

AULA – 08

PONTEIROS

Caio César Teodoro Mendes
NUSP: 7252290

Pointeiros - Analogia

Cada gaveta possui um objeto (variáveis comum)



Gavetas endereçadas

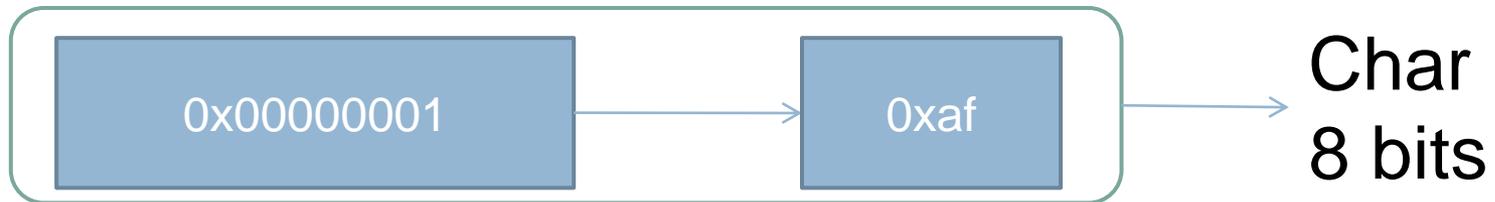
Gaveta possui um endereço (Ponteiro)



