

Software Livre e de Código Aberto

Fernando Castor

Centro de Informática – Universidade Federal de Pernambuco

Faculty of Information and Computing Sciences – Utrecht University



Alguns direitos reservados

Exemplos de Software Livre e de Código Aberto



O que essas organizações tem em comum?



Software Livre

O que é Software Livre?

Ideologicamente

- ▶ Software cujos usuários têm liberdade para
 - ▶ executar
 - ▶ copiar
 - ▶ distribuir
 - ▶ estudar
 - ▶ modificar
 - ▶ melhorar
- ▶ Usuários **controlam** os programas e não o contrário
- ▶ **Premissa básica:** **liberdade** é uma **coisa boa** para indivíduos e sociedade
 - ▶ **Movimento Social**

O que é Software Livre?

Estritamente falando

Um programa é **livre** se seus usuários têm as 4 liberdades essenciais:

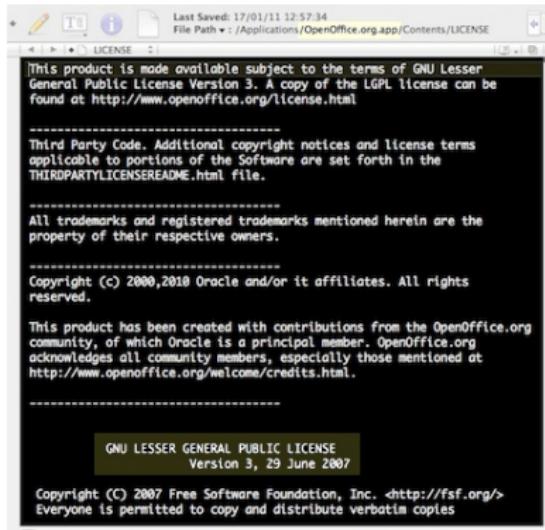
- 0 Liberdade de **executar** o programa, para qualquer fim.
- 1 Liberdade para **estudar** o funcionamento do programa e **modificá-lo** para fazer o que você **desejar**.
- 2 Liberdade para **redistribuir** cópias.
 - ▶ Assim, você ajuda seu vizinho.
- 3 Liberdade para **distribuir** cópias de suas **versões modificadas**.
 - ▶ Assim, a comunidade toda se beneficia com suas modificações.

Acesso ao código fonte é precondição para as liberdades 1 e 3.

O que é Software Livre?

Legalmente

- Software **distribuído** de acordo com uma **licença** que respeita as 4 liberdades



Quem é esse?

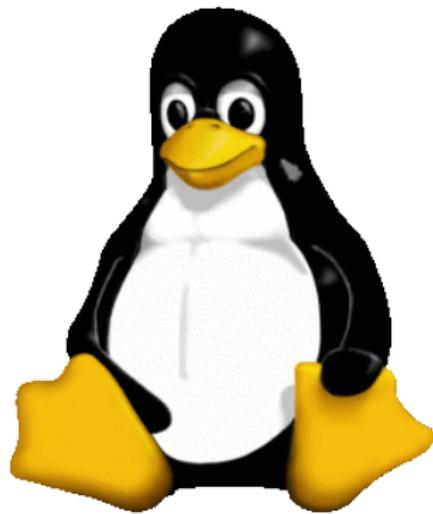


Richard M. Stallman

Este programa é software Livre?



Este programa é software Livre?



Este programa é software Livre?



(considerando apenas a versão gratuita)

Software de Código Aberto

O que é Software de Código Aberto?

Ideologicamente

O foco do movimento Open Source **não é ideológico**.

Em poucas palavras...

- ▶ “Campanha de Marketing” em prol do Software Livre
 - ▶ Com foco em vantagens práticas
 - ▶ Sem viés ideológico
 - ▶ Para chamar a atenção de grandes corporações

Por que “Código Aberto”?

