



Minha primeira contribuição

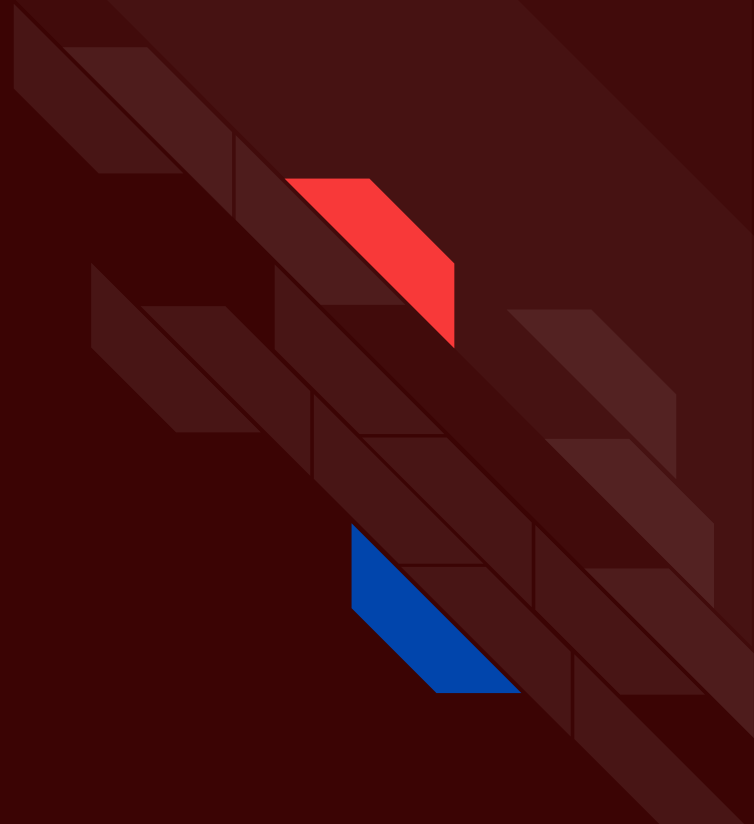
Desmistificando as primeiras contribuições para
projetos open source bem estabelecidos.



Resumo da apresentação

1. Resumo
2. Credenciais
3. O que é uma contribuição?
4. Qual foi a minha primeira contribuição?
5. O que as comunidades estão fazendo?
6. O que você está esperando?

Essa é a apresentação
que eu gostaria de ter
ouvido quando estava
na faculdade





Credenciais

Quem sou eu para falar que é fácil?

- Formado em ciencia da computacao
- Trabalho desenvolvendo GDB
- 5 primeiras contribuições
 - 3 delas com código
- Participei de discussões de como melhorar a experiência dos contribuidores
- Reviso patches consistentemente na lista do GDB



O que é uma contribuição?

1. Código
 - a. Consertar bugs
 - b. adicionar features
 - c.
2. Reportar bugs
 - a. Se os problemas não são reportados, como eles podem ser consertados?
 - b. Escrever bons bug reports é difícil
 - c. suítes de testes raramente são exaustivas.
3. Documentação
 - a. Ninguém gosta de escrever
 - b. Todos os usuários precisam disso
4. Outros
 - a. User experience
 - b. Design
 - c. Arte



O que é uma contribuição?

1. ~~Código~~

- a. ~~Consertar bugs~~
- b. ~~adicionar features~~
- c. ~~...~~

2. Reportar bugs

- a. Se os problemas não são reportados, como eles podem ser consertados?
- b. Escrever bons bug reports é difícil
- c. suítes de testes raramente são exaustivas.

3. Documentação

- a. Ninguém gosta de escrever
- b. Todos os usuários precisam disso

4. Outros

- a. User experience
- b. Design
- c. Arte



Reportando bugs

O que é um bom bug report?

Explicação detalhada do problema: (What? How? When? ...)

