

Universidade Federal do Rio Grande do Norte  
Departamento de Física Teórica e Experimental  
Disciplina: Física Computacional II (2012.2)  
Linux - Comandos Básicos

Nesse curso, iremos utilizar o sistema operacional Linux para o desenvolvimento de nossas atividades. Como é um sistema novo para a maioria dos alunos, iremos apresentar aqui um resumo dos comandos básicos que serão necessários para as atividades mais básicas. Veja que nossa abordagem aqui é fornecer apenas os comandos mais básicos e não pretende ser um curso de Linux. Existe muito material na internet, além de livros. Por exemplo, acesse <http://wiki.ubuntu-br.org/ComandosBasicos> para uma lista de comandos mais extensa.

## 1 Manipulação de arquivos

O sistema operacional Linux se baseia fortemente no seu sistema de arquivos. Praticamente, tudo em Linux está montado nessa estrutura. Assim, é importante aprender a manuseá-la. Existem algumas maneiras de fazer isso. Você pode utilizar alguma software gráfico, como o gerenciador de arquivos da sua interface gráfica, ou a linha de comandos. Nesse pequeno tutorial, iremos mostrar a utilização da linha de comandos.

Basicamente, você precisa realizar operações de listagem, criação, alteração e navegação de arquivos. A estrutura de arquivos no Linux inicia a partir do chamado diretório raiz (“/”). Todos os arquivos e diretórios podem ser referenciados a partir da raiz, utilizando o caminho completo, ex: /home/joao/Desktop, ou um caminho relativo, ex: Desktop. Nesse último caso, o diretório (arquivo) Desktop deve estar imediatamente abaixo do diretório corrente.

Para ilustrar melhor as operações básicas, vamos fazer isso através de exemplos. Para isso, nossa primeira tarefa é abrir um terminal para digitar comandos. Isso pode ser feito de várias maneiras e pode variar entre as diversas interfaces gráficas disponíveis. Se estiver usando o Ubuntu, você pode pressionar as teclas Alt+F2 ao mesmo tempo (se não funcionar, faça uma busca com os termos linux abrir terminal).

```
joao@ubuntu12:~$
```

Deve abrir uma tela com uma linha parecida com a acima. Vamos detalhar as informações mostradas na linha.

- joao : é o nome do usuário está usando o terminal no momento.
- @ : é apenas um separador.
- ubuntu12 : é o nome da máquina onde o terminal está sendo executado.

- `~`: é um mnemônico para o diretório padrão do usuário.
- `$` : Também é um separador, os caracteres após ele, normalmente são comandos.

Claro que essas informações vão mudar de acordo com seu usuário e máquina. A linha acima é chamada prompt de comandos. Os comandos são digitados sempre após o prompt de comandos e são normalmente programas, que podem ser comandos externos e internos.

Vamos agora aos comando básicos. Nas instruções a seguir, os comandos estarão logo após o prompt de comandos e as linhas que aparecem abaixo dele são as saídas dos comandos.

## Comando pwd

Para saber em que diretório você está, execute o comando `pwd` abaixo

```
joao@ubuntu12:~$ pwd
/home/joao
```

O comando acima mostra que você está atualmente no diretório `/home/joao`. Disso podemos concluir também que o diretório padrão do usuário `joao` é `/home/joao`

## Comando ls

Para listar o conteúdo do diretório corrente, utilize o comando `ls`

```
joao@ubuntu12:~$ ls
Área de Trabalho      core      QtSDK
books                  gj         temp
```

No caso acima, vemos que existem alguns arquivos. Para ver uma listagem com mais informações utilize

```
joao@ubuntu12:~$ ls -l
total 108652
drwxr-xr-x  2 joao joao    4096 Abr 30 22:05 Área de Trabalho
drwxrwxr-x  2 joao joao    4096 Jul 22 11:44 books
-rw-----  1 joao joao 30855168 Mai 22 21:29 core
drwxrwxr-x  3 joao joao    4096 Mai  1 10:34 gj
drwxrwxr-x 13 joao joao    4096 Mai 22 21:44 QtSDK
drwxrwxr-x  3 joao joao    4096 Mai  1 11:45 temp
```

O comando acima, lista cada uma das entradas do diretório em uma linha. Essa linha, por sua vez, traz várias informações sobre aquela entrada. O primeiro conjunto de caracteres é `drwxr-xr-x` que nos informa o tipo daquela entrada através da primeira letra “d” que nesse caso é um diretório e as permissões do usuário, do grupo e do restante além do usuário dono do arquivo, tamanho, data da última alteração e nome do arquivo/diretório.

Por ora, o importante conhecermos apenas algumas informações, vamos identificá-las para a primeira e terceira linhas. Você pode inferir para as linhas seguintes.

Para a primeira linha

- Tipo de arquivo: diretório
- Dono do arquivo: joao
- Grupo do arquivo: joao
- Data/Hora da última alteração ou criação: 30/04 do ano corrente às 22:05
- Nome: Área de Trabalho

Para a terceira linha

- Tipo de arquivo: arquivo
- Dono do arquivo: joao
- Grupo do arquivo: joao
- Tamanho do arquivo: 30855168 bytes
- Data/Hora da última alteração ou criação: 22/05 do ano corrente às 21:29
- Nome: core

A sopa de letrinhas drwxr-xr-x nos dá informações sobre as permissões da entrada. A primeira letra pode ser “d” para um diretório ou “-” para um arquivo regular (existem outras possibilidades, mas não vamos explorá-las aqui). Os outro nove caracteres representam as permissões para o usuário (três primeiras), grupo (do quarto até o sexto) e outros (do sétimo até o nono), onde

- r : leitura habilitada
- w : gravação habilitada
- x : execução habilitada (no caso de arquivos regulares) e entrada não habilitada (no caso de diretórios)
- - : não habilitada.