- ▶ A palavra “free” em *Free Software* é problemática
 - ▶ “*Free as in freedom, not free beer*”



- ▶ Com “Código Aberto” a ênfase fica em algo mais palpável, o código

A criadora do termo “Código Aberto”



Christine Peterson

Possivelmente os dois maiores evangelistas do movimento em seus primeiros anos



Eric Raymond e Bruce Perens

Código aberto não significa apenas acesso ao código fonte

Práticas

1. Desenvolvimento distribuído
2. Depuração em massa, com usuários envolvidos no desenvolvimento
3. Software disponível cedo e frequentemente
4. Meritocracia
5. Sem uma hierarquia rígida e sem coerção
6. Abertura do código fonte é uma pré-condição

Software de Código Aberto também tem uma definição

 Open Source Initiative

Search this site:

Navigation

- About the OSI
- ▽ The Open Source Definition
 - Annected
 - License Review Process
 - License Prioritization
 - Open Source Licenses
 - Trademarks and Logo Usage
 - FAQ
 - Open Standards
 - Open Source Education
 - Making Inns
 - Getting Help
 - Donate to the OSI
 - OSI Individual Membership
 - OSI Affiliate Membership
 - Contact OSI
 - Terms of Service
 - OSI Corporate Sponsors

The Open Source Definition

Introduction

Open source doesn't just mean access to the source code. The distributor terms of open-source software must comply with the following criteria:

- 1. Free Redistribution**

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
- 2. Source Code**

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost; preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.
- 3. Derived Works**

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
- 4. Integrity of The Author's Source Code**

The license may restrict source code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
- 5. No Discrimination Against Persons or Groups**

The license must not discriminate against any person or group of persons.
- 6. No Discrimination Against Fields of Endeavor**

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for generic research.
- 7. Distribution of License**

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
- 8. License Must Not Be Specific to a Product**

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed either the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.
- 9. License Must Not Restrict Other Software**

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.
- 10. License Must Be Technology-Neutral**

No provision of the license may be predicated on any individual technology or style of interface.



OpenSource.org site content is licensed under a [Creative Commons Attribution 3.0 License](#) | [Terms of Service](#)

“Open Source” é mais popular que “Free Software”

Microsoft®



 **Alibaba Group**

 **amazon**

ORACLE®

Uber

 **IBM**



Google

Quem é esse?



Quem é esse?



Linus B. Torvalds

Uma palavrinha sobre o Linux...

Primeiras versões baseadas no MINIX

Disponível para o público desde outubro de 1991

A importância do Linux

- ▶ Núcleo usado no GNU (GNU/Linux)
- ▶ Feito notável de colaboração

Atualmente: Mais de 23000 desenvolvedores contribuindo

Informações interessantes: <https://www.openhub.net/p/linux>

- ▶ Principal exemplo de software livre que alcançou o **grande público**

Distribuições do Linux



Diferenças e Semelhanças entre FS e OSS

Em pouquíssimas palavras...

A democracia é um direito moral

vs

A democracia é o sistema mais
eficiente que se conhece

Aspectos Econômicos de Software Livre e de Código Aberto

Eu posso vender software livre? E de código aberto?

Resposta curta: **SIM!**

Na prática

- ▶ Não faz sentido
- ▶ E cobrar por instalações em máquinas, pode?
- ▶ Restringir uso comercial?

Quem contribui?

- ▶ Voluntários/entusiastas/nerds
- ▶ Entidades acadêmicas/governamentais
- ▶ e...

Corporações!!!!



Free as in free beer?