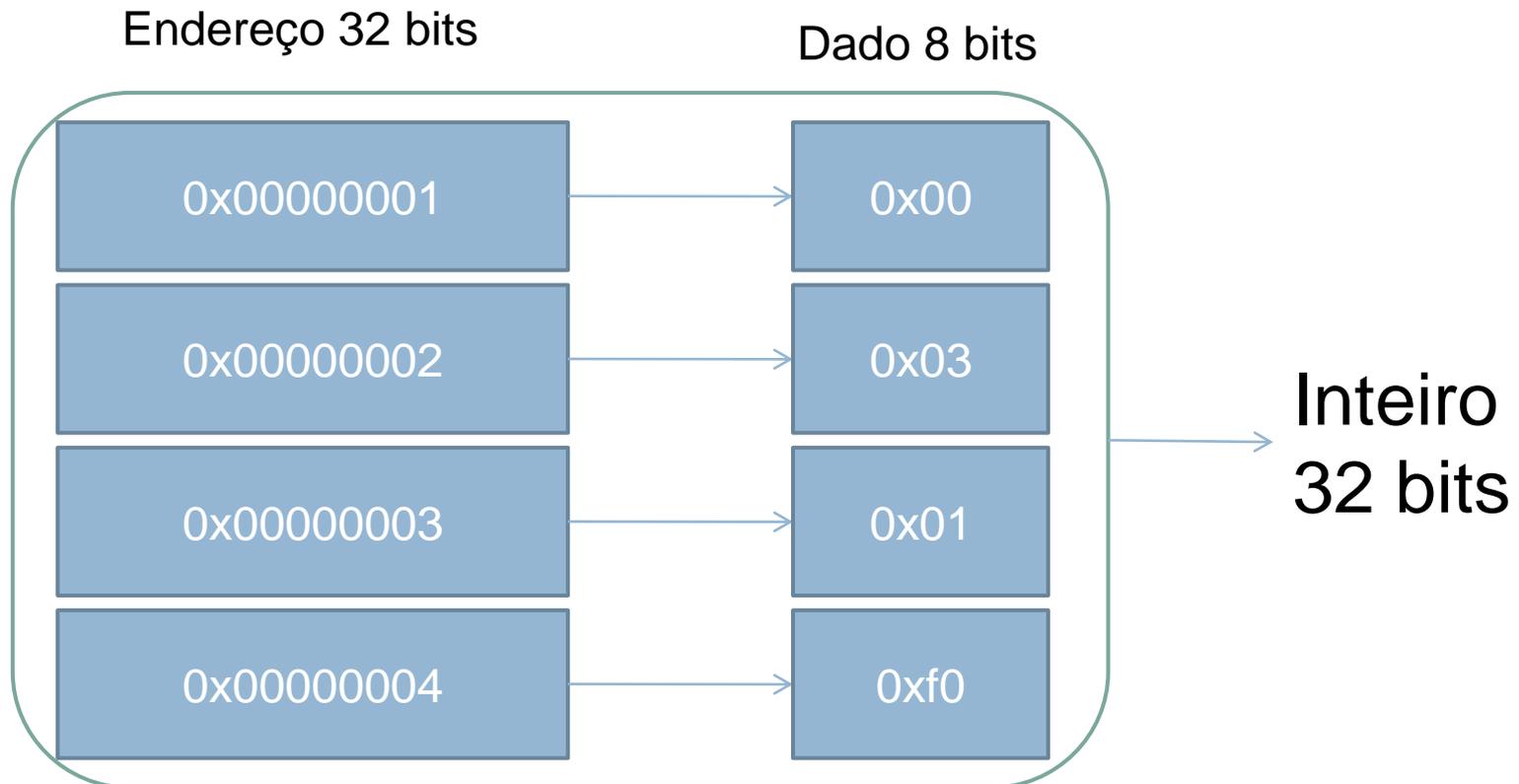
Memória

Endereço 32 bits

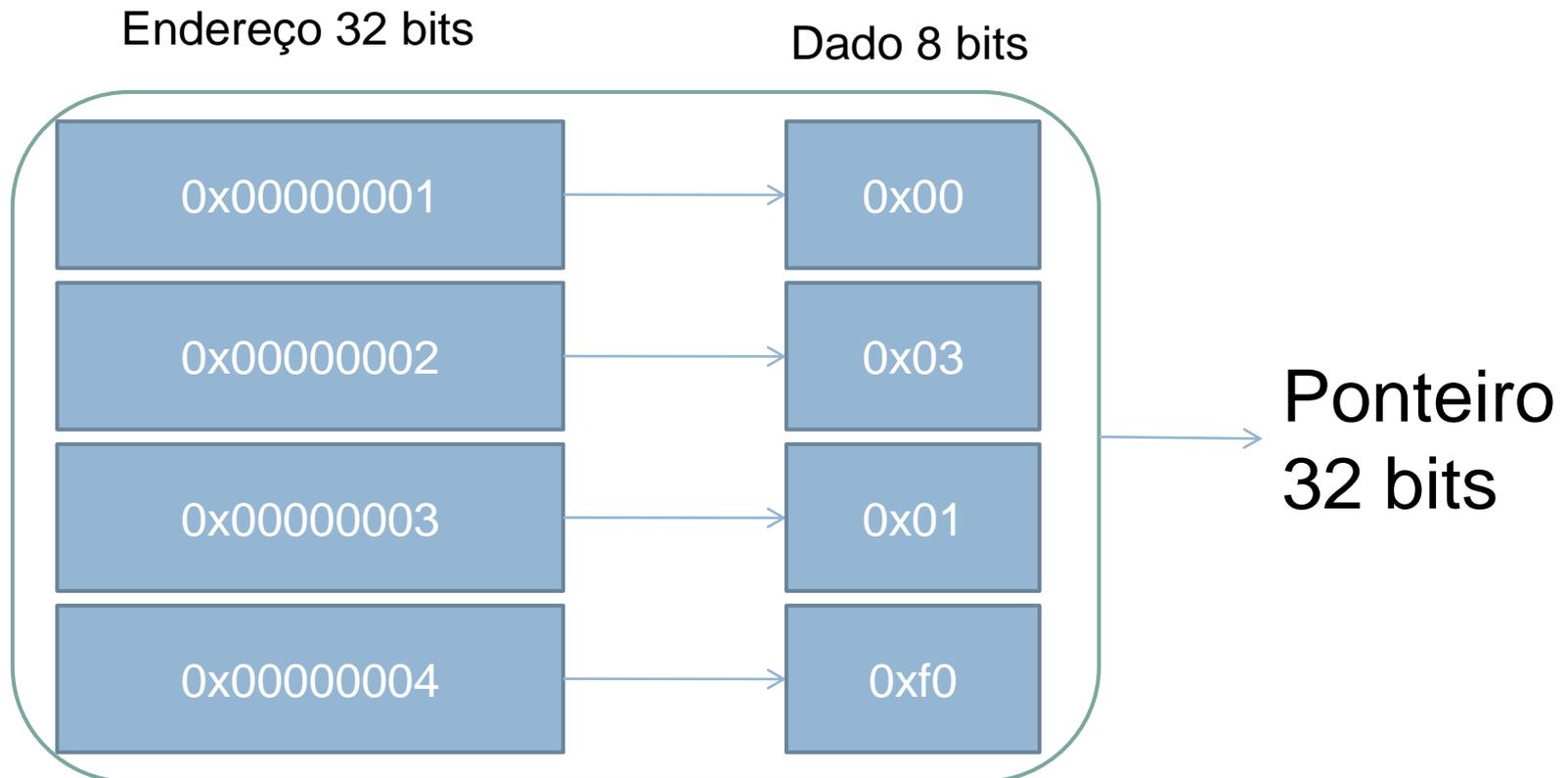
Dado 8 bits



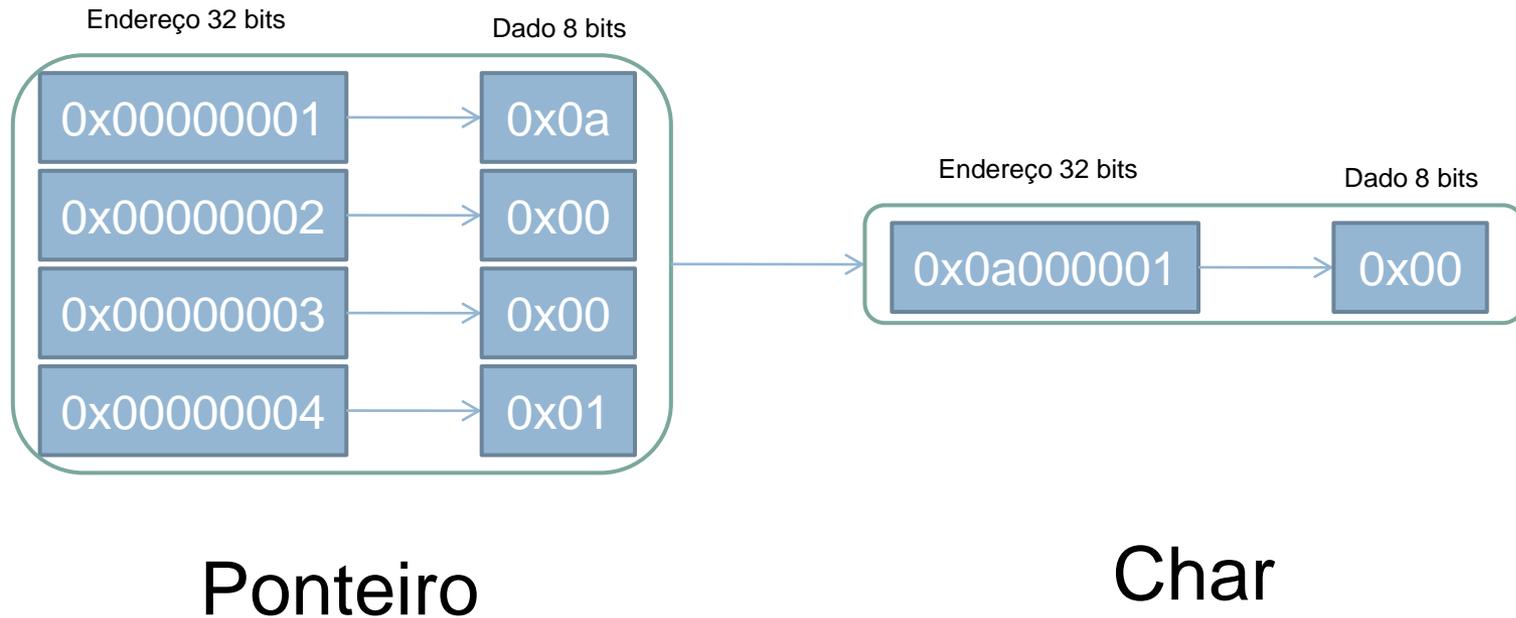
Memória



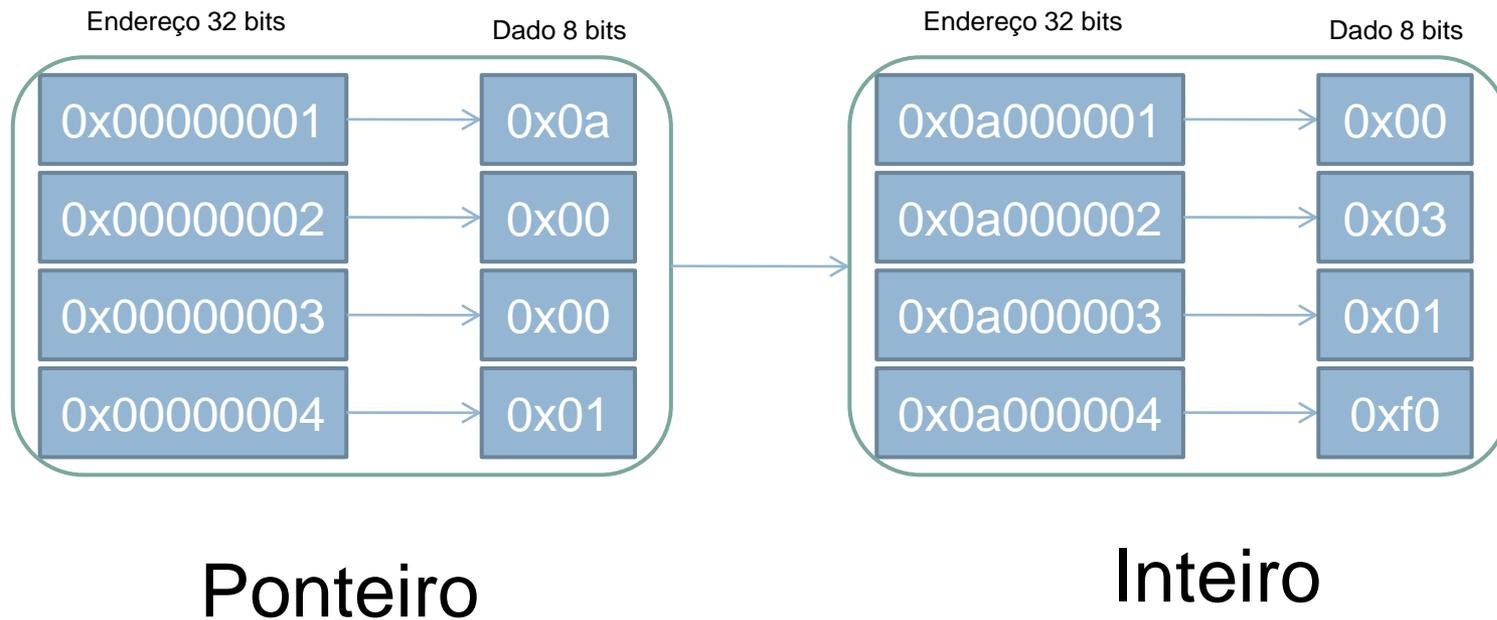
Memória



Memória



