

# Introdução à Programação #17

Henrique Dias @ Pplware

24 de dezembro 2014

## 1 Apontadores

### 1.1 Tamanho e endereço de variáveis

Primeiramente, devo recordar que todos os tipos de dados ocupam um espaço diferente na memória RAM. Para saber quantas *bytes* ocupam um determinado tipo, deve-se recorrer à função *sizeof* cujo único argumento é a variável em questão.

Ou seja, para se saber quantas *bytes* ocupa o tipo `int`, devemos passar como argumento à função *sizeof* uma variável desse tipo.

No seguinte exemplo é imprimido o tamanho de espaço que ocupam quatro tipos de dados e ainda o endereço da memória a que corresponde cada variável.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     char caractere;
7     int inteiro;
8     float Float;
9     double Double;
10
11     printf("Tipo\tNum de Bytes\tEndereço\n");
12     printf("_____\n");
13     printf("Char\t%d byte \t\t%p\n", sizeof(caractere), &caractere);
14     printf("Inteiro\t%d bytes \t\t%p\n", sizeof(inteiro), &inteiro);
15     printf("Float\t%d bytes \t\t%p\n", sizeof(Float), &Float);
16     printf("Double\t%d bytes \t\t%p\n", sizeof(Double), &Double);
17
18     return 0;
19 }
```

O resultado da execução do código acima deverá ser algo semelhante ao seguinte:

Tipo	Num de Bytes	Endereço
Char	1 byte	0028FF1F
Inteiro	4 bytes	0028FF18
Float	4 bytes	0028FF14
Double	8 bytes	0028FF08

Tabela 1: Tamanho e Endereço das Variáveis

Como pode verificar, todos os tipos de dados exceto `char` ocupam mais do que um *byte* na memória RAM. Então, o endereço que é imprimido corresponde apenas ao **primeiro byte** ocupado pela variável.

**Hexadecimal para Decimal** Antes de continuarmos, será útil abordar como se convertem números hexadecimais para decimais. Primeiramente, como já deve ter reparado, os números hexadecimais contam com letras, além de números.

Isto acontece porque a numeração hexadecimal tem como base 16 números enquanto que a decimal tem como base 10 números. Veja então o número que corresponde a cada letra da numeração hexadecimal na tabela seguinte:

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

Tabela 2: Conversão Hexadecimal para Decimal

Iremos converter o número hexadecimal 0028FF1F para decimal. Em primeiro lugar, iremos tirar os primeiros dois zeros que são, neste caso, desprezíveis.

De seguida precisamos conhecer o significado de cada dígito hexadecimal em número decimal que está presente nesse número. Em decimal, os números 2, 8, F e 1 são, respetivamente, 2, 8, 15 e 1.

Para efetuar a conversão, temos que multiplicar cada um desses números por uma potência de base 16 cujo expoente é igual à posição de cada um dos números (da direita para a esquerda). Então, temos:

$$\begin{aligned}
 &(2 * 16^5) + (8 * 16^4) + (15 * 16^3) + (15 * 16^2) + (1 * 16^1) + (15 * 16^0) = \\
 &= 2097152 + 524288 + 61440 + 3840 + 16 + 15 = \\
 &= 2686751
 \end{aligned}$$

Voltando aos endereços das variáveis e convertendo-os para números decimais, obtemos os seguintes valores:

Tipo	Num de Bytes	Endereço
Char	1 byte	2686751
Inteiro	4 bytes	2686744
Float	4 bytes	2686740
Double	8 bytes	2686728

Tabela 3: Endereço das Variáveis em Formato Decimal

Podemos concluir dos dados acima que a variável que declaramos do tipo *char*, no meu computador, ocupa a posição 2686751 enquanto que as outras ocupam mais do que uma.

Relembrando que as *bytes* alocadas para variáveis que ocupam mais do que uma encontram-se localizadas lado a lado, de seguida e que o endereço que obtemos corresponde ao da primeira *byte*, a variável do tipo inteiro que declaramos ocupa as *bytes* cujos endereços (em número decimal) são 2686744, 2686745, 2686746 e 2686747.