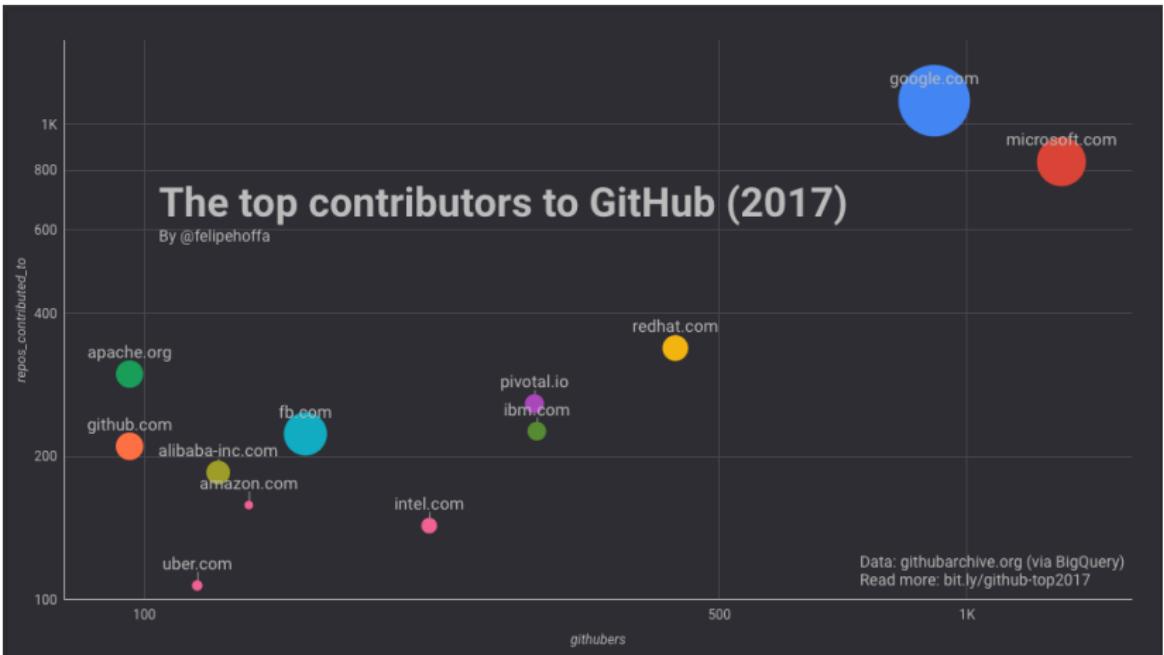
- ▶ A IBM gastou, entre 1995 e 2005, US\$ 1bi no Linux
 - ▶ Para fins de desenvolvimento e promoção

Free as in free beer?

- ▶ A IBM gastou, entre 1995 e 2005, US\$ 1bi no Linux
 - ▶ Para fins de desenvolvimento e promoção
- ▶ E, mais recentemente...

The screenshot shows a news article from WIRED magazine. The header is green with the WIRED logo and navigation links for BUSINESS, CULTURE, GEAR, IDEAS, and SCIENCE. The main title of the article is "IBM Buying Open Source Specialist Red Hat for \$34 Billion". Below the title is a photograph of IBM CEO Ginni Rometty speaking on stage. The caption below the photo reads: "IBM CEO Ginni Rometty in Germany in June. © KRISZTIAN BODI/BLOOMBERG/GETTY IMAGES".

IBM JUST SPENT \$34 billion to buy a software company that gives away its primary product for free.



Free as in free beer?

- ▶ A Oracle (graças a **aquisições**) **era** dona, em 2010, de
 - ▶ MySQL
 - ▶ Java

Free as in free beer?

- ▶ A Oracle (graças a **aquisições**) **era** dona, em 2010, de
 - ▶ MySQL
 - ▶ Java
- ▶ O Google desenvolveu um dos melhores Web Browsers
 - ▶ E **abriu ele**, via projeto Chromium
 - ▶ Incluindo o **engenho V8** para JavaScript

A propósito, falando em Google...



Mais recentemente...



- ▶ 5+ anos de desenvolvimento (primeiro *commit* em 2010!)
- ▶ *open source um ano e meio depois de seu lançamento*

