdicação. Essas restrições aplicam-se ainda a certas responsabilidades sobre a distribuição ou modificação do software.

Por exemplo, ao se distribuir cópias de determinado programa, por uma taxa determinada ou gratuitamente, deve-se informar sobre todos os direitos incidentes sobre esse programa, assegurando-se que os fontes estejam disponíveis assim como a Licença Pública Geral GNU.

A proteção dos direitos envolve dois passos: (1) *copyright* do software e (2) licença que dá permissão legal para cópia, distribuição e/ou modificação do softwares. Ainda para a proteção da FSF e do autor é importante que todos entendam que não há garantias para softwares de livre distribuição. Caso o software seja modificado por alguém e passado adiante, este software não mais refletirá o trabalho original do autor não podendo portanto ser garantido por aquele.

Finalmente, qualquer programa de livre distribuição é constantemente ameaçado pelas

patentes de softwares. Buscamos evitar o perigo de que distribuidores destes programas obtenham patentes individuais, tornado-se seus donos efetivos. Para evitar isso foram feitas declarações expressas de que qualquer solicitação de patente deve ser feita permitindo o uso por qualquer indivíduo, sem a necessidade de licença de uso.

Os termos e condições precisas para cópia, distribuição e modificação seguem abaixo:

A.2.2 Licença Pública Geral GNU

TERMOS E CONDIÇÕES PARA CÓPIA, DISTRIBUIÇÃO E MODIFICAÇÃO

0 Esta licença se aplica a qualquer programa ou outro trabalho que contenha um aviso colocado pelo detentor dos direitos autorais dizendo que aquele poderá ser distribuído nas condições da Licença Pública Geral. O Programa, abaixo refere-se a qualquer software ou trabalho e a um trabalho baseado em um Programa e significa tanto o Programa em si como quaisquer trabalhos derivados de acordo com a lei de direitos autorais, o que significa dizer, um trabalho que contenha o Programa ou uma parte deste, na sua forma original ou com modificações ou traduzido para uma outra língua (tradução está incluída sem limitações no termo modificação).

Atividades distintas de cópia, distribuição e modificação não estão cobertas por esta Licença, estando fora de seu escopo. O ato de executar o Programa não está restringido e a saída do Programa é coberta somente caso seu conteúdo contenha trabalhos baseados no Programa (independentemente de terem sido gerados pela execução do Programa). Se isso é verdadeiro depende das funções executadas pelo Programa.

1 O código fonte do Programa, da forma como foi recebido, pode ser copiado e distribuído, em qualquer media, desde que seja providenciada um aviso adequado sobre os *copyrights* e a negação de garantias, e todos os avisos que se referem à Licença Pública Geral e à ausência de garantias estejam inalterados e que qualquer produtos oriundo do Programa esteja acompanhado desta Licença Pública Geral.

É permitida a cobrança de taxas pelo ato físico de transferência ou gravação de cópias, e podem ser dadas garantias e suporte em troca da cobrança de valores.

2 Pode-se modificar a cópia ou cópias do Programa de qualquer forma que se deseje, ou ainda criar-se um trabalho baseado no Programa, e copiá-la e distribuir tais modificações sob os termos do Parágrafo 1 acima e do seguinte:

- a) Deve existir aviso em destaque de que os dados originais foram alterados nos arquivos e as datas das mudanças;
- b) Deve existir aviso de que o trabalho distribuído ou publicado é, de forma total ou em parte derivado do Programa ou de alguma parte sua, e que pode ser licenciado totalmente sem custos para terceiros sob os termos desta Licença.
- c) Caso o programa modificado seja executado de forma interativa, é obrigatório, no início de sua execução, apresentar a informação de *copyright* e da ausência de garantias (ou de que a garantia corre por conta de terceiros), e que os usuários podem

redistribuir o programa sob estas condições, indicando ao usuário como acessar esta Licença na sua íntegra.

