

Agendamento de tarefas



Agendamento de tarefas

Introdução

O que é o cron?

Nada mais é do que a “**agenda**” do sistema para determinar tarefas a serem executadas em horários específicos. No caso do **cron**, podemos agendar diversas tarefas administrativas, como por exemplo: desligar o sistema, fazer um backup, mandar mensagens no sistema em um determinado dia ou horário em que desejarmos que ocorra essa tarefa.

Vamos supor que você resolveu ser uma pessoa mais organizada, e, cansado de perder seus arquivos de configuração, resolveu fazer sempre um backup, afinal de contas é melhor prevenir do que remediar! Só que você não quer ficar sempre lembrando disso e é justamente aí que entra em cena o **cron**, que irá sempre verificar a agenda, chamada **crontab**, o que o shell precisa executar.



Agendamento de tarefas

Entendendo o cron

Temos que entender a estrutura do **crontab**, ou seja da agenda do cron!
O crontab tem a seguinte estrutura:

* * * * * **Tarefa**

Em um primeiro momento, é difícil de entender a estrutura do crontab, porém vamos detalhar ela a seguir.



Agendamento de tarefas

Estrutura do crontab

Para poder escrever sua tarefa, é necessário entender o que significa cada campo presente no crontab, que são separados por espaços ou pela tecla **<TAB>**.

Seria assim:

```
* * * * * tarefa
| | | | | |_____ Tarefa a ser executada
| | | | | |_____ Dia da semana (0-7)
| | | | | _____ Mês (1-12)
| | | | _____ Dia do mês (1-31)
| | | _____ Hora (0-23)
| | _____ Minuto (0-59)
```



Agendamento de tarefas

Valores das colunas do cron

Minuto

Varia de 00 à 59

Hora

Varia de 00 à 23

Dia

Varia de 01 à 31

OBS: Lembrando sempre que ao executar uma tarefa nos dias acima de 28, essa tarefa não será executada no mês de fevereiro!

Mês

Varia de 01 à 12



Agendamento de tarefas

Valores das colunas do cron

Semana

Varia de 0 à 6 ou de 1 à 7

Fica assim:

0 à 6 significa que é de Domingo a Sábado

1 à 7 significa que é de Segunda a Domingo

Ou seja, tanto o valor 0 quanto o valor 7 é Domingo!

Tarefa

É a tarefa que vai ser executada.



Agendamento de tarefas

Os coringas do crontab

Coringa “*”

Para ajudar no agendamento de tarefas podemos usar os coringas, que são caracteres especiais determinados no agendamento de uma ou mais tarefas.

Por exemplo: devemos ter uma tarefa que desligue o computador todos os dias às 22 horas. Ao invés de digitar todos os dias do mês, todos os meses e todos dias da semana, fazemos isso **usando um único caractere**, que indique todos em cada campo específico; no nosso caso, seria o caracter “*”.



Agendamento de tarefas

Os coringas do crontab

Coringa “*”

Vamos executar uma tarefa (**echo “Estou testando o cron” > /dev/tty1**), enviando a saída desse comando para o **terminal texto 1** em cada hora, ou seja às 13:10, às 14:10, e assim por diante:

```
10 * * * * echo “Estou testando o cron” > /dev/tty1
```

O coringa “/”

Com esse coringa, podemos executar por exemplo uma tarefa de 5 em 5 minutos:

```
*/5 * * * * echo “Estou testando o cron” > /dev/tty1
```



Agendamento de tarefas

Os coringas do crontab

O coringa "-"

Aqui, podemos executar uma tarefa dentro de um certo intervalo, por exemplo de 0 à 5 minutos:

```
0-5 * * * * echo "Estou testando o cron" > /dev/tty1
```

O coringa ",",

Esse é um coringa onde podemos executar uma tarefa em tempos intercalados!

Por exemplo, no minuto 1, 3 e 5:

```
1,3,5 * * * * echo "Estou testando o cron" > /dev/tty1
```



Agendamento de tarefas

O que fizemos agora foi colocar na agenda do cron, para que, toda sexta, às 18 horas, seja feito o **backup** do diretório **/etc**, usando o **tar** com o **gzip**.

Feito isso, podemos salvar o arquivo e sair e aguardar o horário para que o computador execute a tarefa no horário que pedimos.

Vale lembrar que a nossa agenda sempre vai executar a tarefa que pedimos se ela obedecer a uma condição.

Por exemplo:

Quero que seja mostrado na tela às **17 horas** a seguinte mensagem: **“Hoje é Sex Jul 29 17:00:00 BRT 2005”**; ou seja, mostre a saída do comando **date**; porém, o dia tem que ser sexta e tem que ser no **mês 07**, no **dia 29** e tem que ser **17 horas**!



