

Bases Numéricas e Lógica Booleana

1.Sistema Binário

Não é segredo que na computação e na linguagem das máquinas num geral, o seu funcionamento só é possível pela interpretação de comandos na linguagem binária, ou seja, 0 e 1. As máquinas utilizam dessa linguagem para representar, processar e armazenar dados, chamamos os conjuntos formados de 0 e 1 de bits que é a menor unidade de medida de informação em um computador, o conjunto de 8 bits forma 1 byte, que é a unidade de medida mais comum para representar dados em computadores.

2.Sistema Decimal

O sistema decimal, no entanto, não é amplamente utilizado pelo computador da mesma maneira da linguagem binária, é muito utilizada por nós no dia a dia e conta com a representação numérica de (0-9). Entretanto, os usuários não interagem com seus aparelhos diretamente na linguagem binária, usamos a decimal e a linguagem natural e o que ocorre dentro da máquina é uma conversão durante o processamento das ações.

3.Sistema Octal

O sistema octal surge como uma alternativa para diminuir a representação de um número binário que pode vir a ser muito extenso e claro também a facilitar a manipulação humana no sistema. É um sistema considerado antigo e já não tão utilizado, podendo ser encontrado em maquinários mais antigos.

Cada dígito octal equivale a 3 bits e para converter um número decimal para octal, é preciso dividir o valor decimal por 8 de forma progressiva, obtendo um resultado e um resto. O resultado de cada iteração será sempre menor que 7. Para converter um número octal para decimal, é preciso multiplicar cada dígito pelo seu valor de posição e somar os resultados.

4.Sistema Hexadecimal

E por fim o sistema Hexadecimal, é o mais utilizado atualmente por programadores pois é a forma mais compacta de representar números binários. Cada dígito hexadecimal representa exatamente 4 bits, ou seja, 1 "nibble" (metade de 1 byte). É bastante utilizada pois como dito anteriormente os números binários podem vir a ser muitíssimo extensos, o sistema vem justamente para facilitar e simplificar. Um exemplo dessa utilização é a aplicação dos múltiplos de 16 para indicar capacidade de processamento de memórias (32 bits ou 64 bits).

5.Utilização nas Cidades Inteligentes

Com base na definição e maior meio de utilização dos sistemas acima, conseguimos implementar eles numa realidade de uma cidade inteligente. A começar pelo sistema decimal que como já foi dito é a principal ponte de utilização entre o usuário e o sistema, aplicando numa realidade nem todos os operadores e gestores de uma cidade inteligente teriam tempo de realizar o processo de conversão, sem falar na praticidade para todos os cidadãos.

Já o sistema binário é indispensável, afinal é a forma como os dados são armazenados e processados internamente pelos computadores e dispositivos eletrônicos. E é a nível binário que geralmente ocorre a comunicação entre sensores e controladores, podemos citar como exemplo o controle dos semáforos, onde 0 para representar vermelho e 1 para verde, para otimizar o tráfego. Tão indispensável quanto, o sistema hexadecimal pode vir a ajudar na eficiência na transmissão de dados, sistemas binários e hexadecimais ajudam a reduzir a largura de banda necessária para transmitir dados. Como os dispositivos trabalham com binário, a comunicação é mais eficiente, e a conversão para hexadecimal facilita a interpretação e o gerenciamento de endereços, melhorando a eficiência e a confiabilidade das redes de sensores e dispositivos conectados. Ao falar de cidades inteligentes falamos de versatilidade e pluralidade por isso até mesmo a linguagem octal vem a ser utilizada em dispositivos heterogêneos as vezes de cunho analógico.

6. Modelo de Conversão

A fim de aprimorar e colocar em prática o conhecimento teórico acima, foi desenvolvido um modelo de conversão de *string* na linguagem de programação *python* em suas formas binária, octal e hexadecimal.

```
1 def codificar_frase(frase):
2     # Criar listas para armazenar os resultados binário, octal e hexadecimal
3     binario = []
4     octal = []
5     hexadecimal = []
6
7     # Para cada caractere na frase, obtemos o código ASCII e o convertemos
8     for char in frase:
9         # Código ASCII do caractere
10        codigo_ascii = ord(char)
11
12        # Convertendo para binário (remover o prefixo '0b')
13        binario.append(bin(codigo_ascii)[2:].zfill(8)) # zfill(8) para garantir 8 bits
14
15        # Convertendo para octal (remover o prefixo '0o')
16        octal.append(oct(codigo_ascii)[2:].zfill(3)) # zfill(3) para garantir 3 dígitos
17
18        # Convertendo para hexadecimal (remover o prefixo '0x' e tornar em maiúsculo)
19        hexadecimal.append(hex(codigo_ascii)[2:].upper().zfill(2)) # zfill(2) para garantir 2 dígitos
20
21    # Resultados
22    return {
23        'binario': ' '.join(binario),
24        'octal': ' '.join(octal),
25        'hexadecimal': ' '.join(hexadecimal)
26    }
27
28    # Exemplo de uso
29    frase = input("Digite uma frase: ")
30    codificacao = codificar_frase(frase)
31
32    print("Frase original:", frase)
33    print("\nCodificação Binária:")
34    print(codificacao['binario'])
35    print("\nCodificação Octal:")
36    print(codificacao['octal'])
37    print("\nCodificação Hexadecimal:")
38    print(codificacao['hexadecimal'])
```

O código pede ao usuário que digite uma frase e ela é convertida aos sistemas citados, onde é possível se ter uma ideia de como funciona esse processo nas máquinas.

Referências

Napoleon- O que é: Binary Number System?

Meu Artigo - O Software Binário e a Matemática (Parte I)

Infonova - Diferença entre sistemas 32 bits e 64 bits