

Guia de atualização para versão 3.11

Este documento explica o procedimento aplicado ao Sinuca pinpoints para torná-lo compatível com a versão 3.11 do PinPoints. Seu objetivo é servir de referência para futuras atualizações de compatibilidade com as próximas versões do PinPoints.

Todas as alterações necessárias foram efetuadas sobre a seguinte pasta do Sinuca tracer implementado no OrCS:

>> **OrCS/trace_generator/extras/pinplay/PinPoints/scripts**

Preparação

- O primeiro passo é baixar a versão atual do OrCS e instalar o PinPoints nele, de acordo com o documento **Readme OrCS tracer.pdf**.
- Também precisamos de um programa para testar as alterações no tracer, então vamos criar um simples Hello World:

```
#include <stdio.h>

int main() {
    printf("Hello World\n");
    return 0;
}
```

- Já que o PinPlay precisa de um programa linkado estáticamente para rodar (fonte: Zanata), utilize a seguinte flag em sua compilação: **-static**
- Para simplificar, esse programa ficará salvo na minha home:
 - >> ~/helloWorld
 - ou
 - >> /home/war/helloWorld

Execução inicial

Com tudo posicionado, vamos fazer uma execução inicial para verificar tudo. Primeiro, executando o helloWorld:

(lembre-se que copiar direto deste documento pode gerar caracteres incorretos, então prefira re-escrever)

```
~ >> ./helloWorld
Hello World
~ >> 
```

Agora vamos tentar aplicar o Sinuca PinPoints do jeito que está nesse arquivo:

- Navegue até a pasta **trace_generator**:
 - `>> cd ~/OrCS/trace_generator`
- Agora vamos tentar gerar os traços com esse comando gigante:
 - `>> python extras/pinplay/PinPoints/scripts/sinuca_pinpoints.py --delete -lbsp -T --program_name helloWorld --input_name none --command "~/helloWorld > saida.s 2>1" --mode st`
Essa versão do Sinuca tracer ainda rodava com python 2
- No meu caso, a execução termina com o seguinte erro:
 - **sinuca_pinpoints.py ERROR: A problem occurred in phase - Basic block vector generation [gen_BBV]**
 - Como mostrado pelo log, esse foi o último comando executado:
 - `/home/war/OrCS/trace_generator/pin -xyzzzy -reserve_memory whole_program.none/helloWorld.none_5942.address -t /home/war/OrCS/trace_generator/extras/pinplay/bin/intel64/pinplay-driver.so -replay -xyzzzy -replay:basename whole_program.none/helloWorld.none_5942 -replay:playout 0 -bbprofile -slice_size 30000000 -o helloWorld.none_5942.Data/helloWorld.none_5942 -log:mt 0 -- /home/war/OrCS/trace_generator/extras/pinplay/bin/intel64/nullapp`
A partir disso podemos deduzir que o erro ocorreu durante a fase de replay do mecanismo.

Para verificar se isso não é erro do próprio PinPoints, vamos executar a mesma operação com seus arquivos originais ([Fonte](#)):

- Executando o comando: `python3 extras/pinplay/scripts/pinpoints.py --delete -lbsp --program_name helloWorld --input_name none --command "~/helloWorld > saida.s 2>1" --mode st`

Obtemos um novo erro:

- **pinpoints.py ERROR: A problem occurred in phase - Simpoinits generation [Simpoinit]**

Observando o log, temos que esse erro é causado porque o parâmetro `slice_size` é maior que o programa:

```
simpoint.py ERROR: Slice size is greater than the number of instructions. Reduce parameter 'slice_size'.  
Instruction count:      42.574  
Slice size:            30.000.000
```

assim, como nosso `helloWorld` é pequeno, vamos definir esse parâmetro como 10 e executar novamente.

```
>> python3 extras/pinplay/scripts/pinpoints.py --delete -lbsp --program_name helloWorld --  
input_name none --command "~/helloWorld > saida.s 2>1" --mode st --slice_size 10
```

Após isso, o programa já conseguiu completar todas as fases:

```
real      0m0,691s  
user      0m0,573s  
sys       0m0,124s  
  
+++ Finished processing: whole_program.none/helloWorld.none_6891  
*** Finished iteration 2 generating pinballs ***      abril 30, 2020 16:15:47  
+++ Waiting on final concurrent region pinball generation  
*** Finished generating region pinballs [relog_regions] ***      abril 30, 2020  
16:15:47  
*** TRACING: END ***      abril 30, 2020 16:15:47
```

Isso mostra que o problema é com nossos arquivos e não com o PinPoints, assim, vamos atualizá-los.

