

Descrição da Gramática do Assembler NEANDER

Introdução

Este documento descreve a gramática e estrutura compreendida pelo assembler desenvolvido para programas NEANDER. O objetivo deste assembler é interpretar arquivos `.asm` contendo definições de variáveis e instruções de máquina e gerar como saída um arquivo `.mem` binário compatível com o simulador NEANDER da UFRGS.

Seções

O assembler reconhece dois blocos principais em um arquivo `.asm`:

```
.DATA
    <definições de variáveis>
.CODE
    <instruções e diretivas>
```

Gramática da Seção `.DATA`

A seção `.DATA` é usada para definir variáveis nomeadas em posições específicas de memória.

Sintaxe:

`<variavel> DB <valor>`

Regras:

- A palavra-chave `DB` (Define Byte) é obrigatória.
- `<valor>` deve estar em **formato hexadecimal** (ex: `05`, `FF`, etc.).

```
.DATA
A DB 05
B DB 10
```

Gramática da Seção `.CODE`

A seção `.CODE` define o código de máquina propriamente dito.

Instruções válidas:

NOP, STA, LDA, ADD, OR, AND, NOT, JMP, JN, JZ, HLT

Descrição da Gramática do Assembler NEANDER

Sintaxe Geral:

<instrução> <operando>

Observações:

- O operando pode ser:
 - Uma constante numérica hexadecimal (`05`, `0A`, ...)
 - O nome de uma variável definida na seção `.DATA`

```
.CODE  
LDA A  
ADD B  
STA X  
HLT
```

Processamento em Duas Passagens

1. **Primeira Passagem**:

- Lê todos os tokens e determina os endereços de memória para variáveis (labels) e diretivas `ORG`.
- Calcula corretamente o valor do registrador PC com base nas instruções e alocações.

2. **Segunda Passagem**:

- Gera um buffer de 512 bytes com base nas instruções e nas variáveis alocadas.
- Escreve um arquivo `.mem` com o cabeçalho fixo `0x03 0x4E 0x44 0x52` seguido pelo conteúdo da memória.

Formato do Arquivo .mem

- Tamanho total: **516 bytes**
- Início: **4 bytes de cabeçalho fixo**: `0x03 0x4E 0x44 0x52`
- Restante: **512 bytes de memória**
- Cada célula de memória ocupa **2 bytes** (opcode + operando)