

Nome:	Turma:
Professor:	Data:

## Prática 14 - Crontab e AT.

### 1. Objetivo

- Configurar e administrar o serviço de agendamento através do cron e at.

### 2. Recursos Necessários

- Equipamento com o sistema operacional Linux Mandriva Free 2006 instalado ou similar;

### 3. Requisitos Necessários

- Ter conhecimentos básicos do ambiente shell.



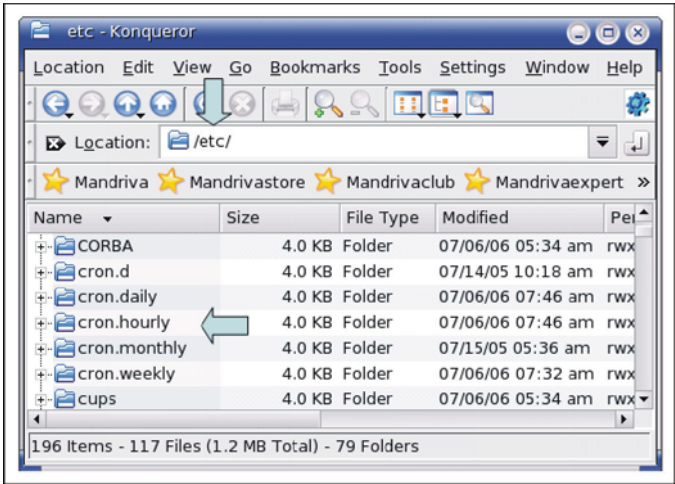
### 4. O que é o Cron?

O cron é um daemon padrão do *Unix* que foi introduzido no *Linux* que permite o agendamento de tarefas, permitindo através de um *script* a execução de um comando ou programa em um momento específico (dia/mês/ano/hora), ou em intervalos regulares realizados através do *crontab*, que é uma espécie de tabela. É um recurso muito usado em tarefas que envolvem arquivamento de *logs*, checagem e manutenção da integridade do sistema operacional, execução de programas/comandos em horários pré-determinados. Apresentamos aqui alguns exemplos para melhor entendimento do recurso e suas potencialidades:

- Checar a integridade dos sistemas de arquivos (e.g. todos os dias às 5:00 am e mandar *e-mail* do resultado para o usuário *root*);
- Verificar se há *e-mails* em suas contas pop (e.g. em intervalos de 15 minutos);

- Rodar o *updatedb* todos os dias (e.g. às 00:30 am);
- Procurar por arquivos core desnecessários e excluí-los (e.g. todas as quintas-feiras, às 3:00 am);
- E muito mais...

Existem quatro diretórios essenciais por default já prontos para serem usados, são eles:

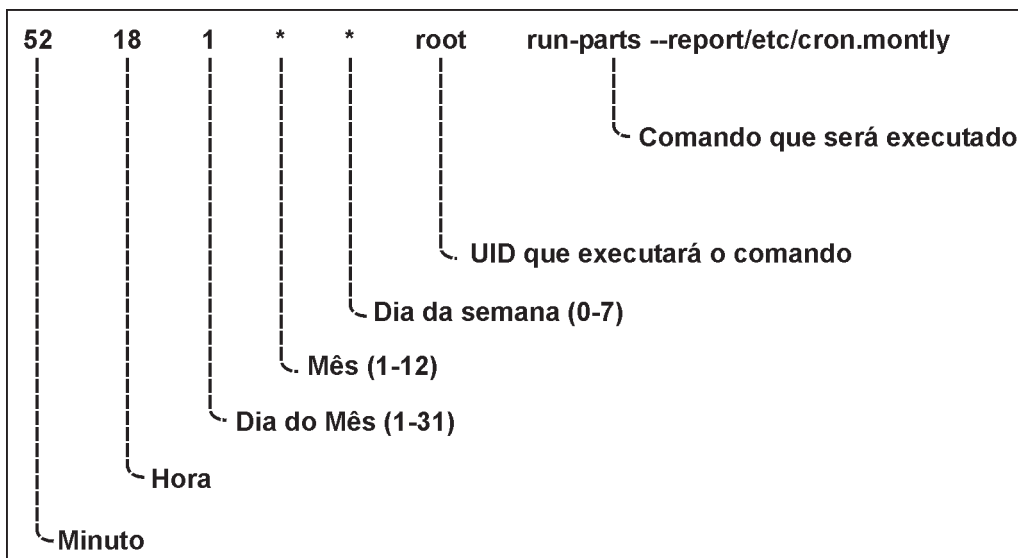
etc - Konqueror	Diretório	Periodicidade	Horário
	/etc/cron.hourly	A cada hora	No minuto 01
	/etc/cron.daily	Diariamente	04:20 am
	/etc/cron.weely	Semanalmente	Domingo às 04:20 am
	/etc/cron.monthly	Mensalmente	Dia 01 às 04:20 am

