文章编号:1009-3907(2008)03-0064-05

浅析C语言中的随机数问题

李柯景

(长春大学 计算机科学技术学院,吉林 长春 130022)

摘 要:介绍了C语言程序设计教学过程中遇到的随机数问题。C语言中可产生随机数的相关函数有四种,它们分别为rand、srand、random和randomize。本文阐述了四种函数的区别及应用,并总结了产生不同范围内随机数的方法。

关键词:随机数;伪随机数;种子;随机数生成器

中图分类号:TP311.11 **文献标识码**:A

0 引 言

经过几轮的C语言程序设计教学,笔者发现在教学过程中,很多实例都涉及到随机数问题。

例如:设计一个猜数游戏,由计算机产生一个随机数magic,若输入数字与magic相符,则输出"You are right!"。否则,输出"You are wrong!"

具体程序如下:

```
#include <stdio.h>
#include <stdlib.h>
main()
{ int guess,magic;
magic=rand();
scanf("%d",&guess);
if (guess==magic)
  printf("You are right!");
else
    printf("You are wrong!");
}
```

经过tc进行编译链接后,输入25,程序显示结果为346并换行显示You are wrong! 此后多次测试结果,无论输入整数范围内的任何值,magic的输出结果均为346。也就是说,只有当输入346时,才会显示You are right! 而我们编程的初衷是想要让程序实现猜数功能,这种显示结果显然并没有实现真正的随机!

又如:让屏幕上显示计算机产生的十个随机数。实现方法如下(程序1)所示:

```
#include "stdio.h"
#include" stdlib.h"
main()
{int i;
for (i=0;i<=9;i++)
   printf("%d ",rand());
```

收稿日期:2008-04-20

作者简介:李柯景(1980-),女,吉林省九台市人,长春大学计算机科学技术学院助教,硕士生,主要从事计算机基础教学研究。

```
}
```

经过编译、链接后,该程序的多次运行结果均为同样的10个数。以上两个实例说明,利用rand函数实现的系统所产生随机数是可预见的随机数!

1 C语言中的随机数函数

1.1 rand()函数

以上两个程序中的随机数都是由随机函数rand()产生的,它的值是0~RAND_MAX之间的一个正整数,RAND_MAX是在头文件<stdlib.R>中通过#define RAND_MAX 0x7FFF定义的符号常量,十六进制的0x7FFF换算成十进制为32767[1]。ANSI标准规定RAND_MAX的值不小于32767,具体的取值大小与编译环境有关,通常我们所使用的TurboC2.0中规定RAND_MAX的最大值为32767。

通过以上两例我们可以看到,使用rand()函数并不能产生真正的随机数。其实计算机不会产生绝对随机的随机数,只能产生"伪随机数"。其实绝对随机的随机数只是一种理想的随机数,计算机只能生成相对的随机数,即伪随机数。实际上,系统是将0~32767之间的整数"随意"地排成了一个"随机数表",程序第一次调用rand()函数是取"随机数表"中的第一个,第二次调用rand()函数是取"随机数表"的第二个,由此类推,程序1中产生的十个随机数,实际上是利用循环变量的变化十次调用了"随机数表"的前十个数。即使我们运行了多次,但都是在同一个系统上,而同一个系统产生的"随机数表"又始终是相同的,因此每一次运行的结果都是一样的。这就是我们把机器中产生的随机数称之为"伪随机数"的原因。

由此可见,以上两个程序要利用计算机实现随机数的功能,就必须使得每次运行结果产生的数不同,而单纯使用rand()函数是无法实现的。

1.2 srand()函数

通过上面的论述我们可以知道,在C语言中,rand()函数可以用来产生随机数,但这种随机数的产生是可以预见的,是一个伪随机数。是根据一个数,我们可以称它为种子,以它为基准以某个递推公式推算出来的一系数,当计算机正常开机后,这个种子的值是定了的,除非你破坏了系统。为了改变这个种子的值,C语言提供了srand()函数。

srand函数是随机数生成器的初始化函数,它的原型为:void srand(unsigned seed);它需要提供一个种子,如:srand(1);直接使用1来初始化种子。srand()函数使用seed作为种子,用来初始化随机数生成器[2]。只要把相同的种子传入srand(),然后调用rand()时,就会产生相同的随机数序列。若我们将程序1作如下修改:

```
#include"stdio.h"
#include"stdlib.h"
main()
{ int i;
    srand(1);
    for(i=0;i<=9;i++)
    printf("%d",rand());
}
```

