

PROJETO INTERDISCIPLINAR

Conversão de Base

Conversão de base decimal para base binária,
octadecimal e hexadecimal.

Alunos:

RGM	Nome
25653130	Giovanni Silva Cerqueira
25647466	Guilherme Jorge da Silva

UNIVERSIDADE CRUZEIRO DO SUL

PROJETO INTERDISCIPLINAR

Conversão de Base

Conversão de base decimal para base binária,
octadecimal e hexadecimal.

Trabalho apresentado como parte do requisito para
aprovação na Disciplina de Projeto Interdisciplinar do curso
de Tecnologia em Análise e Desenvolvimento de Sistemas
da Universidade Cruzeiro do Sul.

Orientadores: Prof. Douglas Almendro e Prof. Marco
Sanches Antonio

São Paulo

2021

Sumário

1. Apresentação:	Erro! Indicador não definido.
1.1 Justificativa e Motivação	Erro! Indicador não definido.
1.2 Dados do Programa.	Erro! Indicador não definido.
2 Requisitos de Programação de Computadores.....	Erro! Indicador não definido.
3 Requisitos de Organização e Arquitetura de Computadores	7
4 Consideração finais.....	9
5 BIBLIOGRAFIA	10

1. APRESENTAÇÃO:

1.1 Justificativa e Motivação

O projeto de conversão de bases numéricas tem como justificativa colocar em prática os conhecimentos adquiridos ao longo do semestre, oferecendo desafios à altura de um curso de Análise e Desenvolvimento de Sistemas, pois reúne, de uma só vez, habilidades adquiridas em diversas áreas da computação, em especial aquelas voltadas à programação e organização e arquitetura de computadores, além de incentivar a pesquisa.

1.2 Dados do Programa

O que ele faz: Realiza a conversão da base decimal para as bases hexadecimal, octal e binária.

Para que serve: para representar em diferentes valores a quantidade definida na forma de representação.

Qual a relevância da utilização desse programa: no mundo da tecnologia digital os dispositivos eletrônicos trabalham em baixo nível com a base numérica binária, pois os números binários são facilmente representados na eletrônica através de pulsos elétricos. Além desses dois, as bases numéricas octal e hexadecimal também são muito utilizadas pela fácil representação.

2. Requisitos de Programação de Computadores

A linguagem utilizada foi o Python 3.9.2

```
print("Converta um numero decimal para outra base\n")
base = input("Qual base deseja converter? \n [1]Binário \n [2]Octodecimal \n [3]Hexadecimal \n Digite a opção: ")
```

O início do programa/código possui um “print” que imprime na tela “Converta um número decimal para outra base”. Posteriormente há um input que armazenará na variável “base” o número que o usuário digitar, que poderá ser “1”, “2” ou “3”, esses números fazem parte do menu que será apresentado para ele. O 1 representa a base binária, o 2 a base octodecimal e o 3 a base hexadecimal.

Com base nisso entra a condição “if” que verifica qual número para definir o próximo passo do código:

```
if base == "1" or base == "2" or base == "3":
```

Se a base for igual a 1, entrará essa condição:

```
if base == "1":
    num1 = int(input("Digite o numero : "))
    resto = "."
    while num1 > 0:
        resto = str(num1 % 2) + resto
        num1 = int(num1 // 2)
    print(f'o numero é {resto}')
```

Nesse momento do código impresso na tela, o input “Digite o número: ”, o usuário digitará o valor que ele deseja converter para binário. O valor será armazenado na variável “num1”. Após isso foi criada e iniciada a variável “resto” com o valor “.”.

Com essas informações é o momento de fazer a conversão, para isso foi utilizado um while, que fará um loop até o valor de “num1” chegar a 0 e dessa forma o programa ficará convertendo o valor digitado do usuário até o último dígito, com o seguinte cálculo:

```
resto = str(num1 % 2) + resto
num1 = num1 // 2
```

A variável “resto” armazenará o valor do resto da divisão de “num1” dividido por 2, converte o resultado desse cálculo em uma string e em seguida insere junto com o valor armazenado anteriormente dentro da variável “resto”, por exemplo:

No primeiro loop foi armazenado o valor 5, no próximo loop, o resultado deste cálculo será armazenado junto com esse 5, ficando 54, pois o valor desse cálculo é convertido em string e é só feito a concatenação.

