

REVISTA

espírito livre

LIBERDADE E
INFORMAÇÃO

<http://revista.espiritolivre.org> | #050 | Maio 2013



Entrevista com Linus Torvalds

Página 36

Dicas para otimização do Ubuntu

Página 53

Virtualização com Vagrant

Página 56

Criando uma rede social na escola

Página 72



S O F T W A R E L I V R E N O

GOVERNO



Atribuição-Compartilhual 3.0 Brasil (CC BY-SA 3.0 BR)

Esta é uma licença simplificada baseada na [Licença Jurídica \(licença integral\)](#)

[Advertência](#)

Você tem a liberdade de:

Compartilhar — copiar, distribuir e transmitir a obra.

Remixar — criar obras derivadas.

fazer uso comercial da obra



Sob as seguintes condições:



Atribuição — Você deve creditar a obra da forma especificada pelo autor ou licenciante (mas não de maneira que sugira que estes concedem qualquer aval a você ou ao seu uso da obra).



Compartilhamento pela mesma licença — Se você alterar, transformar ou criar em cima desta obra, você poderá distribuir a obra resultante apenas sob a mesma licença, ou sob uma licença similar à presente.

Ficando claro que:

Renúncia — Qualquer das condições acima pode ser **renunciada** se você obtiver permissão do titular dos direitos autorais.

Domínio Público — Onde a obra ou qualquer de seus elementos estiver em **domínio público** sob o direito aplicável, esta condição não é, de maneira alguma, afetada pela licença.

Outros Direitos — Os seguintes direitos não são, de maneira alguma, afetados pela licença:

- Limitações e exceções aos direitos autorais ou quaisquer **usos livres** aplicáveis;
- Os **direitos morais** do autor;
- Direitos que outras pessoas podem ter sobre a obra ou sobre a utilização da obra, tais como **direitos de imagem** ou privacidade.


Aviso — Para qualquer reutilização ou distribuição, você deve deixar claro a terceiros os termos da licença a que se encontra submetida esta obra. A melhor maneira de fazer isso é com um link para esta página.

Uma mensagem para o leitor



Recentemente tivemos a notícia que alguns de nossos governantes, aqueles eleitos para nos representar e zelar por nós, se manifestaram contrários a tecnologia nacional, com a assinatura de acordos com empresas estrangeiras. Tal medida vem se tornando cada vez mais comum e pode até passar despercebida sob o olhar da sociedade e até dos "mais entendidos". Entretanto, a comunidade de software livre mostrou-se bastante decepcionada com tais medidas. Um acordo por si só, ainda sem apresentação para o povo de suas premissas e propósitos reais, de forma transparente, já é algo que gera medo, incerteza e dúvida. Sugere-se que em um acordo entre duas partes, ambas se beneficiarão igualmente, logo, acordos não são maléficos por natureza. O que causa estranheza é ele partir de estados tão popularmente conhecidos por suas inovações e projetos relacionados com o software livre. Isto causou o mal-estar que me refiro.

Nossos governos deveriam (e este é o papel deles), zelar por sua população, cuidar de seu

povo. Inovações nacionais e projetos que nasceram ou foram desenvolvidos aqui não poderiam simplesmente serem jogados para escanteio, em detrimento de acordos com multinacionais. Não cabe nem mencionar governos, nem as empresas envolvidas, uma vez que não são poucos tais acordos que vem pipocando aqui e ali. Nós da comunidade, vez ou outra, já nos sentimos deslocados quando fazemos uso de uma tecnologia que nem todos a utilizam. Não ter apoio nem dos que deveriam nos representar, ainda, é algo que entristece e nos faz refletir em até que ponto tal "cuidado e zelo" está realmente sendo realizado. Não é colocado em xeque aqui apenas as responsabilidades de nossos governantes, mas também sua preocupação com a inovação, genialidade nacional e independência tecnológica. Vale ressaltar que não basta termos uma massa de usuários interessados em mudar. É necessário vontade, e algumas vezes, política. Que tal pensarmos um pouco sobre isso? 

João Fernando Costa Júnior
Editor

Diretor Geral

João Fernando Costa Júnior

Editor

João Fernando Costa Júnior

Revisão

Vera Cavalcante e João Fernando Costa Júnior

Arte e Diagramação

Hélio José S. Ferreira e João Fernando Costa Júnior

Jornalista Responsável

Larissa Ventrone Costa - ES00867JP

Capa

Carlos Eduardo Mattos da Cruz e João Fernando Costa Júnior

Colaboradores desta edição

Albino Biasutti Neto, Aleksandro Montanha, Alexandre Oliva, Antônio Carlos Ribas, Camilo de Souza Teixeira, David Jourdain, Edson

Praczyk, Gilberto Sudré, Glyn Moody, Gustavo Freitas, Hailton David Lemos, João Felipe Soares Silva Neto, João Fernando Costa Júnior, João Victor Rocon Maia, José James Figueira Teixeira, José Roberto Colombo Júnior, Julian Carlo Fagotti, Linus Torvalds, Marcelo Soares Souza, Milton Simonetti, Nayan Pacheco, Roney Médice, Selma Regina Gomes e Wagner da Silva Arestides.

Contato

Site: <http://revista.espiritolivres.org>

Email: revista@espiritolivres.org

Telefone: +55 27 8112-4903

ISSN Nº 2236031X

O conteúdo assinado e as imagens que o integram, são de inteira responsabilidade de seus respectivos autores, não representando necessariamente a opinião da Revista Espírito Livre e de seus responsáveis. Todos os direitos sobre as imagens são reservados a seus respectivos proprietários.

03 EDITORIAL

por João Fernando Costa Júnior

05 NOTÍCIAS

por João Fernando Costa Júnior

08 CARTAS

por vários autores

09 PROMOÇÕES

por vários autores

**11 DEPUTADO QUESTIONA
PARCERIA ENTRE ESTADO E
MICROSOFT**

por Antônio Carlos Ribas

**13 ENTREVISTA COM JULIAN
CARLO FAGOTTI**

por João Fernando Costa Júnior

**17 SOFTWARE LIVRE EM XEQUE
MATE NO PARANA**

por Marcelo Soares Souza

20 SOFTWARE LIVRE NO LIMBO

por David Jourdain

**24 O QUE MOVE AS PESSOAS SÃO
OS SONHOS**

por Aleksandro Montanha

**26 O USO DO SOFTWARE LIVRE
NA ADMINISTRAÇÃO PÚBLICA**

por Roney Médice

**30 O USO DO SOFTWARE LIVRE
NO GOVERNO**

por Milton Simonetti

33 FORMATOS LIVRES

por Gilberto Sudré

**36 ENTREVISTA COM LINUS
TORVALDS**

por Glyn Moody

44 POR QUE PERSONALIZAR?

por Camilo de Souza Teixeira

**47 REDE SOCIAL E VOCÊ NO
FACEBOOK**

por Albino Biasutti Neto

49 SALVEM OS HACKERS

por Alexandre Oliva

**53 DICAS PARA OTIMIZAÇÃO DO
UBUNTU**

por Wagner da Silva Arestides

**56 VIRTUALIZAÇÃO DE DESKTOPS
COM VAGRANT**

por João Victor Rocon

**59 AMBIENTE DE PROGRAMAÇÃO
PARA AVR**

por José Roberto Colombo Júnior

**64 O SOFTWARE LIVRE COMO
ALIADO NO COMBATE A
FALSIFICAÇÃO**

por Hailton David Lemos

**68 APROXIMAÇÕES ENTRE A
PEDAGOGIA E O MOVIMENTO DO
SOFTWARE LIVRE**

por Selma Regina Gomes

**72 COMO CRIAR UMA REDE
SOCIAL NA ESCOLA**

por Gustavo Freitas

77 QUADRINHOS

por vários autores

79 AGENDA

por vários autores

Notícias

por João Fernando Costa Júnior

Lançado Debian GNU/Hurd 2013



A equipe de desenvolvimento do GNU/Hurd no projeto Debian lançou uma versão da distribuição baseada no Debian 7 “Wheezy” e o kernel GNU Hurd. Debian GNU/Hurd 2013 é o primeiro grande lançamento da distribuição depois de anos de trabalho de desenvolvimento, mesmo que ele não sendo um lançamento oficial do Debian. O kernel Hurd é um projeto micro-kernel Unix-like baseado no kernel Mach e está em desenvolvimento desde 1990. Debian GNU/Hurd está disponível para a arquitetura i386 e os desenvolvedores estão dizendo que mais de 10 mil pacotes estão disponíveis nos repositórios, que dizem englobar mais de 75% do repositório Debian inteiro. A distribuição usa a versão 1.3.99 do kernel do GNU Hurd que os desenvolvedores dizem ter atingido “um estado muito decente” dados os recursos limitados do projeto, o kernel é complementado pelas habituais ferramentas e programas do Debian. O sistema é instalado com o familiar instalador gráfico do Debian. Debian GNU/Hurd 2013 está disponível para download como um CD, DVD e NetInstall a partir dos [servidores](#) de ports do Debian.

Lançado Diaspora 0.1.0.0



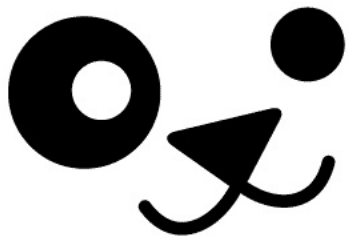
A plataforma social federada de código aberto Diaspora acaba de disponibilizar mais um novo lançamento para o público. Após mover o projeto para o modelo de desenvolvimento dentro da comunidade em Agosto de 2012, os desenvolvedores agora conseguiram lançar a versão 0.1.0.0 do projeto. Essa nova versão inclui funcionalidades como upload de fotos via dispositivos móveis, uma função de preview de posts e navegação via teclado, assim como, um número razoável de correções de bugs. Detalhes no [changelog](#) do projeto.

Lançado Moodle 2.5



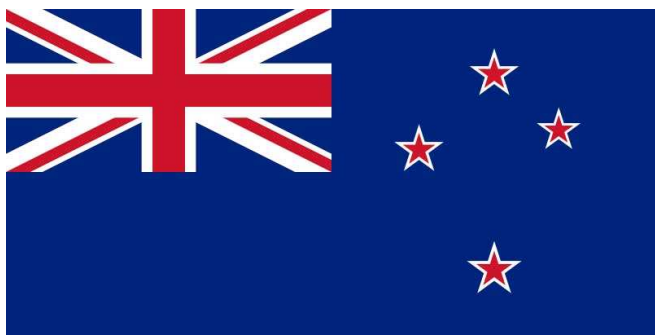
Martin Dougianias, cofundador do Moodle, anunciou o lançamento do Moodle 2.5 após longos meses de trabalho de desenvolvimento. Entre as principais novidades destaca-se o suporte para o OpenBadges da Mozilla, o início de suporte para temas projetados com o framework Bootstrap do Twitter, além de melhorias de usabilidade e acessibilidade. Além do lançamento do Moodle 2.5, houve também atualizações para as séries estáveis 2.2.x, 2.3.x e 2.4.x. Tais informações podem ser encontradas nas [notas de lançamento](#).

Lançado Puppy Linux 5.6



Barry Kauler, desenvolvedor do Puppy Linux, lançou a última versão de sua distribuição GNU/Linux. O Precise Puppy 5.6, assim como o nome sugere, é baseado no Ubuntu 12.04 LTS de codinome “Precise Pangolin”, e inclui um novo setup wizard para o X.org, um kernel não-PAE baseado na versão upstream 3.2.44 do kernel Linux, além de um número de drivers para modems analógicos destinados a usuários que ainda não possuem acesso a conexões com a Internet via banda larga. Uma lista detalhada de suas novas funcionalidades está disponível nas [notas de lançamento](#).

Em favor da inovação no mundo da tecnologia, governo da Nova Zelândia eliminou as patentes de software



O governo da Nova Zelândia anunciou que já não se poderão solicitar mais patentes de software no país para eliminar uma das barreiras que estão detendo a inovação no mundo da tecnologia. Ainda que tecnicamente nunca fossem explicitamente permitidas as patentes de software, a emenda introduzida na lei de patentes vigente vai ainda mais além, ao definir-se basicamente em três princípios - novidade, inventividade e aplicação industrial - que “um programa de computador não é uma

invenção e nem tampouco uma forma de fabricação para os propósitos desta ata”. Com esta decisão, o governo neo-zelandês considera legalmente que as leis atuais de patentes já outorgam proteção suficiente ao software, pois é cada vez mais comum um estancamento da inovação causado pelo patenteamento massivo de supostas inovações que na verdade não são. A decisão poderia ser o argumento, contrariamente ao que os EEUU consideram, para excluir totalmente o software das leis de patentes de qualquer país.

Nuvem do Serpro será 100% em código aberto



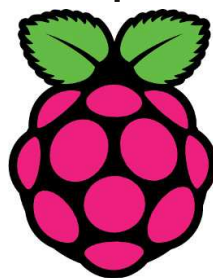
Em agosto, o Serpro planeja concluir o desenvolvimento de sua solução definitiva para processamento em nuvem, migrando os atuais ambientes que dispõe, inclusive proprietários, por força de contratos com a Receita Federal, para uma solução exclusiva em código aberto. O trabalho vem sendo desenvolvido pela regional do Serpro do Paraná, através da plataforma “openstack”, segundo revelou à CDTV do portal Convergência Digital o presidente do Serpro, Marcos Mazoni. “Em agosto nós vamos migrar esses serviços que já implantamos na nossa nuvem, que são ou em soluções proprietárias ou gratuitas, mas não absolutamente livres, passando para o “openstack”, afirmou. Segundo Mazoni, com a nova plataforma, alguns dos principais clientes do Serpro, entre eles o Ministério do Planejamento, Orçamento e Gestão, poderão integrar melhor no ambiente em nuvem, hoje restrito ao seu pessoal da área de desenvolvimento.

Publicações digitais da Unesco serão totalmente livres



A Unesco decidiu tornar suas publicações digitais disponíveis gratuitamente para milhões de pessoas ao redor do mundo, sob licença aberta. Seguindo uma decisão da equipe executiva da Organização, em abril, a Unesco passou a ser o primeiro braço das Nações Unidas a adotar esse tipo de política — que permite a qualquer pessoa baixar, traduzir, adaptar distribuir e compartilhar os conteúdos do banco de publicações e informações, sem qualquer pagamento. Ao adotar esta nova política, a UNESCO alinha a sua prática ao seu trabalho de defesa do acesso aberto e reforça o seu compromisso com o acesso universal à informação e ao conhecimento. O movimento de Acesso Aberto (OA) nasceu na comunidade científica para enfrentar os custos crescentes da literatura científica, que é essencial para os pesquisadores. Uma vasta gama de universidades, instituições e governos o apoiam como uma alternativa ao modelo tradicional de disseminação de conhecimento por meio de publicações acadêmicas caras. A partir de julho de 2013, centenas de publicações digitais da Unesco estarão disponíveis para os usuários em um novo repositório de acesso aberto com uma interface multilíngue. Todas as novas publicações serão lançadas com licença aberta. A Unesco também vai buscar formas de aplicar a decisão de forma retroativa, ou seja, para obras já publicadas. Coeditores serão fortemente incentivados a aderir aos requisitos da nova política.

Raspberry Pi: módulo de câmera já está disponível




RaspberryPi

Em desenvolvimento desde o ano passado, o módulo da câmera Raspberry Pi está finalmente disponível. O módulo pode ser comprado da RS Components ou Element 14 e é baseado em um sensor Omnivision 5647 de 5MPixel que é configurado para oferecer uma resolução de imagem de 2592x1944 ou ainda entregar vídeo com resolução de 1080p a 30fps (frames por segundo). Isso tudo é gerenciado em um pacote de 20x25x10mm que se conecta ao Pi através de um cabo de fita plana. O suporte de software está incluso na distribuição Raspbian do Debian para o Pi. A placa custa £19,24 no Reino Unido, incluso VAT da RS Components.

Revista Espírito Livre fará participação especial no fisl14



O fisl14 promoverá uma atividade especial: o ativista e editor João Fernando Costa Júnior, da revista Espírito Livre, promoverá uma oficina aberta de diagramação com o uso de softwares livres, a partir da qual o material apurado no evento, entre textos e fotos, será usado na produção de uma revista do Fórum, creditada aos colaboradores responsáveis pela cobertura. Quem ainda não fez sua inscrição, corra! 

Coluna do leitor

Esta seção reúne os últimos comentários, mensagens, e-mails e sugestões que recebemos de nossos leitores.

Sou leitor assíduo da revista desde o seu 1º número. É uma leitura fascinante. Sempre abordando temas atuais e relevantes, contribuindo com nosso aprendizado e, mais importante, evangelizando com software livre.

Raphael Silva Souza - Macarani/BA

É uma ótima iniciativa dos autores. Como sou usuário de Linux, tenho dificuldade em encontrar publicações sobre software livre.

Roberto Takeshi Matino - Campinas/SP

O Brasil estava precisando de uma revista Livre e sobre Software Livre. A revista Espírito Livre veio para preencher essa laguna e levar o conhecimento do software livre a todos, sem excessão.

Helder Oliveira - Presidente Prudente/SP

É um Projeto/Iniciativa que tem tudo pra dar certo e está dando certo, não é à toa que está na 42ª Edição. Com certeza tem algumas dificuldades, mas pra alcançar a vitória e aquilo que acreditamos, teremos que superar os desafios encontrados, no caso, nessa grande jornada que é de levar o Software Livre às pessoas. Levar o conhecimento e compartilhando conteúdo significativo e de qualidade. É uma revista que eu gosto e que adoro divulgar. Quem sabe algum dia, quando eu adquirir mais conhecimento, possa também colaborar com a REL e com todos os seus outros leitores sempre levantando a bandeira do SL.

Edson L. de Araújo Silva - Castanhal/PA

A Revista Espírito Livre é uma publicação moderna que nos traz novidades do mundo do software livre e ao mesmo tempo incentiva a todos a buscar cada vez mais informações sobre aplicações livres.

Paulo Henrique P. Cardoso - Valinhos/SP

A Revista Espírito Livre acrescentou um novo horizonte em minha vida de conhecimento e informação.

Filipe Neres de Araújo - Jundiaí/SP

Vou começar a ler. Mas pelo índice, é fundamental para nós professores que trabalhamos com as TICs nas escolas e com Software Livre. Agradeço a oportunidade de utilizar tantas informações importantes. Pretendo compartilhar com os colegas.

Gilberto C. de Oliveira - Santa Maria/RS

Defino em uma palavra: PERFEITA!

Fagner M. Camelo - São João de Meriti/RJ

É uma revista informativa feita por quem tem um ideal e o defende sem sofrer influência de empresas privadas.

Cassio B. de Oliveira - Ilha Solteira/SP

Um importante instrumento de divulgação do Software livre no Brasil.

Elias Alves de Souza - Rio das Ostras/RJ

Uma revista atual que aborda temas relacionados ao crescimento profissional com foco no software livre.

Mário Luiz Moraes da Silva - Tubarão/SC

Uma das poucas revistas, senão a única, que trata do ponto de vista político e filosófico do Software Livre em língua portuguesa. Sempre acompanho e recomendo para minha equipe e meus alunos.

Ricardo Lino Olonca - São Paulo/SP

Simplesmente adoro a revista. Conteúdos variados e com uma proposta ótima do uso de ferramentas open source

Douglas de Andrade - Uberlândia/MG



Promoções

8º SoLiSC
congresso catarinense
de software livre
solisc.org.br



Sorteio de inscrições. Participe!




fisl14
14º Fórum Internacional
Software Livre

Sorteio de inscrições. Participe!

VirtualLink



Soluções e Treinamentos em Linux
Sorteio de kits de CD e DVD.
Clique [aqui](#) para concorrer!



CLUBE DO HACKER
Os melhores estão aqui!

Sorteio de associações para o clube.
Clique [aqui](#) para concorrer!

Relação de ganhadores de sorteios anteriores:

Ganhadores da promoção Clube do Hacker:

1. Milton Bragança Sousa - Porto Velho/RO
2. Daniel Ferreira - Petrolina/PE
3. Fred Mercury Freitas Matos - Presidente Médici/RO

Ganhadores da promoção VirtualLink:

1. Diogo Orro de Campos Viegas - Campo Grande/MS
2. Danielle Maria Siqueira Silva - Guarulhos/SP
3. José de Ribamar de Jesus Costa - São Luis/MA
4. Filipe Araújo - Jundiaí/SP
5. Everton da Rosa - Três de Maio/RS

Ganhadores da promoção PASL.NET.BR:

1. Vanildo Souto Mangueira - São Paulo/SP
2. Marden Laairoy Correia de Oliveira - Maceió/AL
3. Marlon Santos Coutinho - Araçuaí/MG
4. Nelson Martins Ferreira - Mesquita/RJ
5. Roberto Carlos Pereira Junior - Belo Horizonte/MG

Ganhador do Livro "Muito além do antivírus":

1. Douglas de Andrade - Uberlândia/MG



REVISTA **espírito livre** LIBERDADE E INFORMAÇÃO
<http://www.revista.espiritolivre.org/>

SoLiSC

congresso catarinense
de software livre
solisc.org.br



O EVENTO

O SoLiSC é o maior evento de Software livre de Santa Catarina e vai proporcionar a chance de você ampliar seus conhecimentos.

