

Escolhendo software livre em sistemas embarcados

Motivos para se escolher software livre em sistemas embarcados

Michael Opdenacker

Free Electrons

<http://free-electrons.com>

Tradução para o português:
Marcelo Barros de Almeida
(marcelobarrosalmeida@yahoo.com.br)

Criado com [OpenOffice.org 2.x](#)

Escolhendo software livre em sistemas embarcados

© Copyright 2006, Michael Opdenacker

Tradução: Marcelo Barros de Almeida

Usa Creative Commons -Licença ShareAlike 2.0

<http://free-electrons.com>

15 de Set de 2009



Porque você vale a pena

Cuidados intensivos no sistema

Acaba com bugs para sistemas críticos

Gera uma sensação de liberdade imediata

Redução visível dos custos após a primeira aplicação



Direitos de cópia



Créditos - ShareAlike 2.0

Você é livre para

- copiar, distribuir, apresentar e executar trabalhos
- fazer trabalhos derivados
- fazer uso comercial deste trabalho

Sob as seguintes condições

Créditos. Você deve dar crédito ao autor original.

Compartilhe do mesmo modo. Se você alterar, mudar, ou realizar trabalhos usando este como base, você deve redistribuir o trabalho resultante sob uma licença idêntica a esta.

 Para qualquer reuso ou distribuição você deve deixar claro os termos de licença deste trabalho.

• Qualquer uma destas condições podem ser abandonadas se você obtiver uma permissão do detentor dos direitos autorais.

Faça uso justo e o direitos dos outros não serão afetados de forma alguma pelas restrições acima.

Texto da licença:

<http://creativecommons.org/licenses/by-sa/2.0/legalcode>

© Copyright 2006
Michael Opdenacker
michael@free-electrons.com

Documentos originais, atualizações e traduções:
<http://free-electrons.com/articles/reasons>

Correções, sugestões e traduções
são bem vindas!



Escolhendo software livre em sistemas embarcados

© Copyright 2006, Michael Opdenacker

Tradução: Marcelo Barros de Almeida

Usa Creative Commons -Licença ShareAlike 2.0

<http://free-electrons.com>

15 de Set de 2009



Melhor visualizado com...

Este documento é melhor visualizado com um leitor de PDF recente ou com o próprio [OpenOffice.org](#)!

- ▶ Aproveite os links internos e externos. Não hesite em clicar neles!
- ▶ Encontre páginas rapidamente através da procura automática
- ▶ Use miniaturas para navegar no documento de forma rápida

Se você está lendo uma cópia em papel ou HTML, você deveria obter a sua cópia em formatos PDF ou [OpenOffice.org](#) em <http://free-electrons.com/articles/reasons>!



Conteúdo

Sistemas embarcados: vantagens do software livre (SL)

- ▶ Introdução
- ▶ Vantagens para desenvolvedores de sistemas embarcados
- ▶ Vantagens para gerentes de desenvolvimento de sistemas embarcados
- ▶ Vantagens para empresas de sistemas embarcados
- ▶ Vantagens para provedores de conteúdo
- ▶ Vantagens para clientes e usuários
- ▶ Desafios com o software livre



Escolhendo software livre em sistemas embarcados

Introdução



Sobre sistemas embarcados

Definição da [Wikipedia](#) : “um computador de propósito especial, que é completamente encapsulado pelo dispositivo que controla”.

- ▶ Nunca significando ser um computador tradicional de propósito geral
- ▶ Exemplos: celulares, equipamentos para redes, multimídia, transporte, controle industrial...
- ▶ Usa um processador de propósito geral (os mais populares são: [arm](#), [x86](#), [ppc](#)) ou ainda um micro-controlador mais simples.

Veja http://en.wikipedia.org/wiki/Embedded_system



Dura competição no mercado de embarcados

- ▶ Em sistemas embarcados, sistemas com SL abocanham uma porção significativa do mercado de sistemas proprietários tradicionais.
- ▶ Windows CE também.
- ▶ Os maiores concorrentes agora são Linux e Windows CE (posição em 2005, dispositivos coletados por <http://linuxdevices.com> e <http://windowsforddevices.com>)

Cool devices, by embedded OS -- Round Four Scorecard

(click each quantity to view devices)

Device Category	Windows Embedded	Embedded Linux
PDAs, handhelds	120 devices	41 devices
Mobile phones	55 devices	30 devices
VoIP phones/devices	13 devices	15 devices
Robots	(included in other)	11 devices
Audio/video devices	23 devices	68 devices
Thin client devices	45 devices	28 devices
Tablets/webpads	40 devices	14 devices
Gateways, servers, APs	(included in other)	91 devices
Other	52 devices	69 devices
TOTAL:	348 devices	367 devices

