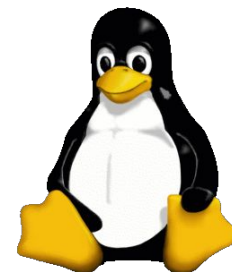


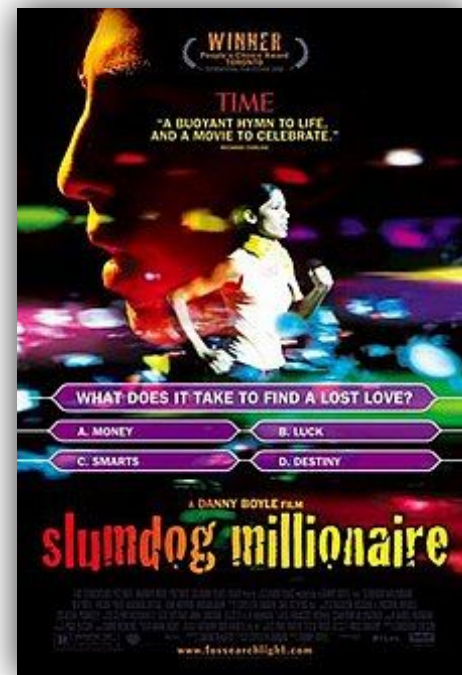
É difícil contribuir para o kernel Linux?

Melissa Wen

22 de Outubro de 2022



Quem quer ser um desenvolvedor do kernel Linux?



Fonte: imdb.com



Motivação



Fonte: rottentomatoes.com



Motivação - Os 13 porquês

Pessoais

Nunca ficar entediado: variedade de uso

Acredita em software livre

Quer conhecer pessoas novas

Ter seu talento reconhecido

Se divertir



Motivação - Os 13 porquês

Pessoais

Nunca ficar entediado: variedade de uso

Acredita em software livre

Quer conhecer pessoas novas

Ter seu talento reconhecido

Se divertir

Profissionais

Boa remuneração

Turbinar a carreira

Feedback instantâneo dos usuários



Freddie Mercury 1986



Motivação - Os 13 porquês

Pessoais

Nunca ficar entediado: variedade de uso

Acredita em software livre

Quer conhecer pessoas novas

Ter seu talento reconhecido

Se divertir

Profissionais

Boa remuneração

Turbinar a carreira

Feedback instantâneo dos usuários

Inovação

Colaboração técnica de alto nível

Aprender coisas novas

Tecnologia de ponta de SO

Novas ideias

Melhorias constantes



Primeiros passos (1)

- **Git e Linguagem C**
- **Fluxo de submissão de patches:**
<https://www.kernel.org/doc/html/latest/process/submitting-patches.html>
- **Estrutura de subsistemas**
 - Fluxo das contribuições
 - Mantenedores e repositórios
 - Modelos de manutenção
 - Único mantenedor
 - Delegação
 - Múltiplos committers



Primeiros passos (2)

- **Baixar um kernel do upstream**

Mainline: <https://github.com/torvalds/linux>

- **Instalar e compilar um kernel customizado**

<https://flusp.ime.usp.br/kernel/Kernel-compilation-and-installation/>

- **Experimentar mudanças no kernel**

https://flusp.ime.usp.br/kernel/play_with_modules/

- **Leia notícias e blog posts**

<https://lwn.net/>

<https://planet.freedesktop.org/>



The Linux Kernel documenter x +

docs.kernel.org

Gmail

The Linux Kernel

6.1.0-rc1

Search docs

A guide to the Kernel Development Process

Submitting patches: the essential guide to getting your code into the kernel

Code of conduct

Kernel Maintainer Handbook

All development-process docs

Core API Documentation

The Linux driver implementer's API guide

Kernel subsystem documentation

Locking in the kernel

Linux kernel licensing rules

How to write kernel documentation

Development tools for the kernel

Kernel Testing Guide

Kernel Hacking Guides

Linux Tracing Technologies

fault-injection

Kernel Livepatching

Rust

The Linux kernel user's and administrator's guide

The kernel build system

Reporting issues

User-space tools

The Linux kernel user-space API guide

» The Linux Kernel documentation

View page source

The Linux Kernel documentation

This is the top level of the kernel's documentation tree. Kernel documentation, like the kernel itself, is very much a work in progress; that is especially true as we work to integrate our many scattered documents into a coherent whole. Please note that improvements to the documentation are welcome; join the linux-doc list at [vger.kernel.org](mailto:linux-doc@kernel.org) if you want to help out.

Working with the development community

The essential guides for interacting with the kernel's development community and getting your work upstream.

- [A guide to the Kernel Development Process](#)
- [Submitting patches: the essential guide to getting your code into the kernel](#)
- [Code of conduct](#)
- [Kernel Maintainer Handbook](#)
- [All development-process docs](#)

Internal API manuals

Manuals for use by developers working to interface with the rest of the kernel.

- [Core API Documentation](#)
- [The Linux driver implementer's API guide](#)
- [Kernel subsystem documentation](#)
- [Locking in the kernel](#)

Development tools and processes

Various other manuals with useful information for all kernel developers.

- [Linux kernel licensing rules](#)
- [How to write kernel documentation](#)
- [Development tools for the kernel](#)
- [Kernel Testing Guide](#)

Leia a documentação



É difícil contribuir para o kernel Linux?
Melissa Wen, 22 de Outubro de 2022



Hello World

- **Corrigindo bugs**
 - Seus próprios problemas
 - O problema dos outros
 - Procurando encrenca
- **Revisando contribuições**
- **Testando ou reportando bugs**
- **Documentando**



Hello World

Programas para novos contribuidores - desenvolvimento upstream

- Google Summer of Code
- Outreachy
- Igalia Coding Experience
- XOrg EVoC
- etc.