Uma oportunidade imperdível de desenvolver suas habilidades na Grande Florianópolis.

Participe, conheça e domine um mundo de novas informações.

TRILHAS

Software Livre em Geral;

Hardware e Sistemas Embarcados;

Administração de Sistemas;

Desenvolvimento.

www.solisc.org.br



Deputado questiona parceria entre Estado e Microsoft

por Antônio Carlos Ribas

O deputado Edson Praczyk- PRB, em pronunciamento na tribuna da Assembleia Legislativa, manifestou sua preocupação com as notícias que estão sendo veiculadas na mídia sobre a assinatura de um Protocolo de Intenções entre o Governo do Estado e a Microsoft.

Este acordo, proporcionaria a utilização do sistema de software proprietário ao invés do software livre nos órgãos públicos ligados ao governo estadual modifi-

cando um sistema que é utilizado atualmente, e que ao longo dos anos propiciou a economia de muitos milhões de reais aos cofres do Estado, fruto da aplicação da Lei Estadual 14.058, de 2003, de sua autoria, que indica o uso preferencial do software livre, em todos os órgãos da Administração Pública direta, indireta, Autárquica e Fundacional do Estado do Paraná, bem como os órgãos autônomos e empresas sob controle estatal.

Este sistema já permitiu não só aos órgãos estaduais, mas também ao próprio governo federal, uma economia extraordinária. Inclusive com o relato de diretores destas estatais como a Sanepar. Seria estranho mudar agora, e exatamente por isso, Praczyk quer saber se isso não atropelará sua lei. Quais os termos deste Protocolo de Intenções, os detalhes deste convênio, e como ficaria o sistema atual que utiliza o software livre? Quais as intenções e os motivos deste convênio?

O deputado ainda destaca que os ganhos com o software livre são de conhecimento de todos os paranaenses. A rede de conhecimento do Estado, através do Portal da

Educação, utiliza o sistema Linux. Por isso quer saber por que o governo vai assinar este Protocolo de Intenções com a Microsoft? Praczyk espera uma resposta por parte do Governo já para os próximos dias. 🇧🇷

Autor: Antonio Carlos Ribas

Foto: Derli Junior/ Edson Praczyk

Entenda o Caso

O governo do Estado do Paraná assinou no dia 09 de abril, durante o Fórum de Líderes de Governo na América Latina e Caribe, no Rio de Janeiro, protocolo de intenções com a diretoria da empresa Microsoft.

O protocolo de intenções prevê cooperação mútua entre o Governo do Estado e a multinacional norte-americana de tecnologia e informática nas áreas de educação, qualificação, inovação e empreendedorismo. A empresa disponibilizará ao Estado programas e conteúdos de capacitação em tecnologia da informação, com o uso de plataformas de aprendizagem virtual.





Entrevista: Julian Carlo Fagotti

por João Fernando Costa Júnior

REVISTA ESPÍRITO LIVRE - Apresente-se para os leitores da Revista Espírito Livre.

JULIAN CARLO FAGOTTI - Meu nome é Julian Carlo Fagotti. Fui assessor da Companhia de Informática do Paraná – Celepar, desde o início da implementação do Software Livre no Governo do Estado, na Gestão do governador Roberto Requião e do presidente da Celepar Marcos Mazoni, em 2002. Sai em 2010.

REVISTA ESPÍRITO LIVRE - Recentemente tivemos uma notícia desanimadora quanto a adoção do software livre no governo do Paraná. Houve manifestações em listas de discussão e notícias a respeito. Qual seu ponto de vista quanto a este acordo?

JULIAN CARLO FAGOTTI - Acordos de cooperação de desenvolvimento tecnológico são sempre bons a princípio. No entanto as vantagens dos dois lados dos cooperados tem que ser equilibradas. No caso da Microsoft eu desconheço acordos feito por ela com tal equilíbrio, porque criar códigos em cima de uma plataforma fechada que causa dependência tecnológica de uma das partes, inviabiliza a equidade antes mesmo de ser digitada a primeira linha de código. Quando estávamos na Celepar fizemos muitos acordos de cooperação com software livre. Cooperamos muito código aberto. Aí é cooperação de verdade.

Quando estávamos no governo optamos pela cooperação com a UFPR, com as comunidades do e-groupware, do PostgreSQL, do hoje chamado LibreOffice, e no primeiro evento internacional de software livre que organizamos, tivemos a Microsoft como patrocinadora do evento. Acabou não dando certo, mas foi uma tentativa de colocar a Microsoft num mesmo patamar de colaboração.



Acordos de cooperação são sempre bons a princípio.



REVISTA ESPÍRITO LIVRE - O Brasil conhece a Celepar e o seu trabalho na popularização e desenvolvimento de ferramentas de software livre. Tal acordo foi como um balde de água fria em anos de trabalho. Na sua opinião, o que levou o governo paranaense a assinar este acordo de intenções com a Microsoft?

JULIAN CARLO FAGOTTI - Opção por Software Livre é política, não está dissociada da política partidária. Existem no Brasil basicamente duas posições a respeito de tecnologia. Um grupo de partido que não acredita na capacidade de desenvolvimento da sua própria gente, e por isso sente necessidade de uma tutela tecnológica, e outro que acredita na capacidade de desenvolvimento próprio que se relaciona de igual para igual com qualquer empresa de qualquer país. Não

foi este acordo que foi um banho de água fria na política de software livre. Foram as eleições de 2010 no Paraná. Mas foi um banho de água fria. Não uma pneumonia com risco de morte. Apenas o Governo do Paraná deixou de fazer, a vida continua, outras empresas públicas e privadas continuam a fazer e usar software livre. Creio que a maioria dos servidores do Datacenter do Governo do Paraná, na Celepar ainda sejam em Linux, por exemplo.

REVISTA ESPÍRITO LIVRE - Sabemos o quanto é difícil implementar uma política de uso de software livre. Quais os reflexos e consequências deste acordo?

JULIAN CARLO FAGOTTI - Duvido da produtividade deste acordo. Eventualmente um ou outro profissional pode ter alguma vantagem, mas dificilmente vá mudar o panorama do desenvolvimento de software no Brasil, pelas razões que já citei.

REVISTA ESPÍRITO LIVRE - Com a assinatura de acordos como esse, quem sai ganhando e quem sai perdendo?

JULIAN CARLO FAGOTTI - Moralmente a Celepar, que era identificada no país e no mundo como desenvolvedora de software, e livre, e foi submetida à posição de consumidora de software.

REVISTA ESPÍRITO LIVRE - Você acha que tal medida tem força suficiente para enfraquecer a comunidade de software livre no Brasil?

JULIAN CARLO FAGOTTI - Não. Tem muito software para desenvolver no mundo. O Brasil ainda desenvolve mais software livre nas estatais do que nas empresas privadas. É com isto que a comunidade de



Não creio que o governo do paraná seja um exemplo a ser seguido.



software livre tem que se preocupar, resolver problemas das empresas do Brasil. As estatais que tem política de software livre estão fazendo sua parte. As que já tiveram deixaram seu legado. Tem muito software desenvolvido pela Celepar que é livre, cabe as comunidades de software livre reivindicá-los, modificá-los e colocar a disposição de cooperativas agrícolas, bancos, varejo, etc.

REVISTA ESPÍRITO LIVRE - Qual o recado que você manda para os nossos governantes que pretendem seguir o exemplo do governador do Paraná?

JULIAN CARLO FAGOTTI - Não creio que o governo do Paraná seja um exemplo a ser

seguido. Mas os governos talvez tenham muito mais em cooperar entre si do que com a Microsoft. O Brasil tem 27 unidades federativas, mais de 5 mil municípios, e um governo federal, uma penca de universidades e empresas sob a mesma legislação, a mesma realidade econômica. Tem muito o que cooperar para desenvolvimento tecnológico. Não quer dizer que não devam fazer projetos com a iniciativa privada também. Ao contrário. Mas o que não dá pra entender é esta predileção pela posição de colonizado. Vamos nos levar a sério!

REVISTA ESPÍRITO LIVRE - O Partido Pirata do Paraná no último dia 25, pediu ao governador uma cópia deste acordo. Como você encara tal ação? Acha que poderá ser bem-sucedida e que trará reflexos positivos?

JULIAN CARLO FAGOTTI - O Estado do Paraná tem a mesma tradição dos atos secretos do Senado. O senado já acabou com isso, no Paraná recomeçou. Mas tem que pedir os contratos, afinal, administração pública, é pública!

REVISTA ESPÍRITO LIVRE - Deixe algumas palavras para os nossos leitores.

JULIAN CARLO FAGOTTI - Escrevemos poucos códigos para resolver nossos problemas. Longe de ser xenófobo, temos que entender que a maioria dos projetos de software livre são para resolver problemas

de empresas dos EUA como bem constatou o Frank Alcântara. Precisamos criar o vínculo com as empresas privadas brasileiras. O software livre precisa entrar no ciclo econômico brasileiro, que está num bom momento, e tem pressa. Quando trouxemos o e-gropware para transformá-lo no Expresso superamos todas as metas de planejamento do projeto original na Alemanha, dando um porte ao projeto que os criadores nem imaginavam. É este tipo de experiência que precisa ser replicada no meio privado. O software livre brasileiro estatal vai bem, apesar de retroceder no Paraná. Quem precisa criar asas é o software livre brasileiro privado. 🇧🇷

JOÃO FERNANDO COSTA JÚNIOR é membro da The Document Foundation – TDF. Idealizador, responsável e editor da Revista Espírito Livre. É administrador por formação e especialista em informática em educação. Moderador da PSL-ES e participante da Tux-ES. Atua ainda como professor de cursos técnicos.



HostGator
HOSPEDAGEM DE SITES

- ✓ Servidores Linux de alto desempenho
- ✓ Painel cPanel em português
- ✓ Transferência e espaço ilimitados

Hospede seu site com uma das **melhores do mundo!**

HOSTGATOR.COM.BR



Software Livre em xeque no Paraná

www.sxc.hu

por Marcelo Soares Souza

Desde 2003, o Governo do Estado do Paraná tornou-se uma referência nacional no que diz respeito a independência e inovação tecnológica sustentável. Seja pelo compromisso firmado pelo Governo do Estado em prol da adoção de soluções livres e de padrões abertos, seja pelas ações exemplares da Companhia de Tecnologia da Informação e Comunicação do Paraná - CELEPAR, a mais antiga empresa pública de informática no Brasil e que neste ano completa 49 anos.

Há quatorze anos, o primeiro projeto de lei versando sobre o uso preferencial de Software Livre pelo Governo Federal foi apresentando na Câmara dos Deputados e continua tramitando até os dias de hoje. Quatro anos depois, foi aprovada e sancionada, no Paraná, as leis 14058/2003 e 14195/2003 que tratam sobre o uso preferencial de soluções tecnológicas livres e de código-fonte aberto pelo Governo do Estado do Paraná.

Desde então, o Governo do Estado do Paraná vem promovendo ações como o Paraná Digital - ação de Inclusão Digital presente em mais de duas mil escolas públicas, atendendo a um milhão e meio de crianças e adolescentes, possibilitando o desenvolvimento socioeconômico sustentável utilizando Software Livre. Muito além da simples formação técnica, está o exemplo de que o desenvolvimento pessoal se dá também pela colaboração e compartilhamento. Com uma base de mais de meio milhão de usuários, distribuídos entre 167 instituições, o Expresso Livre é sem dúvida o maior exemplo de que é possível desenvolver soluções tecnológicas de alta qualidade, e de forma colaborativa, no Brasil. O Expresso Livre nasceu da necessidade de uma ferramenta corporativa de correio eletrônico livre para o Estado do Paraná. Desde o princípio, rapidamente foi ganhando novos colaboradores, novas instituições aderiram e a tornaram uma solução tecnológica de excelência amplamente utilizada e desenvolvida.

Tratando o Software, expressão prática do conhecimento, como um meio de produção e não simplesmente como um produto, o Governo do Paraná possibilitou o surgimento de novos empreendimentos econômicos com bases sólidas sobre as tecnologias disponibilizadas.

Novos empreendimentos econômicos surgiram, utilizando, desenvolvendo e recompartilhando o conhecimento disponibilizado livremente, oferecendo soluções inovadoras, com alto valor agregado. Competitividade é o fruto mais doce da inovação, e inovação é o resultado direto do ousar o novo e ao adotar instrumentos e ações para a promoção de um ambiente aberto o Estado do Paraná lançou bases para um novo paradigma desenvolvimentista. Ao ser pioneiro na abertura dos códigos fontes de suas soluções tecnológicas o Estado do Paraná contribuiu diretamente para o desenvolvimento socioeconômico do estado e do Brasil.

No mesmo ano em que se comemora os 10 anos do compromisso firmado pelo Governo do Estado do Paraná com a liberdade e a independência tecnológica, no dia 09 de Abril o governador do Paraná Beto Richa assinou um acordo de

intenções com a Microsoft no Hotel Copacabana Palace, no Rio de Janeiro. O presidente da Microsoft Brasil sequer teve o trabalho de se deslocar até o Paraná para assinar tal acordo.

Este, claro, não é um fato isolado, em 2008 o Governo do Estado da Bahia firmou acordo parecido com a Microsoft. Nota-se um crescente retrocesso nas políticas e ações públicas na questão da adoção e fomento de ações ligadas ao Software Livre e sobre os Padrões Abertos.

No ano em que se comemora os 30 anos do início do desenvolvimento do Sistema Operacional GNU e os 20 anos desde o registro do primeiro usuário GNU/Linux no Brasil, temos pouco a comemorar. 🇧🇷

MARCELO SOARES SOUZA é bacharel em Informática pela Universidade Católica do Salvador. Consultor, desenvolvedor, técnico, tradutor e docente em tecnologias da informação e comunicação. Atuou em Programas de Inclusão e Cultura Digital do Governo Federal como Casa Brasil, GESAC e no Pontão de Cultura Digital juntaDados.org. Blog: <http://marcelo.juntadados.org>.



NOVO. RÁPIDO. LIVRE.
LIBRE.



The Document Foundation
apresenta:

LibreOffice



Writer



Calc



Impress



Draw



Base

A suíte de escritório em software livre mais avançada.

pt-br.libreoffice.org



Software Livre “no limbo”

Porquê ainda lidamos com notícias sobre adoção, e posterior abandono do software livre?

Como em poucos lugares do mundo, software livre no Brasil tem sido tratado constantemente como um elemento ideológico, onde muitos adotam o uso do software livre como quem gosta de um time de futebol, um clube ou um partido. Contudo, software livre nasceu a partir do conceito de disseminação de conhecimento e ciência, nunca foi fruto “deste” ou “daquele” partido.

O Brasil, infelizmente, vive uma situação surreal onde, mesmo que um determinado software seja avaliado como tecnicamente seguro e estável, ou ainda mais, como a melhor opção para uma determinada finalidade, quando mudam governos, mudam as políticas, mudam os enfoques técnicos, mudam os softwares.

É certo de que este artigo venha a despertar amores e ódios. Infelizmente, sei que temos de lidar com isso! Software livre tem sido SIM ferramenta de palanque. Mas, quando o partido perde, muda tudo, e o software muda também. Será que algumas vertentes da comunidade de software livre no Brasil estão errando em algum ponto, ou talvez em vários pontos?

por David Jourdain

CAUSOU GRANDE MOVIMENTAÇÃO e desconforto nas listas de discussão, nos fóruns e nas comunidades virtuais onde se reúnem membros de diversas comunidades de software livre, quando da divulgação da notícia a respeito da assinatura de um protocolo de intenções entre o Governo do Estado do Paraná e a Microsoft, em que o citado documento prevê cooperação mútua nas áreas de educação, qualificação, inovação e empreendedorismo. A Microsoft disponibilizará ao Estado do Paraná programas e conteúdos de capacitação em tecnologia da informação, com o uso de plataformas de aprendizagem virtual.

Independente de concordar ou não (não é este o objetivo deste artigo), temos que lembrar que ações como estas, se amparadas por análises jurídicas e pareceres favoráveis, são lícitas! Sei que alguns poderão dizer: *"É!! Um cara de software livre falando que um acordo desses é lícito? É um vendido!!"*.

Volto a dizer que não é a intenção deste artigo discutir se é lícito ou não um acordo desses. Muito provavelmente seja! Entretanto, minha intenção é discutir porquê situações como estas são corriqueiras e fazem com que seja visto com certa frequência o software livre ser adotado, para depois de um tempo ser abandonado, para novamente ser adotado e, assim segue, num *"vai e vem"* monótono e frustrante. Aconteceu em várias regiões do Brasil, ora em cidades, ora na administração estadual, e até mesmo na esfera federal onde *"em tese"*, vivemos um governo de continuidade. E até mesmo em empresas do governo federal, *"em tese"* defensoras e investidoras do software livre, que ora divulgam publicamente que investirão no desenvolvimento de software livre, para depois abandoná-los e jogar *"para debaixo do tapete"* toda a divulgação, com se a internet tivesse tapete para esconder estas notícias.

Acontece que, pelo fato do software livre ser avaliado como um elemento ideológico, como de *"esquerda"*, quem é de *"centro-esquerda"*, *"centro"*, *"centro-direita"* ou de *"direita"* não se dá ao trabalho de transpor a imagem ideológica para analisar os benefícios do uso do software livre nas diversas esferas de governo, ou mesmo nas empresas de governo. Se chegou até aqui na leitura, significa que o texto ainda não ofendeu você. Que bom por isso. Voltando ao parágrafo inicial, onde dizia que muito provavelmente o protocolo assinado entre o Governo do Estado do Paraná e a Microsoft seja lícito, eu honestamente creio que seja. Mas a pergunta que precisa ser feita é: **Este protocolo é conveniente para o Estado do Paraná? Para a administração do Estado do Paraná? Para a Educação do Estado do Paraná? Meu avô materno tinha o costume de dizer que *"nem tudo que me é lícito, me convém!"*.**

Não estou questionando o direito da Microsoft de oferecer produtos e serviços a quem quer que seja. Governo ou não. Questiono sim se, após anos de investimentos em uso, treinamento, desenvolvimento de softwares que são utilizados em diversas esferas da administração do Estado do Paraná, é conveniente uma mudança dessas? E *"se"* for, qual será o custo desta mudança? E ainda assim, se o custo compensar, não seria mais produtivo que fosse revertido em novos investimentos para melhorias dos softwares já em uso? Compreendem meu ponto de vista? Se considerarmos estas premissas, estou certo que muitos profissionais de TI, isentos de posicionamento ideológico partidário, chegariam a conclusão de que mudanças tão drásticas assim precisariam ser *"muito justificadas"* para compensar empenho e dedicação para uma mudança dessas. Situação semelhante está ocorrendo já a alguns anos, quando a prefeitura de Munique na Alemanha

entendeu que mudar toda a sua estrutura para software livre, entre diversos fatores positivos, teria o fator de independência de fornecedor e o pleno domínio dos aplicativos que são utilizados, sem esquecer que eles apresentaram planilhas públicas, exibindo que financeiramente era muito fácil justificar a mudança. Mas vale lembrar que a imagem do software livre "precisa" desgrudar da figura de que "software livre é igual a software de graça"! Grande parte dos problemas no Brasil para que projetos de software livre sejam realmente viáveis reside na ilusão de que o software livre "nasce em árvores" e que ninguém precisa pagar por ele. Inclusive, é por este posicionamento que o Brasil é conhecido atualmente como o maior consumidor de software livre no mundo, mas a colaboração brasileira para projetos de software livre de alcance mundial é muito pequena, para não dizer pífia!

MAS EIS O PONTO EM QUESTÃO: É conveniente uma mudança dessas, depois de tanto investimento?

Na verdade, para os que trabalham em repartições públicas subordinadas a administrações municipais e estaduais, sabem que não se deve dedicar muito esforço aos processos implantados pelo governo eleito, pois na próxima eleição ele será substituído e tudo será mudado. Esta triste situação se deve a falta de

compreensão de que políticas partidárias, na maioria absoluta das vezes, são burras! Sim, é isso mesmo que você leu. Políticas partidárias são, por princípio, burras! E são burras porque desconsideram o todo, consideram apenas uma parte do todo, apenas as opiniões do partido ou dos aliados, muitas vezes "aliados" que se unem por motivos não muito republicanos. Sendo assim, como a oposição não foi levada em consideração no processo do projeto das políticas, não se cria uma verdadeira política pública, mas sim uma política partidária. Muda o partido, muda a política! Se eu estiver dizendo alguma novidade para algum dos leitores, por favor, me corrijam!

