

Experiência 02 – *cron* e *crontab*
O seguro morreu de velho. O desconfiado ainda vive.

O primeiro passo é abrir o **crontab**. Para isso, você pode usar editores de textos ou o comando **crontab -e** para editar o arquivo exclusivo de seu usuário. Neste último caso, a edição é feita como se você estivesse usando o vi.

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:

1. **Minutos:** informe números de 0 a 59;
2. **Horas:** informe números de 0 a 23;
3. **Dias do mês:** informe números de 1 a 31;
4. **Mês:** informe números de 1 a 12;
5. **Dias da semana:** informe números de 0 a 7;
6. **Usuário:** é o usuário que vai executar o comando (não é necessário especificá-lo se o arquivo do próprio usuário for usado);
7. **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 informado 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.

Exemplo:

```
# tarefa 1  
00 19 2,10 * * echo "Vamos estudar"
```

Neste exemplo, a frase "Porque nós amamos a liberdade" é exibida às 19 horas e 00 minutos, nos dias 2 e 10, em todos os meses e em todos os dias da semana.

Tarefa 01: faça um script que imprima uma mensagem qualquer na tela a cada 5 minutos.

Tarefa 02: Uma das principais preocupações de pequenas empresas é terem o seu site invadido, pois este site é a vitrine de seus produtos. Porém pequenas empresas não possuem recursos financeiros para manter uma equipe de segurança digital. Sua tarefa é desenvolver um script que faça a comparação de um diretório de arquivos no Linux com um outro diretório modelo. Caso haja diferença entre os diretórios um e-mail deverá ser enviado para o responsável pelo site (você, neste caso) para informar em quais arquivos as mudanças ocorreram. Dessa forma será possível resolver o problema no menor tempo possível. Para que o script seja executado sem a intervenção do administrador do sistema, use o cron para que ele seja executado periodicamente.

Tarefa 03: Incremente a tarefa 02 para que também seja efetuado o reparo dos arquivos que foram modificados.