Introdução

Suponha que você seja dono de um serviço de hospedagem de sites e diariamente precisa disponibilizar, numa pasta acessível por FTP, o arquivo de log de acessos de um cliente. É trabalhoso fazer isso manualmente e na pior das hipóteses, você pode esquecer. Felizmente, o Linux conta com o serviço *cron*, que permite a realização de tarefas programadas em dias e horários determinados por você. O objetivo deste artigo é mostrar como fazer isso.

O que é cron?

O cron pode ser interpretado como um serviço do Linux que é carregado durante o processo de boot do sistema. Trata-se de uma ferramenta que permite programar a execução de comandos e processos de maneira repetitiva ou apenas uma única vez.

Pode-se utilizar o cron para uma infinidade de funções, por exemplo para que o arquivo de log de um cliente fosse disponibilizado todos os dias às 15 horas, por exemplo. Isso é possível porque o daemon (uma espécie de programa que executa tarefas em segundo plano) do cron executa os comandos nas datas e horários especificados. A cada minuto, o cron é acionado para verificar se há tarefas a serem realizadas.

Para executar as tarefas, o cron usa uma espécie de tabela conhecida como *crontab*. O arquivo crontab geralmente fica localizado no diretório /etc, mas também pode estar em um diretório que cria um crontab para cada usuário do sistema (geralmente em /var/spool/cron/), tudo depende das configurações do sistema operacional utilizado.

Como usar o cron

O primeiro passo é abrir o crontab. Para isso, você pode usar editores de textos como vi, emacs ou nano. Também é possível digitar o comando *crontab -e* para editar o arquivo exclusivo de seu usuário. Neste caso, a edição é feita como se você estivesse usando o [vi](http://www.infowester.com/linuxvi.php).

O crontab tem o seguinte formato:

**[minutos] [horas] [dias do mês] [mês] [dias da semana] [usuário] [comando]**

O preenchimento de cada campo é feito da seguinte maneira:

**- Minutos:** informe números de 0 a 59;

**- Horas:** informe números de 0 a 23;

**- Dias do mês:** informe números de 0 a 31;

**- Mês:** informe números de 1 a 12;

**- Dias da semana:** informe números de 0 a 7;

**- Usuário:** é o usuário que vai executar o comando (não é necessário especificá-lo se o arquivo do próprio usuário for usado);

**- Comando:** a tarefa que deve ser executada.

Repare que a ordem desses valores indica o nome correspondente do campo. Por exemplo, no campo mês, 1 a 12 quer dizer de "janeiro a dezembro". No caso de dias da semana, 0 a 6 quer dizer de "domingo a sábado". Note que o número 7 também pode ser usado. Neste caso, assim como o número 0, o 7 equivale ao dia de "domingo".

No lugar desses valores, você pode informar \* (asterisco) para especificar uma execução constante. Por exemplo, se o campo dias do mês conter \*, o comando relacionado será executado todos os dias.

Você também pode informar intervalos no preenchimento, separando os números de início e fim através de - (hífen). Por exemplo, se no campo horas for informando 2-5, o comando relacionado será executado às 2, 3, 4 e 5 horas. E se o comando tiver que ser executado às 2 horas, entre 15 e 18 horas e às 22 horas? Basta informar 2,15-18,22. Nestes casos, você separa os parâmetros por vírgula.

Vamos a um exemplo:

**#tarefa infowester   
30 22 3,14 \* \* echo "Não entre em pânico" > /home/alecrim/infowester.txt**

Neste exemplo, a frase "Não entre em pânico" é inserida no arquivo infowester.txt, dentro do diretório /home/alecrim/, às 22 horas e 30 minutos, nos dias 3 e 14, em todos os meses e em todos os dias da semana. Repare na linha "#tarefa infowester". Trata-se de um comentário. Digite # e tudo o que for digitado na linha não será considerado pelo cron. É um recurso útil para inserir descrições quando se tem várias tarefas a serem executadas.

Comandos do crontab

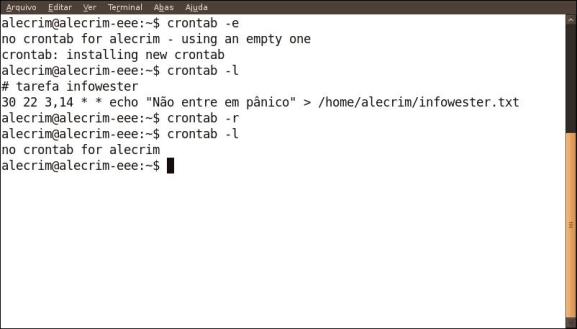
Para acessar o crontab, basta digitar esse nome em um terminal seguido de um parâmetro. Eis a lista de parâmetros disponíveis:

**crontab -e:** conforme já informado, serve para editar o arquivo atual do crontab e criar um, caso não exista;

**crontab -l:** este comando mostra o conteúdo atual do crontab;

**crontab -r:** remove o arquivo atual do crontab.

Como exemplo, analise a imagem abaixo:



Ela mostra que o comando *crontab -e* foi dado para edição (que ocorreu em outro programa). Como não havia nenhum arquivo de crontab para isso, o comando serviu para criar um. Quando esse procedimento foi finalizado, foi dado o comando *crontab -l* para a visualização de seu conteúdo. Em seguida, foi dado o comando *crontab -r* para eliminar esse arquivo. Por fim, foi dado o comando *crontab -l* novamente para comprovar a eliminação.

Finalizando

Os recursos do cron não se limitam aos mostrados anteriormente. Você pode ainda agendar tarefas mais complexas e poupar muito trabalho nessas execuções automáticas. É possível, por exemplo, configurar o cron para emitir-lhe um aviso de quando uma tarefa for executada. A dica para lidar com o cron é testá-lo. Experimente-o bastante. Mesmo que você seja apenas um usuário doméstico, certamente encontrará alguma utilidade para o cron.