Avaliando um pouco além das nossas fronteiras, podemos ver o que ocorre em alguns países que preferiram adotar políticas públicas, em vez de políticas partidárias. Veja o caso de Munique. Todo o programa de mudança para adoção do software livre começou em 2004. De lá para cá, governos mudaram. Ora a esquerda estava no governo, ora a direita, a ora a centro-direita, mas o projeto continuou. E por quê continuou? Continuou pois ficou claro que software livre não é um elemento ideológico. Ele pode ser analisado como útil e com resultados positivos claros, independente de quem está avaliando, se é de "direita" ou de "esquerda", ou seja lá de que lado for! Veja a situação do software livre nos Estados Unidos. Com todo respeito a meus amigos americanos, nem sei se eles sabem o quê é um partido de esquerda. Para eles, se você está "um pouco" perto do centro, você é comunista! E mesmo lá, o software livre é maciçamente utilizado em diversos órgãos de governo, em diversos estados e cidades, e até mesmo em empresas e programas de governo, como por exemplo a Nasa. A Finlândia, que mais próximo chega de uma "esquerda" política, teve governos sucessivos de centro e centro-



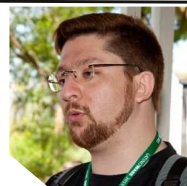
direita, e lá o uso de Linux e outros softwares livres é basicamente questão de honra, além de questão de domínio de conhecimento. Outros governos, como os de diversas cidades e algumas repartições da administração federal francesa, algumas outras na Alemanha, a rede de hospitais públicos da Dinamarca, todos eles países que não tiveram tradição de "esquerda" ou de "direita". Na verdade, alguns destes países têm tradição é na alternância de governos, sem que isso signifique mudanças radicais nas políticas para o Estado, pois elas são públicas e não partidárias.

Um exemplo mais próximo de nós encontra-se no Chile. Após a redemocratização, os partidos acordaram algumas linhas gerais para a criação de políticas públicas para o país, e que é de conhecimento da sociedade. Com isso, se tiver um governo de esquerda, ou de direita, ele terá que seguir claras linhas das políticas públicas planejadas para o desenvolvimento do país. Esta situação é tão clara no Chile, que vemos alunos reclamando do governo, seja ele de direita ou de esquerda, pela falta de aplicação das políticas públicas ou pela demora na

aprovação de novas políticas públicas.

Enquanto vivermos nesta constante "partidarização" do software livre, continuaremos a ver outras circunstâncias semelhantes a estas ocorridas no Estado do Paraná. Ora o software livre está no Governo, ora sai! Ora volta e fica por um tempo, depois sai novamente! Enquanto isso, patinamos enquanto sociedade, esperando que a classe política entenda que a sociedade precisa de mais políticas públicas e menos políticas partidárias. A sociedade agradece, e as comunidades de software livre no Brasil também. A não ser que hajam muitos que gostem deste enfadonho "vai e vem" do software livre no governo. Eu sei que corro o risco de levantar amores e ódios, infelizmente. Mas tudo bem, ninguém é perfeito! Ou será que tem alguém que, baseado em suas crenças políticas, crê que é perfeito? 🇧🇷

DAVID JOURDAIN é membro fundador da The Document Foundation - TDF. Moderador das listas da TDF em língua portuguesa. Tem toda a sua formação na área da Computação. Mais de 12 anos "mexendo" no Kernel Linux. Já foi professor universitário em diversos países, ensinando sobre Kernel Linux e como organizar grupos de desenvolvedores de software livre em Universidades.





O que move as pessoas são os seus sonhos

por Aleksandro Montanha

Estes sonhos se tornam realidade quando o brilho nos olhos irradia e encanta uma legião de interessados em torná-lo tangível, possível e real.

O Software Livre é, em sua essência, uma ideologia, um princípio, um hábito que é conseguido e praticado. Com a mesma força de um movimento revolucionário o caminho da esquerda da tecnologia tomou o seu posto e assumiu uma posição estratégica em diversas áreas e isso, chamou a atenção de muitos.

Vigaristas de plantão se jogaram no meio dessa imensidão de bits e tomaram assentos num atribulado conjunto de siglas, convenções, torneios, eventos, palestras, documentos, projetos e diversos outros movimentos gerados pelos sonhadores.

Muitos desdes sonhadores acordaram de pesadelos e se viram como marionetes comandados pelo pleonismo, “político oportunista”.

Também vimos as gran-

des empresas esponjas, sugando o néctar dos projetos livres e se esbaldando dos benefícios que a exploração predatória lhe oferece.

Por outro lado, sinceramente, alguns de nossos gurus gritando liberdade, liberdade e comemorando a vitória sobre o cadáver de ilustres competidores, não enaltece em nada um movimento tão bonito. Viver criando projetos pelo simples exercício de ser o diferente, não permite a evolução de outros já existentes. Pensar no holofote antes do resultado, não traz perpetuidade, nem para o projeto nem para o seu autor. Portanto, o superficialismo é um mau comum, que atinge quem está dentro e quem não está conosco. Algo deve ser feito, pensado, discutido e planejado. Somos uma classe que ainda não tem representatividade, um bando de ovelhas que ilustram os sonhos de ilustres empresários de sucesso.

Que fica indignado pela troca de plataforma de um governo que era um aliado,


mas por outro lado não se movimenta nem para organizar eventos de software livre.

A Skynet ainda está viva. 🙋

ALEKSANDRO MONTANHA

é mestre em Ciência da Computação, especialista em Análise de Sistemas e em Desenvolvimento com Software Livre. Tem experiência como gerente de projetos e docente de cursos de Graduação e de Pós-Graduação. Atualmente trabalha como consultor de tecnologia aplicada e desenvolvimento de aplicações de processamento de imagens.





O uso de software proprietário pela Administração Pública: uma decisão de retrocesso a sustentabilidade e a economia

por Roney Médice

O surgimento da Administração Pública no Brasil se deu através do fracasso da outorga da administração de territórios à particulares, através do sistema conhecido como capitânias hereditárias e a instituição de um governo geral, em meados de 1549. Com o passar do tempo, na evolução histórica, ocorreram diversas transformações na reforma da administração pública, como por exemplo, a instituição por decreto dos princípios da administração pública até a reforma promovida pela Constituição Federal Brasileira de 1988, tratando dos princípios: Eficiência, Legalidade e Impessoalidade.

A administração pública tem um papel fundamental na sociedade que é garantir que a vontade da coletividade seja exercida dentro da legalidade (o administrador só pode realizar ações que a lei determinar e permitir) e também com Eficiência (que as ações exercidas sejam realizadas com resultados positivos e no menor custo para o erário). Dessa forma, os administradores públicos precisam conciliar a vontade do povo com as necessidades do Estado realizando uma evolução social e econômica dentro dos parâmetros da administração pública.

Estamos vivendo em um momento na história que

muito se fala em crescimento sustentável, gerando desenvolvimento econômico respeitando o meio ambiente e com eficiência. Cabe a cada pessoa conceituar e refletir as ações necessárias para que a sustentabilidade caminhe junto com resultados eficientes e econômicos, proporcionando uma equilibrada e verdadeira evolução.

Para tanto, na administração pública, muitos gestores escolheram o uso do software livre para conseguir um desenvolvimento sustentável e garantiram uma boa economia aos cofres públicos evitando pagar por uma tecnologia que pode ser usada com bons programas gratuitos de qualidade equivalente ao software pago.

Os softwares livres são os programas que podem ser usados, copiados, estudados, modificados, redistribuídos sem restrições e melhor ainda, sem precisar pagar pelo programa. Qual administrador público não quer utilizar software livre em sua gestão para demonstrar eficiência e economia ao seu povo?

A utilização de software livre na gestão pública brasileira já está bem visível há anos. Alguns exemplos de utilização do software livre na administração pública podem ser conferidos no site <http://www.softwarelivre.gov.br>. A tendência natural

é que os softwares proprietários sejam colocados de lado e que se utilize os softwares livres, economizando e muito o valor pago pela licença de uso aos aplicativos desenvolvidos por empresas privadas.

O software livre possui inúmeros benefícios que só tem a somar com a administração pública. Atualmente, governos e prefeituras com recursos escassos, dependentes de verbas para mover a máquina pública, com o uso desse tipo de software, evita gastos desnecessários como o pagamento de nova licença a cada atualização do software proprietário que com o tempo fica obsoleto e inseguro, na medida que o fabricante de um software proprietário deixa de disponibilizar atualizações de segurança e focando em versões mais atuais, com valores cada vez mais caros.

Entretanto, alguns administradores públicos estão andando na contramão do desenvolvimento sustentável e optando por trocar todo o parque tecnológico que utiliza software livre por programas proprietários, pagando um preço alto. Essa ação requer um estudo maior sobre quais os objetivos ou qual a real motivação que leva uma administração a um retrocesso, deixando de lado a economia obtida por não

comprar os programas utilizados atualmente e os seus funcionários já habituados no seu uso.

Seria um jogo de interesse por parte da administração pública a aquisição de software proprietário em vez de um software livre? Quais os beneficiários dessa decisão que demonstra claramente um retrocesso da economia pública? Respostas essas que precisam ser esclarecidas e que o povo precisa ter a informação concreta pois não se pode aceitar meramente alguns atos da administração com base em suposições ou favorecimento pessoal.

O que se espera de uma administração pública é a sua autonomia em escolher qual fabricante ou software será utilizado nos órgãos públicos, independentemente da tecnologia e do desprendimento de um fornecedor em específico. Não se pode escolher um aplicativo apenas pelo conheci-

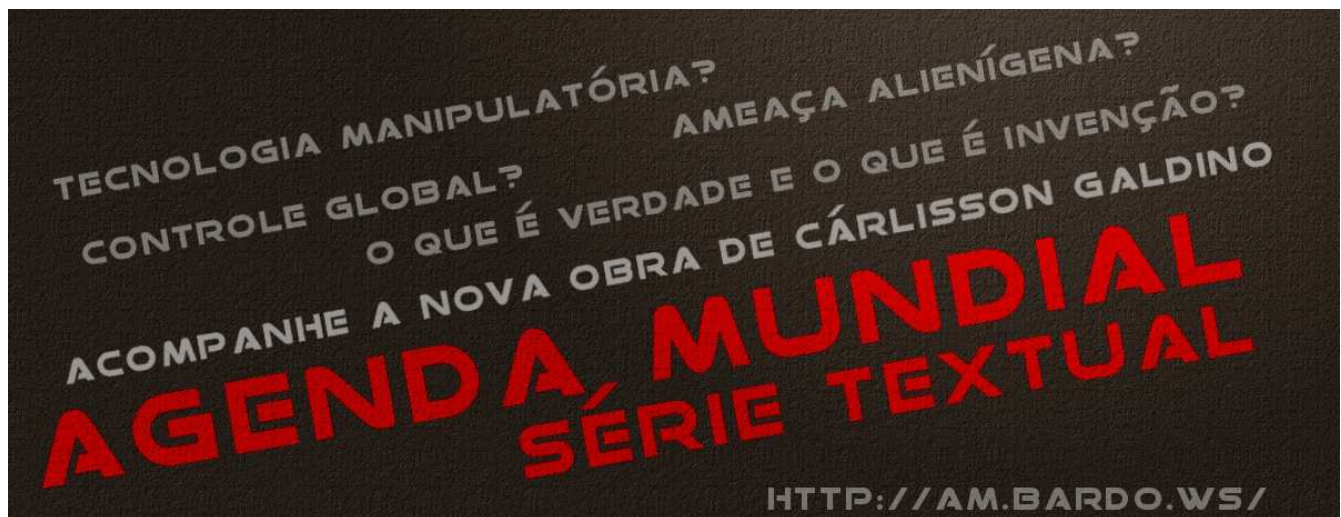
mento intelectual do fabricante. Tem que atender a demanda e de forma coerente e imparcial, sem privilegiar uma determinada empresa especificamente.

A preferência é tentar utilizar tecnologia sem custo, que permite direcionar os recursos que seriam aplicados na compra de software proprietário para outras áreas prioritárias, como a saúde, por exemplo. Não se pode menosprezar o software livre em um momento em que há lei da responsabilidade fiscal, onde o gestor público tem um limite de gasto que não pode comprometer os percentuais legais para evitar responder a processos por improbidade administrativa.

Contudo, esperamos que os nossos governantes levem a sério a administração pública respeitando o que já conquistamos e os avanços que foram conseguidos com o uso de software livre nos órgãos da máquina pública.

Assim, evitaremos que determinadas empresas sejam beneficiadas por decisões estranhas que não condizem com a ansiedade da população, que é realizar os atos administrativos com transparência, imparcialidade e eficiência. 🇧🇷

RONEY MÉDICE é graduado em Ciência da Computação, Direito e MBA em Gestão de Segurança da Informação. Consultor de Segurança da Informação. Perito em Computação Forense com certificação CDFI. Membro do CSA - Cloud Security Alliance, membro do Comitê ABNT/CB-21, Presidente da APECOMFES - Associação de Peritos em Computação Forense do Espírito Santo.



BRASIL FOSSGIS

Georreferenciando o Conhecimento

<http://fossGISbrasil.com.br>



AQUI VOCÊ ENCONTRA CONTEÚDO FEITO SOB MEDIDA DOS TEMAS MAIS ATUAIS E DIVERSIFICADOS SOBRE O MUNDO DAS GEOTECNOLOGIAS LIVRES



<http://twitter.com/fossGIS>



<https://www.facebook.com/FOSSGISBrasil>



O uso do Software Livre no governo

por Milton José Lyrio Simonetti

Como diriam os anarquistas (concepção política): Para que governo?

De fato, não se pode dizer, radicalmente, “sou alheio à política”, nem que “manterho distância da política”. Sem aprofundar na filosofia, estando-se todos imersos em uma sociedade, de alguma forma o controle dessa sociedade nos diz respeito. Ficando-se alheio, permite-se que quem “dá as cartas” continue agindo a seu bem entender. De certa forma, é uma concordância com quem manda.

Olhando-se por esse ângulo, pode-se considerar duas vertentes principais: a pri-

meira defende que o estado, o governo, não deve interferir na vida da sociedade, a qual saberá se “autodirigir” através do mercado. A segunda defende que a mão do estado deverá tomar medidas que regulem a organização da sociedade.

Para se discutir a atuação do governo na adoção do software livre, portanto, e mesmo que não se manifeste um ponto de vista antecipadamente, deve-se ter um alinhamento com uma destas vertentes. Implícita ou explicitamente uma das duas estará embutida em cada ponto de vista.

Nessa perspectiva, uma das defesas do software livre considera o papel do estado na organização social como necessário e determinante. Se forem levados em conta os investimentos que um governo faz em software, vê-se que tais recursos fatalmente influenciarão nas tendências de mercado, valorizando ou não determinada linha de desenvolvimento, pelo próprio conjunto de recursos destinados ao suprimento de suas necessidades. Ainda deve-se considerar os “efeitos colaterais”, pois fatalmente a mão de obra necessária a um ou a outro tipo de software terá capacitação diferente.

Alguns governos, ao contrário, preferem adotar exclusivamente a abordagem financeira como parâmetro de definição, considerando que sua influência no mercado é irrelevante.

Creio que este seja o caso do Governo do Estado do Paraná, por exemplo, que anuncia em alto e bom som (<http://www.celepar.pr.gov.br/modules/noticias/article.php?storyid=1044&tit=-Parana-vai-usar-gratuitamente-solucoes-de-TI-da-Microsoft->) “Paraná vai usar gratuitamente soluções de TI da Microsoft”. Segundo o texto da matéria, “Pelo protocolo firmado com a Microsoft ficam disponíveis opções de programas e softwares que poderão ser implantadas pelo governo nas áreas de educação, qualificação, inovação e empreendedorismo. Entre eles estão capacitação básica em tecnologia da informação e design, comunicação instantânea entre alunos e professores, softwares para projetos de alfabetização em informática e desenvolvimento de empresas iniciantes.”

Esse é um típico exemplo de análise unicamente financeira, onde “efeitos colaterais” são desconsiderados pelos governantes.

Por outro lado, o portal do SERPRO destaca o encerramento do IV Planejamento Estratégico de Ações do Comitê Técnico de Implementação de Software Livre do Governo Federal (CISL). Nessa visão diame-

tralmente oposta à anterior, “o secretário adjunto do CISL, Deivi Kuhn, destacou que o trabalho do comitê contribui para a criação de políticas públicas por parte do Estado. ‘Não é só mudar a tecnologia, mas fazer algo para transformar a sociedade’, diz ele.”.

Já Antônio Carlos Tiboni, coordenador nacional do Programa Serpro de Software Livre, “ressaltou que a troca de experiências entre os diversos órgãos do governo foi um dos pontos fortes do evento. ‘Discutimos questões como ambientes de compartilhamento, planos de participação em cursos e divulgação de casos de sucesso’, explicou Tiboni. Um dos resultados dessa troca foi a definição de ações para que a administração pública federal conheça sua própria experiência no uso de soluções livres”.

Vê-se assim que a abordagem do tema software livre por parte dos governos tem um arcabouço ideológico determinante, seja para o lado de uma vertente ou para outro.

Nesse IV planejamento do CISL (<http://www.softwarelivre.gov.br/contatos-do-cisl/>), por exemplo, as diretrizes para o biênio 2013/2014, em linhas gerais, quanto ao software livre, podem ser agrupadas como:

DIRETRIZES

1.Colaboração

* Estimular o desenvolvimento colaborativo de software, as redes de colaboração e incentivar mudança de cultura organizacional.

2.Formação

* Promover formação contínua do quadro de pessoal da administração pública federal em Software Livre.

3.Adoção de tecnologias abertas

* Adotar soluções baseadas em Software Livre, Hardware Livre e padrões abertos.

4. Sociedade e Transparência

* Popularizar o uso de Software Livre na sociedade;

* Fortalecer movimentos sociais colaborativos em torno de tecnologias livres;

* Disponibilizar dados, processos, meios e códigos que permitam a transparência do governo, a auditabilidade plena e a segurança dos sistemas além de permitir à sociedade conhecer e produzir informação pública para controlar a administração e com ela contribuir.

5. Inovação e desenvolvimento

* Fortalecer negócios em tecnologia abertas;

* Incentivar a adoção, o desenvolvimento e a pesquisa de Software Livre nas instituições de ensino e pesquisa.

6. Sinergia com outras ações de governo

* Garantir a Sinergia com outras ações e programas de governo, trabalhando pelo desenvolvimento e combate a miséria.

Entretanto, há ainda um outro aspecto que pode ser trazido à discussão quando o tema é software: os softwares proprietários, via de regra, tem um custo por usuário muito inferior ao custo efetivamente pago por cada cópia, ou cada licença. Isso já foi demonstrado em outros artigos. Então, se uma licença ou cópia custa 10,00 e é vendido por 100,00, por exemplo, mesmo que a empresa desenvolvedora fique com um lucro estratosférico de 500% (fique com 60,00) ainda restarão 40,00 para agraciar simpatizantes com viagens e benesses. Seria forçar muito a imaginação?

O Governo do Estado do Espírito Santo, por seu lado, atualizou um decreto do governo anterior, atualização publicada no Diário Oficial de 09/04/2012 como decreto n.2991-R, no qual “Institui a Política Estadual de Tecnologia da Informação e Comunicação – PETI, no âmbito do Poder Executivo do Estado”. O texto desse Decreto é longo, embora com muitos rodeios, e

no seu Art. 6 fala que a PETI possui as seguintes diretrizes específicas: I - Plano Diretor de TIC do Poder Executivo; II - Arquitetura de Referência e Padrões; III - Prestação de Serviços; IV - Processo de Aquisição e Gestão de Contratos; V - Aprovação de Projetos; VI - Segurança da Informação; VII - Desenvolvimento de Competências em TIC; VIII - Uso de Software Livre.

A diretriz “Uso de Software Livre” detalha:

“a) os gestores de TIC identificarão oportunidades para utilização de softwares livres, adotando o seu uso quando viável;

b) a adoção de software livre, da mesma forma que para os outros tipos de software, deve obrigatoriamente considerar características e requisitos compatíveis com os adotados para padronização de tecnologias pelo Governo;

c) para adoção do uso de software livre é recomendada análise prévia do custo-benefício, considerando não somente o das licenças, mas sim o total de adoção da solução de software (incluindo customização, implantação, treinamento, suporte, entre outros fatores).”

Tentando considerar a boa intenção do governo do estado, pode-se extrair do texto acima que o governo não proíbe o uso do software livre, mas recomenda a análise prévia do seu custo-benefício... O texto fala por si. 🙏

MILTON JOSÉ LYRIO SIMONETTI é

engenheiro elétrico e especialista em Análise de Sistemas. Já atuou como analista de sistemas na Secretaria de Saúde do Estado e posteriormente no DER onde exerce a função. Já ministrou aulas na UVV, UFES, Univila e proferido palestras em diversos eventos.



Formatos Livres

por *Gilberto Sudré*

Quando falamos de um ambiente completamente livre, a princípio todas as atenções voltam-se para os aplicativos. É fácil imaginar o motivo pois eles são a parte do computador visível para os usuários. Mas, não podemos deixar de observar que existem alguns detalhes que, quando olhados

cuidadosamente, passam a fazer parte central da discussão sobre o uso democrático e irrestrito dos computadores.

Quando usamos uma aplicação qualquer, por exemplo um editor de textos, ao terminar o trabalho precisamos salvar o material digitado em um arquivo.

Neste momento, o usuário precisa escolher como gravar as informações no disco, ou seja, qual a organização das informações armazenadas. Esta escolha determina onde (equipamentos) e como (aplicações) o arquivo será utilizado posteriormente.

Os formatos .DOC, .DOCX, .PPT, PPTX, .XLS, Windows Media Vídeo e Windows Media Audio da Microsoft assim como outros são “proprietários” e seus “donos” os mantêm sob controle e guardados a sete chaves. O objetivo é eliminar a concorrência e manter a dependência dos clientes. Uma vez que os utilize, você será obrigado a fazer uso também de seus aplicativos

para modificação ou impressão.

