Laboratório de Sistemas Operacionais - Aula Prática 08 – 24.05.2016
Sistema Operacional Linux

Trabalhando com agendamento de processos

1. Esta sessão vai mostrar para você como trabalhar com agendamento de processos no ambiente Linux. Inicialmente, vamos criar um novo diretório e chamá-lo de ***Lab08*** na sua pasta atual. Use o seguinte comando: ***mkdir Lab08***. Para ver se ele foi criado, use o comando ***ls –l*** . Ok?

2. Agora, vamos seguir para a nova pasta criada. Digite ***cd Lab08***. Para ter certeza que está no diretório correto, use o comando ***pwd*** e veja se o último arquivo é o ***Lab08***.

3. **Entendendo** o agendamento de processos:
   Entende-se pelo agendamento de tarefas a programação prévia destas, que podem ser comandos, serviços, etc. a serem executados em determinado tempo e/ou com uma determinada freqüência.

   As tarefas a serem agendadas devem ser, portanto, aquelas que por inconveniente de horário ou local não poderão dispor da atenção de ninguém presente ou aquelas que se tornam rotina como backups, relatórios extensos, limpeza de diretórios, etc.

   No Linux temos pelo menos duas maneiras de se fazer esse agendamento: com o comando "**at**" e do "**crontab**".

4. Primeiro vamos entender o funcionamento do agendamento de processos usando o comando (**at**):

   A sintaxe do comando é: **at –[flags] TIME**

   As principais flags do comando at são:

- ***v*** Imprime o número de versão do programa
- ***f*** Informa ao comando para ler as instruções de um arquivo em vez de ler da entrada padrão
- ***l*** Um atalho para o comando ***atq***. Exibe uma relação de jobs agendados, com suas respectivas datas programadas para a execução, do usuário atual.
- ***d*** Um atalho para o comando atm. Remove um job especificado pelo usuário.
- ***m*** Informa ao comando para enviar um e_mail para o usuário quando sua tarefa for executada.
- ***c*** Informa o conteúdo do job informado.

O argumento TIME em sua sintaxe é composto de duas partes, que utilizamos exatamente para informar ao comando ***at***, quando a tarefa deverá ser executada.

A primeira parte do argumento é hora, em formato ***HH:MM***, que a tarefa será executada podendo-se ainda utilizar as siglas ***AM*** ou ***PM***, para horário diurno ou noturno ou o formato 24 horas. Podem ser utilizadas também as palavras-chave "***noon***", "***teatime***" e "***midnight***" indicando os horários 12:00 h, 16:00 h e 00:00 h respectivamente.

A segunda parte do argumento é utilizada para informar ao comando ***at*** o dia em que a tarefa será executada. Podemos utilizar os formatos ***MM/DD/AA***, ***MMDDAA***, ***DD.MM.AA*** ou ainda as palavras-chave "***today***" e "***tomorrow***" indicando a data atual ou o dia seguinte respectivamente.

Por exemplo:
Vamos agendar um comando que liste o conteúdo do diretório raiz e armazene seu resultado no arquivo ***arq1.txt***. Aqui foi escolhido para ser feito as ***08:00 h da manhã do dia 25.05.16***. No seu caso, coloque como horário 5 ou 6 minutos antes do horário atual e o dia de hoje. Por exemplo,

se o horário do servidor for 20h45, coloque no seu comando ***at*** o horário de 20h50.

Note que ao dar o <***enter***> logo após concluir o comando ***at***, o Linux abre um prompt do ***at*** e espera que você coloque o ***comando*** ou o ***programa*** a ser agendado. Faça isso. Ao dar o <***enter***> ele abre novamente o prompt, porque você pode agendar várias tarefas num determinado instante. Se não tiver mais nada a agendar, dê o ***Ctrl + D*** que o agendamento será fechado.

5. Bem, tudo o que você precisaria saber sobre o funcionamento do comando at já foi dito, agora vamos lá, mão na massa. Digite os comandos abaixo: (Lembre da questão do horário que falei dois parágrafos acima)!

at 08:00 25.05.16	Digite <*enter*>
at > ls –l .. > arq1.txt	Digite <*enter*>
at > <EOT>	Tecle Ctrl + D para fechar

Pronto, sua tarefa está agendada. Agora é só aguardar sua execução, mas para você não ficar parado só esperando a execução do comando, vamos fazer o passo seguinte.

6. Agora você deseja saber o que está agendado, então, o Linux oferece o comando ***atq***. O comando ***atq*** mostra as tarefas que estão agendadas, ou seja, em espera.
   - ***atq***

   6 2016-05-25 08:00 a
   >

   Conforme podemos observar no exemplo acima, a saída do comando ***atq*** nos fornece todas as informações necessárias sobre as tarefas atualmente agendadas.

O comando ***atq*** pode ser chamado pelo atalho "***at –l***" que gera a mesma saída.

7. Bem, não podemos perder de vista o agendamento que fizemos no item 5, logo acima. Então, vamos monitorar a sua execução.

   Dê o comando ***date*** e veja se já passou da hora e minuto que você havia agendado. Se não deu ainda, espere mais um pouco, e em seguida, repita o comando ***date***.

