Processos no LINUX

O que sao processos?
A definicao de processo apresentada por Tanenbaum no livro **Sistemas Operacionais**

- Projeto e Implementacao:
Aideia-chave aqui e que umprocesso e umtipo de atividade. Ele tem umprograma, entrada, saida e um estado. Um unico processador pode ser compartilhado entre vários processos, com algum algoritmo de agendamento sendo utilizado para determinar quando parar de trabalhar em um processo e servir a um diferente.
Pense em um processo como a representacao de um programa em execucao utilizando os recursos do computador para realizar alguma tarefa.

Um processo possui estados que definem o seu comportamento, sao eles:

• execucao: o processo esta ativo utilizando a CPU e outros recursos;
• pronto ou espera: o processo esta temporariamente parado permitindo que outro processo execute na sua frente;
• bloqueado: o processo esta parado aguardando a execucao de algum evento para voltar ao estado de execucao.

Alem de possuir esses comportamentos, um processo e capaz de criar outros processos. Quando isso ocorre dizemos que um processo e pai dos outros criados por ele.

Cada processo no Linux recebe um numero para sua identificacao conhecido por PID. Podemos ve-los com o comando ps:

anhanguera:~$ ps aux

O processo init

Quando inicializamos o Linux, o primeiro processo criado e o init, que e conhecido como o pai de todos os outros.

Depois de inicializar o ambiente grafico, quando abrimos o terminal, um processo e criado para controlar o terminal em questao. De forma semelhante, para cada programa aberto tambem sera criado um processo correspondente. O init e responsavel por inicializar todos eles e possui a identificacao de numero 1 no sistema.

A identificacao de processos

Como vimos, um processo e identificado por seu PID (Process Identifier). Esse numero e dado pelo sistema para cada processo, cada PID e unico, entao voce nunca vera dois ou mais processos fazendo uso do mesmo PID.

Cada processo tambem possui um usuario dono, dessa forma, o sistema verifica as permissoes e sabe qual usuario pode executar um determinado processo. A identificacao de donos e feita pelos numeros UID e GID.

No Linux, todo usuario possui um numero de identificacao damesma forma que

os processos. Esse numero e conhecido por UID (User Identifier) e o GID (Group Identifier).

Verificando processos

Verificar e gerenciar processos e uma tarefamuito importante, pois as vezes precisamos interromper um processo a forca ou verificar quais processos estao consumindo mais recursos no computador (CPU, memoria etc).

Veremos alguns comandos novos. Vamos comecar pelo comando ps, que serve para listar os processos em execucao e obter informacoes como PID e UID. Execute o ps para obter a lista de processos do nosso usuario:

```
anhanguera:~$ ps
```

Assim como todos os comandos que vimos ate agora, o ps tambem possui opcoes. Veremos algumas delas:

- a: lista todos os processos existentes;
- u: exibe o nome do usuario dono do processo;
- x: lista os processos que nao possuem relacao com o terminal;
- m: exibe a quantidade memoria consumida por cada processo.

Paramais opcoes man ps, a combinacaomais usada e ps aux – execute e veja a lista de processos em execucao:

```
anhanguera:~$ ps aux

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.3 3516 1908 ? Ss 18:49 0:00 /sbin/init
root 2 0.0 0.0 0 0 ? S 18:49 0:00 [kthreadd]
root 3 0.0 0.0 0 0 ? S 18:49 0:00 [ksoftirq]
root 5 0.0 0.0 0 0 ? S 18:49 0:00 [kworker/]
root 6 0.0 0.0 0 0 ? S 18:49 0:00 [migratio]
root 7 0.0 0.0 0 0 ? S 18:49 0:00 [watchdog]
root 8 0.0 0.0 0 0 ? S< 18:49 0:00 [cpuset]
root 9 0.0 0.0 0 0 ? S< 18:49 0:00 [khelper]
root 10 0.0 0.0 0 0 ? S 18:49 0:00 [kdevtmpf]
```

Note que podemos ver o usuario dono de cada processo na primeira coluna USER. Em seguida temos o PID de cada processo, informacoes de consumo de CPU e memoria, outras informacoes como data e hora de quando o processo foi inicializado, assim como o nome do processo.

Podemos contar todos os processos em execucao fazendo uso de um comando que ainda nao testamos: o wc. Basicamente o que ele faz e contar as linhas de um arquivo ou do conteudo que for exibido no terminal, para isso utilizamos a opcao wc -l. Para informar ao comando que queremos apenas a quantidade de linhas, o que faremos e executar o ps aux em combinacao com o wc -l, usando o | (pipe), que e uma forma de encadeamento de processos:

```
anhanguera:~$ ps aux | wc -l
```

O que o | fez foi encadear a execucao do comando ps aux ao comando wc -l. Assim, ele pegou o resultado gerado do primeiro comando e passou para o segundo.

