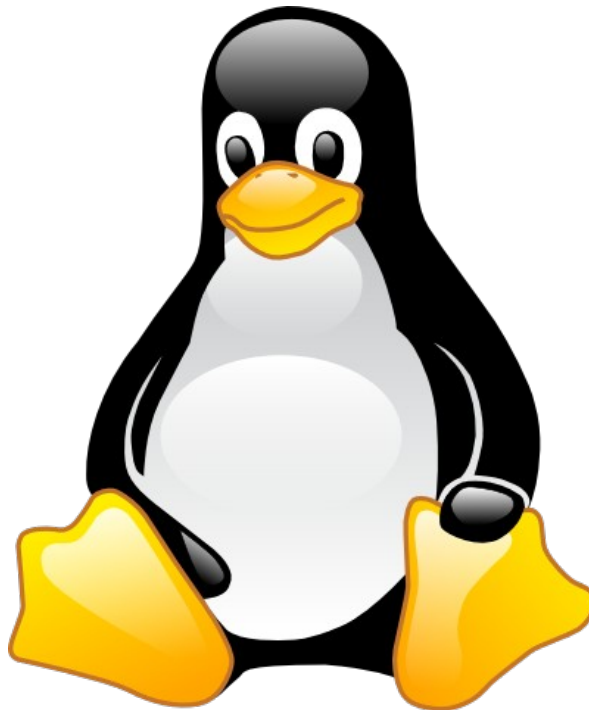


Linux System Administration 455



Aula 9 - 455



Aula 9 - 455

Função de cada um:

tar - só empacota, não reduz tamanho, apenas junta todos os arquivos em um.

gzip/bzip2 - compactador, ou seja, compacta um arquivo reduzindo seu tamanho.

Aula 9 - 455

Seria assim:

gzip - Rápido, porém não muito eficiente na redução de tamanho.

bzip2 - Lento, porém muito eficiente na redução de tamanho.

Aula 9 - 455

Vamos entrar no diretório /tmp para mostrar como funciona:

```
# cd /tmp
```

Aqui, estou colocando um script para criar um arquivo grande:

```
# while true  
> do  
> cat /etc/passwd >> /tmp/arquivo.txt  
> done
```

Deixem esse script executando por uns 30 segundos!

Onde os ">" não precisam ser digitados, pois vai aparecer (> é o prompt secundário).

Para parar basta dar CTRL+C para interromper a execução do script.

Aula 9 - 455

Vamos ver o tamanho do arquivo:

```
# cd /tmp
```

```
# du -h arquivo.txt
```

E aproveita e faz uma cópia do mesmo:

```
# cp arquivo.txt arquivo2.txt
```

Para testar os compactadores, vamos fazer assim:

arquivo.txt - Para o gzip

arquivo2.txt - Para o bzip2

Aula 9 - 455

Então vamos ao exemplo:

```
# cd /tmp (Só para certificar!!! rs)
```

```
# time gzip arquivo.txt
```

```
# time bzip2 arquivo2.txt
```

O comando time é apenas para marcar o tempo que o comando irá usar para terminar!

Aula 9 - 455

Então vou ter dois arquivos no /tmp, um com a extensão gz e outro com bz2:

```
# ls arquivo*  
arquivo.txt.gz  
arquivo2.txt.bz2
```

Agora comparem o tamanho dos dois arquivos.

Qual ficou menor?

```
#du -h arquivo*  
68K arquivo.txt.gz  
24K arquivo2.txt.bz2
```


Aula 9 - 455

Então vamos ao exemplo:

```
# cd /tmp
```

```
# tar -cvf bkp_etc.tar /etc
```

Onde:

-c - Para criar um backup

-v - (verbose) Para mostrar detalhes para você na hora de criar

-f - Para indicar o nome do arquivo. Essa opção sempre vem por último, pois é ela quem define o nome do arquivo.

Então podemos fazer o teste de tamanho e verão que não reduziu tamanho:

```
# du -h /etc
```

```
13M /etc
```

```
# du -h bkp_etc.tar
```

```
13M /etc
```

Aula 9 - 455

Para dizer para o tar que irei executar ele com mais um compactador tenho que acrescentar uma opção ao comando que fizemos:

z- Para compactar com GZIP

j - Para compactar com BZIP2

Então o comando mudaria para:

Tar com Gzip:

```
# cd /tmp # tar -cvzf bkp_etc.tar.gz /etc
```

Onde:

c - Para criar um backup

v - Para mostrar detalhes para você na hora de criar

z - Para compactar com o GZIP

f- Para indicar o nome do arquivo de backup

Aula 9 - 455

Tar com Bzip2:

```
# cd /tmp
```

```
# tar -cvjf bkp_etc.tar.bz2 /etc
```

Onde:

c - Para criar um backup;

v - Para mostrar detalhes para você na hora de criar ;

j - Para compactar com o BZIP2;

f - Para indicar o nome do arquivo de backup.