Veja <http://linuxdevices.com/articles/AT6743418602.html>

Escolhendo software livre em sistemas embarcados

© Copyright 2006, Michael Opdenacker

Tradução: Marcelo Barros de Almeida

Usa Creative Commons -Licença ShareAlike 2.0

<http://free-electrons.com>

15 de Set de 2009



Sistemas embarcados livres

Reusam vários componentes empregados na computação tradicional:

- ▶ Aproveitam toda a vantagem da modularidade e versatilidade de sistemas embarcados livres para desktops e servidores.
- ▶ Bibliotecas compartilhadas: biblioteca C, criptografia, compressão, processamento de dados e formatos, ferramentas gráficas...
- ▶ Kernel do sistema operacional: gerenciamento de processos e memória, protocolos de suporte (como redes), sistemas de arquivos, suporte a arquiteturas e dispositivos (processadores, barramentos, etc)...
- ▶ Ferramentas de desenvolvimento: compiladores, depuradores...



Kernels livres para sistemas embarcados - Linux

<http://kernel.org>

- ▶ O mais popular.
Comunidade de desenvolvimento muito grande e ativa.
- ▶ Suporta várias arquiteturas, dispositivos e protocolos.
- ▶ Requisitos mínimos: processador de 32 bits, 2 MB de RAM, 1 MB de espaço de armazenamento (pode caber até em 500 KB!).
- ▶ Licença: GNU GPL
Código do kernel: use com cuidado para evitar violações de licença.
Excelente para usuários Linux também (mais código livre!).



Kernels livres para sistemas embarcados - uClinux

<http://uclinux.org/> (pronuncia-se como “You See Linux”)

- ▶ Linux para micro-controladores, processadores sem MMU (Unidade de Gerenciamento de Memória).
- ▶ Usado principalmente para sistemas embarcados menores e de baixo custo.
- ▶ Kernel: originalmente um kernel Linux derivado. Agora cada vez mais integrado no ramo principal do Linux.
- ▶ Aplicações: aplicações padrões para Linux embarcado com *patches* para atender restrições especiais de memória.

Muito próximo do Linux embarcado tradicional.



Kernels livres para sistemas embarcados - NetBSD

<http://netbsd.org>

- ▶ Sistema BSD altamente portável.
Suporta várias arquiteturas e placas embarcadas!
- ▶ Comunidade menor mas muito ativa!
- ▶ Requisitos mínimos de armazenamento e RAM similares aos do Linux.
- ▶ Licença: BSD. Sem problemas de violação de licença mas não requer que o código dos drives seja livre (alguns drivers mantidos proprietários?).

Veja <http://foss.in/2005/slides/netbsd-linux.pdf>
para um boa comparação entre Linux e NetBSD



Escolhendo software livre em sistemas embarcados

© Copyright 2006, Michael Opdenacker

Tradução: Marcelo Barros de Almeida

Usa Creative Commons -Licença ShareAlike 2.0

<http://free-electrons.com>

15 de Set de 2009



Kernels livres para sistemas embarcados - eCos

<http://ecos.sourceforge.org/>



- ▶ Sistema embarcado de tempo real bem leve, provido pela Red Hat / Cygnus solutions.
- ▶ Compatível com a maioria das aplicações Unix e Linux.
- ▶ Para sistemas bem pequenos: suporta processadores de 16 bits e algumas centenas de KB de RAM e armazenamento (como 300 KB) são mais do que suficientes.
- ▶ Kernel + aplicação podem caber até mesmo em 50KB (sistema mínimo)!
- ▶ Licença: GPL (com pequenas modificações)



Computadores vs. Sistemas Linux embarcados

GNU Tradicional / Sistema Linux



Navegador web, escritório, multimídia...



ls, vi, wget, ssh, httpd, gcc...

libjpeg, libstdc++, libxml, libvorbis...



Biblioteca GNU C



Kernel Linux

Kernel completo com a maioria das características e com *drivers* para todo tipo de hardware de PC do planeta!!

Sistema Linux embarcado



Interface personalizada

busybox
(ls, vi, wget, httpd...)
dropbear (ssh)...

Gráficos,
navegador web,
servidor de web.

Implementações
muito mais leves!
Sem ferramentas de
desenvolvimento.

libjpeg, libstdc++, libxml, libvorbis...

uClibc

Muito mais leve
do que a
biblioteca C GNU!



Kernel Linux

Kernel leve, somente
com as características
necessárias e *drivers*

Interface
com o usuário

Utilitários de
linha de
comando

Bibliotecas
compartilhadas

Biblioteca C

Kernel

Escolhendo software livre em sistemas embarcados

© Copyright 2006, Michael Opdenacker

Tradução: Marcelo Barros de Almeida

Usa Creative Commons -Licença ShareAlike 2.0

<http://free-electrons.com>

15 de Set de 2009



Licenças em sistemas embarcados livres

► Kernel Linux: GPL

Drivers proprietários (fornecidos separadamente) são cada vez menos tolerados.

Não impõe restrições de licença em bibliotecas ou aplicações.

