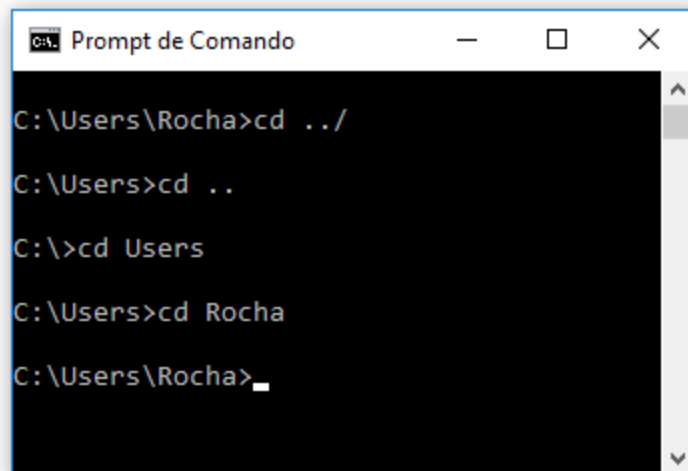




Tutorial Makefile

Como compilar um arquivo .c ?

1. Abra o terminal
2. Vá até a pasta onde está localizado o arquivo usando o comando `cd` (Funciona pra Windows/Mac/Linux)



```
CA: Prompt de Comando
C:\Users\Rocha>cd ../
C:\Users>cd ..
C:\>cd Users
C:\Users>cd Rocha
C:\Users\Rocha>
```

3. Use o comando a seguir:

```
gcc "nome dos arquivos".c -o "nome do executável"
```

Esse comando quer dizer: "Compilador, pegue os arquivos .c e gere um executável"

4. Execute o arquivo gerado usando o comando a seguir:

```
./"nome do executável" (Linux e Mac)
```

```
"nome do arquivo".exe (Windows)
```

```
francisco@francisco-VirtualBox: ~  
francisco@francisco-VirtualBox:~$ ls  
app1.c  Área de Trabalho  examples.desktop  Modelos  Público  Vídeos  
app1.c~ Documentos      Imagens          Música  Transferências  
francisco@francisco-VirtualBox:~$ gcc app1.c -o aplicativo1  
francisco@francisco-VirtualBox:~$ ls  
aplicativo1  Área de Trabalho  Imagens  Público  
app1.c       Documentos        Modelos  Transferências  
app1.c~      examples.desktop  Música   Vídeos  
francisco@francisco-VirtualBox:~$ ./aplicativo1
```

```
C:\Users\55199\Desktop\TutorialMakefile>dir
O volume na unidade C é Windows
O Número de Série do Volume é 4031-2AB0

Pasta de C:\Users\55199\Desktop\TutorialMakefile

03/09/2020 13:51 <DIR> .
03/09/2020 13:51 <DIR> ..
03/09/2020 13:47      827 main.c
03/09/2020 12:44      214 operacoes.c
03/09/2020 12:44      137 operacoes.h
                3 arquivo(s)      1.178 bytes
                2 pasta(s) 141.091.213.312 bytes disponíveis

C:\Users\55199\Desktop\TutorialMakefile>gcc main.c operacoes.c -o out

C:\Users\55199\Desktop\TutorialMakefile>dir
O volume na unidade C é Windows
O Número de Série do Volume é 4031-2AB0

Pasta de C:\Users\55199\Desktop\TutorialMakefile

03/09/2020 13:51 <DIR> .
03/09/2020 13:51 <DIR> ..
03/09/2020 13:47      827 main.c
03/09/2020 12:44      214 operacoes.c
03/09/2020 12:44      137 operacoes.h
03/09/2020 13:51    54.860 out.exe
                4 arquivo(s)    56.038 bytes
                2 pasta(s) 141.091.139.584 bytes disponíveis

C:\Users\55199\Desktop\TutorialMakefile>out.exe
Executando programa...

Fim do programa
C:\Users\55199\Desktop\TutorialMakefile>
```

Além disso, podemos acrescentar outros comandos para gerar o executável:

option	description
<code>gcc -c</code>	compile source files to object files without linking
<code>gcc -Dname[=value]</code>	define a preprocessor macro
<code>gcc -fPIC</code>	generate position independent code for shared libraries
<code>gcc -glevel</code>	generate debug information to be used by GDB
<code>gcc -Idir</code>	add include directory of header files
<code>gcc -llib</code>	link with library file
<code>gcc -Ldir</code>	look in directory for library files
<code>gcc -o output file</code>	write build output to output file
<code>gcc -Olevel</code>	optimize for code size and execution time
<code>gcc -shared</code>	generate shared object file for shared library
<code>gcc -Uname</code>	undefine a preprocessor macro
<code>gcc -w</code>	disable all warning messages
<code>gcc -Wall</code>	enable all warning messages
<code>gcc -Wextra</code>	enable extra warning messages

Como fazer isso mais rápido e fácil?

RESPOSTA: Makefile!!!

Vamos dar nome aos comandos pra facilitar a execução

1. Crie um arquivo com nome "Makefile"

2. Nesse arquivo crie um target (nesse exemplo é o **all**) para gerar o arquivo binário e coloque os comandos que deseja para gerar o executável dentro dele, também crie outro target (nesse exemplo é o **run**) e coloque o comando para executar o arquivo gerado.

