

Universidade Federal de Uberlândia
Curso de Graduação em Engenharia Mecatrônica
Sistemas Embarcados II / Sistemas Digitais para Mecatrônica

Semana 02 - Ambiente de Programação Linux

Relatório referente à leitura e reprodução dos comandos e conceitos referentes ao ambiente de programação Linux, apresentados no livro referência, “Linux: Comandos Básicos e Avançados” de Vivas, Araújo, Pitangui e Assis.

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

Os quatros processos de compilação são:

1. Pré-processamento (Preprocessing)
2. Compilação (Compilation Proper)
3. Montagem (Assembly)
4. Lincagem ou Ligante (Linking)

No pré-processamento, o compilador trabalha com diretivas introduzidas no código que são precedidas pelo caractere #, tal como instruções #include e #define, também incluindo comentários.

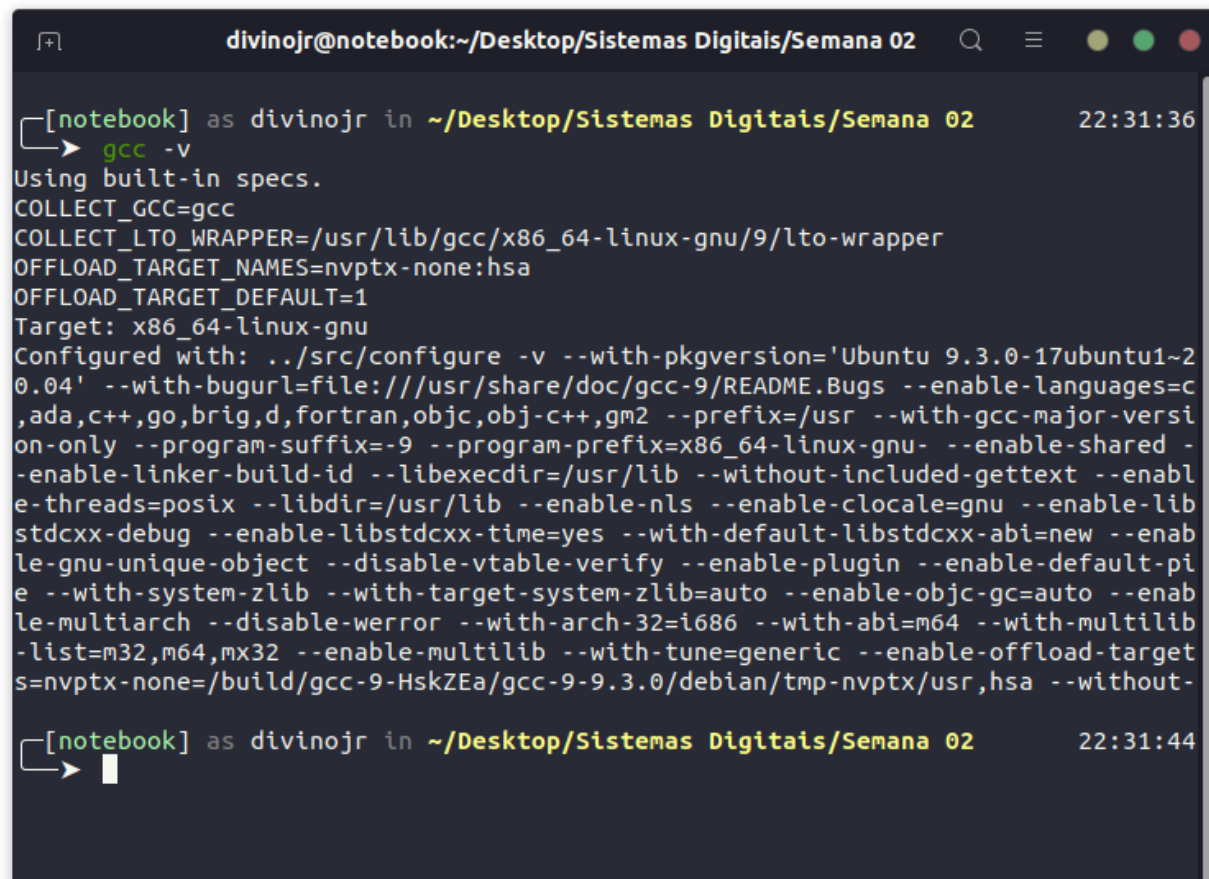
Durante a etapa de compilação, o compilador traduz o código fonte (neste caso, C/C++), gerado na etapa de pré-processamento para a linguagem Assembly.

A etapa de montagem gera um arquivo binário em código de máquina que ao ser processado, estará pronto para ser interpretado pelo processador.

A etapa de lincagem, trabalha na mesclagem (link) dos arquivos objeto (ou binários) gerados na etapa de montagem, gerando o arquivo executável que pode ser utilizado pelos usuários.

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

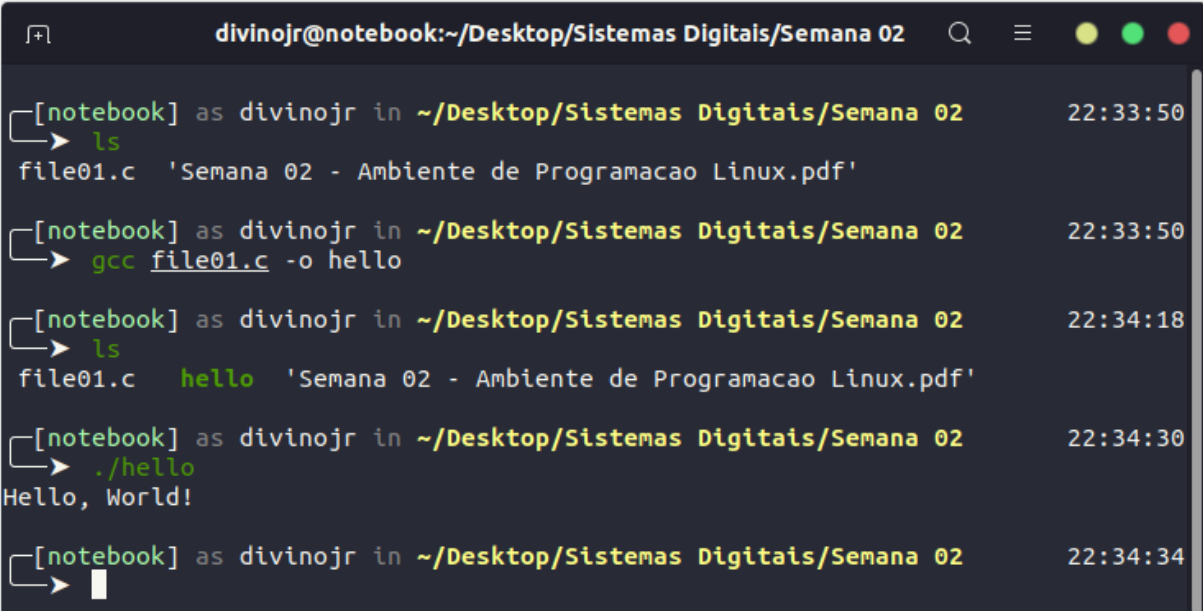
Versão do GCC

A terminal window titled 'divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02' shows the command 'gcc -v' being executed. The output displays the GCC version as 9.3.0-17ubuntu1~20.04 and lists various configuration options and target architecture (x86_64-linux-gnu). The terminal prompt returns to the user.

```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 22:31:36
> gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/9/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:hsa
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 9.3.0-17ubuntu1~20.04' --with-bugurl=file:///usr/share/doc/gcc-9/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++,gm2 --prefix=/usr --with-gcc-major-version-only --program-suffix=-9 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib=auto --enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch=32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none=/build/gcc-9-HskZEa/gcc-9-9.3.0/debian/tmp-nvptx/usr,hsa --without-
```

A instrução a seguir é utilizada para compilar o programa Hello World presente no arquivo file01.c. Onde, o argumento -o representa a especificação do nome do arquivo de saída (hello).

```
//Hello, World!  
#include <stdio.h>  
  
int main(void){  
  
printf("Hello, World!\n");  
  
return 0;  
}
```



```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02  
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 22:33:50  
➤ ls  
file01.c 'Semana 02 - Ambiente de Programacao Linux.pdf'  
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 22:33:50  
➤ gcc file01.c -o hello  
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 22:34:18  
➤ ls  
file01.c hello 'Semana 02 - Ambiente de Programacao Linux.pdf'  
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 22:34:30  
➤ ./hello  
Hello, World!  
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 22:34:34  
➤
```

Para representar os processos de compilação e construção de uma aplicação em C, foi utilizado o seguinte código salvo no arquivo file02.c.

```
#include <stdio.h>
#include "file02.h"

int value1 = 10, value2 = 20;

int main(void){

    printf("O valor da multiplicacao entre %i e %i = %i\n", value1, value2,
    mult(value1, value2));

    return 0;
}
```

Código presente no arquivo file02.h.

```
int mult(int a, int b){

    return a*b;
}
```

Conforme visto anteriormente, a compilação é iniciada com o pré-processamento das diretivas que acompanham o caractere '#'. Para dar início ao pré-processamento, foi utilizado o comando '*gcc -E file02.c*', sendo possível observar algumas das linhas de comandos a seguir.



```
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02
```

18:43:3

5

```
→ gcc -E file02.c
```

```
# 1 "file02.c"
# 1 "<built-in>"
# 1 "<command-line>"
# 31 "<command-line>"
# 1 "/usr/include/stdc-predef.h" 1 3 4
# 32 "<command-line>" 2
# 1 "file02.c"
# 1 "/usr/include/stdio.h" 1 3 4
# 27 "/usr/include/stdio.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 1 3 4
# 33 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 3 4
# 1 "/usr/include/features.h" 1 3 4
# 461 "/usr/include/features.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 1 3 4
# 452 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
# 453 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/long-double.h" 1 3 4
# 454 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 2 3 4
# 462 "/usr/include/features.h" 2 3 4
# 485 "/usr/include/features.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 1 3 4
# 10 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/gnu/stubs-64.h" 1 3 4
# 11 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 2 3 4
# 486 "/usr/include/features.h" 2 3 4
# 34 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 2 3 4
# 28 "/usr/include/stdio.h" 2 3 4
```



```
# 1 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 1 3 4
# 209 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 3 4

# 209 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 3 4
typedef long unsigned int size_t;
# 34 "/usr/include/stdio.h" 2 3 4


# 1 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stdarg.h" 1 3 4
# 40 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stdarg.h" 3 4
typedef __builtin_va_list __gnuc_va_list;
# 37 "/usr/include/stdio.h" 2 3 4


# 1 "/usr/include/x86_64-linux-gnu/bits/types.h" 1 3 4
# 27 "/usr/include/x86_64-linux-gnu/bits/types.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
# 28 "/usr/include/x86_64-linux-gnu/bits/types.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 1 3 4
# 29 "/usr/include/x86_64-linux-gnu/bits/types.h" 2 3 4


typedef unsigned char __u_char;
typedef unsigned short int __u_short;
typedef unsigned int __u_int;
typedef unsigned long int __u_long;


typedef signed char __int8_t;
typedef unsigned char __uint8_t;
typedef signed short int __int16_t;
typedef unsigned short int __uint16_t;
```



divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02



```
typedef signed int __int32_t;  
typedef unsigned int __uint32_t;  
  
typedef signed long int __int64_t;  
typedef unsigned long int __uint64_t;
```

```
typedef __int8_t __int_least8_t;  
typedef __uint8_t __uint_least8_t;  
typedef __int16_t __int_least16_t;  
typedef __uint16_t __uint_least16_t;  
typedef __int32_t __int_least32_t;  
typedef __uint32_t __uint_least32_t;  
typedef __int64_t __int_least64_t;  
typedef __uint64_t __uint_least64_t;
```

```
typedef long int __quad_t;  
typedef unsigned long int __u_quad_t;
```

```
typedef long int __intmax_t;
```



divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02



```
typedef int __key_t;

typedef int __clockid_t;

typedef void * __timer_t;

typedef long int __blksize_t;


typedef long int __blkcnt_t;
typedef long int __blkcnt64_t;


typedef unsigned long int __fsblkcnt_t;
typedef unsigned long int __fsblkcnt64_t;


typedef unsigned long int __fsfilcnt_t;
typedef unsigned long int __fsfilcnt64_t;


typedef long int __fsword_t;
typedef long int __ssize_t;


typedef long int __syscall_slong_t;
```



```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02
typedef struct _G_fpos_t
{
    __off_t __pos;
    __mbstate_t __state;
} __fpos_t;
# 40 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types/__fpos64_t.h" 1 3 4
# 10 "/usr/include/x86_64-linux-gnu/bits/types/__fpos64_t.h" 3 4
typedef struct _G_fpos64_t
{
    __off64_t __pos;
    __mbstate_t __state;
} __fpos64_t;
# 41 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types/__FILE.h" 1 3 4

struct _IO_FILE;
typedef struct _IO_FILE __FILE;
# 42 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types/FILE.h" 1 3 4

struct _IO_FILE;

typedef struct _IO_FILE FILE;
# 43 "/usr/include/stdio.h" 2 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/types/struct_FILE.h" 1 3 4
# 35 "/usr/include/x86_64-linux-gnu/bits/types/struct_FILE.h" 3 4
```



```
extern int remove (const char *__filename) __attribute__ ((__nothrow__ , __leaf__));

extern int rename (const char *__old, const char *__new) __attribute__ ((__nothrow__ , __leaf__));

extern int renameat (int __oldfd, const char *__old, int __newfd,
                    const char *__new) __attribute__ ((__nothrow__ , __leaf__));
# 173 "/usr/include/stdio.h" 3 4
extern FILE *tmpfile (void) ;
# 187 "/usr/include/stdio.h" 3 4
extern char *tmpnam (char *__s) __attribute__ ((__nothrow__ , __leaf__));

extern char *tmpnam_r (char *__s) __attribute__ ((__nothrow__ , __leaf__));
# 204 "/usr/include/stdio.h" 3 4
extern char *tempnam (const char *__dir, const char *__pfx)
    __attribute__ ((__nothrow__ , __leaf__)) __attribute__ ((__malloc__));

extern int fclose (FILE *__stream);
```



```
extern int getchar (void);
```

```
extern int getc_unlocked (FILE *__stream);  
extern int getchar_unlocked (void);  
# 510 "/usr/include/stdio.h" 3 4  
extern int fgetc_unlocked (FILE *__stream);  
# 521 "/usr/include/stdio.h" 3 4  
extern int fputc (int __c, FILE *__stream);  
extern int putc (int __c, FILE *__stream);
```

```
extern int putchar (int __c);  
# 537 "/usr/include/stdio.h" 3 4  
extern int fputc_unlocked (int __c, FILE *__stream);
```

```
extern int putc_unlocked (int __c, FILE *__stream);
```



```
extern void perror (const char *__s);
```

```
# 1 "/usr/include/x86_64-linux-gnu/bits/sys_errlist.h" 1 3 4
```

```
# 26 "/usr/include/x86_64-linux-gnu/bits/sys_errlist.h" 3 4
```

```
extern int sys_nerr;
```

```
extern const char *const sys_errlist[];
```

```
# 782 "/usr/include/stdio.h" 2 3 4
```

```
extern int fileno (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;
```

```
extern int fileno_unlocked (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__f__));
```

```
# 800 "/usr/include/stdio.h" 3 4
```

```
extern FILE *popen (const char *__command, const char *__modes) ;
```

```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02

extern int ftrylockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));

extern void funlockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));
# 858 "/usr/include/stdio.h" 3 4
extern int __uflow (FILE *);
extern int __overflow (FILE *, int);
# 873 "/usr/include/stdio.h" 3 4

# 2 "file02.c" 2
# 1 "file02.h" 1

# 1 "file02.h"
int mult(int a, int b){

    return a*b;
}
# 3 "file02.c" 2

int value1 = 10, value2 = 20;

int main(void){

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:43:54
> █
```

Ao final das informações, é possível notar a estrutura do código no arquivo .c e .h.

Após o pré-processamento, foi realizada a compilação utilizando o comando 'gcc -S file02.c'. Onde a opção '-S' representa a instrução que o compilador recebe para converter o código pré-processado para assembly sem a necessidade de criar um objeto, traduzindo o arquivo 'file02.i' em 'file02.s' que contém o código em assembly.

```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:49:01
> ls
file01.c  file02.h  file02.s
file02    file02.i  'Semana 02 - Ambiente de Programacao Linux.pdf'
file02.c  file02.o

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:49:03
> 
```

A etapa de montagem, consiste na compilação efetiva do script, gerando o código de máquina, com a instrução `gcc -c file02.c`, onde é gerado o arquivo `file02.o`. Logo em seguida, foi executada a etapa de ligamento (ou lincagem), através do comando `gcc -o hello file02.o`, gerando o arquivo executável `hello`.

```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02

printf("O valor da multiplicacao entre %i e %i = %i\n", value1, value2, mul
t(value1, value2));

return 0;
}

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:51:48
> gcc -S file02.c

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:52:13
> ls
file01.c  file02.h  'Semana 02 - Ambiente de Programacao Linux.pdf'
file02.c  file02.s

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:52:17
> gcc -c file02.c

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:52:25
> ls
file01.c  file02.h  file02.s
file02.c  file02.o  'Semana 02 - Ambiente de Programacao Linux.pdf'

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:52:27
> 
```

```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:53:52
> ls
file01.c  file02.h  file02.s
file02.c  file02.o  'Semana 02 - Ambiente de Programacao Linux.pdf'

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:53:52
> gcc -o hello file02.o

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:53:58
> ls
file01.c  file02.o  'Semana 02 - Ambiente de Programacao Linux.pdf'
file02.c  file02.s
file02.h  hello

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:53:59
> ./hello
O valor da multiplicacao entre 10 e 20 = 200

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 18:54:02
> █
```

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

O parâmetro -static vincula um programa estaticamente, desfazendo a necessidade de que haja a existência de uma dependência de bibliotecas dinâmicas em tempo de execução para que o programa seja executado.

Para obter a vinculação estática, é necessário que as versões de arquivo (.a) de suas bibliotecas existam no sistema. então /usr/lib/libc.a /usr/lib/crt1.o etc ...

Em sistemas Linux mais modernos quando um binário é vinculado aos seguintes passos que são seguidos:

- 1) coloca o código no executável por meio de arquivos .o e .a
- 2) coloca referências a arquivos de bibliotecas dinâmicas (.so) isso é resolvido por /lib/ld-linux.so (ou /lib64/ld-linux=x86-64.so) que está sempre em um local conhecido.

Para realizar comandos utilizando a opção `-static` é necessário possuir o pacote `glib-static`. Utilizando o pacote `build-essential`, pode-se checar a versão utilizando o comando `ldd --version`.