Após isso é feito a divisão do “num1” por 2 e armazenado o valor do inteiro dessa divisão dentro da própria variável e será dessa forma que o valor de “num1” chegará até 0, interrompendo o loop.

Por último, é impresso na tela o valor da variável “resto”, que nesse já terá os valores de todas as conversões concatenados.

```
elif base == "2":
    num1 = int(input("Digite o numero : "))
    resto = "."
    while num1 > 0:
        resto = str(num1 % 8) + resto
        num1 = int(num1 // 8)
    print(f'o numero é {resto}')
```

Para a base “octadecimal”, representada pelo “2”, é feito o mesmo procedimento, apenas substituindo o valor da divisão por 8.

```

elif base == "3":

    num1 = int(input("Digite o numero : "))
    resto = "."

    while num1 > 0:
        verificacao = (num1 % 16)
        if verificacao == 10:
            letra = "A"
            resto = letra + resto
            num1 = int(num1 // 16)
        elif verificacao == 11:
            letra = "B"
            resto = letra + resto
            num1 = int(num1 // 16)
        elif verificacao == 12:
            letra = "C"
            resto = letra + resto
            num1 = int(num1 // 16)
        elif verificacao == 13:
            letra = "D"
            resto = letra + resto
            num1 = int(num1 // 16)
        elif verificacao == 14:
            letra = "E"
            resto = letra + resto
            num1 = int(num1 // 16)
        elif verificacao == 15:
            letra = "F"
            resto = letra + resto
            num1 = int(num1 // 16)
        else:
            resto = str(num1 % 16) + resto
            num1 = int(num1 // 16)

    print(f'o numero é {resto}')

```

A base hexadecimal, representada pelo “3”, é um pouco mais complexa, pois é necessário substituir os valores, 10, 11, 12, 13, 14, 15, por A, B, C, D, E, F, respectivamente.

Para isso, foi inserido dentro do while a variável “verificação”, que terá a função de verificar se o resto da divisão do “num1” terá um dos valores 10, 11, 12, 13, 14, 15. Para ser feito essa verificação, foi criando uma condição, que se verifica for igual a um desses valores, ela entrará na condição correspondente e fará o seguinte procedimento:

Será gerado uma variável que armazenará a letra correspondente ao número atendido pela condição, por exemplo, se foi 12, será gerado a variável “letra”, com o valor “C”

Após isso é feito o cálculo:

resto = letra + resto

E caso o resto da divisão não seja nenhum desses números que exigem essa conversão é feito o cálculo padrão:

$\text{resto} = \text{str}(\text{num1} \% 16) + \text{resto}$

Desta forma em cada loop será verificado se será necessário a conversão pela letra ou não, se sim, o programa entra dentro da condição correspondente a essa letra e faz o cálculo informado, caso contrário é feito o cálculo padrão e no final é inserido as letras e os números, gerando o resultado esperado através do “print” da mesma forma das outras bases.

3. Requisitos de Organização e Arquitetura de Computadores

Sistemas de numeração é um conjunto de símbolos usados para representar quantidades e regras que definem a forma de representação. Cada sistema de numeração é apenas um método diferente de representar quantidades. Essas quantidades são sempre as mesmas, o que mudam são os símbolos que as representam. Dentro de sistemas de numeração temos notação posicional e não posicional.

Não posicional: o valor de um símbolo não se altera independente da posição onde ele está, o que muda é o valor final dependendo da posição desses símbolos. Exemplo: algarismo romano.

Posicional: o valor de um símbolo se altera dependendo da posição onde ele está, mudando assim o valor da quantidade. O valor total é a soma dos valores relativos de cada algarismo. Exemplo: 754 fica 754-700-50-4.

Todos os sistemas numéricos posicionais são identificados por uma base.

Os seguintes sistemas de numeração posicionais são alvos deste estudo:

Sistema decimal – possui base 10 e faz uso de dez símbolos para representar todas as quantidades.

Sistema binário – tem base 2 e é representado pelos símbolos 0 e 1.

Sistema octal – possui base 8 e utiliza os algarismos 0 a 7.

Sistema hexadecimal – possui base 16 e para representar as quantidades faz uso dos números 0 a 9 e das letras A a F que correspondem aos números 10 a 15.