Esses requisitos aplicam-se a trabalhos de modificação em geral. Caso algumas seções identificáveis não sejam derivadas do Programa, e podem ser consideradas como partes independentes, então esta Licença e seus Termos não se aplicam àquelas seções quando distribuídas separadamente. Porém ao distribuir aquelas seções como parte de um trabalho baseado no Programa, a distribuição como um todo deve conter os termos desta Licença, cujas permissões estendem-se ao trabalho como um todo, e não a cada uma das partes independentemente de quem os tenha desenvolvido.

Mais do que tencionar contestar os direitos sobre o trabalho desenvolvido por alguém, esta seção objetiva propiciar a correta distribuição de trabalhos derivados do Programa. Adicionalmente, a mera adição de outro trabalho ao Programa, porém não baseado nele nem a um trabalho baseado nele, a um volume de armazenamento ou media de distribuição não obriga a utilização desta Licença e de seus termos ao trabalho.

3 São permitidas a cópia e a distribuição do Programa (ou a um trabalho baseado neste) na forma de código objeto ou executável de acordo com os termos dos Parágrafos 1 e 2 acima, desde que atendido o seguinte:

- a Esteja acompanhado dos códigos fonte legíveis, os quais devem ser distribuídos na forma dos Parágrafos 1 e 2 acima, em mídia normalmente utilizada para manuseio de softwares ou
- b Esteja acompanhado de oferta escrita, válida por, no mínimo 3 anos, de disponibilizar a terceiros, por um custo não superior ao custo do meio físico de armazenamento, uma cópia completa dos códigos fonte em meio magnético, de acordo com os Parágrafos 1 e 2 acima.
- c Esteja acompanhada com a mesma informação recebida em relação à oferta da distribuição do código fonte correspondente. (esta alternativa somente é permitida para distribuições não comerciais e somente se o programa recebido na forma de objeto ou executável tenha tal oferta, de acordo com a sub-seção b acima).

O código fonte de um trabalho é a melhor forma de produzirem-se alterações naquele trabalho. Códigos fontes completos significam todos os fontes de todos os módulos, além das definições de interfaces associadas, arquivos, scripts utilizados na compilação e instalação do executável. Como uma exceção excepcional, o código fonte distribuído poderá não incluir alguns componentes que não se encontrem em seu escopo, tais como compilador, kernel, etc... para o SO onde o trabalho seja executado.

Caso a distribuição do executável ou objeto seja feita através de acesso a um determinado ponto, então oferta equivalente de acesso deve ser feita aos códigos fonte, mesmo que terceiros não sejam obrigados a copiarem os fontes juntos com os objetos simultaneamente.

4 Não é permitida a cópia, modificação, sublicenciamento ou distribuição do Programa, exceto sob as condições expressas nesta Licença. Qualquer tentativa de cópia, modificação,

sublicenciamento ou distribuição do Programa é proibida, e os direitos descritos nesta Licença cessarão imediatamente. Terceiros que tenham recebido cópias ou direitos na forma desta Licença não terão seus direitos cessados desde que permaneçam dentro das cláusulas desta Licença.

5 Não é necessária aceitação formal desta Licença, apesar de que não haverá documento ou contrato que garanta permissão de modificação ou distribuição do Programa ou seus trabalhos derivados. Essas ações são proibidas por lei, caso não se aceitem as condições desta Licença. A modificação ou distribuição do Programa ou qualquer trabalho baseado neste implica na aceitação desta Licença e de todos os termos desta para cópia, distribuição ou modificação do Programa ou trabalhos baseados neste.

6 Cada vez que o Programa seja distribuído (ou qualquer trabalho baseado neste), o recipiente automaticamente recebe uma licença do detentor original dos direitos de cópia, distribuição ou modificação do Programa objeto deste termos e condições. Não podem ser impostas outras restrições nos recipientes.