Atualização

Como visto anteriormente, apenas os scripts relacionados ao Sinuca tracer não funcionam, porém esses scripts foram baseados em uma versão antiga do PinPoints, assim, podemos utilizar os scripts funcionais do PinPoints 3.11 para atualizar o Sinuca tracer.

Para isso, vamos fazer um merge das pastas. Primeiro, copiamos a pasta dos scripts funcionais para o mesmo diretório do Sinuca tracer:

```
>> cp -r ~/OrCS/trace_generator/extras/pinplay/scripts/  
~/OrCS/trace_generator/extras/pinplay/PinPoints/scripts2/
```

Nesse momento a pasta `~/OrCS/trace_generator/extras/pinplay/PinPoints/scripts/` contém o Sinuca tracer antigo e com problemas e a `~/OrCS/trace_generator/extras/pinplay/PinPoints/scripts2/` contém o PinPoints funcional.

Como o Sinuca tracer é uma especialização do PinPoints, vamos copiar seus arquivos para a pasta do PinPoints funcional:

```
>> cd ~/OrCS/trace_generator/extras/pinplay/PinPoints/  
>> cp ./scripts/sinuca_* ./scripts2/
```

A partir de agora, esta pasta `scripts2` é nossa área de trabalho, vamos modificá-la até que tudo esteja funcionando.

Teste inicial

Primeiro vamos voltar à nossa pasta `~/OrCS/trace_generator` para testar nossa primeira mudança:

```
>> cd ~/OrCS/trace_generator
```

Vamos executar o mesmo comando do início mas com a nova pasta:

```
>> python extras/pinplay/PinPoints/scripts2/sinuca_pinpoints.py --delete -lbsp -T --program_name
helloWorld --input_name none --command "~/helloWorld > saida.s 2>1" --mode st
```

Nosso primeiro erro é um módulo não encontrado:

```
>> ImportError: No module named configparser
```

O problema aqui é que esse módulo (do jeito que está escrito) pertence ao python3, já que os novos scripts do PinPlay utilizam essa versão. Assim, executando com ele:

```
>> python3 extras/pinplay/PinPoints/scripts2/sinuca_pinpoints.py --delete -lbsp -T --
program_name helloWorld --input_name none --command "~/helloWorld > saida.s 2>1" --mode st
```

```
import sinuca_kit
File "/home/war/OrCS/trace_generator/extras/pinplay/PinPoints/scripts2/sinuca_
kit.py", line 54, in <module>
    class Sinuca_TracerKit(kit.Kit):
File "/home/war/OrCS/trace_generator/extras/pinplay/PinPoints/scripts2/sinuca_
kit.py", line 110, in Sinuca_TracerKit
    kit_type = config.SINUCA
AttributeError: module 'config' has no attribute 'SINUCA'
```

Mudamos o erro, já é uma vitória hahaha

O problema agora é que o atributo SINUCA não está presente no arquivo config.py, isso porque só copiamos os arquivos que iniciavam com a palavra **sinuca_** da antiga versão.

Não, não podemos só copiar o antigo senão dá ruim.

Então vamos usar o comando **meld** para ver quais as diferenças entre os dois configs:

```
>> cd ~/OrCS/trace_generator/extras/pinplay/PinPoints
```

```
>> meld scripts/config.py scripts2/config.py
```

A maior parte das coisas devem continuar diferentes, tipo isso:

```
import ConfigParser
import optparse
import os
import pickle
import platform
import random
import shutil
import subprocess
import tempfile
import types

# Local modules
#
import msg

#####
import configparser
import optparse
import os
import pickle
import platform
import random
import shutil
import subprocess
import tempfile
import types

# Local modules
#
import msg
from msg import ensure_string
#####
```

Estamos interessados apenas naquilo diretamente relacionado ao sinuca, por exemplo:

```
sinuca_tracer_output = ''
sinuca_tracer_threads = 0
sinuca_tracer_parallel_start = -2
sinuca_tracer_parallel_end = -2
```

Should the parameter 'program name' be used in the whole program?

```
simhome = ''
simpoint_options = ''
slice_size = 0
sniper_root = ''
warmup_length = 0
```

Should the parameter 'program name' be used in the whole program?