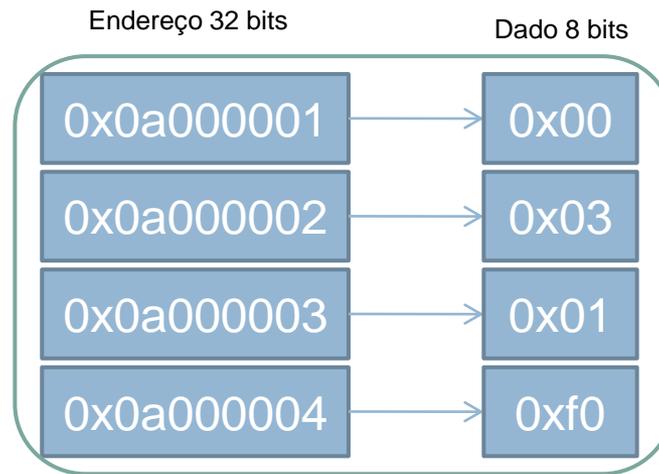
Memória



Programação

```
int dado;  
printf("%d\n", dado);  
printf("%x", &dado);
```

197104
0x0a000001



Inteiro

Programação

```
char dado;  
printf("%c\n", dado);  
printf("%x", &dado);
```

A
0x0a000001

Endereço 32 bits

