Operações Aritméticas Bit a Bit

Mário Leite

...

As quatro operações aritméticas: Adição, Subtração, Multiplicação e Divisão são as primeiras operações básicas da Matemática que todos aprendem a fazer depois de conhecer os dígitos numéricos da Base Decimal. Mas existem outros tipos de operações que envolvem dígitos de outras bases numéricas, em especial a **Base Binária**, cujos dígitos são apenas dois: **0** e **1**. Nesta base pode-se até fazer uma "gracinha" dizendo que **1 + 1 = 10**. Na verdade o resultado NÃO é **dez**; o correto é dizer "**um zero**"; assim como a adição **1 + 1 + 1 = 1 + 10 = 11**; estranho e engraçado, né! MAS não é **"um mais dez igual a onze**"... entenda: houve apenas um deslocamento do bit **1** e o resultado NÃO é onze; é **"um e um**"...
Na verdade, esta operação de adição é feita "bit a bit" envolvendo os tais "deslocamentos" de bits binários (sem redundância) e que na operação "normal" de adição é o nosso conhecido "**vai 1**" quando atingimos a base 10 ou múltiplos dela: "**vai 2"**, **"vai 3"**, ...). Assim, em operações binárias, quando o valor atingido na soma dá **2** tem que aplicar o **"vai 1"** adicionando esse **1** à coluna de soma à esquerda, tal como se faz nas operações na base decimal. Fazer operações binárias (sejam aritméticas ou lógicas) não é complicado, mas pode se tornar bem trabalhoso. O programa **"OperacoesArtimeticasBinarias"** (codificado em Python e C#) resolve esse problema de maneira prática. As **figuras 1** e **2** mostram saídas relativas aos dois códigos, respectivamente.

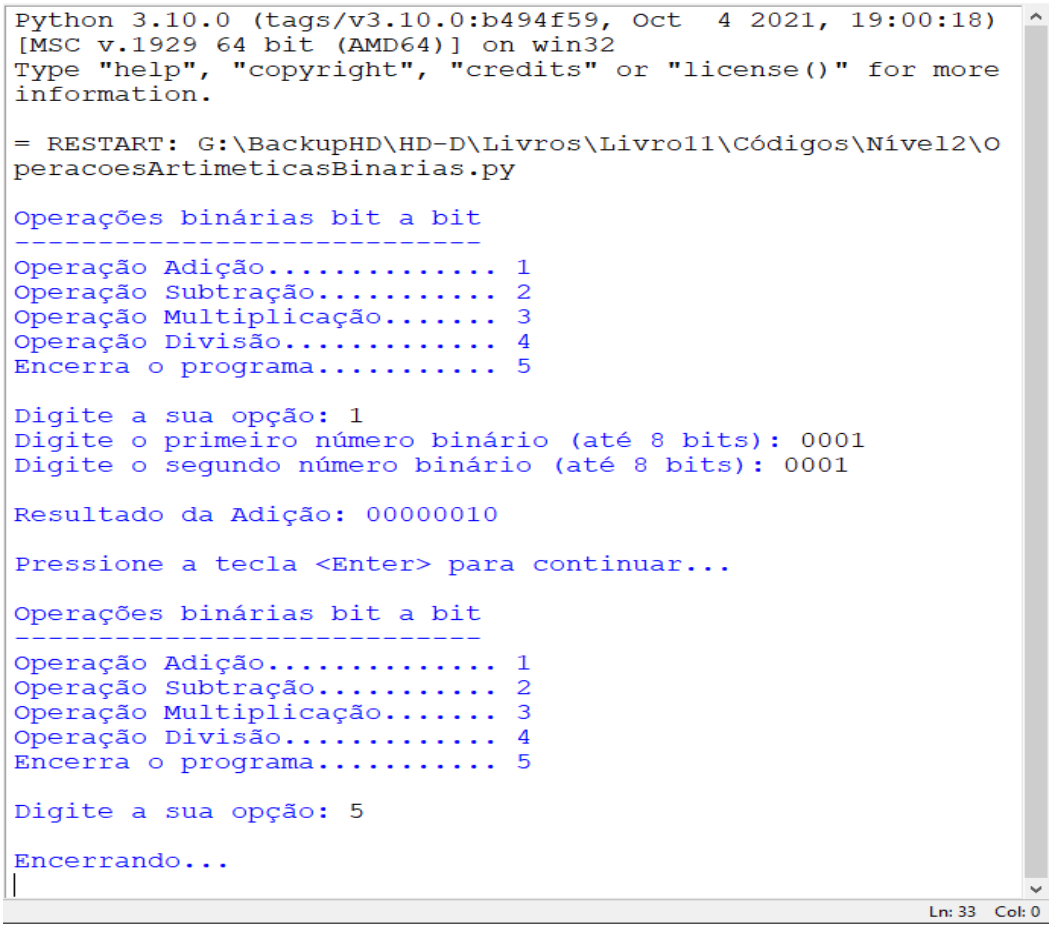

Figura 1 - Saída do programa em Python

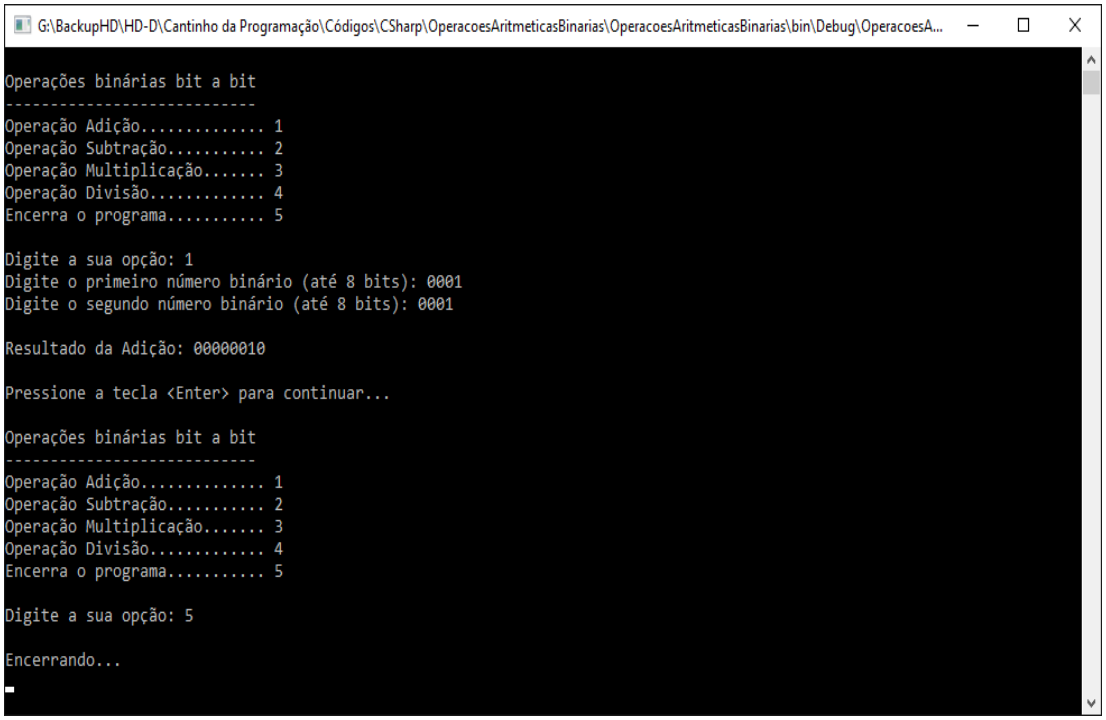

Figura 2 - Saída do programa em C#

```
'''
OperacoesArtimeticasBinarias.py
------------------------------------------------------------------------------
Lê dois números binários (máximo oito bits) e faz as operações Aritméticas:
Adição, Subtração, Multiplicação e Divisão, bit a bit.
