

Assembler

O arquivo `assembler.c` implementa um montador (assembler) responsável por traduzir programas escritos em linguagem de montagem (extensão `.asm`) para arquivos binários executáveis (`.bin`). Esses arquivos binários seguem um formato específico, compatível com a arquitetura de um processador simulado — o qual é interpretado por um programa executor (`executor.c`).

A linguagem de montagem suportada possui uma gramática simples, composta por duas seções principais:

- **.DATA**: seção onde são declaradas variáveis e constantes.
- **.CODE**: seção que contém as instruções a serem executadas.

Seção .DATA

Cada linha dessa seção define uma variável no seguinte formato:

nome DB valor

Ou, caso o valor ainda não esteja definido:

nome DB ?

As variáveis recebem endereços de memória automaticamente, com alinhamento de 2 bytes, sendo o valor inicial armazenado na memória caso esteja disponível.

Seção .CODE

Na seção de código, cada linha representa uma instrução da ISA (Instruction Set Architecture), podendo ou não ter um operando. Instruções são convertidas em uma sequência de 4 bytes na memória binária:

- **Byte 1**: Opcode da instrução.
- **Byte 2**: Padding (0x00).
- **Byte 3**: Endereço do operando (se houver).
- **Byte 4**: Padding (0x00).

Labels (rótulos) seguidos de `:` podem ser usados para marcar endereços no código, mas não afetam diretamente a execução — servem apenas como âncoras para instruções de salto.

Processo de Montagem

O processo de montagem ocorre em duas passagens:

1. Primeira Passagem:

- Identifica e armazena todos os rótulos e variáveis.
- Calcula os endereços corretos para cada símbolo.

2. Segunda Passagem:

- Traduz as instruções para seus respectivos opcodes.
- Resolve operandos com base na tabela de símbolos.
- Preenche a memória do programa.

O assembler inicia a memória com um cabeçalho fixo de 4 bytes: `0x03 0x4E 0x44 0x52` (representando "NDR"), seguido das seções de dados e código. A memória total tem tamanho fixo de **512 bytes**.

Instruções Suportadas

| Instrução | Opcode | Descrição |
|------------|--------|---|
| NOP | 0x00 | Não faz nada (operação nula). |
| STA | 0x10 | Armazena o valor do acumulador em uma variável. |
| LDA | 0x20 | Carrega o valor de uma variável para o acumulador. |
| ADD | 0x30 | Soma o valor de uma variável ao acumulador. |
| SUB | 0x31 | Subtrai o valor de uma variável do acumulador. |
| OR | 0x40 | Realiza OR lógico entre o acumulador e uma variável. |
| AND | 0x50 | Realiza AND lógico entre o acumulador e uma variável. |
| NOT | 0x60 | Inverte todos os bits do acumulador (bitwise NOT). |

| | | |
|------------|------|--|
| JMP | 0x80 | Salta incondicionalmente para uma posição de código. |
| JMN | 0x90 | Salta se o valor do acumulador for negativo. |
| JMZ | 0xA0 | Salta se o valor do acumulador for zero. |
| HLT | 0xF0 | Termina a execução do programa. |