Se pensamos em liberdade de escolha, e isto inclui plataformas de hardware e aplicativos, certamente estes formatos proprietários não são os mais adequados, pois nos obrigam a também utilizar aplicativos do mesmo modelo (proprietário).

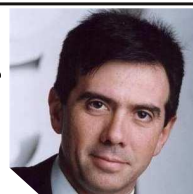
Hoje já podemos escolher padrões abertos como o XML para textos e dados, .ODT (texto), .ODS (planilha) e .ODP (apresentação), o Theora para vídeo e o Ogg para áudio entre outros.

Pensando em igualdade de condições para a concorrência, muito saudável por sinal, isto também vale para compras governamentais

pois se está estabelecido que o formato utilizado é proprietário por consequência já está definido o único fabricante que poderá fornecer os aplicativos que manipularão estes arquivos. 🇧🇷

GILBERTO SUDRÉ é professor, consultor e pesquisador da área de Segurança da Informação.

Comentarista de Tecnologia da Rádio CBN. Articulista do jornal A Gazeta e portal iMasters. Autor dos livros Antenado na Tecnologia, Redes de Computadores e Internet: O encontro de 2 mundos.



DESIGN

[HTTP://CADUNICO.ART.BR](http://cadunico.art.br)



[HTTP://CADUNICO.ART.BR](http://cadunico.art.br)

Se o seu evento esta precisando de uma identidade visual profissional, deixe isto comigo! Sou designer a mais de 15 anos com vasta experiência em promover uma uniformidade gráfica em mídias de eventos. Além de criar toda a campanha, toda a parte de controle de impressão também fica sob minha responsabilidade, graças as excelentes parcerias que fazem com que o custo final fique mais em conta. Além do fato de garantir uma produção final de melhor qualidade! Seguem abaixo nossos combos especiais.

Combo

#1



- * 01 Banner 180 X 90 cm
- * 2500 Filipeta 10 x 21 cm 120g 4 cores na frente e atrás Couchê 120g
- * 50 Crachás de PVC 8,5 x 5,4 cm 4 cores na frente com presilha
- * 2000 Cartazes Couchê 90g 31x44 cm 4 cores na frente

R\$ 1.665,00

Frete não
incluso.

Combo

#2



- * 02 Banners 180 X 90 cm
- * 5000 Filipeta 10 x 21 cm 120g 4 cores na frente e atrás Couchê 120g
- * 100 Crachás de PVC 8,5 x 5,4 cm 4 cores na frente com presilha
- * 5000 Cartazes Couchê 90g 31x44 cm 4 cores na frente
- * 100 Bottons 4,5 cm

R\$ 3.165,00

Frete não
incluso.

Combo

#3



- * 04 Banners 180 X 90 cm
- * 10000 Filipeta 10 x 21 cm 120g 4 cores na frente e atrás Couchê 120g
- * 200 Crachás de PVC 8,5 x 5,4 cm 4 cores na frente com presilha
- * 10000 Cartazes Couchê 90g 31x44 cm 4 cores na frente
- * 200 Bottons 4,5 cm
- * 1000 Pulseiras de identificação
- * 1000 Pasta Com Bolsa Couchê 250g Uv Total

R\$8.050,00

Frete não
incluso.

* O valor de cada combo só é cobrado após a aprovação da arte.

* Entregamos para todo brasil com código de rastreio, junto com a nota fiscal do design e das gráficas.

Caso queira uma campanha personalizada, entre em contato que farei um orçamento sem compromisso.

Maiores detalhes e solicitação de orçamento: <http://cadunico.art.br>

Entrevista: Linus Torvalds

"Hoje eu não leio mais código"

Publicada no final do ano passado, esta entrevista apresenta um pouco de como Linus Torvalds tem desempenhado atualmente suas atividades relacionadas ao desenvolvimento do kernel, os principais eventos relacionados a desenvolvimento de código, bem como sobre a colaboração de empresas como o Google para o desenvolvimento do kernel linux. Esta entrevista, na verdade, foi uma “conversa entre amigos”, pois Glyn e Linus já se conhecem desde 1996.

por Glyn Moody - Tradução: David Jourdain

TIVE A SORTE DE ENTREVISTAR LINUS

bem no início da história do Linux - em 1996, quando ele ainda estava vivendo em Helsinque, e você pode ler os frutos dessa conversa nesta edição da **Wired**. Foi um momento importante para ele, tanto sob o ponto de vista pessoal, pois sua primeira filha havia recém-nascido, mas também sob o ponto de vista de sua carreira. Ele estava prestes a juntar-se ao desenvolvimento de projetos para processadores na empresa Transmeta. Comparado com um “jogo de xadrez”, podemos dizer que este foi um movimento que não deu muito certo, mas que o levou a se mudar para os Estados Unidos, onde permanece até hoje.

Isso fez com que suas viagens a Europa se tornassem raras. Por isso, eu aproveitei o fato de que ele esteve na **LinuxCon Europe 2012** em Barcelona, e fui para entrevistá-lo novamente, para rever os momentos-chave para o kernel do Linux e sua comunidade desde nossa última conversa.

GLYN Olhando para a última década e meia, o que você vê como os eventos-chave no desenvolvimento do kernel?

LINUS Uma coisa importante para mim é sobre todo o trabalho que fizemos no que tange a escalabilidade. Nós saímos do ponto em que estávamos satisfeitos com o suporte a 2 ou 4 CPUs para o ponto onde, basicamente, você pode ter 4000 CPUs. E se você não puder promover a escalabilidade perfeitamente, na maioria das vezes não será o kernel o gargalo. Se o seu trabalho permitir, a escalabilidade poderá realmente ser muito bem executada. E para que chegássemos a este ponto, foi necessário muito esforço.

Particularmente, a SGI trabalhou muito gerando patches para portar algumas centenas de processadores. Contudo, seus patches iniciais não poderiam apenas ser

mesclados ao código-fonte. Não havia nenhuma maneira de podermos utilizar este trabalho feito pela SGI em um PC normal, porque eles adicionaram toda uma infraestrutura para trabalhar com milhares de processadores, e isso era muito distante da realidade, quando você tem apenas acesso a alguns núcleos.

Durante um período, eu tive medo de que teríamos um kernel de alta performance para grandes computadores e o código-fonte acabaria sendo separado do kernel normal. Muitos desenvolvedores trabalhavam duro apenas para ter certeza de que nós tínhamos uma base de código bem limpa, onde você pode dizer no momento da compilação: *“hey, eu quero que o kernel que trabalha para 4.000 CPUs, que ele gere o código para isso e, ao mesmo tempo, se eu disser que não, eu quero que o kernel trabalhe apenas com duas CPUs, e o mesmo código fonte tem que compilar.”*

Olhando lá atrás na linha do tempo, ter esta preocupação logo no início foi muito importante, pois isso fez com que o código-fonte realmente ficasse muito melhor. Todo o esforço que a SGI e outros desenvolvedores fizeram para a unificação destes códigos ao fonte do kernel não funcionariam para uma centena de CPUs distintas, então tivemos que limpá-lo para que ele funcionasse. E todo este trabalho fez com que o código atual do kernel ficasse mais eficiente e melhor de ser mantido. Atualmente, em nosso ambiente de trabalho, é comum termos entre 8 a 16 CPUs, o quê poderia ser um problema. Agora, é como brincadeira de criança.

Mas outras coisas também ocorreram. No outro lado, passamos alguns anos tentando desenvolver soluções para economia de energia, pois as pessoas sabiam que seus telefones tinham verdadeiras gambiarras, especialmente para a arquitetura ARM. Passamos anos fazendo código para gerenciamento de energia,

fazendo quase sempre a mesma coisa, em vez de ter apenas esses cortes especializados de gerenciamento de energia para ARM e os poucos dispositivos que as pessoas se preocupavam para seus celulares, tentamos gerar correções através de todo o kernel. E isso levou uns cinco anos para que conseguíssemos que o nosso trabalho de gerenciamento de energia pudesse atingir a todo o espectro do desenvolvimento.

Quando você adiciona suporte a um novo dispositivo, normalmente isso não tem impacto no resto do kernel. Mas gerenciamento de energia é uma das coisas que impacta sobre todos os milhares de dispositivos que temos no kernel. Impacta principalmente nas funcionalidades elementares, no processo de desligamento de CPUs, nos escalonadores, na VM, impacta em tudo. E isso não apenas afeta tudo mas tem potencial para quebrar tudo, o que torna toda a situação muito dolorosa. Por isso, usamos muito tempo apenas dando dois passos para frente e um passo para trás, pois fazíamos uma melhoria que era uma clara, mas que também quebrava máquinas. E por isso tínhamos que dar um passo atrás apenas para corrigir as máquinas que nós mesmos quebrávamos.

Sendo realista, em cada release, a maior parte é apenas desenvolvimento de driver. O quê é um pouco chato, porque não há nada fundamentalmente interessante em fazer um driver, pois é só para oferecer suporte a um outro chipset ou alguma outra coisa, e ao mesmo tempo, isso é o nosso *“feijão com arroz”* do kernel. Mais da metade do kernel são apenas drivers. Assim, todas as grandes e emocionantes coisas inteligentes que nós fazemos, no final, não são nada quando comparado com todo o trabalho que acabamos de fazer para oferecer suporte a um novo hardware.



GLYN Quais são as principais mudanças na arquitetura, para suportar um novo hardware?

LINUS As trilhas de código referentes ao funcionamento do suporte a USB já foram reescritas várias vezes, sempre que alguma nova funcionalidade é inserida. Neste momento, percebemos que esta funcionalidade ainda não foi prevista por nós e, por isso, temos que reescrever. Senão a funcionalidade simplesmente não vai funcionar. Assim foi com o USB 3.0, que precisou de um novo suporte para os controladores, e verificamos que o quê tínhamos no código não seria o suficiente para trabalhar com as diferentes versões. E esta situação não se trata apenas com USB. É assim também com as PCIs, que logo serão convertidas em PCIe, e tudo isso funcionando via hotplug.

E essa é outra grande diferença entre um Linux tradicional e um Unix tradicional. Você tem uma estação de trabalho Unix e depois que você a inicializa, isso não muda mais! Você não adiciona novos dispositivos. Agora, as pessoas podem adicionar diversos dispositivos via USB. Mas, sendo realista, nem sempre foi assim. E por conta disso, toda uma gama de suporte teve que ser adicionada e, por conta disso, tivemos

que aplicar todas essas mudanças de infraestrutura fundamentais ao código.

GLYN E como anda a comunidade do kernel? Como ela tem evoluído?

LINUS Ela costumava ser restrita e até um pouco bajuladora. Eu não sei quando a mudança ocorreu, mas costumava ser eu e talvez mais 50 desenvolvedores - e nunca houve uma hierarquia profunda entre nós. Hoje em dia, as correções que chegam a mim, às vezes passam por quatro níveis de pessoas. Nós liberamos um novo release a cada três meses e em cada lançamento temos aproximadamente 1.000 pessoas envolvidas. E 500 das 1.000 pessoas basicamente enviam correções de apenas uma única linha para algo realmente trivial - e é assim que alguns colaboram, e alguns deles nunca mais farão outra colaboração, e tudo bem em ser assim. Mas quando você tem milhares de pessoas envolvidas, especialmente quando alguns deles são estes que estão guiados apenas para dar "um tiro" (para gerar apenas uma correção), você não pode contar comigo apenas para ficar pegando estes patches individualmente de todos. Eu não teria tempo para interagir com eles.

Algumas pessoas são especializadas em drivers, e conhecem outras que sabem que são especialistas nessa área particular para drivers e que são responsáveis dentro do desenvolvimento, e por isso eles interagem entre si, escrevem seus drivers e enviam os patches. Eu certamente verei o patch durante este processo, onde raramente existe quatro "camadas" de pessoas, mas frequentemente ao menos duas pessoas estão no meio, entre mim e o desenvolvedor deste patch.

GLYN Então, qual o impacto que isso tem sobre o seu papel atual, no desenvolvimento do kernel?

LINUS Bem, o fato mais relevante é que

hoje eu não leio mais código. Quando um patch já passou por duas pessoas, naquele momento, eu poderia olhar para o patch e dizer: "não, todo seu trabalho foi uma perda de tempo", e eu teria que gerenciar esta situação a um nível bem particular e, francamente, eu não quero fazer isso, pois eu não tenho a capacidade para fazer isso.

Por isso, na maioria das vezes, quando se trata dos responsáveis por subsistemas principais, eu confio neles pois tenho trabalhado com eles por 5, 10, 15 anos. Então eu nem sequer olho para o código. Eles me dizem quais são as mudanças e dão uma visão geral de alto nível. Dependendo da pessoa, podem ser cinco linhas de texto dizendo que isso é mais ou menos o que mudou, e então eles me dão um **diffstat**^[1], que apenas diz que 15 linhas foram alteradas neste arquivo, e 25 linhas foram alteradas naquele arquivo ou poderia ser um diffstat com algumas centenas de linhas, porque há algumas centenas de arquivos que foram alterados. Mas eu nem sequer vejo o código em si, eu apenas digo: "OK, as mudanças aconteceram nesses arquivos, e, a propósito, eu confio em você para mudar esses arquivos, de modo que eu creio que está bom". E então eu acabo dizendo: "Vou aplicar o patch".

GLYN Então, qual é o seu papel agora?

LINUS Em grande parte eu estou na gestão de pessoas. Não no sentido logístico, já que eu não pago ninguém, mas também não tenho que me preocupar que eles tenham acesso a hardware ou algo assim. Em grande parte o que acontece é que eu me envolvo quando as pessoas começam a discutir e ocorre atrito entre elas, ou quando os erros acontecem.

Erros acontecem o tempo todo, mas muitas vezes as pessoas não sabem enviar devidamente um relatório de erros. Elas

enviam o relatório de erros para a lista de discussão Linux Kernel – onde ninguém é capaz de ler muita coisa. Depois que as pessoas descobrem isso na lista de discussão do kernel, muitas vezes começam a me atacar, dizendo: “Hey, esta máquina não funciona mais para mim”. E como eu no primeiro momento nem sequer li o código mas sei quem está no comando, eu acabo sendo o ponto de conexão para os relatórios de bugs e para as solicitações de mudanças reais. Isso é tudo que eu tenho feito, dia a dia: leio e encaminho e-mails. E tudo bem! Eu gosto de fazer isso, mas é bem diferente do que eu fazia antes.

GLYN Então, isso significa que pode haver espaço para você escrever uma outra ferramenta como Git, mas para gestão de pessoas, e não de código?

LINUS Eu não acho que isso vá acontecer. Pode até haver algum ferramental, mas sendo bem realista, a maioria das minhas ações tendem a ser sobre interação humana. Então nós temos ferramentas para descobrir quem é o responsável, para dizer que sabemos que o problema acontece nesta ou naquela área do código, assim como quem mexeu no código na última vez e quem é o mantenedor desse subsistema. Existem tantas pessoas envolvidas que tentam manter o controle de código, e ter alguma automação, por si só, não fazem as coisas funcionarem. Mas ao mesmo tempo, a maioria do trabalho é de interação, e pessoas diferentes trabalham de formas diferentes, assim como ter um processo muito automatizado pode ser realmente doloroso para todos.

Mas estamos indo muito bem! O tipo de pontos de desconforto que tínhamos há dez anos simplesmente não existem mais. E isso é em grande parte porque estamos utilizando esta hierarquia plana. Nós concertamos nossas ferramentas e concertamos nossos fluxos de trabalho. E não é



só comigo, isso acontece através de todo o kernel e não há uma única pessoa que esteja no caminho de qualquer fluxo de trabalho específico.

Eu recebo uma quantidade razoável de e-mails, mas não fico sobrecarregado. Eu adoro ler e-mails no meu celular quando eu viajo, por exemplo. Mesmo durante os intervalos, eu leio e-mails no meu celular, pois 90% deles eu só leio para minha informação e depois arquivo. Eu não preciso fazer nada, pois estava como cópia por haver algum problema acontecendo. Preciso estar ciente disso, mas não preciso fazer nada. Então posso fazer 90% do meu trabalho durante as viagens, mesmo sem ter um computador. À noite, quando volto para o quarto de hotel, eu passo o olho nos outros 10%.

GLYN Há 16 anos, você disse que estava impulsionado principalmente pelo que o mundo lá fora estava pedindo, dado o enorme interesse em celulares e tablets. Qual tem sido o impacto destas demandas sobre o desenvolvimento do kernel?

LINUS No que tange aos tablets, o maior problema tende a ser a gestão de energia, principalmente porque eles são maiores do que os telefones. Têm baterias maiores, mas, por outro lado, as pessoas esperam

que tenham uma vida mais longa, mesmo tendo telas maiores e que se utilizam mais da bateria. Por isso, do lado do kernel, um tablet a partir da perspectiva de hardware e da perspectiva de uso é em grande parte a mesma coisa que um telefone, e isso é algo que nós já sabemos como fazer, principalmente por causa do Android.

A interface de usuário para tablets tem sido a principal “dor de cabeça” - mas que está longe o suficiente do kernel. Em um telefone, o navegador não é um navegador completo - eles costumavam ter os navegadores móveis; sobre os tablets, as pessoas realmente esperam ter um navegador completo - você tem que ser capaz de clicar nesse link pequeno. Assim, a maioria das questões no tablet tem sido no espaço do usuário. Nós tivemos um monte de problemas no kernel durante o desenvolvimento para telefones, mas com os tablets nós recebemos quase que todas as soluções de graça.

GLYN E sobre a computação em nuvem: Quê impacto produziu sobre o kernel?

LINUS O maior impacto tem sido que, mesmo no lado do servidor (mas especialmente quando se trata de computação em nuvem), as pessoas tornaram-se muito mais conscientes do consumo de energia. Todo o trabalho originalmente dirigido para gerenciamento de energia costumava ser dirigido para os celulares, e apenas nos últimos 3-4 anos, as pessoas envolvidas com servidores estão mais conscientes sobre consumo de energia. Muitos destes servidores estão juntos, muitas vezes eles têm o pico de uso bem elevado. Se você olhar para a Amazon, a ordem de magnitude de uso nos períodos de pico é bem maior do que o seu uso regular quando ocioso. Por exemplo, apenas o serviço de vendas da Amazon, no final de novembro e início de dezembro, um mês antes do Natal, executa operações



de negócios tanto quanto fazem nos outros dez meses do ano. O ponto é que eles têm a escala de toda a infraestrutura de hardware baseada para o pico de uso, que é bem maior que o do resto do ano. Por isso, eles só usam um décimo da capacidade instalada. Portanto, ser capaz de não usar energia o tempo todo é muito importante, pois o custo da eletricidade é grande nesses provedores de grandes servidores.

GLYN: Tem gente da Amazon envolvida diretamente com o trabalho de kernel?

LINUS A Amazon não é o maior exemplo. O Google é provavelmente um melhor exemplo, pois eles realmente têm um monte de engenheiros de kernel trabalhando para eles. Na maioria das vezes, o que eles precisam é feito pelo seu próprio pessoal. Eu acho que a Amazon tem tido componentes mais padronizados. Na verdade, eles mudaram a forma de como constroem hardware - eles agora têm o seu próprio projeto de referência de hardware. Usaram para comprar hardware da HP e da Dell, mas acontece que quando você compra 10.000 máquinas, em algum ponto, é mais simples desenhar as máquinas por si mesmo, e ir diretamente

para os fabricantes de equipamentos e dizer: *“Eu quero essa máquina, como está aqui”*. Mas eles só começaram a fazer isso recentemente.

Eu não sei se a Amazon apenas está um pouco atrasada neste ponto ou se o Google é que é mais orientado para tecnologia. Amazon tem trabalhado mais no espaço do usuário, e eles usaram um kernel bastante normal. Já o Google tem trabalhado mais do lado do kernel, por isso eles fizeram seus próprios sistemas de arquivos. Eles costumavam fazer seus próprios drivers para seus discos rígidos porque tinham alguns requisitos especiais.

GLYN Quão útil tem sido para você o trabalho do Google no kernel?

LINUS Por alguns anos - entre cinco ou dez anos atrás - o Google costumava ser este buraco negro. Eles contratavam engenheiros do kernel que desapareciam completamente da face da terra. Eles trabalhavam dentro do Google, e ninguém jamais ouvia falar sobre eles novamente, porque eles faziam apenas o material específico para o Google, e o Google realmente não retornava muita coisa.



Hoje eu não leio mais código ”

Contudo, isso melhorou muito, provavelmente porque o Google ficou um longo tempo trabalhando em nossas versões da 2.4. Eles permaneceram aí durante anos, porque eles tinham feito muitas modificações internas para seus hardwares especializados, e simplesmente atualizar o kernel seria um grande problema para eles. E, em parte, também porque eles tem este projeto do Android e, por isso, eles realmente querem estar mais ativos no todo do desenvolvimento.

Agora eles estão de forma mais ativa e os desenvolvedores não desaparecem mais. Isso fez com que o kernel ficasse melhor, a ponto de muitos dos problemas iniciais se tornassem detalhes, ao invés de se transformarem em enormes buracos. Eles agora tem atuado mais o menos assim: *“OK, nós podemos realmente usar o kernel padrão e, em seguida, fazemos nossos pequenos ajustes em cima”*, em vez de fazer essas grandes cirurgias apenas para fazê-la funcionar na infraestrutura deles.

