U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69215 ALLAN MORINAGA
19FFU109 - 1 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciências sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10 Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10 A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2 A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16 A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  034159      3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $10000111_{(2)} =?_{(10)}$
2. $1155_{(16)} =?_{(10)}$
3. $158_{(10)} =?_{(2)}$
4. $286_{(10)} =?_{(2)}$
5. $7045_{(10)} =?_{(16)}$
6. $2642_{(10)} =?_{(16)}$
7. $100011000_{(2)} + 100101001_{(2)} =?_{(2)}$
8. $11101000_{(2)} + 11011001_{(2)} = ?_{(2)}$
9. $EF6_{(16)} + 1F14_{(16)} =?_{(16)}$
10. $1C78_{(16)} + 7FE_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69822 ANDREA A ALVES
19FFU109 - 2 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciências sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal

Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício

Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10

Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} = ?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10

A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} = ?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2

A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} = ?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16

A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} = ?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário

trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal

A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  034159      3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $10111011_{(2)} = ?_{(10)}$
2. $121D_{(16)} = ?_{(10)}$
3. $287_{(10)} = ?_{(2)}$
4. $299_{(10)} = ?_{(2)}$
5. $7185_{(10)} = ?_{(16)}$
6. $7847_{(10)} = ?_{(16)}$
7. $110010100_{(2)} + 11110111_{(2)} = ?_{(2)}$
8. $11101110_{(2)} + 101110001_{(2)} = ?_{(2)}$
9. $109F_{(16)} + 1E4E_{(16)} = ?_{(16)}$
10. $400_{(16)} + B48_{(16)} = ?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69084 EDSON RODRIGUES VELOSO
19FFU109 - 3 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciênciais sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal

Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício

Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10

Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10

A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2

A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16

A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário

trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal

A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $11111010_{(2)} =?_{(10)}$
2. $C79_{(16)} =?_{(10)}$
3. $299_{(10)} =?_{(2)}$
4. $227_{(10)} =?_{(2)}$
5. $9381_{(10)} =?_{(16)}$
6. $9073_{(10)} =?_{(16)}$
7. $101001000_{(2)} + 100110100_{(2)} =?_{(2)}$
8. $10110110_{(2)} + 111000010_{(2)} = ?_{(2)}$
9. $12C5_{(16)} + 1DC7_{(16)} =?_{(16)}$
10. $5D7_{(16)} + 2204_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69222 ELAINE GREBOGY (OUV)
19FFU109 - 4 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciênciais sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal

Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício

Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10

Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10

A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2

A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16

A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário

trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal

A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $11100110_{(2)} =?_{(10)}$
2. $FB3_{(16)} =?_{(10)}$
3. $285_{(10)} =?_{(2)}$
4. $200_{(10)} =?_{(2)}$
5. $6207_{(10)} =?_{(16)}$
6. $2667_{(10)} =?_{(16)}$
7. $101111111_{(2)} + 110001110_{(2)} =?_{(2)}$
8. $110000010_{(2)} + 100010111_{(2)} =?_{(2)}$
9. $1C2F_{(16)} + 2339_{(16)} =?_{(16)}$
10. $14F7_{(16)} + 18ED_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69091 FAGNER FERREIRA
19FFU109 - 5 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciências sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10 Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10 A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2 A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16 A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $11100101_{(2)} =?_{(10)}$
2. $125B_{(16)} =?_{(10)}$
3. $169_{(10)} =?_{(2)}$
4. $263_{(10)} =?_{(2)}$
5. $1423_{(10)} =?_{(16)}$
6. $3156_{(10)} =?_{(16)}$
7. $111101011_{(2)} + 1000101010_{(2)} =?_{(2)}$
8. $110100001_{(2)} + 111011001_{(2)} =?_{(2)}$
9. $20E0_{(16)} + 2562_{(16)} =?_{(16)}$
10. $1D9D_{(16)} + 1ABA_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69103 FELLIPE RICHTER NEVES
19FFU109 - 6 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciências sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal

Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício

Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10

Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10

A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2

A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16

A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário

trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal

A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  034159      3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $11100010_{(2)} =?_{(10)}$
2. $ABB_{(16)} =?_{(10)}$
3. $285_{(10)} =?_{(2)}$
4. $116_{(10)} =?_{(2)}$
5. $1791_{(10)} =?_{(16)}$
6. $3612_{(10)} =?_{(16)}$
7. $101011010_{(2)} + 111101001_{(2)} =?_{(2)}$
8. $110100011_{(2)} + 1110110_{(2)} = ?_{(2)}$
9. $1D41_{(16)} + 2019_{(16)} =?_{(16)}$
10. $616_{(16)} + 1972_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69110 GABRIEL MELEK DOS SANTOS
19FFU109 - 7 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciênciais sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10 Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10 A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2 A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16 A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $11101101_{(2)} =?_{(10)}$
2. $1349_{(16)} =?_{(10)}$
3. $132_{(10)} =?_{(2)}$
4. $165_{(10)} =?_{(2)}$
5. $3072_{(10)} =?_{(16)}$
6. $8259_{(10)} =?_{(16)}$
7. $10010111_{(2)} + 10110111_{(2)} =?_{(2)}$
8. $100000110_{(2)} + 100001110_{(2)} =?_{(2)}$
9. $14E1_{(16)} + 206C_{(16)} =?_{(16)}$
10. $501_{(16)} + 165D_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69127 GABRIEL ROBERTO DA ROSA
19FFU109 - 8 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciênciais sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal

Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício

Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10

Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10

A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0 + C\times16^1 + 2\times16^2 + 0\times16^3 = 10\times16^0 + 12\times16^1 + 2\times16^2 + 0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2

A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16

A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário

trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal

A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $10111000_{(2)} =?_{(10)}$
2. $D19_{(16)} =?_{(10)}$
3. $175_{(10)} =?_{(2)}$
4. $246_{(10)} =?_{(2)}$
5. $5832_{(10)} =?_{(16)}$
6. $4140_{(10)} =?_{(16)}$
7. $101000000_{(2)} + 100001001_{(2)} =?_{(2)}$
8. $10111100_{(2)} + 11001111_{(2)} = ?_{(2)}$
9. $1CF9_{(16)} + 13A6_{(16)} =?_{(16)}$
10. $448_{(16)} + 22F2_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69134 GABRIEL SOUZA BRAGA
19FFU109 - 9 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciências sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10 Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10 A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2 A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16 A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $100010010_{(2)} =?_{(10)}$
2. $10FD_{(16)} =?_{(10)}$
3. $183_{(10)} =?_{(2)}$
4. $158_{(10)} =?_{(2)}$
5. $4101_{(10)} =?_{(16)}$
6. $5271_{(10)} =?_{(16)}$
7. $101010111_{(2)} + 101101110_{(2)} =?_{(2)}$
8. $111111101_{(2)} + 110000101_{(2)} =?_{(2)}$
9. $C40_{(16)} + 19B1_{(16)} =?_{(16)}$
10. $76D_{(16)} + 1780_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69141 GUILHERME SAITO BANDEIRA
19FFU109 - 10 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciênciais sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10 Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10 A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2 A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16 A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $11010011_{(2)} =?_{(10)}$
2. $12ED_{(16)} =?_{(10)}$
3. $178_{(10)} =?_{(2)}$
4. $181_{(10)} =?_{(2)}$
5. $4419_{(10)} =?_{(16)}$
6. $5543_{(10)} =?_{(16)}$
7. $11101100_{(2)} + 11100110_{(2)} =?_{(2)}$
8. $110100001_{(2)} + 10010011_{(2)} =?_{(2)}$
9. $1093_{(16)} + 1426_{(16)} =?_{(16)}$
10. $1DB7_{(16)} + 1B8C_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69158 HARON AURELIANO TRAVASSOS
19FFU109 - 11 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciênciais sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal

Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício

Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10

Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10

A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2

A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16

A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário

trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal

A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE     166A4
+ 00765B   + 22BB7
--------   -------
  34159      3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $11010110_{(2)} =?_{(10)}$
2. $1035_{(16)} =?_{(10)}$
3. $186_{(10)} =?_{(2)}$
4. $217_{(10)} =?_{(2)}$
5. $9789_{(10)} =?_{(16)}$
6. $6716_{(10)} =?_{(16)}$
7. $111101011_{(2)} + 1000011001_{(2)} =?_{(2)}$
8. $101111000_{(2)} + 101000010_{(2)} =?_{(2)}$
9. $2039_{(16)} + 1AE1_{(16)} =?_{(16)}$
10. $1397_{(16)} + 710_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69239 ICLEIA SANTOS (OUV)
19FFU109 - 12 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciênciais sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal

Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício

Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10

Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10

A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2

A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16

A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário

trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal

A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $100000101_{(2)} =?_{(10)}$
2. $116B_{(16)} =?_{(10)}$
3. $218_{(10)} =?_{(2)}$
4. $169_{(10)} =?_{(2)}$
5. $7521_{(10)} =?_{(16)}$
6. $1770_{(10)} =?_{(16)}$
7. $111100111_{(2)} + 101011111_{(2)} =?_{(2)}$
8. $1000000110_{(2)} + 101100100_{(2)} =?_{(2)}$
9. $FD1_{(16)} + 2096_{(16)} =?_{(16)}$
10. $136E_{(16)} + 1B1D_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69246 MAURICIO KOBAY DO AMARAL
19FFU109 - 13 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciênciais sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal

Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício

Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10

Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10

A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2

A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16

A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário

trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal

A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $11010001_{(2)} =?_{(10)}$
2. $DDD_{(16)} =?_{(10)}$
3. $138_{(10)} =?_{(2)}$
4. $108_{(10)} =?_{(2)}$
5. $5247_{(10)} =?_{(16)}$
6. $8578_{(10)} =?_{(16)}$
7. $10011111_{(2)} + 10011010_{(2)} = ?_{(2)}$
8. $111100110_{(2)} + 11110100_{(2)} = ?_{(2)}$
9. $20F3_{(16)} + 2129_{(16)} =?_{(16)}$
10. $1E5A_{(16)} + 1241_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69253 PEDRO HENNIG
19FFU109 - 14 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciências sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal

Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10

Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10

A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2

A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16

A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal

A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  034159      3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $10100001_{(2)} =?_{(10)}$
2. $F85_{(16)} =?_{(10)}$
3. $129_{(10)} =?_{(2)}$
4. $240_{(10)} =?_{(2)}$
5. $7120_{(10)} =?_{(16)}$
6. $3696_{(10)} =?_{(16)}$
7. $1000101111_{(2)} + 11000100_{(2)} =?_{(2)}$
8. $11100110_{(2)} + 11010000_{(2)} = ?_{(2)}$
9. $261E_{(16)} + B6E_{(16)} =?_{(16)}$
10. $2196_{(16)} + 2013_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69165 RAFAEL DO NASCIMENTO BARBOSA
19FFU109 - 15 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciênciais sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10 Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10 A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2 A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16 A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $1110000_{(2)} =?_{(10)}$
2. $B55_{(16)} =?_{(10)}$
3. $191_{(10)} =?_{(2)}$
4. $297_{(10)} =?_{(2)}$
5. $1850_{(10)} =?_{(16)}$
6. $7659_{(10)} =?_{(16)}$
7. $1000011000_{(2)} + 11011011_{(2)} =?_{(2)}$
8. $100101111_{(2)} + 10111001_{(2)} = ?_{(2)}$
9. $128D_{(16)} + 2005_{(16)} =?_{(16)}$
10. $2013_{(16)} + 264E_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69172 RODRIGO PORTANTIOLO WAGNER
19FFU109 - 16 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciênciais sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10 Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10 A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2 A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16 A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $100100110_{(2)} =?_{(10)}$
2. $D9A_{(16)} =?_{(10)}$
3. $262_{(10)} =?_{(2)}$
4. $270_{(10)} =?_{(2)}$
5. $9123_{(10)} =?_{(16)}$
6. $7054_{(10)} =?_{(16)}$
7. $1000111001_{(2)} + 100010101_{(2)} =?_{(2)}$
8. $10001100_{(2)} + 111000001_{(2)} = ?_{(2)}$
9. $F56_{(16)} + CED_{(16)} =?_{(16)}$
10. $1C56_{(16)} + 26CC_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

