What is a Re-entrant procedure?

What is a re entrant procedure and can you give an example scenario of when it is used?

Edit: Also, can multiple processes access a re entrant procedure in **parallel**?

Please provide a different way of explaining than wikipedia as I don't totally understand their description hence my question *here*

c procedure

edited Aug 11 '11 at 12:53

asked Aug 11 '11 at 12:30


rrazd
645 10 28

4 Answers

The idea behind re-entrancy is that the routine may be called while it is in the middle of executing already and it will still work right.

Generally this is achieved by it using only parameters and local variables declared on the stack (in C terms, no `static` locals). It would also be important that it not lock any global resources during execution.

Now, you may ask, "How would such a weird thing as a routine being run multiple times at once happen?" Well, some ways this could happen are:

- The routine is recursive (or mutually-recursive with some other set of routines).
- It gets called by another thread.
- It gets called by an interrupt.

If any of these happen, and the routine is modifying a global (or C `static` local), then the new execution could potentially wipe out the changes the first execution made. As an example, if that global was used as a loop control variable, it might cause the first execution, when it finally gets to resume, to loop the wrong number of times.

edited Aug 11 '11 at 12:42

answered Aug 11 '11 at 12:35


T.E.D.
27.8k 3 37 92

Note that reentrancy os not necessary for your case (2) (It gets called by another thread). Locking is always sufficient for that case, because either thread can make progress independently. Recursive calling, and calls from signal handlers, are different in that the "outer" call cannot make progress until the "inner" call returns. – R.. Aug 11 '11 at 12:50

Note that a non-re-entrant function can be thread safe by using thread local storage. – Maxim Egorushkin Aug 11 '11 at 13:00

It is a subroutine which can be called when it is already active. For instance recursive

functions are often reentrant. Functions which are called from signal handlers must be reentrant as well. A reentrant function is thread-safe but not all thread-safe one are reentrant.

edited Aug 11 '11 at 12:54

answered Aug 11 '11 at 12:34

AProgrammer

32.4k 3 46 98

Recursive functions are not necessarily reentrant. For example: `int fact(int x) { static int tmp = fact(x-1); return x>1 ? x*tmp : 1; }` – R.. Aug 11 '11 at 12:52

@R..., changed as I agree, but your example is buggy (static variables are initialized only at the first call). – AProgrammer Aug 11 '11 at 12:55

Sorry, I was being lazy trying to fit it into a comment. The correct version is: `int fact(int x) { static int tmp; tmp = fact(x-1); return x>1 ? x*tmp : 1; }` – R.. Aug 11 '11 at 13:04

@AProgrammer, variables of `static` storage duration are initialized *before* the first call, usually at compile time or in some cases at program startup. – Jens Gustedt Aug 11 '11 at 16:29

@Jens, right, I forgot that difference between C and C++. – AProgrammer Aug 11 '11 at 16:35

Have a look at following URL

http://www.eetimes.com/discussion/beginner-s-corner/4023308/Introduction-to-Reentrancy

answered Aug 11 '11 at 12:35

cyber_raj

730 5 16

A reentrant procedure is one in which a single copy of the program code can be shared by multiple users during the same period of time. Re entrance has two key aspects: The program code cannot modify itself and the local data for each user must be stored separately.

In a shared system, reentrancy allows more efficient use of main memory: One copy of the program code is kept in main memory, but more than one application can call the procedure. Thus, a reentrant procedure must have a permanent part(the instructions that make up the procedure) and a temporary part(a pointer back to the calling program as well as memory for local variables used by the program).

Each execution instance, called activation, of a procedure will execute the code in the permanent part but must have its own copy of local variables and parameters. The temporary part associated with a particular activation is referred to as an activation record.

The most convenient way to support reentrant procedures is by means of a stack. When a reentrant procedure is called, the activation record becomes part of the stack frame that is created on procedure call

answered Apr 1 '12 at 21:22

user731914