



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA



FEELT49081 – SISTEMAS DIGITAIS PARA MECATRÔNICA

Prática de Sistemas Digitais para Mecatrônica

Prof. Éder Alves de Moura

SEMANA 02:
AMBIENTE DE PROGRAMAÇÃO LINUX

MATHEUS ALVES DE PAULA

11521EMT008

Uberlândia

Dezembro de 2021

Sumário

1. OBJETIVOS.....	3
2. ATIVIDADES.....	4
2.1. ATIVIDADE 1	4
2.2. ATIVIDADE 2	4
2.3. ATIVIDADE 3	4
3. REFERÊNCIAS.....	5

1. OBJETIVOS

Esta atividade prática tem como objetivo introduzir ao aluno o ambiente de programação Linux, onde o aluno irá revisar as ferramentas de compilação em linguagem C no ambiente Linux.

2. ATIVIDADES

2.1. ATIVIDADE 1

As quatro etapas do processo de compilação são apresentadas a seguir.

- Pré-processamento: nessa etapa o pré-processador trabalha com as diretivas `#include` e `#define` através de um arquivo de código-fonte C++. Sendo que a saída desta etapa é um arquivo C++ "puro" sem diretivas de pré-processador.
- Compilação: essa etapa é realizada em cada saída da etapa anterior, sendo que o compilador analisa o código-fonte C++ puro convertendo-o em código de montagem.
- Montagem: nessa etapa o assembler converte o código de montagem em código de máquina no arquivo de objeto.
- Linker: por fim, nessa etapa o vinculador é o que produz a saída final da compilação a partir dos arquivos de objeto que o compilador produziu. Essa saída pode ser uma biblioteca compartilhada (ou dinâmica) ou um executável.

2.2. ATIVIDADE 2

2.3. ATIVIDADE 3

- a) `-static`: esse parâmetro pode ser utilizado para variáveis globais e locais. Sendo que a sua função é indicar que tal variável é permanente.
- b) `-g`: esse parâmetro tem como função solicitar ao compilador e ao vinculador que gerem e mantenham informações de depuração no nível da fonte no próprio executável.
- c) `-pedantic`: a função desse parâmetro é emitir avisos quando encontrar extensões que são proibidas no processo de compilação.
- d) `-Wall`: a função desse parâmetro é encontrar variáveis não inicializadas.
- e) `-Os`: é um parâmetro de otimização de código cuja funcionalidade é reduzir o tamanho do código e tempo de execução.
- f) `-O3`: esse parâmetro inclui todas as otimizações de `O2`, desenrolamento de loop e outros recursos específicos do processador.

3. REFERÊNCIAS

- [1] <https://qastack.com.br/programming/6264249/how-does-the-compilation-linking-process-work>.
- [2] <https://www.embarcados.com.br/modificadores-de-armazenamento-na-linguagem-c/>
- [3] <https://www.ti-enxame.com/pt/c%2B%2B/gcc-g-o-que-acontecera/971861482/>