

FACULDADE DE TECNOLOGIA DE GARÇA

ANÁLISE E DESENVOLVIMENTO DE SOFTWARE

SISTEMAS OPERACIONAIS II

O Que é Software Livre?

"Software livre" se refere à liberdade dos usuários executarem, copiarem, distribuírem, estudarem, modificarem e aperfeiçoarem o software. Mais precisamente, ele se refere a **quatro liberdades**, 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 possa ajudar ao seu próximo (liberdade nº 2);
- A liberdade de aperfeiçoar o programa, e liberar 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 o usuário tem todas estas liberdades. Portanto, 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 sem pedir ou pagar pela permissão.

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

A liberdade de utilizar um programa significa a liberdade para qualquer tipo de pessoa física ou jurídica utilizar o software em qualquer tipo de sistema computacional, para qualquer tipo de trabalho ou atividade, sem que seja necessário comunicar ao desenvolvedor ou a qualquer outra entidade em especial.

A liberdade de redistribuir cópias deve incluir formas binárias ou executáveis do programa, assim como o código-fonte, tanto para as versões originais quanto para as modificadas. Se não for possível produzir uma forma binária ou executável (algumas linguagens de programação não suportam este recurso), mas deve ser concedida a liberdade de redistribuir essas formas caso seja desenvolvido um meio de criá-las.

De modo que a liberdade de fazer modificações, e de publicar versões aperfeiçoadas, tenha algum significado, deve-se ter acesso ao código-fonte do programa. Portanto, **acesso ao código-fonte é uma condição necessária ao software livre.**

Para que essas **liberdades** sejam reais, elas têm que ser **irrevogáveis desde que não se faça nada errado**; caso o desenvolvedor do software tenha o poder de revogar a licença, mesmo que haja 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** é a regra de que, quando redistribuindo um programa, 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.

Portanto, pode-se pagar para receber cópias do software GNU, ou pode-se obter cópias sem nenhum custo. Mas independente de como obter a cópia, sempre tem a liberdade de copiar e modificar o software, ou mesmo de vender cópias.

"Software Livre" não significa "não-comercial". Um programa livre deve estar disponível para uso comercial, desenvolvimento comercial, e distribuição comercial. O desenvolvimento comercial de software livre não é incomum; tais softwares livres comerciais são muito importantes.

Software Livre é diferente de software em domínio público. O **primeiro**, quando utilizado em combinação com licenças típicas (como as licenças GPL e BSD), **garante os direitos autorais do programador/organização**. O segundo caso acontece quando o autor do software renuncia à propriedade do programa (e todos os direitos associados) e este se torna bem comum.

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 pode que torná-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.) Também é aceitável uma licença que exija que, caso você tenha distribuído uma versão modificada e um desenvolvedor anterior peça por uma cópia dele, você deva enviar uma.

No projeto GNU, usa-se o "**copyleft**" para proteger estas liberdades legalmente para todos. Mas também existe software livre que não é copyleft. Nós acreditamos que hajam razões importantes pelas quais é melhor usar o copyleft, mas se o programa é free-software, mas não é copyleft, todos podem utilizá-lo.

Às vezes regras de controle de exportação e sansões de comércio podem limitar a sua liberdade de distribuir cópias de programas internacionalmente. Desenvolvedores de software não tem o poder para eliminar ou sobrepôr 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**.

Finalmente, note que critérios como os estabelecidos nesta definição do software livre requerem cuidadosa deliberação quanto à sua interpretação. Para decidir se uma licença se qualifica como de software livre, nós a julgamos baseados nestes critérios para determinar se ela se segue o nosso espírito assim como as palavras exatas. Se uma licença inclui restrições impensadas, nós a rejeitamos, mesmo que nós não tenhamos antecipado a questão nestes critérios. Às vezes um requerimento de alguma licença levanta uma questão que requer excessiva deliberação, incluindo discussões com advogados, antes que nós possamos decidir se o requerimento é aceitável. Quando nós chegamos a uma conclusão sobre uma nova questão, nós frequentemente atualizamos estes critérios para tornar mais fácil determinar porque certas licenças se qualificam ou não.

