

Números Pseudoaleatórios

Uso da função `rand()` para
preenchimento aleatório de dados

Números Pseudoaleatórios

- São bastante utilizados em programação para gerar comportamento aleatório para programas
- Exemplos de programas que usam esses números são **jogos** ou **simulações**
- Existem algoritmos para geração de números que são **aparentemente** aleatórios
- Na verdade são gerados a partir de funções matemáticas



Números Pseudo-Aleatórios em C

- A biblioteca `stdlib.h` inclui uma função simples para gerar números pseudoaleatórios

```
int rnd = rand();
```

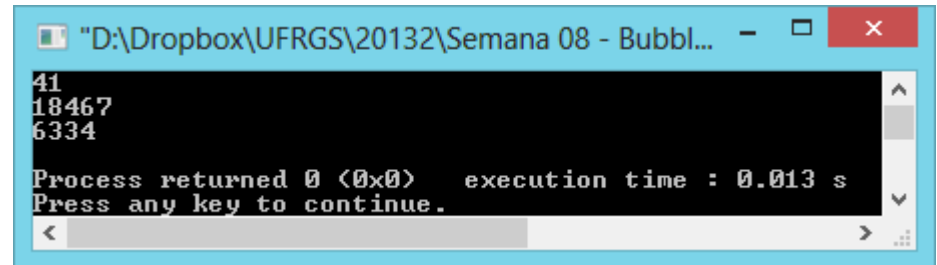
- Faz com que `rnd` receba um valor pseudoaleatório entre zero e `RAND_MAX`

Referência para o manual da função

<http://www.cplusplus.com/reference/cstdlib/rand/>

Exemplo de Código

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5
6      int rnd;
7
8      rnd = rand();
9      printf("%d\n", rnd);
10
11     rnd = rand();
12     printf("%d\n", rnd);
13
14     rnd = rand();
15     printf("%d\n", rnd);
16
17     return 0;
18 }
```



```
"D:\Dropbox\UFRGS\20132\Semana 08 - Bubbl...
41
18467
6334
Process returned 0 (0x0) execution time : 0.013 s
Press any key to continue.
```

Números Pseudo-Aleatórios em C

- Problema 1
 - Como toda a função matemática `rand()` gera sempre o mesmo resultado para uma mesma entrada
 - No caso, o primeiro número aleatório gerado é sempre 41
- Problema 2
 - Como gerar números aleatórios em um intervalo diferente de zero a `RAND_MAX` ?

Resolvendo o Problema 1

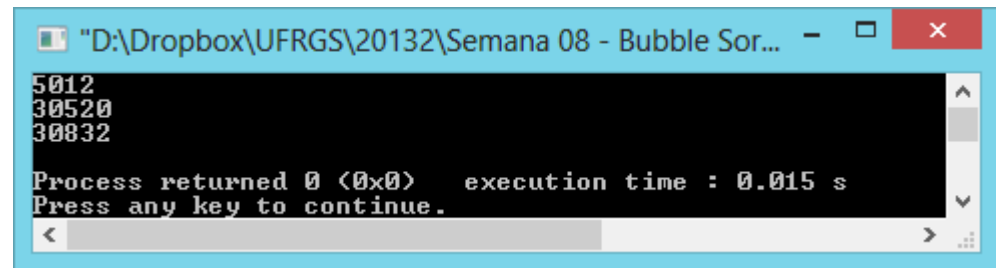
- A entrada para a função `rand()` é chamada de **semente aleatória**
- É possível inicializar a primeira semente com um valor realmente aleatório através da função `srand()`
- Seria uma boa ideia usar o tempo atual em segundos desde 1 de janeiro de 1970 UTC
 - (i.e., *unix timestamp atual*)

```
srand (time(NULL)) ;
```

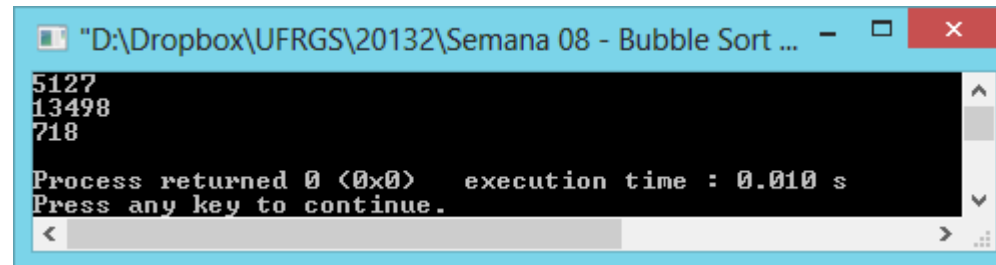
Nesse caso
incluir também
`time.h`

Exemplo de Código

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main() {
6
7      int rnd;
8      // Inicializa a semente aleatória com o tempo atual
9      srand(time(NULL));
10
11     rnd = rand();
12     printf("%d\n", rnd);
13
14     rnd = rand();
15     printf("%d\n", rnd);
16
17     rnd = rand();
18     printf("%d\n", rnd);
19
20     return 0;
21 }
```



```
"D:\Dropbox\UFRGS\20132\Semana 08 - Bubble Sor... - [X]
5012
30520
30832
Process returned 0 (0x0)   execution time : 0.015 s
Press any key to continue.
```



```
"D:\Dropbox\UFRGS\20132\Semana 08 - Bubble Sort ... - [X]
5127
13498
718
Process returned 0 (0x0)   execution time : 0.010 s
Press any key to continue.
```

Resolvendo o Problema 2

- Para gerar um número pseudoaleatório em um determinado intervalo basta usar o operador resto da divisão (%)

- Valor entre 0 e 9

`rnd = rand() % 10;`

- Valor entre -20 e 20

`rnd = rand() % 41 - 20;`

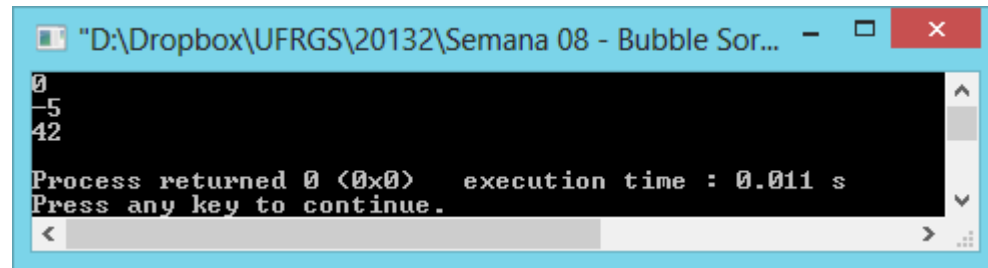
- Valor entre 1 e 100

`rnd = rand() % 100 + 1;`

O resto da divisão de **a** por **b** está sempre entre **0** e **b - 1**

Exemplo de Código

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main() {
6
7      int rnd;
8      // Inicializa a semente aleatória com o tempo atual
9      srand(time(NULL));
10
11     rnd = rand() % 10;
12     printf("%d\n", rnd);
13
14     rnd = rand() % 41 - 20;
15     printf("%d\n", rnd);
16
17     rnd = rand() % 100 + 1;
18     printf("%d\n", rnd);
19
20     return 0;
21 }
```



```
"D:\Dropbox\UFRGS\20132\Semana 08 - Bubble Sor... - [X]
0
-5
42
Process returned 0 (0x0)   execution time : 0.011 s
Press any key to continue.
```