

Guia Foca GNU/Linux

Capítulo 26 - Manutenção do Sistema

Este capítulo descreve como fazer a manutenção de seu sistema de arquivos e os programas de manutenção automática que são executados periodicamente pelo sistema.

26.1 Checagem dos sistemas de arquivos

A checagem do sistema de arquivos permite verificar se toda a estrutura para armazenamento de arquivos, diretórios, permissões, conectividade e superfície do disco estão funcionando corretamente. Caso algum problema exista, ele poderá ser corrigido com o uso da ferramenta de checagem apropriada. As ferramentas de checagem de sistemas de arquivos costumam ter seu nome iniciado por `fsck` e terminados com o nome do sistema de arquivos que verifica, separados por um ponto:

- `fsck.ext2` - Verifica o sistema de arquivos EXT2 ou EXT3. Pode também ser encontrado com o nome `e2fsck`.
- `fsck.ext3` - Um alias para `fsck.ext2`.
- `fsck.minix` - Verifica o sistema de arquivos Minix.
- `fsck.msdos` - Verifica o sistema de arquivos Msdos. Pode também ser encontrado com o nome `dosfsck`.

Para verificar um sistema de arquivos é necessário que ele esteja desmontado caso contrário poderá ocorrer danos em sua estrutura. Para verificar o sistema de arquivos raiz (que não pode ser desmontado enquanto o sistema estiver sendo executado) você precisará inicializar através de um disquete e executar `ofsdsk.ext2`.

26.1.1 `fsck.ext2`

Este utilitário permite verificar erros em sistemas de arquivos EXT2 e EXT3 (*Linux Native*).

```
fsck.ext2 [opções] [dispositivo]
```

Onde:

`dispositivo`

É o local que contém o sistema de arquivos EXT2/EXT3 que será verificado (partições, disquetes, arquivos).

`opções`

`-c`

Faz o `fsck.ext2` verificar se existem agrupamentos danificados na unidade de disco durante a checagem.

`-d`

Debug - Mostra detalhes de processamento do `fsck.ext2`.

`-f`

Força a checagem mesmo se o sistema de arquivos aparenta estar em bom estado. Por padrão, um sistema de arquivos que aparentar estar em bom estado não são verificados.

-F

Grava os dados do cache no disco antes de iniciar.

-l [arquivo]

Inclui os blocos listados no [arquivo] como blocos defeituosos no sistema de arquivos. O formato deste arquivo é o mesmo gerado pelo `programabadbblocks`.

-L [arquivo]

Faz o mesmo que a opção `-l`, só que a lista de blocos defeituosos do dispositivo é completamente limpa e depois a lista do [arquivo] é adicionada.

-n

Faz uma verificação de somente leitura no sistema de arquivos. Com esta opção é possível verificar o sistema de arquivos montado. Será assumido não para todas as perguntas e nenhuma modificação será feita no sistema de arquivos.

Caso a opção `-c` seja usada junto com `-n`, `-l` ou `-L`, o sistema de arquivos será verificado e permitirá somente a atualização dos setores danificados não alterando qualquer outra área.

-p

Corrige automaticamente o sistema de arquivos sem perguntar. É recomendável fazer isto manualmente para entender o que aconteceu, em caso de problemas com o sistema de arquivos.

-v

Ativa o modo verbose (mais mensagens são mostradas durante a execução do programa).

-y

Assume `sim` para todas as questões.

Caso sejam encontrados arquivos problemáticos e estes não possam ser recuperados, o `fsck.ext2` perguntará se deseja salva-los no diretório `lost+found`. Este diretório é encontrado em todas as partições `ext2`. Não há risco de usar o `fsck.ext3` em uma partição `EXT2`.

Após sua execução é mostrado detalhes sobre o sistema de arquivos verificado como quantidade de blocos livres/ocupados e taxa de fragmentação.

Exemplos: `fsck.ext2 /dev/hda2`, `fsck.ext2 -f /dev/hda2`, `fsck.ext2 -vrf /dev/hda1`.

