

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

**Semana 02:**

Prof. Edér Alves de Moura

Amanda Abigail Bonilla - 11711EMT027

UBERLÂNDIA

2022

1) Liste e descreva o que são as 4 etapas do processo de compilação.

- a) Pré-processamento: Nesta etapa o compilador lida com as diretivas que foram introduzidas com o #, como #include e #define, além de suprimir os comentários.
- b) Compilação: Nessa etapa, o compilador vai verificar o código pré-processado e vai traduzir o código fonte, gerado na etapa anterior, para a linguagem da máquina. O compilador lê como um conjunto de instruções a serem executadas, e é bom lembrar que pode dar um erro durante o processo se tiver um erro de sintaxe.
- c) Montagem: Esta etapa irá gerar um arquivo binário em código máquina, pronto para ser interpretado pelo processador.
- d) Lincagem: Finalmente, vamos linkar os arquivos binários (objeto) da etapa anterior para gerar o executável autônomo que será distribuído para os usuários.

2) Desenvolva uma aplicação simples que demonstre o uso de múltiplos arquivos para a construção de uma aplicação em C.

Para mostrar o uso múltiplo de arquivos de uma ação simples como printar “Hello Brasil!”, foram criados três arquivos;

Um main.c

```
C main.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "hello.h"
4
5  int main(){
6      hello();
7      return(0);
8
9  }
```

um hello.h

```
✓ SEMANA02_SD [📁] [📄] [🔄] [📄]
  C hello.c
  C hello.h
  C main.c

C hello.h
1  #ifndef _h_teste
2  #define _h_teste
3
4  void hello(void);
5
6  #endif
7
```

e um hello.c

```
✓ SEMANA02_SD [📁] [📄] [🔄] [📄]
  C hello.c
  C hello.h
  C main.c
  M makefile

C hello.c
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void hello(void){
5      printf("Hello Brasil!\n");
6  }
7
```

e logo nosso makefile

```
✓ SEMANA02_SD [📁] [📄] [🔄] [📄]
  C hello.c
  C hello.h
  C main.c
  M makefile

M makefile
1  text: printy
2  printy: main.o hello.o
3      gcc -o printy main.o hello.o
4
5  main.o: main.c hello.h
6      gcc -o main.c -c -W -Wall -ansi -pedantic
7
8  hello.o: hello.c hello.h
9      gcc -o hello.o hello.c -c -W -Wall -pedantic
10
11 clean:
12     rm -rf *.o *~printy
```

Para executar essa aplicação usamos os comandos

```
$ make text
```

```
$ ./printy
```

- 3) O compilador gcc permite fornecer parâmetros extras, que modificam desde a emissão de erros até o binário final, o otimizando para determinados

comportamentos. Explique a função e crie um exemplo para demonstrar a funcionalidade dos seguintes parâmetros:

a) -static

Parâmetro que especifica que a vinculação deve ser estática

```
compilation terminated.
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ gcc main.c -o teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ ls -ls
total 24
 4 main.c 20 teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ gcc -static main.c -o teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ ls -ls
total 856
 4 main.c 852 teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$
```

b) -g

É um parâmetro que gera informações de depuração padrão não binário.

```
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ gcc main.c -o teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ ls -l teste
-rwxrwxr-x 1 userabhi userabhi 16464 dez 19 13:05 teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ gcc -g main.c -o teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ ls -l teste
-rwxrwxr-x 1 userabhi userabhi 18976 dez 19 13:06 teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$
```

c) -pedantic

Parâmetro que emite todos os avisos exigidos pelo padrão ANSI/ISO C.

```
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ gcc -pedantic main.c -o teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$
```

d) -Wall

Parâmetro que emite todos os avisos que o gcc pode fornecer.

```
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ gcc -Wall main.c -o teste
main.c: In function 'main':
main.c:5:9: warning: unused variable 'total' [-Wunused-variable]
  5 |     int total;
    |         ^~~~~
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$
```

e) -Os

Parâmetro que otimiza a compilação considerando o tamanho.

```
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ gcc -Os main.c -o teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$
```

f) -O3

Parâmetro que otimiza a compilação, habilita a otimização -O2 e mais algumas funções.

```
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$ gcc -O3 main.c -o teste
userabhi@userabhi-VirtualBox:~/Desktop/semana02_SD/teste-02$
```