Conversão de base numérica

Conversão de base numérica é a passagem da representação de um número de uma base numérica para outra, alterando a simbologia para se adequar à nova base. A base que normalmente usamos é a decimal, pois contém dez algarismos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Por exemplo, o número inteiro representado em base decimal como 14, pode ser escrito como '1110' em base binária ou 'E' em base hexadecimal.

Conversão da base decimal para uma base qualquer

A conversão de um número decimal para uma base qualquer consiste em realizar divisões sucessivas e tomar os restos dessas divisões. O divisor usado é o valor da base de destino e os restos das divisões inteiras são a sequência de algarismos da base de destino. Para definir a ordem dos algarismos do resultado da operação, o primeiro resto encontrado, corresponderá ao dígito menos significativo e o último resto encontrado corresponderá ao dígito mais significativo. Exemplo:

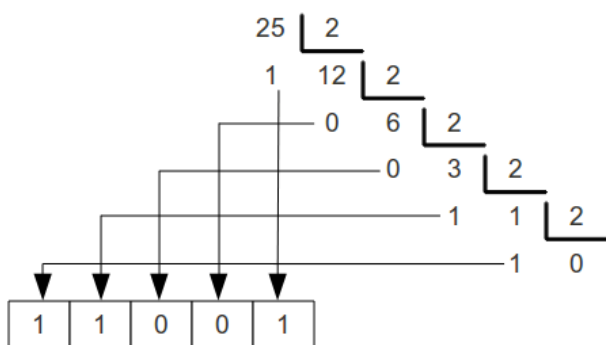


Figura 1. Conversão de base decimal para binária.

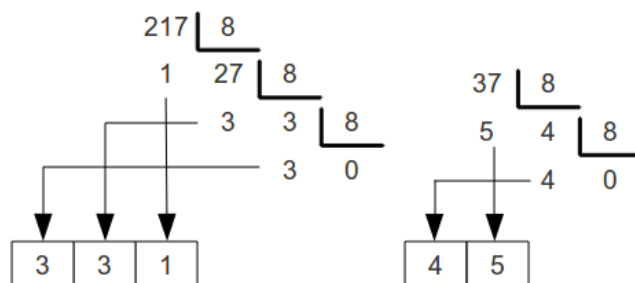


Figura 2. Conversão de base decimal para octadecimal.

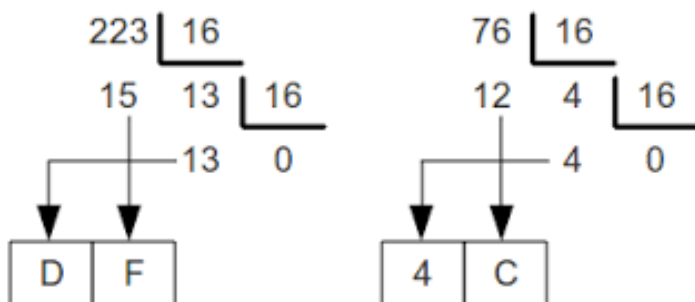


Figura 3. Conversão de base decimal para hexadecimal.

4. CONSIDERAÇÕES FINAIS

Sabíamos como fazer a conversão da base decimal para outras bases através dos cálculos aprendidos nas aulas de Organização e Arquitetura de Computadores. Além disso, tivemos o conhecimento necessário, adquirido na matéria de Programação de Computadores, para utilizar as indentações dentro da linguagem de programação Python a fim de realizar os cálculos de conversão.

Por outro lado, unir a lógica do cálculo matemático de conversão de bases dentro da lógica de programação utilizando o Python foi uma dificuldade encontrada em meio ao processo. Tivemos de testar e experimentar diferentes caminhos, muitos deles errôneos, para chegar ao ponto onde a lógica do cálculo e a lógica do programa, unidas, dessem o resultado esperado. Mas especificamente a conversão para base hexadecimal foi um desafio ainda maior, visto que a contagem numérica vai até 9, e a partir deste número, são utilizadas letras de A a F, representando os números de 10 a 15. Sendo assim, era necessário que o resultado da conversão dentro do programa mostrasse letras para representar estes números. Levamos um tempo considerável para chegarmos à solução deste problema, mas felizmente deu tudo certo.

5. BIBLIOGRAFIA

ALMENDRO, Douglas. **Programação de Computadores**. Universidade Cruzeiro do Sul, São Paulo, 2021.

ANTONIO, Marco Sanches. **Organização e Arquitetura de Computadores**. Universidade Cruzeiro do Sul, São Paulo, 2021.