
Sistema Operacional Linux

Rafael Beserra Gomes, Universidade Federal do Rio Grande do Norte

Uma vez que você praticará os algoritmos em computadores, conheça um pouco mais sobre o sistema operacional que será utilizado na disciplina: o Linux. Este texto contém somente o conhecimento básico para o início da disciplina. Você pode aprender bem mais sobre Linux com alguns tutoriais online. Uma sugestão é o guia Foca (http://www.guiafoca.org/?page_id=14).

1 A organização dos arquivos no Linux

Os arquivos no Linux são dispostos em diretórios (ou pastas¹). Há um motivo claro para isso: se todos os arquivos estivessem em um único local seria difícil para os usuários gerenciar uma quantidade maior de arquivos. Você provavelmente deve estar acostumado a colocar os seus arquivos de música em uma pasta denominada Música e os arquivos de texto em uma pasta denominada Documentos. Isso não impede que você possa colocar uma música na pasta Documentos ou um arquivo de texto na pasta Música – mas poderá ter problemas futuros com esse tipo de organização.

No Linux, todos os diretórios começam com /, o qual é chamado de diretório raiz. Nesse diretório há subdiretórios, como /bin, /usr, /tmp e /home, cada um com seu propósito na organização dos arquivos pessoais dos usuários e os arquivos do sistema.

¹o termo *pasta* também é bastante utilizado, mas há uma diferença conceitual – ainda assim, para os propósitos do curso, você pode considerá-los como sinônimos

O diretório `/home` é o de nosso maior interesse pois é o diretório que armazena os arquivos pessoais dos usuários. Se o seu usuário for, por exemplo, *maria*, então existirá um diretório `/home/maria`. Esse diretório que pertence ao usuário é também chamado de **home do usuário** ou **pasta pessoal**.

Através de uma interface gráfica do Linux é possível navegar nos diretórios através de cliques, entretanto será comum no curso especificar um arquivo ou diretório de forma textual a partir de um diretório qualquer.

1.1 Caminho completo e relativo

Os arquivos e diretórios podem ser especificados pelo caminho completo ou pelo caminho relativo. No primeiro especificamos a localização desde a raiz. Por exemplo, suponha que há um diretório chamado `documentos` na pasta pessoal do usuário *maria* e nesse diretório há um arquivo `relatorio.doc`. Então o caminho completo do arquivo é `/home/maria/documentos/relatorio.doc`. Já no segundo a localização é expressa em função do diretório atual. Por exemplo, se estivermos no diretório `/home/maria` então o caminho relativo do arquivo **relatorio.doc** é `documentos/relatorio.doc`; se estivermos em `/home` então o caminho relativo desse arquivo é `maria/documentos/relatorio.doc`.

Se quisermos nos referir ao diretório imediatamente acima do atual em um caminho relativo, isto é, o diretório que contém o atual, utilizamos `..`. Por exemplo, se estivermos no diretório `/home/maria/documentos/outroDiretorio`, então podemos nos referir àquele arquivo com `../relatorio.doc`. Mais um exemplo: se o diretório atual for `outroDiretorio` e houver um arquivo chamado `imagem.jpg` em `/home/maria/` então este poderia ser especificado com `../../imagem.jpg`.


2 O terminal do Linux

A nossa interação com o sistema operacional é feita através de um *shell*.

Para utilizar o Linux não é necessário uma interface gráfica, embora esta facilite bastante o seu uso. Ao utilizar o Linux do laboratório, após o *login*, você interagirá com o sistema através de um *shell* gráfico. Mas é possível por essa interface gráfica abrir um **terminal** com um outro *shell*: o *bash*.

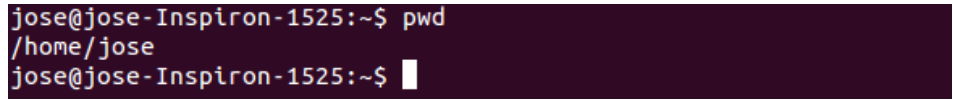
O *bash* é um *shell* por linha de comando e será bastante útil no decorrer da disciplina. A Figura 1 exemplifica o que é exibido ao abrir o *bash*. Geralmente o que é exibido é: o usuário, um `@`, o nome do computador, um `:`, um `~` e um `$`. O `~` significa que o diretório atual é a *home*, ou seja, a sua **pasta pessoal**.

Se você digitar *pwd*, será exibido o **caminho completo** do diretório atual, conforme exemplifica a Figura 2.



```
jose@jose-Inspiron-1525:~$
```

Figure 1: *Bash: um shell por linha de comando.*

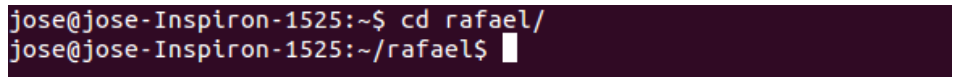


```
jose@jose-Inspiron-1525:~$ pwd
/home/jose
jose@jose-Inspiron-1525:~$
```

Figure 2: *pwd: exibe o caminho completo do diretório atual.*

2.1 Navegando nos diretórios

Para alterar o diretório atual, utilize o comando **cd** (de *change directory*) especificando a seguir o caminho completo ou relativo do diretório para o qual você quer mudar. Veja na Figura 3 um exemplo. Se você omitir o diretório, o diretório é alterado para a sua **pasta pessoal** (vide Figura 4).



```
jose@jose-Inspiron-1525:~$ cd raphael/
jose@jose-Inspiron-1525:~/raphael$
```

Figure 3: *cd: altera o diretório atual. Observe que entre : e \$ aparece o caminho completo, onde a pasta pessoal está abreviada por ~*

Um detalhe que você deve prestar atenção é que se o diretório contiver espaços, estes devem ser expressos por `\espaço`. Por isso, a sugestão é evitar nomes de diretórios com espaços. Veja a seguir outros comandos úteis do *bash*.