A seguir estão disponíveis os códigos e os comandos utilizados.

```
/* Filename: lib_mylib.c
```

```
Código Adaptado da geeksforgeeks
```

```
https://www.geeksforgeeks.org/static-vs-dynamic-libraries/
```

```
*/
```

```
#include <stdio.h>
```

```
void fun_print(void)
```

```
{
```

```
    printf("Funcao executada pela biblioteca estatica\n\n");
```

```
}
```

```
/* Filename: lib_mylib.h */
```

```
/* Código Adaptado da geeksforgeeks
```

```
https://www.geeksforgeeks.org/static-vs-dynamic-libraries/
```

```
*/
```

```
void fun_print(void);
```

```
/* Filename: main.c */
```

```
/* Código Adaptado da geeksforgeeks
```

```
https://www.geeksforgeeks.org/static-vs-dynamic-libraries/
```

```
*/
```

```
#include "lib_mylib.h"
```

```
void main()
```

```
{
```

```
    fun_print();
```

```
}
```



```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02/file03

[notebook] as divinojr ln ~/Desktop/Sistemas Digitais/Semana 02/file03
> ls
driver.o lib_mylib.a lib_mylib.c lib_mylib.h lib_mylib.o main.c

[notebook] as divinojr ln ~/Desktop/Sistemas Digitais/Semana 02/file03
> gcc -c lib_mylib.c -o lib_mylib.o

[notebook] as divinojr ln ~/Desktop/Sistemas Digitais/Semana 02/file03
> ar rcs lib_mylib.a lib_mylib.o

[notebook] as divinojr ln ~/Desktop/Sistemas Digitais/Semana 02/file03
> gcc -c main.c -o main.o

[notebook] as divinojr ln ~/Desktop/Sistemas Digitais/Semana 02/file03
> gcc -o main main.o -L. -l_mylib

[notebook] as divinojr ln ~/Desktop/Sistemas Digitais/Semana 02/file03
> ls
driver.o lib_mylib.c lib_mylib.o main.c
lib_mylib.a lib_mylib.h main main.o

[notebook] as divinojr ln ~/Desktop/Sistemas Digitais/Semana 02/file03
[notebook] as divinojr ln ~/Desktop/Sistemas Digitais/Semana 02/file03
>
```

b) -g

O parâmetro “-g” gera informações de debug a serem utilizadas no GDB debugger, com algumas opções na tabela a seguir:

-g0	Sem informações de Debug
-g1	Mínimas informações de Debug
-g3	Máximas informações de Debug
-g	Informações padrão de Debug

Compilando o arquivo file02.c já apresentado anteriormente podemos observar os resultados a seguir:

```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 19:23:06
> ls
file01.c  file02.o  'Semana 02 - Ambiente de Programacao Linux.pdf'
file02.c  file02.s
file02.h  hello

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 19:23:07
> gcc -g file02.c -o mult

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 19:23:14
> gdb mult
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from mult...
(gdb) run
Starting program: /home/divinojr/Desktop/Sistemas Digitais/Semana 02/mult
0 valor da multiplicacao entre 10 e 20 = 200
[Inferior 1 (process 34165) exited normally]
(gdb) q

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02
> 
```

c) -pedantic

O pedantic habilita alguns avisos que existem como padrão no C mas são desativados para extensões do GCC.