Para maiores informações, visite o site de <http://www.gnu.org/philosophy/philosophy.pt-br.html>

"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".

Software Proprietário x Software Livre

Software Proprietário

É aquele cuja cópia, redistribuição ou modificação são proibidos pelo autor em determinado grau. É necessário solicitar permissão ou pagar para utilizar. Pode ser freeware, shareware, trial ou demo, etc.

O sistema de copyright atribui donos ao software, e esses donos, pela definição apresentada acima, têm o direito de estabelecer regras de distribuição do software. Devido a esse poder, surgiu o conceito de Software Proprietário, um paradigma de distribuição de software vigente desde o início da década de 80, que foi responsável pela criação do chamado Software de Caixinha. Porém, ao contrário do que muitos pensam, simplesmente existir o copyright de um software não basta para que ele seja Software Proprietário. O copyright, ou Direito de Uso, permite que o detentor desse direito estabeleça regras para o uso e distribuição do software. Essas regras são chamadas de licenças de uso e podem estabelecer, por exemplo, se o software pode ou não ser distribuído gratuitamente, se pode ou não ser copiado, entre outras regras.

Software Livre

Alguns tipos de licença:

GPL: a Licença Pública Geral GNU acompanha os pacotes distribuídos pelo Projeto GNU (General Public License). É a mais utilizada, sendo adotada pelo Linux. Ela impede que o software seja integrado em um software proprietário e garante os direitos autorais. Não permite que as liberdades originais sejam limitadas, nem que sejam impostas restrições que impeçam a distribuição da mesma forma que foram adquirido

BSD: a licença BSD foi inicialmente utilizada nos softwares da Berkeley Software Distribution. Ela impõe poucas restrições sobre as formas de uso, alterações e redistribuição do software e, por isso, é chamada de copycenter. O programa pode ser vendido e não precisa incluir o código fonte.

Software em Domínio Público: o autor do software relega a propriedade do programa e este se torna bem comum, ou seja, não possui copyright. Entretanto, o autor pode restringir que modificações sejam feitas.

Freeware: software proprietário que é disponibilizado gratuitamente, mas não pode ser modificado.

Shareware: é o software disponibilizado gratuitamente por um período de tempo ou com algumas funções abertas, mas que implica no posterior pagamento pela sua licença.

Trial: versão de teste de vários softwares. É disponibilizada algumas funções, geralmente por 30 dias, para que o usuário experimente o programa para saber se ele atende às suas necessidades.

Demo: versão de demonstração, semelhante ao Trial. É possível usar o programa por um tempo ou com apenas algumas funções disponíveis.

Open Source: o software de código aberto é aquele que disponibiliza seu código fonte e restringe-se aos termos técnicos da questão. Pode ser livre, ou proprietário. Algumas empresas como IBM, HP, Intel e Nokia investem em software de código aberto.

Histórico do Linux

A união do Laboratório Bell da AT&T, uniu-se a General Electric e o projeto MAC do MIT (Massachusetts Institute of Technology), era voltada para desenvolver um sistema operacional que veio a se chamar MULTICS.

Como é de se esperar o projeto do MULTICS, não atingiu seus propósitos e então os Laboratórios Bell saíram do projeto. Logo em seguida o cientista Ken Thompson da empresa AT&T e um ex-integrante do grupo que desenvolvia o MULTICS, começou a desenvolver um novo sistema.

Este sistema era escrito em Assembly num computador PDP-7. Este foi batizado de UNIX.

Uma das grandes dificuldades em se escrever em Assembly é que praticamente o sistema operacional só vai funcionar numa plataforma. Ou seja ficaria limitado a uma única plataforma. Mas uma vez foi feito um esforço para se construir uma linguagem de programação que possibilitasse qualquer programa ser portado de forma bem mais fácil para outras plataformas. Então criou-se a linguagem C.

O UNIX cresceu muito na AT&T. Nesta época a AT&T não comercializava o UNIX, porém o distribuía de forma gratuita juntamente com o seu código fonte para as universidades para fins educacionais.