Aula 9 - 455

Com esses 3 comandos que eu passei, terei 3 arquivos dentro do meu /tmp!

```
# ls /tmp | grep bkp  
bkp_etc.tar bkp_etc.tar.gz bkp_etc.tar.bz2
```

Podemos comparar seus tamanhos:

```
# du -h bkp*  
9.3M bkp_etc.tar  
2.1M bkp_etc.tar.bz2  
2.5M bkp_etc.tar.gz
```

Aula 9 - 455

Então, tenho que testar a integridade desses arquivos, e, para isso usamos a opção t:

```
# tar -tvf bkp_etc.tar
```

```
# tar -tvzf bkp_etc.tar.gz
```

```
# tar -tvjf bkp_etc.tar.bz2
```

No qual t é a opção que testa! Depois, para descompactar esses arquivos só mudamos para a opção para x que é de extract!

```
# tar -xvf bkp_etc.tar
```

```
# tar -xvzf bkp_etc.tar.gz
```

```
# tar -xvjf bkp_etc.tar.bz2
```

Aula 9 - 455

Caso queira descompactar no / para substituir o /etc antigo no caso de um backup temos que especificar isso com o -C:

```
# tar -xvf bkp_etc.tar -C /
```

```
# tar -xvzf bkp_etc.tar.gz -C /
```

```
# tar -xvjf bkp_etc.tar.bz2 -C /
```

Aula 9 - 455

Agende para 10 minutos no futuro a listagem do diretório /etc/ redirecionando a saída para o terminal 11 (modo texto) e em seguida imprima no mesmo terminal a data e a hora:

Por exemplo: vamos supor que estou em outubro (10), dia 28, ano 2008, e a hora é 15:00 e quero que a tarefa seja executada às 20:00, então faço:

```
# at 20:00 10/28/2008
```

```
at > ls -color /etc/ > /dev/tty11
```

```
at > echo $(date +"%H:%m %M/%d/%Y") >> /dev/tty11
```

```
at > ^d
```

^d = CTRL + D

Aula 9 - 455

Agendada essa tarefa, confirme-a listando todos os agendamentos pendentes:

```
# atq
```

Explore o diretório onde ficam os agendamentos:

```
# cd /var/spool/cron/atjobs
```

```
# ls -la
```


Aula 9 - 455

Vamos realizar outro agendamento, para executar em 3 minutos, para que possamos aprender como apagá-lo:

```
# at HH:mm MM/DD/YYYY
```

```
at > echo "Teste" > /tmp/at.out
```

```
at > ^d
```

Liste os agendamentos correntes e verifique que um novo arquivo foi criado no diretório de spool do at:

```
# atq
```

```
4 Tue Feb 5 14:30:00 2008 a root
```

Aula 9 - 455

Remova o agendamento:

```
# atrm 4
```

Como eu posso verificar se ele foi realmente removido?

```
# atq
```

Aula 9 - 455

```
# dpkg -l | grep cron
```

```
ii cron 3.0pl1-86 management of regular background processing
```

Aula 9 - 455

crontab -e (Para todos os usuários que queiram usar o cron)

Onde -e significa edit.

Nesse arquivo em branco, temos que adicionar uma nova tarefa seguindo essa estrutura:

* * * * * comando/tarefa

Aula 9 - 455

Coluna Função

1 Minuto

2 Hora

3 Dia do mês

4 Mês

5 Dia da semana

6 Programa que será executado

Aula 9 - 455

Onde:

Coluna Função

Minuto -> 0-59

Hora -> 0-23

Dia do mês-> 1-31

Mês -> 1-12

Dia da semana -> 0-7 (o "0" e "7" é domingo, "1" segunda)

Aula 9 - 455

Exemplo:

```
#crontab -e
```

```
00 23 * * * halt
```

Aula 9 - 455

```
#crontab -e
```

```
00 23 * * * halt
```

No comando acima, estou falando para desligar a máquina, todos os dias (3º Campo com *) , de todos os meses (4º Campo com *) sendo qualquer dia da semana (5º Campo com *) as 23h00.

Aula 9 - 455

Por exemplo:

```
00 12 25 12 0 echo "Hoje é Domingo" >> /dev/tty2
```

Aula 9 - 455

Para ver as tarefas que foram agendadas pelos usuários (EU ROOT) digito:

```
# crontab -l -u root
```

```
00 22 * * * halt
```

Onde:

-l Para listar

-u Para especificar o usuário

Aula 9 - 455

Quando mando salvar a regra, ou a tarefa agendada, a mesma irá para um arquivo com o nome do meu usuário:

```
# ls /var/spool/cron/crontabs/root
```

Aula 9 - 455

```
# crontab -e
```

```
00 21 * * 1-5 /root/backup.sh
```

Aula 9 - 455

Vamos supor que você queira um backup que seja feito de 2 em 2 minutos:

```
*/2 * * * * /root/backup.sh
```

Entendendo os operadores do formato de agendamento:

vírgula (,) especifica uma lista de valores, por exemplo: “1,3,4,7,8”;

– especifica um intervalo de valores, por exemplo: 1-15 (de 1 a 15);

* especifica todos os valores possíveis;

/ especifica pulos de valores, por exemplo, se no campo hora utilizarmos */3 o comando será executado às 0,3,6,9,12,15,18,21..

Aula 9 - 455

Os arquivos de crontab dos usuários ficam em:

```
# cd /var/spool/cron/crontabs
```

Aula 9 - 455

Já a crontab do sistema é encontrada no `/etc/crontab` e já possui agendamentos para realizar as tarefas que se encontram nos diretórios:

`/etc/cron.hourly`

`/etc/cron.daily`

`/etc/cron.weekly`

`/etc/cron.monthly`

Aula 9 - 455

A sintaxe de agendamentos de sistema é quase idêntico ao do usuário, à exceção que a crontab do sistema possui um campo USUARIO a mais:

minuto hora dia mês diadasemana USUARIO comando

Repare também que dentro do arquivo /etc/crontab existem 4 agendamentos já definidos: cron.hourly, cron.daily, cron.weekly e cron.monthly.

Aula 9 - 455

Para que o script seja executado diariamente:

```
# cp /root/backup.sh /etc/cron.daily/backups
```

Para que o script seja executado semanalmente:

```
# cp /root/backup.sh /etc/cron.weekly/backups
```

Depois de adicionar o script dentro dos diretórios será necessário reiniciar o daemon do cron (Debian):

```
# /etc/init.d/cron stop
```

```
# /etc/init.d/cron start
```

Leitura sugerida:

```
# man 5 crontab
```