

# Capítulo 8

## Administração do sistema operacional

- Instalação de programas no Linux
- Compactadores e backup
- Arquivos de lote no Linux
- Arquivos de Linux

**C**onhecidos os diferentes tipos de sistemas operacionais existentes no mercado, é o momento de abordar a sua gestão, ou seja, analisar as atividades que fazem parte do cotidiano de um técnico de informática nesse âmbito. É fundamental saber algumas práticas que podem ser adotadas durante o trabalho, seja o profissional um responsável pelo parque de máquinas de sua empresa, seja um programador, com foco direcionado a projetos.

## 8.1. Instalação de programas no Linux

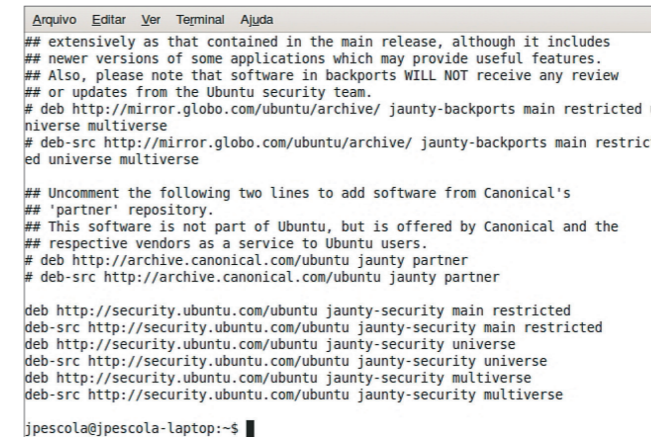
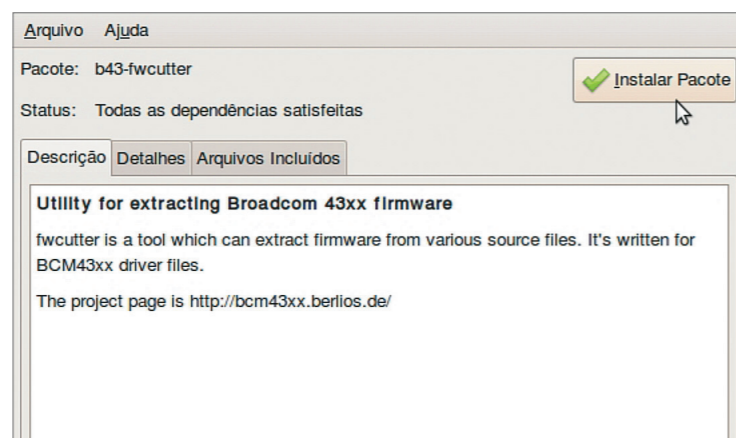
Um bom começo é abordar tudo o que envolve a instalação de programas no Linux, porque a maioria daqueles que se dedicam ao segmento está mais acostumada a fazer isso no Windows. E é simples. Basta clicar sobre o arquivo do instalador e seguir os passos do assistente de instalação (next, next, next, finish...). O Linux também oferece essa possibilidade bem amigável. Acessando os mesmos sites, é possível baixar softwares para Linux e instalar um por um, com um duplo clique do mouse. Mas essa não é a única forma de fazê-lo. Vejamos quais são as outras alternativas.

### 8.1.1. Instalando programas a partir do CD da distribuição

Ao inserir o CD do Ubuntu no drive, conseguimos acessar diversas pastas com arquivos de instalação, executáveis manualmente com um duplo clique do mouse. Esses arquivos têm a extensão “.deb”. Durante a execução, veremos uma janela como a da figura 217. Se você clicar no botão “Instalar Pacote”, o utilitário vai adicionar o novo software ao sistema operacional.

**Figura 217**

Instalando um programa a partir do arquivo no CD.



**Figura 218**

Arquivo “sources.list”.

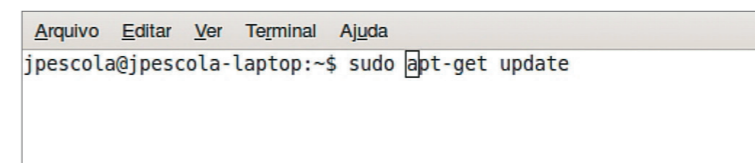
### 8.1.2. Instalando programas a partir de sites de download

Arquivos “.deb” são pacotes de software do Debian, distribuição Linux bastante famosa, da qual derivou o Ubuntu. Portanto, qualquer arquivo com extensão “.deb” pode ser instalado no Ubuntu. Se você encontrar em um site um pacote “.deb” e quiser instalá-lo em seu equipamento, basta baixar o arquivo e dar duplo clique sobre ele, como fez com o arquivo do CD no item anterior.

