

Exercício orientado: Monitoramento de atividade de processo

Instruções

1. Na máquina `workstation`, abra duas janelas de terminal lado a lado. Nesta seção, esses terminais são referidos como *esquerdo* e *direito*. Em cada terminal, faça login na máquina `servera` com o usuário `student`.

```
[student@workstation ~]$ssh student@servera...output omitted...  
[student@servera ~]$
```

2. No shell do terminal esquerdo, use o diretório `/home/student/bin`. No novo diretório, crie um script de shell chamado `monitor`, que gera carga artificial de CPU. Torne o arquivo de script `monitor` executável.

- a. Crie o diretório `/home/student/bin`.

```
[student@servera ~]$mkdir /home/student/bin
```

- b. Crie o arquivo do script no diretório `/home/student/bin` com o conteúdo exibido.

```
[student@servera ~]$vim /home/student/bin/monitor  
#!/bin/bash  
while true; do  
    var=1  
    while [[ var -lt 60000 ]]; do  
        var=$((var+1))  
    done  
    sleep 1  
done
```

Nota

O script de `monitor` é executado até o processo ser encerrado. Ele gera carga de CPU artificial executando sessenta mil cálculos de adição. Depois de gerar o carregamento de CPU, ele fica suspenso por um segundo, redefine a variável e repete.

- c. Torne o arquivo `monitor` executável.

```
[student@servera ~]$chmod a+x /home/student/bin/monitor
```

3. No shell do terminal direito, execute o comando `top`. Redimensione a janela para visualizar o conteúdo do comando.

```
[student@servera ~]$top
top - 12:13:03 up 11 days, 58 min,  3 users,  load average: 0.00, 0.00, 0.00
Tasks: 113 total,  2 running, 111 sleeping,  0 stopped,  0 zombie
%Cpu(s):  0.2 us,  0.0 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 1829.4 total, 1377.3 free, 193.9 used, 258.2 buff/cache
MiB Swap: 1024.0 total, 1024.0 free,  0.0 used. 1476.1 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM     TIME+ COMMAND
5861 root        20   0     0     0     0  I   0.3   0.0   0:00.71 kworker/1:3-events
6068 student    20   0 273564  4300  3688 R   0.3   0.0   0:00.01 top
    1 root        20   0 178680 13424  8924 S   0.0   0.0   0:04.03 systemd
    2 root        20   0     0     0     0  S   0.0   0.0   0:00.03 kthreadd
    3 root         0 -20     0     0     0  I   0.0   0.0   0:00.00 rcu_gp
...output omitted...
```

4. No shell de terminal esquerdo, determine o número de CPUs lógicas na máquina virtual.

```
[student@servera ~]$lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 2...output omitted...
```

5. No shell de terminal esquerdo, execute uma única instância do executável arquivo de script `monitor` em segundo plano.

```
[student@servera ~]$monitor &
[1] 6071
```

6. No shell do terminal direito, monitore o comando `top`. Use os pressionamentos de teclas simples `l`, `t` e `m` para alternar entre linhas de cabeçalho de carga, threads e memória. Depois de observar esse comportamento, certifique-se de que todos os cabeçalhos estão sendo exibidos.
7. Observe a ID de processo (PID) para o processo `monitor`. Visualize o percentual de CPU para o processo, o qual se espera que esteja entre 15% e 25%.

```
[student@servera ~]$top
PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM
TIME+ COMMAND
071 student   20   0  222448    2964    2716 S   18.7   0.2
0:27.35 monitor
...output omitted...
```

Visualize as médias de carga. O valor da média de carga de um minuto é, atualmente, menor que 1. O valor observado poderá ser afetado pela contenção de recursos de outra máquina virtual ou do host virtual.

```
top - 12:23:45 up 11 days,  1:09,  3 users,  load averag
e: 0.21, 0.14, 0.05
```

8. No shell de terminal esquerdo, execute uma segunda instância do executável arquivo de script `monitor` em segundo plano.

```
[student@servera ~]$monitor &  
[2] 6498
```