e

```
UNKNOWN = -1
PINPLAY = 0
SDE = 1
BRPRED = 2
CMPSIM = 3
X86NOAS = 4
X86 = 5
KEIKO = 6
SINUCA = 7
```

```
UNKNOWN = -1
PINPLAY = 0
SDE = 1
BRPRED = 2
CMPSIM = 3
X86NOAS = 4
X86 = 5
KEIKO = 6
```

e

```
pickle.dump(sniper_root, pickle_file)
pickle.dump(sim_kit_type, pickle_file)
```

```
pickle.dump(sinuca_tracer_output, pickle_file)
pickle.dump(sinuca_tracer_threads, pickle_file)
pickle.dump(sinuca_tracer_parallel_start, pickle_file)
pickle.dump(sinuca_tracer_parallel_end, pickle_file)
```

```
except (pickle.PicklingError):
```

```
pickle.dump(sniper_root, pickle_file)
pickle.dump(sim_kit_type, pickle_file)
```

```
except (pickle.PicklingError):
    msg.PrintAndExit('Error writing pickle file: ' + pickle_file)
    pickle_file.close()
```

```
return pickle_name
```

```
def ReadGlobalVars(self, filename):
```

dentre outros....

Tenha certeza de passar todas essas coisas relacionadas para a nova versão de config.py.

Não remova ou altere nada já presente nesse arquivo, apenas adicione aquilo relacionado ao sinuca.

Lembre-se de salvar as modificações antes de fechar o meld.

Segunda tentativa

Com tudo isso feito e o arquivo de configuração atualizado, vamos tentar executar novamente:

```
>>> cd ~/OrCS/trace_generator/
```

```
>>> python3 extras/pinplay/PinPoints/scripts2/sinuca_pinpoints.py --delete -lbsp -T --  
program_name helloWorld --input_name none --command "~/helloWorld > saida.s 2>1" --mode st
```

```
ERROR: Path to the PinPlay kit was not found.  
Default kit location is: /home/war/pinplay
```

Precisamos dizer ao tracer que o PinPlay não está localizado em **~/pinplay**, mas em nossa pasta **~/OrCS/trace_generator**.

Para isso, vamos adicionar um parâmetro ao nosso comando:

```
>>> python3 extras/pinplay/PinPoints/scripts2/sinuca_pinpoints.py --delete -lbsp -T --  
program_name helloWorld --input_name none --command "~/helloWorld > saida.s 2>1" --mode st  
--pinplayhome ~/OrCS/trace_generator/
```

```
sinucapoint.py ERROR: Slice size is greater than the number of instructions. Reduc  
e parameter 'slice_size'.  
Instruction count:      42.594  
Slice size:            30.000.000
```

Novamente temos nosso erro de slice_size, vamos adicionar o parâmetro que o reduz:

```
>>> python3 extras/pinplay/PinPoints/scripts2/sinuca_pinpoints.py --delete -lbsp -T --  
program_name helloWorld --input_name none --command "~/helloWorld > saida.s 2>1" --mode st  
--pinplayhome ~/OrCS/trace_generator/ --slice_size 10
```

```
*** Finished generating region pinballs [relog_regions] ***      abril 30, 2020  
17:12:47  
sinuca_tracer_output:  
sinuca_tracer_threads:      0  
sinuca_tracer_parallel_start: -2  
sinuca_tracer_parallel_end:  -2  
  
+++ ERROR: Please set sinuca_tracer_output.  
  
*****  
*****  
sinuca_pinpoints.py ERROR: A problem occurred in phase - Could not find the sinu  
ca_tracer_output parameter.  
  
*** TRACING: END ***      abril 30, 2020 17:12:47
```

Aparentemente passamos por mais fases! Com novos problemas de configuração, está na hora de criarmos um arquivo de configuração antes que cada comando fique do tamanho de uma página e impossível de gerenciar ;)

Criando arquivo de configuração

Dentro de ~/OrCS/trace_generator, vamos criar o arquivo **config.cfg** com o seguinte conteúdo (baseado no erro anterior e em nossa gigante linha de comando):

- [Parameters]
- *program_name: helloWorld*
- *input_name: none*
- *command: "~/helloWorld > saida.s 2>1"*
- *mode: st*
- *pinplayhome: /home/war/OrCS/trace_generator/ (Deve ser o caminho completo, sem aspas)*
- *slice_size: 10*
- *sinuca_tracer_output: "result.helloWorld.trace"*
- *sinuca_tracer_threads: 1*
- *sinuca_tracer_parallel_start: -1*
- *sinuca_tracer_parallel_end: -1*

Lembre-se que copiar pode gerar caracteres incorretos, tipo as aspas, e não coloque essas bolinhas antes :)