### 8.1.3. Gerenciador de pacotes do Linux (apt-get e synaptic)

Um conceito importante sobre os sistemas Linux é o de servidores de pacotes. Já sabemos que um pacote é um software que pode ser instalado no Linux. E um servidor de pacotes é uma máquina disponível na rede (ou na internet), para tornar disponíveis diversos pacotes a serem baixados por distribuições Linux, como o Ubuntu. Assim que é instalado, o Ubuntu configura, automaticamente, um servidor de pacotes do país cuja linguagem foi escolhida no início do processo. Essa configuração é feita no arquivo “sources.list” (lista de fontes de software), que fica na pasta “/etc/apt” (figura 218). A partir desse momento, podemos utilizar o comando “apt-get” (gerenciador de pacotes usado no Ubuntu) e o comando “apt-cache” (programa usado para manter um banco de dados) para instalar, de maneira extremamente fácil, um novo pacote no PC.

É recomendável atualizar a lista de pacotes atual, antes de instalar um, pois novos pacotes são criados e atualizados diariamente no servidor. Para fazer essa atualização, digite o comando “sudo apt-get update” (figura 219-A). Atenção: para utilizar o “apt-get”, é preciso manter a conexão com a internet, pois o servidor de pacotes não está na máquina nem na rede local.



**Figura 219-A**

Atualizando a lista de pacotes.

#### DICA

Podemos alterar o endereço “http” do servidor que será utilizado para baixar os pacotes. Basta consultar os que estão disponíveis no site do Ubuntu e trocar o que estava no “sources.list”. Lembre-se de salvar as alterações antes de fechar o arquivo.

**Figura 219-B**

Atualizando a lista de pacotes com o servidor remoto.

```
Arquivo Editar Ver Terminal Ajuda
jpescola@jpescola-laptop:~$ sudo apt-get update
[sudo] password for jpescola:
Ign cdrom://Ubuntu 8.04_Hardy_Heron_ - Release i386 (20080423) hardy/main Translation-pt_BR
Ign cdrom://Ubuntu 8.04_Hardy_Heron_ - Release i386 (20080423) hardy/restricted Translation-pt_BR
Atingido http://mirror.globo.com jaunty Release.gpg
Obter:1 http://mirror.globo.com jaunty/main Translation-pt_BR [256kB]
Atingido http://security.ubuntu.com jaunty-security Release.gpg
Ign http://security.ubuntu.com jaunty-security/main Translation-pt_BR
Ign http://security.ubuntu.com jaunty-security/restricted Translation-pt_BR
Ign http://security.ubuntu.com jaunty-security/universe Translation-pt_BR
Ign http://security.ubuntu.com jaunty-security/multiverse Translation-pt_BR
56% [1 Translation-pt_BR 129600/256kB 50%]
```

Se o resultado for parecido com o da figura 219-B, é sinal de que os pacotes foram atualizados com sucesso. Caso haja algum erro durante a atualização, abra novamente o arquivo “sources.list” e confira se o endereço do servidor está correto.

O próximo passo é consultar a lista de pacotes utilizando uma palavra-chave. Isso nos permite fazer uma busca, no servidor, à procura de pacotes que têm algo a ver com a palavra “game” (jogo). O resultado retorna os nomes dos aplicativos, seguidos de uma pequena descrição. Fazer uma busca é importante, pois precisaremos do nome correto do pacote para instalá-lo, como mostra a figura 220.

Chegou o momento de instalar o pacote. Digite o comando “apt-get install” (o parâmetro “install” significa instalar), seguido do nome do pacote que selecionamos ao digitar o comando anterior para consulta (figura 221).

Para que o pacote escolhido funcione corretamente, são necessários outros pacotes, que são detectados e instalados pelo utilitário, conforme ilustra a figura 222.

**Figura 220**

Busca, por pacotes, via palavra-chave.

```
Arquivo Editar Ver Terminal Ajuda
Atingido http://mirror.globo.com jaunty-updates Release
Atingido http://mirror.globo.com jaunty/main Packages
Atingido http://mirror.globo.com jaunty/restricted Packages
Atingido http://mirror.globo.com jaunty/main Sources
Atingido http://mirror.globo.com jaunty/restricted Sources
Atingido http://mirror.globo.com jaunty/universe Packages
Atingido http://mirror.globo.com jaunty/universe Sources
Atingido http://mirror.globo.com jaunty/multiverse Packages
Atingido http://mirror.globo.com jaunty/multiverse Sources
Atingido http://mirror.globo.com hardy/main Packages
Atingido http://mirror.globo.com hardy/restricted Packages
Atingido http://mirror.globo.com hardy/universe Packages
Atingido http://mirror.globo.com hardy/multiverse Packages
Atingido http://mirror.globo.com jaunty-updates/main Packages
Atingido http://mirror.globo.com jaunty-updates/restricted Packages
Atingido http://mirror.globo.com jaunty-updates/main Sources
Atingido http://mirror.globo.com jaunty-updates/restricted Sources
Atingido http://mirror.globo.com jaunty-updates/universe Packages
Atingido http://mirror.globo.com jaunty-updates/universe Sources
Atingido http://mirror.globo.com jaunty-updates/multiverse Packages
Atingido http://mirror.globo.com jaunty-updates/multiverse Sources
Baixados 1102kB em 6s (167kB/s)
Lendo listas de pacotes... Pronto
jpescola@jpescola-laptop:~$ apt-cache search game
```

