

# Lição 6



## Observabilidade

# Objetivos

Ao final desta lição, o estudante será capaz de:

- Trabalhar com o conjunto de ferramentas DTrace
- Realizar comandos no Solaris que permitem ver em ação: CPU, processos, memória e E/S

# DTrace

- Conjunto de ferramentas do Solaris 10
- Desenvolvido como uma ferramenta de diagnóstico do sistema
- Utiliza probes
- Probes mantêm um registro de tudo



# Probes BEGIN e END

- O probe BEGIN é iniciado quando um processo é iniciado
- A sessão será executada quando o probe listado na descrição for ativado e o predicado for verdadeiro

**alo.d**

```
BEGIN
{
  trace ("Alô Mundo!");
}
END
{
  trace ("Adeus!");
}
```



# Organização dos probes

- Probes são identificados pelos ID e pelo seu nome
- Provedores são módulos do núcleo que contêm o código dos probes
- Exemplo de provedores:
  - DTrace
  - Lockstat
  - Profile
  - Syscall
  - VMinfo
  - Proc
  - Sched

# Variáveis

- Variáveis no DTrace não possuem tipos de dados, isso significa que este é determinado durante a primeira atribuição de valor
- O `dtrace::BEGIN` é comumente usado para inicializar variáveis

## **countdown.d**

```
dtrace:::BEGIN
{
    ctr = 10;
}
profile:::tick-1sec
{
    trace(ctr);
    ctr--;
}
```

```
dtrace:::END
{
    trace("Obrigado por usar meu
programa.");
}
```



# Predicados

- Predicados agem como uma instrução para um *script*  
**countdown.d**

```
dtrace:::BEGIN
{
    ctr = 10;
}
profile:::tick-1sec
/ ctr > 0 /
{
    trace(ctr);
    ctr--;
```

```
profile:::tick-1sec
/ ctr == 0/
{
    trace(ctr);
    exit(0);
}
dtrace:::END
{
    trace("0 tempo acabou!");
}
```



# Comando printf

- mostra o valor de um determinado atributo na tela
- Sintaxe:

```
printf("O tempo é %d segs. %s", ctr, msg);
```

- Tipo do atributo:
  - %d
  - %s
  - %f
  - %x
  - %%





# Funções agregadas

- A sintaxe básica da função agregada é:
- `@name[key] = aggfunc(args)`
- Lado Esquerdo:
  - **@** indica que estamos definindo uma função agregada
  - **name** é um nome qualquer para o agregado
  - **key** é um atributo cujos valores tornam-se índices do array
- Lado Direito:
  - `count()`
  - `sum(exp)`
  - `avg(exp)`
  - `min(exp)` e `max(exp)`
  - `quantize(exp)`



# Scripts no DTrace

```
syscall::read:entry
{
    t = timestamp;
}
```

```
syscall::read:return
{
    delay = timestamp - t;
    printf("%s(%d) tempo em  
%s: %d nsecs\n",  
execname, pid, probefunc,  
delay);  
    t = 0;  
}
```

```
syscall::read:entry
{
    self->t = timestamp;
}
```

```
syscall::read:return
{
    self->delay = timestamp  
- self->t;  
    printf("%s(%d) tempo no  
método %s: %d nsecs\n",  
execname, pid,  
probefunc, self-  
>delay);  
    self->t = 0;  
}
```



# Scripts no DTrace

```
syscall::read:entry  
{  
    self->t = timestamp;  
}
```

```
syscall::read:return  
{  
    self->delay = timestamp - t;  
    @[execname] = quantize[self->delay];  
    self->t = 0;  
}
```



# DTrace toolkit

- Diretório do DTrace:
  - apps/ - scripts específicos de aplicação
  - cpu / - scripts para análise de CPU
  - disk/ - scripts para análise de I/O
  - docs/ - documentação
  - mem/ - scripts para análise de memória
  - proc/ - scripts para análise de processos



# Informações da CPU

- Comando vmstat
  - kthr – número de threads nos seguintes estados:
    - r, b e w
  - memory – memória utilizada em kilobytes
    - swap e free
  - page – informação sobre como a memória está sendo utilizada
  - disk – informação sobre as operações em disco por segundo
  - faults – informação sobre os traps do sistema
  - cpu – percentual de uso da CPU
    - us, sy e id



# Comando uptime

```
# uptime
```

```
10:50am  up 3 day(s), 5 min(s), 2 users,  
load average: 0.11, 0.04, 0.02
```



# Scripts no DTrace

- No diretório /opt/DTT/cpu
  - cputypes.d – lista a informação sobre cada CPU
  - loads.d – mostra a média de carga
  - intbycpu.d – mostra o número de interrupções manipuladas por cada CPU
  - runocc.d – mostra as execuções que estão em uma queue

