

Opção	Descrição
-pass-exit-codes	Sair com o código de erro mais alto de uma fase.
--help	Exibir essas informações.
--target-help	Exibir opções de linha de comando específicas do destino.
use '-v --help'	(Use '-v --help' para exibir opções de linha de comando de subprocessos).
--version	Exibir informações sobre a versão do compilador.
-dumpspecs	Exibir todas as strings de especificação internas.
-dumpversion	Exibir a versão do compilador.
-dumpmachine	Exibir o processador de destino do compilador.
-print-search-dirs	Exibir os diretórios no caminho de pesquisa do compilador.
-print-libgcc-file-name	Exibir o nome da biblioteca companheira do compilador.
-print-file-name=<lib>	Exibir o caminho completo para a biblioteca <lib>.
-print-prog-name=<prog>	Exibir o caminho completo para o componente do compilador <prog>.
-print-multiarch	Exibir o triplo GNU normalizado do destino usado como um componente no caminho da biblioteca.
-print-multi-directory	Exibir o diretório raiz para versões do libgcc.
-print-multi-lib	Exibir o mapeamento entre opções de linha de comando e vários diretórios de pesquisa de biblioteca.
-print-multi-os-directory	Exibir o caminho relativo para bibliotecas do sistema operacional.
-print-sysroot	Exibir o diretório de bibliotecas de destino.
-print-sysroot-headers-suffix	Exibir o sufixo sysroot usado para encontrar cabeçalhos.
-Wa	<options> Passar opções <options> separadas por vírgula para o montador.
-Wp	<options> Passar opções <options> separadas por vírgula para o pré-processador.
-Wl	<options> Passar opções <options> separadas por vírgula para o linker.
-Xassembler <arg>	Passar <arg> para o montador.
-Xpreprocessor <arg>	Passar <arg> para o pré-processador.
-Xlinker <arg>	Passar <arg> para o linker.
-save-temps	Não excluir arquivos intermediários.
-save-temps=<arg>	Não excluir arquivos intermediários.
-no-canonical-prefixes	Não canonicalizar caminhos ao construir prefixos relativos para outros componentes do gcc.
-pipe	Usar pipes em vez de arquivos intermediários.
-time	Cronometrar a execução de cada subprocesso.
-specs=<file>	Substituir as especificações internas pelo conteúdo de <file>.
-std=<standard>	Supor que as fontes de entrada são para <standard>.
--sysroot=<directory>	Usar <directory> como o diretório raiz para cabeçalhos e bibliotecas.
-B <directory>	Adicionar <directory> aos caminhos de pesquisa do compilador.
-v	Exibir os programas invocados pelo compilador.
-###	Como -v mas opções entre aspas e comandos não executados.
-E	Pré-processar apenas; não compilar, montar ou ligar.
-S	Compilar apenas; não montar ou ligar.
-c	Compilar e montar, mas não ligar.

-o <file>	Colocar a saída em <file>.
-pie	Criar um executável independente de posição.
-shared	Criar uma biblioteca compartilhada.
-x <language>	Especificar o idioma dos arquivos de entrada seguintes. Os idiomas permitidos incluem: c c++ assembler none 'none' significa reverter para o comportamento padrão de adivinhar o idioma com base na extensão do arquivo.

## Compilar o Arquivo C/C++ para Código Objeto:

**1. Compilar o arquivo fonte (.c ou .cpp):** Use um compilador como ``gcc`` para arquivos C ou ``g++`` para arquivos C++. Por exemplo:

```
gcc -c arquivo.c -o arquivo.o // Para arquivos C
```

```
g++ -c arquivo.cpp -o arquivo.o // Para arquivos C++
```

Isso criará um arquivo objeto chamado **arquivo.o**.

## Compilar Objetos para Código Executável:

**2. Linkar os objetos:** Depois de ter os arquivos objetos (**.o**), você precisa linká-los para criar o executável final. Use o mesmo compilador e inclua todos os arquivos objetos necessários. Por exemplo:

```
gcc arquivo1.o arquivo2.o -o executavel // Para arquivos C
```

```
g++ arquivo1.o arquivo2.o -o executavel // Para arquivos C++
```

Isso criará um executável chamado executavel.

## Resumo:

- Compile cada arquivo C/C++ individualmente para código objeto usando **-c**.
- Depois de ter todos os arquivos objeto necessários, **link-os** juntos para criar o executável final.