

# Arquitetura de Computadores

## REPRESENTAÇÃO DOS DADOS

Janeiro/2020

**CURSO SUPERIOR DE ANALISE E DESENVOLVIMENTO DE SISTEMAS**

FACULDADE DE TECNOLOGIA DE AMERICANA

Prof. Ms Diógenes de Oliveira

# Unidades de Medidas

# Unidades de Medidas

| Unidade | Sigla | Capacidade |
|---------|-------|------------|
| Byte    | B     | 8 bits     |
| Kilo    | K     | $2^{10}$   |
| Mega    | M     | $2^{20}$   |
| Giga    | G     | $2^{30}$   |
| Tera    | T     | $2^{40}$   |
| Peta    | P     | $2^{50}$   |
| Exa     | E     | $2^{60}$   |
| Zetta   | Z     | $2^{70}$   |

escreva o valor de x nas unidades indicadas: (escreva somente a expressão matemática)

a)  $65.536 = xK$

$$65.536 / 2^{10}$$

b)  $12.288K = xM$

$$12.288 \times 2^{10} / 2^{20}$$
$$12.288 / 2^{10}$$

c)  $19.922.944 = xM$

$$19.922.944 / 2^{20}$$

d)  $8GB = xBytes$

$$8 \times 2^{30}$$

e)  $262.144bits = xKbits$

$$262.144 / 2^{10}$$

f)  $10Gbits = xBytes$

$$10 \times 2^{30} / 8$$
$$10 \times 2^{27}$$

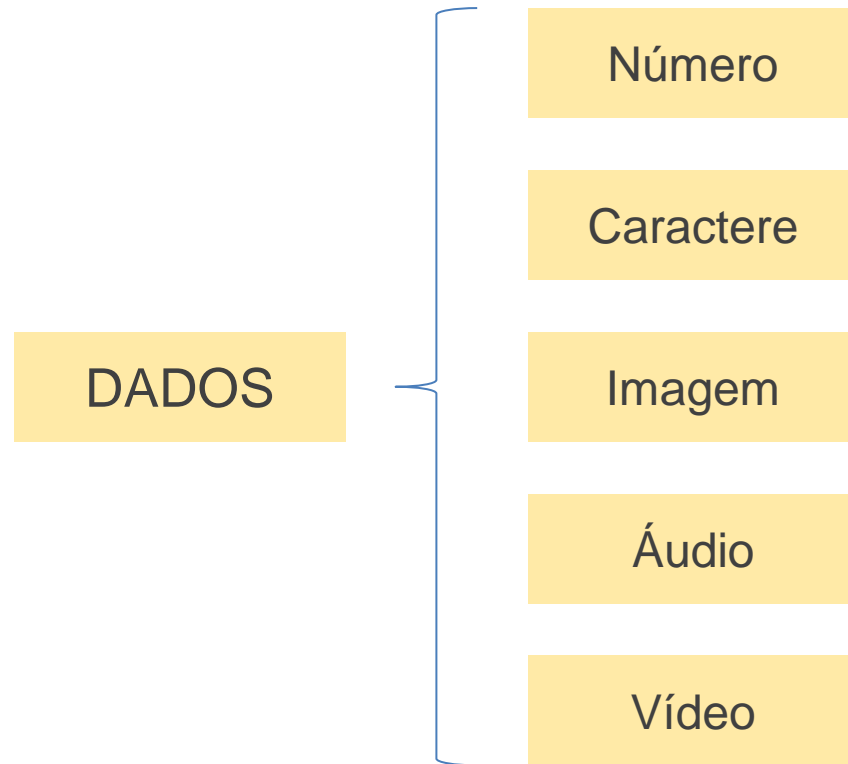
| Unidade | Sigla | Capacidade |
|---------|-------|------------|
| Byte    | B     | 8 bits     |
| Kilo    | K     | $2^{10}$   |
| Mega    | M     | $2^{20}$   |
| Giga    | G     | $2^{30}$   |
| Tera    | T     | $2^{40}$   |
| Peta    | P     | $2^{50}$   |
| Exa     | E     | $2^{60}$   |
| Zetta   | Z     | $2^{70}$   |

Quantos números diferentes podem ser criados com 5 (cinco) algarismos binários?

$$2^5 = 32 \text{ números}$$

00000  
00001  
00010  
00011  
00100  
00101  
00110  
00111  
01000  
01001  
01010  
01011  
01100  
...  
11111

# Representação de Dado



# Números

| PALAVRA | COM SINAL  | SEM SINAL (unsigned)           |
|---------|--|--------------------------------|
| 8 bits  | -128 a<br>127  | 0 a 255                        |
| 16 bits | - 32.768 a<br>32.767                                       | 0 a 65.535                     |
| 32 bits | - 2.147.483.648 a<br>2.147.483.647                         | 0 a 4.294.967.295              |
| 64 bits | - 9.223.372.036.854.775.808 a<br>9.223.372.036.854.775.807 | 0 a 18.446.744.073.709.551.615 |

# Números

|   |   |
|---|---|
| <code>unsigned char a = 65;</code>            | 0100 0001                               |
| <code>unsigned char e = 0x41;</code>          | 0100 0001                               |
| <code>unsigned int16 d = 33056;</code>        | 1000 0001 0010 0000                     |
| <code>unsigned int32 f = 523.934.450</code>   | 0001 1111 0011 1010 1001 1010 1111 0010 |
| <code>unsigned int32 h = 4.294.967.295</code> | 1111 1111 1111 1111 1111 1111 1111 1111 |
| <code>char j = -65;</code>                    | 1100 0001                               |
| <code>char j = 10;</code>                     | 0000 1010                               |
| <code>char m = 44;</code>                     | 0010 1100                               |
| <code>char n = m + j;</code>                  | 0011 0110                               |



# Números Negativos

`char c = -65;`

**1**100 0001



O bit mais a esquerda é o bit de sinal

1000 0000 = -128

0111 1111 = +127

`int32 g = 0x-1F3A9AF2`

**1**001 1111 0011 1010 1001 1010 1111 0010  
1 1111 0011 1010 1001 1010 1111 0010

`int32 h = -1`

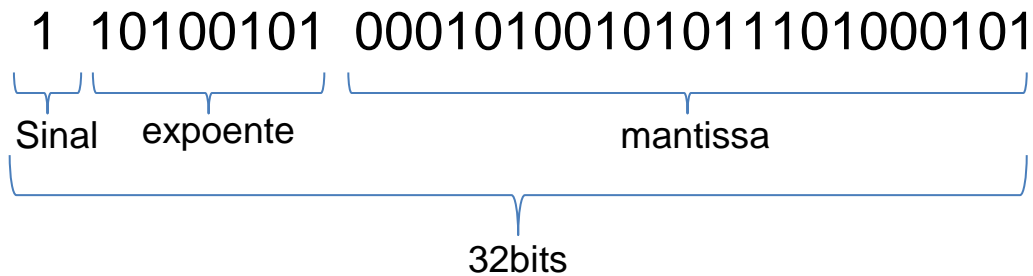
**1**000 0000 0000 0000 0000 0000 0000 0001

# Números de ponto flutuante

Especificação do padrão IEEE (Instituto de Engenharia Elétrica e Eletrônica)

| PARÂMETRO                                    | PRECISÃO SIMPLES | PRECISÃO DUPLA |
|--|------------------|----------------|
| Tamanho (bits) da célula de MP               | 32               | 64             |
| bit de sinal (mais a direita)                | 1                | 1              |
| Tamanho do expoente (num. de bits)           | 8                | 11             |
| Tamanho da mantissa ou fração (num. de bits) | 23               | 52             |
|  |                  |                |

Ex:



# Texto, Caractere

`char b="A";`      0100 0001

`char[] b="FATEC";`    0100 0110 0100 0001 0101 0100 0100 0101 0100 0011



Código do caractere **F**  
na tabela ASCII

`char c="65";`      0011 0110 0011 0101

`char d=65;`      0100 0001

# ASCII - American Standard Code for Information Interchanged

| DEC |     |      | 0    | 16   | 32   | 48   | 64   | 80   | 96   | 112  | 128  | 144  | 160  | 176  | 192  | 208  | 224  | 240  |
|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
|     | HEX |      | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | A    | B    | C    | D    | E    | F    |
|     |     | BIN  | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| 0   | 0   | 0000 | NUL  | DLE  | SPC  | 0    | @    | P    | `    | p    | Ç    | É    | á    |      | Á    |      | ÿ    |      |
| 1   | 1   | 0001 | SOH  | DC1  | !    | 1    | A    | Q    | a    | q    | ü    | æ    | í    |      |      |      | ß    |      |
| 2   | 2   | 0010 | STX  | DC2  | "    | 2    | B    | R    | b    | r    | é    | Æ    | ó    |      |      |      |      |      |
| 3   | 3   | 0011 | ETX  | DC3  | #    | 3    | C    | S    | c    | s    | â    | ô    | ú    |      |      |      |      |      |
| 4   | 4   | 0100 | EOT  | DC4  | \$   | 4    | D    | T    | d    | t    | ä    | ö    | ñ    |      |      |      |      |      |
| 5   | 5   | 0101 | ENQ  | NAK  | %    | 5    | E    | U    | e    | u    | à    | ò    | Ñ    |      |      |      |      |      |
| 6   | 6   | 0110 | ACK  | SYN  | &    | 6    | F    | V    | f    | v    | å    | û    | ª    |      |      |      |      |      |
| 7   | 7   | 0111 | BEL  | ETB  | '    | 7    | G    | W    | g    | w    | ç    | ù    | º    |      |      |      |      |      |
| 8   | 8   | 1000 | BS   | CAN  | (    | 8    | H    | X    | h    | x    | ê    | ÿ    | ¿    |      |      |      |      |      |
| 9   | 9   | 1001 | HT   | EM   | )    | 9    | I    | Y    | i    | y    | ë    | Ö    |      |      |      |      |      |      |
| 10  | A   | 1010 | LF   | SUB  | *    | :    | J    | Z    | j    | z    | è    | Ü    |      |      |      |      |      |      |
| 11  | B   | 1011 | VT   | ESC  | +    | ;    | K    | [    | k    | {    | ï    | ¢    | ½    |      |      |      |      |      |
| 12  | C   | 1100 | FF   | FS   | ,    | <    | L    | \    | l    | !    | î    | £    | ¼    |      |      |      |      | n    |
| 13  | D   | 1101 | CR   | GS   | -    | =    | M    | ]    | m    | }    | ì    | ¥    | í    |      |      |      |      |      |
| 14  | E   | 1110 | SO   | RS   | .    | >    | N    | ^    | n    | ~    | Ä    | Pt   | «    |      |      |      |      |      |
| 15  | F   | 1111 | SI   | US   | /    | ?    | O    | _    | o    | del  | Å    |      | »    |      |      |      |      | del  |

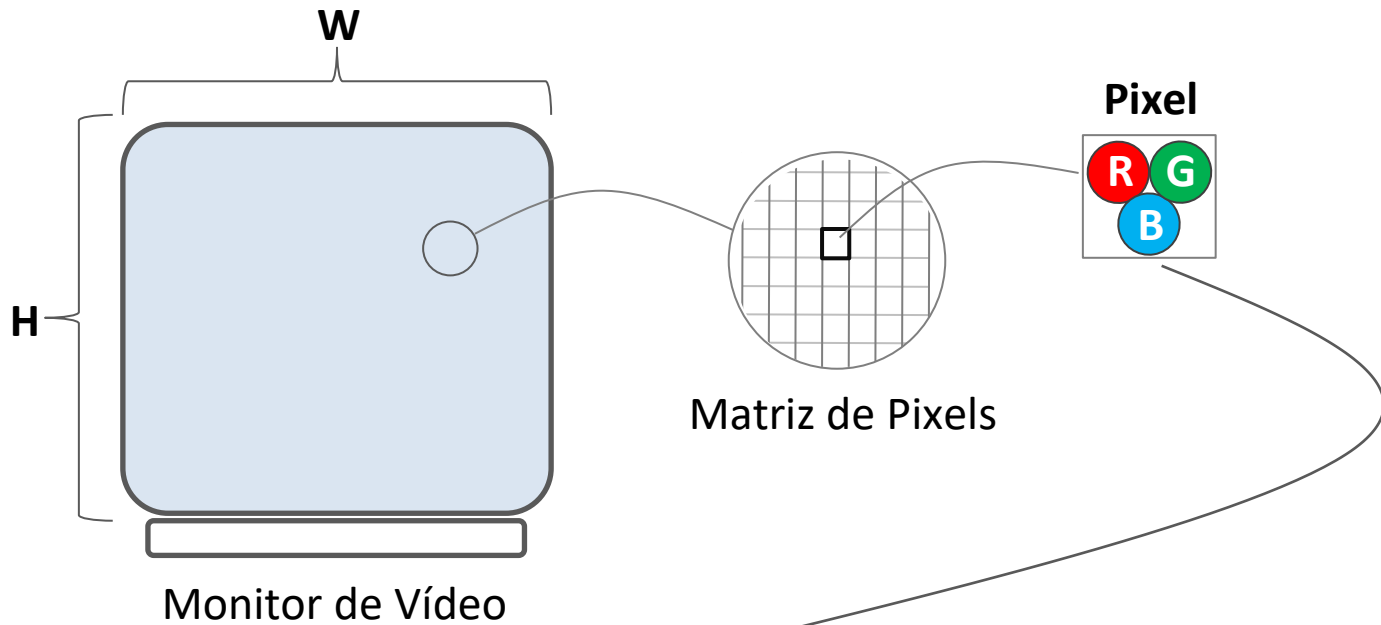
Exemplo:

Tabela padrão

Depende da cultura instalada

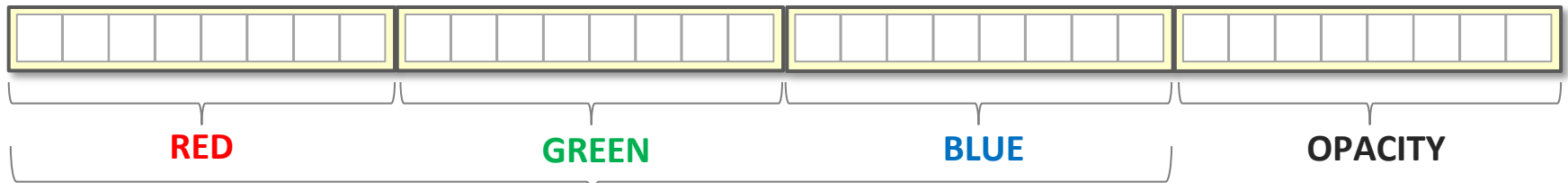
$$\mathbf{A} = 0100\ 0001_2 = 41_{16} = 65_{10}$$

# Imagem



Monitor de Vídeo

24 bits



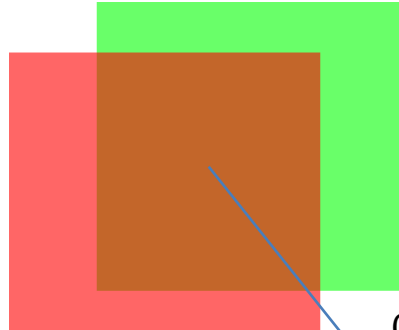
24 bits  $\Rightarrow 2^{24} = 16.777.216$  de cores