► Biblioteca C: LGPL

Permite criar aplicações proprietárias

► Bibliotecas compartilhadas:

LGPL / BSD: aplicações proprietárias permitidas
ou GPL: aplicações proprietárias não permitidas

► Aplicações:

Do tipo BSD: podem ser proprietárias

GPL: podem co-existir (sem modificações) com aplicações proprietárias.
Mas é necessário fornecer o código fonte para o usuário.

**Repleta de espaço
para aplicações
proprietárias ou
específicas do
sistema!**



O modelo de negócio GNU GPL

- ▶ Um paradoxo?
Com GNU GPL você pode ter mais lucros do que com licenças “mais liberais” !
- ▶ Bibliotecas comerciais: Qt da Trolltech, MiniGUI da Feynman...
Licença: Dupla GPL / proprietária
Para desenvolver software proprietário, é necessário pagar por licença especial para os detenedores dos direitos..
- ▶ Bibliotecas da comunidade: GTK, SDL, libvorbis...
A maioria disponível sob licenças LGPL ou BSD.



Escolhendo software livre em sistemas embarcados



Vantagens para desenvolvedores de sistemas embarcados





Inovação e valor agregado

- ▶ Vários componentes prontos para o uso em grande parte do sistema. Permite que o foco seja na parte inovadora do produto, naquilo que o diferencia.
- ▶ Você não precisa esperar por meses ou anos para que alguma características seja implementada por alguém. Pelo menos, você pode implementar as características críticas que você precisa.
- ▶ Inovação constante. As características evoluem a passos rápidos, algumas vezes antes mesmo que você precise delas!
- ▶ Possível portar o Linux para uma nova arquitetura inovadora. Fácil portar todo o seu sistema então.

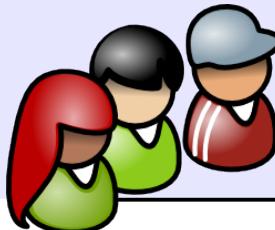




Qualidade de Software

- ▶ Desenvolvedores de componentes chaves (GNU, kernel Linux, uClibc, busybox, servidores http...) são programadores excelentes.
- ▶ O código é também por outros programadores excelentes. Código ruim não pode ser escondido. Auditorias independentes para descoberta de vulnerabilidades é possível (e realizada nos projetos maiores).
- ▶ Com software proprietários, o desenvolvimento de software é completamente fechado. O código pode ser criado por times de programadores que trabalham em turnos, 24 /7, apressados para escrever o máximo de código possível.
- ▶ Você pode estimar a qualidade de um dado componente através do retorno do usuário em lista de emails da comunidade.

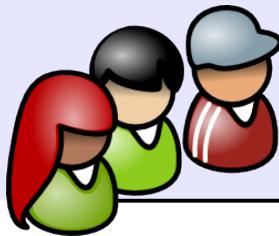




Controle

- ▶ Os desenvolvedores estão no controle!
- ▶ Liberdade irrestrita para uso do software, para qualquer finalidade, por quanto tempo for necessário.
- ▶ Liberdade irrestrita para tomar decisões a respeito do sistema. Sem atualizações obrigadas por causa de um fornecedor de um sub-sistema.
- ▶ Sem caixas pretas no sistema que não podem ser corrigidas, modificadas ou mesmo melhoradas, com problemas que admitem apenas remendos.
- ▶ Liberdade para escolher entre várias alternativas técnicas.

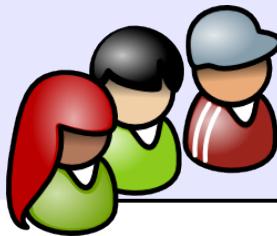




Flexibilidade

- ▶ Várias aplicações de terceiros disponíveis
- ▶ Tudo no sistema pode ser personalizado, da interface do usuário ao kernel.
- ▶ Comparado com soluções proprietárias, é muito mais fácil de integrar com sistemas diferentes e de adaptar a necessidades específicas.
- ▶ Scripts e encadeamento de comandos básicos.
Podem realizar tarefas bastante sofisticadas a um custo muito baixo.





Escolhas

- ▶ Várias possibilidades de escolha para a mesma funcionalidade (por exemplo, vários servidores de http disponíveis). Pode-se realmente escolher o mais próximo dos requerimentos.
- ▶ Possibilidade de trocar um componente por outro que faça a mesma função, mesmo em etapas posteriores de desenvolvimento.
- ▶ Muitas ferramentas de desenvolvimento e teste. Pode-se escolher a que melhor se adapta a forma de trabalho.
- ▶ Escolha do S.O. para desenvolvimento: GNU/Linux, Unix e até mesmo Windows! Sem necessidade de trocar o S.O. Do desktop.





“Closer to the metal”

“Use the Source, Luke”

- ▶ Com sistemas proprietários você não tem certeza de que não existirão outros processos fazendo ações que não irão interferir nas sua aplicação.
- ▶ Acesso aos fontes e interfaces de baixo nível: você pode obter um entendimento detalhado de qualquer parte do sistema, sem “caixas pretas”. Junto com o conhecimento a respeito do seu próprio sistema, chances reais de encontrar a causa principal dos problemas.