7 No caso de decisões judiciais ou alegações de uso indevido de patentes ou direitos autorais, restrições sejam impostas que contradigam esta Licença, estes não isentam da sua aplicação. Caso não seja possível distribuir o Programa de forma a garantir simultaneamente as obrigações desta Licença e outras que sejam necessárias, então o Programa não poderá ser distribuído.

Caso este Parágrafo seja considerado inválido por qualquer motivo particular ou geral, o seu resultado implicará na invalidação geral desta licença na cópia, modificação, sublicenciamento ou distribuição do Programa ou trabalhos baseados neste.

O propósito deste parágrafo não é, de forma alguma, incitar quem quer que seja a infringir direitos reclamados em questões válidas e procedentes, e sim proteger as premissas do sistema de livre distribuição de software. Muitas pessoas têm feito contribuições generosas ao sistema, na forma de programas, e é necessário garantir a consistência e credibilidade do sistema, cabendo a estes e não a terceiros decidirem a forma de distribuição dos softwares.

Este Parágrafo pretende tornar claro os motivos que geraram as demais cláusulas destas Licença.

8 Caso a distribuição do Programa dentro dos termos desta Licença tenha restrições em algum País, quer por patentes ou direitos autorais, o detentor original dos direitos autorais do Programa sob esta Licença pode adicionar explicitamente limitações geográficas de distribuição, excluindo aqueles Países, fazendo com que a distribuição somente seja possível nos Países não excluídos.

9 A Fundação de Software de Livre Distribuição (FSF - Free Software Foundation) pode publicar versões revisadas ou novas versões desta Licença Pública Geral de tempos em tempos. Estas novas versões manterão os mesmos objetivos e o espírito da presente versão, podendo variar em detalhes referentes a novas situações encontradas.

A cada versão é dada um número distinto. Caso o Programa especifique um número de versão específico desta Licença a qual tenha em seu conteúdo a expressão - ou versão mais atualizada-, é possível optar pelas condições daquela versão ou de qualquer versão mais atualizada publicada pela FSF.

10 Caso se deseje incorporar parte do Programa em outros programas de livre distribuição de softwares é necessária autorização formal do autor. Para softwares que a FSF detenha os direitos autorais, podem ser abertas exceções desde que mantido o espírito e objetivos originais desta Licença.

AUSÊNCIA DE GARANTIAS

11 UMA VEZ QUE O PROGRAMA É LICENCIADO SEM ÔNUS, NÃO HÁ QUALQUER GARANTIA PARA O PROGRAMA. EXCETO QUANDO TERCEIROS EXPRESSEM-SE FORMALMENTE O PROGRAMA É DISPONIBILIZADO EM SEU FORMATO ORIGINAL, SEM GARANTIAS DE QUALQUER NATUREZA, EXPRESAS OU IMPLÍCITAS, INCLUINDO MAS NÃO LIMITADAS, A GARANTIAS COMERCIAIS E DO ATENDIMENTO DE DETERMINADO FIM. A QUALIDADE E DESEMPENHO SÃO DE RISCO EXCLUSIVO DOS USUÁRIOS, CORRENDO POR SUA CONTA OS CUSTOS NECESSÁRIOS A EVENTUAIS ALTERAÇÕES, CORREÇÕES E REPAROS JULGADOS NECESSÁRIOS.

12 EM NENHUMA OCASIÃO, A MENOS QUE REQUERIDO POR DECISÃO JUDICIAL OU POR LIVRE VONTADE, O AUTOR OU TERCEIROS QUE TENHAM MODIFICADO O PROGRAMA, SERÃO RESPONSÁVEIS POR DANOS OU PREJUÍZOS PROVENIENTES DO USO OU DA FALTA DE HABILIDADE NA SUA UTILIZAÇÃO (INCLUINDO MAS NÃO LIMITADA A PERDA DE DADOS OU DADOS ERRÔNEOS), MESMO QUE TENHA SIDO EMITIDO AVISO DE POSSÍVEIS ERROS OU DANOS.

