

O espaço de memória, quando alocado de forma dinâmica, reserva espaços de memória contíguas. Isso pode ser comprovado no seguinte trecho:

```
for (i = 0; i < LINHA; i++) {  
    for (j = 0; j < COLUNA; j++) {  
        printf(" %p ", &Matriz[i][j]);  
    }  
    printf("\n");  
}
```

O resultado desse loop mostra que, cada elemento está localizado a uma distância de 4 bytes, que é o tamanho do tipo 'int', em relação ao elemento anterior, seguindo uma sequência contínua.

Segue o teste feito com uma matriz 5x5.

Endereço do ponteiro --> 0061FF14

Usando Malloc

008B18B0 008B18B4 008B18B8 008B18BC 008B18C0
008B18D0 008B18D4 008B18D8 008B18DC 008B18E0
008B18F0 008B18F4 008B18F8 008B18FC 008B1900
008B1910 008B1914 008B1918 008B191C 008B1920
008B1930 008B1934 008B1938 008B193C 008B1940

Usando Calloc

008B15F0 008B15F4 008B15F8 008B15FC 008B1600
008B1610 008B1614 008B1618 008B161C 008B1620
008B1630 008B1634 008B1638 008B163C 008B1640
008B1650 008B1654 008B1658 008B165C 008B1660
008B1670 008B1674 008B1678 008B167C 008B1680

Imprimindo os valores

0 1 2 3 4

1 2 3 4 5

2 3 4 5 6

3 4 5 6 7

4 5 6 7 8

- A alocação contígua continua com tamanhos maiores, como essa matriz 15x15:

Endereço do ponteiro --> 0061FF14

Usando Malloc

00BF18F8 00BF18FC 00BF1900 00BF1904 00BF1908 00BF190C 00BF1910
00BF1914 00BF1918 00BF191C 00BF1920 00BF1924 00BF1928 00BF192C
00BF1930

00BF15F0 00BF15F4 00BF15F8 00BF15FC 00BF1600 00BF1604 00BF1608
00BF160C 00BF1610 00BF1614 00BF1618 00BF161C 00BF1620 00BF1624
00BF1628

00BF1638 00BF163C 00BF1640 00BF1644 00BF1648 00BF164C 00BF1650
00BF1654 00BF1658 00BF165C 00BF1660 00BF1664 00BF1668 00BF166C
00BF1670

00BF1680 00BF1684 00BF1688 00BF168C 00BF1690 00BF1694 00BF1698
00BF169C 00BF16A0 00BF16A4 00BF16A8 00BF16AC 00BF16B0 00BF16B4
00BF16B8

00BF16C8 00BF16CC 00BF16D0 00BF16D4 00BF16D8 00BF16DC 00BF16E0
00BF16E4 00BF16E8 00BF16EC 00BF16F0 00BF16F4 00BF16F8 00BF16FC
00BF1700

00BF1710 00BF1714 00BF1718 00BF171C 00BF1720 00BF1724 00BF1728
00BF172C 00BF1730 00BF1734 00BF1738 00BF173C 00BF1740 00BF1744
00BF1748

00BF1758 00BF175C 00BF1760 00BF1764 00BF1768 00BF176C 00BF1770
00BF1774 00BF1778 00BF177C 00BF1780 00BF1784 00BF1788 00BF178C
00BF1790

00BF3A10 00BF3A14 00BF3A18 00BF3A1C 00BF3A20 00BF3A24 00BF3A28
00BF3A2C 00BF3A30 00BF3A34 00BF3A38 00BF3A3C 00BF3A40 00BF3A44
00BF3A48

00BF3A58 00BF3A5C 00BF3A60 00BF3A64 00BF3A68 00BF3A6C
00BF3A70 00BF3A74 00BF3A78 00BF3A7C 00BF3A80 00BF3A84 00BF3A88
00BF3A8C 00BF3A90

00BF3AA0 00BF3AA4 00BF3AA8 00BF3AAC 00BF3AB0 00BF3AB4
00BF3AB8 00BF3ABC 00BF3AC0 00BF3AC4 00BF3AC8 00BF3ACC 00BF3AD0
00BF3AD4 00BF3AD8

00BF3AE8 00BF3AEC 00BF3AF0 00BF3AF4 00BF3AF8 00BF3AFC
00BF3B00 00BF3B04 00BF3B08 00BF3B0C 00BF3B10 00BF3B14 00BF3B18
00BF3B1C 00BF3B20

00BF3B30 00BF3B34 00BF3B38 00BF3B3C 00BF3B40 00BF3B44 00BF3B48
00BF3B4C 00BF3B50 00BF3B54 00BF3B58 00BF3B5C 00BF3B60 00BF3B64
00BF3B68

00BF3B78 00BF3B7C 00BF3B80 00BF3B84 00BF3B88 00BF3B8C 00BF3B90
00BF3B94 00BF3B98 00BF3B9C 00BF3BA0 00BF3BA4 00BF3BA8 00BF3BAC
00BF3BB0

