

## LAB 0: Introdução ao ambiente de trabalho | Introduction to linux environment.

OBJETIVO: SABER COMPILAR E FAZER A LINKAGEM DE PROGRAMAS C EM AMBIENTE LINUX.

De que forma posso compilar e fazer a linkagem em Linux? Como interagir com o terminal de comandos? O que é gdb? Como utilizar o ddd? O que é o make? O que é uma aplicação de gestão de versões (git)? Como colocar, atualizar e recuperar o código no github?

GOAL: TO KNOW HOW TO COMPILE AND LINK IN LINUX ENVIRONMENT.

How can we compile and link a program? How to interact with the terminal? How to use gdb and ddd? How can I use 'make'? How can I use github to keep my code and manage its versions?

Guião de apoio ao laboratório de introdução no ambiente de trabalho e de teste em programas em C.

Para apoio à interação com Linux, existem vários documentos disponíveis, pequenos manuais e o livro "How Linux Works".

## PARTE I: Usar Linux | PART I: Using Linux

### 1. Instalar o LINUX numa máquina virtual

Este passo deverá ser dado para quem pretende instalar o Linux no seu computador pessoal (em alternativa poderá instalar o linux em *dual boot* ([ver aqui](#))).

- a. Instale o Oracle VM VirtualBox
- b. Descarregue a imagem (OVA ) OSC-2016 fornecida pelo Professor.
- c. Dentro da janela do Oracle VM VirtualBox, deverá atribuir um nome e definir o tipo de sistema operativo que no caso é Linux, versão Debian de 32-bit.
- d. A imagem carregada tem apenas um utilizador, o **OSCReader**. Deve clicar sobre esse utilizador (o do seu ambiente de trabalho) e depois introduzir a palavra passe **osc**.
- e. Para partilhar uma pasta entre o **SO** hóspede e o **SO** convidado, deverá seguir as instruções indicadas na página do "Oracle VM VirtualBox" e pedir ajuda ao Professor.

## 2. Novo utilizador

Ao iniciar o sistema operativo Linux do laboratório, fica associado a um utilizador genérico. Deve criar um utilizador próprio para poder sempre trabalhar no mesmo ambiente de trabalho. Para isso, deverá escolher como utilizador o seu primeiro e último nome separado por um ifen – por exemplo, *jose-cascalho*. | When starting a LINUX OS in the lab, you enter in a generic user. We must create a unique user to have always the same desktop. To do it, we must select one user name - e.g. first and last name connected by an ifen like jose-cascalho.

Utilize a interface através das janelas disponibilizada pelo GNOME – sistema de janelas do linux instalada no *Ubuntu*. Seleccione os *settings* no espaço de aplicações disponíveis. Procure *Users* e adicione um novo utilizador. Não se esqueça de seleccionar uma *password*. | Using the windows interface, select settings in the space of applications. Search for Users and add a new user. Don't forget to add a password.

Abra a janela, no topo do canto direito do ecrã. Escolha *Switch User* ou *Log Out* e entre como com o novo utilizador. | Open the window at the to right of the screen and select Switch User or Log Out.

No terminal de comandos execute a instrução: `gsettings set org.gnome.desktop.background show-desktop-icons true`. Esta instrução vai permitir ver no Desktop, os icons dos documentos que vai criando nessa diretoria. | In the terminal execute the command `gsettings set org.gnome.desktop.background show-desktop-icons true`. This command will show in the screen the icons created in the Desktop.

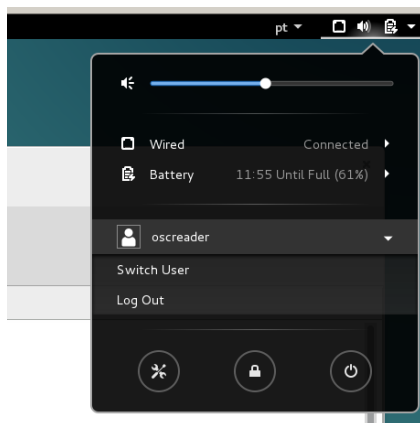


fig.1: Janela para adicionar novo utilizador.

