



**República Bolivariana de Venezuela
Aldea Universitaria Liceo Fray Pedro de Agreda**

Prof. Elías Cisneros 23/08/09

**Lenguaje C++
Números aleatorios**



Usted es libre de:

- Copiar, distribuir y comunicar públicamente la obra .
- Hacer obras derivadas .

Bajo las condiciones siguientes:

	Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).
	No comercial. No puede utilizar esta obra para fines comerciales.
	Compartir bajo la misma licencia. Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.



Aleatoriedad

La aleatoriedad es un campo de definición que, en matemáticas, se asocia a todo proceso cuyo resultado no es previsible más que en razón de la intervención del azar. El resultado de todo suceso aleatorio no puede determinarse en ningún caso antes de que este se produzca. Por consiguiente, los procesos aleatorios quedan englobados dentro del área del cálculo de probabilidad. La palabra aleatorio se usa para expresar una aparente carencia de propósito, causa, u orden.

Los números aleatorios fueron investigados primero en el contexto de las apuestas, y muchos dispositivos aleatorizados tales como los dados, las cartas, y las ruletas, fueron primero desarrollados para ser usados en apuestas. La habilidad de producir justamente números aleatorios es vital a la apuesta electrónica, y como tal, los métodos usados para crearlos son usualmente regulador por instituciones gubernamentales.

Números aleatorios

Un número aleatorio es un resultado de una variable al azar especificada por una función de distribución. Cuando no se especifica ninguna distribución, se presupone que se utiliza la distribución uniforme continua en el intervalo $[0,1)$.

Un número pseudo-aleatorio es un número generado en un proceso que parece producir números al azar, pero no lo hace realmente. Las secuencias de números pseudo-aleatorios no muestran ningún patrón o regularidad aparente desde un punto de vista estadístico, a pesar de haber sido generadas por un algoritmo completamente determinista, en el que las mismas condiciones iniciales producen siempre el mismo resultado.

Un Generador de números aleatorios es un componente o funcionalidad que crea números o símbolos para un programa software en una forma que carezca de un patrón evidente, y que así parezcan ser números aleatorios. La mayor parte de los generadores de números aleatorios son, en realidad, pseudoaleatorios: se calcula (o introduce internamente) un valor X_0 , llamado generalmente semilla, y, a partir de él, se van generando X_1 , X_2 , X_3 . Siempre que se parta de la misma semilla, se obtendrá la misma secuencia de valores.

Número aleatorios en C++

A veces es útil usar números aleatorios en nuestros programas, el campo de aplicación puede ser bastante amplio, simular juegos de dados, rangos de temperaturas, calificaciones, edades, etc. Para realizar la implementación de números pseudo - aleatorios en lenguaje C++ es necesarios incorporar la librería `stdlib.h`, la generación de estos números es utilizaremos las siguientes funciones:

Función rand



Generador de números aleatorios. La función "rand" devuelve un número pseudo aleatorio entre 0 y **RAND_MAX**. La constante **RAND_MAX** está definida en `stdlib.h`. La secuencia de números retornada cada vez que es invocada la función `rand` es generado por un algoritmo determinístico, el algoritmo que genera la secuencia de números utiliza una semilla para cambiar dicha secuencia. La semilla se establece mediante la función `srand` que sea detallado más adelante.

Sintaxis:

`int rand(void);`

Valor de retorno:

La función `rand()` devuelve un número entero entre 0 y `RAND_MAX`.

Parametros de la función	Rango de valores
<code>rand() % 11;</code>	0 a 10
<code>rand () % 11 + 20;</code>	20 a 30
<code>rand()%11 -5</code>	-5 a 5
<code>rand () % (N-M+1) + M;</code>	Forma más general, entre M y N con N mayor que M. Obteniendo un número que está entre M y N.

Ejemplo 1. Un programa empleando números aleatorios es el siguiente:

```
1. using namespace std;
2. #include <iostream>
3. #include <stdlib.h>
4. int main(void)
5. {
6.     int i;
7.     int veces=10;
8.     int resultado;
9.     cout <<RAND_MAX<<endl;
10.    for (i=0;i<veces;i++)
11.    {
12.        resultado = rand()%11 -5;
13.        cout <<resultado<<endl;
14.    }
15.    cin.get();
16.    return 0;
17. }
```



Función srand()

Es una función que inicializa el generador de números aleatorios. Ahora bien, si ejecutamos varias veces nuestro programa, la secuencia de números aleatorios se repite. Supongamos que tenemos un programa que escribe 3 números aleatorios entre **0** y **10**. Lo ejecutamos una vez y sale, por ejemplo 4, 7 y 9, lo ejecutamos por segunda vez y vuelve a salir 4, 7 y 9; y la tercera vez sale lo mismo. Esto ocurre porque la función **rand()** calcula los números aleatorios a partir de un número inicial que como vimos anteriormente es llamado semilla, utiliza un algoritmo para el cálculo de primer número, para el segundo número, realiza unas operaciones basado en el resultado anterior y de esta forma obtiene el segundo número y así sucesivamente.