00BF3BC0 00BF3BC4 00BF3BC8 00BF3BCC 00BF3BD0 00BF3BD4
00BF3BD8 00BF3BDC 00BF3BE0 00BF3BE4 00BF3BE8 00BF3BEC 00BF3BF0
00BF3BF4 00BF3BF8

00BF3C08 00BF3C0C 00BF3C10 00BF3C14 00BF3C18 00BF3C1C 00BF3C20
00BF3C24 00BF3C28 00BF3C2C 00BF3C30 00BF3C34 00BF3C38 00BF3C3C
00BF3C40

Usando Calloc

00BF3C50 00BF3C54 00BF3C58 00BF3C5C 00BF3C60 00BF3C64 00BF3C68
00BF3C6C 00BF3C70 00BF3C74 00BF3C78 00BF3C7C 00BF3C80 00BF3C84
00BF3C88

00BF04B0 00BF04B4 00BF04B8 00BF04BC 00BF04C0 00BF04C4 00BF04C8
00BF04CC 00BF04D0 00BF04D4 00BF04D8 00BF04DC 00BF04E0 00BF04E4
00BF04E8

00BF6C38 00BF6C3C 00BF6C40 00BF6C44 00BF6C48 00BF6C4C 00BF6C50
00BF6C54 00BF6C58 00BF6C5C 00BF6C60 00BF6C64 00BF6C68 00BF6C6C
00BF6C70

00BF6D10 00BF6D14 00BF6D18 00BF6D1C 00BF6D20 00BF6D24 00BF6D28
00BF6D2C 00BF6D30 00BF6D34 00BF6D38 00BF6D3C 00BF6D40 00BF6D44
00BF6D48

00BF6F50 00BF6F54 00BF6F58 00BF6F5C 00BF6F60 00BF6F64 00BF6F68
00BF6F6C 00BF6F70 00BF6F74 00BF6F78 00BF6F7C 00BF6F80 00BF6F84
00BF6F88

00BF7388 00BF738C 00BF7390 00BF7394 00BF7398 00BF739C 00BF73A0
00BF73A4 00BF73A8 00BF73AC 00BF73B0 00BF73B4 00BF73B8 00BF73BC
00BF73C0

00BF6CC8 00BF6CCC 00BF6CD0 00BF6CD4 00BF6CD8 00BF6CDC
00BF6CE0 00BF6CE4 00BF6CE8 00BF6CEC 00BF6CF0 00BF6CF4 00BF6CF8
00BF6CFC 00BF6D00

00BF6BF0 00BF6BF4 00BF6BF8 00BF6BFC 00BF6C00 00BF6C04 00BF6C08
00BF6C0C 00BF6C10 00BF6C14 00BF6C18 00BF6C1C 00BF6C20 00BF6C24
00BF6C28

00BF7070 00BF7074 00BF7078 00BF707C 00BF7080 00BF7084 00BF7088
00BF708C 00BF7090 00BF7094 00BF7098 00BF709C 00BF70A0 00BF70A4
00BF70A8

00BF7340 00BF7344 00BF7348 00BF734C 00BF7350 00BF7354 00BF7358
00BF735C 00BF7360 00BF7364 00BF7368 00BF736C 00BF7370 00BF7374
00BF7378

00BF70B8 00BF70BC 00BF70C0 00BF70C4 00BF70C8 00BF70CC 00BF70D0
00BF70D4 00BF70D8 00BF70DC 00BF70E0 00BF70E4 00BF70E8 00BF70EC
00BF70F0

00BF7268 00BF726C 00BF7270 00BF7274 00BF7278 00BF727C 00BF7280
00BF7284 00BF7288 00BF728C 00BF7290 00BF7294 00BF7298 00BF729C
00BF72A0

00BF6C80 00BF6C84 00BF6C88 00BF6C8C 00BF6C90 00BF6C94 00BF6C98
00BF6C9C 00BF6CA0 00BF6CA4 00BF6CA8 00BF6CAC 00BF6CB0 00BF6CB4
00BF6CB8

00BF7148 00BF714C 00BF7150 00BF7154 00BF7158 00BF715C 00BF7160
00BF7164 00BF7168 00BF716C 00BF7170 00BF7174 00BF7178 00BF717C
00BF7180

00BF6DA0 00BF6DA4 00BF6DA8 00BF6DAC 00BF6DB0 00BF6DB4
00BF6DB8 00BF6DBC 00BF6DC0 00BF6DC4 00BF6DC8 00BF6DCC 00BF6DD0
00BF6DD4 00BF6DD8

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

A função com malloc pode conter lixo, ou valores aleatórios, visto que ele apenas reserva um bloco de memória. Diferentemente, calloc garante uniformidade, já que o espaço é inicializado com zeros, conforme indicado na documentação