------------------------------------------------------------------------------
Data: 17/09/2023
Autor: Mário Leite
'''
import time

TamBin = 8  #define o padrão da palavra para 1 byte (8 bits)

def FazerAdicao(bin1, bin2):
    carry = 0
    resultado = ""

    #Certifica se ambos os números tenham o mesmo comprimento
    while(len(bin1) < TamBin):
        bin1 = "0" + bin1
    while(len(bin2) < TamBin):
        bin2 = "0" + bin2

    for i in range(TamBin - 1, -1, -1):
        bit1 = int(bin1[i])
        bit2 = int(bin2[i])
        soma = bit1 + bit2 + carry
        resultado = str(soma % 2) + resultado
        carry = soma // 2

    return resultado
```

```
#-------------------------------------------------------------------------
def FazerSubtracao(bin1, bin2):
    emprestimo = 0
    resultado = ""
    #Certifica que ambos os números tenham o mesmo comprimento
    while(len(bin1) < TamBin):
        bin1 = "0" + bin1
    while(len(bin2) < TamBin):
        bin2 = "0" + bin2
    for i in range(TamBin - 1, -1, -1):
        bit1 = int(bin1[i])
        bit2 = int(bin2[i])
        diferenca = bit1 - bit2 - emprestimo
        if(diferenca < 0):
            diferenca += 2
            emprestimo = 1
        else:
            emprestimo = 0

        resultado = str(diferenca) + resultado
    return resultado
#-------------------------------------------------------------------------------
def FazerMultiplicacao(bin1, bin2):
    num1 = int(bin1, 2)
    num2 = int(bin2, 2)
    resultado = bin(num1 * num2)[2:]

    if(len(resultado) > TamBin):
        resultado = resultado[-TamBin:]
    while(len(resultado) < TamBin):
        resultado = "0" + resultado
    return resultado
#-------------------------------------------------------------------------------
def FazerDivisao(bin1, bin2):
    if not (ValidarBinario(bin1) and ValidarBinario(bin2)):
        return "Número(s) binário(s) inválido(s)"
    #Certifica se ambos os números tenham o mesmo comprimento
    while(len(bin1) < (len(bin2))):
           bin1 = "0" + bin1
    while(len(bin2)) < (len(bin)):
           bin2 = "0" + bin2
    quociente = ""
    resto = "0"  #começa com um "0" à direita
    for i in range(len(bin1)):
        dividend = resto + bin1[i]
        if(dividend >= bin2):
            quociente += "1"
            resto = bin(int(dividend, 2) - int(bin2, 2))[2:]
        else:
            quociente += "0"
            resto = dividend
    #Remova os zeros à esquerda no quociente
    quociente = quociente.lstrip("0")
    #Se o quociente estiver vazio significa que a divisão é exata
    if(quociente == ""):
        quociente = "0"
    return quociente
```

```python
#-------------------------------------------------------------------------
def ValidarBinario(bin_str):
    if((len(bin_str) < 4) or (len(bin_str)) > TamBin):
        return False
    for bit in bin_str:
        if((bit != "0") and (bit != "1")):
            return False
    return True
#=========================================================================================
#Programa principal
while(True):
    print("\nOperações binárias bit a bit")
    print("----------------------------")
    print("Operação Adição.............. 1")
    print("Operação Subtração........... 2")
    print("Operação Multiplicação....... 3")
    print("Operação Divisão............. 4")
    print("Encerra o programa........... 5")
    print()

    op = abs(int(input("Digite a sua opção: ")))

    if(op == 5):
        print("\nEncerrando...")
        time.sleep(3)  #aguarda três segundos
        break

    if((op < 1) or (op > 4)):
        print("\nOpção inválida. Tente novamente.")
        continue

    bin1 = input("Digite o primeiro número binário (até 8 bits): ")
    bin2 = input("Digite o segundo número binário (até 8 bits): ")

    if((not(ValidarBinario(bin1)) and (ValidarBinario(bin2)))):
        print("\nNúmero(s) binário(s) inválido(s). Tente novamente.")
        continue

    if(op == 1):
        resultado = FazerAdicao(bin1, bin2)
        print("\nResultado da Adição:", resultado)

    elif(op == 2):
        resultado = FazerSubtracao(bin1, bin2)
        print("\nResultado da Subtração:", resultado)

    elif(op == 3):
        resultado = FazerMultiplicacao(bin1, bin2)
        print("\nResultado da Multiplicação:", resultado)

    elif(op == 4):
        resultado = FazerDivisao(bin1)
        print("\nResultado da Divisão por:", resultado)

    input("\nPressione a tecla <Enter> para continuar...")