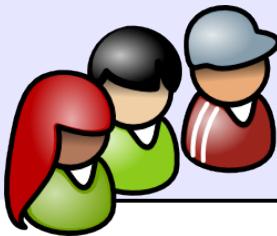




Facilidade de desenvolvimento

- ▶ Pode-se desenvolver aplicações para o desktop primeiro, usando as mesmas ferramentas de desenvolvimento. Você não perderá nada quando mover para a plataforma embarcada desejada.
- ▶ Padrões: Interfaces do Kernel GNU/Linux e bibliotecas desenvolvidas com um foco forte em conformidade aos padrões. Seus produtos estarão de acordo com padrões e serão interoperáveis com outros padrões também.

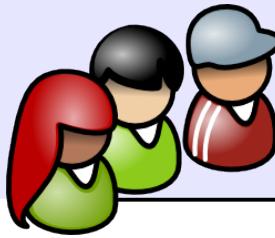




Mais fácil de escrever o código

- ▶ Fontes disponíveis: muitos exemplos acessíveis tornam mais fácil aprender como as coisas funcionam e como escrever o código.
- ▶ Mais fácil de escrever o código: você pode modificar e reusar código existente se está desenvolvendo com uma licença compatível. Particularmente útil para criar coisas similares (device driver, interface gráfica...)





Recursos e suporte disponíveis

- ▶ Comunidades enormes de desenvolvedores e usuários. Muitas pessoas com vontade de ajudar e compartilhar experiências.
- ▶ Muitos recursos on-line disponíveis: documentação, HOWTOs, fórum, lista de emails. Use a sua ferramenta de busca web preferida para encontrá-los!
- ▶ Mais fácil para compartilhar experiências com outros usuários de outras empresas. Fornecedores de software proprietário tendem a isolar seus clientes, de forma que a única fonte de informação são seus vendedores.

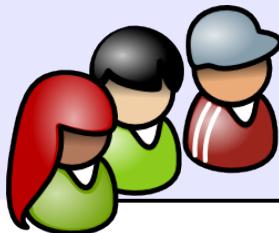




Desenvolvimento aberto

- ▶ Você tem acesso direto aos desenvolvedores. Muito útil para levantar novas questões! Chances reais para influenciar em suas escolhas.
- ▶ A chance de acessar os desenvolvedores não depende do tamanho da empresa em que você trabalha.
- ▶ Possibilidade de conversas técnicas profundas com os desenvolvedores, investigando questões, avaliando soluções, obtendo comentários valiosos, experiência e realimentação.
- ▶ Acesso a todos os arquivos das listas de email de desenvolvimento. Pode-se entender porque as decisões foram tomadas.

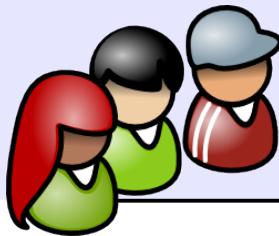




Aprendizagem valiosa

- ▶ Menos APIs proprietárias para aprender, que você não irá usar novamente com outro produto.
- ▶ O que você aprende não fica obsoleto na próxima versão de software ou S.O.. Com Unix e GNU/Linux, você ainda pode usar o que aprendeu 15 anos atrás.
- ▶ Você pode usar a mesma ferramenta de desenvolvimento quando se mudar para outra companhia. Sua habilidade não perde valor.

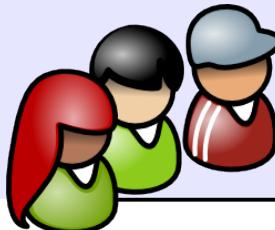




Propriedade do trabalho

- ▶ Sistemas proprietários: ao final do período de avaliação, você tem que pagar ou perder tudo que implementou.
- ▶ Contribuições para projetos abertos: você não tem que implementar tudo de novo quando se mudar para outra companhia.

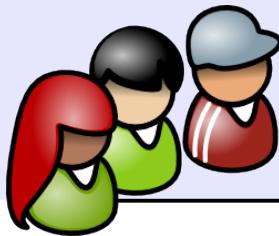




Custo

- ▶ Custos reduzidos graças a adição somente do custo do desenvolvimento
- ▶ Custo zero de software
 - ▶ Facilidades para aprovação da gerência, pelo menos para avaliações.
 - ▶ Permite que se use as mesmas ferramentas in casa e no escritório.
 - ▶ Excelente para pessoas que desenvolvem sistemas embarcados como hobby.

