

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?

1. Executar um programa em C e usar o depurador ddd. | Executing a C program and using the ddd debugger.

1.1 Executar um programa em C | Executing a program in C

No terminal utilizamos o comando 'gcc -c name-of-file -lm' para compilar o programa (sem linkagem). Ao fazer ls -l vai descobrir que existe um ficheiro objeto com extensão '.o' pronto para a linkagem. A operação de compilação e linkagem ou só linkagem faz-se com o comando 'gcc -o name-of-file name-of-file.c'. Faça a compilação e execute o programa 'calculadora.c'. Note que o parâmetro '-lm' permite incluir a biblioteca de matemática. | In the terminal 'gcc -c name-of-file -lm' compiles the program without linkage. Use ls -l to see the new object file with '.o' extension created in the directory | To get an executable do 'gcc -o name-of-file name-of-file.c'. Note that '-lm' parameter is used to include math library.

1.2 Instalar o depurador gdb e a interface para depurador ddd

No terminal de comandos vai instalar novo software. Em primeiro lugar terá de instalar uma biblioteca que apoia a execução do depurador da GNOME: | In the terminal you are going to install a new software but first you need to install the following library:

sudo aptitude install libc6-dev [ou sudo apt-get install libc6-dev]

Se tudo correu bem, instale agora o gdb e o ddd: | Please install gdb and ddd:

- Procure pelo nome do pacote gdb: *sudo aptitude search gdb*
- Instale o pacote com o nome certo. | Install the package with the name found.
- Procure pelo nome do pacote ddd: *sudo aptitude search ddd*
- Instale o pacote com o nome certo. | Install the package with the name found.