9. No shell de terminal direito, observe a ID do processo (PID) para o segundo processo `monitor`. Visualize o percentual de CPU para o processo, o qual também se espera que esteja entre 15% a 25%.

```
[student@servera ~]$top  
PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %ME  
M      TIME+ COMMAND  
6071 student   20   0   222448    2964    2716 S   19.0   0.  
2    1:36.53 monitor  
6498 student   20   0   222448    2996    2748 R   15.7   0.  
2    0:16.34 monitor  
...output omitted...
```

Novamente, visualize a média de carga de um minuto, a qual ainda será menor que 1. Aguarde pelo menos um minuto para que o cálculo se ajuste à nova carga de trabalho.

```
top - 12:27:39 up 11 days,  1:13,  3 users,  load averag  
e: 0.36, 0.25, 0.11
```

10. No shell de terminal esquerdo, execute uma terceira instância do executável arquivo de script `monitor` em segundo plano.

```
[student@servera ~]$monitor &  
[3] 6881
```

11. No shell de terminal direito, observe a ID do processo (PID) para o terceiro processo `monitor`. Visualize o percentual de CPU para o processo, o qual também se espera que esteja entre 15% e 25%.

```
[student@servera ~]$top  
PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %ME  
M      TIME+ COMMAND
```

M	TIME+	COMMAND
6881	student 20 0 222448 3032 2784 S 18.6 0.	
2	0:11.48	monitor
6498	student 20 0 222448 2996 2748 S 15.6 0.	
2	0:47.86	monitor
6071	student 20 0 222448 2964 2716 S 18.1 0.	
2	2:07.86	monitor

Para que a média de carga fique acima de 1, você deve iniciar mais processos `monitor`. A configuração da sala de aula tem duas CPUs, portanto, somente três processos não são suficientes para enfatizá-la. Inicie mais três processos `monitor` em segundo plano. Visualize a média de carga de um minuto novamente, a qual agora espera-se que seja maior que 1. Aguarde pelo menos um minuto para que o cálculo se ajuste à nova carga de trabalho.

```
[student@servera ~]$monitor &
[4] 10708
[student@servera ~]$monitor &
[5] 11122
[student@servera ~]$monitor &
[6] 11338
```

```
top - 12:42:32 up 11 days, 1:28, 3 users, load averag
e: 1.23, 2.50, 1.54
```

12. Quando terminar de observar os valores de média de carga, encerre cada um dos processos `monitor` no comando `top`.

- No shell do terminal, pressione **k** para observar o prompt abaixo dos cabeçalhos e acima das colunas.

```
...output omitted...
PID to signal/kill [default pid = 11338]
```

- O prompt escolhe os processos de `monitor` no topo da lista. Pressione **Enter** para eliminar o processo.

```
...output omitted...  
Send pid 11338 signal [15/sigterm]
```

- c. Pressione **Enter** novamente para confirmar o sinal SIGTERM 15.

Verifique se o processo selecionado não está mais presente no comando `top`. Se a PID existir, repita essas etapas para encerrar os processos e substitua o sinal SIGKILL 9 quando solicitado.

```
 6498 student    20    0  222448    2996    2748 R   22.9  
0.2   5:31.47 monitor  
 6881 student    20    0  222448    3032    2784 R   21.3  
0.2   4:54.47 monitor  
11122 student    20    0  222448    2984    2736 R   15.3  
0.2   2:32.48 monitor  
 6071 student    20    0  222448    2964    2716 S   15.0  
0.2   6:50.90 monitor  
10708 student    20    0  222448    3032    2784 S   14.6  
0.2   2:53.46 monitor
```

13. Repita a etapa anterior para cada processo de `monitor` restante. Verifique se nenhum processo `monitor` permanece no comando `top`.
14. No shell de terminal direito, pressione **q** para sair do comando `top`. Feche o terminal direito.
15. Retorne à máquina `workstation` como o usuário `student`.

```
[student@servera ~]$exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Encerramento

Na máquina `workstation`, altere para o diretório pessoal do usuário `student` e use o comando `lab` para concluir este exercício. Essa etapa é importante para garantir que recursos de exercícios anteriores não afetem exercícios futuros.

```
[student@workstation ~]$lab finish processes-monitor
```

Isso conclui a seção.