Também altere o caminho para o seu pinplayhome de acordo com seu usuário. (digite o comando **pwd** no terminal quando estiver na pasta **trace_generator** para obter esse caminho)

Resultado:

```
home > war > OrCS > trace_generator > ⚙ config.cfg
1  [Parameters]
2
3  program_name: helloWorld
4  input_name: none
5  command: "~/helloWorld > saida.s 2>1"
6  mode: st
7
8  pinplayhome: /home/war/OrCS/trace_generator/
9  slice_size: 10
10
11 sinuca_tracer_output: "result.helloWorld.trace"
12 sinuca_tracer_threads: 1
13 sinuca_tracer_parallel_start: -1
14 sinuca_tracer_parallel_end: -1
```


Terceira tentativa

Vamos executar a partir do nosso arquivo de configuração recentemente criado:

```
>>> cd ~/OrCS/trace_generator
```

```
>>> python3 extras/pinplay/PinPoints/scripts2/sinuca_pinpoints.py --delete -lbsp -T --cfg config.cfg
```

```
pwd;sinuca_replay_dir.py --replay_dir helloWorld.none_11329.pp --log_options " -
output "result.helloWorld.trace" -threads 1 -parallel_start -1 -parallel_end -1
" --sim_add_filename --cfg config.cfg --global_file global.dat.14330
/home/war/OrCS/trace_generator

real    0m0,000s
user    0m0,000s
sys     0m0,000s
bash: sinuca_replay_dir.py: command not found

*** Finished running simulator on region pinballs: [sim_regions] ***   abril
30, 2020 17:32:11

*****
*****
sinuca_pinpoints.py ERROR: A problem occurred in phase - simulator on region pin
balls: [sim_regions]

*** TRACING: END ***   abril 30, 2020 17:32:11
```

A parte mais importante aqui é que novos problemas realmente são avanços ;)

A mensagem nos informa que aquele primeiro comando gigante não pôde ser executado, visto que ele não encontrou o **sinuca_replay_dir.py**

Observando mais atentamente, isso ocorreu porque o comando foi executado sem nenhum diretório de referência, ou seja, o diretório desse arquivo deveria estar em PATH.

Mas mudar isso não parece uma boa, ficaria muito dependente do ambiente do usuário, assim, vamos descobrir onde isso está sendo executado e alterar esse caminho.

Procurando o nome **sinuca_replay_dir.py**, descobrimos que ele é definido no arquivo **sinuca_pinpoints.py**, então vamos alterar esse caminho. Porém, por ser algo variável, não queremos que esse arquivo seja editado toda vez, portanto, vamos criar uma variável chamada **all_scripts_path** no arquivo **config.py** e usá-la para alterar esse caminho. Assim no arquivo **sinuca_replay_dir.py** alteramos a definição da variável **sim_replay_cmd** para:

```
83     # Simulator replay script
84     #
85     sim_replay_cmd = config.all_scripts_path + 'sinuca_replay_dir.py'
```

E no arquivo config.py adicionamos a nova variável com o caminho correto (linha 115):

```
# Global variables
#
verbose = False
debug   = False

# Scripts location from trace_generator folder
all_scripts_path = '/home/war/0rCS/trace_generator/extras/pinplay/PinPoints/scripts2/'
```

Lembrando que estamos trabalhando na pasta scripts2, quando virar a principal essa linha deve ser alterada

Quarta tentativa

Executando novamente:

```
>>> python3 extras/pinplay/PinPoints/scripts2/sinuca_pinpoints.py --delete -lbsp -T --cfg config.cfg
```

```
Traceback (most recent call last):
  File "/home/war/OrcS/trace_generator/extras/pinplay/PinPoints/scripts2/sinuca_replay_dir.py", line 43, in <module>
    import sinuca_kit
  File "/home/war/OrcS/trace_generator/extras/pinplay/PinPoints/scripts2/sinuca_kit.py", line 49, in <module>
    import config
  File "/home/war/OrcS/trace_generator/extras/pinplay/PinPoints/scripts2/config.py", line 39, in <module>
    import configparser
ImportError: No module named configparser

*** Finished running simulator on region pinballs: [sim_regions] ***   abril 30, 2020 20:44:47

*****
*****
sinuca_pinpoints.py ERROR: A problem occurred in phase - simulator on region pinballs: [sim_regions]

*** TRACING: END ***   abril 30, 2020 20:44:47
```

Novamente temos o erro do **configparser**, ou seja, precisamos alterar o arquivo que deu erro para ser executado com o python3.

Para isso, basta mudarmos a primeira linha do arquivo **sinuca_replay_dir.py** de:

```
>>> #!/usr/bin/env python
```

para

```
>>> #!/usr/bin/env python3
```

Quinta tentativa

Executando com essa alteração, temos um novo erro ;)

```
sinuca_replay_dir.py ERROR: Undefined kit type in method replay_dir.Replay(): 7

*** Finished running simulator on region pinballs: [sim_regions] ***   abril
30, 2020 20:49:40

*****
*****
sinuca_pinpoints.py ERROR: A problem occurred in phase - simulator on region pin
balls: [sim_regions]
```

Esse erro nos informa que um tipo de kit (também não sei o que é :P) não foi definido no método **replay_dir.Replay()**. Buscando essa mensagem de erro, descobrimos que ela sai do arquivo **replay_dir.py** que é um dos arquivos originais do PinPlay. Usando o **meld** novamente entre esse arquivo e o **replay_dir.py** da versão anterior do Sinuca tracer:

```
>> cd ~/OrCS/trace_generator/extras/pinplay/PinPoints/
```

```
>> meld scripts/replay_dir.py scripts2/replay_dir.py
```

Descobrimos algumas diferenças relacionadas ao 7º tipo, o tipo SINUCA:

<pre>if config.sim_kit_type == config.BRPRED: import brpred_kit sim_kit = brpred_kit.BrPredKit() elif config.sim_kit_type == config.SINUCA: import sinuca_kit sim_kit = sinuca_kit.Sinuca_TracerKit() elif config.sim_kit_type == config.CMPSIM: import sde_cmppsim_kit sim_kit = sde_cmppsim_kit.CMPSimKit() else: msg.PrintAndExit('Undefined kit type in method r str(config.sim_kit_type))</pre>	<pre>.. NOTE: If you are adding a kit for a new simulator, # to modify this code. # if config.sim_kit_type == config.BRPRED: import brpred_kit sim_kit = brpred_kit.BrPredKit() elif config.sim_kit_type == config.CMPSIM: import sde_cmppsim_kit sim_kit = sde_cmppsim_kit.CMPSimKit() else: msg.PrintAndExit('Undefined kit type in method r str(config.sim_kit_type))</pre>
--	--

e

<pre># Add the simulator knob to specify the file for the # the simulator. # # ~ log_opt += ' ' + sim_kit.GetSimOutputFile(basename_f</pre>	<pre># Add the simulator knob to specify the file for the # the simulator. # log_opt += ' ' + sim_kit.GetSimOutputFile(basename_f</pre>
---	---

Aplicando apenas essas duas alterações, vamos executar novamente: **Lembre-se de salvar ;)**

Lembrando, o comando usado é:

```
>> cd ~/OrCS/trace_generator
```

```
>> python3 extras/pinplay/PinPoints/scripts2/sinuca_pinpoints.py --delete -lbsp -T --cfg config.cfg
```

Temos um erro conhecido a partir dessa execução:

```
10r3_warmup0_prolog0_region10_epitlog0_003_0-16317.0 --log_options " -output resu
lt.helloWorld.trace -threads 1 -parallel_start -1 -parallel_end -1 " --global_f
ile global.dat.2042 --cfg config.cfg
bash: sinuca_replayer.py: command not found

real    0m0.001s
user    0m0.001s
sys     0m0.000s
```

Novamente um arquivo não foi encontrado, dessa vez o **sinuca_replayer.py**.

Assim, como com o **sinuca_replay_dir.py**, vamos completar seu caminho.

Com um `ctrl+shift+f` no Visual Studio Code, buscamos onde esse nome foi atribuído a uma variável, e descobrimos que ele é atribuído à **replayer_cmd** no arquivo **sinuca_replay_dir.py**, agora, vamos completar esse caminho com a variável que criamos no **config.py**:

```
54  
55     # Branch predictor simulator python script to replay one pinball.  
56     #  
57     replayer_cmd = config.all_scripts_path + '|sinuca_replayer.py'
```

Porém esse arquivo não importa o **config.py**, então vamos adicionar essa dependência junto aos outros imports:

```
39     import sys  
40  
41     # Local modules  
42     #  
43     import sinuca_kit  
44     import replay_dir  
45     import config
```