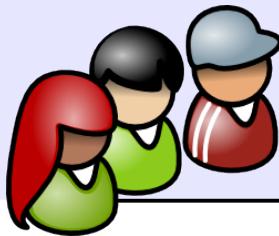




Liberdade e iniciativa

- ▶ Dispensa a aprovação da gerência para a construção de um protótipo ou demo.
- ▶ Sem: compra de software, requisições de versões de avaliação, necessidade de registro, horas perdidas decifrando termos de contrato ...
- ▶ Você pode começar o projeto imediatamente e colocar as idéias em ação.

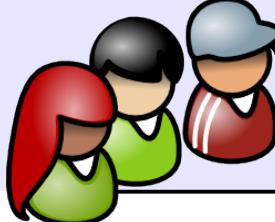




Menos chatices legais

- ▶ Muitas ferramentas são disponíveis sob um pequeno número de licenças bem conhecidas e examinadas. FAQs disponíveis.
- ▶ Software proprietário sempre vem com seus próprios termos de licenciamento. Não é necessário perder tempo tentando entendê-los ou procurando por armadilhas antes de aceitá-lo.
- ▶ Licenças S.L. são mais fáceis de se entender. A GPL foi escrita por um programador (e então revista por um advogado).





Vantagens para o desenvolvedor - Resumo

- ▶ Inovação, valor agregado
- ▶ Qualidade de software
- ▶ Controle
- ▶ Flexibilidade
- ▶ Escolhas
- ▶ “Closer to the metal”
- ▶ Facilidade de desenvolvimento
- ▶ Mais fácil de escrever código
- ▶ Suporte e recursos
- ▶ Desenvolvimento aberto
- ▶ Aprendizagem valiosa
- ▶ Propriedade do trabalho
- ▶ Custo
- ▶ Liberdade e iniciativa
- ▶ Menos chatices legais



Escolhendo software livre em sistemas embarcados



Vantagens para gerentes de sistemas embarcados





Controle (1)

- ▶ Decisões técnicas não dependem dos fornecedores de software.
- ▶ Você possui os dispositivos que cria.
Precisa ser capaz de fazer qualquer escolha a respeito deles.
- ▶ Sistemas operacionais proprietários embarcados possuem grandes corporações por trás, impondo seus termos, preços e escolhas técnicas. Pode haver um conflito de interesses.

“Você deseja perder seu tempo tentando obter correções do seu fornecedor de S.O. já que estas correções não trazem retorno para eles ? ”

Escolhendo software livre em sistemas embarcados

© Copyright 2006, Michael Opdenacker

Tradução: Marcelo Barros de Almeida

Usa Creative Commons -Licença ShareAlike 2.0

<http://free-electrons.com>

15 de Set de 2009





Controle (2)

- ▶ Atualizações de software com fornecedores proprietários: eles podem mudar ou revogar seus termos de licença, deixando-o preso a sua versão atual.
- ▶ Sem problemas para manter sua aplicação proprietária (a menos que você use bibliotecas específicas com licença Copyleft).
Isto é o que adiciona valor ao seu sistema. É sua a decisão.
- ▶ Escolha de fornecedores e vendedores para o mesmo produto. Qualquer um pode entrar no mercado de suporte, treinamento e desenvolvimento. Fornecedores tem que tentar ser os melhores. Eles não possuem nada que você dependa.





Custo efetivo

- ▶ Sem royalties para pagar
- ▶ Reuso máximo de código. Recursos podem ser focados somente no valor adicionado ao sistema, gerando um diferencial em relação aos concorrentes.
- ▶ Menor tempo para chegar ao mercado.





Propriedade

- ▶ Desde que você respeite os termos da licença S.L., ninguém jamais será capaz de revogar seus direitos para continuar a usar a sua plataforma ou ferramentas. Sem possibilidade de perder o seu investimento devido a “violações de licença”.
- ▶ Propriedade plena da plataforma quando ela em produção.





Gerenciamento de software facilitado

- ▶ Sem necessidade de gerenciar compra de software. Os projetos podem começar imediatamente tão logo a decisão seja tomada.
- ▶ Sem necessidade de gerenciar prestação de contas de royalties.
- ▶ Sem necessidade de procurar por software pirata usado pelos funcionários.
- ▶ Sem necessidade de revisão das licenças de software pela equipe advogados repetidas vezes para cada software de terceiros.
- ▶ Menos tempo gasto com equipe de vendas de software tentando vender mais e mais produtos. Sem necessidade de lutar por descontos e fazer comprometimentos de compra de outros produtos.





Custos de aprendizagem

- ▶ Reuso de software: força de trabalho experiente fácil de encontrar. Curvas de aprendizagem mais longas com software proprietário.
- ▶ Facilidade de participar das tomadas de decisões com o seu time. Muitos artigos técnicos, análises e fontes de informação disponíveis na Internet.



