



KernelDevDay Warmup

FLUSP: FLOSS at USP



FLUSP: FLOSS at USP

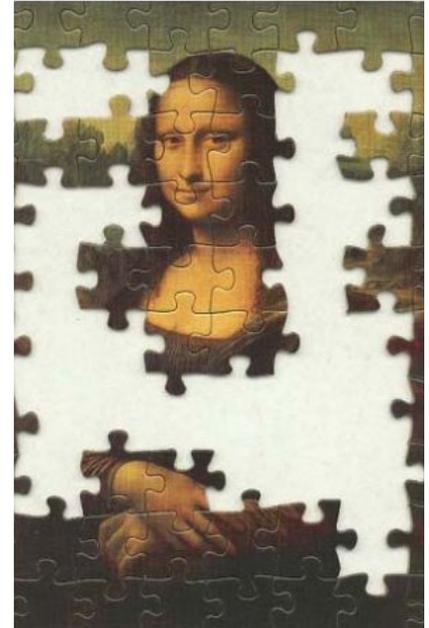
- Alunos de graduação e pós graduação focados em contribuir para projetos FLOSS
- Linux, GCC, Git, etc.
- <https://flusp.ime.usp.br>
 - Agenda de reuniões
 - Meios de contato



Estrutura da comunidade Linux

Modularização

- Linux é dividido em “subsistemas”
- Não confundir com “microkernel”, oposto de “monolítico”



Mantenedores

- Zelam pelo código de um subsistema, driver ou arquivo
- Revisam e aprovam contribuições, gerenciam as branches, etc.
- O arquivo MAINTAINERS: lista de mantenedores
- Recuperando um mantenedor: `kw m` ou [scripts/get_maintainers.pl](#)

Repositórios

- Cada subsistema tem seu fork
- <https://git.kernel.org/pub/scm/linux/kernel/git/>
- Algumas organizações têm seus próprios repositórios oficiais.
- Dica: Observe as branches do repositório e descubra em qual trabalhar.
Às vezes a mais atual não é a master!

Comunicação

- Lista de emails, IRC, etc. (Cada subsistema tem suas particularidades)
- Listas: <http://vger.kernel.org/vger-lists.html>
- Contribuições são enviadas e revisadas nas listas
- Archives: LKML, MARC, etc.
- **Importante: O padrão nas listas é plain text e resposta inline**

On Mon, 2018-11-05 at 16:59 -0200, Lucas Santos wrote:

> Hello,
>

Hey,

Sorry for the late reply.
I deferred answering your email, and nearly forgot.

> I'm looking at the code of impedaance-analyzer/ad5933, trying to
> understand
> what is needed to move it to the main tree. I am part of a study group
> and we
> are willing to work on it.

Cool :)
Thanks

Revisões

- Feitas pelos mantenedores e outros contribuidores
- Corrija e reenvie utilizando os prefixos v2, v3, ...
 - Exemplo: “[PATCH v2] cripto: add new config option”
- **Dica: Espere alguns dias antes de enviar novas versões, para receber mais revisões**

Processo de Desenvolvimento

Timeline de desenvolvimento

1. Encontrar onde contribuir
2. Escrever o código
3. Testar o código (compilação e possíveis testes)
4. Verificar o codestyle
5. Escrever uma boa mensagem de commit
6. Encontrar os destinatários do patch
7. Enviar o patch
8. Ao receber feedbacks, aplicar as correções e re-enviar o patch

Overview dos Repositórios

- COPYING
- Documentation
- Kbuild
- LICENSES
- **drivers**
- MAINTAINERS
- Makefile
- arch
- crypto
- README
- fs
- include
- init
- ipc
- kernel
- lib
- mm
- net
- samples
- scripts
- security
- sound
- tools

Mão na Massa: Ambiente e IIO

Configurando ambiente

1. Git
2. Baixar código fonte
3. KW
4. Inscrever-se na mailing list
5. Filtros do Gmail
6. Git send-email
7. Compilação
8. Qemu e VM

Git

- Checar instalação: `git --version`
- Configurar:
 - `git config --global user.name "<your name>"`
`git config --global user.email "<your email>"`
 - Use nome completo (ou parte dele). **Não usar** pseudônimos.

Baixar código fonte

- `git clone https://git.kernel.org/pub/scm/linux/kernel/git/jic23/iio.git`
- Dica: Visite o site acima. É um bom viewer online do repositório

Instalar KW

- Conjunto de scripts para facilitar o desenvolvimento no kernel
- <https://github.com/rodrigosiqueira/kworkflow>
- Mudar para a branch “unstable” e rodar `./setup.sh -i`

Inscriver-se na Mailing List + Filtro do Gmail

- <http://vger.kernel.org/vger-lists.html#linux-iio>
 - Email para majordomo@vger.kernel.org
 - Sem assunto
 - Corpo: "subscribe linux-iio"
- Filtros do Gmail

Has the words

Doesn't have



Skip the Inbox (Archive it)

...

Apply the label:

Git send-email

- Lembre-se: queremos enviar patches em plain text
- Alguns clientes de email (como o GMail web) trocam *tabs* por *espaços* ou inserem HTML
- Como configurar e usar:
<https://flusp.ime.usp.br/git/2019/02/15/sending-patches-by-email-with-git>

Compilando o Kernel

- O super arquivo .config
- Como conseguir um .config para a sua máquina:
 - `zcat /proc/config.gz > .config`
 - `cp /boot/config-`uname -r` .config`
- Habilitar/desabilitar módulos: `make nconfig`
- Apenas drivers enabled: `make localmodconfig`
- Compilar: `make ARCH=x86_64 -jX` (X=número de threads)
- Referência para compilar/instalar:
<https://flusp.ime.usp.br/others/2019/02/16/Kernel-compilation-and-installation/>

Qemu e VM (opcional)

- É legal ter uma VM para instalar e testar o kernel modificado
- Recomendamos usar o Qemu
- Referência:
<https://flusp.ime.usp.br/others/2019/02/15/use-qemu-to-play-with-linux/>

Considerações no IIO

- Docs: <https://www.kernel.org/doc/html/v4.14/driver-api/iio/index.html>
- Desenvolver na branch 'testing'
- Drivers:
 - drivers/staging/iio/*
 - drivers/iio/*
 - Procure o datasheet do dispositivo online ;)

Leituras

Leituras importantes

- Sobre o processo de desenvolvimento:
<https://www.kernel.org/doc/html/v4.15/process/2.Process.html> e
<https://www.kernel.org/doc/html/latest/process/submitting-patches.html>
- Primeiro patch: <https://kernelnewbies.org/FirstKernelPatch>
- Boas mensagens de commit: <https://chris.beams.io/posts/git-commit/>
- Linux codestyle:
<https://www.kernel.org/doc/html/v4.10/process/coding-style.html>
- Filosofia do patch: <https://kernelnewbies.org/PatchPhilosophy>
- Docs nos repos: `Documentation` e `Documentation/process`