**Figura 221**

Lista de pacotes relacionados com a palavra-chave “game”.

```
Arquivo Editar Ver Terminal Ajuda
moagg-data - data files for the moagg game
moaggedit - map editor for the Moagg game
opendb - Web-based lending database written in PHP
pcsx-bin - Sony PlayStation emulator -- binary
pcsx-i18n - Sony PlayStation emulator -- extra languages
psemu-input-omnjoy - Controller/keyboard plugin for PSX emulators
psemu-input-padjoy - Controller plugin for PSX emulators
psemu-sound-alsa - ALSA sound plugin for PSX emulators
psemu-sound-oss - OSS sound plugin for PSX emulators
psemu-video-x11 - software graphics plugin for PSX emulators
schafkopf - popular Bavarian card game for KDE
starfighter - 2D scrolling shooter game
starfighter-data - data files for Starfighter game
superkaramba-kde4 - a program based on karamba improving the eyecandy of KDE 4
vertex - a GTK+OpenGL 3D modeller
worlded - world editor for Arkkrpg
xbat - classic shoot 'em up game for X11
xmms-gbs - Gameboy sound player plugin for XMMS
chdrvfont - Fonte Kuo Chiao 16x16 para terminal de console chinês CHDRV.
alien-arena-dbg - debugging symbols for alien-arena
alien-arena-server-dbg - debugging symbols for alien-arena-server
libdscaler - Video deinterlacer plugins from the DScaler project
robotour - control mobile robots in this programmer's game
jpescola@jpescola-laptop:~$
```

**Figura 222**

Instalando um novo pacote.

```
Arquivo Editar Ver Terminal Ajuda
jpescola@jpescola-laptop:~$ sudo apt-get install robotour
Lendo listas de pacotes... Pronto
Construindo árvore de dependências
Lendo informação de estado... Pronto
Os pacotes extra a seguir serão instalados:
 libsox-fmt-alsa libsox-fmt-base libsox1 libwxbase2.6-0 libwxgtk2.6-0 sox
Pacotes sugeridos:
 libsox-fmt-all
Os NOVOS pacotes a seguir serão instalados:
 libsox-fmt-alsa libsox-fmt-base libsox1 libwxbase2.6-0 libwxgtk2.6-0
 robotour sox
0 pacotes atualizados, 7 pacotes novos instalados, 0 a serem removidos e 0 não a
tualizados.
É preciso baixar 4458kB de arquivos.
Depois desta operação, 12,7MB adicionais de espaço em disco serão usados.
Você quer continuar [S/n]?
```

Antes de começar a baixar os pacotes necessários e suas dependências, no entanto, o utilitário solicita uma confirmação e informa a quantidade de arquivos e o tamanho total em MB (Megabytes) de todos. Para continuar o processo, pressione a tecla “Enter”. Pronto. O pacote é instalado com sucesso.

## 8.2. Compactadores e backup

Provavelmente você já deve ter recebido um e-mail com um arquivo anexado cuja extensão seja “.zip” ou “.rar”: eles passaram por um processo de compactação (compressão). O motivo é simples: quanto menor o arquivo, mais rápido será o seu envio como anexo, pois ele utiliza menos espaço em disco.

Existem diversos tipos de compactadores no mercado, além de diferentes formatos de compactação. Atualmente, os formatos RAR e ZIP têm sido os mais utilizados, mas o 7Z também é muito bem aceito, em especial por ser um software do tipo OpenSource (código aberto). Cada um desses formatos pode ser adotado tanto no Windows quanto no Linux. Basta instalar um software de compactação que possua suporte para o formato escolhido. Alguns até dispõem de suporte para mais de um formato de compressão e descompressão.

### DICA

1. Nem sempre os novos pacotes criam atalhos no menu “Aplicativos” do Ubuntu. Então, pode ser necessário digitar o comando para chamar o aplicativo no prompt de comandos (terminal).
2. Quando não quiser mais o programa ou se precisar de mais espaço em disco, você poderá desinstalá-lo utilizando o parâmetro “remove” do comando “apt-get”.

### 8.2.1. Compactadores nativos do Linux

O comando “tar” está presente em qualquer distribuição Linux e permite armazenar ou compactar pastas ou conjuntos de arquivos em um único arquivo. Sua sintaxe é a seguinte:

**tar [parâmetros] [nome do arquivo] [itens a serem arquivados]**

Imagine uma pasta chamada “backup”, com arquivos a compactar. Primeiramente digite o comando “du” para ver qual é o espaço ocupado em disco pela pasta (figura 223).

Para arquivá-la em um arquivo com a extensão “.tar”, digite o comando mostrado na figura 224.

Digite novamente o comando “du” para visualizar o tamanho do arquivo “.tar” criado, conforme a figura 225-A.

