

Desenvolvimento Aberto



Debugging II

Igor dos Santos Montagner (igorsm1@insper.edu.br)

Aulas passadas

- Compilando seu projeto
- "Regras" de debugging do livro
- Dicas de como se encontrar no projeto

Hoje

- "Regras" de debugging do livro

9 regras de debug

1. UNDERSTAND THE SYSTEM
2. MAKE IT FAIL
3. QUIT THINKING AND LOOK
4. DIVIDE AND CONQUER
5. CHANGE ONE THING AT A TIME
6. KEEP AN AUDIT TRAIL
7. CHECK THE P LUG
8. GET A FRESH VIEW
9. IF YOU DIDN'T FIX IT, IT AIN'T FIXED

Destrinchando as 9 regras

1. UNDERSTAND THE SYSTEM:

- Compile o projeto
- Rode sua versão
- Saiba dizer onde estão os fontes e por qual critério estão organizados

2. MAKE IT FAIL

- Reproduza o bug

Destrinchando as 9 regras

3. QUIT THINKING AND LOOK

- Encontre no código onde o bug pode estar
- Comece geral (em qual arquivo está a funcionalidade?) e vá restringindo (em qual função o bug "explode"?)

4. DIVIDE and CONQUER

- Faça uma lista de tarefas contendo modificações que você acha que resolveriam o problema e que você precisa aprender para fazê-las
- Ordene-as de acordo com sua facilidade.

Destrinchando as 9 regras

5. CHANGE ONE THING AT A TIME

6. KEEP AN AUDIT TRAIL

- Log de como você encontrou onde mexer
- Log de todos arquivos de interesse e seus usos
- Log de todas as pesquisas feitas
- Log de todas as modificações feitas

Destrinchando as 9 regras

7. CHECK THE PLUG

8. GET A FRESH VIEW

- Professor
- Colegas trabalhando no mesmo projeto
- IRC, Mailing list, Issues e PR

Destrinchando as 9 regras

9. IF YOU DIDN'T FIX IT, IT AIN'T FIXED



Como se encontrar no projeto

Problema: Como encontro em qual arquivo mexer?

Ferramenta: comando `grep` permite buscar por strings em todos os arquivos de uma pasta.

Solução:

```
`$ grep [OPTIONS] PATTERN FILES
```

- `PATTERN` : expressão regular
- `FILES` : lista de diretórios ou arquivos

Como se encontrar no projeto

Exemplo 1: buscar todos arquivos nas pasta atual (.) e subpastas com o texto "dialog" ignorando maiúsculas/minúsculas.

```
$ grep -r -i dialog .
```

Mais info:

- `man grep` ([disponível online](#))
- Seção *Regular Expressions* é particularmente relevante

Como se encontrar no projeto

Exemplo 2: Listas todos os arquivos `.cpp` que fazem algum include

```
`$ grep -r --include "*.cpp" "#include" .
```

```
[igor@haute-normandie supercomp-2]$ grep -r --include "*.cpp" "#include" .  
./github/multi-core/03-intro-omp/calculo_pi.cpp:#include <stdio.h>  
./github/multi-core/03-intro-omp/calculo_pi.cpp:#include <omp.h>  
./github/multi-core/03-intro-omp/exemplo1.cpp:#include <iostream>  
./github/multi-core/03-intro-omp/exemplo1.cpp:#include <omp.h>  
./github/multi-core/02-fork-join/exemplo1-threads.cpp:#include <thread>  
./github/multi-core/02-fork-join/exemplo1-threads.cpp:#include <iostream>  
./github/simd/funcs.cpp:#include <math.h>  
./01-introducao/src/teste.cpp:#include "funcs.cpp"  
./01-introducao/src/teste.cpp:#include <iostream>  
./01-introducao/src/sqrt_simd.cpp:#include <x86intrin.h> //Extensoes SSE  
./01-introducao/src/sqrt_simd.cpp:#include <bits/stdc++.h> //Bibliotecas STD  
./01-introducao/src/funcs.cpp:#include <math.h>  
./01-introducao/src/funcs.cpp:#include <iostream>  
./01-introducao/src/funcs.cpp:#include <chrono>
```

Esta semana

- Duas aulas para finalizar seu primeiro PR
- Usem as 9 regras!
- O quê acontece depois do PR? [\[1\]](#)

Desenvolvimento Aberto



Debugging II

Igor dos Santos Montagner (igorsm1@insper.edu.br)