2.2 Outros comandos úteis de bash

- *ls*: lista os arquivos da pasta atual;
- *mkdir dir*: cria um diretório chamado *dir*; evite espaços e caracteres especiais;
- *cat arquivo*: exibe o arquivo na tela em modo texto;
- *rm arquivo*: apaga o arquivo
- *wget url*: baixa o arquivo que está na URL especificada.

```
jose@jose-Inspiron-1525:~/rafael/pdfs$ cd
jose@jose-Inspiron-1525:~$ pwd
/home/jose
jose@jose-Inspiron-1525:~$
```

Figure 4: *cd: altera o diretório atual.*

2.3 Auto-completar e histórico de comandos

Dois recursos bastante úteis do *bash* é o auto-completar e o histórico de comandos. Para utilizar o auto-completar, use da tecla *tab*. Para navegar no histórico de comandos, utilize a seta para cima ou para baixo. Experimente!

2.4 Abrindo programas via terminal

Especificamente no Ubuntu podemos abrir os programas clicando no primeiro ícone da barra e digitando no campo de busca o nome do programa que desejamos abrir (vide Figura 5).

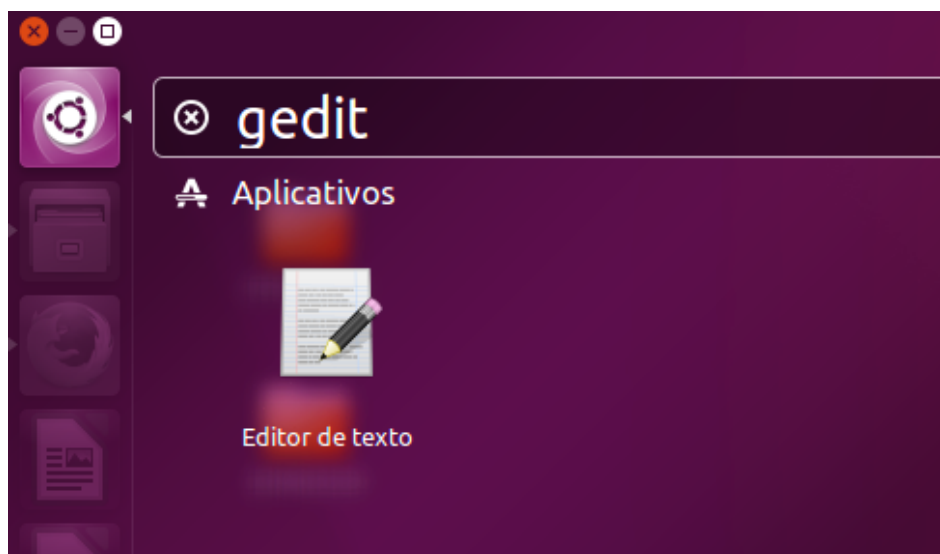


Figure 5: *Abrindo um programa (gedit no exemplo) via interface gráfica.*

Outra opção é abrir no *bash* digitando o nome do programa. Nesse caso há ainda duas opções. Na primeira opção (Figura 6) basta digitar o nome do programa e apertar enter. Nesse caso, você somente terá controle novamente do *bash* quando o programa aberto encerrar. Na segunda opção (Figura 7) digite

também `&` ao final e aperte enter. Nesse caso, o programa é aberto em segundo plano e você já passa a ter controle de volta do *bash*.

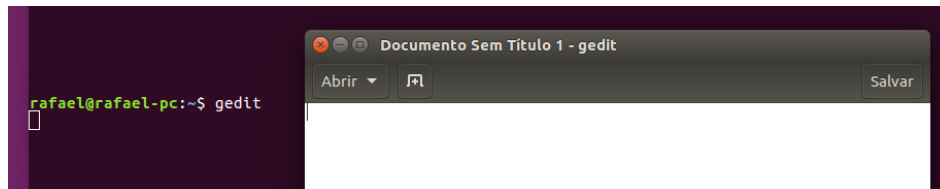


Figure 6: *Abrindo um programa (gedit no exemplo) via bash.*

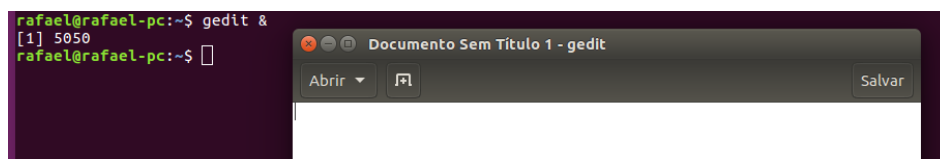


Figure 7: *Abrindo um programa (gedit no exemplo) via bash.*