Veja que o arquivamento diminui um pouco o tamanho do arquivo, mas isso não chega a ser uma compactação, pois esse processo converte os bits repetitivos da estrutura de todos os arquivos, fazendo com que seja necessário menos espaço em disco para armazenamento. Compacte o arquivo, ainda utilizando o comando “.tar”, mas agora com o ZIP (figura 225-B). Agora, o arquivo resultante tem 725 KB. É menos de um quarto do tamanho da pasta de origem.

**Figura 223**

Visualização do espaço em disco usado pela pasta “backup”.

```
Arquivo Editar Ver Terminal Ajuda
jpescola@jpescola-laptop:~$ du backup.tar
3003 backup.tar
jpescola@jpescola-laptop:~$ du backup
3064 backup
jpescola@jpescola-laptop:~$
```

**Figura 224**

Arquivando a pasta “backup”.

```
Arquivo Editar Ver Terminal Ajuda
jpescola@jpescola-laptop:~$ 7zr a backup.7z backup
```

**Figura 225-A**

Visualizando o tamanho do arquivo “.tar gerado”.

```
Arquivo Editar Ver Terminal Ajuda
jpescola@jpescola-laptop:~$ du backup.tar
3003 backup.tar
jpescola@jpescola-laptop:~$ du backup
3064 backup
jpescola@jpescola-laptop:~$
```

**Figura 225-B**

O arquivo compactado, menor do que o arquivado, e a pasta de origem.

```
Arquivo Editar Ver Terminal Ajuda
jpescola@jpescola-laptop:~$ du backup.tar
3003 backup.tar
jpescola@jpescola-laptop:~$ du backup
3064 backup
jpescola@jpescola-laptop:~$ tar czf backup.tar.gz backup
jpescola@jpescola-laptop:~$ du backup.tar.gz
725 backup.tar.gz
jpescola@jpescola-laptop:~$
```

```
Arquivo Editar Ver Terminal Ajuda
jpescola@jpescola-laptop:~$ sudo apt-get install rar
```

**Figura 226**

Instalação do “rar”.

```
Arquivo Editar Ver Terminal Ajuda
jpescola@jpescola-laptop:~$ tar cf backup.tar backup
```

**Figura 227**

Compactando em formato “rar”.

### 8.2.2. Instalação de novos compactadores, usando o gerenciador de pacotes

Agora instale o compactador “rar”, utilizando o comando “apt-get”, como se viu no item anterior (figura 226).

Em seguida crie um arquivo compactado com o formato “rar”, como ilustra a figura 227.

Observe que é preciso digitar o comando “rar”, seguido do parâmetro “a”, que significa “add” (adicionar). O próximo passo é o nome do arquivo resultante (backup.rar) e, por último, a pasta que será compactada pelo comando. Veja com que tamanho o arquivo “backup.rar” ficou (figura 228).

Com o formato rar, o arquivo resultante é ainda menor. Isso acontece porque o seu algoritmo de compactação é mais eficiente que o utilizado pelo formato “zip”. Justamente por comprimir mais, o “rar” precisa de um tempo maior para efetuar todos os cálculos de compactação e, por isso, é mais lento que o “zip”.

Já o 7Z é do tipo OpenSource, o que significa que diversas pessoas constantemente ajustam seu algoritmo, para torná-lo cada vez mais eficiente. Faça um teste prático. Instale-o via “apt-get” (figura 229).

```
Arquivo Editar Ver Terminal Ajuda
jpescola@jpescola-laptop:~$ du backup.rar
673 backup.rar
jpescola@jpescola-laptop:~$
```

**Figura 228**

Visualização do espaço em disco usado por “backup.rar”.

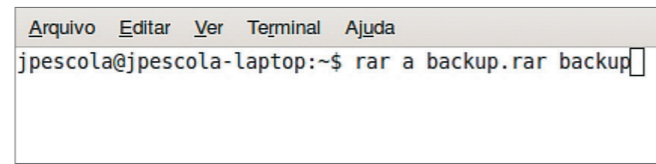
```
Arquivo Editar Ver Terminal Ajuda
jpescola@jpescola-laptop:~$ sudo apt-get install p7zip
```

**Figura 229**

Instalando o compactador para usar o formato “7Z”.

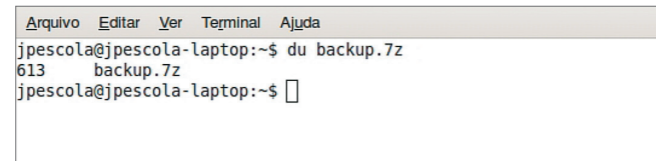
**Figura 230**

Compactando com “7zr”.



**Figura 231**

O arquivo compactado com “7z” é menor que os demais compactados.



Para compactar a mesma pasta, com o comando “7zr”, faça como mostra a figura 230.

Veja que o parâmetro “add” ainda é o mesmo, mas agora é preciso especificar uma extensão diferente para o arquivo (“.7z”), como aparece na figura 231. Fica evidente que o “7z” conseguiu realmente compactar mais do que os outros formatos.