FIM DA LICENÇA

Como aplicar estes termos a novos softwares?

Caso se tenha desenvolvido um novo programa e se deseje a sua ampla distribuição para o público, a melhor forma de consegui-lo é torná-lo um software de livre distribuição, o qual qualquer um possa distribuí-lo nas condições desta Licença.

Para tanto basta anexar este aviso ao programa. É aconselhável indicar ainda no início de cada arquivo fonte a ausência de garantias e um apontamento para um arquivo contendo o texto geral desta Licença, como por exemplo:

<nome do programa e função> Copyright (C) 199X <Autor>

Este programa é um software de livre distribuição, que pode ser copiado e distribuído sob os termos da Licença Pública Geral GNU, conforme publicada

pela Free Software Foundation, versão 2 da licença ou (a critério do autor) qualquer versão posterior.

Este programa é distribuído na expectativa de ser útil aos seus usuários, porém NÃO TEM NENHUMA GARANTIA, EXPLÍCITAS OU IMPLÍCITAS, COMERCIAIS OU DE ATENDIMENTO A UMA DETERMINADA FINALIDADE. Consulte a Licença Pública Geral GNU para maiores detalhes.

Deve haver uma cópia da Licença Pública Geral GNU junto com este software em inglês ou português. Caso não haja escreva para Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Autor@mail.com.br

Endereço

Caso o programa seja interativo, apresente na sua saída um breve aviso quando de seu início como por exemplo:

Gnomovision versão 69, Copyright ©199X Yoyodine

Gnomovision NÃO POSSUI NENHUMA GARANTIA; para detalhes digite mostre garantia. Este é um software de livre distribuição e você está autorizado a distribuí-lo dentro de certas condições. Digite mostre condição para maiores detalhes.

Os comandos hipotéticos `mostre garantia` e `mostre condição` apresentarão as partes apropriadas da Licença Pública Geral GNU. Evidentemente os comandos podem variar ou serem acionados por outras interfaces como clique de mouse, etc...

A.3 Open Source Initiative

A versão original² desta licença encontra-se em www.opensource.org/docs/definition_plain.html. Esta definição é muito similar à Debian Free Software Guidelines (DFSG), contida nas distribuições Debian do sistema GNU/Linux³.

INÍCIO DA LICENÇA

Além do código fonte, os termos de distribuição do *software com código aberto* (*Open Source Software*) devem obedecer aos seguintes critérios:

- 1. Redistribuição Livre** A licença não pode restringir a venda ou a cessão do software como um componente de uma distribuição, ou pacote, contendo programas de várias origens distintas. A licença não pode exigir o pagamento de *royalties*, ou em moeda, neste tipo de venda.
- 2. Código Fonte** O programa deve incluir o código fonte, e a licença deve permitir a distribuição do código fonte, bem como do código compilado (binário). Quando alguma forma de um produto não é distribuída com o código fonte, deve existir um meio bem-conhecido de se obter o código

²A tradução da licença foi efetuada por Roberto A Hexsel.

³Disponível em www.debian.org/social_contract

fonte, a custo não maior que o da reprodução, mas preferencialmente sem custo e através da Internet. O código fonte deve ser a forma preferencial a partir da qual um programador modificaria o programa. Código escrito deliberadamente de forma confusa não é permitido. Formas intermediárias, tais como a saída de um preprocessor ou tradutor, não são permitidas.