U Positivo UTFPR PUCPr
16/03/2019 - 11:22:21.4
Prof Dr P Kantek
(pkantek@up.edu.br)
Aritmética decimal, binária e hexadecimal VIVO036b V: 1.01
69189 WILSON DE MELO
19FFU109 - 17 apos 18/04, 50%
/ _____ / _____

Aritmética Decimal, binária e Hexadecimal

A aritmética tal como a conhecemos se desenvolveu a partir dos últimos 10.000 anos. Nesse aspecto, a matemática é privilegiada. Estivéssemos falando de ciências sociais ou mesmo de tecnologia aplicada, estaríamos tratando do último modismo, ou da última invenção. Mas, a matemática é eterna. Pitágoras e Euclides continuam tão verdadeiros e frescos (no bom sentido, por favor) como eram há 2400 anos. Aliás, se algum dia, descobrirmos uma civilização extraterrena, com biologia completamente distinta, seres com 7 braços e 3 pernas que vivem a bilhões de anos luz da Terra, pode-se garantir: o Teorema de Pitágoras lá continua valendo: $a^2 + b^2 = c^2$ ainda que eventualmente tenha outro nome e outros símbolos, talvez o Pitágoras deles seja um camarão inteligente. É batata!

A primeira grande invenção na aritmética foi o número zero (devido a matemáticos indianos). Com ele criou-se um sistema de notação posicional. A base escolhida foi a decimal, muito provavelmente devido a termos 10 dedos, já que estes são os primeiros e muito efetivos instrumentos de contagem de que dispõe o ser humano.

Eis os fatos sabidos até aqui: nosso sistema de numeração é decimal, (base=10), posicional e usa os dígitos arábicos 0,1,...9. Uma curiosidade: embora diversos povos do planeta usem distintos alfabetos para a escrita, (cirílico, katakana, hangul, árabe, grego,...) todos os principais parecem concordar com os dígitos arábicos para escrever números. Assim o número 1003 é 1003 em qualquer rincão culto da Terra.

A mecânica da contagem é usar os 9 primeiros dígitos, começando no um e ao chegar ao 10, que é quando se completa uma base, e se acabam os dígitos unitários, escreve-se o um (o primeiro) seguido de um zero.

Aqui, a primeira novidade. Nenhuma generalidade se perde (e se ganha espaço) se a origem das contagens for o zero e não o um. Dentro da ciência da computação, em muitos momentos a origem da contagem será o zero. Fiquemos espertos.

Quando, os engenheiros ingleses construiram o primeiro computador de que se tem notícia, seguindo os planos teóricos do matemático Alan Turing (esse sim, provavelmente foi um extraterrestre...) estiveram diante de uma decisão tecnológica: como representar uma quantidade dentro de uma máquina ?

A primeira hipótese foi fazer uma analogia com um circuito elétrico e variar alguma grandeza física (elétrica) proporcionalmente à quantidade em estudo. Aliás alguns computadores analógicos foram assim construídos. Imprecisão, dificuldades imensas na programação, e baixa generalidade foram fatais a este projeto. O caminho estava em outro lugar: em computadores digitais. Aliás, esta tendência que começou na década de 40, está hoje entre nós na digitalização de imagens, sons, filmes, livros, raios-x, relógios etc etc.

