

Universidade de São Paulo
ICMC - Instituto de Ciências Matemáticas e Computação

SSC0902 - Organização e Arquitetura de Computadores (2021/1)
20 de junho de 2021

Redigido por:

Natan Henrique Sanches (11795680) - Bel. em Ciências da Computação

Paulo Henrique de Souza Soares (11884713) - Bel. em Ciências da Computação

Osni Brito de Jesus (11857330) - Bel. em Ciências da Computação

Alvaro José Lopes (10873365) - Bel. em Ciências da Computação

CONVERSOR ENTRE BASES NUMÉRICAS

Uma prática dirigida em Assembly MIPS

1 Introdução

O atual projeto, de caráter pedagógico, consistiu na construção de um conversor entre as principais bases numéricas utilizadas no âmbito da computação (binária, decimal e hexadecimal) utilizando para sua natureza a linguagem montadora *Assembly* relativa à arquitetura MIPS.

2 Objetivo

O presente relatório possui o objetivo de redigir acerca do primeiro projeto prático da disciplina SSC0902 - Organização e Arquitetura de Computadores, explicando a construção e expondo os resultados satisfatórios obtidos a partir da finalização do projeto. Além disso, a elaboração do projeto contribuiu de maneira considerável para o entendimento do fluxo de uma linguagem montadora, bem como o trabalho direto com registradores e memória-pilha.

3 Ferramentas utilizadas

- Simulador MARS MIPS, da Universidade de Missouri.

4 Modelagem prática

O conversor de bases pôde ser confeccionado utilizando dos artifícios da linguagem montadora da arquitetura MIPS, através principalmente do manuseio direto de registradores para efetuar os cálculos aritméticos.

O programa permite a conversão de números da base binária, decimal ou hexadecimal para alguma outra desse grupo, independentemente do fluxo. Sua construção também conta com o tratamento de erros para o eventual caso de dígitos ou bases inválidos informados de entrada, além da análise de um possível *overflow* caso o valor informado seja muito grande.

De maneira geral, tornou-se preferível que o número fosse lido como uma cadeia de caracteres (`ascii`) e, posteriormente, fosse convertido para decimal independentemente da base de entrada. Com o valor decimal do número em mãos e interpretando-o como um número inteiro, podemos realizar a conversão para a base desejada de maneira fácil.

5 Resultados

Como proposto, o programa converte qualquer fluxo de bases dentre as disponíveis (Figuras 1, 2, 3, 4, 5).

```
Bem vindo ao conversor de base!  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de origem: d  
Digite o número para conversão: 1024  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de destino: b  
O número convertido é: 1000000000  
-- program is finished running --
```

Figura 1: Conversão de um número decimal para um número binário

```
Bem vindo ao conversor de base!  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de origem: b  
Digite o número para conversão: 101011  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de destino: d  
O número convertido é: 43  
-- program is finished running --
```

Figura 2: Conversão de um número binário para um número decimal

```
Bem vindo ao conversor de base!  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de origem: H  
Digite o número para conversão: FFF  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de destino: B  
O número convertido é: 1111111111  
-- program is finished running --
```

Figura 3: Conversão de um número hexadecimal para um número binário

```
Bem vindo ao conversor de base!  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de origem: H  
Digite o número para conversão: A43EF  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de destino: d  
O número convertido é: 672751  
-- program is finished running --
```

Figura 4: Conversão de um número hexadecimal para um número decimal

```
Bem vindo ao conversor de base!  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de origem: b  
Digite o número para conversão: 1000101010  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de destino: H  
O número convertido é: 22A  
-- program is finished running --
```

Figura 5: Conversão de um número binário para um número hexadecimal

Além disso, é detectado quando o usuário propõe uma entrada inválida/incompatível com os padrões que o programa foi estruturado, seja ela bases inexistentes nesse contexto, dígitos inválidos para a base requisitada ou números que gerarão po-

tenciais *overflows* (i.e. números maiores do que $2^{32} - 1$), como exemplificado nas figuras 6, 7, 8 e 11.

```
Bem vindo ao conversor de base!  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de origem: K  
  
Erro: Dado de entrada inválido. Programa finalizado retornando 1.  
  
-- program is finished running --
```

Figura 6: Identificação do erro a partir da informação de uma base numérica inexistente

```
Bem vindo ao conversor de base!  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de origem: b  
Digite o número para conversão: 102  
  
Erro: Dado de entrada inválido. Programa finalizado retornando 1.  
  
-- program is finished running --
```

Figura 7: Identificação do erro a partir de dígitos inválidos para a base binária

```
Bem vindo ao conversor de base!  
Opções: (B)inário      (D)ecimal      (H)exadecimal  
Digite a letra correspondente à base numérica de origem: H  
Digite o número para conversão: 9M737AF  
  
Erro: Dado de entrada inválido. Programa finalizado retornando 1.  
  
-- program is finished running --
```

Figura 8: Identificação do erro a partir de dígitos inválidos para a base hexadecimal

```
Bem vindo ao conversor de base!
Opções: (B)inário      (D)ecimal      (H)exadecimal
Digite a letra correspondente à base numérica de origem: D
Digite o número para conversão: 999999999999999999999999
Opções: (B)inário      (D)ecimal      (H)exadecimal
Digite a letra correspondente à base numérica de destino: B

Erro: Dado de entrada excede o limite permitido.

-- program is finished running --
```

Figura 9: Identificação de *overflow* no número informado de entrada

```
Bem vindo ao conversor de base!
Opções: (B)inário      (D)ecimal      (H)exadecimal
Digite a letra correspondente à base numérica de origem: a

Erro: Dado de entrada inválido. Programa finalizado retornando 1.

-- program is finished running --
```

Figura 10: Identificação da primeira opção inválida

```
Bem vindo ao conversor de base!
Opções: (B)inário      (D)ecimal      (H)exadecimal
Digite a letra correspondente à base numérica de origem: b
Digite o número para conversão: 1010
Opções: (B)inário      (D)ecimal      (H)exadecimal
Digite a letra correspondente à base de destino: a

Erro: Dado de entrada inválido. Programa finalizado retornando 1.

-- program is finished running --
```

Figura 11: Identificação da segunda opção inválida

6 Conclusão

A atividade prática cumpriu com efeito seu objetivo de familiarizar o grupo com artifícios de linguagem montadora, tais como seu fluxo de trabalho, funcionamento e suas instruções, a manipulação de registradores diretos e a escrita/carregamento de informações da memória-pilha. Apesar de restrito à arquitetura MIPS, o conhecimento adquirido pode ser aplicado para outras arquiteturas.

7 Bibliografia

Referências

- [1] MIPS Quick Tutorial. Encontrado em https://minnie.tuhs.org/CompArch/Resources/mips_quick_tutorial.html.
- [2] D .A. Patterson and J. L. Hennessy. *Computer Organization and Design - The Hardware Interface*. Elsevier, fifth edition, 2014.