- 3. Trabalhos Derivados** A licença deve permitir modificações e a produção de trabalhos derivados, e deve permitir que estes sejam redistribuídos sob os mesmos termos da licença do software original.
- 4. Integridade do Código Fonte do Autor** A licença pode restringir a distribuição de código fonte modificado, *somente* se a licença permite a distribuição de “arquivos de diferenças” (*patch files*) juntamente com o código fonte, permitindo assim que o programa modificado seja produzido durante a sua construção. A licença deve explicitamente permitir a distribuição de software construído a partir do código fonte modificado. A licença pode exigir que as versões derivadas possuam um nome distinto do original ou um número de versão distinto daquele do software original.
- 5. Não-discriminação Contra Pessoas ou Grupos** A licença não pode discriminar contra qualquer pessoa ou grupo de pessoas.
- 6. Não-discriminação Contra Forma de Utilização** A licença não pode restringir a forma de utilização do programa. Por exemplo, a licença não pode restringir a utilização do programa em uma empresa, ou na pesquisa em Genética.
- 7. Distribuição da Licença** Os direitos associados a um programa devem se aplicar a todos a quem o programa é redistribuído sem necessidade da emissão de uma licença adicional a, ou por, aqueles.
- 8. Licença Não Pode Ser Específica a um Produto** Os direitos associados ao programa não podem depender de o programa pertencer a uma distribuição em particular. Se o programa é extraído da distribuição e usado ou distribuído de acordo com os termos da licença do programa, todos aqueles a quem o programa é redistribuído possuem os mesmos direitos outorgados quando da distribuição do software original.
- 9. Licença Não Pode Contaminar Outro Software** A licença não pode impor restrições a outros programas que são distribuídos juntamente com o software licenciado. Por exemplo, a licença não pode insistir em que todos os programas distribuídos em um mesmo meio físico sejam software de código aberto.

FIM DA LICENÇA

A.4 Licença BSD

Copyright © 2000 <SOMEONE>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.5 Licença X.Org

X.Org – Terms and Conditions

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1999,2000,2001 Compaq Computer Corporation

Copyright © 1999,2000,2001 Hewlett-Packard Company

Copyright © 1999,2000,2001 IBM Corporation

Copyright © 1999,2000,2001 Hummingbird Communications Ltd.

Copyright © 1999,2000,2001 Silicon Graphics, Inc.

Copyright © 1999,2000,2001 Sun Microsystems, Inc.

Copyright © 1999,2000,2001 The Open Group

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES

OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

X Window System is a trademark of The Open Group.

OSF/1, OSF/Motif and Motif are registered trademarks, and OSF, the OSF logo, LBX, X Window System, and Xinerama are trademarks of the Open Group. All other trademarks and registered trademarks mentioned herein are the property of their respective owners.

Apêndice B

Livros sobre Linux

Este Apêndice contém a lista de títulos de livros que contém a palavra Linux, obtida do catálogo eletrônico de uma grande livraria de Curitiba em 21 de julho de 2001. Alguns dos títulos constavam do catálogo como indisponíveis.

1. *Administração de Linux for Dummies*, Michael Bellomo, Ed. Campus
2. *Aprendendo Red Hat Linux*, Bill McCarthy, Ed. Campus
3. *Borland Kylix: Delphi para Linux*, Marcos dos Santos Ferreira, Ed. Érica
4. *Como Instalar Red Hat Linux 7*, William von Hagen, Ed. Campus
5. *Conectiva Linux 4 - Guia de Consulta e Aprendizado*, Gorki S C Oliveira, Ed. Book Express
6. *Configurando um Servidor Linux para Intranet/Internet*, Hide Tsuji, Ed. Makron Books
7. *Desvendando Linux*, Michael J Tobbler, Ed. Campus
8. *Dominando 110% Intranet em Ambiente Linux*, Antonio Marcelo, Ed. Brasport
9. *Dominando Linux*, Matt Welsh, Ed. Ciência Moderna
10. *Ferramentas Poderosas para Redes em Linux*, Paul G Sery, Ed. Ciência Moderna
11. *Firewalls em Linux*, Antonio Marcelo, Ed. Brasport
12. *Guia Completo do Linux*, Peter Norton, Ed. Berkeley Brasil
13. *Guia do Administrador de Redes Linux 2*, Olaf Kirch, Ed. Conectiva
14. *Guia do Administrador de Sistemas Linux 1*, Lars Wirzenius, Ed. Conectiva
15. *Guia do Servidor Linux 3*, diversos autores, Ed. Conectiva
16. *Guia do Usuário do Conectiva Linux 4.0*, diversos autores, Ed. Conectiva
17. *Internet em Ambiente Linux*, Antonio Marcelo, Ed. Brasport