## 3. Interagir com o LINUX através do TERMINAL de COMANDOS

### 3.1 Identificar a shell

No LINUX, a emulação do que era um antigo terminal de comandos ainda está presente na atividade de interação com o computador (assim como no Windows).

A linguagem de interação depende da *shell* utilizada.

Coloque o comando “echo \$SHELL” para obter informação da shell que utiliza.

### 3.2 Ficheiro de configuração da shell partindo do comando *.bashrc*

Qualquer shell tem um conjunto de ficheiros de configuração. No LINUX, os ficheiros que se iniciam com ‘.’ estão escondidos.

Só são visíveis quando usamos o comando ls com a opção -a (all): `ls -a`. Execute os comandos:

- a) `cd $home`.
- b) `ls -a`
- c) `cat .bashrc`

### 3.3. Modo de administrador (*superuser*) e fazer *shutdown* da máquina.

O UNIX tem duas formas de interagir com o utilizador. Uma como administrador que tem acesso a comandos de gestão da máquina. A outra como simples utilizador. | UNIX has two different ways to interact with user through shell commands, as *superuser* which has access to commands for OS management and the other as simple user.

3.3.1 No terminal escreva `su` de “*superuser*”. Introduza a palavra chave. Passou para o modo de administrador. Verá alterações na indicação de linha do cursor com a passagem do sinal \$ para # no “*bash shell*”. | Use the command `su` to change to *superuser*. The command line has now the signal # meaning that you are now has an administrator profile.

3.3.3 Escreva `exit` para sair de modo administrador. | You leave administrator profile using `exit`.

3.3.4 Escreva `shutdown -h now`. Uma informação no ecrã dir-lhe-á que não existe esse comando. | Write `shutdown -h now`. The shell informs you that there is no such command.

3.3.5 Experimente agora passar novamente para administrador e experimente novamente a instrução de encerramento da máquina. O que acontece? | Enter again as administrator and execute the last instruction. What happens?

### 3.4 Editar um texto em Linux a partir da SHELL

Existem muitos editores de texto em LINUX que trabalho diretamente com a shell como por exemplo o *nano*. Experimente o *nano*, executando o comando na shell: `nano calculadora.c` | There are plenty of text editors for LINUX that work under the shell. The one most used is *nano*. Try it editing the file `calculadora.c`, by executing the command `nano calculadora.c`.

Iremos trabalhar com o editor *gedit* inicialmente. Mais para a frente, podemos utilizar o *bluefish*. O primeiro editor é muito simples. Experimente o *gedit* utilizando o comando na shell: `gedit calculador.c` | Most of the time we'll using *gedit*. *Bluefish* will be used later on, with more complex C programs. Try *gedit* using the command `gedit`.

### 3.5 Instalar compilador e compilar um ficheiro utilizando o comando ‘gcc’

Para instalar software no LINUX, existem formas expeditas que permitem controlar as versões instaladas dos pacotes. Elas procuram na base de dados se existe determinado software e depois é testam se é possível instalá-lo. | To install software in LINUX there are simple ways using commands that control the versions installed. They search in a data base if there is a certain software and then test if it is possible to install it.

Usa os seguintes comandos para instalar o compilador gcc | Use the following commands to install the gcc compiler.

```
apt-get update
apt-get search gcc
apt-get install gcc
```

Teste o compilador, executando a compilação do ficheiro `calculador.c`. Depois do comando, um ficheiro executável aparece na listagem de comandos. | Test the gcc compiler with the file `calculador.c`. After the command a new file executable will be present in the list of files at the directory.