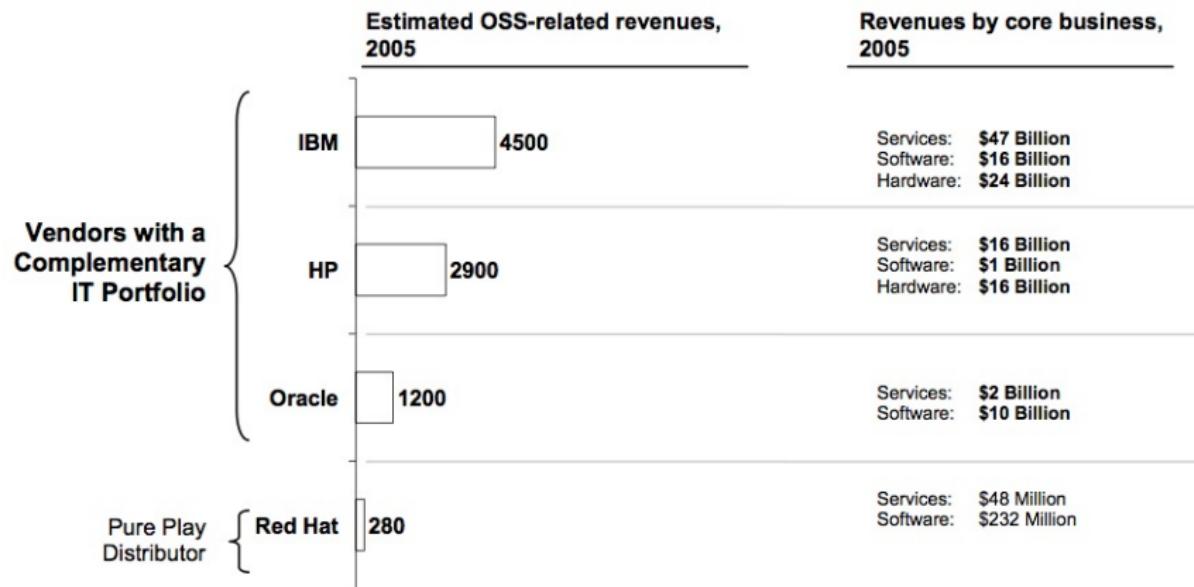


Receita relacionada a FLOSS vs. receita por área

Corporações não tem fins humanitários...

Receita relacionada a FLOSS vs. receita por área

Corporações não tem fins humanitários...





Como consumidores consomem mídia

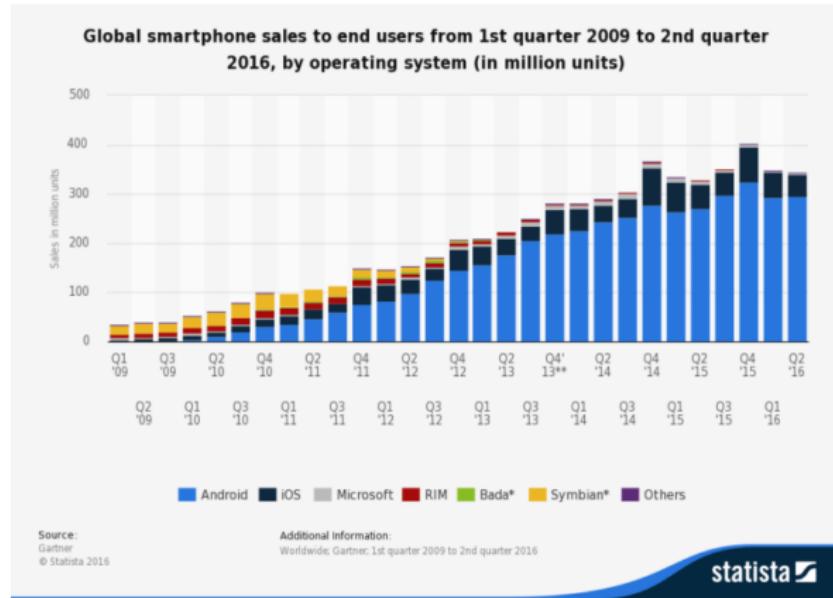
Dados de Out/2014. Google comprou a Android Inc. em 2005.