# Processos

```
# ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	0	0	0	Dec 10	?	0:12	sched

- **UID** – identificação do usuário do processo
- **PID** – identificação do processo
- **PPD** – identificação do processo pai
- **C** – coluna obsoleta
- **STIME** – tempo de inicialização de um processo
- **TTY** – terminal de controle
- **TIME** – tempo que um processo está executando na CPU
- **CMD** – comando usado para inicializar o processo





# Scripts para processos no DTrace

- No diretório /opt/DTT/proc
  - sampleproc – inspeção em muitas CPUs
  - writebytes.d e readbytes.d – leitura e escrita dos bytes
  - syscallbyproc.d e syscallbypid.d – sistema de chamadas
  - filebyproc.d – lista de arquivos abertos
  - crash.d – aplicações que falharam



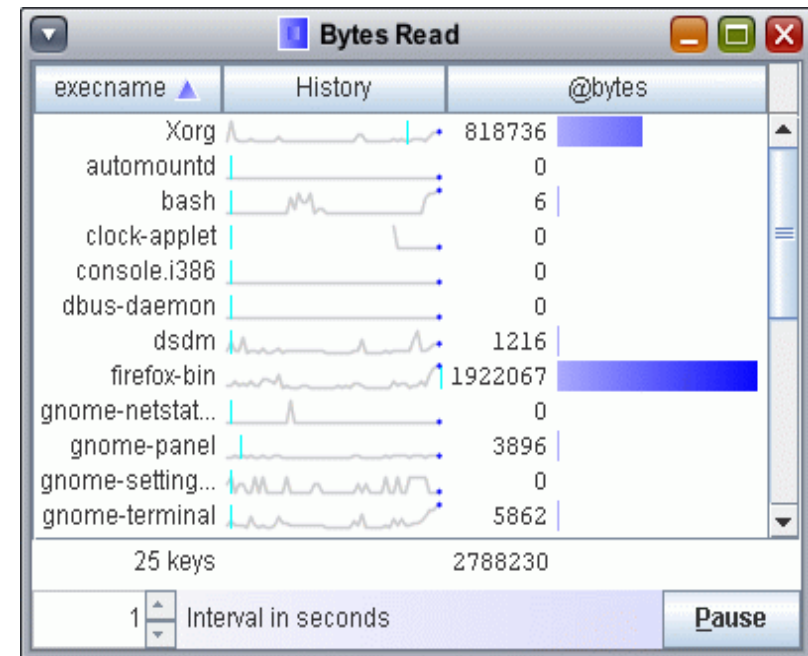
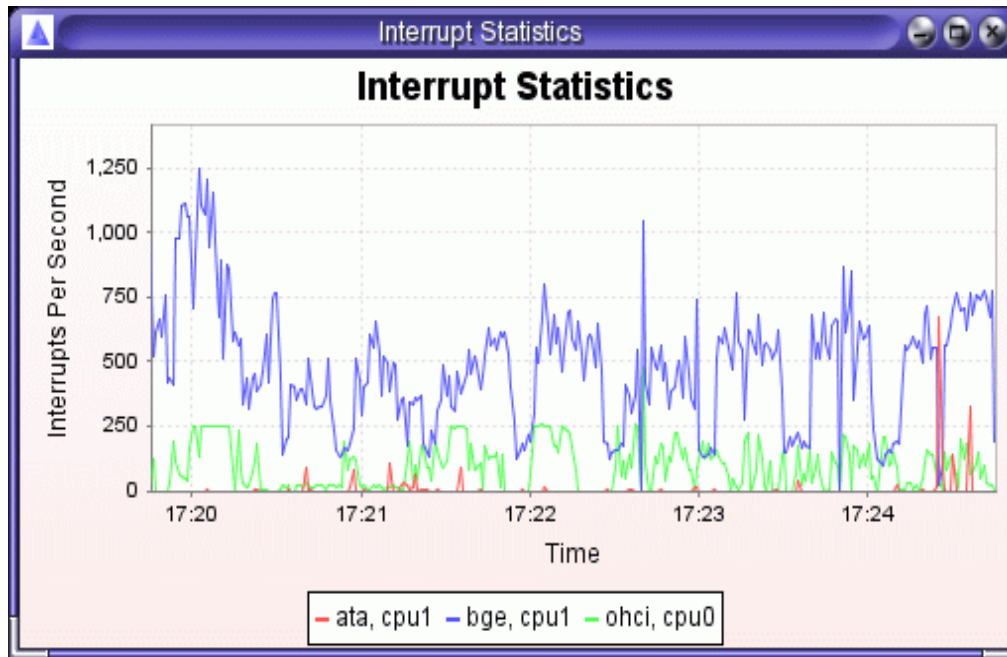
# Scripts para Memória no DTrace

- No diretório /opt/DTT/mem
  - vmstat.d – utilizado para escrever em D
  - xvmstat – mostrar mais informações em relação ao vmstat
  - swapinfo.d – mostra informações da memória virtual
  - minfbypid.d – detecta um grande consumidor de memória

# Scripts para Disco Rígido no DTrace

- No diretório /opt/DTT/disk
  - iofile.d – mostra o tempo de espera para entrada e saída
  - diskhits – verifica a E/S e a média de carga de um arquivo
  - iotop – lista os eventos de E/S por processo
  - iosnoop – monitora eventos de E/S

# Chime



<http://www.opensolaris.org/os/project/dtrace-chime/>

# Sumário

- DTrace
  - Uso de Probes
  - Predicados, comando printf, funções agregadas
  - Toolkit



# Parceiros

- Os seguintes parceiros tornaram JEDI<sup>TM</sup> possível em Língua Portuguesa:

