

NAME

`drand48_r`, `erand48_r`, `lrand48_r`, `nrand48_r`, `mrnd48_r`, `jrand48_r`, `srand48_r`, `seed48_r`, `lcong48_r` - generate uniformly distributed pseudo-random numbers reentrantly

SYNOPSIS

```
#include <stdlib.h>

int drand48_r(struct drand48_data *buffer, double *result);

int erand48_r(unsigned short xsubi[3],
             struct drand48_data *buffer, double *result);

int lrand48_r(struct drand48_data *buffer, long int *result);

int nrnd48_r(unsigned short int xsubi[3],
            struct drand48_data *buffer, long int *result);

int mrnd48_r(struct drand48_data *buffer, long int *result);

int jrand48_r(unsigned short int xsubi[3],
            struct drand48_data *buffer, long int *result);

int srand48_r(long int seedval, struct drand48_data *buffer);

int seed48_r(unsigned short int seed16v[3],
            struct drand48_data *buffer);

int lcong48_r(unsigned short int param[7],
            struct drand48_data *buffer);
```

Feature Test Macro Requirements for glibc (see [feature_test_macros\(7\)](#)):

All functions shown above: `_SVID_SOURCE` || `_BSD_SOURCE` || `_XOPEN_SOURCE`

DESCRIPTION

These functions are the reentrant analogs of the functions described in [drand48\(3\)](#). Instead of modifying the global random generator state, they use the supplied data *buffer*.

Before the first use, this struct must be initialized, for example, by filling it with zeros, or by calling one of the functions `srand48_r()`, `seed48_r()`, or `lcong48_r()`.

RETURN VALUE

The return value is 0.

ATTRIBUTES

Multithreading (see [pthreads\(7\)](#))

The `drand48_r()`, `erand48_r()`, `lrand48_r()`, `nrand48_r()`, `mrnd48_r()`, `jrand48_r()`, `srand48_r()`, `seed48_r()`, and `lcong48_r()` functions are thread-safe.

CONFORMING TO

These functions are GNU extensions and are not portable.

SEE ALSO

[drand48\(3\)](#), [rand\(3\)](#), [random\(3\)](#)

COLOPHON

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <http://www.kernel.org/doc/man-pages/>.