**Etapas de Compilação da Linguagem C**

**Bernardo de Cesaro1, Carlos Rodrigues1, Felipe Tavares1, João Alberto do Nascimento1**

1Escola Politécnica - PUCRS – Porto Alegre – RS – Brasil

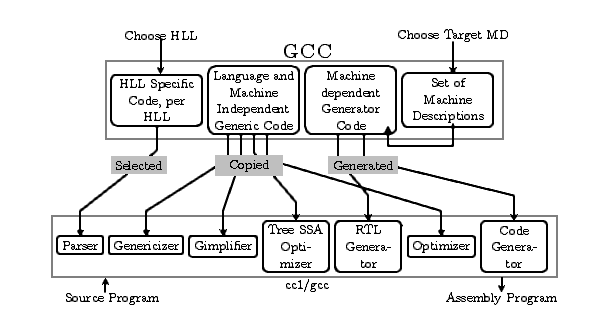
***Resumo.*** *Este artigo apresenta o funcionamento do compilador GCC para a linguagem C, discorrendo cada etapa de compilação, como o formato de arquivo gerado, o comando envolvido no processo, bem como o propósito e composição destes arquivos.*

**1. Etapas de Compilação**

A linguagem de programação C, surgida Sendo assim, apresentamos então as diferentes etapas de compilação da linguagem C, passando por diferentes contextos e etapas, visando comunicar codificação com o mais baixo nível, o hardware.

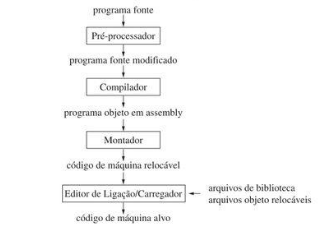
O *GNU Compiler Collection*, ou *GCC*, é o software responsável pela compilação da linguagem C em código executável de máquina. Podemos usar a ferramenta para analisar as diferentes etapas de compilação pelo terminal, observando os arquivos e linguagens geradas a cada instrução, sendo utilizado principalmente para migrar código-fonte para diferentes sistemas.

Figura 1: A estrutura do compilador GCC.

Fonte: The GCC Compiler Generation Framework. (3)

Podemos então distribuir sua compilação em quatro etapas distintas para a geração de código de máquina executável, real sendo elas: pré-processamento, compilação, montagem e ligação.

Figura 2: As etapas de compilação

Fonte: Aho (2008). (1)

**1.1 Pré-processamento**

Nesta etapa o pré-processador lê o código-fonte relacionado e aplica mudanças para que o programa possa ser compilado. Entre as possíveis mudanças estão resolução de diretrizes da linguagem (como o #include), a retirada de comentários do código e a possível otimização do mesmo. Para obtenção do pré-processamento, é utilizado o comando -E seguido do arquivo desejado. Para um melhor entendimento, o comando completo para pré-processar um programa em C seria:

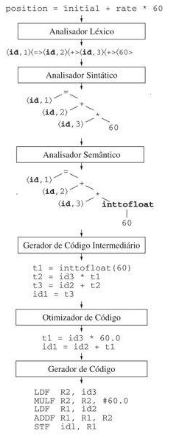
“gcc -E arquivo.c”

A partir desse momento, o programa já está pronto para ser compilado.

**1.2 Compilação**

O compilador tem como função reescrever o programa em um programa equivalente, com isso, efetuar o correto acesso a memória, e organizar os acessos dinâmicos e estáticos. Durante essa etapa onde a aquisição de códigos definidos nos headers (acessos dinâmicos) é realizado. Essa varredura do código é feita em duas etapas distintas, a primeira consiste na análise da estrutura gramatical, onde verifica se existe estruturas mal formadas e erros léxicos. Após essa análise, e a validação transcorrer, é realizada síntese do programa que constitui em criar um fluxo tokens que pode ser representado como uma árvore de prioridades, o conjuto dessas árvores formam o código intermediário. A próxima etapa é otimizar o código intermediário, melhorar acessos, reduzir cálculos aritméticos e alocar a memória evitando uso de variáveis vazias e referências redundantes.

Figura 2: As etapas de compilação



Fonte: Aho (2008). (1)

O comando para realizar a compilação é : “gcc -S arquivo.c”

**1.3 Montagem**

Nesta etapa, em que o código do arquivo em assembly “.s” da etapa anterior é transformado em código de máquina, dando origem ao arquivo objeto, com extensão “.o”. Este arquivo arquivo objeto, com extensão “.o”. Este arquivo contém efetivamente o código que pode ser interpretado pelo processador, mas ainda não é o programa completo, pois ainda é necessário que todos os arquivos sejam ligados, como é descrito na próxima etapa. O comando para realizar a montagem é:

*“*gcc -c arquivo.s*”*

**1.4 Ligação**

A etapa de ligação define justamente o final, a transformação de código em um arquivo executável. Após a fragmentação de código e geração de novos arquivos em diferentes linguagens e contextos, a união destes de forma otimizada resulta então no software descrito em uma linguagem de alto nível. O comando para ligar é:

“gcc -o arquivo.c”

**2. Referências**

1 - Aho. Alfred V., et al. (2008) “Compiladores: Princípios, técnicas e ferramentas”, In: Pearson Education do Brasil. São Paulo, Brasil.

2 - Allain, Alex. “Compiling and linking”. Disponível em: <<https://www.cprogramming.com/compilingandlinking.html>> Acessado em Mar. 2019  
  
3 - Disponível em: <<https://www.cse.iitb.ac.in/grc/intdocs/gcc-basic-info.html>>. Acessado em Mar. 2019

4 - Casavella, Eduardo. “Ciclo de desenvolvimento de um programa em linguagem C”

<<http://linguagemc.com.br/desenvolvendo-um-programa-em-linguagem-c/>> Acessado em: 22:13

O presente documento encontra-se disponível através do seguinte endereço na web:

<https://github.com/nascimentoJoao/gccexplained/>