32 bits

# RGB

#FF 00 00 3C

#00 FF 00 3C



1111 1111 0000 0000 0000 0000 0011 1100

0000 0000 1111 1111 0000 0000 0011 1100

#C2 66 29 FF

1100 0010 0110 0110 0010 1001 1111 1111

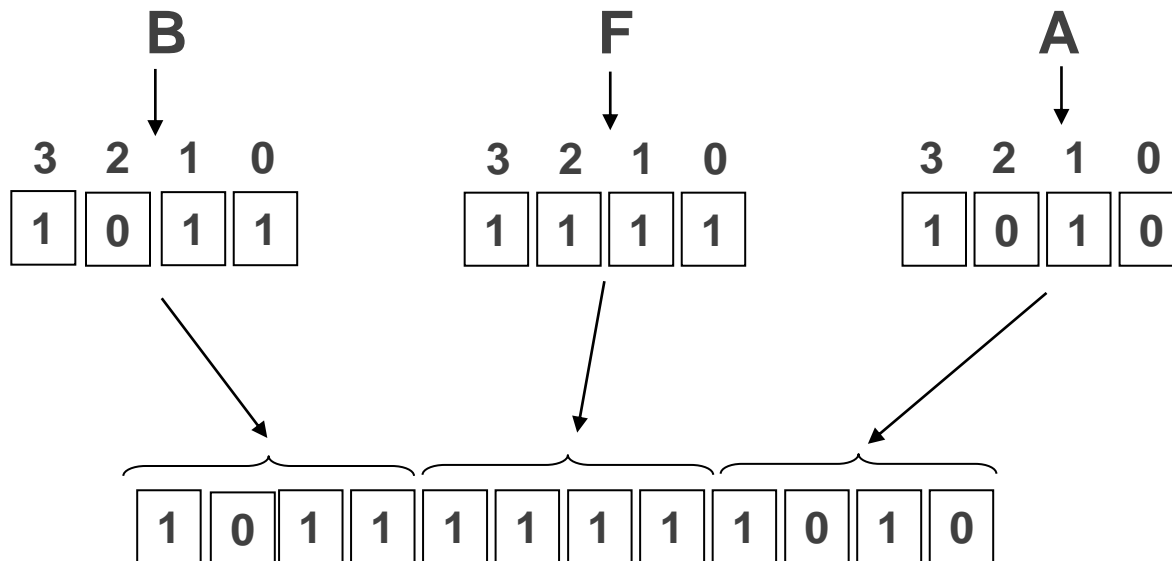
# Conversão de Bases Numéricas

## Hexadecimal p/ Binário

|     |    |
|-----|----|
| 0 = | 0  |
| 1 = | 1  |
| 2 = | 2  |
| 3 = | 3  |
| 4 = | 4  |
| 5 = | 5  |
| 6 = | 6  |
| 7 = | 7  |
| 8 = | 8  |
| 9 = | 9  |
| A = | 10 |
| B = | 11 |
| C = | 12 |
| D = | 13 |
| E = | 14 |
| F = | 15 |

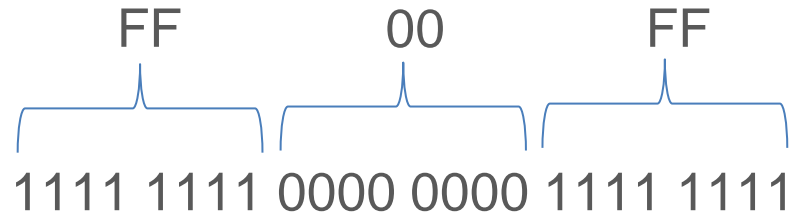
Hexadecimal

Decimal

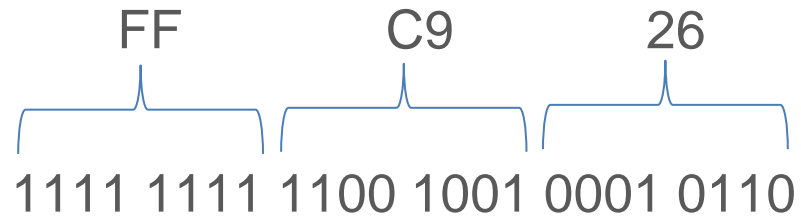


# Cor do objeto RGB

#FF00FF



#FFC926



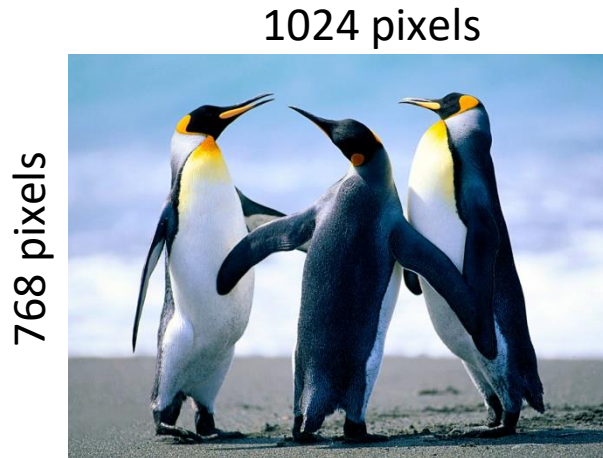


# Memória Principal

Representação das Informações na MP

## Imagem Gráfica

Exemplo:



$$1024 \times 768 = \mathbf{786.432 \text{ pixels}}$$

$$786.432 \times 32 \text{ bits} = \mathbf{25.165.824 \text{ bits}}$$

$$25.165.824 / 8 = \mathbf{3.145.728 \text{ Bytes}}$$

$$3.145.728 / 2^{20} = \mathbf{3 \text{ MB}}$$

# Memória Principal

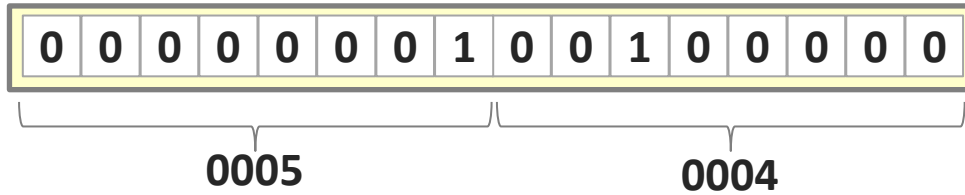
Representação das Informações na MP

char a = 65;

char b = "A";

char c = -65;