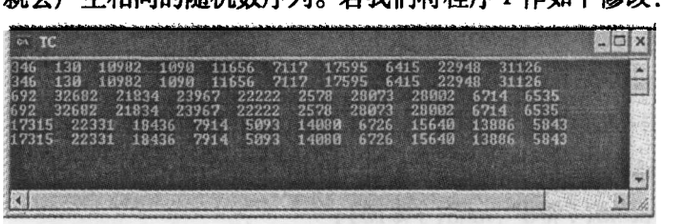

图1　程序运行结果

再将srand()内参数值分别修改为srand(2),srand(50),每个参数值分别运行两次,程序结果如图1所示。

由此可见,srand()函数中的seed不同,产生的随机数列也就不相同。反之,若srand()函数中的seed相同,那么产生的序列就是可以预见的。

种子是产生伪随机数的初始依据。种子相同产生的伪随机数也就相同。如果程序本身不设置种子,那么系统默认种子为1,因此每当需要更新的伪随机数的时候应重设种子,使得每次执行结果都能够产生不同的数列,即让结果看起来是实现了真正的随机效果。

重设种子的方法有很多,利用srand((unsigned)(time(NULL))是一种产生不同种子的不错选择。time()函数原型包含在系统头文件<time.h>中,功能是返回从1970年1月1日零时到当前时间为止的秒数。由于当前时间是时刻变化的,所以每次运行程序时调用srand函数的参数值均不同(种子(seed)不同),

从而产生的随机数也不同。则程序1修改如下：

```
#include" stdio. h"
#include" stdlib. h"
#include" time. h"
main( )
{ int i;
    srand( (unsigned)time(NULL));
    for(i =0;i < =9;i + + )
          printf(" % d ",rand( ));
}
```

通过上面的论述，我们可以知道：现在的C编译器都提供了基于ANSI标准的伪随机数生成器函数，用来生成随机数。它们就是rand( )和 srand( )函数。这两个函数的工作过程如下：首先给srand( )提供一个种子，它是一个unsigned int类型，其取值范围从0～65535；然后调用rand( )，它会根据提供给srand( )的种子值返回一个随机数(在0到32767之间)；根据需要多次调用rand( )，从而不间断地得到新的随机数。

1.3 randomize( )和random(num)的使用

在Turbo c2.0的<stdlib. h>中我们可以看到如下内容：

```
#define random(num) (rand( ) % (num))
#define randomize( ) srand( (unsigned)time(NULL))
```

由此可见，randomize( )函数同样也是随机数生成器初始化函数。只要我们在使用rand( )前调用randomize( )函数，同样也可以实现不可预见的随机数序列。

random(num)函数中参数的值决定了输出随机数的取值范围。如：

```
main( )
{……
randomize( );
printf(" % d",random(10));
……}
```

为输出一个[0,10)之间的随机数，若将random(num)中参数值改为100，为输出一个[0,100)之间的随机数。同样的，若程序改为：

```
main( )
{……
randomize( );
printf(" % d",rand( )%10);
……
}
```

也可以产生[0,10)之间的随机数。

由此可见，random(num)跟rand( )函数使用方法相近，但是它可以通过参数的设定直接限制随机数产生的范围。由于很多实际应用对于产生的随机数范围有所限制，比如猜数游戏，不能让系统产生的随机数范围过大。因此可以通过使用random函数来解决。

综上所述：C语言中可以产生随机数的方法可借助于以下四种函数来实现。

(1)int rand(void)

得到一个0到0x7FFF之间之间的随机数。

(2) int random(int num)

得到一个[0,num－1]之间的随机数。

(3) void randomize( )

通过time函数来得到一个随机数,此数将成为起始发生数据。可与rand、random配合使用。

(4) void srand(unsigned number)