26.2 reiserfsck

Verifica um sistema de arquivos `reiserfs` em sistema de arquivos.

`reiserfsck [opções] [dispositivo]`

dispositivo

Dispositivo que contém o sistema de arquivos `reiserfs` que será verificado.

opções

-a

Mostra detalhes sobre o sistema de arquivos e sai

-j arquivo

Especifica um arquivo de Journal alternativo usado pelo sistema de arquivos.

-q quiet

Não exibe mensagens sobre o status da checagem do sistema de arquivos.

-S

Constrói a árvore de todos os blocos do dispositivo.

O `reiserfsck` possui outros modos de operação além de checagem (o padrão), para detalhes veja a página de manual do programa.

Exemplos: `reiserfsck /dev/hda1`, `reiserfsck -S /tmp/arq-reiserfs`.

26.3 fsck.minix

Verifica o sistema de arquivos *minix* em um dispositivo.

`fsck.minix [opções] [dispositivo]`

Onde:

dispositivo

Partição, disquete ou arquivo que contém o sistema de arquivos *Minix* que será verificado

opções

-f

Verifica o sistema de arquivos mesmo se ele estiver perfeito.

-r

Permite reparo manual do sistema de arquivos

-a

Permite um reparo automático do sistema de arquivos. É recomendado fazer o reparo manual.

-v

Verbose - Mostra detalhes durante a execução do programa

-S

Exibe detalhes sobre os blocos de root.

Exemplo: `fsck.minix -f /dev/hda8, fsck.minix -vf /dev/hda8`

26.4 badblocks

Procura blocos defeituosos em um dispositivo. Note que este **apenas** pesquisa por blocos defeituosos, sem alterar a configuração do disco. Para marcar os blocos defeituosos para não serem mais usados, utilize a opção `-l` do `fsck` (veja [fsck.ext2, Seção 26.1.1](#)).

`badblocks [opções] [dispositivo]`

Onde:

`dispositivo`

Partição, disquete ou arquivo que contém o sistema de arquivos que será verificado.

`opções`

`-b [tamanho]`

Especifica o `[tamanho]` do bloco do dispositivo em bytes

`-o [arquivo]`

Gera uma lista dos blocos defeituosos do disco no `[arquivo]`. Este lista pode ser usada com o programa `fsck.ext2` junto com a opção `-l`.

`-s`

Mostra o número de blocos checados durante a execução do `badblocks`.

`-v`

Modo verbose - São mostrados mais detalhes.

`-w`

Usa o modo leitura/gravação. Usando esta opção o `badblocks` procura por blocos defeituosos gravando alguns padrões (0xaa, 0x55, 0xff, 0x00) em cada bloco do dispositivo e comparando seu conteúdo.

Nunca use a opção `-w` em um dispositivo que contém arquivos pois eles serão apagados!

Exemplo: `badblocks -s /dev/hda6, badblocks -s -o bad /dev/hda6`

26.5 defrag

Permite desfragmentar uma unidade de disco. A fragmentação é o armazenamento de arquivos em áreas não sequenciais (uma parte é armazenada no começo a outra no final, etc), isto diminui o desempenho da unidade de disco porque a leitura deverá ser interrompida e feita a movimentação da cabeça para outra região do disco

onde o arquivo continua, por este motivo discos fragmentados tendem a fazer um grande barulho na leitura e o desempenho menor.

A desfragmentação normalmente é desnecessária no GNU/Linux porque o sistema de arquivos *ext2* procura automaticamente o melhor local para armazenar o arquivo. Mesmo assim, é recomendável desfragmentar um sistema de arquivos assim que sua taxa de fragmentação subir acima de 10%. A taxa de fragmentação pode ser vista através do `fsck.ext2`. Após o `fsck.ext2` ser executado é mostrada a taxa de fragmentação seguida de *non-contiguos*.