18. *Introdução ao Linux - Como Instalar e Configurar*, Jairo Araújo, Ed. Ciência Moderna
19. *Introdução ao Linux Slackware 4.0*, Augusto B D'Oliveira, Ed. Ciência Moderna
20. *Kernel do Linux*, Scott Maxwell, Ed. Makron Books
21. *Kylix - A Ferramenta Linux para Programadores*, Adilson S Dias, Ed. Ciência Moderna
22. *Kylix Delphi para Linux - Guia Prático de Programação*, Bruno Sonnino.
23. *Linux - Guia Prático em Português*, Heverton Anunciação, Ed. Érica
24. *Linux - Manual do Usuário 3.0 Guarani*, MD9 Distrib. Informática
25. *Linux - O Guia Essencial*, Ellen Siever, Ed. Campus
26. *Linux - Rápido e Fácil para Iniciantes*, Lisa Lee, Ed. Campus
27. *Linux - Red Hat Conectiva Guia do Administrador e do Usuário*, Cesar A K Grosman, Ed. Book Express
28. *Linux - Série Curso Básico Rápido*, Carlos Cardoso, Axcel Books do Brasil Ed.
29. *Linux - Série para Dummies*, John Hall, Ed. Campus
30. *Linux 5 em 24 Horas Passo a Passo*, Bill Ball, Ed. Ciência Moderna
31. *Linux 5.0 - Estudo Dirigido*, José Augusto Manzano, Ed. Érica
32. *Linux Administração e Suporte*, Chuck V Tibet, Novatec Ed.
33. *Linux Bancos de Dados*, Fred Butzen, Ed. Ciência Moderna
34. *Linux Bash - Aprenda Rápido*, Carlos Portella, Visual Books Ed.
35. *Linux - Da Instalação às Redes*, Alexandre M Mattos, Livros e Editora Infobook
36. *Linux Device Drivers*, Alessandro Rubini, Ed. Market Books do Brasil
37. *Linux e os Hackers - Proteja Seu Sistema*, Wayne R Mendes, Ed. Ciência Moderna
38. *Linux e Seus Servidores*, João E Mota Filho, Ed. Ciência Moderna
39. *Linux - Ferramentas Anti-Hackers*, Antonio Marcelo, Ed. Brasport
40. *Linux - Interface Gráfica KDE, Guia de Consulta Rápida*, Frederico S Reis, Novatec Ed.
41. *Linux para Dummies*, Dee-Ann LeBlanc, Ed. Campus
42. *Linux para Leigos Passo a Passo*, Manuel A Ricart, Ed. Ciência Moderna
- ...
62. *Utilizando Samba & Flagship em Português para Linux*, Emiliano Soares Monteiro, Ed. Érica