Ao construir uma variável digital (ao invés de uma analógica) é possível controlar sua exatidão, garantir ausência de erros e obter a precisão tão grande quanto se queira. A pergunta agora é: que tipo de variável ? A tentação é criar uma variável decimal (com 10 estados digitais). Mais fácil de falar do que fazer. Imaginem um circuito que a 0V indica 0, 1V indica 1, ... até 9V indica 9. Basta cair um raio, haver uma interferência e um 7 vira 9 fácil, fácil. A opção foi radical: variáveis binárias, contendo apenas dois valores, e completamente opostos um do outro. Digamos que 0 é +5V e 1 é −5V. Ao cair um raio, um 5V pode virar 4V ou até 3V, mas o circuito é suficiente "esperto" para não se confundir: 3V está muito mais perto de 5V do que de −5V.

Assim, chega-se à aritmética binária: a base de todos os computadores digitais (e mais genericamente de toda a eletrônica digital). Toda a circuitaria de um computador está construída assim. Deixando de lado os detalhes sórdidos (como 5V e −5V) vamos passar a usar a aritmética binária: apenas dois dígitos 0 e 1.

Continuam valendo TODAS as regras da aritmética: adição, produto, subtração, divisão, logaritmo, etc etc.

Acompanhe no exemplo: 0+0=0, 0+1=1, 1+0=1 e 1+1=0 e vai um ou seja, 1+1=10. Resta um problema: o tamanho dos números: Por exemplo, o número $55786803_{(10)}$ é igual a $11010100110011110100110011_{(2)}$. Um número em base 2 contém em média $log10 \div log2 = 3,32$ algarismos a mais do em base 10. Nada que assuste, principalmente tendo em vista a aumento explosivo dos limites máximos de memória de qualquer circuitozinho.

Por tudo isso, a unidade básica de registro (chamado, algo incongruentemente de memória) em um computador é o BIT (BInary digiT), que como vimos pode conter 0 ou 1. Usa-se abreviar o BIT por um "b" minúsculo. Devido à pobreza desse elemento individual, os bits são agrupados em unidades maiores, chamadas BYTES e compostas por 8 bits. Usa-se abreviar o BYTE por um "B" maiúsculo. Deve-se este conceito ao engenheiro americano Claude Shanon. Se, em um bit só podem ser armazenados dois valores, em um byte, podem-se guardar $2^8 = 256$ estados, devidamente numerados de 0 até 255.

Sistema Hexadecimal

Os dígitos binários são tudo o que o computador consegue manusear. Mas, vimos que se torna cansativo e sujeito a erros para o ser humano, manipular cadeias enormes de zeros e uns. A solução para isso, está em usar um segundo sistema, o hexadecimal. A palavra hexadecimal tem a ver com a base=16. Ressalte-se que ele é uma modificação do sistema binário (que insisto: é o único que o computeador entende), criada exclusivamente para facilitar a vida dos humanos que têm que pensar em binário. Essa simplificação consiste em trocar 4 dígitos binários por 1 único dígito hexadecimal. Em um sistema de base 16, vai-se precisar de 16 dígitos. Na falta de melhor opção, vão-se usar os 10 dígitos numéricos (0..9) e depois 6 letras começando no A (A..F).

Antes de mostrar as tabelas, vamos fazer uma contagem:

em decimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

em binário 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100 ,1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, ...

em hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ...

Agora, uma tabela de conversão:

decimal	binário	hexad.
0	0000.0000	00
1	0000.0001	01
2	0000.0010	02
9	0000.1001	09
10	0000.1010	0A
15	0000.1111	0F
16	0001.0000	10
255	1111.1111	FF

Perceba-se que dentro de um byte (8 bits) é costume pôr um ponto separando as duas metades do byte. É mais ou menos o mesmo que escrever 34.567 ao invés de 34567. Facilita a leitura.

Exercício

Construa a tabela acima para os 256 primeiros valores. Comece em 0 e termine em 255.