A ferramenta de desfragmentação usada no GNU/Linux é o `defrag` que vem com os seguintes programas:

- `e2defrag` - Desfragmenta sistemas de arquivos *Ext2*.
- `defrag` - Desfragmenta sistemas de arquivos *Minix*.
- `xdefrag` - Desfragmenta sistemas de arquivos *Xia*.

O sistema de arquivos deve estar desmontado ao fazer a desfragmentação. Se quiser desfragmentar o sistema de arquivos raiz (/), você precisará inicializar através de um disquete e executar um dos programas de desfragmentação apropriado ao seu sistema de arquivos. A checagem individual de fragmentação em arquivos pode ser feita com o programa `frag`.

ATENÇÃO: Retire cópias de segurança de sua unidade antes de fazer a desfragmentação. Se por qualquer motivo o programa de desfragmentação não puder ser completado, você poderá perder dados!

```
e2defrag [opções] [dispositivo]
```

Onde:

dispositivo

Partição, arquivo, disquete que contém o sistema de arquivos que será desfragmentado.

-d

Debug - serão mostrados detalhes do funcionamento

-n

Não mostra o mapa do disco na desfragmentação. É útil quando você inicializa por disquetes e recebe a mensagem "Failed do open term Linux" ao tentar executar o `e2defrag`.

-r

Modo somente leitura. O `defrag` simulará sua execução no sistema de arquivos mas não fará nenhuma gravação. Esta opção permite que o `defrag` seja usado com sistema de arquivos montado.

-s

Cria um sumário da fragmentação do sistema de arquivos e performance do desfragmentador.

-v

Mostra detalhes durante a desfragmentação do sistema de arquivos. Caso mais de uma opção `-v` seja usada, o nível de detalhes será maior.

`-i [arquivo]`

Permite definir uma lista de prioridades em que um arquivo será gravado no disco, com isto é possível determinar se um arquivo será gravado no começo ou final da unidade de disco. Esta lista é lida do [arquivo] e deve conter uma lista de prioridades de -100 a 100 para cada inodo do sistema de arquivos. Arquivos com prioridade alta serão gravados no começo do disco.

Todos os inodos terão prioridade igual a zero caso a opção `-i` não seja usada ou o inodo não seja especificado no [arquivo]. O [arquivo] deverá conter uma série de linhas com um número (inodo) ou um número prefixado por um sinal de igual seguido da prioridade.

`-p [numero]`

Define o [numero] de buffers que serão usados pela ferramenta de desfragmentação na realocação de dados, quanto mais buffers mais eficiente será o processo de realocação. O número depende de quantidade memória RAM e Swap você possui. Por padrão 512 buffers são usados correspondendo a 512Kb de buffer (em um sistema de arquivos de blocos com 1Kb).

Exemplo: `e2defrag -n -v /dev/hdb4, e2defrag -r /dev/hda1`

26.6 Verificando e marcando setores danificados em um HD

Um dos sintomas de um disco rígido que contém setores danificados (bad blocks) é a mudança repentina do sistema de arquivos para o modo somente leitura, o aparecimento de diversas mensagens no syslog indicando falha de leitura do hd, uma pausa de segundos no sistema junto com o led de atividade de disco ligado. Se isto acontece com você, uma forma de solucionar este inconveniente é executar o teste na superfície física do disco para procurar e marcar os blocos problemáticos como defeituosos.

Em alguns casos, os blocos defeituosos ocorrem isoladamente no disco rígido, não aumentando mais sua quantidade, entretanto, se o número de blocos danificados em seu disco está crescendo em um curto espaço de tempo, comece a pensar na troca do disco rígido por um outro. Existem empresas que recuperam HDs mas pelo valor cobrado por se tratar de um serviço delicado, só compensa caso você não tenha o backup e **realmente** precisa dos dados do disco.