Hello World

Programas para novos contribuidores - desenvolvimento upstream

- Google Summer of Code
 - Verão do hemisfério norte
 - Remuneração regional
 - Primeira experiência em OSS
 - Múltiplos projetos de software upstream
 - Contribuidor e mentor propõem um projeto



Hello World

Programas para novos contribuidores - desenvolvimento upstream

- Outreachy
 - Duas rodadas por ano
 - Remuneração fixa
 - Etapa preparatória de envio de patches
 - Foco em diversidade



Hello World

Programas para novos contribuidores - desenvolvimento upstream

- Igalia Coding Experience
 - Projetos na área de participação da empresa
 - Remuneração fixa
 - Inscrições no verão do hemisfério norte
 - Flexibilidade para início e dedicação
 - Foco em diversidade e primeiras contribuições a OSS



Hello World

Programas para novos contribuidores - desenvolvimento upstream

- XOrg EVoC
 - Projetos na pilha gráfica do Linux
 - Remuneração fixa
 - Sem período específico para início e fim
 - Para estudantes



A comunidade

- Desenvolvedores com diferentes backgrounds
 - Profissionais e disciplinadas
- Distribuída
 - Geograficamente
 - Profissionalmente (empresas)
- Lista de discussões é o ponto de encontro
- Sim, fazemos amizades!



A comunidade



Fonte: br.pinterest.com



Fazer parte da comunidade

- Comunicação escrita: e-mails e blog posts
 - Seja educado e cuidado com o tom
 - Não seja agressivo ou irônico
 - Respeite o trabalho dos outros
 - Divulgue seu trabalho
 - "Show me the code"



Fazer parte da comunidade

- Habilidade de sintetizar mudanças e atrair revisores
- Comece com pequenas mudanças
- Tome seu tempo para estudar (para sempre)



Fazer parte da comunidade

- Habilidade de sintetizar mudanças e atrair revisores
- Comece com pequenas mudanças
- Tome seu tempo para estudar (para sempre)
- Resista



Fonte: revistaquem.globo.com



Desafios

"Dracarys"



Fonte: HBO



Desafios

- **Perfis tóxicos**
- **Comunicação escrita**
- **Curva de aprendizado íngreme**
- **Sobrecarga de trabalho dos mantenedores**
 - Maior resistência a mudança
 - Diferentes critérios de aceitação: automação e documentação
- **Retenção de conhecimento**



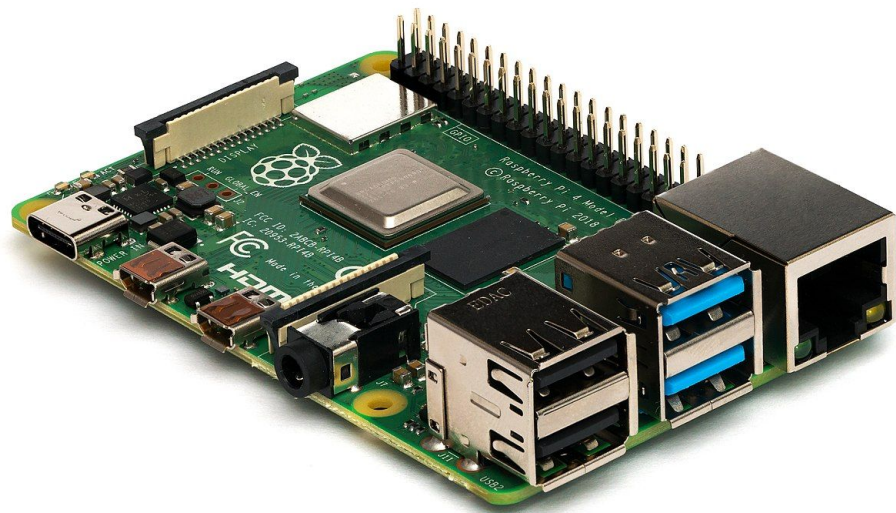
Você quer jogar um jogo?



Desenvolver drivers de GPU

- **Subsistema DRM (o DRM do bem)**
 - Drivers de GPU
 - Interface com o espaço do usuário
- **Múltiplas fabricantes de GPU**
 - Intel, AMD, Broadcom, ARM, Qualcomm, etc.
- **Falta de acesso externo a especificações**
- **Entender a pilha gráfica do Linux**





É difícil contribuir para o kernel Linux?
Melissa Wen, 22 de Outubro de 2022







DRM - Direct Rendering Manager

- **Renderização 3D**
 - Drivers do Mesa
 - OpenGL / Vulkan
- **Display - Tela**
 - Ajustes: resolução, rotação, correção de cor...
 - Atualização de frames
 - Interfaces de vídeo
- **Gerenciamento de memória**
 - Escalonador



DRM - Comunidade

- **Manutenção em grupo**
 - Repositório central
 - Repositório de mantenedores das fabricantes
- **Comunidade muito ativa**
 - Mailing list: dri-devel (<https://lists.freedesktop.org/mailman/listinfo/dri-devel>)
 - IRC: #dri-devel on OFTC
 - XOrg Foundation: <https://www.x.org/wiki/>
 - XDC: <https://www.youtube.com/c/XOrgFoundation>



Mas afinal, é difícil?





É difícil contribuir para o kernel Linux?
Melissa Wen, 22 de Outubro de 2022









É difícil contribuir para o kernel Linux?
Melissa Wen, 22 de Outubro de 2022



