**Sumário**

[**1. Estrutura do Projeto 2**](#_h9v7vaace9cy)

[**2. Especificações Técnicas 3**](#_e2f9066hqkc3)

[**4. Exemplo de Entrada e Saída 4**](#_d4zmhnuq5wnu)

[**5. Limitações e Melhorias Futuras 4**](#_drc0i58oc79m)

[**6. Referências 5**](#_yqj3m9pnvvow)

Documentação do Interpretador de Instruções Hexadecimal

Disciplina: Arquitetura de Computadores

Objetivo: Ler instruções em hexadecimal de um arquivo, interpretá-las em Assembly x86 e gerar código de máquina correspondente.

# **1. Estrutura do Projeto**

Arquivos

| Arquivo | Função |
| --- | --- |
| main.c | Lê o arquivo instrucoes.txt e chama o interpretador em Assembly. |
| interpretador.asm | Decodifica as instruções hexadecimais e exibe o código de máquina. |
| instrucoes.txt | Arquivo de entrada com instruções em hexadecimal (1 instrução/linha). |

**Fluxo do Programa**

1. Entrada:

* O arquivo instrucoes.txt contém instruções em hexa (ex: 10A2, 2FF0).

2. Processamento:

* O C lê o arquivo e passa cada instrução para o Assembly.
* O Assembly converte o hexa em binário, extrai o opcode e identifica a instrução.

3. Saída:

* Exibe no terminal a instrução Assembly + valor hexadecimal (ex: ADD 0x10A2).

# **2. Especificações Técnicas**

**Instruções Suportadas**

| Opcode (4 bits) | Instrução | Exemplo (hex) |
| --- | --- | --- |
| 0x0 | ADD | 01A3 → ADD |
| 0x1 | SUB | 12F0 → SUB |
| 0x2 | MOV | 20B1 → MOV |
| 0x3 | JMP | 3FF0 → JMP |
| Outros | Inválido | 4ABC → Instrução desconhecida |

**Registradores e Memória**

O código usa registradores EAX, EBX, ECX, EDX para conversão e comparação.

A pilha (stack) é usada para chamadas de função e passagem de parâmetros.

**3. Compilação e Execução**

Linux (GCC + NASM)

**Compilar o Assembly:**

nasm -f elf32 interpretador.asm -o interpretador.o

**Compilar o C e linkeditar:**

gcc -m32 main.c interpretador.o -o interpretador

**Executar:**

./interpretador

# **4. Exemplo de Entrada e Saída**

Arquivo instrucoes.txt

10A2

2FF0

30B1

4ABC

Saída no Terminal

ADD 0x10A2

SUB 0x2FF0

JMP 0x30B1

Instrução desconhecida: 0x4ABC

# **5. Limitações e Melhorias Futuras**

**Limitações Atuais**

- Suporta apenas 4 instruções básicas (ADD, SUB, MOV, JMP).

- Não executa o código de máquina, apenas exibe a interpretação.

**Melhorias Propostas**

1. Ampliar o conjunto de instruções (ex: CMP, XOR, CALL).

2. Implementar um simulador que execute as instruções em um registrador virtual.

3. Adicionar operandos (ex: ADD AX, BX).

# **6. Referências**

Intel x86 Assembly Manual.

NASM Documentation.

"Computer Organization and Design" (David Patterson & John Hennessy).

Nota:

O projeto foi testado em Linux (Ubuntu).

O código está modularizado para facilitar a adição de novas instruções.

Link do repositório:

**GitHub** <https://github.com/hubzinho/Interpretador-asm>