GLYN Finalmente, você diz que passa a maior parte do seu tempo respondendo e-mails: Como alguém que sempre me pareceu um hacker por excelência, isso não lhe preocupa?

LINUS Eu não diria que me preocupa. Acabo não fazendo programação tanto quanto, às vezes, eu gostaria. Por outro lado, tem alguns tipos de programação que eu não quero fazer mais. Quando eu tinha 20 anos eu gostava de fazer drivers de dispositivo. Se eu nunca mais tiver que fazer um único driver de dispositivo na minha vida de novo, eu vou ficar muito feliz. Este é o tipo de dor de cabeça que eu prefiro não ter mais.

Eu realmente gostei de fazer Git, foi muito divertido. Quando comecei o projeto, comecei a fazer a programação no espaço do usuário, o que eu não tinha feito há 15

anos, e foi como: “Uau, isso é tão fácil”. Não preciso me preocupar com tantas coisas; eu tenho uma pilha infinita; o **malloc**^[2] simplesmente funciona! Mas, no espaço do kernel, você tem que se preocupar com bloqueio, tem que se preocupar com segurança, tem que se preocupar com o hardware. Fazer o Git foi um alívio. Mas depois acabou ficando chato.

O outro projeto que eu ainda estou envolvido é o Subsurface, que é um programa de monitoria de mergulho. Também tivemos um grande problema a algum tempo atrás com o site *kernel.org*. Foi muito doloroso para os mantenedores, e até o FBI se envolveu para descobrir o que aconteceu. Durante dois meses, não tivemos quase nenhum desenvolvimento do kernel. Bem, as pessoas ainda estavam desenvolvendo para o kernel, mas o site principal, onde todo mundo se reunia, estava inativo, e um monte de desenvolvedores core do kernel passou muito tempo verificando se alguém tinha realmente quebrado algo em suas máquinas. As pessoas ficaram um pouco paranoicas.

Assim, por um par de meses o meu trabalho principal, que era integrar o trabalho de outras pessoas, basicamente ficou de lado, porque a integração do nosso

site havia se perdido. Então, eu ainda acabo fazendo programação, mas no final, eu sempre volto para o kernel. 🇧🇷

Definições:

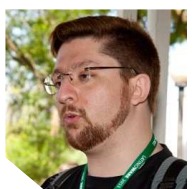
[1] Diffstat lê o resultado do comando diff e mostra um histograma das inserções, eliminações e modificações em cada arquivo. Diffstat é comumente usado para fornecer um resumo das mudanças em grandes arquivos de patches complexos.

[2] É um termo da computação que designa uma função da biblioteca padrão das linguagens de programação C e C++, para requisitar alocação dinâmica de memória. É uma forma abreviada de escrever memory allocation (alocação de memória).

GLYN MOODY é escritor, blogueiro e palestrante. Tem escrito, dado palestras e feito consultoria sobre o uso comercial da internet desde o início de 1994, e sobre código aberto desde 1995. Seu livro, "Rebel Code: Linux and the Open Source Revolution", é uma história detalhada do software livre e foi publicado em 2001.



DAVID JOURDAIN é membro fundador da The Document Foundation - TDF. Moderador das listas da TDF em língua portuguesa. Tem toda a sua formação na área da Computação. Mais de 12 anos "mexendo" no Kernel Linux. Já foi professor universitário em diversos países, ensinando sobre Kernel Linux e como organizar grupos de desenvolvedores de software livre em Universidades.



Imagine o fórum oficial da Espírito Livre na sua cidade!

Entre em contato conosco!

Fórum espírito livre

Por que personalizar?

por Camilo de Souza Teixeira

H

á cinco anos, mais ou menos, eu entrei em uma loja (não lembro do que e nem para que) e notei que a interface do sistema operacional usado no PC da loja, que lembrava o Windows XP, era personalizada!

Ou seja, a interface não tinha somente o plano de fundo com o logo da empresa, mas os ícones usados também eram referentes à empresa: o Menu Iniciar tinha o formato do logo da empresa; a Lixeira era uma caneca (também com o logo da empresa); na Área de trabalho, além da Lixeira, só haviam dois ícones: um descrito como Cliente e outro como Administração; a barra de ferramentas era de cor cinza metálica com o nome da empresa escrito em alto-relevo na cor vermelha; só o relógio parecia padrão do Windows®!

Na época achei, que se tratava do XP ou até do 98 modificado. Inclusive pesquisei na internet para saber era como era feito, mas não encontrei nada.

Mas, quando tive acesso ao meu primeiro Linux - Ubuntu 10.04, comprado em banca de jornal a uns 2 anos, me dei conta, por causa da facilidade de personalização, que talvez aquele sistema operacional que vi anos atrás fosse Linux e não Windows. Ou poderia realmente ser Windows?

A certeza eu só vou ter se voltar àquela loja - que não lembro mais onde era, para perguntar!

Mas está aí uma questão de segurança que geralmente não pensamos: a impossibilidade de identificar visualmente o sistema operacional de agências comerciais, públicas e bancárias!

Pois acredito que algumas pessoas que aplicam golpes nestas agências via internet, tais como o envio de vírus ou invasão do sistema, teriam como necessidade descobrir primeiramente que sistema operacional é utilizado pela agência, pois assim a invasão seria mais fácil, certo?

E para isso alguns desses criminosos chegam a ir até suas “agências vítimas” para identificar o sistema operacional usado. Acredito que a maioria dos criminosos virtuais ou fornecedores de informações para criminosos ajam dessa forma. Como diz o ditado: “informação é poder”, e no caso do mundo digital esse ditado mostra realmente o seu potencial, e dessa maneira o criminoso pode se preparar e estabelecer sua estratégia de ataque mais facilmente e as chances dele ser detectado são muito pequenas. Agora, sem está informação o criminoso correrá mais riscos de ser detectado e rastreado e, conseqüentemente, preso!

Pois a personalização da qual falo, deixaria o sistema operacional realmente com sua cara, além de descaracterizar totalmente a interface padrão do sistema. Imagine, por exemplo, a “sua cabeça” como o Menu Iniciar ou uma “casinha de cachorro” como lixeira - sem o cachorro quando a lixeira está vazia e com o

cachorro quando a lixeira está cheia. Isso tornaria inviável a identificação visual do sistema.

Com a personalização da interface, há a possibilidade de colaboração mutua, no quesito segurança, entre as plataformas (mesmo sem um acordo formal entre elas) usadas pelas agências, uma vez que a identificação do sistema é dificultada, torna-se difícil descobrir quais ferramentas ou cuidados serão necessários para a invasão.

E a meu ver, no caso do Linux, há uma dificuldade ainda maior que no Windows para se invadir o sistema, pois além da distribuição, o criminoso tem que saber a versão do mesmo, se quiser ter algum êxito.

Isso porque o kernel do Linux costuma variar de versão para versão, até mesmo entre a mesma distribuição de Linux. E mesmo que o código fonte seja livre no Linux, ou seja, qualquer um pode ter acesso, as brechas e falhas são diferentes de versão para versão. Com o isso o conhecimento para invadir também tem que ser diferente!

Pois, invadir um sistema é como construir tuneis. Você não pode usar a mesma técnica que constrói tuneis nas entranhas de uma montanha para construir um túnel que passa embaixo de um rio!

Por isso, o modo operante que o criminoso usa para invadir o Windows não funciona no Linux e, da mesma forma, o modo operante para invadir o Ubuntu não funciona no Kurumin e por aí vai!

E é aí que mora a necessidade do criminoso em saber qual sistema operacional a vítima está usando, pois com tentativa e erro, ele corre o risco de ser preso ou de, no mínimo, alertar sua presa, fazendo com que está se previna contra ele ou até mesmo “arme uma arapuca”. Ou seja, é como “puxar o rabo da onça”, você pode conseguir, mas as chances de ser pego por ela são grandes.

E a meu ver, no caso do Linux, há uma dificuldade ainda maior que no Windows para se invadir o sistema, pois além da distribuição, o criminoso tem que saber a versão do mesmo, se quiser ter algum êxito"

Lamento que, aqui no Brasil, o Linux não tenha ganhado tanta força quanto deveria, pois só a questão desta personalização do sistema que impede sua identificação visual, geraria muitos empregos. Esse tipo de serviço não seria procurado somente por estas agências, mas até por usuários domésticos como eu.

Existem os temas disponíveis na internet para baixarmos, mas estou falando de temas de “propriedade pessoal” e não de temas que pertencem a uma empresa e possuem código fonte fechado. Ou seja, temas que refletem a sua personalidade (afinal eu adoraria ter um sistema com visual - pelo menos os ícones, inspirados nos meus desenhos, pois adoro desenhar). Assim a turma que sabe fazer ícones e temas, mas que estão desempregados, poderiam ter uma oportunidade de renda, além das empresas que (como a Canonical) disponibilizam distribuições Linux de maneira gratuita, poderiam oferecer este serviço de personalização do sistema de maneira comercial. Empresas, pessoas e outras entidades que se interessassem por esse tipo de serviço, poderiam manter-se financeiramente e ainda assim manteriam a suas distribuições de sistema operacional

de forma gratuita ao mesmo tempo em que geram mais vagas de trabalho no mercado.

Afinal, as possibilidades de geração de emprego e renda na área da informática são inimagináveis, principalmente no Linux. Estou falando de empregos e geração de renda de forma legal, ou seja, que não infringem a lei, tais como: a personalização de sistemas (já mencionado), a instalação e a configuração de sistemas e programas, a adaptação de sistemas para pessoas especiais, configuração de redes, etc...

Em um país cuja maioria, da população ainda está desempregada, está é uma questão que devia ser mais defendida. Não acham? 🙏

CAMILO DE SOUZA TEIXEIRA

é formado em pedagogia e iniciante no mundo maravilhoso mundo do sistema operacional GNU/Linux.





Rede social e você no Facebook

por Albino Biasutti neto

Olá, estava pensando em como deletar uma conta da empresa na rede social do Mark Zuckerberg. Eu já tentei várias vezes. Impressionante.

Primeiro devemos pensar para que serve uma rede social, qual é o *MEU (SEU)* objetivo nela, o impacto que ela vai ocasionar para mim e na minha vida. Esses são os principais pontos que devemos analisar. Depois ler toda a política de privacidade da mesma, e o que ela nos beneficia e privilegia.

Rede Social para marketing, publicidade e anúncios, são seus enfoques primordiais para um site, blog, projeto, comunidade, movimento, ativismos, entusiastas e muitos outros substantivos que podemos acrescentar. Qual será a sua finalidade para ela? Obter "*amigos*" virtuais, reencontros, amores, fazer fofoca, vasculhar a vida das outras pessoas, ter ciúmes das "*fotos*" que as pessoas postam, **inveja**, são tantos né.

O impacto que ela vai ocasionar na minha vida

A princípio você será visto pelo *MUNDO*, todas as pessoas que possuem acesso na internet, de Nova Iorque - EUA, até Madagascar - África (não desmerecendo o país/cidade, por favor!). Sua vida será exposta, por quê? Você posta uma mensagem ou foto sua pessoal, um "amigo" seu compartilha, e o "amigo" do "amigo" compartilha e assim sucessivamente, lembrando que, os "amigos" podem fazer esse compartilhamento público e não só para seus "amigos" da rede. Minha vida será prejudicada? Talvez sim ou não. A resposta será **não** se você adicionar as pessoas que tem confiança. Será **sim** a partir do momento que você criou uma conta na rede, está exposto para que, as principais forças de segurança, saibam de seus movimentos: FBI, CIA, Polícia Federal e outros. E das piores situação, pessoas maliciosas que desejam fazer o mau a você.

O importante é saber que está sendo vigiado(a) em todos os locais para onde anda, passeia, curte, suas maiores atividades de lazer, seus prazeres, ideias, lugares que gosta e não gosta, suas críticas e **TUDO**.

A política de privacidade, puts.. o que é isso? São os termos que você está de acordo ou não em assumir depois de registrar na rede social, estiver de acordo, registre. São as linhas principais de tudo que for fazer na internet, ler o que os sites pedem para registro. Por quê? Muitos não se responsabilizam por nada do que você faz. Qualquer problema é de sua responsabilidade, sem aval para ela. Se algum órgão de segurança solicita seus dados, obrigatoriamente terá que liberar, sob mandado judicial. Em resumo: não te beneficia em nada.

Retornando ao primeiro parágrafo do texto, quando deletava minha conta do facebook, e quando retornava, meus dados

estavam intactos por lá. Nisso andei vasculhando as ferramentas que a rede disponibiliza, encontrei os registros de atividades, bloqueio de apps, privacidade(s), ligação com outros sites (não vou falar mais, se você possui conta deve olhar).

Fui no *Registro de Atividades*, acreditem tudo que eu fiz (ou fazia) na rede social ou ligação com sites (e outros), estava registro nela. Ou seja, ela sabe demais da sua própria vida do que você mesmo (paradoxo). É de se impressionar mesmo. Então como deletar por completo e sair "limpo" de lá, acredito que deletando todos seus registros e depois sua conta e não voltar nela mais. Estou nesse processo no Facebook. E as outras redes sociais? Estou analisando, deletando alguns registros desnecessários pela web, e partindo apenas para o identi.ca e talvez o Twitter.

Abraços! 🇧🇷

Licença Copyright

Esta obra está licenciada sob a Licença Creative Commons CC BY-SA (Attribution ShareAlike, ou Atribuição e Compartilhamento pela mesma licença) 2.0 Unported. Para ver uma cópia dessa licença, visite

<http://creativecommons.org/licenses/by-nc-sa/3.0> ou envie uma carta ao Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

ALBINO BIASUTTI NETO

*é um apaixonado por Software Livre e Linux, e também é um Geek assumido.
Site: www.albino.ws.
Siga-me: identi.ca/bino28*



Salvem os Hackers

por *Alexandre Oliva*

É um sarro como repórteres são atraídos por palestras com “hacker” no título; parecem abelhas rondando copos de refrigerante dietético. Por mais que tentem, não vão encontrar ali nem a “pimenta” nem o açúcar que esperavam. Os padrões desses repórteres acabam “vítimas” da própria campanha de difamação que lançaram contra nós, hackers. Por isso mesmo, nada mais justo que usarmos técnicas de artes marciais para direcionar a força do oponente contra si mesmo, né?

Pois então, a primeira coisa que se esclarece ao “abelhudo” é que hacker não é criminoso virtual; isso é uma distorção para nos demonizar. Hackers são seres curiosos, amantes do conhecimento, da descoberta, da engenhosidade. Ganham pontos entre seus pares não por quanto conseguem extorquir dos outros, mas pelo brilho do conhecimento que compartilham. Por que alguém demonizaria gente assim, que compartilha conhecimento, por vezes através de artifícios engenhosos?

Para entender, basta pensar em diversos hackers perseguidos e demonizados pelas elites ao longo dos tempos: Leonardo da Vinci hackeava suas obras de arte para nelas codificar sua verdadeira posição (não) religiosa; Richard Stallman é frequente vítima de ataques pessoais que pretendem desmerecer e desviar a atenção da sua mensagem de libertação dos usuários, que tanto assusta os dominadores; os criadores e mantenedores do The Pirate Bay foram processa-

dos e condenados à prisão por publicar o equivalente digital a uma lista telefônica; Julian Assange, fundador do Wikileaks, ousa publicar segredos vazados que outros jornais igualmente publicaram mas, ao contrário dos demais, não aceita se calar em face das ameaças de morte, acusações e cortes de fontes de recursos para pagar sua defesa. Jacob Applebaum, desenvolvedor da rede de anonimização TOR que ativistas de direitos humanos usam em todo o mundo para escapar dos bloqueios e da censura a que seus governos tentam sujeitá-los, é sempre barrado pela polícia de fronteira de seu próprio país, que insiste que ele revele suas senhas de criptografia para que acessem os dados em seus computadores; a pressão aumentou depois que ele passou a colaborar com o Wikileaks. Aaron Swartz foi perseguido até a morte, com ameaças de décadas de prisão, por baixar artigos demais de um repositório científico ao qual ele tinha acesso ilimitado; suspeita-se que a pressão tenha advindo também de sua colaboração com Wikileaks.

A rigor, pouco há de novo nisso tudo: historicamente, a elite sempre esteve um passo adiante da população em geral em termos tecnológicos, usando esse passo adiante para dominar e

manter controle sobre os demais. Douglas Rushkoff, numa palestra no SWSX 2010 sobre seu livro “Programa ou Serás Programado”, apontou que a “invenção” do alfabeto permitiu a criação de registros históricos escritos, mas o poder de ler e interpretar os textos escritos, via de regra as escrituras sagradas, ficou restrito ao clero. Quando surgiu a imprensa, os livros deixaram de ser algo restrito à nobreza, porém era a realeza quem permitia ou não a publicação das obras: tornou-se o novo instrumento de controle do pensamento e portanto do comportamento. Quando chegou a Internet, a rede em que pessoas se conectam através de computadores interligados, passou a ser possível a qualquer um publicar o que bem entendesse. A elite demorou a perceber que essa ferramenta, criada por hackers, poderia se tornar uma nova revolução, mas hoje, governos e grandes empresas se aliam para tentar controlar a Internet, por vezes bloqueando sites para evitar a publicação de certas informações, por vezes filtrando o acesso a informação indesejável à classe dominante, e incansavelmente tentando tirar dos usuários o controle sobre seus computadores: temos uma geração de blogueiros, mas o controle sobre os meios

está nas mãos de programadores, ou melhor, daqueles que pagam seus salários. Quem não programa (ou usa um computador que impede a programação) é programado; se crê usuário, mas é usado.

Perdoem-me os não letrados nessa disciplina, mas programar é fundamental! Afinal, ensinamos a ler e escrever não porque todos vão se tornar escritores profissionais, mas porque é uma competência essencial para a comunicação, para a vida em sociedade. O mesmo se pode dizer da aritmética, de outros idiomas. Ora, linguagens de programação, antes de serem “de programação” são linguagens, que crianças se mostram capazes de adquirir e dominar com extrema facilidade. Já imaginou se, em vez de ficarem limitados a máquinas de escrever com TV digital e som 5.1 e ao que desenvolvedores de aplicativos acharam que seria vantajoso lhes oferecer, pudessem se tornar mais eficientes programando para automatizar suas próprias tarefas repetitivas (coisa que o computador foi criado para fazer)? Por que deveria ser limitado a poucos esse superpoder de evitar a frustração, pela incapacidade de expressar o que desejam (se é que conseguem formular o desejo para si mesmos), pela incapacidade de sequer compreender os có-

digos que governam o funcionamento de sua própria máquina? Analfabetos algumas vezes têm alguma vaga noção de quanto perdem por não saber ler, mas à classe dominante não interessa que as pessoas tomem ciência de quanto perdem por não poderem controlar seus computadores: assim fica mais fácil monitorarem e controlarem a vida dos dominados através de seus computadores. Incapazes de acessar, compreender e alterar os códigos explícitos na programação dos computadores que controlam parte cada vez maior da realidade à sua volta, que esperança têm essas pessoas de desvendar, compreender e influenciar os códigos e protocolos implícitos nas interações sociais, no estabelecimento e manutenção das estruturas de classes sociais?

Por isso me parece que o caminho para uma sociedade mais justa passa necessariamente pela educação, mas uma educação libertadora, cidadã, ética, de hackers, não uma (con)formação de gado humano. É preciso instigar a curiosidade e o compartilhamento de informações, não os segredos dos códigos-fonte e as restrições jurídicas e tecnológicas que se impõem sobre muitos materiais educacionais. Incentivar o questionamento,

em vez de autoritariamente exigir um respeito à suposta autoridade. Mostrar os prazeres do aprender, do pesquisar, do descobrir e do inventar, construindo cada um seu próprio caminho para uma maior diversidade, em vez de todos decorarem o mesmo conteúdo, seguindo a mesma trilha demarcada até chegarem ao abatedouro.

Escolas devem dar o exemplo: ensinavam a compartilhar lanches e brinquedos levados à aula; por que não livros e programas de computador, que, quando digitais, são compartilhados não por divisão, mas por multiplicação? Por que a escola, em vez de ensinar valores humanizantes, se vê quase forçada a ensinar que é proibido ajudar ao próximo? Não! Isso está errado! É preciso ensinar a controlar as máquinas, não que elas não possam ser controladas por nós porque seus controles são secretos. Cada vez que um aluno pergunta ao professor e o professor responde “é segredo, não posso lhe ensinar isso”, ou, pior, “não mexa, criatura!”, mata a curiosidade, mata um filhote de hacker antes que ele cresça e “traga problemas”; reforça no estudante o auto-estereótipo de consumidor passivo, não de agente cocriador de ideias e da realidade!

