UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
Curso de Graduação em Engenharia Mecatrônica
Sistemas Digitais para Mecatrônica

SEMANA 2 (FEELT49081)

Ambiente de Programação Linux

Prof. Eder Alves de Moura

Pedro Otavio Pereira Rangel (11821EMT008)

Uberlândia, março de 2023.

SUMÁRIO

Add Headings (Format > Paragraph styles) and they will appear in your table of contents.

2)

a)

As etapas de compilação são:

1) Pré-processamento: nesta etapa, o pré-processador lê o código fonte e realiza diversas operações, como a inclusão de arquivos de cabeçalho, a expansão de macros e a remoção de comentários. O resultado é um código fonte modificado que será compilado posteriormente.

2) Compilação: nesta etapa, o compilador lê o código fonte pré-processado e produz código objeto. O código objeto é uma forma intermediária do código executável e contém instruções em linguagem de máquina que ainda precisam ser vinculadas com outras bibliotecas e arquivos objeto.

3) Vinculação (Linking): nesta etapa, o vinculador (ou linker) lê os arquivos objeto gerados na etapa anterior e as bibliotecas de sistema e os vincula juntos em um único arquivo executável. Ele resolve referências a funções e variáveis externas e aloca memória para o código e os dados.

4) Geração do executável: nesta etapa, o carregador lê o arquivo executável gerado na etapa anterior e carrega-o na memória do sistema operacional. O código executável agora pode ser executado pelo sistema operacional, realizando as funções para as quais foi projetado.

b)

i) -static: Esse parâmetro faz com que o compilador vincule as bibliotecas necessárias estaticamente ao executável. Isso significa que todas as bibliotecas necessárias serão incorporadas ao executável final, o que tornará o arquivo executável maior, mas não precisará mais das bibliotecas compartilhadas no momento da execução.

ii) -g: Esse parâmetro faz com que o compilador inclua informações de depuração no arquivo executável. Essas informações são usadas por depuradores para rastrear erros no código.

iii) -pedantic: Esse parâmetro faz com que o compilador emita avisos sobre o uso de extensões da linguagem C ou C++. Ele faz com que o compilador

seja mais rigoroso ao verificar o código.

iv) -Wall: Esse parâmetro faz com que o compilador emita avisos para todos os problemas de possível erro que ele encontrar durante a compilação. Isso inclui avisos para coisas como variáveis não inicializadas ou variáveis usadas sem serem declaradas.

v) -Os: Esse parâmetro faz com que o compilador otimize o código para diminuir o tamanho do executável. Ele faz com que o compilador remova informações desnecessárias e otimize o código para reduzir seu tamanho.

vi) -O3: Esse parâmetro faz com que o compilador otimize o código para obter o melhor desempenho possível. Isso inclui otimizações como a reordenação de instruções e a eliminação de código redundante. Esse parâmetro pode aumentar o tempo de compilação, mas geralmente resulta em um código executável mais rápido.