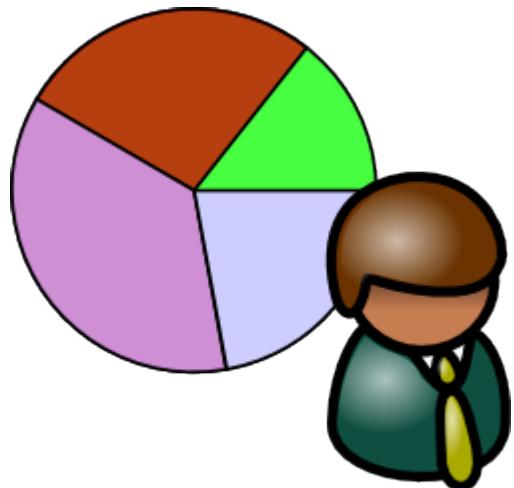


Vantagens para gerentes - Resumo

- ▶ Controle
- ▶ Custo efetivo
- ▶ Propriedade
- ▶ Gerenciamento de software facilitado
- ▶ Custos de aprendizagem

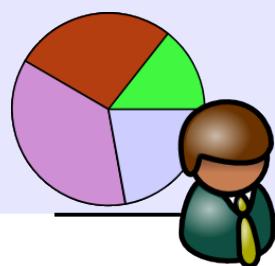


Escolhendo software livre em sistemas embarcados



Vantagens para empresas de sistemas embarcados

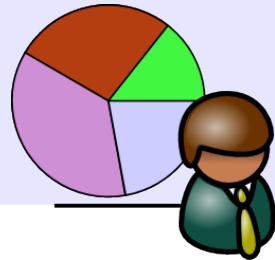




Vantagens para empresas (1)

- ▶ Inclua todas as vantagens mencionadas para funcionários e gerentes: controle, inovação, custo, tempo para colocar o produto no mercado ...
- ▶ Menos trabalho para o departamento de compras de software. Sem a necessidade de ter uma única ferramenta corporativa (pelo menos não por razões de custo).
- ▶ Se você escolhe Windows CE: Microsoft é um concorrente potencial com dinheiro suficiente para entrar em qualquer mercado. Se eles possuem o S.O. que o seu produto concorrente roda, você está em problemas!
- ▶ Custos de litígio: fabricantes pequenos de sistemas embarcados não podem arcar com disputas judiciais com grandes fornecedores de software.

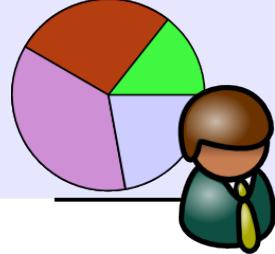




Vantagens para empresas (2)

- ▶ Construir dispositivos abertos baseados S.L. pode gerar uma imagem muito positiva da sua empresa para a comunidade de desenvolvedores de S.L. Facilita a atração de desenvolvedores talentosos.
- ▶ S.L. garante que seus investimentos de logo prazo são seguros e podem lhe levar adiante.





Vantagens para empresas - Resumo

- ▶ Controle, inovação, custo, tempo para chegar ao mercado
- ▶ Garante investimentos de longo prazo
- ▶ Evita concorrência de fornecedores de software
- ▶ Evita custos de processos judiciais proibitivos
- ▶ Habilidade para atrair talentos



Vantagens para os fornecedores de conteúdo



- ▶ Vários padrões, players compatíveis no mercado para tocar o que eles produzem: reduz o custo de produção.
- ▶ Eles podem criar e testar sua produção com ferramentas usando os mesmos codecs ou bibliotecas que estão nos players embarcados.
- ▶ Codecs sem patentes ou royalties, (como Ogg Vorbis ou Ogg Theora): sem taxas para criação de conteúdo.



Escolhendo software livre em sistemas embarcados



Vantagens para consumidores e usuários finais





Vantagens para usuários finais (1)

- ▶ Propriedade dos dados: graças aos formatos abertos e ao S.L., as pessoas preservam a capacidade de abrir os arquivos que possuem, por quanto tempo for necessário. Muito útil quando se está trocando o hardware.
- ▶ Qualidade e confiabilidade: a maioria dos componentes do sistema já foram usados por várias pessoas em dispositivos diferentes e a sua confiabilidade foi provada.
- ▶ Segurança e privacidade: S.L. componentes são submetidos a verificações por várias pessoas e organizações. Não é possível esconder suas falhas ou armadilhas por um tempo longo.





Vantagens para usuários finais (2)

- ▶ Dispositivos cheios de características e uso agradável (fontes *anti-aliased*, menus transparentes...). Componentes S.L. tem um projeto menos óbvio ou falhas no uso.
- ▶ Dispositivos com sistemas proprietários: frequentemente oferecem uma escolha limitada de software proprietário de baixa qualidade ou freeware.
- ▶ Consumidores: eles estão no controle. Com os fontes, eles podem ajustar o sistema para as suas necessidades. Capacidade de modificar ou adicionar software (se o dispositivo permite atualização de firmware).