Si volvemos a ejecutar el programa desde el principio, el número inicial o la semilla que usa **rand()** es el mismo, con lo que la secuencia de números aleatorios es la misma, ya que las operaciones del algoritmo de cálculo de números pseudo aleatorios son las mismas. Para evitar este problema existe la función **srand()**, a la que se le pasa de parámetro un número que se utilizará como número inicial para las cuentas. A esta función sólo se debe llamar una vez en nuestro programa.

¿Qué número le ponemos a la función **srand()**?. No podemos ponerle un número fijo, porque entonces no hemos hecho nada. No podemos ponerle un número obtenido con **rand()**, porque la primera vez siempre nos dará el mismo y el resultado será igual que si le ponemos un número fijo. Debemos buscar la forma de obtener un número que sea distinto en la ejecución de cada programa. Hay dos mecanismos que se utilizan habitualmente para ello:

- La fecha/hora del sistema. Este valor cambia si ejecutamos el programa en distinto instante de tiempo. Tendríamos que arrancar el programa dos veces en el mismo segundo para obtener la misma secuencia de números aleatorios. En C de linux esta fecha/hora se obtiene con la función **time()** .

`srand (time(NULL));`

- El número de proceso del programa. El primer programa que se arranca cuando se enciende el computador con el sistema operativo linux, tiene el número de proceso 1, el segundo el 2, el tercero el 3 y así sucesivamente. Cuando arrancamos nuestro programa, se le asignará el número que le toque, por ejemplo, el 215. Cuando lo volvamos a arrancar, se le asignará el que le toque (puede ser 216 si no hemos ejecutado nada entre medias o 345, si nos hemos entretenido con otras cosas). Después de ejecutar nuestro programa varios miles de veces, el número de proceso puede que se repita, pero ya no nos acordaremos de la secuencia que se sacó la primera vez. El número de proceso se obtiene con **getpid()** .

`srand (getpid());`

Sintaxis:

`void srand(unsigned semilla);`

La función **srand** sirve para cambiar el origen del generador de números aleatorios.



Valor de retorno:

"srand" no devuelve ningún valor.

Ejemplo 2. El siguiente programa que utiliza la función srand con la semilla del valor retornado por la función time() para asegurar que en cada corrida los números generados por la función rand.

```
1. /* rand example: guess the number */
2. using namespace std;
3. #include <iostream>
4. #include <stdlib.h>
5. #include <time.h>
6. int main ()
7. {
8.     int iSecreto, iAdivina;
9.     /* Inicializa la semilla para el numero aleatorio*/
10.    srand ( time(NULL) );
11.    /* Se genera el numero secreto */
12.    iSecreto = rand() % 10 + 1;
13.    do {
14.        cout<<"Adivine el número del 1 al 10: "<<endl;
15.        cin>>iAdivina;
16.        if (iSecreto<iAdivina) cout<<"El numero secreto es menor"<<endl;
17.        else if (iSecreto>iAdivina) cout<<"El numero secreto es mayor"<<endl;
18.    } while (iSecreto!=iAdivina);
19.    puts ("Felicitaciones!");
20.    return 0;
21. }
```

Funciones para generar números aleatorios con decimales

Las funciones drand48() y srand48()

Para generar un número aleatorio con decimales entre 0.0 (incluido, puede salir el 0,0) y 1.0 (excluido, nunca saldrá 1.0). A partir de ahí es fácil transformarlo al rango que queramos. Si queremos un rango entre 10.0 y 20.0 (o entre M y N siendo N mayor que M y ambos con decimales, aunque sean .0)

```
#include <stdlib.h>
...
numero = drand48() * (20.0-10.0) + 10.0;
numero = drand48() * (N-M) + N;
```



De la misma forma que antes, la secuencia de números aleatorios se repetirá cada vez que ejecutemos nuestro programa y de igual manera, tenemos manera de cambiar la "semilla" de esa secuencia. Hay que llamar a la función `srand48()` pasándole un entero que sea distinto en cada ejecución del programa. Nuevamente, tenemos las dos opciones anteriores

```
srand48(time(NULL));  
srand48(getpid());
```

Ejercicios

Construya un programa que simule el peso de un gimnasio. El programa debe generar aleatoriamente la edad, estatura y peso de cada usuario. Utilice un ciclo de `do-while` con respuesta del operador para calcular los datos de un nuevo usuario. Al finalizar indique el promedio de edades, estaturas y pesos.

Referencias

- http://es.wikipedia.org/wiki/N%C3%BAmero_aleatorio
- http://es.wikipedia.org/wiki/Generador_de_n%C3%BAmeros_aleatorios
- http://es.wikipedia.org/wiki/N%C3%BAmero_pseudo-aleatorio
- <http://es.wikipedia.org/wiki/Aleatoriedad>
- <http://cplusplus.com/reference/cstdlib/rand/>
- <http://www.conclase.net/c/curso/index.php?cap=902d>
- <http://es.answers.yahoo.com/question/index?qid=20090711182720AAFlwa3>
- <http://www.chuidiang.com/clinix/funciones/rand.php>