As tarefas são definidas através do arquivo */etc/crontab* e por arquivos individuais dos usuários em */var/spool/cron/crontabs/[usuário]*, criados através do programa *crontab*.

**Obs.:** A distribuição *Debian*, adicionalmente utiliza os arquivos no diretório */etc/cron.d* como uma extensão para o */etc/crontab*.

## 5. Formato de um arquivo *Crontab*

O arquivo */etc/crontab* tem o seguinte formato, apresentado no exemplo a seguir:



Campo	Valores Válidos Permitidos
Minuto	Valor entre 0 e 59
Hora	Valor entre 0 e 23
Dia do Mês	Valor entre 0 e 31
Mês	Valor entre 0 e 12, identificando os meses de Janeiro a Dezembro.
Dia da Semana	Valor entre 0 e 7, identificando os dias entre Domingo a Sábado. Note que tanto os números 0 e 7, equivalem aqui ao domingo.
Usuário	O usuário especificado será usado para executar o comando.
Comando	Comando que será executado, podendo ser usados os respectivos parâmetros utilizados na linha de comando.

**Obs.:** Podemos utilizar também além dos valores válidos, caracteres especiais, como segue abaixo:

Caractere	Exemplo	Significado
Asterisco ( * )	01 * * * *	De hora em hora, no minuto 01. O asterisco pode ser usado nos campos de data e hora para especificar todo o intervalo disponível.
Hífen ( - )	1-5	Intervalo de 1 a 5. O hífen serve para especificar períodos de execução (incluindo o número inicial/final).
Vírgula ( , )	1,3,5,7	Os números 1, 3, 5 e 7. A vírgula serve para especificar uma lista de números.
Barra ( / )	* / 5	De 5 em 5. Passos são especificados através de uma barra invertida.

## 6. Dominando a Especificação de Horários no Crontab

Especificar horários e frequência desejados no crontab não é uma experiência complexa. Basta conhecer algumas regras simples.

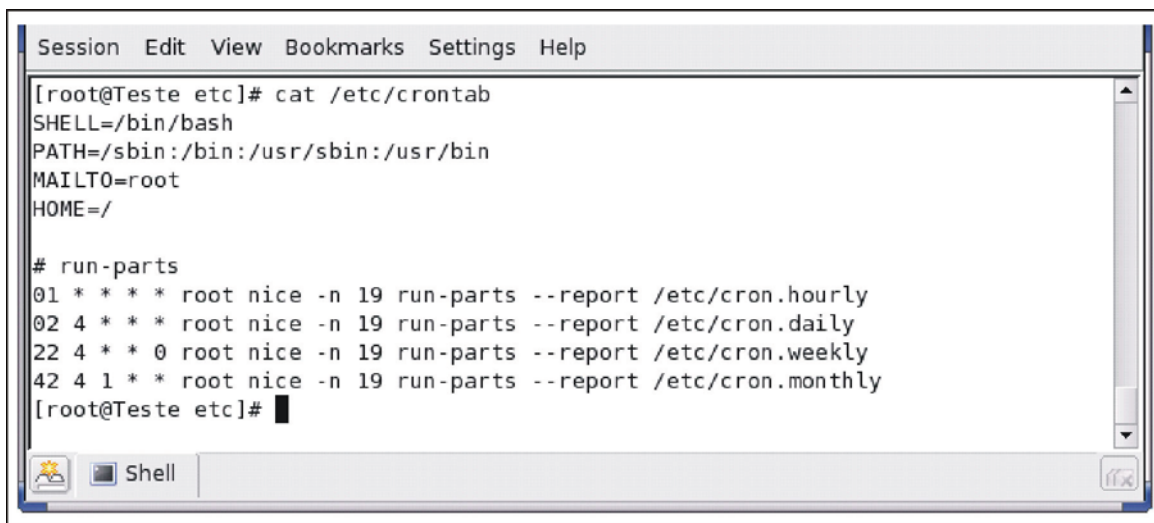
Horário					Significado
*	*	*	*	*	De minuto em minuto.
01	*	*	*	*	De hora em hora, no minuto 01.
00	18	*	*	*	Diariamente às 18 horas.
00	18	15			Mensalmente, no dia 15 às 18 horas.
00	18	*	*	5	Semanalmente, nas sextas-feiras às 18 horas.
00	18	*	*	1-5	De segunda a sexta-feira às 18 horas.
00	18	*	*	1,3,5	De segunda, quarta e quinta-feira às 18 horas.
00	18	*/3	*	*	De 3 em 3 dias, às 18 horas.
*/15	*	*	*	*	Diariamente de 15 em 15 minutos.
00	8-18	*	*	1-5	De hora em hora, no horário das 8 às 18 horas, de segunda a sexta-feira.
00	18	1-5,7,9	3	*	Às 18 horas dos dias 1, 2, 3, 4, 5, 7 e 9 do mês de Março.