Utilizando o arquivo file02.c como exemplo, temos:

```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02/file02

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02/file02 20:48:11
> ls
file02.a  file02.c  file02.h  file02.o  file02.s

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02/file02 20:48:11
> gcc -Wall -Wextra -pedantic -o file02 file02.c

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02/file02 20:48:18
> ls
file02  file02.a  file02.c  file02.h  file02.o  file02.s

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02/file02 20:48:20
> ./file02
0 valor da multiplicacao entre 10 e 20 = 200

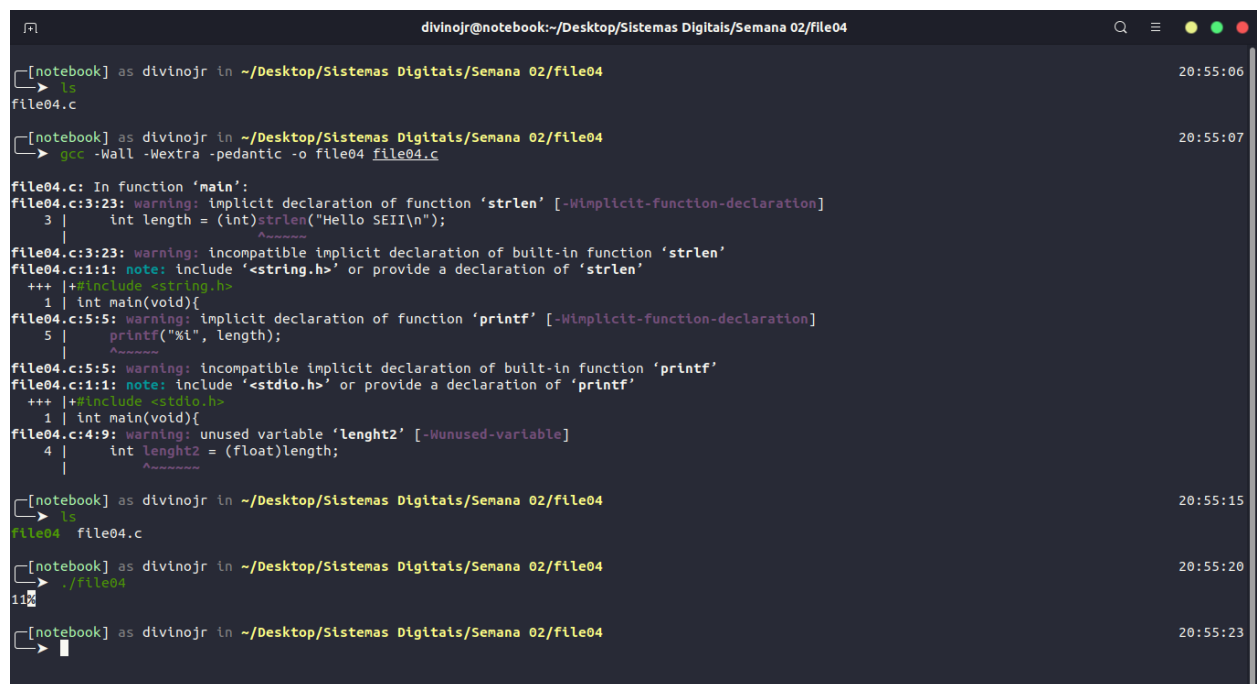
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02/file02 20:48:23
> 
```

Para o programa a seguir, que contém variáveis inutilizadas, temos a seguinte resposta:

```
int main(void){

    int length = (int)strlen("Hello SEII\n");
    int lenght2 = (float)length;
    printf("%i", length);

    return 0;
}
```



```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02/file04
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02/file04 20:55:06
> ls
file04.c
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02/file04 20:55:07
> gcc -Wall -Wextra -pedantic -o file04 file04.c

file04.c: In function 'main':
file04.c:3:23: warning: implicit declaration of function 'strlen' [-Wimplicit-function-declaration]
   3 |     int length = (int)strlen("Hello SEII\n");
     |                               ^
file04.c:3:23: warning: incompatible implicit declaration of built-in function 'strlen'
file04.c:1:1: note: include '<string.h>' or provide a declaration of 'strlen'
+++ |+#include <string.h>
   1 | int main(void){
file04.c:5:5: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]
   5 |     printf("%i", length);
     |     ^
file04.c:5:5: warning: incompatible implicit declaration of built-in function 'printf'
file04.c:1:1: note: include '<stdio.h>' or provide a declaration of 'printf'
+++ |+#include <stdio.h>
   1 | int main(void){
file04.c:4:9: warning: unused variable 'lenght2' [-Wunused-variable]
   4 |     int lenght2 = (float)length;
     |         ^
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02/file04 20:55:15
> ls
file04  file04.c
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02/file04 20:55:20
> ./file04
11
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02/file04 20:55:23
>
```

d) -Wall

O comando “-Wall” habilita mensagens de warning do compilador. Ao executar o código do arquivo file03.c foram encontrados avisos sobre a variável inutilizada.