#Fim do programa "OperacoesArtimeticasBinarias.py" --------------======-----
```

```
/*
OperacoesArtimeticasBinarias.py
-------------------------------------------------------------------------------
Lê dois números binários (máximo oito bits) e faz as operações Aritméticas:
Adição, Subtração, Multiplicação e Divisão, bit a bit.
-------------------------------------------------------------------------------
Data: 17/09/2023
Autor: Mário Leite
*/
//-------------------------------------------------------------------------------
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

public class OperacoesAritmeticasBinarias
{
    private const int TamBin = 8;

    private static string FazerAdicao(string bin1, string bin2)
    {
        int carry = 0;
        string resultado = "";

        while (bin1.Length < TamBin)
            bin1 = "0" + bin1;
        while (bin2.Length < TamBin)
            bin2 = "0" + bin2;

        for (int i = TamBin - 1; i >= 0; i--)
        {
            int bit1 = int.Parse(bin1[i].ToString());
            int bit2 = int.Parse(bin2[i].ToString());

            int soma = bit1 + bit2 + carry;
            resultado = (soma % 2) + resultado;
            carry = soma / 2;
        }

        if (carry == 1)
            resultado = "1" + resultado;

        return resultado;
        Console.ReadKey();
    }
    //-------------------------------------------------------------------------------
    private static string FazerSubtracao(string bin1, string bin2)
    {
        int emprestimo = 0;
        string resultado = "";

        while (bin1.Length < TamBin)
            bin1 = "0" + bin1;
        while (bin2.Length < TamBin)
            bin2 = "0" + bin2;

        for (int i = TamBin - 1; i >= 0; i--)
        {
            int bit1 = int.Parse(bin1[i].ToString());
            int bit2 = int.Parse(bin2[i].ToString());
```

```
            int diferenca = bit1 - bit2 - emprestimo;
            if (diferenca < 0)
            {
                diferenca += 2;
                emprestimo = 1;
            }
            else
            {
                emprestimo = 0;
            }
            resultado = diferenca + resultado;
        }

        return resultado;
    }
    //-----------------------------------------------------------------------------
    private static string FazerMultiplicacao(string bin1, string bin2)
    {
        int num1 = Convert.ToInt32(bin1, 2);
        int num2 = Convert.ToInt32(bin2, 2);
        string resultado = Convert.ToString(num1 * num2, 2);

        if (resultado.Length > TamBin)
            resultado = resultado.Substring(resultado.Length - TamBin);

        while (resultado.Length < TamBin)
            resultado = "0" + resultado;
        return resultado;
    }
    //-----------------------------------------------------------------------------
    private static string FazerDivisao(string bin1, string bin2)
    {
        if (!ValidarBinario(bin1) || !ValidarBinario(bin2))
            return "Números binários inválidos";

        while (bin1.Length < bin2.Length)
            bin1 = "0" + bin1;
        while (bin2.Length < bin1.Length)
            bin2 = "0" + bin2;

        string quociente = "";
        string resto = "0";

        for (int i = 0; i < bin1.Length; i++)
        {
            string dividend = resto + bin1[i];

            if (Convert.ToInt32(dividend, 2) >= Convert.ToInt32(bin2, 2))
            {
                quociente += "1";
                resto = Convert.ToString(Convert.ToInt32(dividend, 2) -
                        Convert.ToInt32(bin2, 2), 2);
            }
            else
            {
                quociente += "0";
                resto = dividend;
            }
        }
        quociente = quociente.TrimStart('0');
        if (string.IsNullOrEmpty(quociente))
            quociente = "0";
        return quociente;
    }
```

```
//-------------------------------------------------------------------==========
private static bool ValidarBinario(string binStr)
{
    if (binStr.Length < 4 || binStr.Length > TamBin)
        return false;

    foreach (char bit in binStr)
    {
        if (bit != '0' && bit != '1')
            return false;
    }