Conhecendo agora a nomenclatura utilizada no crontab, explique o significado da instrução abaixo:

00	18	*	*	*	nemo rm -f /home/nemo/tmp/*
----	----	---	---	---	-----------------------------

## 7. Visualizando o conteúdo de um arquivo Crontab

- A.** Ao ligar a máquina, na tela de logon do KDM, logue-se no sistema como o usuário **root**.
- B.** Abra um novo terminal TTY no modo texto ( <Ctrl> + <ALT> + <F1> )  
**Obs.:** Para efeito de material didático, nós estaremos usando o ambiente gráfico que é similar.
- C.** Digite o comando **cat** como mostrado a seguir, aplicado ao arquivo “*crontab*”.

A terminal window titled "Session Edit View Bookmarks Settings Help" showing the output of the command `cat /etc/crontab`. The output lists system environment variables and four cron jobs for the `run-parts` directory. The prompt is `[root@Teste etc]#`.

```
[root@Teste etc]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root nice -n 19 run-parts --report /etc/cron.hourly
02 4 * * * root nice -n 19 run-parts --report /etc/cron.daily
22 4 * * 0 root nice -n 19 run-parts --report /etc/cron.weekly
42 4 1 * * root nice -n 19 run-parts --report /etc/cron.monthly
[root@Teste etc]#
```

Com base na resposta do sistema de seu computador, que informações foram obtidas e qual a sua importância:

---

---

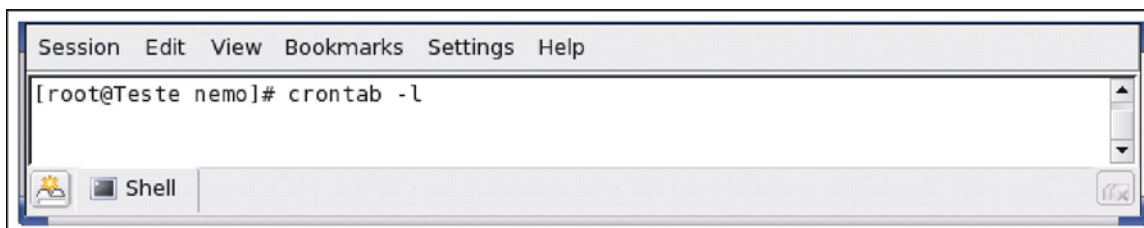
## 8. Comandos do Crontab

Para configurar um *crontab* seja o global (*root*) ou por usuário, nós utilizamos o comando ***crontab***, precedido de parâmetros específicos, que configuram uma determinada ação. A seguir uma relação:

Comando	Função
<code>crontab -e</code>	Edita o <i>crontab</i> atual do usuário.
<code>crontab -l</code>	Exibe o atual conteúdo do <i>crontab</i> do usuário.
<code>crontab -r</code>	Remove o <i>crontab</i> do usuário.