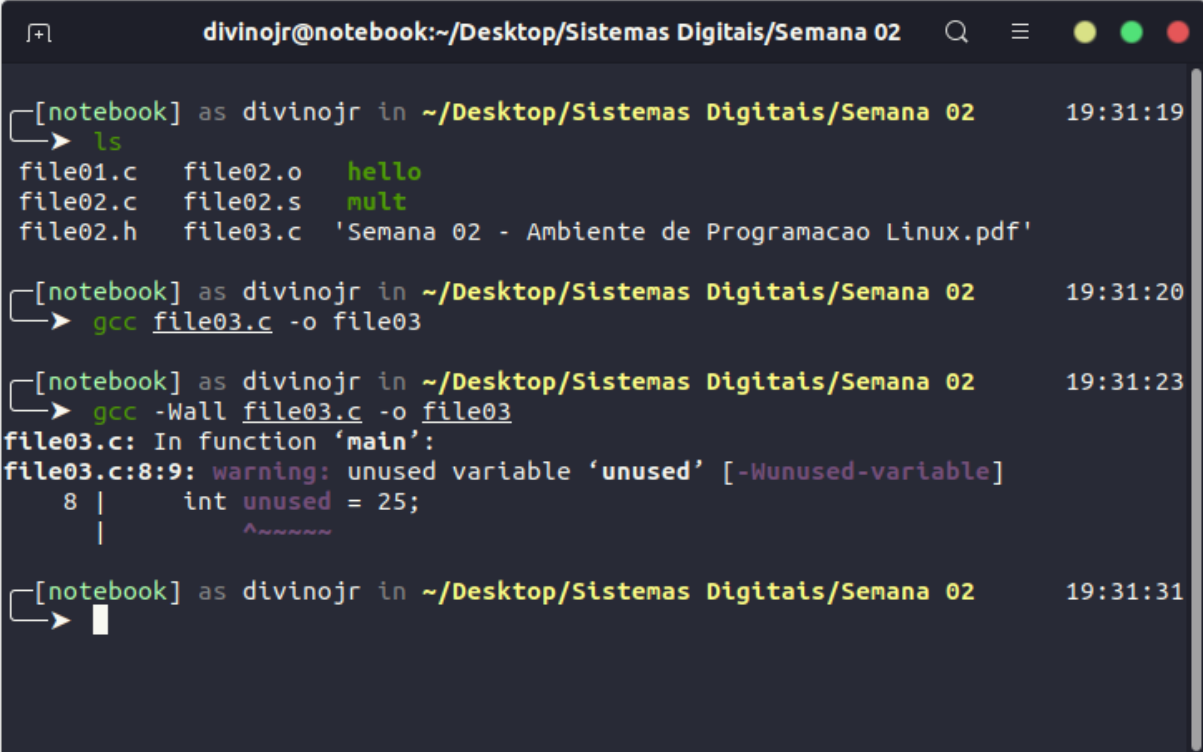
```
#include <stdio.h>

int value1 = 10, value2 = 20;

int main(void){

    int unused = 25;
    printf("O valor da multiplicacao entre %i e %i = %i\n", value1, value2,
    mult(value1, value2));

    return 0;
}
```



```
divinojr@notebook:~/Desktop/Sistemas Digitais/Semana 02

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 19:31:19
> ls
file01.c  file02.o  hello
file02.c  file02.s  mult
file02.h  file03.c  'Semana 02 - Ambiente de Programacao Linux.pdf'

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 19:31:20
> gcc file03.c -o file03

[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 19:31:23
> gcc -Wall file03.c -o file03
file03.c: In function 'main':
file03.c:8:9: warning: unused variable 'unused' [-Wunused-variable]
     8 |     int unused = 25;
       |         ~~~~~
[notebook] as divinojr in ~/Desktop/Sistemas Digitais/Semana 02 19:31:31
> 
```

REFERÊNCIAS

[1] Kurt Wall. Linux Programming Unleashed. SAMS, 2007.

[2]

<https://medium.com/permalink-univesp/guia-b%C3%A1sico-de-compila%C3%A7%C3%A3o-com-gcc-7516912df0bb>

[3] <https://ichi.pro/pt/as-quatro-fases-da-compilacao-de-um-programa-c-116431640209681>

[4] <https://www.geeksforgeeks.org/static-vs-dynamic-libraries/>