

## Exercícios – Objetivo 107.2

1. O comando `date +%d/%m/%Y %T.%N` mostra a data e hora do sistema com precisão de nanossegundos. Use a cron para executar esse comando uma vez por minuto, jogando a saída em um arquivo. Depois de alguns minutos, observe a saída e veja quanto atraso ocorre nas execuções. **ATENÇÃO:** é necessário escapar os sinais de % com contrabarra (\%), pois eles são interpretados de maneira especial na crontab. Para detalhes, consulte o manual (`man 5 crontab`).
2. Gere uma linha para a crontab de sistema para executar o comando `ping -c 3 www.udesc.br` a cada 5 minutos, registrando o resultado no arquivo `/var/log/ping.log`. O arquivo deve acumular a saída de todas as execuções, não apenas da execução mais recente.
3. O site <http://crontab-generator.org/> oferece uma interface que gera linhas de crontab de acordo com critérios especificados pelo usuário.
  - (a) Use o site para gerar uma linha de crontab equivalente à solicitada no exercício 2.
  - (b) Que diferenças existem entre a linha que você construiu e a linha gerada pelo site? É possível dizer se o site gera linhas para crontabs de sistema ou de usuário?
4. O que acontece se você adicionar uma linha com erro a uma crontab? Por exemplo, edite a crontab do usuário `torvalds` e tente acrescentar a linha abaixo:

```
0 8-18/2 * * echo Teste
```

5. O comando abaixo exibe um lembrete na interface gráfica:

```
$ notify-send -u critical "Backup" "Não esquecer do backup semanal"
```

Crie uma linha de crontab para executar o comando abaixo todas as sextas-feiras às 16h, desde que o seu usuário (`torvalds`, por exemplo) esteja logado na interface gráfica.

DICA: o comando `who` lista os usuários logados no sistema; o usuário logado no console gráfico aparece com a indicação `(:0)`.

6. O que acontece se um usuário estiver listado tanto no arquivo `/etc/cron.allow` quanto no arquivo `/etc/cron.deny`? Ele será autorizado a editar sua crontab ou não? (Verifique isso na prática, em vez de procurar na documentação, Google, StackOverflow, ...)
7. Remova sucessivamente os arquivos `cron.allow` e `cron.deny`, verificando a cada passo como isso afeta usuários presentes e ausentes nesses arquivos.
8. O comando `at now + 1 min` irá executar os comandos **daqui** a 1 minuto ou **em até** 1 minuto?
9. Se você usar `at 12:00`, os comandos serão executados ao meio-dia ou à meia-noite?
10. Repita o exercício 6 com os arquivos `/etc/at.allow` e `at.deny`. Depois, remova sucessivamente os arquivos `at.allow` e `at.deny`, e veja como isso afeta usuários presentes e ausentes nesses arquivos.
11. Considere a seguinte linha em uma crontab:

```
0 0 13 * 5 cmd
```

O comando `cmd` será executado à meia-noite de toda sexta-feira 13 ou à meia-noite de toda sexta-feira e de todo dia 13?