Para fazer uma checagem de HD no sistema de arquivos `ext2` ou `ext3`, proceda da seguinte forma:

- Se possível, faça um backup de todos os dados ou dos dados essenciais da partição será checada.
- Inicie o sistema por um disquete de boot ou CD de recuperação. Este passo é útil pois em alguns casos, pode ocorrer a perda de interrupção do disco rígido e seu sistema ficar paralisado. Só o método de checar o HD usando um disquete de boot lhe fará agendar uma parada no sistema e notificar os utilizadores, evitando sérios problemas do que fazendo isto com um sistema em produção.

- Execute o `badblocks` usando a opção `-o` para gravar os possíveis blocos defeituosos encontrados para um arquivo: `badblocks -v -o blocos-defeituosos.lista /dev/hd??`.
Substitua o dispositivo `/dev/hd??` pelo dispositivo que deseja verificar. A checagem do `badblocks` deverá ser feita para cada partição existente no disco rígido. O tempo de checagem dependerá da velocidade do disco rígido, velocidade do barramento, cabo de dados utilizado, velocidade de processamento e é claro, do estado do disco rígido (quantos setores defeituosos ele tem).
 - Após concluir o `badblocks`, veja se foram encontrados blocos defeituosos. Caso tenha encontrado, siga para o próximo passo.
 - Para marcar os blocos encontrados pelo `badblocks` como defeituosos, execute o comando: `fsck.ext3 -l blocos-defeituosos.lista -f /dev/hd??`.
Substitua o dispositivo, pelo dispositivo que verificou com o `badblocks`. O arquivo `blocos-defeituosos.list` contém a lista de blocos gerada pelo `badblocks` que serão marcados como defeituosos. Para mais detalhes sobre as opções de checagem usada pelos programas, veja [badblocks, Seção 26.4](#) e [fsck.ext2, Seção 26.1.1](#).
-

26.7 Limpando arquivos de LOGS

Tudo que acontece em sistemas GNU/Linux pode ser registrado em arquivos de log em `/var/log`, como vimos anteriormente. Eles são muito úteis por diversos motivos, para o diagnóstico de problemas, falhas de dispositivos, checagem da segurança, alerta de eventuais tentativas de invasão, etc.

O problema é quando eles começam a ocupar muito espaço em seu disco. Verifique quantos Megabytes seus arquivos de LOG estão ocupando através do comando `cd /var/log;du -hc`. Antes de fazer uma limpeza nos arquivos de LOG, é necessário verificar se eles são desnecessários e só assim zerar os que forem dispensáveis. Não é recomendável apagar um arquivo de log pois ele pode ser criado com permissões de acesso indevidas (algumas distribuições fazem isso). Você pode usar o comando: `echo -n >arquivo` ou o seguinte shell script para zerar todos os arquivos de LOG de uma só vez (as linhas iniciante com `#` são comentários):

```
#!/bin/sh

cd /var/log

for l in `ls -p|grep '/'`; do

echo -n >$l &>/dev/null

echo Zerando arquivo $l...

done

echo Limpeza dos arquivos de log concluída!
```

Copie o conteúdo acima em um arquivo com a extensão `.sh`, dê permissão de execução com o `chmod` e o execute como utilizador `root`. É necessário executar este script para zerar arquivos de log em subdiretórios de `/var/log`, caso sejam usados em seu sistema.

Algumas distribuições, como a Debian GNU/Linux, fazem o arquivamento automático de arquivos de LOGs em arquivos `.gz` através de scripts disparados automaticamente pelo `cron`. **ATENÇÃO: LEMBRE-SE QUE O SCRIPT ACIMA APAGARÁ TODOS OS ARQUIVOS DE LOGs DO SEU SISTEMA SEM POSSIBILIDADE DE RECUPERAÇÃO. TENHA ABSOLUTA CERTEZA DO QUE NÃO PRECISARÁ DELES QUANDO EXECUTAR O SCRIPT ACIMA!**

26.8 Recuperando partições apagadas

Caso tenha apagado uma partição acidentalmente ou todas as partições do seu disco, uma forma simples de recuperar todos os seus dados é simplesmente recriar todas as partições com o tamanho **EXATAMENTE** igual ao existente anteriormente. Isto deve ser feito dando a partida com um disquete ou CD de inicialização. Após recriar todas as partições e seus tipos (83, 82 8e, etc), execute novamente o lilo para recriar o setor de boot do HD e garantir que a máquina dará o boot.