HOW CONSUMERS SPEND MEDIA TIME (HH:MM) EACH MONTH

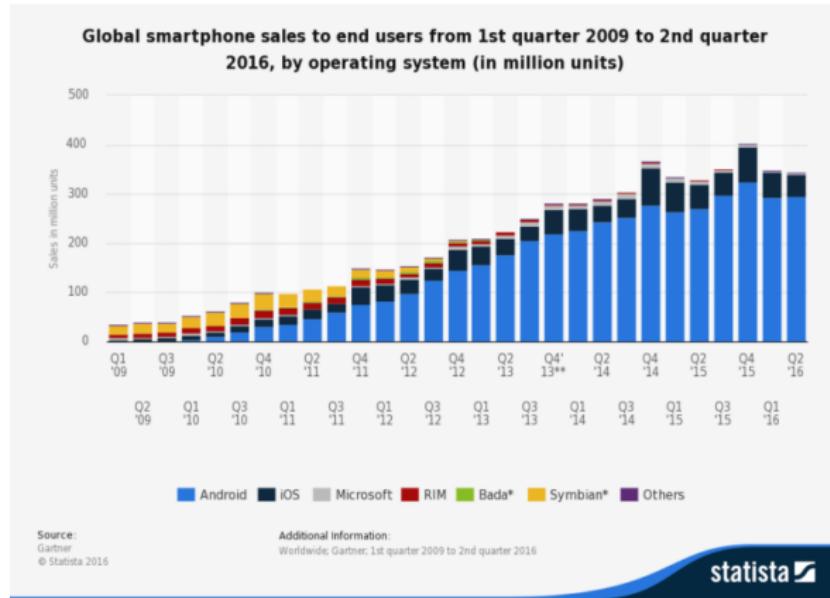
— + CHANGE SINCE 2012 — - CHANGE SINCE 2012



Smartphones com o sistema Android



Smartphones com o sistema Android

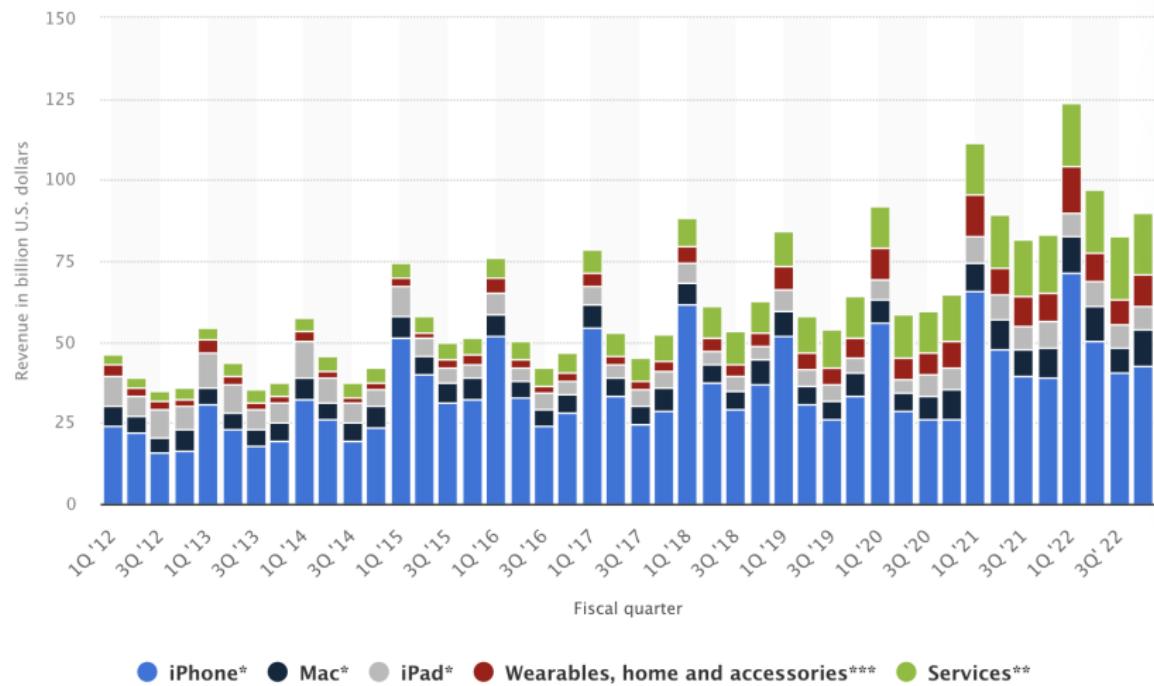


Cada um desses smartphones...

- ▶ funciona bem com as ferramentas do Google, incluindo Play Store;
- ▶ coleta informações que o Google pode usar.
- ▶ Em 2021: mais de 3 bilhões de dispositivos Android!