**Obs.:** Para maiores informações sobre o comando e parâmetros, consulte o comando ***man***.

**A.** Digite o comando abaixo para saber qual o conteúdo do *crontab* do seu usuário.

A terminal window titled "Session Edit View Bookmarks Settings Help" showing the command `crontab -l` being entered. The prompt is `[root@Teste nemo]#`.

```
[root@Teste nemo]# crontab -l
```

Com base na resposta do sistema de seu computador, o que você conclui:

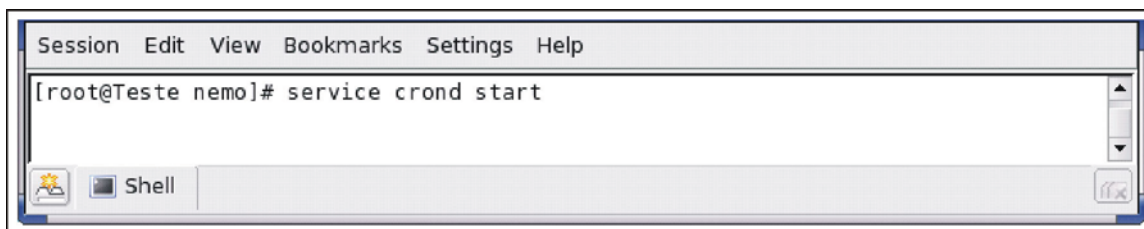
---

---

## 9. Iniciando o Serviço Crontab

O **crond** é um *daemon* do sistema cuja função é executar comandos e programas em datas e horários determinados especificadas no arquivo de configuração `/etc/crontab`.

Logo para que possamos agendar qualquer tarefa no sistema, é necessário que o serviço `crond` esteja carregado. Na maioria das distribuições, por *default* o serviço já está em execução, logo não será necessário reinicializá-lo. Entretanto, supondo que seja necessário iniciar o serviço, digite o comando abaixo:



```
Session Edit View Bookmarks Settings Help
[root@Teste nemo]# service crond start
```

The image shows a terminal window with a menu bar (Session, Edit, View, Bookmarks, Settings, Help) and a title bar (Shell). The command `[root@Teste nemo]# service crond start` has been entered and is being executed.

Com base na resposta do sistema de seu computador, o que você conclui:

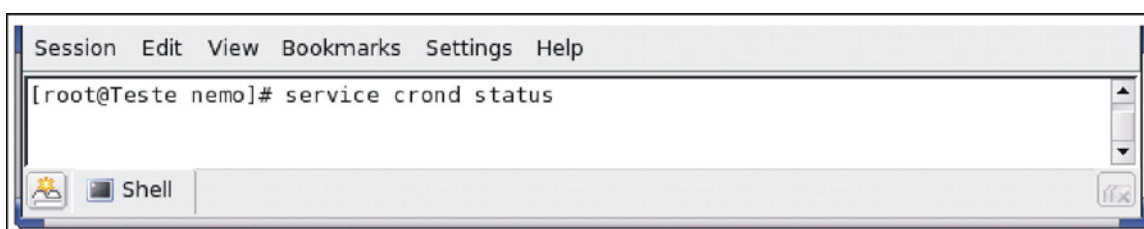
---

---

## 10. Agendando execução de um comando através do Crontab

Tendo iniciado o serviço, vamos agora gerar um pequeno exemplo de uma tarefa a ser agendada pelo *crontab*.

- A. Dentro do ambiente do *shell*, digite o comando abaixo para saber se o *crontab* esta funcionando ou não:



```
Session Edit View Bookmarks Settings Help
[root@Teste nemo]# service crond status
```

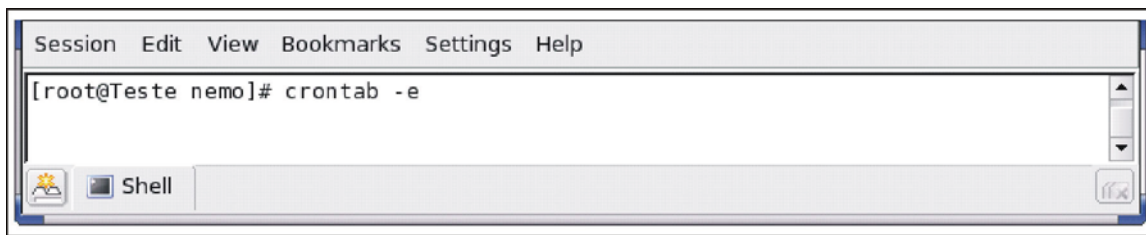
The image shows a terminal window with a menu bar (Session, Edit, View, Bookmarks, Settings, Help) and a title bar (Shell). The command `[root@Teste nemo]# service crond status` has been entered and is being executed.