A recuperação desta forma é possível porque quando se cria ou apaga uma partição, você está simplesmente delimitando espaço onde cada sistema de arquivos gravará seus dados, sem fazer nenhuma alteração dentro dele. Assim, é também útil manter uma cópia dos tamanhos usados durante o processo de criação das partições para ser usado como recuperação em uma possível emergência.

26.9 Recuperando a senha de root perdida

Uma situação que você deve ter se deparado (ou algum dia ainda vai se deparar) é precisar alterar a senha de root e não sabe ou não lembra a senha atual. Esta situação também pode ser encontrada quando ocorre uma falha de disco, falha elétrica, reparos em uma máquina que não detém sua manutenção, etc. A melhor notícia é que a alteração da senha de root é possível e não apresenta problema qualquer para o sistema. Existem várias formas para se fazer isto, a forma que descreverei abaixo assume que você tem acesso a um outro dispositivo de partida que não seja o HD do Linux (*CD-ROM, disquetes, outro disco rígido*, etc). Assim, mesmo que encontre uma senha de BIOS em uma máquina, poderá colocar o disco rígido em outra máquina e executar estes procedimentos.

OBS: Estes procedimentos tem fins didáticos e administrativos, não sendo escritos com a intenção de fornecer mal uso desta técnica. Entender a exposição de riscos também ajuda a desenvolver novas técnicas de defesa para sistemas críticos, e estas são totalmente possíveis e as mais usadas documentadas neste guia.

- Como primeiro passo consiga um CD de partida ou disquete de uma distribuição Linux. Normalmente os mesmos CDs que usou para instalar sua distribuição também são desenvolvidos para permitir a manutenção do sistema, contendo ferramentas diversas e um terminal virtual disponível para trabalhos manuais (tanto de instalação como manutenção).
- Vá até a BIOS da máquina e altere a ordem de inicialização para que seu sistema inicialize a partir do disquete ou CD-ROM (dependendo do método escolhido no passo anterior).
- Inicialize a partir do Disquete/CD-ROM.
- Na maioria dos casos você provavelmente estará utilizando o CD-ROM que usou para instalar sua distribuição. Imediatamente quando o programa de instalação for iniciado, pressione **ALT+F2** para alternar para o segundo

terminal virtual do sistema. O segundo terminal esta sempre disponível nas distribuições distribuições Debian, Red Hat, Conectiva, Fedora, etc.

- O próximo passo será montar sua partição raiz para ser possível alterar sua senha de root. Para isto, crie um diretório onde a partição será montada (por exemplo, `/target`) e execute o comando `mount /dev/hda1 /target` (assumindo que `/dev/hda1` é a partição que contém seu sistema de arquivos raiz (/)).
- Entre no diretório `/target` (`cd /target`) e torne-o seu diretório raiz atual com o comando: `chroot ..`
- digite `passwd` e entre com a nova senha de superutilizador.
- saia do `chroot` digitando `exit`
- Digite `sync` para salvar todas as alterações pendentes para o disco e reinicie o sistema (pressionando-se as teclas `CTRL+ALT+DEL`, `init 6`, `reboot`).
- Retire o CD da unidade de discos e altere sua BIOS para dar a partida a partir do disco rígido.
- Teste e verifique se a senha de root foi alterada.

Normalmente as distribuições seguem o padrão FHS, mantendo binários de administração necessários para recuperação do sistema em caso de panes dentro da partição `/`, se este não for o caso de sua distribuição (hoje em dia é raro), você terá que montar sistemas de arquivos adicionais (como o `/usr`, `/var`) ou então o comando `passwd` não será encontrado ou terá problemas durante sua execução.