Vantagens para usuários finais - Resumo

- ▶ Propriedade dos dados graças graças aos formatos abertos e ao S.L.
- ▶ Qualidade e confiabilidade
- ▶ Segurança e privacidade
- ▶ Dispositivos cheios de características e uso agradável.
Biblioteca de software relativamente grande.
- ▶ Sistemas personalizados (se atualização de firmware estiver disponível).



Escolhendo software livre em sistemas embarcados

Desafios com Software Livre



Escolhendo software livre em sistemas embarcados

© Copyright 2006, Michael Opdenacker

Tradução: Marcelo Barros de Almeida

Usa Creative Commons -Licença ShareAlike 2.0

<http://free-electrons.com>

15 de Set de 2009



Mitos a respeito de S.L.

- ▶ “Serei forçado a compatilhar o meu código com o mundo todo.”
Errado: sem problemas para criar aplicações proprietárias, exceto se forem usadas bibliotecas Copyleft (raras) ou modifica um programa Copyleft.
- ▶ “Não existe suporte nem treinamento.”
Errado: existem inclusive várias empresas competindo por suporte!



Desafios técnicos

- ▶ Mudança de gerenciamento: atualizações e versões frequentes de S.L. (“alvo móvel”). Críticas a congelar componentes cedo demais.
- ▶ Frequentemente precisa-se escolher entre várias soluções. Falta de tempo dos desenvolvedores para investigar soluções e seguir as novidades do S.L. Consultores especializados podem ajudar!
- ▶ Necessidade de escolher soluções duradouras. Necessidade de estimar o tamanho da comunidade de desenvolvedores e usuários, quão ativo é o desenvolvimento, a importância da tecnologia subjacente, disponibilidade de prestadores de serviço ...
- ▶ Frequentemente muitos recursos e documentação espalhados. Requer experiência em procura.



Fraquezas do Linux

- ▶ Linus é um sistema operacional de propósito geral, suportando sistemas que vão do pequeno ao enorme.
- ▶ Desta forma, ele pode não ser ótimo e pode não ser capaz de bater uma solução personalizada bem ajustada.
“Com a generalidade vem a ineficiência”.
- ▶ Linux pode requerer um quantia significativa de ajustes pra fazê-lo suportar os requerimentos específicos do seu sistema embarcado.
- ▶ Por outro lado, não ser ótimo pode ser valorizado devido aos custos menores de desenvolvimento.



Questões legais

- ▶ Ausência de jurisprudência sobre a validade da licença GPL em países diferentes. A GPL não menciona ou não pode mencionar as leis aplicáveis em todos os países.
- ▶ Questões relacionadas a mistura de código com licenças diferentes.
- ▶ Quão legítimas são as licenças ? Tem certeza de que os licenciadores são realmente os verdadeiros proprietários? Eles também tem o consentimento dos seus empregadores ? Muito dependente da lei local também. Projetos principais como GNU ou Linux estão agora muito cuidadosos com as contribuições.
- ▶ Patentes de software (USA e Japão): implementações abertas mas ainda taxa de patentes para pagar !



Atração excessiva!

Cuidado: seus engenheiros podem ter um julgamento tendencioso!

- ▶ S.L. é realmente atrativo. Ele tem características excelentes e muitos engenheiros são bastante familiares com algumas de suas soluções.
- ▶ Software como Linux tem cada vez mais energia em sistemas embarcados. Muitas pessoas estão inclinadas a usá-lo ou experimentá-lo.
- ▶ Desta forma, muitos engenheiros irão ter problemas em ser objetivos no processo de decisão.

Tenha certeza de que você ponderou todos os fatos, custos, prós e contras, cada caso é um caso !



Desafios com S.L. - Resumo

- ▶ Soluções genéricas podem não ser ótimas. Ajustes requeridos. Entretanto, diminui custos de desenvolvimento.
- ▶ Questões legais: validade da GPL no país local, código misto, legitimidade dos donos da propriedade, patentes de software.
- ▶ Excesso de atração: engenheiros desejam usar S.L., mesmo quando ele não é a melhor solução.



Sites úteis

LinuxDevices.com: <http://linuxdevices.com>

- ▶ Boletim semanal com novidades e anúncios sobre dispositivos embarcados rodando Linux.
- ▶ Artigos, relatórios e catálogo de dispositivos com Linux embarcado.
- ▶ Um site excelente para seguir as novidades da indústria.
- ▶ Também útil para descobrir detalhes a respeito de soluções que os concorrentes adotam!



Escolhendo software livre em sistemas embarcados

© Copyright 2006, Michael Opdenacker

Tradução: Marcelo Barros de Almeida

Usa Creative Commons -Licença ShareAlike 2.0

<http://free-electrons.com>

15 de Set de 2009



Sobre o autor

O autor deste documento

- ▶ Tem uma experiência exaustiva tanto com soluções proprietárias quanto abertas
- ▶ Tem feito uma comparação realmente objetiva e independente entre as duas abordagens
- ▶ Sem nenhum interesse em nenhum tipo de progresso no S.L.



Daí, você pode confiar cegamente neste relatório!