该函数和rand函数配合使用,产生随机数的起始发生数据。

此外,使用随机数函数之前应注意在程序前加上编译预处理命令#include "stdlib.h"(如程序1所示)。

2 不同范围内随机数的产生

通过上面论述,C语言中利用以上函数产生的随机数无论多次运行重复与不重复它们都是整数。若想得到的不同范围内的随机数,需要在使用以上几种函数的同时,对程序加以限制。

(1)若想要产生[0,1)之间的随机数,可以使用rand()/(double)(RAND_MAX)或者rand()/(float)(RAND_MAX)的方法。

例如下面的程序:

```
……
main( )
{……
srand( (unsigned)time( NULL ) );
  for( i = 0; i < 10;i++ )
  printf( "%5.2f\n",rand()/32767.0);
}
```

如果想取更大范围的随机浮点数,比如0~100,则应将最后一条语句修改为:

```
printf( "%5.2f\n", rand()/32767.0*100);
```

(2)若想要产生[40,50)之间的随机数,程序如下:

```
……
main( )
{int i;
srand((unsigned)time(NULL));
  for(i = 0; i < 10;i++ )
  printf("%d\n", random(10) +40); }
```

该程序产生40~50之间的随机数,序列中可能包含40,但一定不包括50,若想在随机序列中含有50,则应将最后一条语句修改为printf("%d\n", random(11) +40);。以此类推,想要产生任意范围内的随机整数,只需要记住一个通用的公式。要取[a,b)之间的随机整数(包括a,但不包括b),使用:(rand() % (b - a)) + a或者random(b - a) + a。

同样的,若要产生[a,b)之间的随机浮点数,也可使用同样的方法。如若想要产生[40,50)之间的随机浮点数,则应将最后一条语句修改为:printf("%5.2f\n",rand()/32767.0*10 +40); 那么,同样的我们也只需要记住一个通用的公式。要取[a,b)之间的随机整数(包括a,但不包括b),使用:(rand()/(float)(RAND_MAX)*(b - a)) + a或者random(RAND_MAX) /(float)(RAND_MAX)*(b - a) +a。

3 不重复随机数的产生

很多时候,我们要求产生的随机数序列不仅要在一定的限制范围内,还需要满足无重复值的出现,而利用上面介绍过的随机数函数并不能够满足这个要求。为了能够产生不重复的随机数序列可以采用不同的编程方法来实现。

比如:要产生10个不重复的随机数序列。可以使用如下算法:

(1)设置数组a[10],把0~9这10个数依次预先存放在数组a中;

(2)再利用随机数函数产生的随机数作为数组的下标;产生一个随机下标,我们就取出对应数组中的值输出;

(3)然后再依次用数组最后一个值来替换该下标数组中的值;

(4)再将产生随机下标的范围减少1;

(5)重复执行(2)(3)(4)操作。

参考程序如下:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 10
main()
{ int a[N];
int x, i;
for (i=0; i<N; i++)
   a[i] = i;
srand((int)time(NULL));
for (i=0; i<N; i++)
{ x = random(N-i);
printf("%d ", a[x]);
a[x] = a[N-1-i];
}
}
```

修改符号常量N的值,即可求得更大范围内的不重复的随机序列。当然,产生不重复随机数的方法不仅一种,以上程序仅供参考。

4 结 语

我们在日常生活中也经常会遇到关于随机数的问题,如随机抽奖、随机点名等等。本文介绍了C语言中产生随机数的几种函数的使用及产生不同范围内和不重复随机数的方法,对一些实例应用过程中可预见的结果进行了分析,加强了学生对计算机产生伪随机数概念的理解,有助于学生更好地掌握随机数函数,从而为现实生活中利用随机数函数解决实际问题打下了基础。

参考文献:

[1] 李丽娟.C语言程序设计教程[M].北京:人民邮电出版社, 2008.
[2] 马保东. C语言程序设计基础[M].天津:南开大学出版社, 2006.
[3] 王晓东.算法设计与分析[M].北京:清华大学出版社, 2005.
[4] 张淑梅,李勇.计算机产生随机数的方法[J].数学通报,2006,45(3):44-45.
[5] 苏小红,陈惠鹏,温东新,等.C语言程序设计教程[M]. 北京:电子工业出版社, 2004.

责任编辑:钟 声

Brief analysis of random number question in C language

LI Ke-jing
(Computer Science and Technology Instiute, Changchun University, Changchun 130022, China)

Abstract: This paper introduces random number question in the C language teaching process. There are four kinds of random number function in C language. They are rand, srand, randomand and randomize. This article elaborates the differences and the application of these functions, and summarizes the method of the different scope of random number.

Keywords: random number; pseudo random number; seed; random number generator