

# GNU Compiler Collection Primeiros Passos



Reunião de Férias do  
FLUSP: FLOSS at USP



# Estrutura da comunidade

# Modularização

<https://www.gnu.org/software/gcc/svn.html>

Principal: trunk

# Mantenedores

- Zelam pelo código de um subsistema: revisam/aprovam contribuições, gerenciam as branches, etc.
- O arquivo MAINTAINERS: lista de mantenedores
- Às vezes, o maintainer do arquivo fica no cabeçalho do mesmo

# Repositórios

- <https://gcc.gnu.org/viewcvs/gcc/>
- Dica: Observe as branches do repositório e descubra em qual trabalhar.  
Às vezes a mais adequada não é a trunk!

# Comunicação

- Lista de emails, IRC, etc.
- Inscrever-se em uma ou mais listas do seu interesse
  - Conjunto de todas as listas: <https://gcc.gnu.org/lists.html>
  - Principais: gcc, gcc-patches
- Contribuições são enviadas e revisadas nas listas
- Archives.
- **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
- Envie patches com o prefixo [PATCH]
- Não se esqueça do ChangeLog
- Se estiver corrigindo um PR (Problem Report), insira o ID do mesmo.
- Corrija e reenvie utilizando os prefixos v2, v3, ...
- **Dica: Espere alguns dias antes de enviar novas versões, para receber mais revisões**

# Processo de Desenvolvimento

# Timeline

1. Encontrar onde contribuir
2. Escrever o código
3. Testar o código (compilação e possíveis testes)
4. Verificar o codestyle
5. Gerar patch
6. Escrever um bom título para o email e uma breve explicação sobre o patch
7. Atualizar o ChangeLog e inserir a mudança no corpo do email
8. Enviar o patch para a lista gcc-patches
9. Ao receber feedbacks, aplicar as correções e reenviar o patch

# Mão na Massa: Ambiente

# Configurando ambiente

1. SVN
2. Baixar código fonte
3. Inscrever-se na mailing list
4. Envio de mensagens como Texto puro no Gmail
5. Filtros do Gmail
6. Compilação

# SVN

- Checar instalação: `svn --version`
- Ver histórico de commits: `svn log`
- Atualizar o repositório com o global: `svn update`

# Baixar código fonte

- `svn checkout svn://gcc.gnu.org/svn/gcc/$repositório`
- Substitua `$repositório` por `trunk`, por exemplo

# Inscriver-se na Mailing List + Filtro do Gmail

- <https://gcc.gnu.org/lists.html>
  - Responder o email recebido
- Filtros do Gmail

Has the words

Doesn't have



Skip the Inbox (Archive it)

...

Apply the label:

# Compilando o GCC

- Crie uma pasta e navegue até ela
- Execute o configure do GCC a partir desta pasta
- `make -j NUM_THREADS`
- `make DESTDIR=diretório install`
- Referência para compilar/instalar: <https://gcc.gnu.org/wiki/InstallingGCC>

# Compilando o GCC

- Compilar para x86 normalmente não tem complicações
- Outras arquiteturas têm suas particularidades

Testes

# Testes no GCC

- Todo patch não-trivial enviado precisa de pelo menos um test case
- Algum cuidado a respeito de onde colocar os testes
  - <https://gcc.gnu.org/onlinedocs/gcc-4.7.0/gccint/C-Tests.html>
- `make check`
- <https://gcc.gnu.org/install/test.html>

Leituras

# Leituras Interessantes

- Dicas para iniciantes: <https://gcc.gnu.org/ml/gcc/2018-11/msg00186.html>
- Unofficial Gcc Newbes:  
<https://dmalcolm.fedorapeople.org/gcc/newbies-guide/index.html>
- Sobre o processo de desenvolvimento:  
<https://www.gnu.org/software/gcc/contribute.html>
- <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/>
- Gnu codestyle: <https://www.gnu.org/prep/standards/standards.html>
- Filosofia do patch: <https://kernelnewbies.org/PatchPhilosophy>