Fontes de receita da Apple

(in billion U.S. dollars)



● iPhone* ● Mac* ● iPad* ● Wearables, home and accessories*** ● Services**

Dois pontos de vista complementares

Software pode ou não ser um diferencial de negócios

1. Software livre não é a principal fonte de receita para IBM, HP, Google ou Apple
2. Mas **ajuda a dividir custos e riscos** para produtos que **não** são um diferencial de negócios

Dois pontos de vista complementares

Software pode ou não ser um diferencial de negócios

1. Software livre não é a principal fonte de receita para IBM, HP, Google ou Apple
2. Mas **ajuda a dividir custos e riscos** para produtos que **não** são um diferencial de negócios
3. Se software não é a diferenciação, é um centro de **custo**, não de **lucros**

Dois pontos de vista complementares

Software pode ou não ser um diferencial de negócios

1. Software livre não é a principal fonte de receita para IBM, HP, Google ou Apple
2. Mas **ajuda a dividir custos e riscos** para produtos que **não são um diferencial de negócios**
3. Se software não é a diferenciação, é um centro de **custo**, não de **lucros**

Tecnologia meio vs. tecnologia fim

1. Software pode ser uma tecnologia “possibilitadora”
2. Desenvolvimento de software proprietário é uma **minoria**
 - Menos de 10% das vagas para desenvolvimento de software
 - Mais de 70% estão em empresas que são “usuárias de TI”

Se tiverem que aprender apenas uma coisa, aprendam isso

Vale a pena abrir o código para

1. software que **não é um diferencial** de negócios
2. software que é uma **tecnologia possibilitadora** para vários negócios

Porque os custos e o risco do desenvolvimento são divididos

Como Você Pode se Beneficiar?

Software livre é grátis

Pode aumentar sua empregabilidade!

Education



Experience

- *September 2009 - December 2009.* Travelled from Japan to England without flying.
- *December 2008 - September 2009.* Mathematical Systems, Tokyo: statistical software consultancy. Member of Knowledge Engineering team doing custom development in Common Lisp for NTT.
 - Monitoring system for large numbers (100k) of routers, with Holt-Winters' seasonal exponential time series forecasting to detect aberrations.
 - Visualisation of large (cyclic) directed graphs by condensing nodes while attempting to preserve the graph structure.
 - Efficient storage and retrieval of routing tables.
 - Portable persistent memory-mapped object database integrating into Lisp object system with the Meta-Object Protocol (MOP), including a precise computing garbage collector and atomic transactions (*transact!*).
- *January 2006 - November 2008.* Travelled in the Middle East, South East Asia, China, North Korea, India, and Japan.
- *October 2004 - December 2005.* Alphamosaic (bought by Broadcom), Cambridge: semiconductor company specialising in low-power video. Engineer on the software research team.
 - Embedded C software development; system-wide and video codec performance optimization, dynamic DMA resource management infrastructure, lightweight reliable mutual exclusion without malloc, H.264 video decode (dramatically increased performance of an already heavily optimized system, directly benefiting the video iPod project), virtual file system layer, device drivers.
 - Algorithm evaluation for hardware design: image sensor pipeline (image processing in Matlab), and simulated performance of different i-cache and d-cache sizes and associativities.
 - On-site customer support at Samsung on DMB (mobile TV) phone project. Starting with a dysfunctional project, rebuilt trust with Samsung.
 - High-level ASIC design and software emulation of components for next generation platform (JPEG acceleration).
- *Winter 2003.* Volunteer at NGO wifi ISP in Mozambique.
- *Easter 2003.* Hospital in Bethlehem. Volunteer IT system administrator. Continued regularly until 2007.
- *2002-2004.* Trinity College Students' Union.
- *2001-2002.* Cambridge University Computing Society. Chairman.
- *Summer 2000.* Linux Information Systems AG, Munich: productivity software start-up. Intern.
- *Spring 2000.* Linux kernel. Contributor. USB driver for a scanner; accepted and integrated into the Linux kernel (`/drivers/usb/microlink.c`). Usenpace program to assist reverse engineering (`usb-robo!`).
- *1998.* Unreleased game framework in C++ with dynamic code generation.

Recent hobby projects

- `teepedee2`: a high-performance web application platform in Common Lisp, directly calling Linux system calls. Benchmarked serving more than 10k dynamically generated pages per second on one processor core.
- `dysfunkycom`, ranked seventh out of about three hundred teams in the ICFP 2009 programming competition.