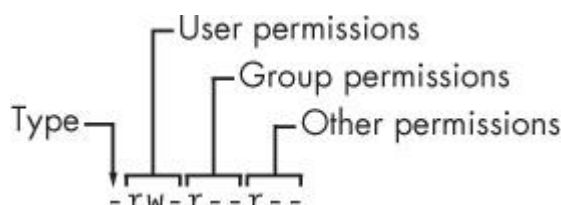
```
gcc -o calculador calculador.c
ls -l
```

Execute o programa da seguinte forma: | Execute the program as following:

```
calculador +
```

### 3.6. Permissão dos ficheiros | Files permissions

Todos os ficheiros em UNIX têm um conjunto de permissões que determinam se podem ser lidos, modificados ou se são executáveis. O comando `ls -l` apresenta essas permissões. | Every Unix file has a set of permissions that determine whether you can read, write, or run the file. Running `ls -l` displays the permissions.



**fig 2. Permissão dos ficheiros | Files permission**

Ao executar o comando `ls -l calculadora.c` surge no ecrã uma informação semelhante a esta:

```
-rw-r--r-- 1 jmc jmc 1042 Mar 2 2017 calculadora.c
```

Comparando com a figura 2, pode concluir-se que o ficheiro em questão pode ser lido e modificado pelo utilizador e apenas lido por utilizadores do mesmo grupo (utilize o comando `groups` para ver a que grupos pertence). Finalmente, qualquer outro utilizador pode ler o ficheiro.

| The user permissions pertain to the user who owns the file. In the preceding example, that's *jmc*. The second set, group permissions, are for the file's group (somegroup in the example). Any user in that group can take advantage of these permissions. Finally, every user can read the file.

Para modificar permissões utiliza-se o comando `chmod`. Primeiro, selecciona-se o tipo de permissões que se pretende alterar e depois dá-se instruções para adicionar o remover essas permissões. As letras estão associadas a *group*(g), *world*(o) ou *user*(u). | To change permissions, use the `chmod` command. First, pick the set of permissions that you want to change, and then pick the bit to change. For example, to add group (g) and world (o, for "other") read (r) permissions

to file, you could run the following commands.

```
chmod g+w calculadora.c
```

```
chmod o+w calculadora.c
```

Podemos alterar ambas as permissões | Or you could do it all in one shot:

```
chmod go+r file
```

Para remover permissões, utiliza-se o sinal - | To remove these permissions, use `go-r` instead of `go+r`.

```
chmod go-r file
```

Por vezes utiliza-se o formato de permissão com modos absolutos listado na figura 3.

Mode	Meaning	Used For
644	user: read/write; group, other: read	files
600	user: read/write; group, other: none	files
755	user: read/write/execute; group, other: read/execute	directories, programs
700	user: read/write/execute; group, other: none	directories, programs
711	user: read/write/execute; group, other: execute	directories

**fig.3.** Modos associados à mudança de permissão. O comando `chmod 644 file` executa o mesmo que `chmod u+rw go+r file`.

As diretorias também têm permissões. Uma diretoria só pode listar o seu conteúdo se for possível lê-la. Mas apenas se consegue aceder a um ficheiro numa diretoria se ela for executável. Um erro habitual é retirar a permissão de execução das diretorias utilizando os modos. | Directories also have permissions. You can list the contents of a directory if it's readable, but you can only access a file in a directory if the directory is executable. One common mistake people make when setting the permissions of directories is to accidentally remove the execute permission when using absolute modes.

