## NAME

**random**, **srandom**, **srandomdev**, **initstate**, **setstate** - non-cryptographic pseudorandom number generator; routines for changing generators

## LIBRARY

Standard C Library (libc, -lc)

## SYNOPSIS

**#include <stdlib.h>**

*long*
**random**(*void*);

*void*
**srandom**(*unsigned int seed*);

*void*
**srandomdev**(*void*);

*char **
**initstate**(*unsigned int seed*, *char *state*, *size_t n*);

*char **
**setstate**(*char *state*);

## DESCRIPTION

**The functions described in this manual page are not secure. Applications which require unpredictable random numbers should use arc4random(3) instead.**

Unless initialized with less than 32 bytes of state, the **random**() function uses a non-linear additive feedback random number generator employing a default table of size 31 long integers to return successive pseudo-random numbers in the range from 0 to (2**31)-1. The period of this random number generator is very large, approximately 16*((2**31)-1).

If initialized with less than 32 bytes of state, **random**() uses the poor-quality 32-bit Park-Miller LCG.

The **random**() and **srandom**() functions are analagous to rand(3) and srand(3).

Like rand(3), **random**() is implicitly initialized as if **srandom**(*1*) had been invoked explicitly.

The **srandomdev**() routine initializes the state array using random numbers obtained from the kernel. This can generate states which are impossible to reproduce by calling **srandom**(), because the succeeding terms in the state buffer are no longer derived from the Park-Miller LCG algorithm applied to a fixed seed.

The **initstate**() routine initializes the provided state array of *uint32_t* values and uses it in future **random**() invocations. (Despite the *char* * type of *state*, the underlying object must be a naturally aligned array of 32-bit values.) The size of the state array (in bytes) is used by **initstate**() to decide how sophisticated a random number generator it should use -- the more state, the better the random numbers will be. (Current "optimal" values for the amount of state information are 8, 32, 64, 128, and 256 bytes; other amounts will be rounded down to the nearest known amount. Using less than 8 bytes will cause an error.) The *seed* is used as in **srandom**(). The **initstate**() function returns a pointer to the previous state information array.

The **setstate**() routine switches **random**() to using the provided state. It returns a pointer to the previous state.

Once a state array has been initialized, it may be restarted at a different point either by calling **initstate**() (with the desired seed, the state array, and its size) or by calling both **setstate**() (with the state array) and **srandom**() (with the desired seed). The advantage of calling both **setstate**() and **srandom**() is that the size of the state array does not have to be remembered after it is initialized.

With 256 bytes of state information, the period of the random number generator is greater than 2**69 which should be sufficient for most purposes.

## DIAGNOSTICS

If **initstate**() is called with less than 8 bytes of state information, or if **setstate**() detects that the state information has been garbled, NULL is returned.

## SEE ALSO

arc4random(3), lrand48(3), rand(3), random(4)

## HISTORY

These functions appeared in 4.2BSD.

## AUTHORS

Earl T. Cohen