Com base na resposta do sistema de seu computador, o que você conclui:

---

---

**B.** Dentro do ambiente do *shell*, digite o comando abaixo para editarmos o *crontab* do usuário *root*:



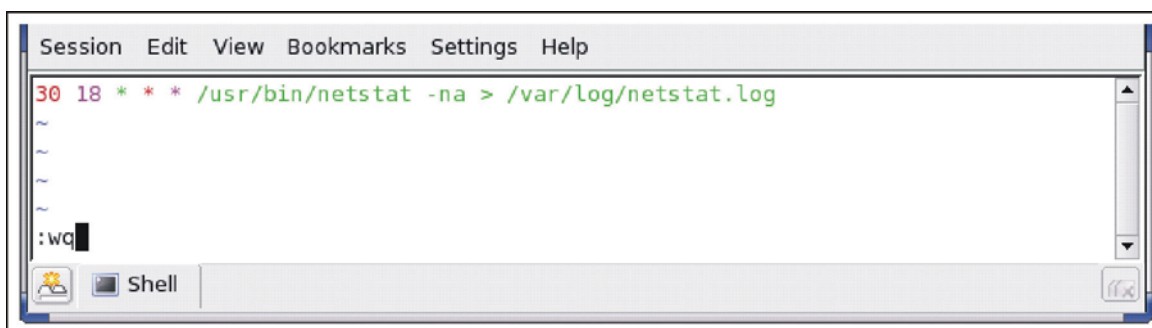
```
[root@Teste nemo]# crontab -e
```

Qual foi a resposta do sistema de seu computador e o que você conclui:

---

---

**C.** Dentro do ambiente do *vi*, digite o comando abaixo para editarmos o *crontab* do usuário *root* para agendarmos uma tarefa para rodar todos os dias às 18:30:



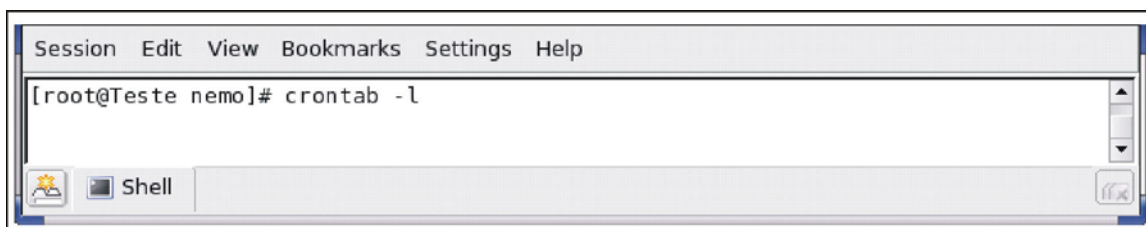
```
30 18 * * * /usr/bin/netstat -na > /var/log/netstat.log
```

Qual foi a resposta do sistema de seu computador após a conclusão:

---

---

**D.** Através do comando abaixo, observe se o *crontab* do usuário *root* foi salvo corretamente:



```
[root@Teste nemo]# crontab -l
```

Com base na resposta do sistema de seu computador, o que você conclui:

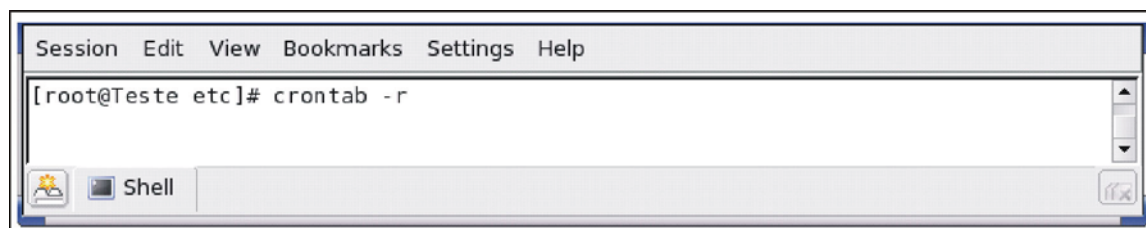
---

---

E. Sabendo que XX é seus próximos 5 minutos e YY é sua hora atual, escreva o *crontab* abaixo:

XX	YY	*	*	*	/bin/echo "Teste de crontab" > /tmp/teste.log
----	----	---	---	---	---