Como ajudar

Se você apóia este documento, você pode ajudar ...

- ▶ Enviando correções, sugestões e traduções
- ▶ Enviando mais idéias: mais vantagens e desafios
- ▶ Requerendo a sua organização para contratar sessões de treinamento realizadas pelo autor destes documentos (veja <http://free-electrons.com/training>)
- ▶ Falando a respeito dele para seus amigos, colegas e comunidade de software livre local.
- ▶ Adicionando links para o nosso material on-line no seu website, aumentando a nossa visibilidade nos resultados de máquinas de busca.



Agradecimentos

- ▶ Ao projeto OpenOffice.org, por suas ferramentas de apresentação e processamento de texto que satisfizeram todas a minhas necessidades.
- ▶ Aos contribuintes do projeto <http://openclipart.org> pelos belos cliparts de domínio público
- ▶ Aos membros de toda a comunidade de Software Livre, por compartilharem o melhor deles: seu trabalho, seu conhecimento, sua amizade.
- ▶ A Bill Gates, por nos deixar tanto espaço para inovação !

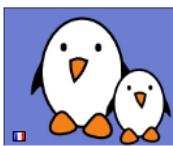
As pessoas que ajudaram, enviaram correções ou sugestões:

<Tenha seu nome adicionado aqui!>





Related documents



Free Electrons
Embedded Freedom

HOME DEVELOPMENT SERVICES TRAINING DOCS COMMUNITY COMPANY BLOG

Recent blog posts

ELC Europe in Grenoble

Free Electrons at ELC

Linux kernel 2.6.29 - New

features for embedded
users

The Buildroot project

begins a new life

FOSDEM 2009 videos

USB-Ethernet device for
Linux

Program for Embedded
Linux Conference 2009

announced

Public session changes

Real hardware in our
training sessions

Call for presentations for
the LSM embedded track

Docs

Most of the below documents are presentations used in our [training sessions](#), or in technical conferences.

License

 All our documents are available under the terms of the [Creative Commons Attribution-ShareAlike 3.0 license](#). This essentially means that you are free to download, distribute and even modify them, provided you mention us as the original authors and that you share these documents under the same conditions.

Linux kernel

- [Embedded Linux kernel and driver development](#)
- [New features in Linux 2.6 \(since 2.6.10\)](#)
- [Kernel initialization](#)
- [Porting Linux to new hardware](#)
- [Power management in Linux](#)
- [Linux PCI drivers](#)
- [Block device drivers](#)
- [Linux USB drivers](#)
- [DMA](#)

Architecture specific documents

- [ARM Linux specifics](#)
- [Linux on TI OMAP processors](#)

Embedded Linux system development

- [Embedded Linux system development](#)
- [Real time in embedded Linux systems](#)
- [Block filesystems](#)
- [Flash filesystems](#)
- [Free software development tools](#)
- [The U-boot bootloader](#)
- [The GRUB bootloader](#)
- [The blob bootloader](#)
- [Hotplugging with udev](#)
- [Introduction to uClinux](#)
- [Java in embedded Linux](#)
- [Embedded Linux optimizations](#)
- [Audio in embedded Linux systems](#)
- [Multimedia in embedded Linux systems](#)
- [Embedded Linux From Scratch... in 40 minutes!](#)
- [Building embedded Linux systems with Buildroot](#)
- [Developing embedded distributions with OpenEmbedded](#)
- [The Scratchbox development environment](#)

Miscellaneous

- [Introduction to the Unix command line](#)
- [SSH](#)
- [Linux virtualization solutions \(with an embedded perspective\)](#)
- [Advantages of Free Software and Open Source in embedded systems](#)
- [Introduction to GNU/Linux and Free Software](#)

All our technical presentations
on <http://free-electrons.com/docs>

- ▶ Linux kernel
- ▶ Device drivers
- ▶ Architecture specifics
- ▶ Embedded Linux system development



How to help

You can help us to improve and maintain this document...

- ▶ By sending corrections, suggestions, contributions and translations
- ▶ By asking your organization to order development, consulting and training services performed by the authors of these documents (see <http://free-electrons.com/>).
- ▶ By sharing this document with your friends, colleagues and with the local Free Software community.
- ▶ By adding links on your website to our on-line materials, to increase their visibility in search engine results.

Linux kernel

Linux device drivers
Board support code
Mainstreaming kernel code
Kernel debugging

Embedded Linux Training

All materials released with a free license!

Unix and GNU/Linux basics
Linux kernel and drivers development
Real-time Linux, uClinux
Development and profiling tools
Lightweight tools for embedded systems
Root filesystem creation
Audio and multimedia
System optimization

Free Electrons

Our services

Custom Development

System integration
Embedded Linux demos and prototypes
System optimization
Application and interface development

Consulting and technical support

Help in decision making
System architecture
System design and performance review
Development tool and application support
Investigating issues and fixing tool bugs