Bug Reports tem templates por um motivo

Description of problem:

Version-Release number of selected component (if applicable):

How reproducible:

Steps to Reproduce:

- 1.
- 2.
- 3.

Actual results:

Expected results:

Additional info:

Template para
reportar bugs no o
GDB do fedora



Reportando bugs

O que é um bom bug report?

Explicação detalhada do problema: (What? How? When? ...)

Projetos precisam de pessoas testando os updates!

Diferentes setups e hardware podem causar bugs diferentes.

O próprio fedora pede para desenvolvedores testarem

Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

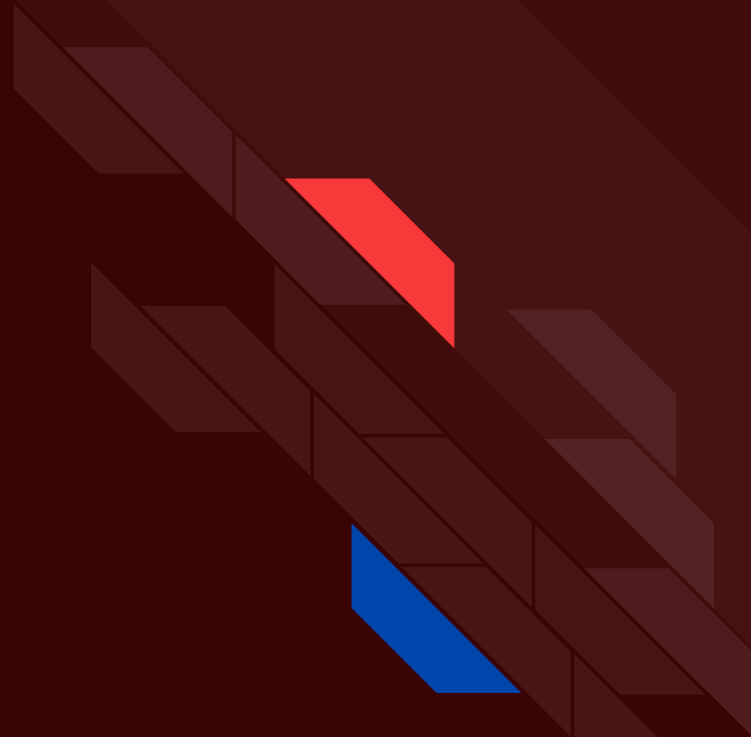
Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37

Re: Donate 1 minute of your time to test upgrades from F36 to F37



Alguns setups de testes podem dar resultados diferentes

```
> if { ![gdb_attach $testpid \
> -      -pattern "main.*at .*$::main_srcfile:.*"] } {
> +      -pattern ".*main.*at .*$::main_basename.c:.*"] } {
```

... when I run this test using Clang (9.0.1) without this patch, the test passes. I notice that when you ran the test the source file reported at the breakpoint doesn't include the `${srcdir}`.

I don't know why you're only seeing a relative path, but I guess, as far as this test is concerned, just checking for the final filename is fine.

email:

<https://sourceware.org/pipermail/gdb-patches/2022-September/191796.html>



Reportando bugs

O que é um bom bug report?

Explicação detalhada do problema: (What? How? When? ...)

Projetos precisam de pessoas testando os updates!

Diferentes setups e hardware podem causar bugs diferentes.

Sem pessoas fazendo coisas “incomuns” no dia a dia. Projetos não sabem o que precisam resolver

LLVM/Clang não integra 100% com GDB

Falta informação de privacidade em typedefs:

<https://github.com/llvm/llvm-project/issues/57608>

Clang não informa sobre aliases de variáveis

<https://github.com/llvm/llvm-project/issues/52664>

GDB não acha macros da linha de comando no clang

https://sourceware.org/bugzilla/show_bug.cgi?id=29034

GDB não acha a classe base em herança virtual

https://sourceware.org/bugzilla/show_bug.cgi?id=16841



Documentação

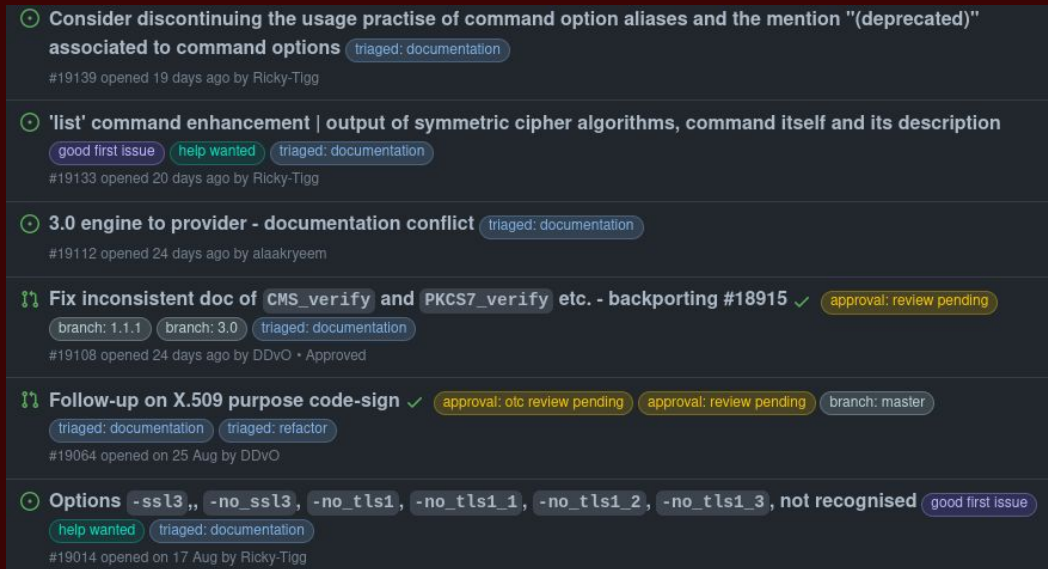
É a parte chata de escrever código, mas precisa ser feito

Quantas vezes vocês já acharam uma biblioteca e gostariam que as funções fossem explicadas melhor?

Além de manual/documentação online, vários projetos usam exemplos para mostrar como as coisas funcionam

..... Alguém precisa escrever esses exemplos, e nem sempre quem introduz a feature faz isso.

OpenSSL tem 135 bugs específicos para documentação



Consider discontinuing the usage practise of command option aliases and the mention "(deprecated)" associated to command options triaged: documentation
#19139 opened 19 days ago by Ricky-Tigg

'list' command enhancement | output of symmetric cipher algorithms, command itself and its description good first issue help wanted triaged: documentation
#19133 opened 20 days ago by Ricky-Tigg

3.0 engine to provider - documentation conflict triaged: documentation
#19112 opened 24 days ago by alaaakryeem

Fix inconsistent doc of CMS_verify and PKCS7_verify etc. - backporting #18915 ✓ approval: review pending
branch: 1.1.1 branch: 3.0 triaged: documentation
#19108 opened 24 days ago by DDvO • Approved

Follow-up on X.509 purpose code-sign ✓ approval: otc review pending approval: review pending branch: master
triaged: documentation triaged: refactor
#19064 opened on 25 Aug by DDvO

Options -ssl3,, -no_ssl3, -no_tls1, -no_tls1_1, -no_tls1_2, -no_tls1_3, not recognised good first issue
help wanted triaged: documentation
#19014 opened on 17 Aug by Ricky-Tigg

Lista de issues no github do openssl, buscando por “triaged: documentation”



Minha(s) primeira(s) vez(es)

- Clang e gcc:
 - Apenas os reports de bug.
- QEMU:
 - O projeto inicial era tão grande que pedi socorro
 - <https://lists.gnu.org/archive/html/qemu-devel/2021-04/msg00497.html>
 - E ainda fiz tudo errado!
 - <https://lists.gnu.org/archive/html/qemu-devel/2021-04/msg01561.html>
- GDB:
 - <https://sourceware.org/pipermail/gdb-patches/2021-October/182338.html>
 - 4 revisões para adicionar parentesis ao redor de “print”s (burrice minha)
- FreeBSD não conta - eu conhecia o reviewer pessoalmente e ele não revisou o patch direito, eu acho.