Conversão de base 2 para base 10

Multiplique cada dígito pelas potências crescente de 2, começando à esquerda com 2^0 e some o resultado. Acompanhe no exemplo:

Quanto é $110100_{(2)} =?_{(10)}$ Faça-se: $0\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4+1\times2^5 = 0+0+4+0+16+32 = 52 = 52_{(10)}$.

Conversão de base 16 para base 10

A mesma coisa: Multiplique cada dígito pelas potências crescente de 16, começando à esquerda com 16^0 e some o resultado. Acompanhe no exemplo:

Quanto é $02CA_{(16)} =?_{(10)}$ Faça-se: $A\times16^0+C\times16^1+2\times16^2+0\times16^3 = 10\times16^0+12\times16^1+2\times16^2+0\times16^3 = 10+192+512+0 = 714_{(16)}$.

Conversão de base 10 para base 2

A regra é através da apropriação dos sucessivos restos da divisão inteira do número por 2. Acompanhe no exemplo:

Quanto é $52_{(10)} =?_{(2)}$. Divida-se 52 por 2. O resultado é 26 e o resto é 0. Depois $26 \div 2 = 13$ e o resto é 1. Depois $13 \div 2 = 6$ e o resto é 1. Depois $6 \div 2 = 3$, resto 0. $3 \div 2 = 1$, resto 1. $1 \div 2 = 0$, resto 1. Tomando os restos de trás para a frente, obtem-se o número binário: 110100, que é o número buscado.

Conversão de base 10 para base 16

A regra é a mesma: a apropriação dos sucessivos restos da divisão inteira do número por 16. Acompanhe no exemplo:

Quanto é $714_{(10)} =?_{(16)}$. Divida-se 714 por 16. Dá 44, com resto 10. Depois, $44 \div 16 = 2$, resto 12. Depois, $2 \div 16 = 0$, resto 2. Tomando os restos fica 2.12.10 ou mais propriamente 2CA, que é o número buscado. Para a realização deste tipo de raciocínio, convém decorar a seguinte tabela de potências crescentes de 2 (e de 16)

2^0	1		
2^1	2	1 b	
2^2	4		
2^3	8		
2^4	16		
2^5	32		
2^6	64		
2^7	128		
2^8	256	1 B	10^0 B
2^9	512		
2^{10}	1024	1 KB	$\approx 10^3$
2^{12}	4096		
2^{16}	65536		
2^{20}	1048576	1 MB	$\approx 10^6$
2^{30}		1 GB	$\approx 10^9$
2^{40}		1 TB	$\approx 10^{12}$
2^{50}		1 PB	$\approx 10^{15}$
2^{60}		1 EB	$\approx 10^{18}$
2^{70}		1 ZB	$\approx 10^{21}$
2^{80}		1 YB	$\approx 10^{24}$

Adição em binário trivial, assim como a soma em base decimal. Veja-se as regras (são apenas 4) que estão no começo desta folha.

Adição em hexadecimal

A regra é a mesma da adição em decimal. Somam-se os valores das duas parcelas e se o resultado é igual ou maior do que a base, tira-se uma base e "vai um". Acompanhe nos exemplos:

```
  02CAFE      166A4
+ 00765B    + 22BB7
--------    -------
  34159       3925B
```

☞ Para você fazer

Resolva os exercícios seguintes e informe os resultados na base pedida.

1. $10101010_{(2)} =?_{(10)}$
2. $B32_{(16)} =?_{(10)}$
3. $263_{(10)} =?_{(2)}$
4. $279_{(10)} =?_{(2)}$
5. $3457_{(10)} =?_{(16)}$
6. $4008_{(10)} =?_{(16)}$
7. $110001111_{(2)} + 101101110_{(2)} =?_{(2)}$
8. $100010000_{(2)} + 111100010_{(2)} =?_{(2)}$
9. $1FEB_{(16)} + 191A_{(16)} =?_{(16)}$
10. $2314_{(16)} + 14D6_{(16)} =?_{(16)}$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.