Assim, a sequência drwxr-xr-x, nos diz que a entrada correspondente é um diretório, o usuário dono pode ler, gravar e entrar no diretório. O grupo pode ler, não pode gravar e pode entrar no diretório, os outros podem ler, não podem gravar e podem entrar no diretório.

## Comando mkdir

Para criar diretórios utilizamos o comando mkdir. Por exemplo, o diretório fc2012:

```
joao@ubuntu12:~$ mkdir fc2012
```

Veja que no exemplo acima utilizamos o caminho relativo, de maneira que o diretório será criado imediatamente abaixo do diretório corrente.

## Comando cd

Para mudar para algum diretório você deve usar o comando `cd`. Por exemplo, para mudar para o diretório criado no exemplo anterior, utilize

```
joao@ubuntu12:~$ cd fc2012
joao@ubuntu12:~/fc2012$
```

Observe que o prompt de comandos mudou.

Para retornar para um diretório acima utilize

```
joao@ubuntu12:~/fc2012$ cd ..
joao@ubuntu12:~$
```

Para retornar para o diretório padrão (HOME), esteja onde estiver, digite simplesmente `cd` e dê enter. No exemplo abaixo, mudamos para o diretório `/var/log` e voltamos ao diretório padrão com o comando `cd`

```
joao@ubuntu12:~$ cd /var/log
joao@ubuntu12:/var/log$ cd
joao@ubuntu12:~$
```

Existem ainda comandos para remover, copiar, listar conteúdo de arquivos, etc. Mas não iremos explorá-los aqui, acesse <http://wiki.ubuntu-br.org/ComandosBasicos> para uma lista mais completa.

## 2 Compilação e execução de programas

Outras atividades que faremos com frequência são a edição, compilação e execução de arquivos. Nessa seção, iremos mostrar como editar, compilar e executar um programa. Vamos criar um primeiro programa para exibir apenas uma mensagem na tela do computador, o objetivo é saber a sequência de passos que utilizaremos. Para programas mais complicados, a sequência de passos é basicamente a mesma, o que muda é apenas o programa fonte.

Nossa sequência consistirá do seguinte:

1. criar um diretório de trabalho.
2. navegar até o diretório de trabalho.
3. criar um arquivo contendo o código fonte do nosso programa.
4. compilar o código fonte.
5. executar o programa e observar seu resultado.

Vamos agora detalhar a sequência acima, veja que você deve abrir um terminal para poder executar os comandos mostrados abaixo

1. criar um diretório de trabalho, criaremos o fc2012.

Se você acabou de abrir um terminal, já estará no seu diretório \$HOME, caso contrário execute um `cd <enter>` para ter certeza que executará a os comandos abaixo a partir do \$HOME. Pode ser que você já tenha o diretório fc2012, caso contrário execute

```
joao@ubuntu12:~$ mkdir fc2012
joao@ubuntu12:~$
```

2. navegar até o diretório de trabalho. Um recurso interessante do Linux (na verdade do \$SHELL que usamos o BASH) é a completção de comandos, você digita o comando até um certo ponto e pressiona a tecla TAB algumas vezes para ver os possíveis complementos. Experimente, no exemplo abaixo, digitar `cd fc ;TAB;` para ver o que acontece

```
joao@ubuntu12:~$ cd fc2012
joao@ubuntu12:~/fc201
```

3. criar um arquivo contendo o código fonte do nosso programa.

Para isso, você pode usar qualquer editor de arquivos de sua escolhas. Vamos usar o gedit, ele é um editor semelhante ao notepad do Windows, com mais recursos. Digite

```
joao@ubuntu12:~/fc2012$ gedit hello.c &
[1] 4969
joao@ubuntu12:~/fc2012$
```

o “&” acima serve para deixar o programa executando em segundo plano e liberar o terminal.

Insira o conteúdo abaixo e salve o programa

```
#include "stdlib.h"
#include "stdio.h"

main() {
    printf("hello world\n");
}
```

4. compilar o código fonte. Como estamos escrevendo um programa em C, utilizamos o comando abaixo

```
joao@ubuntu12:~/fc2012$ gcc -o hello hello.c -lm
joao@ubuntu12:~/fc2012$
```

A opção `-o hello`, informa que iremos criar um programa executável chamado hello, o trecho `hello.c` é o nome do nosso arquivo fonte, a opção `-lm` indica que iremos usar a biblioteca matemática padrão (essa opção não

seria necessária aqui, mas ao longo do nosso curso iremos precisar dela de modo que é interessante compilar nossos programas com essa opção).

Nesse momento é normal ocorrerem erros de compilação, nesses casos a própria mensagem de erro dá pistas de onde está ocorrendo o erro. Veja o número de linha e o erro e tente consertá-lo no seu código fonte.

5. executar o programa e observar seu resultado. Para executar o programa compilado e ver seu resultado, simplesmente execute:

```
joao@ubuntu12:~/fc2012$ ./hello
hello world
joao@ubuntu12:~/fc2012$
```

Nesse tutorial aprendemos os comandos básicos necessários para criar um diretório, navegar para ele, criar um código fonte, compilá-lo e executá-lo. Tudo isso usando o sistema operacional Linux. Mais adiante, iremos explorar um pouco mais os recursos do sistema operacional.