### 3.7 Ajuda nos comandos na SHELL

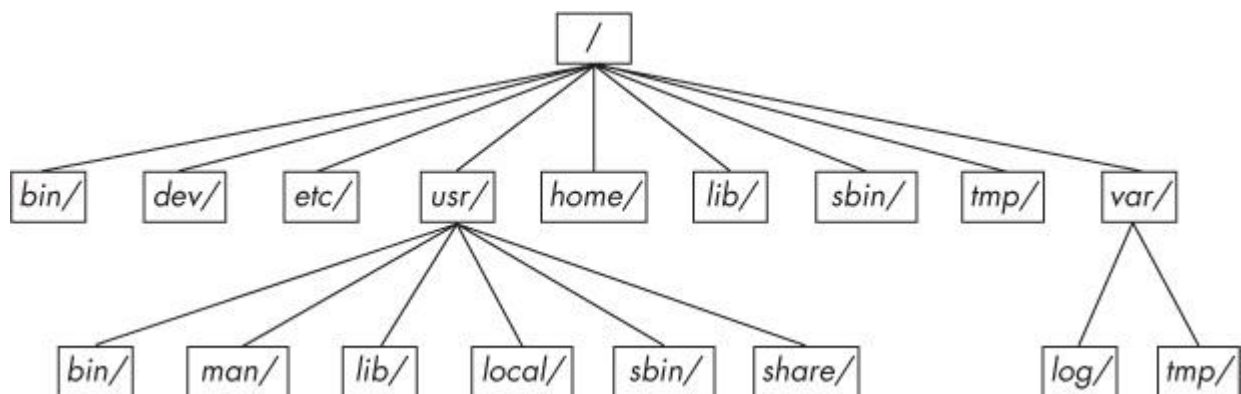
O manual de LINUX associado aos comandos (man) permite aceder a instruções de diferentes comandos. Utilize-o para identificar os seguintes comandos. | LINUX man command have informations about the different commands interface for the shell in LINUX. Use the command to identify each one of the following commands.

- cd ( cd .. e cd /)
- pwd
- mkdir
- rmdir
- rm
- mv
- ls (-l -a -la -lha)

Na listagem de ajuda existe habitualmente um comando help. Experimente obter ajuda de determinado comando executando instruções do tipo cd --help para a listagem de comandos apresentada. | You can also use an help command that usually is called using --help, e.g. cd --help.

## 4. Árvore de diretorias em LINUX | Linux Directory Hierarchy Essentials

Os detalhes da diretoria são apresentados na figura 4. Pode ser consultado mais detalhe sobre esta hierarquia em <http://www.pathname.com/fhs/>. Reparar que é por baixo de home/ que surgem as diretorias associadas aos utilizadores do LINUX. | The details of the Linux directory structure are outlined in the Filesystem Hierarchy Standard, or FHS (<http://www.pathname.com/fhs/>), but a brief walkthrough should suffice for now. Figure 4 offers a simplified overview of the hierarchy, showing some of the directories under /, /usr, and /var. Notice that the directory structure under /usr contains some of the same directory names as . The directory /home is the directory that receives the users of the system.



## 5. Exercícios

**5.1** Crie uma pasta designada por *exercicios\_C*. Mude-se para essa pasta. Nessa pasta crie um programa chamado *primeiro.c* utilizando o gedit. Introduza o seguinte programa: | Create the directory *exercicios\_C*. In this directory create the program *primeiro.c* using gedit. Add the following program:

```
/* A first program in C */
main()
{
    printf("Welcome to C!");
}
```

Faça a compilação e linkagem desse programa utilizando o gcc. Execute o programa. | Compile and execute the program.

**5.2** Crie uma nova pasta designada por *primeiro*. Passe o ficheiro para essa pasta. Elimine os ficheiros que estão na pasta *exercicios\_C*. | Create a new directory below the one already created, e dubbed *primeiro*. Move the file to this new directory. Eliminate the files from the present directory.

**5.3** Mude as permissões do ficheiro *primeiro.c* de forma a que não seja possível modificá-lo. Após essa alteração tente modificá-lo utilizando o editor nano. | Change the permissions of the file *primeiro.c* to unreadable. Try to modify it after that using nano text editor.

**5.4** Volte a colocar as permissões para tornar o ficheiro *primeiro.c* modificável. Nessa diretoria faça novamente a compilação do programa. Liste o programa utilizando o comando 'ls -a'. Verifique que o programa é executável. Retire a permissão para a sua execução e teste se é possível executá-lo dessa forma. | Put back the initial permissions to the file *primeiro.c*. In its directory compile again the program. List the files in the directory using 'ls -a' and test if the program has the permission to be executable. Change its permissions to remove that permission. Try to run the program.