   Ah, sim, agora a data do agendamento foi ultrapassada. Então, dê o comando ***ls –l*** e veja se há no seu diretório o arquivo ***arq1.txt***. Achou? Verifique a data e hora de sua criação, não é a mesma do seu agendamento? Ok.

8. Vamos continuara a explorar a potência do agendamento. Você também pode remover tarefas agendadas e não executadas, e para fazer isso você pode usar o comando ***atrm***.

   A sintaxe do comando ***atm*** é: ***atm n*** onde ***n*** é o número do job a ser removido do agendamento.

   Vamos criar três agendamentos, conforme abaixo:
   (Novamente, os horários e data acima são para exemplo. Lembre-se de agendar sempre 5 minutos para frente do horário do seu servidor e como data, use a data do dia de hoje).

   Passo1: Vamos agendar o primeiro serviço:

   at 08:10 26.05.16 <enter>
   at > ls –l .. > arq2.txt <enter>
   at > <EOT> Ctrl + D para fechar

Passo2: Vamos agendar o segundo serviço:

at 08:15 26.05.16 <enter>

at > ls –l .. > arq3.txt <enter>

at > <EOT> Ctrl + D para fechar

Passo3: Vamos agendar o terceiro serviço:

at 08:20 26.05.16 <enter>

at > ls –l .. > arq4.txt <enter>

at > <EOT> Ctrl + D para fechar

Passo4: Verifique todos os serviços agendados e seus números de Jobs

- ***atq***

O resultado será algo parecido com a lista abaixo:

2 2016-05-26 08:10 a

3 2016-05-26 08:15 a

4 2016-05-26 08:20 a

>

Passo5: Remova o segundo job agendado

- ***atrm 3***

(Aqui o número 3 é só um exemplo. Você deve colocar o número que aparece ao dar o comando ***atq***)

Passo5: Repita o comando ***atq*** para ver os Jobs agendados

- ***atq***

O resultado será algo parecido com a lista abaixo:

2 2016-05-26 08:10 a

4 2016-05-26 08:30 a

>

O comando ***atm*** pode ser chamado pelo atalho ***at –d*** .

9. Podemos também fazer agendamento usando o comando ***crontab***.

   A sintaxe do crontab é: ***crontab –[flags]***

   O comando ***crontab***, assim como o comando ***at***, é utilizado para o agendamento de tarefas. Entretanto, utilizando-se o comando ***at*** executamos as tarefas agendadas apenas uma vez, enquanto o comando ***crontab*** é utilizado exatamente para agendarmos aquelas tarefas repetitivas e rotineiras como um backup, por exemplo.

   Suas flags são:

   l Lista a crontab atual.
   r Remove a crontab atual.
   u<usuário> Age sobre a crontab do usuário informado no lugar do argumento do usuário.
   e Edita a crontab atual.

   Para criarmos uma nova ***crontab***, devemos usar a flag –e ("***crontab –e***") o que fará com que o comando crontab execute o editor-padrão do ambiente definido por meio da variável EDITOR, criando um novo arquivo e portanto este estará vazio nesse momento. A partir de então, devemos preenchê-lo respeitando a sintaxe do ***crontab***.

   A saída do ***crontab*** é a mesma forma que se sai de um arquivo aberto pelo editor ***vi***.

   O layout do arquivo ***crontab*** contém cinco colunas iniciais, separadas pelo caractere espaço, que indicam data, hora e freqüência com que a tarefa

que estará definida a partir da sexta coluna deve ser executada. Podemos gerar quantas linhas desejarmos respeitando esse mesmo layout.

minuto	hora	dia	mês	Dia da semana	comando

Onde devemos entrar com valores aceitáveis para cada campo segundo a tabela abaixo:

Campo	Descrição
Minuto	Números de 0 a 59
Hora	Números de 0 a 23
Dia	Números de 1 a 31
Mês	Devem ser especificados números de 1 a 12 ou nomes dos meses com 3 letras como jan, fev, etc.
Dia da semana	Devem ser especificados números de 0 a 6 onde 0 é Domingo ou o nome de cada dia com três letras como Sun, mon, etc.
Comando	Deve ser especificado qualquer comando a ser executado.

Desta maneira se, por exemplo, desejarmos executar um comando qualquer todo dia 1 de cada mês às 14:00 h, a sintaxe correta para criar a primeira linha de nossa crontab seria:

0 14 1 * * rm –f /tmp/*

10. Agora que temos várias informações do comando ***crontab***, vamos explorá-lo na prática.

Seu desafio é o seguinte: Faça um backup de todos os seus arquivos feitos até hoje e coloque o resultado no arquivo que chamaremos de backupgeral.

O comando seria ***tar –cvf backupgeral ../****

Peço que você crie oito situações diferentes:

(1) Backup todos os dias as 14:00

(2) Backup apenas às segundas-feiras de cada mês as 14:00

(3) Backup nos dias 1,5,10,15,20,25 e 30 às 14:00

(4) Backup no dia 10 de cada mês, porém todas as horas de cada dia

(5) Backup conforme anterior, porém todo dia do mês às 15:00

(6) Backup conforme anterior, porém a cada 2 minutos, todos os dias do mês.

"A vida é peça de teatro que não permite ensaios. Por isso, cante, chore, ria e viva intensamente, antes que a cortina se feche e a peça acabe sem aplausos."

Chaplin