### 8.3. Arquivos de lote (batch)

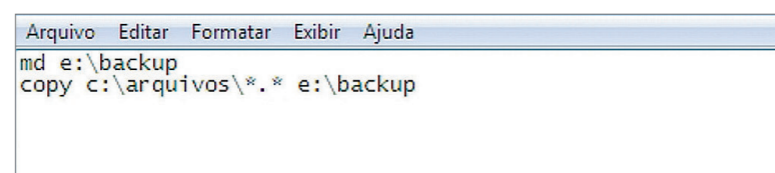
Qualquer sistema operacional permite a criação de arquivos batch, também conhecidos como de lote e que servem para automatizar tarefas repetitivas e cotidianas do administrador do sistema. Com um arquivo de lote, podemos configurar backups, gerar relatórios etc. Podem ser automatizados, se forem colocados em conjunto com os respectivos programas de agendamento de tarefas. Arquivos batch devem ser criados com o uso de um editor de textos simples (padrão ASCII) e salvos com a extensão correspondente, para que o sistema operacional o reconheça. Normalmente, seu conteúdo é formado por conjuntos de comandos da linguagem de cada sistema operacional; portanto, é preciso conhecer esses comandos para criar arquivos de lote.

#### 8.3.1. Criação de arquivos de lote usando comandos do prompt do Windows

Vamos aprender agora a criar um arquivo “Batch” no Windows utilizando comandos básicos do prompt, para gerar uma pasta de backup no pen-drive. Copie os arquivos de alguns diretórios do sistema operacional para essa pasta. Na prática, apenas dois comandos são adicionados ao arquivo: o primeiro cria a pasta “backup” na unidade de disco “E”; o segundo copia todos os dados da pasta “Arquivos” localizada na unidade “C” para a recém-criada (figura 232).

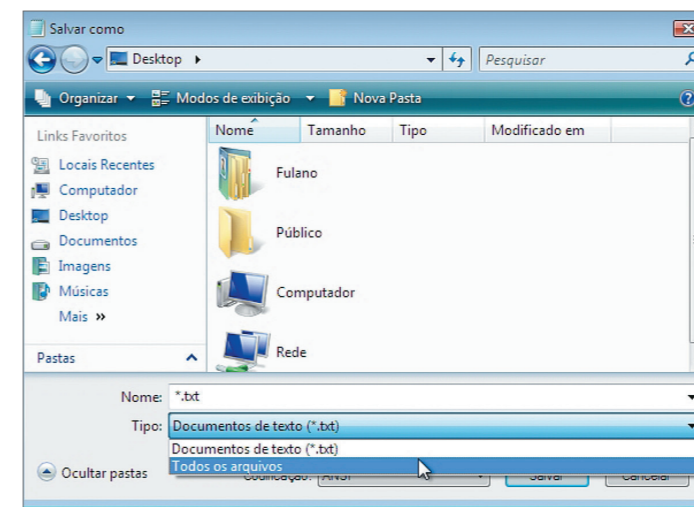
**Figura 232**

Criação de um arquivo batch de backup.



**Figura 233**

Salvando o arquivo “.bat”

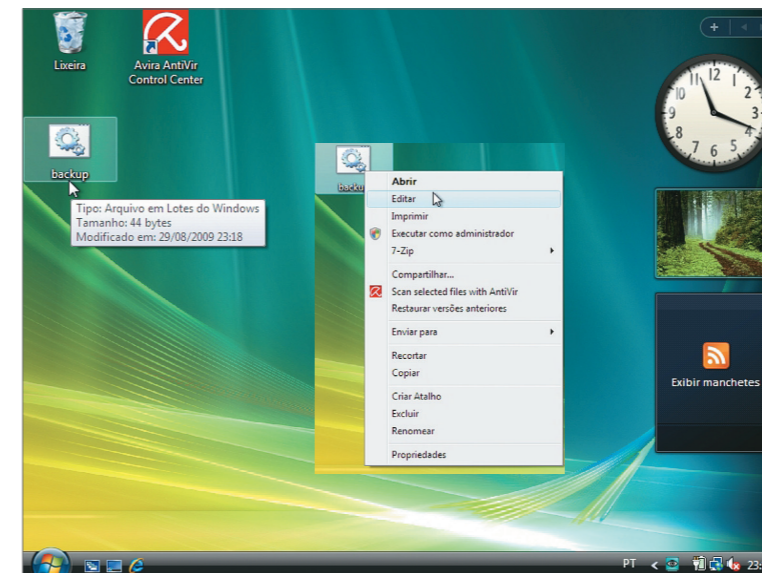


O próximo passo é salvar o arquivo com o nome “backup.bat”. No caso do Windows, é necessário usar a extensão “.bat” para que o sistema operacional o reconheça como sendo do tipo batch. Para salvar corretamente é necessário alterar o tipo do documento para “Todos os Arquivos”. Caso contrário, o arquivo terá extensão “.txt”. Então, digite o nome do arquivo “batch” no campo “Nome”, escolha o diretório de destino (no nosso caso é o Desktop) e clique no botão “Salvar” (figura 233).