A saida do wc nos mostra quantos processos ao todo temos rodando em nosso computador.

Outro comando que ainda nao vimos e o grep, que procura por uma expressao que pode ser uma palavra ou frase em um arquivo, ou ainda pode funcionar como filtro na saida de comandos. Vamos usa-lo para filtrar a lista de processos gerada com ps -A e buscar todos os processos do apache que estiverem em execucao. A opcao -A do comando ps e para exibir todos os processos em execucao mas sem detalhes.

```
anhanguera:~$ ps -A | grep apache

1166 ? 00:00:00 apache2
1184 ? 00:00:00 apache2
1185 ? 00:00:00 apache2
1186 ? 00:00:00 apache2
1187 ? 00:00:00 apache2
1188 ? 00:00:00 apache2
anhanguera:~$
```

Foram retornados somente os processos com nome apache, pois o grep realizou um filtro para ignorar todo o resto e exibir o que procuravamos. Experimente executar ps -A para ver o tamanho da lista de processos.

Veremos agora outro comando bastante usado para verificar processos, o top, que acompanha os processos atualizando as informacoes quase em tempo real.

Execute no terminal o top para ve-lo funcionando.

Na parte superior, logo nas primeiras linhas, temos informacoes sobre o sistema, com numero total de processos, uso da CPU, uso da memoria. Em seguida, temos a lista dos processos existentes. Para obter ajuda sobre o uso, tecle h e, para sair do top, tecle q. O top possui muitas opcoes, veja algumas em man top.

Uma opcao interessante e acompanhar os processos de um determinado usuario do sistema. Para isso usamos a opcao -u e o nome do usuario:

```
anhanguera:~$ top -u daniel
```

Dessa forma, estamos executando um filtro para exibir somente os processos pertencentes ao usuario anhanguera.

Uma outra opcao ao top e o htop, que tem uma interface mais amigavel. Ele nao vem instalado por padrao, entao vamos instalar para conhece-lo:

```
anhanguera:~$ sudo apt-get install htop
```

Apos a instalacao, execute htop:

Bem melhor, nao e mesmo?

Agora e possivel identificar rapidamente as informacoes. O consumo de CPU e memoria ficou mais amigavel, e na barra inferior existe um menu com opcoes. Por exemplo, tecle F1 para obter ajuda:

Navegue um pouco no htop, leia a documentacao, para sair tecle q ou F10.

O que sao sinais de processos?

No Linux os sinais sao uma forma de comunicacao usada pelos processos para que o sistema consiga interferir em seu funcionamento. Na pratica, ao receber um sinal com instrucoes, um processo interpreta a acao que foi especificada no sinal e a executa.

Alguns dos sinais mais conhecidos e usados por processos sao:

• KILL: sinal com funcao de encerrar um processo;
• TERM: termina o processo apos ele finalizar uma tarefa;
• STOP: interrompe a execucao de um processo;
• CONT: ativa a execucao de um processo que foi interrompido.

Para entendermelhor sobre sinais, vamos verumpouco do comando kill. Este comando e usado para o envio de sinais a processos, sua sintaxe e simples e depende apenas do PID de um processo.

Vamos interromper a execucao do mysql enviando um sinal de STOP para o seu processo. Precisamos primeiro do numero PID do mysql:

```
anhanguera:~$ ps -A | grep mysql
1047 ? 00:00:00 mysqld
anhanguera:~$
```

Agora que temos o numero PID do processo do mysql, vamos enviar o sinal de STOP para ele, interrompendo sua execucao:

```
anhanguera:~$ sudo kill -STOP 1047
anhanguera:~$
```

Para entender o que realmente ocorreu, tente conectar ao servico do mysql:

```
anhanguera:~$ sudo mysql -u root -p
Enter password:
```

Nada vai acontecer, ficaremos o dia inteiro olhando esta tela pois o processo de execucao do mysql foi interrompido. Abra um novo terminal, envie o sinal CONT para ativar a execucao do processo que foi interrompido e veja o que acontece no terminal onde estamos tentando nos conectar ao mysql:

```
anhanguera:~$ sudo kill -CONT 1047
anhanguera:~$
```

Ao executar o envio do sinal CONT para o processo interrompido do mysql, veja que ele voltou a funcionar normalmente.

