UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
Curso de Graduação em Engenharia Mecatrônica

RELATÓRIO II DE SISTEMAS DIGITAIS(FEELT49081)

COMPILAÇÃO DE ARQUIVOS NO LINUX

Prof. Éder Alves de Moura

Guilherme Vitor dos Santos Rodrigues 11321EMT009

Uberlândia, Dezembro de 2021.

Sumário

1. Introdução

Um sistema embarcado é o conjunto de hardware e software para aplicações especificas. Este pode conter um microprocessador ou microcontrolador com o computador escondido no sistema. Assim, para sua utilização é necessário o domínio de sistemas operacionais abertos como por exemplo o Linux. E neste relatório é possível identificar aplicações em uma maquina virtual.

2. Objetivos

Como objetivos, para a Semana 2, temos a compilação de programas no Linux utilizando o GCC, tanto para arquivos em linguagem C quanto para arquivos em linguagem python.

3. Comandos de compilação de programas no Linux

Para ocorrer o processo de compilação, é necessário que ocorram também as seguintes etapas:

- Pré-processamento
- Compilação
- Montagem
- Vinculação de arquivos

Então, para este processo ser realizado no Linux precisamos criar o arquivo c e depois realizar as etapas de compilação. Primeiramente um projeto em c é criado:

```c
/*
 * Listing 3.1
 * hello.c – Canonical “Hello, world!” program 4   4  */
#include <stdio.h>

int main(void)
{
    fprintf(stdout, "Hello, Linux programming world!\n");
     return 0;
}
```

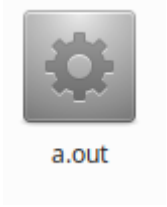
Agora com este arquivo se chamando *file.c*, utilizando o terminal:

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues$ gcc file.
c -o file
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues$ ./file
Hello, Linux programming world!
```

Tem-se então o processo de montagem e vinculação de arquivos de um programa em c. Para a realização de aplicações em c é necessário utilizar o gcc, que tem por objetivo fornecer parâmetros extras, desde a emissão de erros até o binário final e otimizando para determinados comportamentos. Então determinados comandos serão realizados e seus resultados serão mostrados.

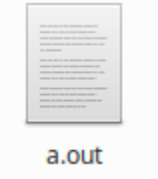
- -static: Tem como resultado a criação do arquivo abaixo

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues/Semana02/a
rquivos_de_teste$ gcc file.c -static
```



- -g: Inclui uma informação padrão de debug

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues/Semana02/a
rquivos_de_teste$ gcc file.c -g
```



- -pedantic: Geralmente exibe todos os erros e *warnings* presente na aplicação

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues/Semana02/a
rquivos_de_teste$ gcc file.c -pedantic
/usr/bin/ld: warning: cannot find entry symbol rrors; defaulting to 000000000000
1060
```

- -Wall: Emite todos os warnings uteis que o gcc pode prover

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues/Semana02/a
rquivos_de_teste$ gcc file.c -Wall
```

- -Os:

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues/Semana02/a
rquivos_de_teste$ gcc -Os file.c
```

- -O3: Envolve a otimização de código, no caso todas as 2 otimizações

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues/Semana02/a
rquivos_de_teste$ gcc -O3 file.c
```

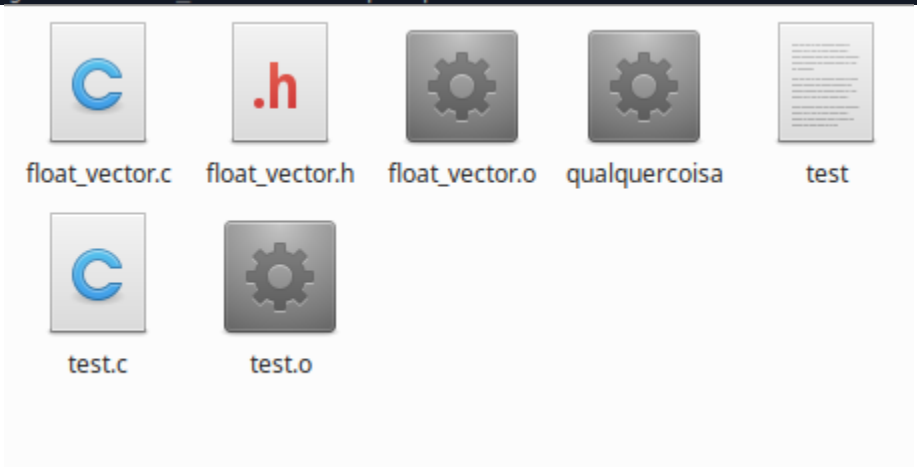
4. Atividades da playlist de vídeos

Após extrair os arquivos temos que compilar o arquivo como no comando abaixo:

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues/Semana02/v
ideo01$ gcc -c float_vector.c
```

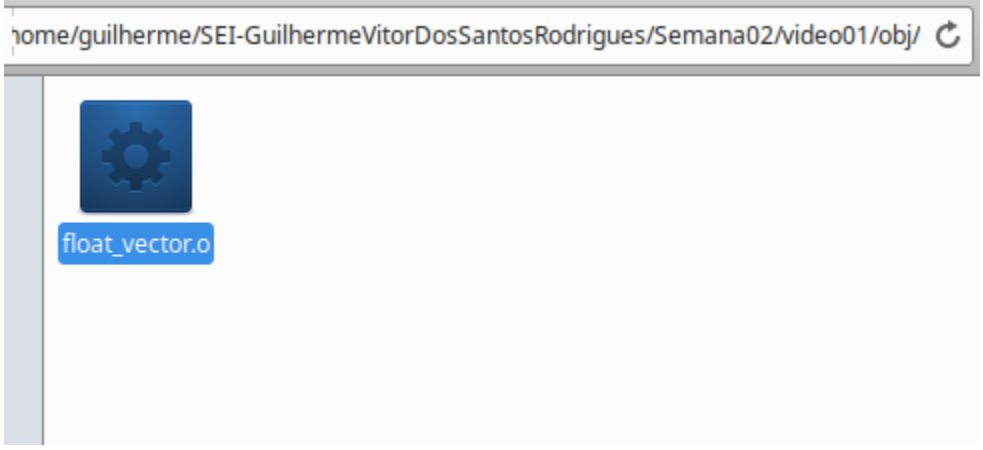
Renomeando o programa:

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues/Semana02/v
ideo01$ gcc -c float_vector.c -o qualquercoisa
```



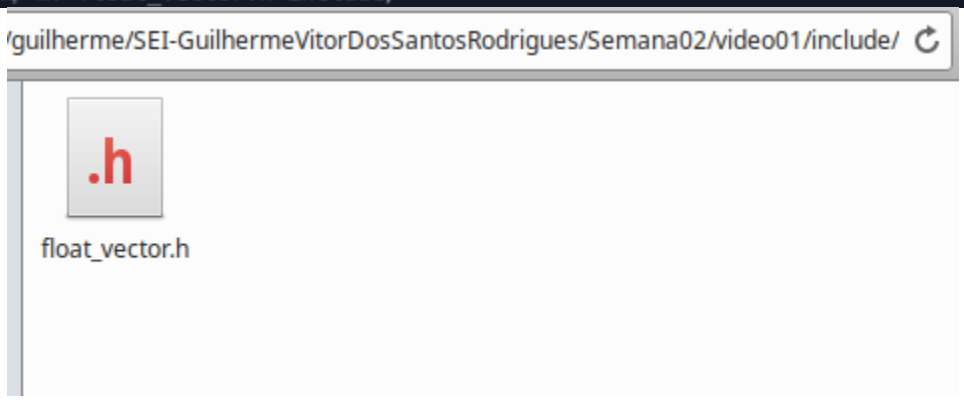
Criando uma pasta chamada *obj*, podemos mover os arquivos *.o* para ela com o comando abaixo:

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues/Semana02/v
ideo01$ gcc -c float_vector.c -o obj/float_vector.o
```



Criando uma pasta chamada *include*, podemos mover os arquivos *.h* para ela com o comando abaixo:

```
guilherme@guilherme-VirtualBox:~/SEI-GuilhermeVitorDosSantosRodrigues/Semana02/v
ideo01$ mv float_vector.h include/
```



Os códigos dos próximos vídeos se encontram no GitHub.

Referências

[1] - “Kurt Wall. Linux Programming Unleashed". SAMS, 2007.capitulo 3