NAME

drand48, erand48, lrand48, nrand48, mrand48, jrand48, srand48, seed48, lcong48 – generate uniformly distributed pseudo-random numbers

SYNOPSIS

#include <stdlib.h>

double drand48(void);

double erand48(unsigned short *xsubi***[3]);**

long lrand48(void);

long nrand48(unsigned short *xsubi***[3]);**

long mrand48(void);

long jrand48(unsigned short *xsubi***[3]);**

void srand48(long *seedval***);**

unsigned short *seed48(unsigned short *seed16v***[3]);**

void lcong48(unsigned short *param***[7]);**

MT-LEVEL

Safe

DESCRIPTION

This family of functions generates pseudo-random numbers using the well-known linear congruential algorithm and 48-bit integer arithmetic.

Functions **drand48( )** and **erand48( )** return non-negative double-precision floating-point values uniformly distributed over the interval [0.0, 1.0).

Functions **lrand48( )** and **nrand48( )** return non-negative long integers uniformly distributed over the interval $[0, 2^{31})$.

Functions **mrand48( )** and **jrand48( )** return signed long integers uniformly distributed over the interval $[-2^{31}, 2^{31})$.

Functions **srand48( )**, **seed48( )**, and **lcong48( )** are initialization entry points, one of which should be invoked before either **drand48( )**, **lrand48( )**, or **mrand48( )** is called. (Although it is not recommended practice, constant default initializer values will be supplied automatically if **drand48( )**, **lrand48( )**, or **mrand48( )** is called without a prior call to an initialization entry point.) Functions **erand48( ), nrand48( ),** and **jrand48( )** do not require an initialization entry point to be called first.

All the routines work by generating a sequence of 48-bit integer values, X_i, according to the linear congruential formula

$$X_{n+1} = (aX_n + c)_{\mathrm{mod}\ m} \qquad n \geq 0.$$

The parameter $m = 2^{48}$; hence 48-bit integer arithmetic is performed. Unless **lcong48( )** has been invoked, the multiplier value a and the addend value c are given by

$$a = \mathrm{5DEECE66D}_{16} = 273673163155_{8}$$
$$c = \mathrm{B}_{16} = 13_{8}.$$

The value returned by any of the functions **drand48( )**, **erand48( )**, **lrand48( )**, **nrand48( )**, **mrand48( )**, or **jrand48( )** is computed by first generating the next 48-bit X_i in the sequence. Then the appropriate number of bits, according to the type of data item to be returned, are copied from the high-order (leftmost) bits of X_i and transformed into the returned value.

The functions **drand48( )**, **lrand48( )**, and **mrand48( )** store the last 48-bit X_i generated in an internal buffer. X_i must be initialized prior to being invoked. The functions **erand48( )**, **nrand48( )**, and **jrand48( )** require the calling program to provide storage for the successive X_i values in the array specified as an argument when the functions are invoked. These routines do not have to be initialized; the calling program must place the desired initial value of X_i into the array and pass it as an argument. By using different arguments, functions **erand48( )**, **nrand48( )**, and **jrand48( )** allow separate modules of a large program to generate several *independent* streams of pseudo-random numbers, that is, the sequence of numbers in each stream will *not* depend upon how many times the routines have been called to generate numbers for the other streams.

The initializer function **srand48( )** sets the high-order 32 bits of X_i to the 32 bits contained in its argument. The low-order 16 bits of X_i are set to the arbitrary value $330E_{16}$.

The initializer function **seed48( )** sets the value of X_i to the 48-bit value specified in the argument array. In addition, the previous value of X_i is copied into a 48-bit internal buffer, used only by **seed48( )**, and a pointer to this buffer is the value returned by **seed48( )**. This returned pointer, which can just be ignored if not needed, is useful if a program is to be restarted from a given point at some future time — use the pointer to get at and store the last X_i value, and then use this value to reinitialize using **seed48( )** when the program is restarted.

The initialization function **lcong48( )** allows the user to specify the initial X_i, the multiplier value a, and the addend value c. Argument array elements *param[0-2]* specify X_i, *param[3-5]* specify the multiplier a, and *param[6]* specifies the 16-bit addend c. After **lcong48( )** has been called, a subsequent call to either **srand48( )** or **seed48( )** will restore the ‘‘standard’’ multiplier and addend values, a and c, specified above.

SEE ALSO

rand(3C)