



Linux Ubuntu 26.04 Manual Pages on command 'srand.3'

\$ man srand.3

rand(3) Library Functions Manual rand(3)

NAME

rand, rand_r, srand - pseudo-random number generator

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#include <stdlib.h>

int rand(void);

void srand(unsigned int seed);

[[deprecated]] int rand_r(unsigned int *seedp);
```

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

rand_r():

Since glibc 2.24:

_POSIX_C_SOURCE >= 199506L

glibc 2.23 and earlier

_POSIX_C_SOURCE

DESCRIPTION

The rand() function returns a pseudo-random integer in the range 0 to RAND_MAX inclusive (i.e., the mathematical range [0, RAND_MAX]).

The srand() function sets its argument as the seed for a new sequence of pseudo-random integers to be returned by rand(). These sequences are repeatable by calling srand() with the same seed value.

If no seed value is provided, the rand() function is automatically seeded with a value of 1.

RETURN VALUE

ATTRIBUTES

For an explanation of the terms used in this section, see `attributes(7)`.

? Interface	? Attribute	? Value	?
-------------	-------------	---------	---

? rand(), rand_r(), srand() ? Thread safety ? MT-Safe ?

VERSIONS

STANDARDS

rand()

rand_r()

Page 2/4

HISTORY

rand()

srand()

SVr4, 4.3BSD, C89, POSIX.1-2001.

rand_r()

POSIX.1-2001. Obsolete in POSIX.1-2008.

EXAMPLES

POSIX.1-2001 gives the following example of an implementation of rand() and srand(), possibly

useful when one needs the same sequence on two different machines.

```
static unsigned long next = 1;

/* RAND_MAX assumed to be 32767 */
int myrand(void) {
    next = next * 1103515245 + 12345;
    return((unsigned) (next/65536) % 32768);
}

void mysrand(unsigned int seed) {
    next = seed;
}
```

The following program can be used to display the pseudo-random sequence produced by rand()

when given a particular seed. When the seed is -1, the program uses a random seed.

```
#include <stdio.h>

#include <stdlib.h>

int
main(int argc, char *argv[])
{
    int r;
    unsigned int seed, nloops;
    if (argc != 3) {
        fprintf(stderr, "Usage: %s <seed> <nloops>\n", argv[0]);
        exit(EXIT_FAILURE);
    }
    seed = atoi(argv[1]);
    nloops = atoi(argv[2]);
```

```
if (seed == -1) {  
    seed = arc4random();  
    printf("seed: %u\n", seed);  
}  
srand(seed);  
for (unsigned int j = 0; j < nloops; j++) {  
    r = rand();  
    printf("%d\n", r);  
}  
exit(EXIT_SUCCESS);  
}
```

SEE ALSO

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

Linux man-pages 6.17

2026-02-08

[rand\(3\)](#)