26.10 Tarefas automáticas de manutenção do sistema

Os arquivos responsáveis pela manutenção automática do sistema se encontram em arquivos individuais localizados nos diretórios `/etc/cron.daily`, `/etc/cron.weekly` e `/etc/cron.monthly`. A quantidade de arquivos depende da quantidade de pacotes instalado em seu sistema, porque alguns programam tarefas nestes diretórios e não é possível descrever todas, para detalhes sobre o que cada arquivo faz veja o cabeçalho e o código de cada arquivo.

Estes arquivos são executados pelo `cron` através do arquivo `/etc/crontab`. Você pode programar quantas tarefas desejar, para detalhes veja [cron, Seção 26.11](#) e [at, Seção 26.12](#). Alguns programas mantêm arquivos do `cron` individuais em `/var/spool/cron/crontabs` que executam comandos periodicamente.

26.11 cron

O `cron` é um daemon que permite o agendamento da execução de um comando/programa para um determinado dia/mês/ano/hora. É muito usado em tarefas de arquivamento de logs, checagem da integridade do sistema e execução de programas/comandos em horários determinados.

As tarefas são definidas no arquivo `/etc/crontab` e por arquivos individuais de utilizadores em `/var/spool/cron/crontabs/[utilizador]` (criados através do programa `crontab`). Adicionalmente a distribuição Debian utiliza os arquivos no diretório `/etc/cron.d` como uma extensão para `/etc/crontab`.

Para agendar uma nova tarefa, basta editar o arquivo `/etc/crontab` com qualquer editor de texto (como o `ae` e o `vi`) e definir o mês/dia/hora que a tarefa será executada. Não é necessário reiniciar o daemon

do cron porque ele verifica seus arquivos a cada minuto. Veja a seção [O formato de um arquivo crontab, Seção 26.11.1](#) para entender o formato de arquivo cron usado no agendamento de tarefas.

26.11.1 O formato de um arquivo crontab

O arquivo `/etc/crontab` tem o seguinte formato:

```
52 18 1 * * root run-parts --report /etc/cron.monthly
```

```
| | | | | | |
```

```
| | | | | | \_ Comando que será executado
```

```
| | | | | |
```

```
| | | | | \_ UID que executará o comando
```

```
| | | | |
```

```
| | | | \_ Dia da semana (0-7)
```

```
| | | |
```

```
| | | \_ Mês (1-12)
```

```
| | |
```

```
| | \_ Dia do Mês (1-31)
```

```
| |
```

```
| \_ Hora
```

```
|
```

```
\_ Minuto
```

Onde:

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 de Domingo a Sábado). Note que tanto 0 e 7 equivalem a Domingo.

utilizador

O utilizador especificado será usado para executar o comando (o utilizador deverá existir).

comando

Comando que será executado. Podem ser usados parâmetros normais usados na linha de comando.

Os campos do arquivo são separados por um ou mais espaços ou tabulações. Um asterisco * pode ser usado nos campos de data e hora para especificar todo o intervalo disponível. O hífen – serve para especificar períodos de execução (incluindo a o número inicial/final). A vírgula serve para especificar lista de números. Passos podem ser especificados através de uma /. Veja os exemplos no final desta seção.

O arquivo gerado em `/var/spool/cron/crontabs/[utilizador]` pelo `crontab` tem o mesmo formato do `/etc/crontab` exceto por não possuir o campo `utilizador` (UID), pois o nome do arquivo já identifica o utilizador no sistema.

Para editar um arquivo de utilizador em `/var/spool/cron/crontabs` ao invés de editar o `/etc/crontab` use `crontab -e`, para listar as tarefas daquele utilizador `crontab -l` e para apagar o arquivo de tarefas do utilizador `crontab -r` (adicionalmente você pode remover somente uma tarefa através `docrontab -e` e apagando a linha correspondente).

OBS: Não esqueça de incluir uma linha em branco no final do arquivo, caso contrário o último comando não será executado.