Referências Bibliográficas

- [1] Nikolai Bezroukov. Open source software development as a special type of academic research (critique of vulgar raymondism). Technical report, First Monday, October 99. www.firstmonday.dk/issues/issue4_10/bezroukov/index.html.
- [2] M A Castilho, R J S Carmo, and R A Hexsel. Aparafusando parafusos: Um modelo de laboratório de computação com qualidade e otimização de recursos. In *Workshop de Educação em Informática*, pages 1–10. Sociedade Brasileira de Computação, August 2001.
- [3] Roberto Di Cosmo. CyberSnare. Technical report, Ecole Normale Supérieure de Paris, Março 1998. www.dmi.ens.fr/~dicosmo.
- [4] B J Dempsey, D Weiss, P Jones, and J Geenberg. A quantitative profile of a community of open source linux developers. Technical report, UNC Open Source Research Team, osrt@metalab.unc.edu, Outubro 1999. www.ibiblio.org/osrt/developro.html.
- [5] C DiBona, S Ockman, and M Stone. *Open Sources: Voices from the Open Source Revolution*. O'Reilly & Associates, 1999. ISBN 1-56592-582-3.
- [6] Jonathan Eunice. Beyond the cathedral, beyond the bazaar. Technical report, Maio 1998. www.illuminata.com/public/content/cathedral.
- [7] R A Hexsel, A E Urban, and R S M de Barros. Arquitetura do Sistema Cartão Nacional de Saúde. *Anais do VIII Congresso Brasileiro de Informática em Saúde*, Setembro 2002. www.inf.ufpr.br/~roberto/cnsCBIS02.pdf.
- [8] Forest J Cavalier III. Some implications of bazaar size. Technical report, Agosto 1998. www.mibsoftware.com/bazdev.
- [9] cWare Inc. The Linux Storm. Technical report, 2000. www.cwareco.com/linux_storm.html.
- [10] B F Leão, J Z da Silva, F B Toledo, and R E Cunha. Agente cartão: Integração de sistemas externos ao Sistema Cartão Nacional de Saúde. *Anais do VIII Congresso Brasileiro de Informática em Saúde*, Setembro 2002.
- [11] Josh Lerner and Jean Tirole. The simple economics of open source. Working paper 7600, National Bureau of Economic Research, Março 2000. www.nber.org/papers/w7600.

- [12] Hans Liesenberg and Daniel Weller. Software aberto no Brasil: indicadores e casos de uso. Technical report, Sociedade da Informação – Brasil, MCT, Março 2002. www.portalbrasil.org.br/Software_Aberto.
- [13] Nathan Newman. The origins and future of open source software. White paper, NetAction, 1999. www.netaction.org.
- [14] Office of the e-Envoy. Open source software use within UK Government. Technical report, e-Government Interoperability Framework – UK, www.e-envoy.gov.uk, Julho 2002. www.e-envoy.gov.uk/news/newonsite/oss-policy.htm.
- [15] Committee on Information Technology Literacy, Computer Science, Telecommunications Board, Commission on Physical Sciences, Mathematics, and Applications. *Being Fluent with Information Technology*. National Research Council, Washington D.C., 2000. [//books.nap.edu/books/030906399X/html/R1.html](http://books.nap.edu/books/030906399X/html/R1.html).
- [16] Working Group on Libre Software. *Free Software/Open Source: Information Society Opportunities for Europe?* Information Society Directorate General of the European Community, Abril 2000. Versão 1.2 (*work in progress*), eu.conecta.it/.
- [17] Working Group on Libre Software. *Training, promotion and explanation of benefits*. Information Society Directorate General of the European Community, 2000. eu.conecta.it/paper/Training_promotion_explanat.html.
- [18] Nic Peeling and Julian Satchell. Analysis of the impact of open source software. White Paper KI/SEB/CR010223, QinetiQ, Outubro 2001. www.govtalk.gov.uk/interoperability.
- [19] Larry L Peterson and Bruce S Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann, 2nd edition, 2000. ISBN 155860577-0.
- [20] Bryan Pfaffenberger. Linux in higher education: Open source, open minds, social justice. *Linux Journal*, 17, March 2000. www2.linuxjournal.com/articles/currents/017.html.
- [21] Eric S Raymond. The Cathedral and the Bazaar. Technical report, Maio 1997. www.tuxedo.org/~esr/writings/cathedral-bazaar/.
- [22] Eric S Raymond. Homesteading the Noosphere. Technical report, Abril 1998. www.tuxedo.org/~esr/writings/homesteading/.
- [23] Eric S Raymond. The Magic Cauldron. Technical report, Junho 1999. www.tuxedo.org/~esr/writings/magic-cauldron/.
- [24] Richard Stallman. *The GNU Manifesto*. Free Software Foundation, 1985. www.gnu.org/gnu/manifesto.html.
- [25] Danny Thorpe. *Is Linux Ready for Delphi*. Inprise-Borland, 2000. <http://community.borland.com/article/0,1410,21171,00.html>.
- [26] Ming-Wei Wu and Ying-Dar Lin. Open source software development: An overview. *IEEE Computer*, 34(6):33–38, June 2001.

