

## Documentação do Assembler Neander

O assembler Neander transforma código assembly em código de máquina para a arquitetura Neander. Este documento descreve a gramática do assembly aceito pelo assembler e o formato do código de máquina gerado.

### Gramática do Assembly Neander

A gramática do assembly aceito pelo assembler Neander é informal e pode ser descrita da seguinte forma:

#### Estrutura Básica

Um programa em assembly Neander consiste em uma sequência de instruções, uma por linha. Cada instrução pode ser precedida opcionalmente por um rótulo.

<programa> ::= <linha>\*

<linha> ::= [<rotulo> ":" ] [<instrucao> [<argumento>]] [<comentario>]

#### Rótulos

Os rótulos são identificadores que representam endereços de memória e podem ser usados como destinos para instruções de salto.

<rotulo> ::= <identificador>

<identificador> ::= [a-zA-Z\_][a-zA-Z0-9\_]\*

#### Instruções

O assembly Neander suporta nove instruções básicas:

<instrucao> ::= "STA" | "LDA" | "ADD" | "SUB" | "NOT" | "JMP" | "JN" | "JZ" | "HLT"

#### Argumentos

As instruções, exceto NOT e HLT, exigem um argumento que pode ser um valor hexadecimal ou um rótulo.

<argumento> ::= <valor\_hex> | <rotulo>

<valor\_hex> ::= [0-9A-F]+

#### Comentários

Comentários começam com um ponto e vírgula (;) e continuam até o final da linha.

<comentario> ::= ";" <qualquer\_caractere>\*

#### Instruções Suportadas

## **Mnemônico Opcode Descrição**

STA	0x10	Armazena o valor do acumulador em um endereço de memória
LDA	0x20	Carrega um valor da memória para o acumulador
ADD	0x30	Adiciona um valor da memória ao acumulador
SUB	0x40	Subtrai um valor da memória do acumulador
NOT	0x60	Inverte os bits do acumulador
JMP	0x80	Salta para um endereço especificado
JN	0x90	Salta se o flag N estiver ativo (acumulador negativo)
JZ	0xA0	Salta se o flag Z estiver ativo (acumulador zero)
HLT	0xF0	Para a execução

## **Formato de Código de Máquina**

O código de máquina gerado pelo assembler tem o seguinte formato:

1. Cabeçalho de 4 bytes (0x52444E03)
2. 512 bytes de dados, onde:
  - Cada instrução ocupa dois bytes na memória:
    - Primeiro byte: código da instrução (opcode)
    - Segundo byte: argumento (endereço ou valor)

## **Exemplos de Código Assembly Válido**

### **Exemplo:**

LDA 10 ; Carrega o valor do endereço 10 para o acumulador

ADD 11 ; Adiciona o valor do endereço 11 ao acumulador

STA 12 ; Armazena o resultado no endereço 12

HLT ; Termina a execução

## **Processo de Montagem**

O assembler Neander realiza a montagem em duas passagens:

1. **Primeira Passagem:**

- Identifica e registra todos os rótulos e seus endereços correspondentes na tabela de símbolos.
- Calcula o tamanho total do programa.

## **2. Segunda Passagem:**

- Gera o código de máquina para cada instrução.
- Resolve referências a rótulos usando a tabela de símbolos criada na primeira passagem.

### **Limitações do Assembler**

1. Máximo de 256 linhas de código.
2. Máximo de 256 bytes de memória endereçável.
3. Rótulos limitados a 31 caracteres.
4. Não suporta macros ou diretivas de assembler complexas.
5. Não suporta expressões aritméticas em argumentos.