

Testes de execução

Atividade II: Assembly MIPS

Disciplina: Organização e Arquitetura de Computadores (2021.1)

Professora: Sarita Mazzini Bruschi

Estudante 1: Gustavo Henrique Brunelli (11801053)

Estudante 2: Mateus Israel Silva (11735042)

Estudante 3: Matheus Henrique de Cerqueira Pinto (11911104)

Estudante 4: Pedro Lucas de Moliner de Castro (11795784)

Instituição: Instituto de Ciências Matemáticas e de Computação – ICMC/USP

São Carlos, 17 de junho de 2021.

Resumo

O presente relatório é referente aos testes de execução do programa de conversão de bases proposto pela segunda atividade da disciplina de Organização e Arquitetura de Computadores. Neste, serão exibidas as execuções de conversão de número das bases binária para binária, binária para decimal, binária para hexadecimal; decimal para binária, decimal para decimal, decimal para hexadecimal; hexadecimal para binária, hexadecimal para decimal e hexadecimal para hexadecimal. Além disso, serão demonstrados os resultados de execução para todos os casos especiais previamente descritos pelo enunciado da atividade.

Palavras-chave: Assembly MIPS, Conversão de bases.

1 Considerações iniciais

1.1 Sobre o código

Buscando manter a `main()` genérica e reduzir o número de `branches`, os seguintes passos foram aplicados:

- a) O número da entrada, independente da base, é lido como uma `string`.
- b) A `string` é convertida para maiúsculo para padronizar a representação de dígitos hexadecimais.
- c) O número passa por uma validação de seus dígitos conforme a base especificada para entrada.
- d) O número é convertido para um decimal representado em `int` a partir da função `para_decimal()` com checagem de overflow no processo.
- e) O decimal é convertido para uma nova `string` construída com a base especificada para a saída a partir da função `decimal_para()`.

Uma decisão importante para a generalização das funções foi a utilização de uma `string` auxiliar `dígitos` equivalente a "0123456789ABCDEF" que representa os possíveis dígitos do número ordenados por valor. A diferenciação da base é feita usando somente a `substring` de tamanho equivalente a base, ou seja, com 2 tem-se "01", com 10 tem-se "0123456789" e com 16 tem-se "0123456789ABCDEF".

A abordagem padronizada de converter o número da base de entrada para decimal e, depois, para a base de saída pode causar algumas execuções redundantes como, por exemplo, na conversão de decimal para decimal. Porém, tal formatação simplifica o código e reaproveita funções em diferentes locais, permitindo uma melhor modularização e concentrando os possíveis tratamentos de erro. Portanto, nesse caso foram priorizadas a legibilidade e simplicidade sobre a eficiência.

1.2 Sobre a construção do relatório

Para a demonstração do correto funcionamento do programa desenvolvido, serão apresentados os *outputs* obtidos no programa MARS. A fim de facilitar as comparações entre bases de origem e destino, foram adotados, para todos os casos, o mesmo conjunto numérico (4294967295, 77 e 7919, em base decimal) para todas as seções.

2 Conversões

2.1 Com binário como base de origem

2.1.1 Para binário

a) 11111111111111111111111111111111

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
11111111111111111111111111111111
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero convertido:
11111111111111111111111111111111
-- program is finished running --
```

b) 1001101

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
1001101
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero convertido:
1001101
-- program is finished running --
```

c) 1111011101111

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
1111011101111
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero convertido:
1111011101111
-- program is finished running --
```

2.1.2 Para decimal

a) 11111111111111111111111111111111

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
11111111111111111111111111111111
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
4294967295
-- program is finished running --
```

b) 1001101

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
1001101
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
77
-- program is finished running --
```

c) 1111011101111

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
1111011101111
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
7919
-- program is finished running --
```

2.1.3 Para hexadecimal

a) 11111111111111111111111111111111

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
11111111111111111111111111111111
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Numero convertido:
FFFFFFFF
-- program is finished running --
```

b) 1001101

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
1001101
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Numero convertido:
4D
-- program is finished running --
```

c) 1111011101111

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
1111011101111
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Numero convertido:
1EEF
-- program is finished running --
```

2.2 Com decimal como base de origem

2.2.1 Para binário

a) 4294967295

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
4294967295
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero convertido:
11111111111111111111111111111111
-- program is finished running --
```


b) 77

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
77
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero convertido:
1001101
-- program is finished running --
```

c) 7919

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
7919
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero convertido:
1111011101111
-- program is finished running --
```

2.2.2 Para decimal

a) 4294967295

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
4294967295
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
4294967295
-- program is finished running --
```

b) 77

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
77
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
77
-- program is finished running --
```

c) 7919

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
7919
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
7919
-- program is finished running --
```

2.2.3 Para hexadecimal

a) 4294967295

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
4294967295
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Numero convertido:
FFFFFFFF
-- program is finished running --
```

b) 77

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
77
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Numero convertido:
4D
-- program is finished running --
```

c) 7919

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
7919
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Numero convertido:
1EEF
-- program is finished running --
```

2.3 Com hexadecimal como base de origem

2.3.1 Para binário

a) FFFFFFFF

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
FFFFFFF
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero convertido:
11111111111111111111111111111111
-- program is finished running --
```

b) 4D

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
4D
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero convertido:
1001101
-- program is finished running --
```

c) 1EEF

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
1EEF
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero convertido:
1111011101111
-- program is finished running --
```

2.3.2 Para decimal

a) FFFFFFFF

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
FFFFFFFF
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
4294967295
-- program is finished running --
```

b) 4D

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
4D
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
77
-- program is finished running --
```

c) 1EEF

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
1EEF
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
7919
-- program is finished running --
```

2.3.3 Para hexadecimal

a) FFFFFFFF

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
FFFFFFF
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Numero convertido:
FFFFFFF
-- program is finished running --
```

b) 4D

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
4D
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Numero convertido:
4D
-- program is finished running --
```


c) 1EEF

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
1EEF
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Numero convertido:
1EEF
-- program is finished running --
```

3 Casos especiais

3.1 Erro na letra que indica a base

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
C
Opcao invalida!
J
Opcao invalida!
A
Opcao invalida!
B
Digite o numero a ser convertido:
10001
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
17
-- program is finished running --
```

3.2 Números não representáveis na base escolhida

a) Binária

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
9
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero invalido!
-- program is finished running --
```

b) Decimal

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
9F
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero invalido!
-- program is finished running --
```

c) Hexadecimal

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
9J
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Numero invalido!
-- program is finished running --
```

3.3 Hexadecimal com letras minúsculas

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
H
Digite o numero a ser convertido:
4d
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Numero convertido:
77
-- program is finished running --
```

3.4 Números maiores do que 32 bits

a) Base binária

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:
1111111111111111111111111111111111
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Overflow detectado!
-- program is finished running --
```

b) Base decimal

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
4294967296
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Overflow detectado!
-- program is finished running --
```

c) Base hexadecimal

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Digite o numero a ser convertido:
4294967296
Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
D
Overflow detectado!
-- program is finished running --
```

3.5 Número vazio

```
Selecione a base do numero de entrada:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Digite o numero a ser convertido:

Selecione a base do numero de saida:
[B] Binario
[D] Decimal
[H] Hexadecimal
B
Numero invalido!
-- program is finished running --
```