A mensagem sutilmente ditada pelas elites dominan-

tes parece ser “não ouse compreender ou influenciar”: consuma, consuma, consuma, sempre passivamente. Consuma software e computadores do jeito que eles vêm; consuma obras culturais, mas não ouse querer transformá-las, adaptá-las ou remixá-las; consuma os meios de comunicação, mas sempre da forma restrita mais lucrativa para os intermediários; consuma a sociedade do jeito que ela vem; consuma o videogame eleitoral e tenha fé que a urna que nem você nem ninguém pode auditar conte seu voto direitinho, ou, se você for um pouquinho mais cínico, que o último a corromper o processo de totalização (que portanto decide o resultado final) tenha sido o seu candidato favorito. A pseudodemocracia em que vivemos é assim: os debates são fictícios ou repletos de distrações, os eleitos representam os interesses daqueles que financiam suas campanhas (não daqueles que os elegeram), e as informações a que temos acesso através da grande imprensa vem de pouquíssimas agências de notícias com interesses alinhados entre si, ainda que seja publicada em diversos meios para dar uma ilusão de diversidade, o que reforça a aparente veracidade noticiada. É uma prisão para sua mente, Neo!

Para romper as amarras das redes antissociais que nos aprisionam, precisamos começar formando hackers livre-pensantes, fomentando sua curiosidade e seu espírito de compartilhamento. Interessados que são, capacitados para compreender e modificar os códigos dos computadores e dos meios de comunicação, terão alguma chance de resgatar o controle sobre eles. Sem essas amarras, com acesso a informação menos parcial, quicá poderão desvendar os códigos e protocolos das redes antissociais e restabelecer a democracia, que, graças à atual tecnologia, poderia até mesmo voltar a ser direta, dispensando intermediários corruptos ou corrompíveis. Como hackers, terão até mesmo o interesse em participar ativamente; tendo resgatado o controle sobre os mei-

os de comunicação e a liberdade de expressão, e mantido a sede de informação e de compartilhamento, poderão tomar decisões informadas e ética. Porque assim somos nós, hackers!

Como a alternativa é que continue a crescer a manipulação dos fatos, a censura e a perseguição aos que buscam informação, conhecimento e uma sociedade mais justa, imploro: salvem os hackers, ou salve-se quem puder! 🇧🇷

Copyright 2013 Alexandre Oliva

Esta obra está licenciada sob a Licença Creative Commons CC BY-SA (Attribution ShareAlike, ou Atribuição e Compartilhamento pela mesma licença) 3.0 Unported. Para ver uma cópia dessa licença, visite <http://creativecommons.org/licenses/by-sa/3.0/> ou

envie uma carta ao Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

Cópia literal, distribuição e publicação da íntegra deste artigo são permitidas em qualquer meio, em todo o mundo, desde que sejam preservadas a nota de copyright, a URL oficial do documento e esta nota de permissão.

<http://www.fsfla.org/svnwiki/blogs/lxo/pub/hackers>

ALEXANDRE OLIVA

Conselheiro da Fundação Software Livre América Latina, mantenedor do Linuxlibre, evangelizador do Movimento Software Livre e engenheiro de compiladores na Red Hat Brasil. Graduado na Unicamp em Engenharia de Computação e Mestrado em Ciências da Computação.



Um foguete

Dicas para otimização do UBUNTU

por Wagner da Silva Arestides

Acelere os softwares instalados no Ubuntu com o **prelink**. Para quem ainda não conhece, o **prelink** é um software livre escrito por Jakub Jelínek da Red Hat para sistemas operacionais compatíveis com o POSIX, principalmente o GNU/Linux porque ele modifica executáveis ELF. Destina-se a aceleração de um sistema, reduzindo o tempo que um programa precisa para iniciar.

Primeiramente instale-o através do apt-get ou pela Central de Programas do Ubuntu. Lance o terminal teclando CTRL+T e proceda dessa maneira:

```
sudo apt-get install prelink -y
```

Há problemas para iniciar o Skype depois que o **prelink** é executado, então desta forma edite esse arquivo de configuração abaixo com o editor de textos de sua preferência. Nesse exemplo usarei o 'nano' que é um editor em modo texto bem simples e fácil de usar:

```
sudo nano /etc/prelink.conf
```

Insira a linha abaixo, logo após as linhas

que seguem com o "-b":

```
-b /usr/bin/skype
```

Tecle CTRL+O pra salvar e CTRL+X pra sair e rode-o depois da seguinte maneira:
sudo prelink -amR

Para implementá-lo no sistema toda vez que você atualiza ou instala algum software no sistema, proceda da seguinte forma:

- Edite, com o editor de sua preferência, o arquivo `/etc/default/prelink` alterando de "unknown" para "yes" a linha "PRELINKING=unknown". Salve e feche o arquivo de configuração.

- Execute a linha de comando abaixo:

```
sudo /etc/cron.daily/prelink
```

- Crie o arquivo `/etc/apt/apt.conf` para que a implementação de execução do **prelink** seja iniciada toda vez que você instalar ou atualizar o sistema inserindo essa linha:

```
"DPkg::Post-Invoke {"echo Running\ prelink, please\ wait...;/etc/cron.daily/prelink";}";
```

Veja o exemplo:

```
sudo nano /etc/apt/apt.conf
```

```
DPkg::Post-Invoke {"echo Running  
prelink, please  
wait...;/etc/cron.daily/prelink";}
```

Tecla **CTRL+O** pra salvar e **CTRL+X** para sair. Atualize ou até mesmo instale algum pacote para testar a implementação do *prelink* no sistema. Será exibido no final de cada atualização ou instalação de pacotes a mensagem "*Running prelink, please wait...*".

Preload

Pra complementar a dica acima, instale o **preload**:

```
sudo apt-get install preload -y
```

O **preload** reduzirá o tempo de inicialização das aplicações otimizando assim a abertura posterior de um software que fora fechado anteriormente.

Para quem usa uma placa de vídeo integrada ou APU da Intel *Sandy Bridge* instale o PPA Glasen:

```
sudo apt-add-repository  
ppa:glasen/intel-driver -y && sudo apt-  
get update && sudo apt-get dist-upgrade  
-y && sudo apt-get install xserver-xorg-  
video-intel -y
```

Logo após crie o arquivo */etc/X11/xorg.conf* inserindo as informações a seguir:

```
sudo nano /etc/X11/xorg.conf
```

```
Section "Device"  
Identifier "Card0"
```

```
Driver "intel"  
Option "AccelMethod" "sna"  
EndSection
```

Digite **CTRL+O** pra salvar e **CTRL+X** pra sair.

Reinicie o sistema para testar:


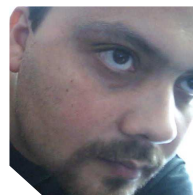
```
sudo init 6
```

Caso não dê certo o procedimento remova o arquivo */etc/X11/xorg.conf*:

```
sudo rm -f /etc/X11/xorg.conf
```

Implementei essa dica no meu netbook e depois consegui usar o Unity 3D no mesmo. 🙌

WAGNER DA SILVA ARESTIDES
é entusiasta do sistema operacional
GNU/Linux e software livre há 9 anos.



Na VirtualLink, você encontra desde
Treinamentos Oficiais em Linux até as
melhores Soluções em TI do mercado.

VirtualLink
Soluções e Treinamentos em Linux
www.virtuallink.com.br

Vai faltar marca página para tanto conteúdo!



- + Conteúdo
- + Dicas e truques
- + Assuntos extendidos
- + Entrevistas e mais...

REVISTA

blogosfera

www.revistablogosfera.com.br



Virtualização de desktops com VAGRANT

por João Victor Rocon Maia

O que é o Vagrant?

Vagrant^[1] é uma tecnologia de virtualização com licença The MIT License mas que, como outras grandes tecnologias da atualidade, tem um empresa envolvida, HASHICORP^[2]. O Vagrant usa o VirtualBox^[3] como base de virtualização o que pode parecer estranho já que ele vem para ajudar nesse ponto. Mas não se assustem, vou explicar melhor a respeito.

Com a missão de ajudar a todos o Vagrant pode ser instalado facilmente via instaladores simplificados^[4] ou de maneiras mais interativa através do código fonte^[5]. No meu Blog pessoal^[6] eu abordo sobre o assunto e um problema relacionado a instalação e manutenção de uma máquina virtual (VM - *virtual machine*) com o Vagrant.

Benefícios de usar o Vagrant

Um exemplo clássico de como o Vagrant pode ajudá-lo ocorre quando da chegada de um novo empregado na empresa. Em geral, nos primeiros dias, esse empregado passa por todo aquele processo de apresentação da empresa, do produto e, por fim, da configuração do seu ambiente de trabalho.

É na configuração do ambiente de trabalho, que muitos acabam perdendo muito tempo, por causa de erros ou até mesmo, pela demora da configuração. Com isso, empregado e empresa, perdem tempo de produtividade. Porém, com o Vagrant isso pode ser acelerado através de VMs prontas. O empregado receberia sua VM toda pronta e assim já iniciaria suas atividades.

A prova desses benefícios pode ser vista na comunidade de desenvolvedores com o Ruby On Rails que possui uma VM para ajudar os iniciantes a colaborarem com o Rails. Além disso, o que tem ajudado o mundo Open Source a apresentar esse crescimento são as grandes empresas como Mozilla, O'Reilly e Nokia por exemplo.

Colocando a mão na massa

Nos passos seguintes, seria ideal que usar Linux ou Mac OS X para conseguir acompanhar. Caso contrário, alguns passos eventualmente poderão ser diferentes.

Criando a primeira Box

Recomendo para quem esta começando, ou não tenha muita familiaridade com o terminal, utilizar o instalador que for mais adequado, apresentado no link^[4]. É hora de criar sua primeira VM mas é necessário, primeiramente, ter uma Box.



O que são essas Box's? O Vagrant por padrão trabalha apenas com o Ubuntu de Box prontas^[7] mas é possível instalar outros sabores de Linux ou até mesmo BSD através de Box de terceiros^[8]. Para adicionar uma Box ao seu repositório é bem simples:

```
$ vagrant box add precise32\  
http://files.vagrantup.com/precise32.box
```

Qual o significado da linha acima? Após o "add" estamos informando qual vai ser o nome da Box para que você possa usar posteriormente e, a url que vem em seguida, é a Box que será adicionada. No lugar da url poderia também, ser o caminho no disco para uma Box própria.

Iniciando a primeira VM

Agora que você já tem uma Box na sua máquina, podemos criar nossa primeira VM. O processo é bem simples:

```
$ mkdir ~/minha_vm
```

```
$ vagrant init precise32
```

Caso tudo ocorra bem algumas mensagens vão surgir no seu terminal e por fim um arquivo chamado Vagrantfile será criado. Eliminando os comentários do arquivo você deve obter algo parecido com o código abaixo:

```
# -*- mode: ruby -*-  
  
# vi: set ft=ruby :  
  
Vagrant::Config.run do |config|  
  
  config.vm.box = "precise32"  
  
end
```

Quem já conhece ou trabalha com Ruby deve ter percebido que a DSL do Vagrant é Ruby e nessa simples configuração já temos uma VM pronta. Vamos iniciá-la:

```
$ vagrant up

[default] Attempting graceful shutdown
of VM...

[default] Clearing any previously set
forwarded ports...

[default] Forwarding ports...

[default] -- 22 => 2222 (adapter 1)

[default] Creating shared folders meta-
data...

[default] Clearing any previously set
network interfaces...

[default] Booting VM...

[default] Waiting for VM to boot. This
can take a few minutes.

[default] VM booted and ready for use!

[default] Mounting shared folders...

[default] -- v-root: /vagrant
```

Veja que os passos são simples e rápidos. Tente comparar quanto tempo você teria gasto para fazer a mesma coisa instalando o Ubuntu na VM através do CD de instalação. Repare, no log de inicialização da VM, alguns detalhes:

- **[default] -- 22 => 2222 (adapter 1)**: isso é um mapeamento da porta 22 da VM para a porta 2222 da sua máquina hospedeira. Isso é útil para o acesso ssh;

- **-- v-root: /vagrant**: isso é um ponto de montagem em comum entre a sua máquina hospedeira e VM. Significa que na pasta **/vagrant** da VM você consegue acessar tudo dentro da pasta **~/minha_vm** da máquina hospedeira.

Agora, o mais importante, acessar a VM criada, nada mais natural de como todo o processo de criação da VM:

```
$ vagrant ssh
```

Pronto! Sua VM está pronta e você já consegue acessá-la e manipular como desejar. Lembrando que, por padrão, você vai acessar ao usuário “vagrant” que tem configurações de super usuário. Então, qualquer comando que requeira o “*sudo*” vai funcionar perfeitamente.

Para sair da VM basta usar o comando “*exit*” e para desligar “*vagrant halt*”.

Conclusão

Nesse artigo conseguimos ver como é fácil e rápido trabalhar com virtualização usando o Vagrant. Isso foi apenas uma pequena demonstração do que podemos fazer com ele. No meu Blog pessoal^[9] tenho outros artigos sobre o assunto exemplificando passos mais avançados caso tenham interesse no assunto. Espero ter conseguido passar o conhecimento dessa ferramenta e o mais importante é que seja de grande utilidade para você. Obrigado por ler e até o próximo! 🐦

Referências

- [1] <http://www.vagrantup.com/>
- [2] <http://www.hashicorp.com/>
- [3] <https://www.virtualbox.org/>
- [4] <http://downloads.vagrantup.com/>
- [5] <https://github.com/mitchellh/vagrant>
- [6] <http://va.mu/cQgI>
- [7] <http://va.mu/cQgI>
- [8] <http://www.vagrantbox.es/>
- [9] <http://blog.joaovrmaia.com/>

JOÃO VICTOR ROCON MAIA é formado em Engenharia de Computação pela Universidade Federal do Espírito Santo. Usuário de Software Livre desde 2005 e envolvido com o grupo local de usuários de SL - Tux-ES. Possui experiência com tarefas de administração de sistemas Linux e desenvolvimento de soluções Web. Email: maia.jvrm@gmail.com - Github: <https://github.com/joaovrmaia>



Ambiente de programação para AVR

Desenvolvendo aplicações micro controladas utilizando micros controladores da ATMEL, com IDE eclipse, programando em C e no Linux

por José Roberto Colombo Júnior

Os microcontroladores ATMEL são referência no mercado de sistemas embarcados.

Microcontroladores clássicos como os PIC, já estão cedendo lugar para esses novos concorrentes.

E isso não é em vão. Além da excelente estabilidade e imunidade a ruídos, estes “garotos” possuem uma fantástica arquitetura interna. Contando com muitos periféricos já embutidos, eles tornaram-se perigosos concorrentes para qualquer fabricante desse tipo de componente.

Uma das principais vantagens destes microcontroladores, na opinião do autor deste texto, é o conjunto de compiladores de C e C++, chamados de **avr-gcc** e **avr-g++**. No Linux, a maioria

das distribuições possuem meios muito simples para a instalação destes programas. No Windows, eles podem ser instalados através do programa **WinAVR**.

Pode não parecer, mas possuir um microcontrolador com seu compilador bem organizado abre um leque muito grande de opções para sua utilização. Veja o caso do PIC. Existe mesmo um compilador C *opensource*? E o acesso a esta ferramenta, é fácil? E uma última pergunta, a estabilidade do código compilado é boa?

O feito mas importante para a popularização deste microcontrolador, foi o Arduino. Criado para auxiliar hobistas, radioamadores e leigos, acabou crescendo e incorporando em suas bibli-

otecas, quase todo o hardware interno dos microcontroladores AVR, tornando-se uma fantástica plataforma para a rápida criação de protótipos.

Além disso, pessoas do mundo todo escreveram bibliotecas para hardware externo como por exemplo, display de 7 segmentos, display LCD, dispositivos de comunicação sem fio Zig-Bee, buzzers, controle de servomotores e motores de passo, entre outros.

Além do microcontrolador, o hardware do Arduino pode ser expandido com os famosos shields, que são placas de expansão que contém os mais variados circuitos, incluindo os acima citados e muito mais.

É sempre bom possuir uma interface de programa-

ção (IDE) organizada e com suporte a pelo menos dois botões: compilar e enviar o código gerado para o microcontrolador. Nesse artigo, mostrarei como instalar um plugin para programação de microcontroladores AVR no Eclipse, que roda nativamente no Linux e no Windows.

O Eclipse é uma ferramenta de código aberto para programação em geral, em Java, C, C++, entre outros. Em nosso caso, tanto a versão do Eclipse clássico como o especial para C/C++ servirão para a instalação do plugin.

Para instalar o plugin no Debian, Ubuntu e derivados, é necessário rodar o seguinte comando no terminal:

```
$ sudo apt-get install\
eclipse eclipse-cdt
```

Para instalar o Eclipse no Gentoo Linux, basta rodar o seguinte comando no terminal:

```
# emerge -avq eclipse
```

No caso de outra distribuição Linux, basta utilizar o gerenciados de pacotes padrão da distro. No Windows, é necessário acessar o site do Eclipse⁽¹⁾ e fazer o download do binário para esse sistema operacional.

É importante ressaltar que o binário do Eclipse vem em um arquivo compactado e é necessário extrair o conteúdo dentro de



Figura 1: Display HD4470 (Fonte: <http://www.pictronics.com.br>)

uma pasta onde o usuário normal não tenha permissão de escrita, por exemplo a pasta Arquivos de Programas.

Com o Eclipse instalado, é preciso instalar o compilador de C *avr-gcc*. No caso do Windows, o compilador é disponibilizado em um pacote de programas, chamado WinAVR. Para instalá-lo, deve-se acessar o site do projeto e fazer o download da última versão. A instalação do pacote é muito fácil, basta seguir as instruções apresentadas na tela pelo instalador.

No Linux, existem diversas formas de instalar o compilador *avr-gcc*. No Debian e derivados, pode-se instalá-lo através do seguinte comando:

```
$ sudo apt-get install\
gcc-avr avr-libc\
avrdude
```

Veja que além do pacote do compilador, serão instala-

das as bibliotecas básicas (*stdio.h*, *stdlib.h*, entre outras) e o programa que realiza o link de comunicação entre o computador e o microcontrolador, o *avrdude*. Este programa possui o objetivo de pegar o nosso arquivo “binário” e copiá-lo para a memória flash (e EEPROM) do microcontrolador.

Para instalar os compiladores do Gentoo Linux, é necessário o uso do programa *cross-dev*, que é responsável por instalar compiladores para *cross-platform*. A ideia aqui, é utilizar um computador “super crescido”, para compilar por exemplo, o kernel do Linux para os antigos AMD K6-II de 400 MHz. No nosso caso, vamos instalar um compilador para sistemas de 8 bits!

A instalação é muito simples no Gentoo, basta seguir o tutorial de instalação do Arduino, que foi muito bem escrito. O tutorial está

hospedado no seguinte endereço; <http://va.mu/cPJo>.

Ao término da instalação, haverá um pequeno problema (apenas no Gentoo). Os arquivos com os *defines* para cada microcontrolador da ATMEL não são copiados automaticamente para a área de trabalho do compilador. Para arrumar isso, simplesmente copie o arquivo correspondente ao microcontrolador que está utilizando para a pasta onde está o seu projeto, que será descrita mais adiante.

Por fim, falta a instalação do plugin para microcontroladores ATMEL, desenvolvido para Eclipse. Sua instalação é exatamente a mesma em qualquer plataforma, já que é instalado de dentro do próprio Eclipse.

Para instalá-lo, basta abrir a interface IDE Eclipse e clicar sobre a aba “*Help*” e então, sobre o item “*Install New Software...*”, como mostrado na **Figura 2**.

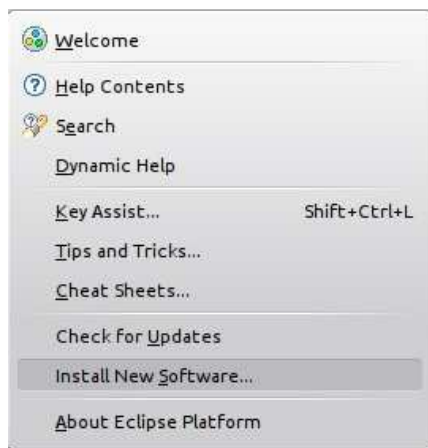


Figura 2: Menu “Help” do Eclipse, para instalação do plugin

Será aberta uma nova janela, parecida com a mostrada na **Figura 3**:

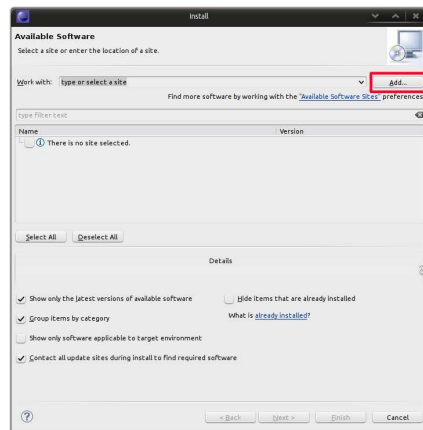


Figura 3: Menu de instalação de plugins do Eclipse

Nessa janela, são mostrados os plugins que estão sendo instalados no momento. Para iniciar uma nova instalação, clique sobre o botão **Add**, como destacado em *vermelho* na **Figura 3**. Será aberta uma nova janela como a mostrada na **Figura 4**:

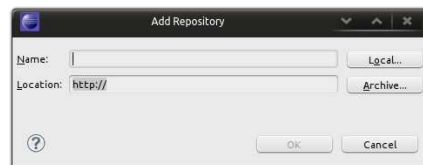


Figura 4: Menu de adição de novo repositório

O campo “**Name**” deve receber um nome qualquer de escolha do usuário, de forma que fique fácil identificar qual plugin ele gerenciará. Ele servirá somente para o propósito de identificação. Já o campo “**Locati-**

on” deve receber a URL do repositório do AVR Plugin, que é: <http://avr-eclipse.sourceforge.net/updatesite>.