Clang

- Bug reports. Fazendo reports de qualidade, é rapido deles serem verificados e consertados. Exemplos:
- <https://github.com/llvm/llvm-project/issues/52661>
 - Bug report ruim, levou multiplos meses ate receber uma resposta



billionai commented on 13 Dec 2021



Running clang with -O0 -g doesn't create epilogue information for trivial epilogues. A code like

```
1 int main(){
2   fun_a( fun_b(), fun_c());
3   return 0;
4 }
```

when debugged using GDB will go from line 3 directly to `__libc_start_call_main()`. This creates a problem for GDB's testsuite, since it expects being able to step once again, to line 4.

It also can make the debugging experience confusing for users, since if they step into `fun_c` and use the command `next`, the program will already be at line 3 here, instead of back at line 3 and able to step into `fun_b` or `fun_a`.



Clang

- Bug reports. Fazendo reports de qualidade, é rapido deles serem verificados e consertados. Exemplos:
- <https://github.com/llvm/llvm-project/issues/52661>
 - Bug report ruim, levou multiplos meses ate receber uma resposta
- <https://github.com/llvm/llvm-project/issues/57608>
 - Bug report bom. Recebeu uma resposta em uma semana, um patch em um mes.

billionai commented 20 days ago • edited by VoltrexMaster ↗



when compiling a program such as

```
class c{
public:
    typedef double public_double;
    c(double dd) : d(dd) { }
private:
    public_double d;
};

int main(){
    c tmp(1);
    c::public_double dob = 0;
    return (int)dob;
}
```



clang can correctly identify the access level, but when analysing the class using `readelf -w` we find the following debug information for the typedef:

```
<1><3e>: Abbrev Number: 5 (DW_TAG_typedef)
<3f>  DW_AT_type       : <0x57>
<43>  DW_AT_name       : (indexed string: 0x6): public_double
<44>  DW_AT_decl_file  : 0
<45>  DW_AT_decl_line  : 3
```

While g++ gives the following debuginfo:

```
<2><53>: Abbrev Number: 8 (DW_TAG_typedef)
<54>  DW_AT_name       : (indirect string, offset: 0x62): public_double
<58>  DW_AT_decl_file  : 1
<59>  DW_AT_decl_line  : 3
<5a>  DW_AT_decl_column: 20
<5b>  DW_AT_type       : <0x77>
<5f>  DW_AT_accessibility: 1      (public)
```

The lack of `DW_AT_accessibility` means that GDB is unable to fully understand the class, and ptype can't inform the user about access levels of the typedef.



Gcc

Também foi bug report.

Reportar se é um problema novo ou regressão, trabalhar em reprodutores pequenos e descrever de maneira concisa é muito mais importante que tentar entender o problema sem olhar para o código

blarsen 2022-03-22 18:50:43 UTC

[Description](#)

Created [attachment 52665](#) [[details](#)].
minimal testcase

When compiling the attached test case with the options `-g -O0`, there is a bug in the line table generation.

The first instruction of `merge_sort` is assigned to line 17 instead of line 2, then backtracks to line 5. It generates some confusion when trying to debug the code, as GDB prints the wrong line when that instruction is being evaluated.

My guess is that g++ sees the early exit and assumes that part of the "if" is related to that exit, and mislabels that instruction. I tried removing the "return" statement and putting the rest of the code inside an else block, and the line table was closer to what I expected.