O `cron` define o valor de algumas variáveis automaticamente durante sua execução; a variável `SHELL` é definida como `/bin/sh`, `PATH` como `/usr/bin:/bin`, `LOGNAME`, `MAILTO` e `HOME` são definidas através do arquivo `/etc/passwd`. Os valores padrões destas variáveis podem ser substituídos especificando um novo valor nos arquivos do `cron`.

Exemplos de um arquivo `/etc/crontab`:

```
SHELL=/bin/sh
```

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
```

```
00 10 * * * root sync
```

```
# Executa o comando sync todo o dia as 10:00
```

```
00 06 * * 1 root updatedb
```

```
# Executa o comando updatedb toda segunda-feira as 06:00.
```

```
10,20,40 * * * * root runq
```

```
# Executa o comando runq todos os dias e a toda a hora em 10, 20 e 40 minutos.
```

```
*/10 * * * * root fetchmail
```

```
# Executa o comando fetchmail de 10 em 10 minutos todos os dias
```

```
15 0 25 12 * root echo "Feliz Natal"|mail john
```

```
# Envia um e-mail as 0:15 todo o dia 25/12 para john desejando um feliz natal.
```

```
30 5 * * 1-6 root poff
```

```
# Executa o comando poff automaticamente as 5:30 de segunda-feira a sábado.
```

26.12 at

O `at` agenda tarefas de forma semelhante ao `cron` com uma interface que permite a utilização de linguagem natural nos agendamentos. Sua principal aplicação é no uso de tarefas que sejam disparadas somente uma vez. Uma característica deste programa é a execução de aplicativos que tenham passado de seu horário de execução, muito útil se o computador é desligado com frequência ou quando ocorre uma interrupção no fornecimento de energia.

Para utilizar o `at`, instale-o com o comando: `apt-get install at`. O próximo passo é criar os arquivos `/etc/at.allow` e `at.deny`. Estes arquivos são organizados no formato de um utilizador por linha. Durante o agendamento, é verificado primeiro o arquivo `at.allow` (lista de quem pode executar comandos) e depois o `at.deny` (lista de quem NÃO pode executar comandos). Caso eles não existam, o agendamento de comandos é permitido a todos os utilizadores.

Abaixo seguem exemplos do agendamento através do comando `at`:

```
echo ls | at 10am today
```

Executa as 10 da manha de hoje

```
echo ls | at 10:05 today
```

Executa as 10:05 da manha de hoje

```
echo ls | at 10:05pm today
```

Executa as 10:05 da noite de hoje

```
echo ls | at 22:05 today
```

Executa as 22:05 da noite de hoje

```
echo ls | at 14:50 tomorrow
```

Executa o comando amanhã as 14:50 da tarde

```
echo ls | at midnight
```

Executa o comando a meia noite de hoje

```
echo ls | at midnight tomorrow
```

Executa o comando a meia noite de amanhã

```
echo ls | at noon
```

Executa o comando de tarde (meio dia).

```
at -f comandos.txt teatime
```

Executa os comandos especificados no arquivo "comandos.txt" no horário do café da tarde (as 16:00 horas).

```
at -f comandos.txt +3 minutes
```

Executa os comandos especificados no arquivo "comandos.txt" daqui a 3 minutos. Também pode ser especificado "hours" ou "days".

```
at -f comandos.txt tomorrow +3 hours
```

Executa os comandos especificados no arquivo "comandos.txt" daqui a 3 horas no dia de amanhã. (se agora são 10:00, ela será executada amanhã as 13:00 da tarde).

Todas as tarefas agendadas são armazenadas em arquivos dentro do diretório `/var/spool/cron/atjobs`. A sintaxe de comandos para gerenciar as tarefas é semelhante aos utilitários do `lpd`: Para ver as tarefas, digite `atq`. Para remover uma tarefa, use o comando `atrm` seguido do número da tarefa obtida pelo `atq`.