F. Dentro do ambiente do shell, digite o comando abaixo para removermos o crontab do usuário root:



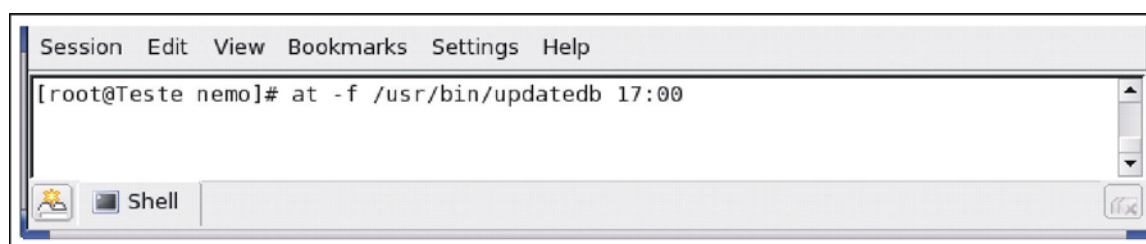
## 11. O que é o At?

O comando "at" permite agendar a execução de tarefas de forma semelhante ao cron com uma interface que permite a utilização de linguagem simples nos agendamentos. Uma das suas principais utilizações, é a execução de aplicativos que por força das circunstâncias, os mesmos não foram executados no horário apropriado, e.g. uma interrupção no fornecimento de energia.

## 12. Agendando Execução de Um Comando através do Comando at

Para agendar uma tarefa, basta usar a opção **-f** para especificar qual comando deve ser executado, seguido do horário desejado.

A. Dentro do ambiente do *shell*, digite o comando *at* com opção **-f** num pequeno exemplo de uma tarefa a ser agendada:



Qual foi a resposta do sistema de seu computador após a execução do comando:

---

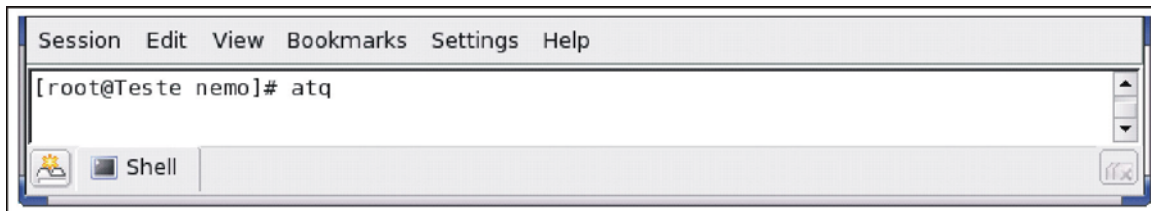
---

**Obs.:** O comando *updatedb* foi agendado para ser executado às 17:00 da presente data (hoje). Caso já tenha se passado o horário em questão, ele será automaticamente agendado para o mesmo horário no dia seguinte.



### 13. Fazendo o levantamento dos comandos agendados com o comando at

- A. Dentro do ambiente do *shell*, digite o comando abaixo para saber quais comandos estão agendados através do *at*:



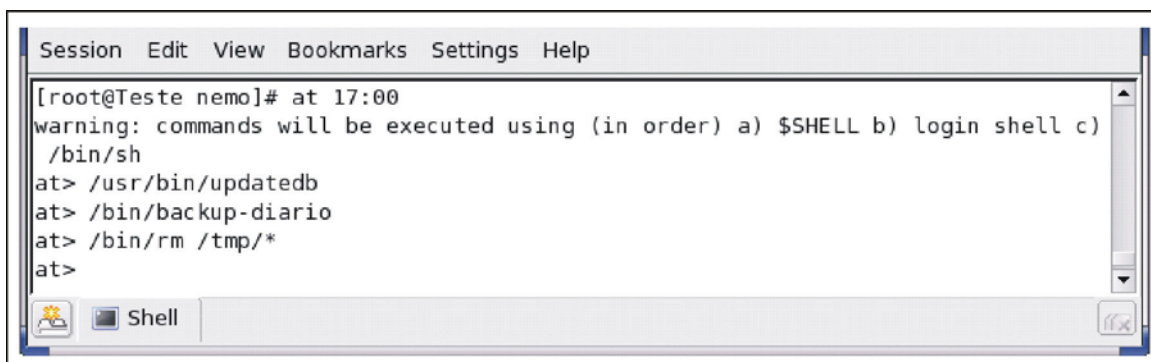
```
Session Edit View Bookmarks Settings Help
[root@Teste nemo]# atq
```

Qual foi a resposta do sistema de seu computador após a execução do comando:

**Obs.:** É possível agendar vários comandos para vários horários diferentes para um mesmo dia. Se houver mais de um comando para ser executado num mesmo horário, coloque-os através um arquivo, como um script shell executável.

