Universidade Federal de Campina Grande – UFCG
Centro de Engenharia Elétrica e Informática – CEEI
Departamento de Sistemas e Computação – DSC
Professor: Reinaldo Gomes
Disciplina: Administração de Sistemas

Prática 10 – Processos no Linux

Essa prática é destinada ao entendimento de alguns dos principais comandos sobre gerenciamento de processos no linux.

1. Liste os processos existente no sistema operacional.

```
$ ps
$ ps -u (exibe mais detalhes sobre os processos)
```

2. Liste os processos existente no sistema operacional que também pertençam a outros usuários (-a):

```
$ ps -au
```

3. Liste os processos existente no sistema operacional que também pertençam a outros usuários (-a) e que não estão vinculados a terminais (-x).

```
$ ps -aux
```

4. Liste os processos ativos, exibindo apenas as seguintes informações: UID, PID, PPID, COMMAND e STAT.

```
$ ps -eo uid,pid,ppid,cmd,stat
```

5. Execute a sequencia de comandos abaixo correspondente ao um script simples e a execução desse script em *backgroud* (segundo plano).

```
$ echo 'echo ola' > teste.sh
$ echo 'sleep 100' >> teste.sh
$ echo 'echo adeus' >> teste.sh
$ chmod a+x teste.sh
$ ./teste.sh & (pressione <enter> para retornar ao prompt de comando)
```

6. Execute o comando abaixo para exibir os processos em *background*. Observa a diferença entre os comandos abaixo.

```
$ jobs
$ jobs -l
```

7. Mate o processo que está em executando em backgroud.

```
$ kill %1
```

8. O comando kill tem como objetivo enviar sinais para os processos. Há vários sinais diferentes que o kill pode enviar para um processo. Veja a lista de todos os sinais conhecidos.

```
$ kill -l
```

A função de cada sinal pode ser obtida consultando a man page: "**$ man 7 signal**".

9. Execute os comandos conforme orientação a seguir.

Inicie três comandos ping no mesmo shell, executando-os em *backgroud*.

```
$ ping 1.1.1.1 &> /dev/null &
$ ping 2.2.2.2 &> /dev/null &
$ ping 3.3.3.3 &> /dev/null &
```

Liste os processos em backgroud.

```
$ jobs -l
```

Volte a executar um dos processos em *foreground* (primeiro plano).

```
$ fg %2 (ou $ fg 2)
```

Supondo que seja necessário trocar o estado de Executando (*Running*) do segundo processo para Parado (*Stopped*). No terminal com a segunda tarefa em execução, conforme comando anterior, pressione a combinação de teclas Ctrl+z (referenciada como **^Z**). Veja a ilustração abaixo.

```
analista@debian:~$ fg %2
ping 2.2.2.2 &>/dev/null
^Z
[2]+  Parado                 ping 2.2.2.2 &>/dev/null
```

Liste novamente os processos em *backgroud*.

$ jobs -l (ou somente $jobs)

Observe que a tarefa 2 está em estado de *Stopped*. Para trazer novamente a tarefa para o estado *Running*, execute os comandos abaixo.

$ kill -cont %2 ou ($ kill -sigcont %2) ou ($ kill -cont PID)

$ jobs

Outras formas de visualizar informações sobre processos:

$ pgrep ping

$ pgrep ping -d ' ' (seleciona um espaço como novo delimitador para o PIDs exibidos)

$ pgrep ping -l -f (exibe o PID e a linha de comando de todos os processos com nome ping)

$ pgrep ping -l -f -o (exibe o processo mais antigo iniciado pelo sistema com o nome ping)

$ pgrep ping -l -f -n (exibe o processo mais novo iniciado pelo sistema com o nome ping)

$ pgrep ping -l -f -v (exibe os processos que não tenham o nome ping)

Temos também a opção de usar o comando pidstat para vermos informações mais detalhadas de algum processo de maneira contínua:

$ pidstat -p pid tempo_atualizacao

Agora mate todos os processos ping.

$ killall ping (ou pkill ping)

$ jobs

Caso os processos ainda persistam em memória, execute o comando: **$ killall -9 ping**

10. Para visualizar informações sobre processos em tempo real, execute os comandos a seguir.

$ top

$ top -d 5 (define o período de atualização)

$ top -p pid1 [-p pid2 ...] (define processos a serem exibidos)

Informações de memória RAM e swap são obtidas no inicio do comando top. Digite a tecla **q** para sair do comando top.

11. Outro comando para exibir informações sobre a quantidade de memória disponível e swap.

$ free -h

12. Para exibir informações sobre a hora atual, quanto tempo o sistema está ativo, a quantidade de usuário conectados no sistema e a carga de trabalho, execute o comando a seguir.

