

# Conversor de Decimal para Hexadecimal em Assembly

## Objetivo

O programa recebe um número inteiro (máximo 255) digitado pelo usuário em formato decimal e converte esse número para a base hexadecimal, exibindo o resultado como uma string de caracteres no terminal.

## Funcionamento do Código

### Secao .data

``mensagem``: string que solicita ao usuário um número.

``erro``: mensagem de erro caso o número seja invalido.

``hex_table``: tabela com os caracteres hexadecimais (**0-F**).

### Secao .bss

``buffer``: armazena o número digitado pelo usuário (até 3 dígitos e newline).

``resultado``: armazena a string resultante da conversão hexadecimal (até 8 caracteres, mas usamos menos).

``tam_hex``: quantidade de caracteres hexadecimais gerados.

## Etapas do Código

### 1. Impressão da Mensagem Inicial

Usa a syscall ``int 0x80`` com ``eax=4`` para imprimir a string ``mensagem`` na tela.

### 2. Leitura do Numero Decimal

Usa ``int 0x80`` com ``eax=3`` para ler a entrada do usuário para o ``buffer``.

Converte os caracteres ASCII (`'0'-'9'`) em valor decimal:

Multiplica o total anterior por 10 e soma o novo dígito.

### 3. Conversão Decimal Hexadecimal

Divide sucessivamente o número por 16.

Cada resto vira um caractere, obtido por índice em ``hex_table``.

Os caracteres são armazenados na ordem inversa no vetor ``resultado``.

### 4. Impressao do Resultado

Os caracteres são impressos do último para o primeiro, com auxílio do registrador ``esi``.

Cada caractere é copiado para ``resultado + 7``, e então impresso com `syscall`.

### 5. Encerramento

``int 0x80`` com ``eax=1`` encerra o programa.

### Observacoes:

O código foi ajustado para evitar ``Segmentation Fault``, limitando o acesso à memória ao utilizar índices positivos e controle de tamanho.

