

Baltazar Alic Borges da Silva – 11711EMT022

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

- Preprocessing

É a primeira fase de tradução que é iniciada por na chamada do compilador. As tarefas comuns realizadas pelo pré-processamento são substituição de macro, teste de diretivas de compilação condicional e inclusão de arquivo. Você pode chamar o pré-processador separadamente para processar texto sem compilar. A saída é um arquivo intermediário, que pode ser inserido para tradução subsequente.

- Compilation Proper

Decodifica o código em C e converte em linguagem de montagem (assembly).

- Assembly

Compila o código assembly para linguagem de máquina.

- Linking

Vincula o programa principal à suas dependências.

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

Resposta no git

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:

Os exemplos estão no git

a) -static

Uma biblioteca estática serve para garantir que os usuários possam executar seu programa porque no caso de bibliotecas compartilhadas, o código que o programa precisa para ser executado é vinculado dinamicamente no tempo de execução, em vez de estaticamente no tempo de compilação. Se a biblioteca compartilhada de que seu programa requer não estiver instalada no sistema do usuário, ela obterá erros e não poderá executar o programa.

b) -g

A opção -g pode ser qualificada com 1, 2 ou 3 para especificar a quantidade de informações de depuração a incluir. O nível padrão é 2 (-g2), que inclui extensas tabelas de símbolos, linha números e informações sobre variáveis locais e externas. Nível 3 de informações de depuração inclui todas as informações de nível 2 e todas as macrodefinições presentes. Nível 1 gera apenas informações

suficientes para criar retrocessos e despejos de pilha. Isso não gerar informações de depuração para variáveis locais ou números de linha.

c) -pedantic

Diz ao gcc para emitir todos os avisos exigidos pelo estrito padrão ANSI / ISO C. Qualquer programa que usa extensões proibidas será rejeitado.

d) -Wall

Emite todos os avisos geralmente úteis que o gcc pode fornecer.

e) -Os

A opção -O diz ao gcc para reduzir o tamanho do código e o tempo de execução, essa opção tem 3 níveis -O1 -O2 e -O3, que deve ser escolhida dependendo do nível de otimização pretendido.

f) -O3

A otimização -O1 inclui saltos de thread e pops de pilha adiados. Otimização de salto de linha tentam reduzir o número de operações de salto; pops de pilha adiados ocorrem quando o compilador permite que os argumentos se acumulem na pilha conforme as funções retornam e então aparecem simultaneamente, em vez de apresentar os argumentos aos poucos à medida que cada função chamada retorna. As otimizações de nível O2 incluem toda a otimização de primeiro nível mais ajustes adicionais que envolvem o agendamento de instruções do processador. Nesse nível, o compilador toma o cuidado de fazer certifique-se de que o processador tem instruções para executar enquanto espera pelos resultados de outras instruções ou latência de dados do cache ou da memória principal. A implementação é altamente específico do processador. As opções de -O3 incluem todas as otimizações de O2, desenrolamento de loop e outros recursos específicos do processador.