Um ícone característico sinaliza que o arquivo é executável (Batch). Basta, agora, dar um duplo clique sobre esse ícone para abrir uma janela do prompt na tela (figura 234). Mas ela não permanece aberta e há uma razão para isso: é que não foi usado um comando para parar a execução e ele pode ser útil para que você veja o resultado do script. Para alterar o script, basta clicar com o botão direito do mouse sobre o arquivo e escolher a opção “Editar”.

**Figura 234**

Arquivo criado na área de trabalho (desktop).



**Figura 235**

Adicionando o comando “pause”.

```
Arquivo Editar Formatar Exibir Ajuda
md e:\backup
copy c:\arquivos\*. * e:\backup
pause|
```

Adicione o comando “Pause” no final do código, fazendo com que ele pare nesse ponto antes de fechar a janela do prompt (figura 235).

### 8.3.2. Comandos adicionais aplicáveis em arquivos de lote

Além dos comandos do prompt, é possível adicionar outros pertencentes aos scripts batch do Windows. Se você preferir, pode colocar também um caractere @ na frente de cada comando, para que ele não seja apresentado no prompt. Assim, somente o resultado do comando aparecerá (figura 236).

O comando **“color”** também pode ser útil. Permite configurar uma cor específica para a execução do script e diferenciá-lo do prompt normal ou mesmo de outros scripts. Já o “title” permite especificar um título para a janela do prompt que está executando o script (figura 237).

Observe que adicionamos um caractere que significa maior (>) e a palavra “nul” à frente do “pause”. Isso faz com que o retorno do “pause”, que é a frase “pressione qualquer tecla para continuar...”, permaneça como oculto do prompt, no final da execução (figura 238).

**Figura 236**

Caracteres como @ ocultam os comandos.

```
backup - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
md e:\backup
copy c:\arquivos\*. * e:\backup
pause|
```

**Figura 237**

Incrementando o script.

```
backup - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
@color f0
@title Script de Backup
@md e:\backup
@copy c:\arquivos\*. * e:\backup
@pause>nul
```

**Figura 238**

Script personalizado com título, cor e pausa.

```
Script de Backup
c:\arquivos\nome.txt
1 arquivo(s) copiado(s).
```

```
backup - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
@color f0
@title Script de Backup
@echo Criando pasta de destino
@md e:\backup
@echo Copiando arquivos, aguarde...
@copy c:\arquivos\*. * e:\backup
@echo Fim do backup, tenha um bom dia!
@pause>nul
```

**Figura 239**

Uso do “echo” para informar melhor o usuário.

```
Script de Backup
Criando pasta de destino
Copiando arquivos, aguarde...
c:\arquivos\nome.txt
1 arquivo(s) copiado(s).
Fim do backup, tenha um bom dia!
```

**Figura 240**

Resultado final do script de backup.

Provavelmente, nenhum usuário leigo utilizará esse script. Mesmo assim, é importante frisar que ainda está faltando alguma coisa. Talvez informações sobre o que está sendo feito durante a execução do script.

Para exibir dizeres ou frases no script, podemos utilizar o comando “echo” (figura 239).

Agora execute o script e observe o resultado. É aconselhável que você aproveite para criar seu próprio arquivo “batch” e automatizar seu backup diário (figura 240).

### 8.3.3. Funções avançadas aplicáveis em arquivos de lote

Há diversas funções avançadas que podem ser aplicadas em arquivos de lote do Windows. Vejamos, aqui, apenas dois desses conceitos, dos muitos que podem ser adicionados a partir de pesquisas de documentação específicas na internet. Podemos direcionar a saída de um comando para um arquivo gerando um relatório, no final do backup, com o nome dos arquivos copiados e a data de realização do backup – entre outras opções. Utilizando um sinal de maior (>) após o comando, direcionamos a saída de um comando para um dispositivo diferente do monitor. Pode ser um arquivo, por exemplo. Dois sinais de maior (>>) direcionam a saída para o arquivo, incrementando seu conteúdo. Se usarmos um sinal só, os dados que antes estavam armazenados no arquivo serão perdidos. Mas com dois sinais eles se mantêm e a nova informação é incrementada.

Outro recurso útil é o de substrings. É possível pegar uma parte de uma string ou variável de ambiente e armazenar em uma variável ou apresentar na tela. Exemplo: “echo %date%” apresenta a data atual na tela e “%date~-0,3%” exibe somente os três primeiros caracteres de “%date%” (a partir do índice 0, pega três caracteres), ou seja, só o dia da semana. Trabalhar com parâmetros também é bem fácil. O sinal “%1” representa o primeiro parâmetro passado na linha de comando. O “%2” significa o segundo.

**DICA**

Faça uma boa revisão do que aprendeu sobre os comandos do Windows e do Linux antes de começar a trabalhar com arquivos de lote.

## 8.4. Arquivos de lote no Linux

### 8.4.1. Introdução aos shell scripts

No Linux também é possível criar arquivos de lote, que devem ser salvos com a extensão “.sh”. A linguagem de scripts utilizada para criar arquivos de lote é chamada de “shell script”. É extremamente poderosa, permitindo laços de repetição, condições etc. Gerar um shell script requer um editor de textos-padrão ASCII, como o gedit do Ubuntu.