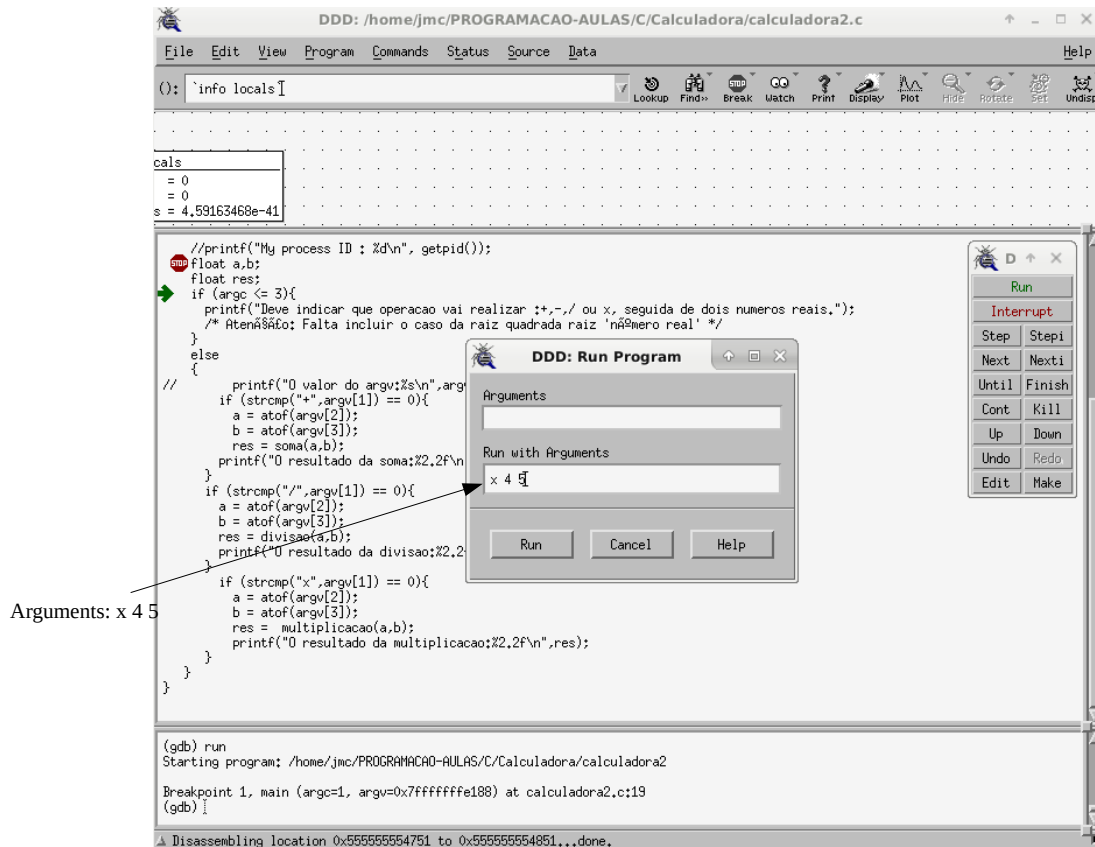
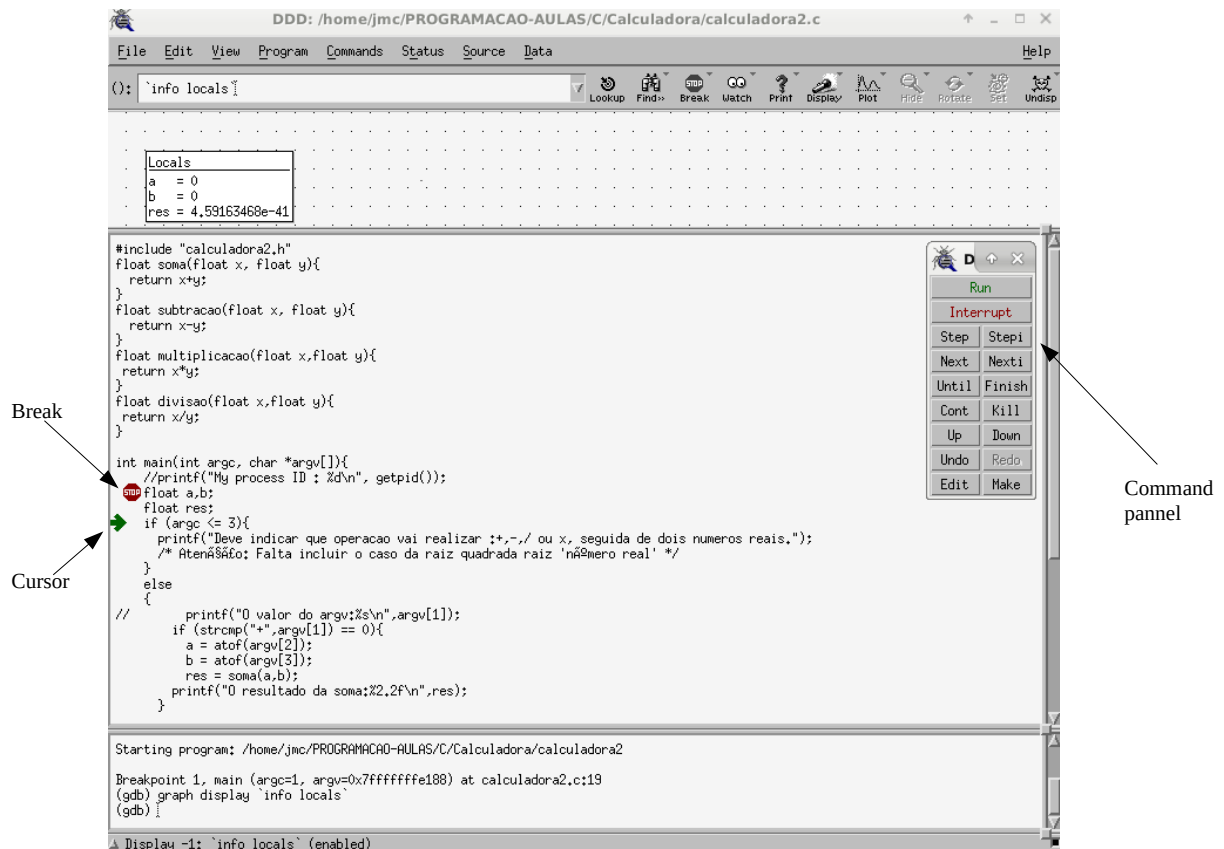
1.3 Testar o depurador | To test the debugger.

Para utilizar o depurador, terá de compilar o programa utilizando o parâmetro '-g'. | To use the debugger you have to compile using '-g'.

Descarregue o programa 'calculadora2.c' e execute: | Compile using the command and execute the debugger:

gcc -g -o calculadora2 calculadora2.c -lm

ddd calculadora2



2. Utilizar a ferramenta make para compilar um programa | Using make to compile a program.

No LINUX bem como noutros sistemas operativos o processo de compilação e linkagem é, já há bastante tempo, apoiado por ferramentas de software que os tornam automáticos. Neste ponto apenas mostramos a possibilidade de um deles, o make. Para mais detalhes, poderá consultar o manual respetivo (<https://www.gnu.org/software/make/>).

| In the LINUX as well as in other OS, the compilation and linkage process can be automatic if supported by software tools like the *make* tool. (<https://www.gnu.org/software/make/>).

2.1 Considere este pequeno programa guardado no ficheiro *makefile* na diretoria Calculadora onde estão os outros dois ficheiros. | Consider the makefile and add it to the Calculadora directory where the other two files must be.

