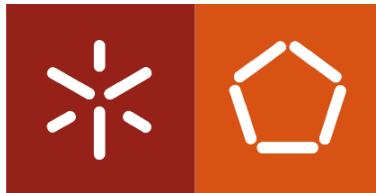


# Dtrace

## Desenvolvimento de programas

Hugo Gião  
PG41073

June 22, 2020



# Contents

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Ambiente utilizado</b>	<b>3</b>
2.1	Traçado das chamadas a <i>system call open</i> . . . . .	3
2.2	Estatísticas relativas a abertura e criação de ficheiros . . . . .	3
2.3	Replicação do programa <i>strace -c</i> . . . . .	4
<b>3</b>	<b>Conclusões</b>	<b>5</b>

# 1 Introdução

O trabalho apresentado neste documento consiste na criação de um conjunto de *scripts Dtrace* com funcionalidades distintas. O objetivo por detrás da criação destas *scripts* consiste em ganhar uma maior familiaridade com a linguagem D e o utilitário *Dtrace* de modo a conseguir utilizar os mesmos num trabalho futuro esse consistente em analisar diferentes implementações de um algoritmo.

Este documento descreve as várias *scripts* criadas e elabora sobre as estratégias utilizadas para a implementação das mesmas.

## 2 Ambiente utilizado

Para efeitos de teste e desenvolvimento das *scripts* foi utilizada uma maquina virtual *Oracle linux*, o virtualizador utilizado foi o *VirtualBox* e foram utilizados dois cores e 1 *GiB* de memória *RAM* nesta maquina virtual

### 2.1 Traçado das chamadas a *system call open*

A primeira *script* implementada traça as chamadas a *system call open* para abrir ficheiros com */etc* no seu caminho, para cada chamada detetada é imprimida uma linha contendo o nome do ficheiro executável associado ao processo, os *pid,uid* e *gid* associados ao processo que realizou a chamada, o caminho para o ficheiro a ser aberto, as *flags* passadas como argumento, a *probe* responsável por detetar esta chamada e o valor de retorno da chamada ao sistema.

A estratégia utilizada para implementar as funcionalidades descritas acima consistiu em primeiro guardar os valores de *arg0* e *arg1* da *probes syscall:open:entry*, esses valores correspondem ao ficheiro e *flags* utilizadas respetivamente, posteriormente é utilizada a *probe syscall:open:return* para os valores cujo caminho do ficheiro contenha */etc*, a distinção dessas chamadas com as restante é realizada utilizando a função *strstr*, é depois imprimida a informação descrita anteriormente com o auxilio da informação guardada e os valores nos campos *execname,pid,uid,gid* e *arg1* desta *probe*. O *script* associado a este programa consiste no *ex1.d*:

```
[dtrace@dtraceBase assignment2]$ ./ex1.d
```

```
cat(pid:1822,uid:1000,gid:1000) tried to open file /etc/ld.so.cache with
flags 524288 called open with return value 3
```

```
cat(pid:1822,uid:1000,gid:1000) tried to open file /etc/inittab with flags
0 called open with return value 3
```

Listing 1: Output do primeiro programa

```
cat /etc/inittab >> /tmp/test
```

Listing 2: Comandos utilizados para produzir o output do programa

### 2.2 Estatísticas relativas a abertura e criação de ficheiros

O segundo programa implementado imprime periodicamente com um valor em segundos passado como argumento os valores associados ao número de tentativas de abertura de ficheiros, o número de tentativas de criação de ficheiros e as tentativas de criar ficheiros bem-sucedidas agrupados por *PID*. Para implementar este programa utilizei as *probes syscall:open:entry* e *syscall:open:return*. Para contabilizar as tentativas de abertura de ficheiros existentes e de novos ficheiros o programa verifica para cada entrada na *probe syscall:open:entry* se as *flags* incluem a *flag O\_CREATE*, o valor das *flags* é guardado. Para contabilizar o sucesso das operações é utilizado o argumento *errno* da *probe syscall:open:return*. A

impressão periódica é feita utilizando a *probe tick* com o argumento passado na linha de comandos. O ficheiro associado a esta *script* é o *ex2.d*:

---

```
Date: 2020 Jun 13 23:11:12
```

---

```
Number of tentatives to open existing files
```

1995	cat	4
------	-----	---

---

```
Number of tentatives to create new files
```

1995	bash	1
------	------	---

---

```
Failllure to open existing files
```

---

```
Failllure to create news files
```

---

```
Success to open existing files
```

1995	bash	1
1995	cat	4

---

```
Success to create news files
```

1995	bash	1
------	------	---

Listing 3: Output do segundo programa

## 2.3 Replicação do programa *strace -c*

O terceiro programa implementado consiste numa reimplementação de algumas funcionalidades do comando *strace -c* utilizando o *Dtrace*. Estas funcionalidades consistem em medir durante a execução de um programa passado como argumento o número de chamadas a cada *system calls* realizadas pelo mesmo e tempo despendido nas mesmas em segundos, para além disso também é imprimido o tempo de execução do programa.

Para implementar as funcionalidades descritas utilizei as *probes syscall:::entry* e *syscall:::return*, essas *probes* medindo o número de chamadas a cada *system call* associada ao processo a ser observado e para cada acionamento da *syscall:::entry* é guardado o *timestamp* da mesma que é utilizado no acionamento da *syscall:::return* no momento do seu acionamento pela mesma *syscall*. O programa é depois corrido com a opção *-c* e com o programa a ser analisado como argumento. O ficheiro associado a esta *script* é o *ex3.d*:

```
time spent in seconds sice the start of this program 12
system calls frequency
```

access	2
arch_prctl	2
exit_group	2
fadvise64	2
mummap	2
write	2

read	6
brk	8
mprotect	8
open	8
newfstat	10
close	12
mmap	16

time spent on each system call in seconds

fadvise64	0
arch_prctl	0
access	0
write	0
munmap	0
read	0
newfstat	0
brk	0
close	0
mprotect	0
open	0
mmap	0

Listing 4: Output do terceiro programa

### 3 Conclusões

A implementação destes programas permitiu ganhar alguma familiaridade com a ferramenta *Dtrace* esta que será útil em trabalhos futuros nomeadamente uso desta ferramenta para a análise e deteção de problemas de performance de aplicações existentes tal como a própria monitorização e recolha de estatísticas sobre o uso de vários sistemas.

## References

- [1] *dtrace.org*. 2020. URL: <http://dtrace.org/blogs/about/>.
- [2] *DTracing Hardware Cache Counters*. 2013. URL: <https://www.joyent.com/blog/dtracing-hardware-cache-counters>.
- [3] *Oracle Linux DTrace Tutorial*. 2019.
- [4] *Sun HPC ClusterTools 8 Software User's Guide*. 2008. URL: <https://docs.oracle.com/cd/E19356-01/820-3176-10/Dtrace-mpiperuse.html>.