Dado 8 bits

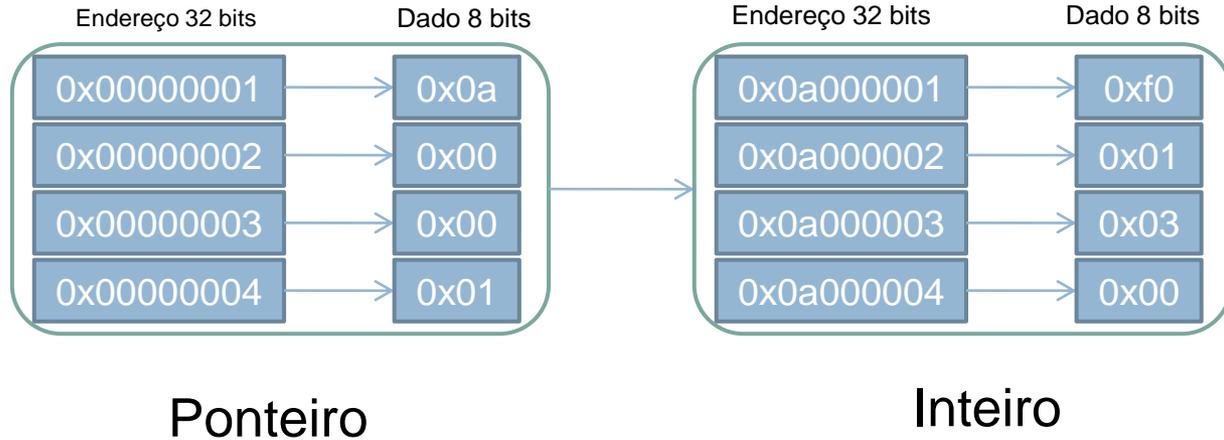
0x0a000001

0x41

Programação

```
int dado;  
int * ponteiro;  
ponteiro = &dado;  
printf("%d\n", dado);  
printf("%x\n", &dado);  
printf("%x\n", ponteiro);  
printf("%x", &ponteiro);
```