    return true;
}
//=============================================================================
public static void Main(string[] args)
{
    while (true)
    {
        Console.WriteLine("\nOperações binárias bit a bit");
        Console.WriteLine("---------------------------");
        Console.WriteLine("Operação Adição.............. 1");
        Console.WriteLine("Operação Subtração........... 2");
        Console.WriteLine("Operação Multiplicação....... 3");
        Console.WriteLine("Operação Divisão............. 4");
        Console.WriteLine("Encerra o programa........... 5");
        Console.WriteLine();

        Console.Write("Digite a sua opção: ");
        int op = Math.Abs(int.Parse(Console.ReadLine()));

        if (op == 5)
        {
            Console.WriteLine("\nEncerrando...");
            System.Threading.Thread.Sleep(3000); // aguarda três segundos
            break;
        }

        if (op < 1 || op > 4)
        {
            Console.WriteLine("\nOpção inválida. Tente novamente.");
            continue;
        }

        Console.Write("Digite o primeiro número binário (até 8 bits): ");
        string bin1 = Console.ReadLine();

        if (bin1.Length > TamBin || !IsBinary(bin1))
        {
            Console.WriteLine("\nNúmero binário inválido. Tente novamente.");
            Console.Write("\nPressione a tecla <Enter> para continuar...");
            Console.ReadLine();
            continue;
        }
        Console.Write("Digite o segundo número binário (até 8 bits): ");
        string bin2 = Console.ReadLine();

        if (bin2.Length > TamBin || !IsBinary(bin2))
        {
            Console.WriteLine("\nNúmero binário inválido. Tente novamente.");
            Console.Write("\nPressione a tecla <Enter> para continuar...");
            Console.ReadLine();
            continue;
        }
        string resultado = "";
```

```
            if (op == 1)
            {
                if (ValidarBinario(bin1) && ValidarBinario(bin2))
                {
                    resultado = FazerAdicao(bin1, bin2);
                    Console.WriteLine($"\nResultado da Adição: {resultado}");
                }
                else
                    Console.WriteLine("\nNúmero(s) binário(s) inválido(s). Tente
                                       novamente.");
            }
            else if (op == 2)
            {
                if (ValidarBinario(bin1) && ValidarBinario(bin2))
                {
                    resultado = FazerSubtracao(bin1, bin2);
                    Console.WriteLine($"\nResultado da Subtração: {resultado}");
                }
                else
                    Console.WriteLine("\nNúmero(s) binário(s) inválido(s). Tente
                                       novamente.");
            }
            else if (op == 3)
            {
                if (ValidarBinario(bin1) && ValidarBinario(bin2))
                {
                    resultado = FazerMultiplicacao(bin1, bin2);
                    Console.WriteLine($"\nResultado da Multiplicação: {resultado}");
                }
                else
                    Console.WriteLine("\nNúmero(s) binário(s) inválido(s). Tente
                                       novamente.");
            }
            else if (op == 4)
            {
                if (ValidarBinario(bin1) && ValidarBinario(bin2))
                {
                    resultado = FazerDivisao(bin1, bin2);
                    Console.WriteLine($"\nResultado da Divisão: {resultado}");
                }
                else
                    Console.WriteLine("\nNúmero(s) binário(s) inválido(s). Tente
                                       novamente.");
            }

            Console.Write("\nPressione a tecla <Enter> para continuar...");
            Console.ReadLine();
        }
    }
    //---------------------------------------------------------------------------
    private static bool IsBinary(string input)
    {
        foreach (char c in input)
        {
            if (c != '0' && c != '1')
                return false;
        }
        return true;
    }
} //Fim do programa "OperacoesArtimeticasBinarias".cs--------------------------
```