

Random Number Generation

Ritankar Mandal

NAME

```
1 #include <stdlib.h>
2
3 int rand(void);
4 int rand_r(unsigned int *seedp);
5 void srand(unsigned int seed);
```

DESCRIPTION: `rand()`

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

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

DESCRIPTION: `srand()`

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.

The `srand()` function returns no value.

NOTES

The versions of `rand()` and `srand()` in the Linux C Library use the same random number generator as `random(3)` and `srandom(3)`, so the lower-order bits should be as random as the higher-order bits.

However, on older `rand()` implementations, and on current implementations on different systems, the lower-order bits are much less random than the higher-order bits. Do not use this function in applications intended to be portable when good randomness is needed. (Use `random(3)` instead.)

Generate random fractional number between 0 and 1 inclusive

```
1  /* returns a random fractional number between 0 and 1
   *   inclusive */
2  double randGen()
3  {
4      double temp;
5      temp = rand();
6      return (temp/RAND_MAX);
7  }
```

Generate random integer number between lower and upper inclusive

```
1  /* returns a random integer number between lower and
   2     upper inclusive */
3  int randGenRangeInt(int lower, int upper)
4  {
5     int range, randNum;
6     double temp;
7
8     temp = rand();
9     range = upper - lower + 1;
10    randNum = ((int)temp % range) + lower;
11
12    return randNum;
}
```