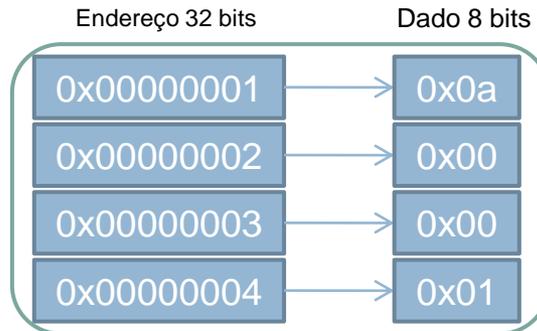
197104
0x0a000001
0x0a000001
0x00000001



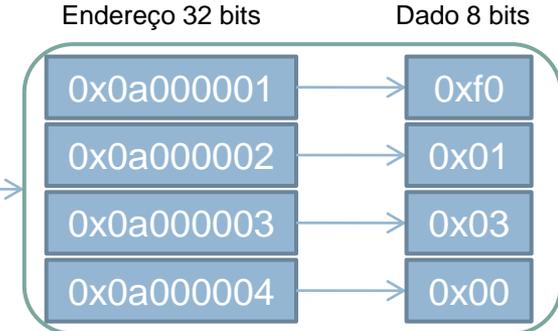
Programação

```
int dado;  
int * ponteiro;  
ponteiro = &dado;  
printf("%d\n", dado);  
printf("%d\n", &dado);  
Printf("%d", *ponteiro);
```