O UNIX cresceu tanto que a AT&T, aproveitando este grande filão de mercado, começou a comercializá-lo. Empresas do mundo todo desenvolveram aplicações comerciais para o UNIX.

A origem do nome Unix, vem do MULTICS. Primeiro foi chamado de Unics. Depois virou realmente UNIX. Brian Kernighan, também pesquisador da Bell Labs, foi quem deu esse nome.

O UNIX foi desenvolvido no final da década de 1960 e vai seguindo seu percurso de crescimento.

Características do UNIX

1. Sistema multitarefa e multiusuário
2. Disponível para diversas plataformas.
3. Sistema robusto e confiável.
4. Sistema maduro.

A principal desvantagem do UNIX é por ser muito caro e só funcionar em plataformas de hardware com preço inacessível a maioria dos profissionais.

Entre 1977 e 1981, a AT&T, alterou o Unix, fazendo algumas mudanças particulares e lançou o System III. Em 1983, após mais uma série de modificações, foi lançado o conhecido Unix System IV, que passou a ser vendido. Até hoje esse sistema é usado no mercado, tornando-se o padrão internacional do Unix. Esse sistema é comercializado por empresas como IBM, HP, Sun, etc. O Unix, é um sistema operacional muito caro e é usado em computadores poderosos (como mainframes) por diversas multinacionais.

Versões do UNIX comerciais:

- » HP-UX
- » AIX
- » Solaris
- » IRIX

Apresentando o Linux

Como se percebe o UNIX é inacessível a maioria das pessoas, pois é caro e exige um hardware adequado.

O Dr. Andrew Tanenbaum motivado para fins educacionais, criou um sistema operacional chamado MINIX, baseado nos processadores INTEL 8086 que era na época a plataforma mais barata e atual.

O Minix só poderia ser usado para fins acadêmicos devido as suas limitações técnicas. Só endereçava até 1MB de cada vez. Não tinha memória virtual.

O que é GNU e a FSF?

A Fundação do Software Livre – FSF (Free Software Foundation) fundada por Richard Stallman no ano de 1983. Foi criado também o projeto GNU que tinha a responsabilidade de criar um clone do UNIX. Um detalhe deste clone é que tinha que ser livre e não poderia usar código fonte no UNIX pois tinha uma versão livre do UNIX – o BSD.

O projeto não deu certo em partes, pois chegou ao final da década de 80 sem cumprir seus propósitos que eram criar o clone do UNIX. Mas o fracasso não foi total.

Não se conseguiu fazer o clone do UNIX mas foram criadas as ferramentas necessárias para que Linux Torvalds conseguisse desenvolver o Linux.

Então, Linus Benedict Torvalds aluno da Universidade de Helsinque, na Finlândia usou tudo o que tinha até o momento: O Minix e as Ferramentas do GNU.

Linus, inicia o desenvolvimento de um kernel que fosse um clone do UNIX que possuísse memória virtual, multitarefa preemptiva e capacidade de multiusuários. Era um trabalho que ele sozinho não conseguiria terminar em tempo hábil.

Então, ele disponibilizou o código fonte do que ele veio a chamar de Linux.

O nome Linux surgiu da mistura de Linus + Unix. Linus é o nome do criador do Linux, Linus Torvalds.

Alguns linuxistas mais radicais chamam o sistema operacional do pingüim de GNU/LINUX, já que Linux Torvalds, conseguiu algum sucesso usando as ferramentas do projeto GNU.

Origem do nome Linux

O nome Linux surgiu da mistura de Linus + Unix. Linus é o nome do criador do Linux, Linus Torvalds. E Unix, é o nome de um sistema operacional de grande porte.

O nome do mascote Tux (o pingüim do Linux), foi escolhido por votação, pelo próprio Linus. Segundo ele "gostaria de um pingüim cheio, satisfeito por ter comido muitos peixes". Diz a lenda que o motivo por trás da escolha do Tux como mascote foi uma mordida que Linus levou de um pingüim em um zoológico.

Fonte:

<http://softwarelivre.org>

<http://www.gnu.org/>

<http://www.gnu.org/philosophy/philosophy.pt-br.html>

<http://www.softwarelivre.gov.br/>

<http://br-linux.org/>