Por fim, o botão **OK** será habilitado. Clique sobre ele. Será aberta uma nova janela, parecida com a mostrada na **Figura 5**.

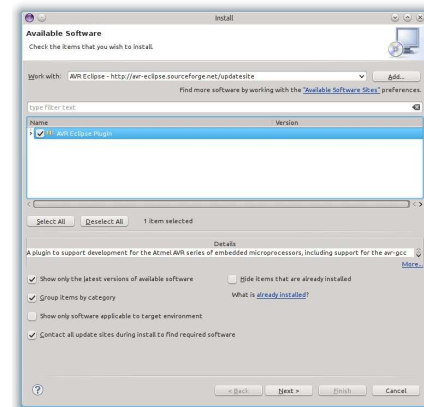


Figura 5: Seleção do plugin

Agora, basta clicar em **NEXT**, aceitar a licença GPL e aguardar a instalação do mesmo. Pronto! Reiniciar o Eclipse encerrará a instalação. Podemos agora, utilizar o Eclipse para escrever programas na linguagem C para microcontroladores AVR.

Para criar um novo projeto, a fim de começar a programar, basta clicar no menu *File*, em seguida no item *New* e por fim, *C Project* para programar em linguagem C ou *C++ Project* para programar em linguagem C++.

Será aberta uma nova janela, como demonstra a **Figura 6** (na próxima página):

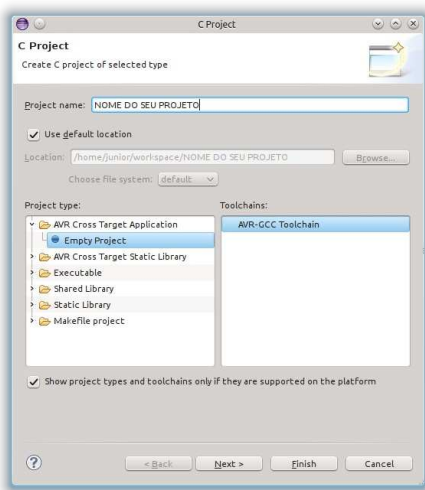


Figura 6: Criação de um novo projeto

Nesta caixa, o usuário terá de escolher entre dois tipos de projeto: **AVR Cross Target Application**, que é a aplicação comum ou o **AVR Cross Target Static Library**, que servirá para a criação de bibliotecas estáticas. Para aplicações comuns, basta escolher a primeira opção.

Em seguida, altere o nome do seu projeto, para algum de sua preferência. Evite o uso de caracteres especiais (como acentuação) e espaços. Então, clique em **Next**.

Na próxima etapa, há as opções **Debug** e **Release**. Se o usuário tiver interesse na simulação do projeto em algum simulador, como por exemplo o **simulavr**, será necessário manter a caixa **Debug** ativada. Com a caixa **Release** ativada, o Eclipse gerará um arquivo **HEX** para ser carregado no microcontrolador. Feita essa escolha, clique em **Next**.

Será aberta uma nova caixa, para a seleção do microcontrolador e da sua respectiva frequência de clock. Basta fazer as devidas seleções. Por fim, clique em **Finish**. Seu projeto já está criado.

Para realmente começar a programar, clique com o botão direito sobre o nome do seu projeto localizado no **Project Explorer** do Eclipse, à esquerda da tela. Então, escolha a opção **New** e por fim, **C Source File**. Para criar um arquivo do tipo cabeçalho, escolha **H Header File**.

Você pode verificar se tudo está funcionando corretamente, inserindo esse código ao seu primeiro programa (somente para testar):

```
int main(void) {
    while (1) {} // Loop infinito

    return (1); // Nunca acontecerá
}
```

Vamos compilar esse programa clicando sobre o botão **Build All**, como mostrado na **Figura 7**, ou simplesmente utilizar o atalho **CTRL+B**.

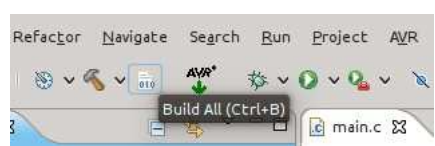


Figura 7: Botão para compilar o projeto

Se tudo correu bem, então você terá uma saída parecida com a mostrada a seguir:

```
Device: atmega8
Program:160 bytes (1.0% Full)
(.text + .data + .bootloader)
Data: 0 bytes (0.0% Full)
(.data + .bss + .noinit)
Finished building: sizedummy
```

Para fazer **UPLOAD** do seu projeto para o microcontrolador, será necessário que você tenha um programador (USBASP, por exemplo). Para configurá-lo no Eclipse, o usuário poderá seguir esse tutorial, que está muito bem escrito e detalhado: <http://va.mu/cPLv>

Eu mantenho no GITORIOUS algumas bibliotecas que escrevi para o microcontrolador ATMEGA8. No entanto, para portá-las para outros microcontroladores, são necessárias pouquíssimas alterações no código. É claro que essas bibliotecas estão sob a licença GPL. Para baixá-las, acesse a página da web: <http://va.mu/cPL5>

Nos próximos artigos, explicarei como utilizar essas bibliotecas, incluindo exemplos de aplicação em hardware. 🙋

JOSÉ ROBERTO COLOMBO JÚNIOR

é aluno de graduação do último semestre em engenharia elétrica na UNESP de Ilha Solteira, representante do corpo discente na Comissão Local de Informática desde 2010 e membro do Laboratório de Ultra Som.



Revista Segurança Digital, você já ouviu falar de nós?

Super Nerds, Geeks viciados em tecnologia ou Super-heróis do mundo digital, se você acha que estamos falando disto então você está um pouco enganado.

Realmente somos SUPER-FÃS de tecnologia, mas somos pessoas normais, com família para criar, contas para pagar e alguns com patrão para aguentar (hehehehe). O que nos difere mesmo é nosso DNA, marcado pelo gosto de compartilhar informação e conhecimento. Para nós isto não é obrigação, é prazer!

Mas, de certa forma, todos que tem responsabilidades e compromissos tem um pouco de herói em si. Se você acha que os SUPER-HERÓIS de verdade são aqueles com capa vermelha e um "S" no peito, então é melhor você parar de ver TV e começar a ler a revista SEGURANÇA DIGITAL.

<http://www.segurancadigital.info>
www.facebook.com/segurancadigital

REVISTA
**Segurança
Digital**

O software livre como aliado ao combate a falsificação: reconhecimento de padrões em imagens

por Hailton David Lemos

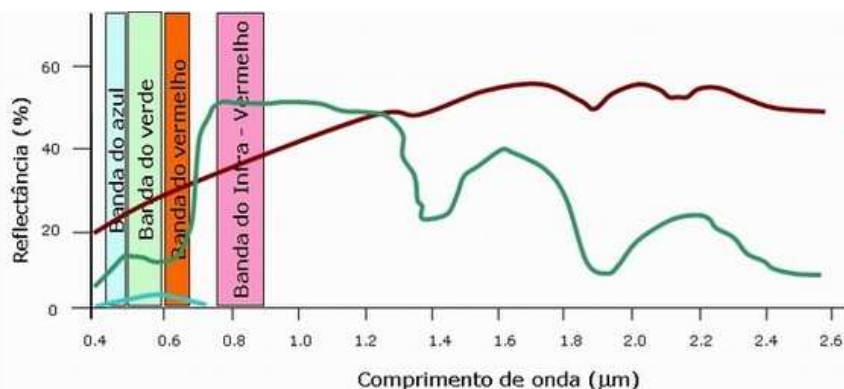
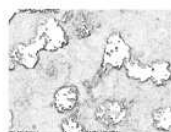
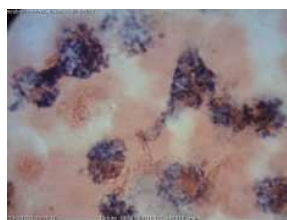
Hoje vivemos em uma sociedade em que cada vez mais estamos comprando gato por lebre, em função do grande número de falsificações existentes no mercado. Por vezes adquirimos um produto que julgamos ser original, quando na verdade são verdadeiras falsificações. Detectar esse tipo de fraude não é um processo trivial ou fácil. Entretanto com o avanço dos algoritmos de processamento de imagens o combate a falsificação está ganhando um aliado

importante. Temos por exemplo a adulteração das gasolinas pela adição de solventes, que é uma prática ilegal, e que ocorre com frequência. A adição ilegal de solventes além de prejudicar os consumidores acarreta em perdas de arrecadação por sonegação fiscal. Para ajudar nesse combate o uso do software livre pode ser uma ótima alternativa, definindo, por exemplo, padrões de reconhecimento de imagens. E uma das funções do reconhecimento de padrões em imagens pode ser o de



determinar uma maneira para que possam ser detectadas fraudes, e esse é o assunto que estamos tratando nesse artigo.

O objeto discutido será a fraude em líquidos, tais como combustíveis, bebidas e outros. A partir de um padrão definido, é possível desenvolver um algoritmo que seja capaz de ser estendido para detecção de fraude em outros materiais também. Por exemplo, suponha que se deseja criar um padrão para distinguir um produto falso do verdadeiro. O primeiro passo é entender o que é o produto a ser analisado e a partir daí começar a desenvolver o trabalho. Vamos tomar como exemplo o uísque, que é um produto bastante falsificado. O uísque é uma aguardente de cereais, com uma característica comum que é a fermentação dos grãos e a destilação máxima de 80% de álcool para os derivados do milho e de 90% de álcool para os derivados de outros grãos. O uísque ganha 60% do seu sabor baseado no tipo de barril usado no seu envelhecimento. E todas essas características físico-químicas devem ser levadas em consideração no momento do processamento das imagens que serão capturadas digital-mente. A imagem digital é uma representação em uma região discreta, limitada através de um conjunto finito de valores inteiros que representam cada um



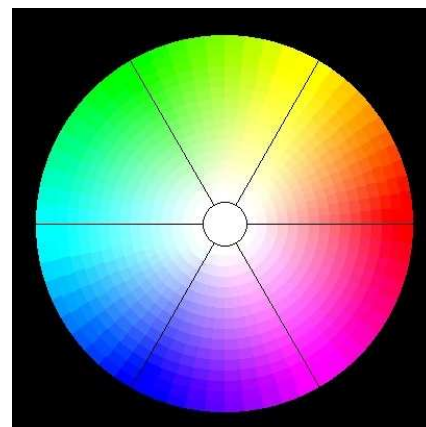
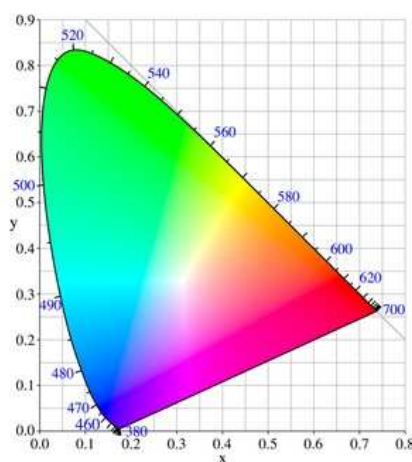
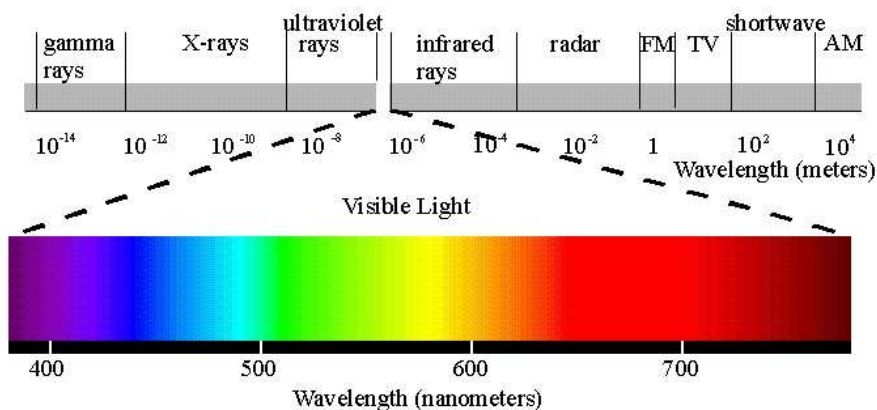
dos seus pontos. As imagens podem ser binárias, monocromáticas ou coloridas. Uma imagem colorida é uma imagem onde a cor em cada ponto é definida através de três grandezas: luminância, matiz e saturação. A luminância está associada com o brilho da luz, a matiz com o comprimento de onda dominante e a saturação com o grau de pureza ou intensidade do matiz. A imagem colorida também permite medir a temperatura da cor, que é a temperatura de um corpo negro ideal, que irradia luz de tonalidade semelhante à da fonte de luz.

Neste artigo, a opção foi trabalhar com imagens coloridas, tiradas com equipamento que reflete um espectro ultravioleta

que incide sob o líquido a ser analisado - o uísque, com o objetivo de diminuir ruídos e ter um maior poder de absorbância, uma vez que ruídos podem distorcer a imagem e dificultar o reconhecimento de padrões. A absorbância ou absorvência é a capacidade intrínseca dos materiais em absorver radiações em frequência específica. Usualmente, tal propriedade é empregada na análise de soluções em química analítica.

A opção pelo espectro ultravioleta se deu pelo seu poder de absorbância, levando-se em conta que existem vários elementos e reações químicas presentes no líquido em estudo, uísque. Para identificar as formas e tamanhos a ferramenta adotada foi a

morfometria, que é um padrão matemático que estuda as formas pertencentes à mesma população estatística e possibilita a identificação de populações de organismos vivos ou reações químicas, que podem assumir formas ou tamanhos diferentes conforme o ambiente em que se desenvolveram ou que estão expostos. A cor também é uma característica essencial na análise proposta, entretanto a cor exibida depende do dispositivo que está sendo utilizado, e para contornar qualquer distorção, foi adotada um padrão neste trabalho : o padrão CIE. Esse padrão é um espaço de cor puramente matemático e, portanto independente de dispositivos. Nesse trabalho as imagens são convertidas para os espaços de cores, HSB, HSL, HSV, CIE Lab, CIE Luv e CIE XYZ e dados como brilho, luminância e cromaticidade são utilizados nos cálculos, e pequenas diferenças no espaço de cores podem fazer uma grande diferença. A cromaticidade da cor, por exemplo, é especificada por dois parâmetros derivados de X e Y, dois dos três valores normalizados que são funções de todos os três valores tristimulares X, Y e Z, com o calculo da curva de potência espectral, que dá o comprimento



de onda da energia.

As leis da mistura aditiva também é outro ponto de apoio fundamental, que também é utilizado para caracterização do produto, falso e verdadeiro. A unidade de medida utilizada nos cálculos da imagem é o pixel, o menor ponto que forma uma imagem digital, o menor elemento num dispositivo de exibição, ao qual é possível atribuir-se uma cor, e cada pixel tem seu próprio endereço. Uma representação adequada de um pixel consiste em uma área retangular ou quadrada. Três aspectos são muito importantes nesse

contexto de representação: a vizinhança, a conectividade e as distancias. A relação de distância indica o quão perto ou quão longe um pixel se encontra do outro geograficamente no plano ou no espaço. A distancia Euclidiana é um padrão que foi adotado para os cálculos.

Uma imagem possui uma série de informações armazenadas, e uma das mais relevantes se refere à cor ou ao tom de seus pixels, e o histograma foi o padrão estático adotado para apresentar os determinados níveis de cor detectados em cada um dos pixels que

formam a imagem em estudo. Pode-se considerar que o primeiro passo para análise e entendimento da imagem consiste em particioná-la de modo que seja possível explicitar regiões representativas da imagem. A segmentação se refere ao processo de dividir uma imagem digital em múltiplas regiões, com um conjunto de pixels, com o objetivo de simplificar para facilitar a sua análise. A detecção de bordas, que é uma técnica de processamento de imagem para determinar pontos de uma imagem digital em que a intensidade luminosa muda repentinamente, também foi adotada, e gera informações estatísticas usadas na inteligência computacional da análise. O algoritmo K-means, algoritmo que fornece uma classificação de informações de acordo com os próprios dados, também é adotado para fazer os agrupamentos das informações. A classificação é baseada em análise e comparações entre os valores numéricos dos dados obtidos com os processos já descritos. Desta maneira, o algoritmo automaticamente vai fornecer uma classificação sem a necessidade de nenhuma supervisão humana. Para gerar as classes e classificar as ocorrências, o algoritmo faz uma

comparação entre o valor da distância euclidiana e também de cada cor por meio da distancia das cores no gráfico de cromaticidade do padrão CIE. O algoritmo vai analisar todos os dados coletados das cores e também da imagem além dos dados morfométricos e criar as classificações. A implementação dessa ferramenta esta baseada em software livre. A linguagem de programação adotada foi o Perl com os pacotes gráficos GD e Imagemagick, além de pacotes matemáticos. Os resultados geram informações para serem utilizados e processados nos programas como R, SCILAB, OpenOffice, Excel e MathLab que servem de apoio a inteligência da ferramenta. 🌐

Referências:

Aoe, Jun-ichi. *Computer Algorithms: string pattern matching strategies.* Los Alamitos, CA, USA: IEEE Computer Society Press, 1994.

Berry, Michel W. *Survey of text mining: clustering, classification, and retrieval -* New York - NY - USA - Springer - 2003.

Building Bioinformatics Solutions with Perl, R and MySql - Bessant, Conrad; Shadforth, Ian; Oakey, Darren. (2009)

Crochemore, Maxime; RYTTER, Wojciech. *Jewels of Stringology text Algorithms.* Singapore: World Scientific Publishing, Co, 2003.

Gonzalez, Rafael C., Woods, Richard. *Digital image processing -* Prentice-Hall - 2007.

Valentin, Jean Louis. *Ecologia Numérica: uma introdução à análise multivariada de dados ecológicos.* Rio de Janeiro: Interciência, 2000.

Pedrini, Hélio. *Análise de imagens digitais: princípios, algoritmos e aplicações.* Thomson Learning - 2008.

Jain, Anil K. *Fundamentals of digital image processing.* Printice Hall - 2008.

Dougherty, Geoff. *Digital image processing for medical applications.* Cambridge - 2009.

Oradski, Gary. Kaehler, Adrian. *Learning OpenCV.* O'Reilly - 2008.

Haykin, Simon. *Redes Neurais: princípios e pratica.* Bookman - 2008.

Sanders, Jason. Kandrot, Edward. *CUDA by example. An introduction to General-Purpose GPU programming.* NVIDIA - 2011.

Eisenberg, J. David. *SVG Essentials.* O'Reilly - 2002.

Ammeraal, Leen. Zhang, Kang. *Computacao gráfica para programadores Java.* LTC - 2008.

Dorigo, Marco. Stutzle, Thomas. *Ant Colony Optimization.* Massachusetts Institute of Technology - 2004.

Azeredo, Eduardo. Conci, Aura. *Computação gráfica: Geração de imagens.* Campus - 2003.

Floreano, Dario. Mattiussi, Claudio. *Bio-Inspired artificial intelligence: theories, methods, and technology.* Institute of Technology - 2008.

O'Rourke, Joseph. *Computacional Geometry in C.* Cambridge - 2008.

HAILTON DAVID

LEMOs é mestrando em Engenharia de Produção e Sistemas. Bacharel em Administração de Empresas, tecnólogo em internet e redes de computadores, licenciando em ciências biológicas, especialista em tecnologia da informação, matemática e estatística. MBA em Planejamento e Gestão Estratégica. Trabalha com desenvolvimento de sistemas há mais de 25 anos.



Aproximações entre a Pedagogia e o movimento do Software Livre

por Selma Regina Gomes

Este texto objetiva apresentar uma reflexão envolvendo a Pedagogia e a filosofia defendida pelo movimento de Software Livre. A intenção é pontuar algumas aproximações possíveis entre as duas áreas, compartilhando o depoimento de uma Pedagoga que acompanha nos bastidores, a luta em prol do entendimento do software livre como uma filosofia.

Que pontos comuns se pode extrair da Pedagogia e do movimento do Software livre? Existe uma relação bem mais estreita que se pode cogitar a princípio. Como profissional formada em Pedagogia e simpatizante do movimento do Software Livre, resolvi escrever como percebo as aproximações entre estas duas áreas.

Há mais ou menos seis anos, tenho presenciado algumas discussões a respeito de Software Livre. Sou Pedagoga e meu contato com o mundo informatizado não vai além do acesso normal à internet para pesquisas, partilha de novidades com os amigos, leitura, entretenimento e claro a digitação de textos necessários a minha profissão.

Ouvi pela primeira vez a palavra software livre, quando na instituição onde trabalhava, os administradores resolveram trocar um sistema operacional por outro, que todos os conhecidos diziam ser muito “complicado”. Procurei in-

formações sobre o assunto, comecei a manusear os computadores da instituição esperando estar diante de uma tarefa quase impossível de ser realizada. Percebi que não havia tanta diferença entre o novo sistema e o outro. Adaptei-me rapidamente ao manuseio do novo sistema operacional.

Tempos depois, encontrei um grupo de pessoas que falavam sobre Software Livre como se estivessem falando de um conhecimento revolucionário, defendiam a propagação da ideia e se empenhavam em fazê-lo como se defendessem o país de um inimigo. Por que eles são tão fanáticos assim? Pensava eu a partir do meu conhecimento aligeirado sobre a questão.