Int16 d = 288;



Int32 e = 85000; = 1 01001100 00001000

## MEMÓRIA PRINCIPAL

0000

0001

0002

0003

0004

0005

0006

0007

0008

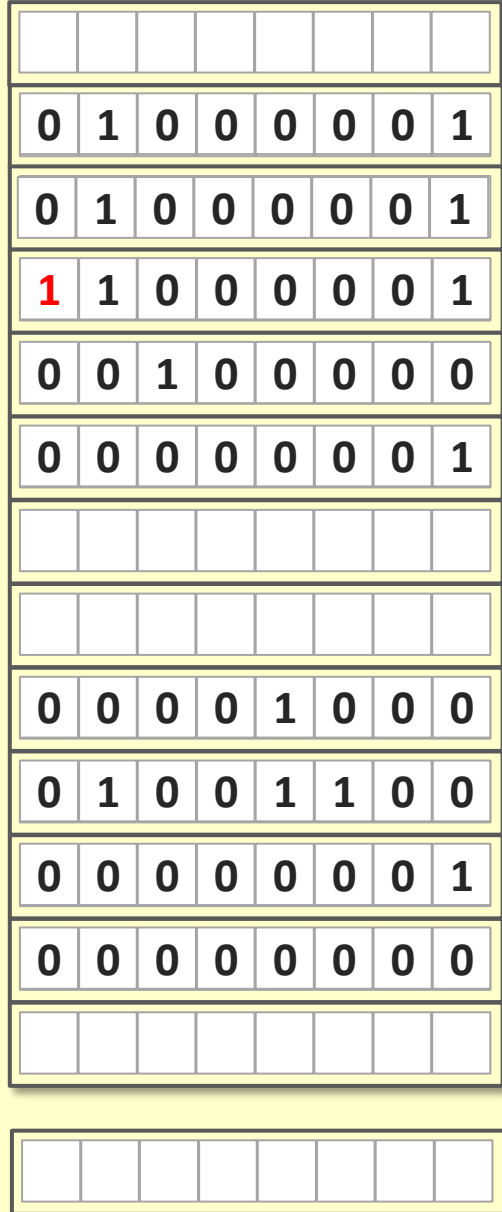
0009

0010

0011

0012

N-1

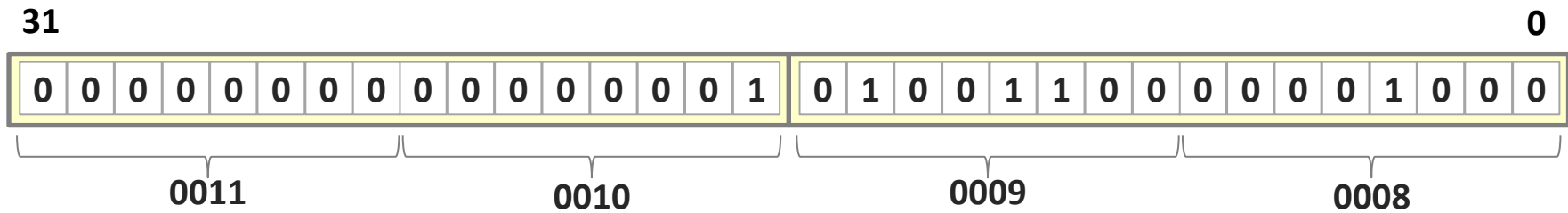


# Memória Principal

Representação das Informações na MP

Exemplo:

**Int32 e=85000;** = 10100110000001000



## Exercício em linguagem C++

```
unsigned char a = 65;  
unsigned char b = 'A';  
unsigned char c = 0x41;  
char d = 255;  
unsigned e = 255;
```

```
printf("%c \n %x \n",a,&a);  
printf("%c \n",a+2);  
printf("%d \n %x \n",b,&b);  
printf("%d \n",b-1);  
printf("%c \n %x \n",c,&c);  
printf("%d \n",c);  
printf("%x \n",c);  
printf("%d \n",d);
```

$$65_{10} = 41_{16} = 01000001_2 = 'A'$$

## REFERÊNCIAS BIBLIOGRÁFICAS:

- [1] TANEMBAUM, A. S. **Organização Estruturada de Computadores**, Livros Técnicos e Científicos, 2000. 460p.
- [2] MONTEIRO, M.A. **Introdução à Organização de Computadores**, 5a ed. Livros Técnicos e Científico Editora SA, 2007. 695p