Sexta tentativa

Com tudo isso pronto vamos executar novamente com nosso conjunto de comandos:

```
>> cd ~/OrCS/trace_generator
```

```
>> python3 extras/pinplay/PinPoints/scripts2/sinuca_pinpoints.py --delete -lbsp -T --cfg config.cfg
```

```
Traceback (most recent call last):
  File "/home/war/OrCS/trace_generator/extras/pinplay/PinPoints/scripts2/sinuca_replayer.py", line 49, in <module>
    import sinuca_kit
  File "/home/war/OrCS/trace_generator/extras/pinplay/PinPoints/scripts2/sinuca_kit.py", line 49, in <module>
    import config
  File "/home/war/OrCS/trace_generator/extras/pinplay/PinPoints/scripts2/config.py", line 39, in <module>
    import configparser
ImportError: No module named configparser
```

Novamente o erro da biblioteca do python3, dessa vez no **sinuca_replayer.py**.

Como já estamos cansados, vamos converter todos os arquivos que começam com **sinuca_** para python3, alterando sua primeira linha e colocando parênteses nos prints.

Executando novamente...

```
+++ Finished processing: helloWorld.none_23754_t0r7_warmup0_prolog0_region10_epilog0_007_0-07566.0
WaitJobs() unexpected error occurred: non-zero exit code

*** Finished running simulator on region pinballs: [sim_regions] ***   abril 30, 2020 21:30:34

*****
*****
sinuca_pinpoints.py ERROR: A problem occurred in phase - simulator on region pinballs: [sim_regions]
```

Agora, apesar dos traços terem sido gerados, tivemos um erro durante a simulação. Isso ocorre porque nosso programa é muito pequeno, assim, algumas configurações padrão do PinPoints acabam impedindo que esse programa funcione nele (minha teoria :P).

Teste final

Já que nosso pequeno programa não deu boa, vamos tentar com outro um pouco maior. Usei um programa que fiz durante minha graduação para inversão de matrizes, usando métodos meio complicados que já esqueci hahah

Nota: não esqueça de colocar a entrada do seu programa, senão ele fica tempo infinito esperando hahah

```
+++ Finished processing: helloWorld.none_24940_t0r4_warmup1500_prolog0_region20_
epilog0_004_0-01177.0
Inserted Output File Name = result.helloWorld.trace
Real Static File = result.helloWorld.trace.tid0.stat.out.gz => READY !
Real Dynamic File = result.helloWorld.trace.tid0.dyn.out.gz => READY !
Real Memory File = result.helloWorld.trace.tid0.mem.out.gz => READY !
Loading /home/war/ICC/base/invmat, Image id = 1
```

```
+++ Finished processing: helloWorld.none_24940_t0r13_warmup1500_prolog0_region11_
epilog0_013_0-06190.0

real    0m0,272s
user    0m0,217s
sys     0m0,057s

+++ Finished processing: helloWorld.none_24940_t0r2_warmup1500_prolog0_region12_
epilog0_002_0-04567.0

*** Finished running simulator on region pinballs: [sim_regions] ***   abril 3
0, 2020 21:44:02

*** TRACING: END ***   abril 30, 2020 21:44:02
```

Sucesso! Agora já temos nossa versão atualizada! Basta copiar estes arquivos da pasta **scripts2** para a pasta **scripts** de uma instalação limpa, alterar o **all_scripts_path** do **config.py** para a pasta original (scripts) e feito!

Extras

Se em algum ponto isso acontecer, vocês esqueceram de compilar o sinuca tracer.

```
/home/war/OrCS/trace_generator/pin -xyzzzy -virtual_segments 1 -reserve_memory
helloWorld.none_18896.pp/helloWorld.none_18896_t0r3_warmup0_prolog0_region10_e
pilog0_003_0-16317.0.address -t /home/war/OrCS/trace_generator/extras/pinplay/
bin/intel64/sinuca_tracer.so -replay -xyzzzy -replay:basename helloWorld.none_18
896.pp/helloWorld.none_18896_t0r3_warmup0_prolog0_region10_epilog0_003_0-16317.0
-replay:playout 0 -output result.helloWorld.trace -threads 1 -parallel_start
-1 -parallel_end -1 -log:mt 0 -- /home/war/OrCS/trace_generator/extras/pinplay
/bin/intel64/nullapp
E: Unable to load /home/war/OrCS/trace_generator/extras/pinplay/bin/intel64/sinu
ca_tracer.so
```

Para resolver basta ir na pasta e digitar make:

```
>> cd ~/OrCS/trace_generator/extras/pinplay/sinuca_tracer/
```

```
>> make
```

Quando terminar de compilar tudo, provavelmente o erro sumirá na próxima execução.