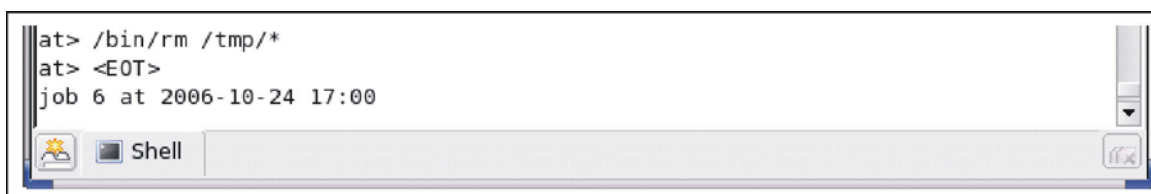
### 14. Gerando tarefas simultâneas num mesmo horário com o comando at

- A. Para agendar várias tarefas simultâneas de forma interativa, no ambiente do **shell**, digite o comando **at** seguido do horário sem a opção **-f** como segue:



```
Session Edit View Bookmarks Settings Help
[root@Teste nemo]# at 17:00
warning: commands will be executed using (in order) a) $SHELL b) login shell c)
/bin/sh
at> /usr/bin/updatedb
at> /bin/backup-diario
at> /bin/rm /tmp/*
at>
```

Para terminar a lista de comandos a serem executados, digite <Ctrl> <D>.

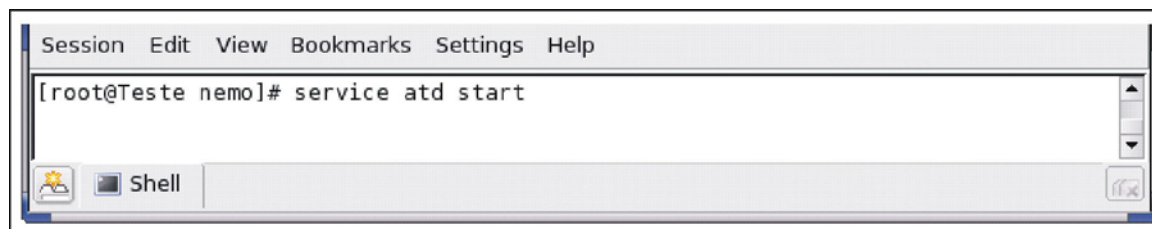


```
at> /bin/rm /tmp/*
at> <EOT>
job 6 at 2006-10-24 17:00
```

Será incluída a instrução <EOT> finalizando assim o processo.

## 15. Iniciando o Serviço at

O at, da mesma forma do cron, para que o agendamento funcione, é necessário que o serviço esteja ativo. Geralmente este já está em funcionamento por default, caso contrário é necessário carregá-lo.



## 16. Arquivos de Controle do Serviço at

É importante observar que há dois arquivos de controle, usados para permitir ou coibir o acesso ao comando at pelos usuários:

Arquivo	Função
/etc/at.allow	Lista os usuários que têm permissão para usar o comando at.
/etc/at.deny	Lista os usuários que não têm permissão para usar o comando at.

**Obs.:** O arquivo “*at.allow*” tem precedência sobre o “*at.deny*”, o que significa que se ambos existirem, o “*at.deny*” nunca será usado. Caso nenhum deles exista, somente o usuário “*root*” poderá utilizá-lo e se o “*at.deny*” existir e estiver vazio, todos os usuários terão permissão de utilizar o comando at.

## Conclusão

Quais são as vantagens de se agendar a execução de comandos e programas num Sistema Operacional?

---

---

---

---

Qual seria a interpretação do comando agendado: 30 4 1,15 \* 5

---

---

---

---



Se você pudesse optar por realizar um procedimento através do “crontab” ou o “at”, qual você escolheria e justifique?

---

---

---

---