O comando kill tambem pode ser usado com o numero do sinal em vez do seu nome, por exemplo, -9 representa o sinal KILL. Para entender melhor o que o sinal KILL faz, abra o editor vim em um terminal e em outro terminal busque pelo PID do editor. Em seguida, envie o sinal -9 para o processo:

```
anhanguera:~$ ps -A | grep vim

2488 pts/0 00:00:00 vim

anhanguera:~$ sudo kill -9 2488
[1]+ Killed vim
anhanguera:~$
```

Ao enviar o sinal KILL para o processo do vim, ele e encerrado. Recebemosuma mensagem informando que o processo foi morto, ou seja, a execucao do aplicativo foi finalizada.

Outro comando bastante usado e o killall, quando nao sabemos o PID ou quando temos varios processos domesmo aplicativo comvarios PIDs em execucao, por exemplo o apache:

```
anhanguera:~$ ps -A | grep apache

2548 ? 00:00:00 apache2
2553 ? 00:00:00 apache2
2554 ? 00:00:00 apache2
2555 ? 00:00:00 apache2
2556 ? 00:00:00 apache2
2557 ? 00:00:00 apache2
anhanguera:~$
```

Note que temos varios PIDs e, por isso, enviar um sinal de cada vez para cada PID torna-se complicado. Neste caso usamos o killall e o nome do processo em vez do PID. Para interromper a execucao do processo, fariamos:

```
anhanguera:~$ sudo killall -STOP apache2
anhanguera:~$
```

Verifique acessando o endereco http://localhost e note que a pagina inicial do apache nao ira carregar. Envie o sinal CONT para retornar a execucao normal do processo:

```
anhanguera:~$ sudo killall -CONT apache2
anhanguera:~$
```

Leiamais sobre o kill e o killall em suas documentacoes – esses comandos sao importantes e bastante utilizados.

Os estados de um processo

Basicamente existem 4 estados para um processo. Apos sua criacao, o seu estado corrente e executavel; quando um processo esta aguardando alguma rotina para ser executado, dizemos que ele esta dormindo – esse estado e chamado de dormente –; se um processo esta congelado e por algum motivo nao pode ser executado, dizemos que seu estado e parado; se um processo e considerado morto, ou seja, foi finalizado, nao esta mais em execucao mas por algum motivo ainda existe, dizemos que seu estado e de um processo zumbi.

Processos e suas prioridades

Durante sua execucao, um processo pode ter prioridade em relacao aos outros. Para entendermelhor como funcionamas prioridades, vamos ver o conceito de gentileza. Imagine um processo em execucao sendo gentil (oferecendo a gentileza) de deixar um processo com prioridade maior passar a sua frente e ser executado antes. Os processos trabalham com niveis de gentileza, que podem ser definidos atraves do comando nice e um numero entre -19 e 19, que determina o quao gentil um processo pode ser. Quantomaior for o numero definido,mais gentil o processo sera, logo quanto menor for o numero, maior a sua prioridade.

Normalmente nao precisamos determinar as prioridades de um processo pois o Linux trabalha de forma inteligente para fazer isso.

Emalguns casos, por exemplo tarefas de backup, setamos a prioridade para que o processo nao consuma de forma inesperada recursos do computador comomemoria e CPU. Nesses casos, podemos usar 2 comandos: o nice e o renice.

Por enquanto, vamos testar o renice e alterar a prioridade do processo que esta

executando o mysql:
anhanguera:~$ ps -A | grep mysql
2340 ? 00:00:03 mysqld
anhanguera:~$

Agora que temos o PID do processo podemos alterar sua prioridade, para isso faca:

anhanguera:~$ sudo renice -19 2340
2340 (process ID) old priority 0, new priority -19
anhanguera:~$

Note que o comando nos informa a prioridade antiga que era 0 para a nova prioridade que e -19. Processos com prioridade 0 sao intermediarios pois estao bem no meio dos extremos. Ao setarmos a prioridade -19, estamos informando que esse processo e pouco gentil e todos os outros deixa-lo-ao passar na frente.

Vamos alterar novamente setandouma nova prioridade e tornando esse processo gentil:

anhanguera:~$ sudo renice +15 2340
2340 (process ID) old priority -19, new priority 15
anhanguera:~$

Agora temos um processo gentil de prioridade 15, que ira permitir que outros passem a sua frente e sejam executados antes. E possivel ver a prioridade dos processos no htop:

Coluna NI mostra a prioridade dos processos

Note, na coluna NI, todas as prioridades com 0 e apenas uma opcao com valor

15 que e a do processo que foi alterado.