This seems to be a long standing bug, as I can reproduce it with g++ 8.5.0 and g++ 11.2.1, though 8.5.0 sets the line as 16 instead.



FreeBSD

Primeira contribuição de todas, porém essa foi diferente

Eu conhecia o reviewer pessoalmente (era parte do time na época), e eu pessoalmente acho que não teria deixado passar o código que ele deixou hoje em dia.

A contribuição foi um “if” adicional na função bcopy.

<https://reviews.freebsd.org/D28776>



QEMU

De novo, a primeira submissão foi um pouco anormal.

Como eu não escolhi e era novo na comunidade, eu mandei um email pessoal pro mantenedor da parte do código que eu ia mexer. Ele respondeu dizendo que ia ajudar e pedindo para conversar na lista publica, assim outras pessoas podem olhar e responder mais rapido, ou podem rever no futuro.

<https://lists.gnu.org/archive/html/qemu-devel/2021-04/msg00497.html> - Multiplas pessoas ajudando de varios jeitos possiveis e explicando coisas que até hoje eu nao tenho noção.

<https://lists.gnu.org/archive/html/qemu-devel/2021-04/msg01561.html> - Primeira versão ainda foi toda errada, varios problemas em todos os patches.

GDB

na minl

GDB te

a suíte

O scrip

e outro

e outro

e outro!

```
Hi
On 2021-10-01 15:02, Bruno Larsen via Gdb-patches wrote:
> Since python 2 is no longer supported on most distributions, updated
> the
```

Change updated -> update in the subject as well.

```
I'
The patch is OK with that changed.
```

```
Br
L: Simon
```

```
"python". We aim for having a Python 3 script. If it is still runnable
> as a Python 2 script with not too much effort, then it's just a bonus.
```

```
>
The patch LGTM other than that.
```

do rodando

n!

<https://sourceware.org/pipermail/gdb-patches/2021-October/182338.html>



E as comunidades?

As comunidades que eu faço parte sabem da dificuldade de novas pessoas contribuírem

Algumas coisas que eu ouvi que querem fazer é:

- Priorizar patches de novos contribuintes - tanto em tempo quanto em tom
- Tentar acomodar novas ferramentas - Como fazer IDEs saberem lidar com os projetos?
- Tentar parecer menos assustadores - Open Office hours da glibc, por exemplo
- Tentando simplificar o processo de review -

<https://sourceware.org/pipermail/gdb/2022-September/050282.html>

E elas já são muito mais tranquilas que as histórias do ~~Lebo~~ Linus mau.



E os outros

Algumas coisas que eu já sei que acontecem são:

- Bugs “bons para iniciantes”
- Tutoriais de como compilar/testar/contribuir para o projeto
- Discussões na lista sobre como resolver problemas - [exemplo](#)
- Ferramentas para facilitar submissão de patches por email - <https://sr.ht/>
- Incentivando estudantes a participarem de conferências (Cauldron dava inscrições gratuitas)

Re: [comitted] gdb: add Torbjörn Svensson to gdb/MAINTAINERS

Luis Machado luis.machado@arm.com

Fri Oct 14 11:21:32 GMT 2022

- Previous message (by thread): [\[comitted\] gdb: add Torbjörn Svensson to gdb/MAINTAINERS](#)
- Next message (by thread): [\[committed\]\[gdb/testsuite\] Add cond_wrap proc](#)
- **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

```
+Luis Suzuki <luis.suzuki@systemz.ibm.com>  
> +Torbjörn Svensson <torbjorn.svensson@torbjorn.se>  
> Alfred M. Szmidt <ams@gnu.org>  
> Ali Tamur <tamur@google.com>  
> David Taylor <david.taylor@arm.com>
```

Congrats. Happy GDB hacking.



E vocês?

O que estão esperando?



Obrigado!

Contato:

Bruno P. Larsen

blarsen <at> redhat.com

telegram: @flango

github: @billionai