Agendamento de tarefas

A tarefa anterior ficaria assim:

```
0 17 29 7 5      echo "Hoje é `date`"
```

Isso quer dizer que, às 17 horas, do dia 29/07, será mostrada a mensagem **"Hoje é Sex Jul 29 17:00:01 BRT 2005"**.

No caso, a nossa mensagem só vai ser exibida no computador novamente daqui a seis anos quando for de novo 29/07, sexta-feira e for 17 horas!



Agendamento de tarefas

Consultando as tarefas

O **cron** deve consultar o que foi escrito no **crontab** para ver se o que pedimos vai ser executado pelo shell. Para ver então essa agenda, usamos o comando:

```
# crontab -l
```

Onde o “-l” é de **list**. Ou seja, me mostre o que tem na agenda. Todas as tarefas agendadas são localizadas dentro do diretório **/var/spool/cron/crontabs** no sistema.

E aí podemos ver se o que foi agendado está no horário certo; se tudo estiver ok, é só aguardar o cron ordenar ao shell que execute o que pedimos, ou seja, o que está dentro da agenda!



Agendamento de tarefas

Apagando as tarefas

Para **deletar** todas as tarefas agendadas no crontab, temos o seguinte comando:

```
# crontab -r
```

Onde a opção “-r” é para **remove**, ou seja, apagar a agenda do cron.

Agora, se você quiser simplesmente remover apenas uma tarefa, basta abrir novamente a agenda com o comando **crontab -e** e apagar a tarefa que estava agendada lá!



Agendamento de tarefas

O arquivo **/etc/crontab**

O **root**, como administrador do sistema, tem a agenda individual dele, onde pode guardar os seus compromissos e tem a agenda global, que é onde ele anota os compromissos que serão executados por um determinado usuário!

Ou seja, o compromisso que o root anotar na agenda global, vai ser executado por um usuário que pode ser ele mesmo ou um usuário mortal! Essa agenda global, fica no **/etc/crontab**.

Diferente da agenda individual, essa agenda tem 7 campos ao invés de 6! E o campo a mais que temos nessa agenda, é o campo **user**.

Essa agenda global tem a seguinte estrutura:

m h dom mon dow user command



Agendamento de tarefas

O arquivo `/etc/crontab`

Bom, nessa estrutura temos:

m = minuto (0-59)

h = hora (0-23)

dom = indica dia do mês (day of month)

mon = aqui está indicando mês (month)

dow = indica dia da semana (day of week)

user = qual usuário será o dono dessa tarefa

command = tarefa que será executada



Agendamento de tarefas

O agendador at

Vamos conhecer agora outro agendador do sistema, o **at**. Ele também tem a mesma função que o cron, ou seja, ele cuida para que o shell execute as tarefas que foram agendadas, mas ele é bem diferente do cron.

Para se ter uma idéia, nem uma agenda específica (arquivo de configuração) ele possui! Ele guarda as tarefas que devem ser executadas pelo shell no diretório **/var/spool/cron/atjobs**.

Só que o at só vai lembrar o shell de executar a tarefa apenas uma vez! É aí que ele é diferente do cron, porque na agenda do cron, o que estiver marcado lá, ele sempre vai executar! Já o at não, executou uma vez, não executa de novo!

Não deixando o at de lado, ele também tem sua importância! **Ele será útil quando quisermos que uma tarefa seja feita apenas uma única vez!**



Agendamento de tarefas

O agendador at

Por exemplo:

```
# echo "O at não e et" > at.txt | at 02pm today
```

Repare que, apesar de não exigir que o shell refaça a tarefa dentro do que foi especificado como o cron, o at é detalhista na questão de horários.

Acima, no exemplo, estamos pedindo para que hoje, às **14 horas**, mostre a mensagem "**O at não é et**" e jogue essa mensagem no arquivo **at.txt** dentro do **/tmp**. E, para que não aconteça erros, eu tenho que falar para o at, que é 2 horas da tarde (por isso o **pm** do lado do 02) de **hoje (today)**.



Agendamento de tarefas

O agendador at

Vendo a tarefa agendada

O at, assim como o cron, também mostra a lista de “coisas que o funcionário shell deve fazer”. E essa lista é mostrada com o comando:

```
# atq
```

Sendo que nessa lista que o at mostra, cada tarefa tem o seu número, e que não deve ser confundido com o PID! O número que aparecer na lista é o próprio at que fornece.

Agora, se você errar quando agendar a tarefa não tem como editar o arquivo, tem que remover a tarefa e adicionar novamente!

Para isso, basta pegar o número da tarefa, e usar o comando:

```
# atrm número_da_tarefa
```



Bibliografia

Linux – Guia do Administrador do Sistema

Autor: Rubem E. Pereira

Editora: Novatec

Manual Completo do Linux (Guia do Administrador)

Autor: Evi Nemeth, Garth Snyder, Trent R. Hein

Editora: Pearson Books

Guia Foca GNU/Linux

<http://focalinux.cipsga.org.br/>