197104
0x0a000001
197104



Ponteiro



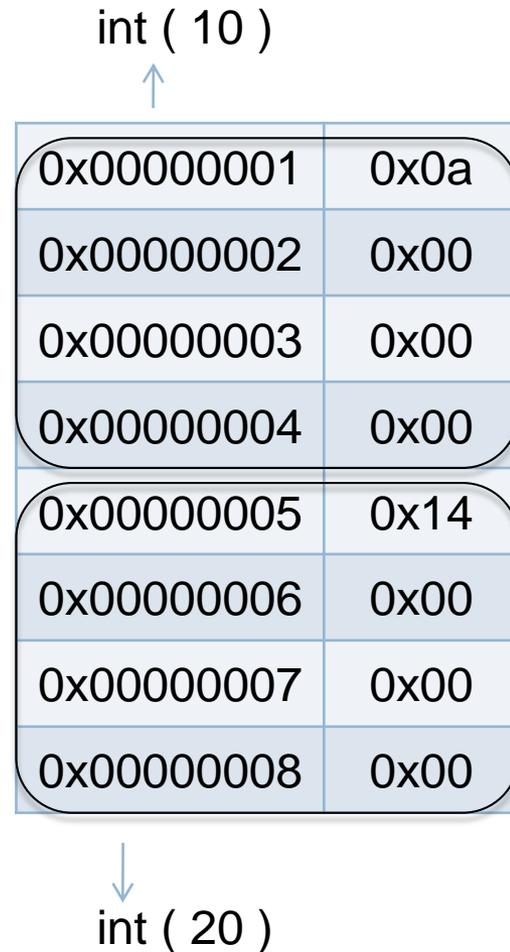
Inteiro

Programação

```
int vetor[2] = { 10, 20};  
int * ponteiro;  
ponteiro = vetor;  
printf("%d\n", vetor[0]);  
printf("%x\n", &vetor[0]);  
printf("%d\n", vetor[1]);  
printf("%x\n", &vetor[1]);
```

```
printf("%d\n", *ponteiro);  
ponteiro++;  
printf("%d\n", *ponteiro);  
printf("%x", ponteiro);
```

```
10  
0x00000001  
20  
0x00000005  
10  
20  
0x00000005
```

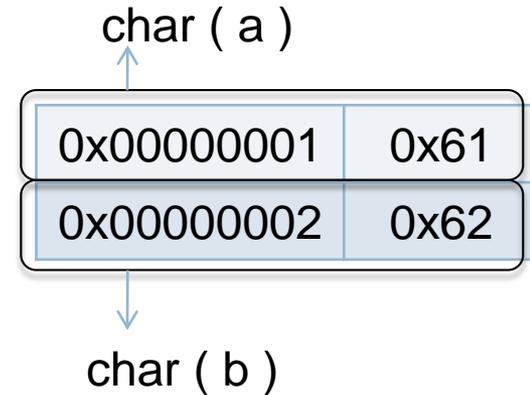


Programação

```
char vetor[2] = { 'a', 'b'};  
int * ponteiro;  
ponteiro = vetor;  
printf("%c\n", vetor[0]);  
printf("%x\n", &vetor[0]);  
printf("%c\n", vetor[1]);  
printf("%x\n", &vetor[1]);
```

```
printf("%c\n",  
ponteiro[0]);  
printf("%c\n",  
ponteiro[1]);  
printf("%x", ponteiro);
```

```
a  
0x00000001  
b  
0x00000002  
a  
b  
0x00000001
```



Programação

```
char vetor[2] = { 'a', 'b'};  
int * ponteiro;  
ponteiro = vetor;  
printf("%c\n", *(&vetor[1]));
```

b

char (a)

0x00000001	0x61
0x00000002	0x62

char (b)

Escopo

```
float funcao()
{
    int a=0;
    a++;
}

Int main()
{
    int a = 0;
    funcao();
    printf("%d", a);
}
```

Passagem de parâmetros

```
float calcMedia(float e1, float e2, float e3, float e4...)
```



```
float calcMedia(float *vetorE);
```

VANTAGENS:

- Economia de memória
- Praticidade
- Desempenho

Passagem de parâmetros

```
float calcMedia(float *vetorE);
```

```
float vetor[20];  
...  
calcMedia(vetor);
```

```
float vetor[20];  
float *vetorE;  
...  
vetorE = vetor; //Equivalente mas em outro escopo!!!
```

Exercício

- Implementar uma função que guarde todos os nomes e médias em um arquivo utilizando passagem de parâmetros por referência.