Pode aumentar sua empregabilidade!

Education

Experience

- *September 2009 - December 2009.* Travelled from Japan to England without flying.
- *December 2008 - September 2009.* Mathematical Systems, Tokyo: statistical software consultancy. Member of Knowledge Engineering team doing custom development in Common Lisp for NTT.
 - Monitoring system for large numbers (100k) of routers, with Holt-Winters' seasonal exponential time series forecasting to detect aberrations.
 - Visualisation of large (cyclic) directed graphs by condensing nodes while attempting to preserve the graph structure.
 - Efficient storage and retrieval of routing tables.
 - Portable persistent memory-mapped object database integrating into Lisp object system with the Meta-Object Protocol (MOP), including a precise compacting garbage collector and atomic transactions (*mainly*).
- *January 2008 - November 2008.* Travelled in the Middle East, South East Asia, China, North Korea, India, and Japan.
- *October 2004 - December 2005.* Alphamosaic (bought by Broadcom), Cambridge: semiconductor company specialising in low-power video. Engineer on the software research team.
 - Embedded C software development; system-wide and video codec performance optimization, dynamic DMA resource management infrastructure, lightweight reliable mutual exclusion without malloc, H.264 video decode (dramatically increased performance of an already heavily optimized system, directly benefiting the video iPod project), virtual filesystem layer, device drivers.
 - Algorithm evaluation for hardware design: image sensor pipeline (image processing in Matlab), and simulated performance of different i-cache and d-cache sizes and associativities.
 - On-site customer support at Samsung on DMB (mobile TV) phone project. Starting with a dysfunctional project, rebuilt trust with Samsung.
 - High-level ASIC design and software emulation of components for next generation platform (JPEG acceleration).
- *Winter 2003.* Volunteer at NGO with ISP in Mozambique.
- *Easter 2003.* Hospital in Bethlehem. Volunteer IT system administrator. Continued regularly until 2007.
- *2002-2004.* Trinity College Students' Union.
- *2001-2002.* Cambridge University Computing Society. Chairman.
- *Summer 2000.* Linux Information Systems AG, Munich: productivity software start-up. Intern.
- **Spring 2000.** Linux kernel. Contributor. USB driver for a scanner; accepted and integrated into the Linux kernel (`/drivers/usb/microtek.c`). Userspace program to assist reverse engineering (`usb-robot`).
- *1998.* Unreleased game framework in C++ with dynamic code generation.

Recent hobby projects

- `teepedee2`: a high-performance web application platform in Common Lisp, directly calling Linux system calls. Benchmarked serving more than 10k dynamically generated pages per second on one processor core.
- `dysfunkycom`, ranked seventh out of about three hundred teams in the ICFP 2009 programming competition.

- *Spring 2000.* Linux kernel. Contributor. USB driver for a scanner; accepted and merged (drivers/usb/microtek.c). Userspace program to assist reverse engineering (usb-ri).

Há muitas oportunidades para trabalhar com FLOSS!

Screenshot of the Indeed.com job search interface for "Open source" jobs.

The search bar shows "What Open source" and "Where City, state, zip code, or "remote"".

A tip message says: "Tip: Enter your city or zip code in the "where" box to show results in your area."

Filter options include: Date posted, Remote, Salary estimate, Job type, Encouraged to apply, Location, Company, Experience level, and Education.

Job listing for "Data Entry Clerk - Remote" at The Bynes Company Office Solutions, LLC:

- Sort by: relevance - date
- 75,486 jobs
- Estimated \$32.7K - \$41.4K a year
- Part-time
- Description: Prepares source data for computer entry by compiling and sorting information.

Job listing for "Open Source Targeter Analyst" at Leidos:

- Leidos ★★★★☆ 1,386 reviews
- McLean, VA 22102
- \$84,500 - \$175,500 a year - Full-time
- You must create an Indeed account before continuing to the company website to apply
- Apply on company site

Sua empresa pode economizar!