$ uptime

13. Para exibir informações do comando uptime e quem está conectado ao sistema, execute o comando a seguir.

$ w

14. Execute os comandos abaixo para manipular com a prioridade de um processo.

Atribuindo o parâmetro nice para um processo. Observe que por padrão o nice é zero e a prioridade é 20. O comando abaixo configura o nice como 10 e altera a prioridade do processo (soma 10 a prioridade do processo).

$ nice -n 10 sleep 300 &

Observe a prioridade do processo com o comando abaixo e anote o PID do processo.

$ ps -lax | grep sleep

(soma 10 a prioridade do processo).

```
$ renice 15 PID (ou renice -n 15 -p PID)
$ ps -lax | grep sleep
```

15. Outra opção que também podemos considerar para o monitoramento de CPU e memória é usar o comando **vmstat.**

```
$ vmstat [tempo_atualizacao]
```

A saída se divide em categorias: processos, memória, swap, uso de disco, system para a indicação do número de vezes que o kernel alterna para executar seu código e CPU.

```
procs -----------memória--------- ---swap-- ----e/s---- -sistema- ------cpu-----
 r  b   swpd  livre   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 14138992  79920 910804    0    0   442    19  123  284  3  1 85 12  0
 0  0      0 14143424  79964 911036    0    0    18   184  235  568  0  0 98  2  0
 0  0      0 14143432  79964 910908    0    0     0     0  175  382  0  0 100  0  0
 0  0      0 14143424  79972 910908    0    0     0     6  172  393  0  0 100  0  0
 0  0      0 14143416  79972 910908    0    0     0     0  167  362  0  0 100  0  0
 0  0      0 14143804  79972 910908    0    0     0     0  186  411  0  0 100  0  0
 0  0      0 14143796  79972 910908    0    0     0    20  168  353  0  0 100  0  0
 0  0      0 14143584  79972 911168    0    0     0     0  491 2234  2  1 98  0  0
 2  1      0 14044124  80812 986288    0    0 16500     0 1081 3941  2  1 89  7  0
 0  0      0 14027252  81664 993092    0    0  7034     0  671 15559  1  1 90  8  0
 0  0      0 14046056  81664 973400    0    0     0     0  590 3744  1  0 98  0  0
 0  0      0 14050288  81672 968924    0    0     4   252  578 3024  1  0 98  1  0
 0  0      0 14060968  81704 978552    0    0   278    90  571 5114  1  0 97  1  0
 0  0      0 14079756  81704 959688    0    0     0     0  214  560  0  0 100  0  0
 0  0      0 14080004  81704 959992    0    0     0     0  342 1528  0  0 99  0  0
```

16. Para monitorarmos especificamente as operações em disco ao invés de usarmos apenas as informações apresentadas no vmstat pode usar o **iostat**, que nos fornece uma riqueza maior de informações.

```
$ iostat [tempo_atualizacao]
```

Nele são apresentadas informações de transferência de dados por segundo, dados lidos por segundo, dados escritos por segundo, total de dados lidos e total de dados escritos, todos em kilobytes.

```
Linux 4.15.0-118-generic (Thunder)      07/10/2020      _x86_64_        (8 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1,61    0,00    0,75   15,05    0,00   82,60

Device             tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
loop0             0,39         8,45         0,00       1061          0
loop1             0,28         0,88         0,00        110          0
loop2             0,48         8,48         0,00       1064          0
loop3             0,31         0,91         0,00        114          0
loop4             0,14         0,35         0,00         44          0
loop5             0,41         8,41         0,00       1055          0
loop6             0,14         0,35         0,00         44          0
loop7             0,28         0,88         0,00        110          0
sda             160,99      4135,90       161,46     519014      20261
loop8             0,41         8,48         0,00       1064          0
loop9             0,41         8,37         0,00       1050          0
loop10            0,33         2,69         0,00        337          0
loop11            0,41         8,42         0,00       1057          0
loop12            0,33         2,69         0,00        337          0
loop13            0,44         8,48         0,00       1064          0
loop14            0,14         0,35         0,00         44          0
loop15            0,31         2,66         0,00        334          0
loop16            0,31         2,64         0,00        331          0
loop17            0,40         8,46         0,00       1062          0
loop18            0,32         0,92         0,00        115          0
loop19            0,41         8,48         0,00       1064          0
loop20            0,15         0,37         0,00         46          0
loop21            0,39         8,46         0,00       1062          0
loop22            0,04         0,06         0,00          8          0
```