Índice Remissivo

A

Agência facilitadora, 30
apache, 9, 11, 21
Apple,
 MacIntosh, 2
 MacOS, 17
Applix, 21
AT&T, 8
autor, 3, 10

B

Berkeley Software Distribution, *veja* BSD
bind, 21
Borland, 18, 21
BSD, 3, 5, 40
BSD-Unix, 8, 11, 20

C

C, 3, 11, 20
código,
 binário, 3
 executável, 2, 3
 fonte, 2, 3, 10
ciclo de vida, 10
Compaq, 18
Conectiva, 11, 13, 15, 20, 22
copyleft, 4, 33
copyright, 4
Corel, 21

D

Debian, 3, 5, 13, 22
 DFSG, 5, 39
Delphi, 18
DENATRAN, 25
distribuição, 3, 22
DNS, 21
Donald Knuth, 9

E

emacs, 8, 17, 21
Entidade Homologadora, 30
Eric S Raymond, 9

F

Free Software Foundation, *veja* FSF
FreeBSD, 8
FSF, 3, 8, 9, 21, 32, 34

G

gcc, 8, 9, 20, 21
gdb, 21
GNOME, 9, 16, 21, 22
GNU, 32, 34
 Projeto, 8
GNU/Linux, 2, 8, 15, 17, 22, 39
GPL, 5, 8, 9, 20, 34

H

hardware,
 drivers, 15
 obsolescência, 13
Hewlett-Packard, 11
HTTP, 11

I

IBAM, 27
IBM, 7, 11, 15, 21
IETF, 7, 29
implementação de referência, 7
Imposto de Renda, 23, 26, 27, 29
inchaço do software, 13
Infoworld, 14
Internet, 6, 7, 13, 15, 20
Internet Engineering Task Force, *veja* IETF

J

Java, 3

K

KDE, 9, 16, 21, 22
Kylix, 18, 21

L

Leslie Lamport, 9
liberdades do software, 32
licença, 2, 3, 5
Linus Torvalds, 8
Linux, 5, 8, 17
Lotus, 21

M

Mandrake, 22
Microsoft, 6, 17, 22–24
 Windows, 2, 18, 27
 Windows98, 17, 18
Mosaic, 11

N

NetBSD, 8
Netcraft, 21
Netscape, 6, 11, 17

O

Open Software Development Laboratory, 11
Open Source Initiative, *veja* OSI
OpenBSD, 8
OpenOffice, 16, 21, 22
Oracle, 21
OSDL, 11
OSI, 3, 5, 39

P

pacote, 3
peer review, 14
perl, 3, 9, 21
plataforma modelo, 14
Projeto GNU, 5

R

rau-tu, 20
Receita Federal, 26, 27
Red Hat, 11, 13
Request For Comments, *veja* RFC
RFC, 7
Richard Stallman, 8

S

Samba, 11
SAP, 21
sendmail, 17, 21
software bloat, 13
StarOffice, 6, 21, 22
Sun,
 Solaris, 2

T

TCP/IP, 20
The Cathedral and the Bazaar, 9
trabalho derivado, 3

U

Unix, 2, 8, 11, 16, 17

V

versão original, 3

W

W3C, 29
Windows, *veja* Microsoft

X

X Window System, 5, 42
X.org, 5, 41
XML, 29