### 8.4.2. Scripts básicos utilizando comandos do Linux

Vamos criar nosso primeiro script para Linux. Inicialmente abra o Gedit ou o Editor de Textos no menu Aplicativos / Acessórios / Editor de Textos do Gnome.

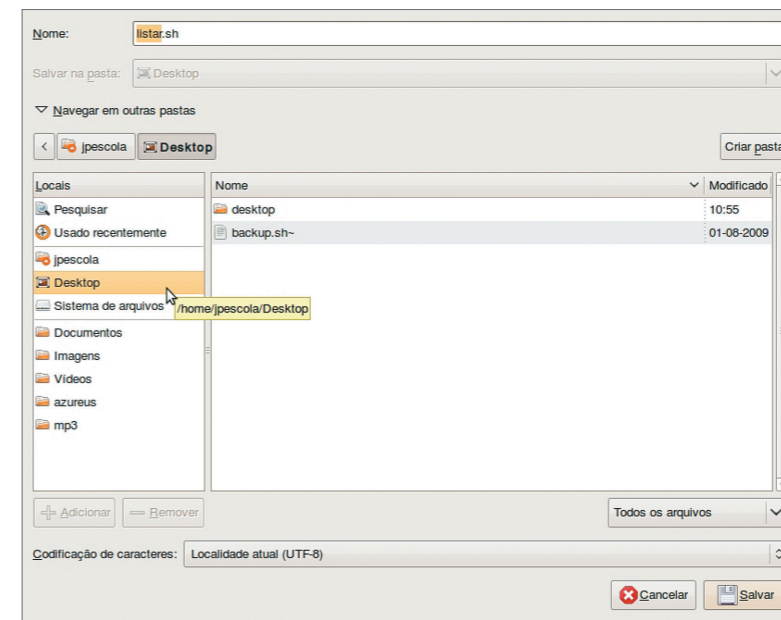
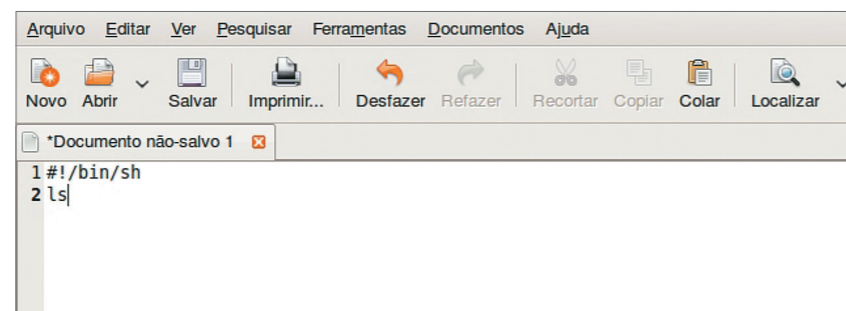
A primeira linha do script deve conter um `#!/bin/sh`, que representa a aplicação a ser utilizada para executar o script. Isso significa que na pasta “bin” existe um comando, chamado “sh”, que será usado para interpretar o script quando você digitar seu nome para ser executado.

Todos os comandos que você conhece para o terminal do Linux podem ser inseridos nos scripts. Então, vamos começar criando um script simples de listagem do conteúdo da pasta atual, como fizemos com o arquivo de lote do Windows.

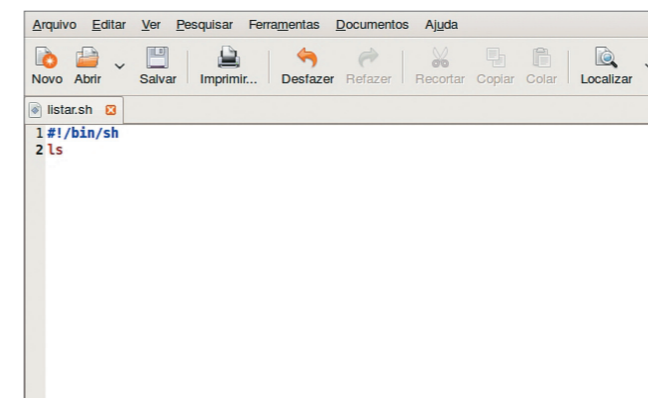
Veja que adicionamos apenas duas linhas de código. Na primeira configuramos o comando que será utilizado para executar o script. Na segunda, o comando “ls”, que lista o conteúdo do diretório atual, como mostra a figura 241. Para salvar o arquivo, clique no botão “Salvar” ou no menu “Arquivo / Salvar”. Será apresentada a tela apresentada na figura 242.

Agora digite o nome do arquivo e escolha a pasta de destino. “Desktop”, como se sabe, representa a área de trabalho do usuário atual. Ao clicar no botão “Salvar”, você verá que o arquivo será criado em sua área de trabalho com um ícone característico do Gnome. Perceba que o Gedit (Editor de Textos) torna coloridos os comandos de acordo com a linguagem que está sendo utilizada. Ele converte os comandos em cores pré-configuradas no momento em que salvamos o arquivo com a extensão “.sh” (figura 243).