O formato de cada linha é: | The format of each line is:

```
target: dependencies
      command 1
      command 2
      ...
```

target corresponde ao ficheiro que é criado com os comandos. As dependências – *dependencies* – identificam quais os ficheiros que têm de existir para ser executado os comandos. | *target* is the file to be created by the commands. The *dependencies* identify which files must exist to execute the commands.

```
#Simple example of makefile
calculadora: calculadora.o
      gcc -o calculadora $(objects) -lm
calculadora.o: calculadora.c calculadora.h
      gcc -c calculadora.c
clean:
      rm calculadora $(objects)
```

fig.1: Exemplo de makefile.

Para testar este pequeno programa, executa as instruções: | Use the following instructions to test the make:

```
make
make clean
```

2.2 O ficheiro seguinte utiliza o programa calculadora2. Crie o *makefile* em baixo numa nova pasta designada calculadora2. Adicione também os ficheiros divisao.c, calculadora2.c e calculadora2.h | The following file uses the program calculadora2. Use a new makefile in a new directory dubbed calculadora2. Add also the files divisao.c, calculadora2.c and calculadora2.h to that directory. Compile e execute. | Compile and execute.

```
#Ficheiro makefile
CC=gcc
CFLAGS=-I.
LIBS = -lm

calculadora2: calculadora2.o divisao.o
      $(CC) -o calculadora2 calculadora2.o divisao.o $(CFLAGS) $(LIBS)
calculadora2.o: calculadora2.c
      $(CC) -c -o calculadora2.o calculadora2.c $(CFLAGS) $(LIBS)
divisao.o: divisao.c
      $(CC) -c -o divisao.o divisao.c $(CFLAGS) $(LIBS)

.PHONY: clean
clean:
      rm calculadora2 calculadora2.o divisao.o
```

fig.2: *makefile*.

3. GITHUB

Esta descrição foi adaptada [daqui](#). Existem muitos locais na Internet onde poderá aceder a informação sobre como utilizar o git. Na disciplina ir-se-á utilizar o git sobre o Linux. | The information in this section is based on this site. There are plenty of sites where the same information can be gathered.

3.1 Instalando o Git | Installing Git

Para a instalação basta abrimos o terminal e digitar: | To install you just need to collect the software package using the command:

```
sudo apt-get install git-core
```

Ele mostrará o espaço necessário para a instalação e mostrará quais os pacotes que serão instalados. Basta confirmar que todos os pacotes serão baixados e instalados. | It shows the space and the necessary packages to install.

Note que poderá ter de executar antes de instalar o *git*, a instrução de atualização da base de dados guardada no SO | You may have to update the database stored in the apt applications before installing the git using the following command:

```
sudo apt-get update
```

3.2 Configurando Git | Configure

```
git config --global user.name "Seu nome aqui" git  
config --global user.email "seuemail@seudominio.com"
```

Para verificar se tudo está certo, basta digitar no Terminal: | To test is everything is ok:

```
git config --global --list
```

3.3 Criando repositório no GitHub

Agora vamos criar um repositório lá no GitHub. Entre no GitHub, entre no seu perfil (tem de criar um perfil se ainda não o criou). | In GitHub select your profile (create a new one if you don't have any)

Clique em *New Repository*, colocando o nome do seu projeto, no nosso caso "so-c-project". O projeto criado será público. | Click in New Repository and add the name for the project "so-c-project"

Com o nosso repositório criado no GitHub vamos criar o nosso projeto agora localmente. No terminal vamos criar um diretório para o nosso projeto e iniciar o repositório do Git: | The project is created and now we create the project locally in our machine.

```
mkdir so-c-project  
cd so-c-project  
git init
```

Vamos identificar o repositório local com o repositório github. A seguir ao *origin* está a localização do repositório no GitHub. | We identify the place in the cloud where our project is.

```
git remote add origin git@github.com:seuloginnoglobit/so-c-project.git
```

Em seguida podemos criar o ficheiro README. Ele irá ser colocado no espaço do repositório, no github. | We add a README file (the usual procedure) and then prepare to add it to the cloud

```
touch README  
git add README git commit -m "inicio repositório com readme"
```

Repare que, neste momento, a operação “commit” prepara localmente o envio do ficheiro para o GitHub. Para o enviar basta digitar o seguinte comando: | Finally using the push command we send it to the cloud, synchronizing the data in you laptop with the git repository.

```
git push -u origin master
```

Pronto! Agora o repositório tem informação sobre o README. | That's it| Your first file is in the cloud (nothing inside the file but it's a file!)

3.4 Gerindo o repositório | Managing the repository

Sempre que quiser sincronizar o repositório do github com o local deverá fazer na diretoria do repositório: | Every time you want to synchronize with repository, you must pull the info from the master to origin.

```
git pull
```

Para sincronizar as alterações locais de forma a ficarem disponíveis no repositório, deverá fazer: | Every time you need to add the local changes you can do the following commands:

```
git add *  
git commit -m "alterações ...."  
git push -u origin master
```