IMPORTANTE: Sempre use a tecla "tab" antes de colocar o comando para funcionar. (Dar espaço não funciona)

```
all:
    gcc main.c -o out

run:
    ./out
```

Obs: Por padrão o comando *make* entra no primeiro target

Agora é só usar os targets no terminal!

1. Use o comando "make" que irá automaticamente ir para o target "all"
2. Use o comando run para rodar o programa

```

C:\Users\55199\Desktop\TutorialMakefile>dir
O volume na unidade C é Windows
O Número de Série do Volume é 4031-2AB0

Pasta de C:\Users\55199\Desktop\TutorialMakefile

03/09/2020  14:57  <DIR>          .
03/09/2020  14:57  <DIR>          ..
03/09/2020  13:47              827 main.c
03/09/2020  14:56              739 Makefile
03/09/2020  12:44              214 operacoes.c
03/09/2020  12:44              137 operacoes.h
                4 arquivo(s)          1.917 bytes
                2 pasta(s) 141.084.483.584 bytes disponíveis

C:\Users\55199\Desktop\TutorialMakefile>make
gcc main.c operacoes.c -o out

C:\Users\55199\Desktop\TutorialMakefile>make run
./out
Executando programa...

fim do programa
C:\Users\55199\Desktop\TutorialMakefile>
```

Obs: Se estiver no cmd do windows o comando "make" do mingw é "mingw32-make", para usar make no terminal você pode ir na pasta bin do mingw e trocar o nome do .exe "mingw32-make" para "make".

Adicione "@" antes do comando que ele não irá aparecer no terminal

```
all:
    @gcc main.c -o out

run:
    @./out
```

```
Prompt de Comando

C:\Users\55199\Desktop\TutorialMakefile>dir
O volume na unidade C é Windows
O Número de Série do Volume é 4031-2AB0

Pasta de C:\Users\55199\Desktop\TutorialMakefile

03/09/2020  14:58    <DIR>        .
03/09/2020  14:58    <DIR>        ..
03/09/2020  13:47             827 main.c
03/09/2020  14:58             741 Makefile
03/09/2020  12:44             214 operacoes.c
03/09/2020  12:44             137 operacoes.h
                4 arquivo(s)          1.919 bytes
                2 pasta(s) 141.083.779.072 bytes disponíveis

C:\Users\55199\Desktop\TutorialMakefile>make

C:\Users\55199\Desktop\TutorialMakefile>make run
Executando programa...

Fim do programa
C:\Users\55199\Desktop\TutorialMakefile>
```

Tá, o que mais posso fazer com isso?

▼ Substituir os comandos com variáveis

```
CC=gcc
SOURCES=main.c operacoes.c
BINARY=out
#Use $(nome da variável)
all:
    $(CC) $(SOURCES) -o $(BINARY)

run:
    ./$(BINARY)
```

▼ Fazer uma mensagem de apresentação

```
CC=gcc
SOURCES=main.c operacoes.c
```

```

BINARY=out

greetings:
    @echo "Welcome to Makefile Tutorial"

all:
    $(CC) $(SOURCES) -o $(BINARY)

run:
    ./$(BINARY)

```

▼ Fazer vários tipos de executáveis

```

CC=gcc
BINARY=out
SOURCES=main.c operacoes.c
INCLUDES=.
LIBRARIES=-lm #Math.h

greetings:
    @echo "Welcome to Makefile Tutorial"

compile0:
    gcc main.c operacoes.c -o out

compile1:
    $(CC) $(SOURCES) -o $(BINARY)

compile2: #Com warnings
    @$(CC) -Wall $(SOURCES) -o $(BINARY)

compile3: #Com gdb
    @$(CC) -g $(SOURCES) -o $(BINARY)
    #gdb out
    #(gdb) run
    #(gdb) quit

compile4: #Adicionando os .h
    @$(CC) $(SOURCES) -o $(BINARY) -I$(INCLUDES)
    #Ao incluir os .h, no seu arquivo .c você pode dar #include usando <> ao invés de ""

compile5: #Linkando lib
    @$(CC) $(SOURCES) -o $(BINARY) $(LIBRARIES)

compile6: #Otimizando
    @$(CC) -O3 $(SOURCES) -o $(BINARY)

run:
    @./$(BINARY)

```


▼ Remover o binário gerado

```
clean:
  @rm $(BINARY)
  #Remove o binário gerado
```

▼ Zipar os arquivos

```
zip:
  @zip -r Operacoes.zip *.c *.h Makefile
  #Gera um zip com os arquivos
```

Links úteis:

Como funciona o Makefile

O Makefile é um arquivo utilizado pelo programa GNU make para instruí-lo como compilar e linkar programas. Esse artigo visa explicar o formato desse arquivo, quais os benefícios e como usá-

 <https://blog.pantuza.com/tutoriais/como-funciona-o-makefile>



GCC C compiler

GCC is a short of GNU Compiler Collection, a C compiler for Linux.

 <https://www.rapidtables.com/code/linux/gcc.html>