**Figura 241**  
Digitando um script no Editor de Textos (Gedit).



**Figura 242**  
Salvando o script na área de trabalho com o nome “listar.sh”.



**Figura 243**  
O gedit aplica cores à sintaxe de acordo com a linguagem correspondente.

O próximo passo é permitir a execução do arquivo. No Linux, isso é necessário porque, por padrão, os arquivos são criados apenas com permissão de leitura e gravação para o usuário. Para tornar o arquivo executável, ou seja, para que o sistema operacional execute os comandos internos do script como se fosse um programa, temos de utilizar o comando **“chmod +x arquivo.sh”**. Após digitar o comando de permissão, vamos executar o script com o comando **“./listar.sh”**. Para isso substituiremos o nome “listar.sh” pelo nome do script que está sendo criado, caso seja escolhido um nome diferente. Lembre-se de que o Linux é case-sensitive. Isso significa que, se você criar um arquivo com o nome “Falar.sh”, não conseguirá executá-lo com o comando “falar.sh”.

“chmod” é o comando que permite alterar as permissões de um arquivo qualquer do Linux; “+x” adiciona permissão de execução no arquivo; e “arquivo.sh” é o nome do arquivo ao qual queremos adicionar permissão de execução.

## 8.5. Arquivos de lote no Linux (estruturas de decisão, laços)

Podemos adicionar, também, estruturas de decisão e laços de repetição em nossos shell scripts. Vamos, então, aprender a fazer isso com exemplos práticos.

**Figura 244**

Adicionando estrutura de decisão.

```

1 #!/bin/sh
2 teste=1
3 if [ $teste = 1 ];then
4     ls
5 else
6     echo "Código incorreto"
7 fi
    
```

**Figura 245**

Utilizando o comando "read".

```

1 #!/bin/sh
2 echo "Digite 1 para listar o diretório atual"
3 read teste
4 if [ $teste = 1 ];then
5     ls
6 else
7     echo "Código incorreto"
8 fi
    
```

**8.5.1. Implementação de estruturas de decisão em scripts**

Se você já programa em alguma linguagem, vai ter facilidade para trabalhar com shell scripts. Entretanto, alguns detalhes precisam ser estudados, pois a sintaxe aqui é um pouco diferente. Um "Se" (estrutura de decisão) é representado pela palavra IF (Se), cuja condição deve ser apresentada entre colchetes, seguido da palavra-chave THEN (então), como ilustra a figura 244.

Veja que criamos uma variável chamada "teste", que recebe inicialmente o valor 1. Faremos a validação dessa variável utilizando a estrutura de decisão IF. Agora vamos pedir ao usuário que digite o código durante a execução do script, utilizando o comando "read" (ler), conforme mostra a figura 245.

Ao executar o script, será solicitado o código para listar o conteúdo do diretório atual. Se o usuário digitar algum código diferente de 1, a mensagem "Código incorreto" será apresentada (ainda a figura 245). Para usar o "senão", utilize "elif", pois esta é a sintaxe correta do shell script.

**8.5.2. Implementação de laços de repetição em scripts**

O script pode permanecer executando indefinidamente até que o usuário pressione um código de saída. Isso permite que ele seja utilizado por tempo indeterminado. Ao terminar o trabalho, você só precisa pressionar 0, por exemplo, para fechá-lo. Para fazer isso, vamos implementar um laço de repetição (figura 246). No primeiro exemplo, criamos um laço infinito que poderá ser finalizado pressionando a tecla "CTRL+C" no teclado.

**Figura 246**

Criando um laço de repetição com o comando "while".

```

1 #!/bin/sh
2 while :
3 do
4     echo "Digite 1 para listar o diretório atual"
5     read teste
6     if [ $teste = 1 ];then
7         ls
8     else
9         echo "Código incorreto"
10    fi
11 done
    
```

Em seguida, adicionamos o comando "while". Assim, enquanto o usuário não pressionar as teclas de finalização "CTRL+C", o script vai continuar em processo de execução. Adicionamos, agora, um comando "elif" para validar também o 2 como uma solicitação de listagem detalhada do diretório no script. Atenção: qualquer código diferente de 1 ou 2 vai resultar numa mensagem de erro pelo script (figura 247).

Em vez de pressionar CTRL+C para finalizar o script, é possível fazer uma validação melhor, solicitando a tecla 0 para sair, por exemplo. É importante frisar que a linguagem "Shell Script" é extremamente poderosa. É aconselhável aprofundar-se no tema consultando materiais específicos.

**Figura 247**

Adicionando uma condição ao laço "while".

```

1 #!/bin/sh
2 while :
3 do
4     echo "Digite 1 para listar o diretório atual\n0 2 para
5     uma listagem detalhada..."
6     read teste
7     if [ $teste = 1 ];then
8         ls
9     elif [ $teste = 2 ]; then
10        ls -lh
11    else
12        echo "Código incorreto"
13    fi
14 done
    
```