



Relatório 02

Semana 02 - Ambiente de Programação Linux

Mateus Oliveira Costa (11711EAU018)

UBERLÂNDIA – MG

14/08/2023

Comandos Linux – Exercício 02

Para execução deste relatório será utilizado a distribuição Ubuntu 22.04 do sistema Linux no WSL2.

```
PS C:\Windows\System32> wsl
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
```

1. gcc “nome_do_arquivo.c” -o “nome_do_arquivo_de_saída.exe”

Este comando é utilizado para compilar arquivos utilizando o compilador do GCC, como de descrito adiante. Uma vez no diretório correto, compile o arquivo usando o GCC com o seguinte comando:

- **gcc -o seuarquivo seuarquivo.c**

Neste comando, "-o seuarquivo" define o nome do arquivo de saída gerado após a compilação. Substitua "seuarquivo" pelo nome que você deseja dar ao arquivo executável resultante. E "seuarquivo.c" é o nome do arquivo de código-fonte que você deseja compilar.

Pressione Enter para executar o comando. O GCC irá compilar o código-fonte e gerar um arquivo executável com o nome que você especificou. Se não houver erros no código-fonte, você verá o prompt novamente, e o arquivo executável estará no mesmo diretório onde você executou o comando.

Para executar o programa, use o seguinte comando no terminal:

- **./seuarquivo**

Substitua "seuarquivo" pelo nome do arquivo executável que você escolheu.

Segue exposto na imagem adiante a aplicação dos comandos descritos.

```
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace$ cd ex-01/
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace/ex-01$ code test.c
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace/ex-01$ gcc test.c -o test
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace/ex-01$ ls -l
total 20
-rwxr-xr-x 1 mateusoliv_dev mateusoliv_dev 15960 Aug 14 19:59 test
-rw-r--r-- 1 mateusoliv_dev mateusoliv_dev    69 Aug 14 19:59 test.c
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace/ex-01$ ./test
Hello!
```

2. gcc -c “nome_do_arquivo”

Este comando permite compilar arquivos com o GCC que não possuem abstratos de dados como exemplo a ausência de uma função “main()”. Assim, serão gerados arquivos objetos do tipo “.o”, que são códigos de máquinas em binário. Como exposto na imagem abaixo.

```
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace$  
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace$ cd ex-02  
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace/ex-02$ ls -l  
total 8  
-rwxr-xr-x 1 mateusoliv_dev mateusoliv_dev 2754 Aug 14 21:01 float_vector.c  
-rwxr-xr-x 1 mateusoliv_dev mateusoliv_dev 629 Aug 14 21:01 float_vector.h  
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace/ex-02$ gcc -c float_vector.c  
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace/ex-02$ ls -l  
total 16  
-rwxr-xr-x 1 mateusoliv_dev mateusoliv_dev 2754 Aug 14 21:01 float_vector.c  
-rwxr-xr-x 1 mateusoliv_dev mateusoliv_dev 629 Aug 14 21:01 float_vector.h  
-rw-r--r-- 1 mateusoliv_dev mateusoliv_dev 4600 Aug 14 21:04 float_vector.o
```

3. gcc -c “Nome_do_arquivo” -o “Nome_da_pasta/nome_do_arquivo.o”

Este comando permite compilar arquivos com o GCC que não possuem abstratos de dados como exemplo a ausência de uma função “main()” e salvá-los em um diretório específico. Assim, serão gerados arquivos objetos do tipo “.o”, que são códigos de máquinas em binário. Como exposto na imagem abaixo.

```
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace/ex-03$ gcc -c float_vector.c -o obj/float_vector.o  
mateusoliv_dev@DESKTOP-M080RHV:~/semb2_workspace/ex-03$ tree  
.  
├── float_vector.c  
├── float_vector.h  
└── obj  
    └── float_vector.o  
  
1 directory, 3 files
```

4. gcc -c float_vector.c -I include/ -o obj/float_vector.o

Este comando permite compilar arquivos com o GCC que não possuem abstrações de dados, como por exemplo a ausência de uma função 'main()' e salvá-los em um diretório específico. Assim, serão gerados arquivos objeto do tipo '.o', que contêm código de máquina em formato binário. Além disso, a opção '-I' indica para o GCC que, caso o arquivo com abstrações de dados tenha inclusões (includes), supostamente essas possam ser encontradas no diretório subsequente descrito, conforme mostrado na imagem adiante.

```
mateusoliv_dev@DESKTOP-M080RHHV:~/semb2_workspace/ex-04$ mkdir include
mateusoliv_dev@DESKTOP-M080RHHV:~/semb2_workspace/ex-04$ ls -l
total 16
-rwxr-xr-x 1 mateusoliv_dev mateusoliv_dev 2754 Aug 14 23:20 float_vector.c
-rwxr-xr-x 1 mateusoliv_dev mateusoliv_dev 629 Aug 14 23:20 float_vector.h
drwxr-xr-x 2 mateusoliv_dev mateusoliv_dev 4096 Aug 14 23:35 include
drwxr-xr-x 2 mateusoliv_dev mateusoliv_dev 4096 Aug 14 23:35 obj
mateusoliv_dev@DESKTOP-M080RHHV:~/semb2_workspace/ex-04$ mv float_vector.h include/
mateusoliv_dev@DESKTOP-M080RHHV:~/semb2_workspace/ex-04$ cd obj/
mateusoliv_dev@DESKTOP-M080RHHV:~/semb2_workspace/ex-04/obj$ rm float_vector.o
mateusoliv_dev@DESKTOP-M080RHHV:~/semb2_workspace/ex-04/obj$ cd ..
mateusoliv_dev@DESKTOP-M080RHHV:~/semb2_workspace/ex-04$ gcc -c float_vector.c -o obj/float_vector.o
float_vector.c:1:10: fatal error: float_vector.h: No such file or directory
  1 | #include "float_vector.h"
    | ^~~~~~
compilation terminated.
mateusoliv_dev@DESKTOP-M080RHHV:~/semb2_workspace/ex-04$
mateusoliv_dev@DESKTOP-M080RHHV:~/semb2_workspace/ex-04$ tree .
.
├── float_vector.c
├── include
│   └── float_vector.h
└── obj

2 directories, 2 files
mateusoliv_dev@DESKTOP-M080RHHV:~/semb2_workspace/ex-04$ gcc -c float_vector.c -I include/ -o obj/float_vector.o
```