BLACK DUCK SOFTWARE ESTIMATES DEVELOPMENT COST OF OPEN SOURCE SOFTWARE AT \$387 BILLION



Reuse of Open Source Software Could Save U.S. Companies \$22 Billion a Year

WALTHAM, Mass., April 14, 2009 - Black Duck Software, a leading provider of products and services for accelerating software development through the managed use of open source software (OSS), estimates the total development cost of open source software at more than \$387 Billion U.S.

To put this figure into perspective, the cost to develop open source software code is equivalent to nearly 50 percent of the U.S. government's recently passed stimulus bill, the \$787 billion "American Recovery and Reinvestment Act."

Black Duck Software has the industry's most comprehensive database of open source software and related metadata. According to the company's research, there are over 200,000 open source projects representing over 4.9 billion lines of code. Using its detailed knowledge of open source projects and applying standard industry cost estimation techniques, Black Duck estimates that the total development cost of OSS exceeds \$387 billion and represents a collective investment of more than two million developer years.

An additional analysis, which estimates that 10 percent of IT application development spending is redundant with existing open source projects, indicates U.S. companies could realize savings of more than \$22 billion a year through the reuse of OSS in application development.

Sua empresa pode economizar!

BLACK DUCK SOFTWARE ESTIMATES DEVELOPMENT COST OF OPEN SOURCE SOFTWARE AT \$387 BILLION



Reuse of Open Source Software Could Save U.S. Companies \$22 Billion a Year

WALTHAM, Mass., April 14, 2009 - Black Duck Software, a leading provider of products and services for accelerating software development through the managed use of open source software (OSS), estimates the total development cost of open source software at more than \$387 Billion U.S.

To put this figure into perspective, the cost to develop open source software code is equivalent to nearly 50 percent of the U.S. government's recently passed stimulus bill, the \$787 billion "American Recovery and Reinvestment Act."

Black Duck Software has the industry's most comprehensive database of open source software and related metadata. According to the company's research, there are over 200,000 open source projects representing over 4.9 billion lines of code. Using its detailed knowledge of open source projects and applying standard industry cost estimation techniques, Black Duck estimates that the total development cost of OSS exceeds \$387 billion and represents a collective investment of more than two million developer years.

An additional analysis, which estimates that 10 percent of IT application development spending is redundant with existing open source projects, indicates U.S. companies could realize savings of more than \$22 billion a year through the reuse of OSS in application development.

Fonte de dados para pesquisa

conf.researchr.org/home/msr-2022

co-located with ICSE 2022 Write a Blog

MAY 18-20 VIRTUAL, MAY 23-24 IN-PERSON, 2022, PITTSBURGH, PA, USA

**19TH INTERNATIONAL CONFERENCE ON
MINING SOFTWARE REPOSITORIES (MSR 2022)**

Attending ▾ Program ▾ Tracks ▾ Organization ▾ Q Search Series ▾ Sign in Sign up

David Lawrence Convention Center
Pittsburgh, PA, USA

Skim **MSR 2022 Tracks**

Centro de Informática **Utrecht University**

Obrigado!

Resumo

- ▶ (0) executar, (1) estudar e modificar, (2) distribuir e (3) distribuir versões modificadas
- ▶ Software Livre e Software de Código Aberto são dois lados da mesma moeda
- ▶ Vale a pena abrir o código para
 1. software que **não é um diferencial** de negócios
 2. software que é uma **tecnologia possibilitadora** para vários negócios
- ▶ Há muitas formas de se beneficiar profissionalmente com FLOSS

Slides: https://github.com/fernandocastor/software_livre

Licença



Você tem o direito de:

- Compartilhar — copiar e redistribuir o material em qualquer formato
- Adaptar — remixar, transformar, e criar a partir do material

De acordo com os seguintes termos:

Attribution — Você deve atribuir o devido crédito, fornecer um link para a licença e indicar se foram feitas alterações. Você pode fazê-lo de qualquer forma razoável, mas não de uma forma que sugira que o licenciante o apoia ou aprova o seu uso.

NonCommercial — Você não pode usar o material para fins comerciais.

ShareAlike — Se você remixar, transformar ou criar a partir do material, tem de distribuir suas contribuições sob a mesma licença

Mais informações em <http://creativecommons.org/licenses/by-nc-sa/3.0/br/>