A curiosidade me levou a indagar sobre o assunto e fazer algumas leituras que conduziram à reflexão que pretendo demonstrar neste texto, buscando as aproximações possíveis entre a Pedagogia e o movimento do Software Livre. A primeira constatação a que cheguei, partilhando das reflexões dos adeptos deste movimento é que Software não é a mesma coisa que Linux. Este é apenas um sistema operacional baseado em software livre. Parece ingênua e óbvia esta constatação, mas não é, visto que a grande maioria das pessoas iniciam seu pensamento sobre este tema com

a ideia de que as duas coisas são uma só.

A partir daí busquei confrontar as ideias postuladas pelo movimento com o conhecimento adquirido por mim em minha experiência como profissional da educação. Então, cheguei à conclusão que existem aproximações, tênues, entre as duas áreas que precisam ser desveladas, confrontadas. Acredito que ao partilhar as aproximações por mim percebidas, desenvolvo uma ação educativa, ainda sem a profundidade desejada, mas que pode contribuir para o entendimento e a aproximação das duas áreas.

Da minha formação como pedagoga, trago a consciência de que toda ação humana é uma prática educativa, independente dos resultados serem positivos ou não. E que uma ação pode ser pedagógica quando traz em seu bojo uma intencionalidade. A Pedagogia como área do conhecimento que trata especificamente do fenômeno educativo, procurando compreendê-lo numa relação dialética com as relações sociais e políticas na sociedade, tem o compromisso ético de promover o conhecimento de tal forma que promova a liberdade de pensamento e, conseqüentemente, a autonomia do ser humano, habilitando-o para a resolução de problemas, bem como a

atuação crítica e consciente na construção da sua realidade sociocultural.

Este é o primeiro ponto de convergência entre a Pedagogia e o movimento do Software Livre, pois ambos defendem a liberdade de pensamento. E neste aspecto gosto de pensar numa definição de liberdade à maneira do pedagogo brasileiro Paulo Freire, que nos deixou como legado a ideia de que a liberdade “é uma conquista, e não uma doação, exige permanente busca. Busca permanente que só existe no ato responsável de quem a faz. Ninguém tem liberdade para ser livre: pelo contrário, luta por ela precisamente porque não a tem.” (PAULO FREIRE, 1987, p. 34).

Vivemos em uma realidade em que a hegemonia de uma classe privilegiada, que se impõe sobre a égide do consumismo e da exploração, abafa a voz e o pensamento de muitos, considerados os “oprimidos”, segundo Freire.

Este autor, falando sobre os que compõem a classe dos “opressores”, diz que estes alimentam a crença de que “ter mais” é um privilégio, um direito intocável, pela sua condição de posse confundida com poder. Os seus contrários, os que “não tem”, ao exigirem mudança e liberdade para participarem na construção da sociedade, estão assumindo um comportamento de subversão. Daí a necessidade do controle, “E quanto mais controlam os oprimidos, mais os transformam em 'coisa', em algo que é como se fosse inanimado” (1987, p. 46).

Em face destas considerações, reafirmo a importância da Pedagogia e do movimento do Software Livre, por seu compromisso ético, responsabilidade e adesão à luta pela liberdade de pensamento. Luta que busca uma mudança “revolucionária”, por meio da qual os “oprimidos” tomam consciência das razões de sua exploração e assumem sua própria luta pela conquista da liberdade e de sua afirmação no mundo.

Este é um dos aspectos, decorrente da busca pela liberdade de pensamento, que aproxima os postulados da Pedagogia e do movimento do Software Livre, a defesa contra a exploração e luta pela inserção crítica e ativa dos indivíduos na sociedade. Acredita-se que à

medida que os sujeitos vão desvelando o “mundo dos opressores”, vão deixando de ser marionetes, e assumem o controle de seu comportamento, modificam sua visão de mundo e de sua ética.

Outro aspecto que considero aproximar estas duas áreas e que pretendo abordar, mas me referindo à questão do direito à aquisição do conhecimento, diz respeito à iniciativa de criação de um sistema operacional, com um código fonte aberto para leitura, modificações e redistribuição sem restrições, que numa primeira leitura pode parecer apenas uma questão técnica, mas, na verdade, carrega em si a filosofia do acesso livre ao conhecimento, um direito inalienável do ser humano.

Percebo aí um vínculo com o princípio básico da Pedagogia. Os pilares do Software Livre: A liberdade de executar, de estudar e adaptar às necessidades, de redistribuir de modo que se possa ajudar ao próximo, de aperfeiçoar e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie, são regras básicas do Software Livre, que se aproximam da convicção que tenho como Pedagoga de que a aquisição do conhecimento é um direito de todos. Todo indivíduo pode e deve ter acesso ao conhecimento produzido

pela humanidade; toda pessoa é livre para interpretar o conhecimento oferecido à sua maneira e adaptá-lo às suas necessidades; é obrigação daquele que tem o conhecimento, compartilhá-lo com o seu próximo; todo conhecimento é passível de aperfeiçoamentos, de mudança, de transformação em benefício da humanidade.

Estes pilares refletem o direito à participação coletiva na construção da sociedade, significam dar ao sujeito a oportunidade de opinar na coletividade com o mesmo nível de influência e dar a ele a capacidade de

ser consultado para as tomadas de decisão que dizem respeito à direção da sociedade em que vive. Esta é a defesa da Pedagogia, do movimento do Software Livre e de qualquer área do conhecimento que defenda a democracia como ponto de partida para a organização de uma sociedade.

Com estas considerações, espero ter lançado uma fagulha para a reflexão sobre as aproximações possíveis entre estas duas áreas, que na minha opinião, apesar de parecerem no campo prático distantes, no campo teórico comungam de ideias comuns. 🙏

Referências

FREIRE, Paulo. *Pedagogia do oprimido.* Rio de Janeiro: Paz e Terra, 1987.

SELMA REGINA GOMES

Mestre em Educação,
Pedagoga e
Especialista em
Psicologia Educacional,
funcionária efetiva da
Secretaria Estadual de
Educação do Estado de Goiás.



TIC na educação: como criar uma rede social na escola

por Gustavo Freitas

Criar uma rede social na escola. Você já pensou nisso?

Hoje as redes sociais são o grande atrativo das novas tecnologias para os alunos. No celular, no tablet, no computador, eles estão sempre pensando em como acessar as redes sociais. Que tal trazer esse entusiasmo para dentro do ambiente da escola? Que tal criar uma rede social na escola?

Sei que a tarefa de criar uma rede social na escola não é simples e será necessário que toda a escola abrace o projeto, não somente a diretoria e o corpo docente, mas

também os pais e responsáveis e, principalmente, os alunos. Por isso resolvi escrever o artigo para levar a conhecer algumas ferramentas que irão ajudar no processo.

COMO CRIAR UMA REDE SOCIAL NA ESCOLA

Existem várias maneiras de criar uma rede social na escola. As ferramentas computacionais não serão o problema aqui, basta que você encontre alguém com o mínimo de conhecimento para ajudar o projeto a alavancar. E tenho certeza que se o projeto for bem planejado e apresentado

aos pais, algum membro da comunidade certamente saberá instalar e configurar uma das ferramentas necessárias.

PLANEJAMENTO É TUDO

Mais importante que instalar e configurar uma ferramenta computacional para criar a rede social de sua escola é fazer o planejamento de como funcionará e quais os objetivos pedagógicos da proposta. Esse processo de planejamento é essencial e será determinante para o sucesso do projeto e deve ter a participação de pais e alunos.

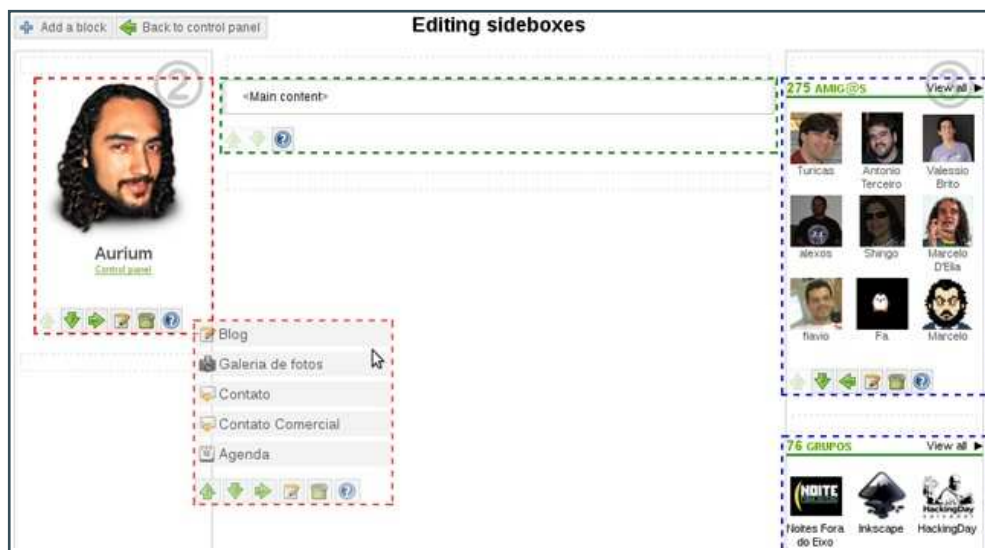
Não adianta criar uma rede social que limita os alunos pois eles não a usarão. Por isso a opinião deles é tão importante. Não adianta criar uma rede social que o corpo docente não entende a função dentro do processo de ensino-aprendizagem. Por isso a opinião dos professores é tão importante. Não adianta criar uma rede social que os pais não participaram da construção. Eles vão pensar que é somente mais uma distração para seus filhos na internet.

TRÊS FERRAMENTAS PARA CRIAR UMA REDE SOCIAL NA ESCOLA

Vou relacionar algumas ferramentas gratuitas que podem ser utilizadas para criar uma rede social na escola.

NOOSFERO

Noosfero é uma plataforma web de código aberto para redes sociais e que ainda permite a criação de blogs, e-Portfolios, CMS, RSS, discussão temática, agenda de eventos, fórum e web chat.



NOOSFERO



ELGG

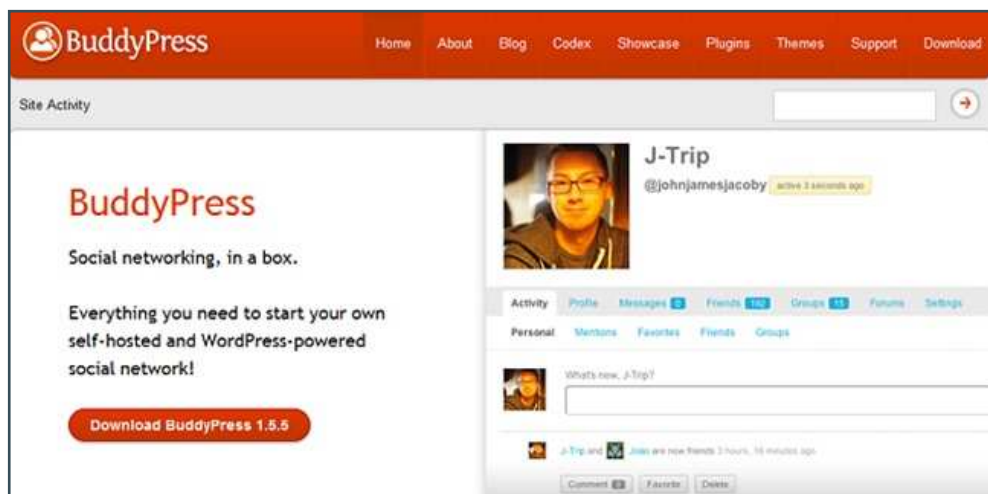
ELGG

Elgg é um premiado motor de código aberto de rede social que fornece uma estrutura robusta sobre a qual você pode criar uma rede social na escola. Fornece uma estrutura robusta sobre a qual pode construir todos os tipos de ambientes sociais, desde um campus da sua universidade, escola ou

faculdade ou uma plataforma de colaboração interna para a sua organização através de uma ferramenta de comunicação que facilita a comunicação entre a sua empresa e os seus clientes. Foi eleito, em 2008, a melhor plataforma aberta de rede social pela Infoworld.

BUDDYPRESS

BuddyPress foi criado em 2008, enquanto o produtor trabalhava para adicionar recursos de redes sociais para um site WordPress. O primeiro lançamento oficial estável foi em Maio de 2009. A plataforma cresceu desde então e transformou-se no pacote dinâmico que hoje está disponível. Assim como WordPress, BuddyPress é uma plataforma totalmente open source. Tudo desde o código, a documentação, temas, plugins e extensões são todos construídos pela comunidade BuddyPress. Isto significa que qualquer um pode ajudar o projeto, contribuindo com o seu tempo e conhecimento. Com BuddyPress você pode criar uma rede social na escola de maneira simples e eficiente.



BUDDYPRESS

CONCLUSÃO

Incluir a TIC na Educação não é uma tarefa simples. Precisa haver planejamento e vontade de fazer diferente. Criar uma rede social na escola não é tão simples, mas pode ser feito e se houver um planejamento e objetivos pedagógicos envolvidos, pode ser uma grande revolução na sua escola. 🌐

Links:

[Noosfero](#)

[Elgg](#)

[BuddyPress](#)

GUSTAVO FREITAS é bacharel em Sistemas de Informação, pós-graduado em planejamento educacional e docência do Ensino Superior, desenvolvedor web, consultor para monetização de blogs, problogger e autor de e-books relacionados a criação e monetização de blogs.



Imagine o fórum oficial da Espírito Livre na sua cidade!

Entre em contato conosco!

Fórum espírito livre

IV Evento de Integração

10110111011001010101101001001010110110110010101101
Win Linux
110100101010110100100101010110110110110010101101
Day

Ambientes de Integração. Evolução Tecnológica

WINLINUX DAY 2012

Por **Nayanne Pacheco** - Acessora de Comunicação | VirtualLink.

Inovação e Qualidade

Foram marcas registradas no grande sucesso que foi a 4ª Edição do Evento de Integração WinLinux Day 2012, que este ano recebeu a honrosa chancela da SBC – Sociedade Brasileira de Computação, e aconteceu no Centro de Convenções e Feiras da Amazônia - Hangar em Belém-PA, nos dias 30 de Novembro e 01 de Dezembro.



Pará é palco de um dos Maiores Eventos Tecnológicos do Brasil

O evento 100% paraense abordou o tema: Ambientes de Integração e Evolução Tecnológica e teve como proposta demonstrar as mais novas Tecnologias e Ambientes de Integração baseados na interoperabilidade entre os sistemas Windows e Linux. De acordo com Alexandre Viana, Diretor da VirtualLink, empresa idealizadora do evento, é necessário que o olhar tecnológico seja expandido. "O Winlinux Day veio para contribuir com o desenvolvimento científico e tecnológico do Pará e Região, estimulando o aprender fazendo! Nos preocupamos em oferecer workshops de qualidade através de profissionais de empresas renomadas do cenário nacional e local como: Kaspesky, Dataprev e VirtualLink, bem como através da participação das comunidades Joomla Brasil e Tá Safo", revelou.



Finalidade

" O Winlinux Day veio para contribuir com o desenvolvimento científico e tecnológico do Pará e Região (...)"

o Alexandre Viana - Diretor VirtualLink

A finalidade do evento foi proporcionar a quebra de paradigmas no que se refere à adoção conjunta destes dois Sistemas, favorecendo na absorção de novos conhecimentos e, ainda, esclarecer dúvidas através da demonstração de Soluções e Tecnologias que existem para estas duas Plataformas.

Matriz: Rua: Joaquim Nabuco, 123 | Nazaré Belém-PA | CEP:66055-300.

Filial: Rua Senador Antônio Lemos, 609-Sala 207 Centro -Castanhal-PA | CEP: 68740-010.

comercial@virtuallink.com.br

www.virtuallink.com.br

Contatos: 91- 3224-7122 | 3241-4474 | 3241-0006

O EVENTO

Foram elaboradas "Ilhas Tecnológicas", um espaço com computadores e soluções tecnológicas onde garantiu-se a interação com todas as tecnologias apresentadas, o que possibilitou uma maior troca de informações, conhecimentos e experiências entre todos os participantes.



Os Participantes interagindo nas Ilhas Tecnológicas



Interação com as Soluções apresentadas



Presença de Autoridades



Workshops com demonstrações práticas



Publico Massivo durante os dois dias de Evento



Equipe VirtualLink, Idealizadora do Evento

A EXPECTATIVA PARA 2013



Na ocasião, Viana esclareceu a motivação para realizar este grandioso evento:

"Nossa expectativa e motivação é que a Região Norte seja reconhecida nacionalmente como uma referência de grandes eventos tecnológicos. Sei que podemos e, por isso, buscamos contribuir com parcela significativa, para a formação de uma Sociedade melhor preparada, informada e instruída, favorecendo diretamente na geração de mão-de-obra qualificada, o que nos leva a crer que devemos sempre acreditar em nossos sonhos", concluiu.

Matriz: Rua: Joaquim Nabuco, 123 | Nazaré
Belém-PA | CEP:66055-300.

Filial: Rua Senador Antônio Lemos,609-Sala 207
Centro -Castanhal-PA | CEP: 68740-010.

comercial@virtuallink.com.br

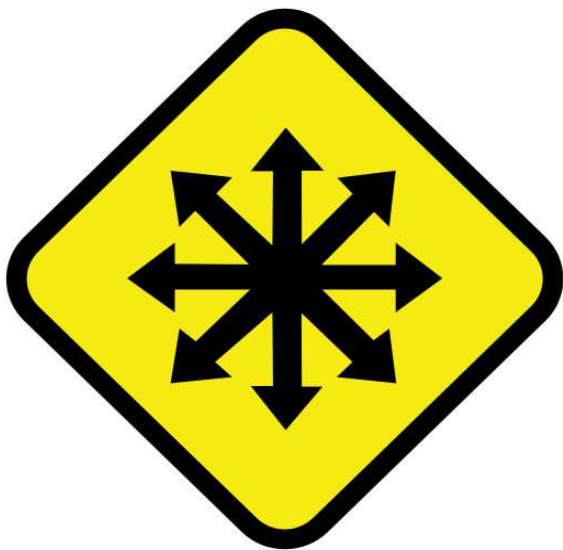
www.virtuallink.com.br

Contatos: 91- 3224-7122 | 3241-4474 | 3241-0006

Quadrinhos

por José James Figueira Teixeira e João Felipe Soares Silva Neto





fisl14

14º Fórum Internacional
Software Livre
A tecnologia que liberta

3 a 6 de julho de 2013 | Centro de Eventos PUCRS
Porto Alegre - RS

PARTICIPE!

Inscrições abertas até 23 de junho
ou no local do evento



Dúvidas e Inscrições
www.fisl.org.br

Realização



Associação
Software Livre.Org



Agenda

Evento: Firefox OS Game Hackathon

Local: São Paulo/SP
Data: 08 e 09/06/2013

Evento: 2º BHack

Local: Belo Horizonte/MG
Data: 22 e 23/06/2013

Evento: 14º Forum Internacional de Software Livre - FISL

Local: Porto Alegre/RS
Data: 03 a 06/07/2013

Evento: V FSLDC - Fórum de Software Livre de Duque de Caxias

Local: Duque de Caxias/RJ
Data: 20/07/2013

Evento: PGBR 2013

Local: Porto Velho/RO
Data: 15 a 17/08/2013

Evento: Consegi 2013

Local: Brasília/DF
Data: 13 a 15/08/2013

Evento: III Fórum de Internet no Brasil

Local: Belém/PA
Data: 03 a 05/09/2013

Evento: CloudConf

Local: São Paulo/SP
Data: 10 e 11/09/2013

Evento: 8º SOLISC

Local: São José/SC
Data: 20 e 21/09/2013

Evento: SFD 2013 - Software Freedom Day

Local: Diversas cidades
Data: 21/09/2013

Evento: Plone Conference 2013

Local: Brasília/DF
Data: 02 a 08/10/2013

Evento: PythonBrasil[9]

Local: Brasília/DF
Data: 02 a 04/10/2013

Evento: LATINOWARE 2013

Local: Foz do Iguaçu/PR
Data: 16 a 18/10/2013

Evento: V FTSL

Local: Curitiba/PR
Data: 24 e 25/10/2013

agenda / entre aspas



Quando as pessoas temem o governo, isso é tirania. Quando o governo teme as pessoas, isso é liberdade.



Thomas Jefferson (13 de abril de 1743 – 4 de julho de 1826) foi o terceiro presidente dos Estados Unidos, e o principal autor da declaração de independência daquele país.

Fonte: Quem Disse

REVISTA

espírito livre

LIBERDADE E
INFORMAÇÃO

Liberdade e
compartilhamento
de informação e
conhecimento

A Revista Espírito Livre é uma
publicação construída também
através da colaboração dos leitores.

Tecnologia

Software Livre

GNU/Linux

Redes

LibreOffice

Opinião

Entrevistas

E muito mais

Então

Não fique para trás!
Colabore!



Entre em
contato conosco.

revista@espiritolivre.org

Acesse a edição mensal gratuita:
<http://revista.espiritolivre.org>
E confira!

