

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Take the 2-minute tour



What kind of code can be called "re-entrant"?

Braintree_PayPal
**START ACCEPTING APPLE PAY, PAYPAL,
 BITCOIN, VENMO, CARDS AND WHATEVER'S NEXT.**

INTEGRATE NOW

Could someone tell me what code can be called "re-entrant" code? I came across this word when reading some real time operating system. What disciplines must be stuck to in order for code to be "re-entrant" code?

Many thanks =8^D

operating-system

edited Feb 16 '10 at 16:13



GWLlosa

11.2k 7 46 83

asked Feb 16 '10 at 16:06



smwikipedia

6,179 14 76 177

3 strangely enough the first google hit: [en.wikipedia.org/wiki/Reentrant_\(subroutine\)](http://en.wikipedia.org/wiki/Reentrant_(subroutine)) – Mitch Wheat Feb 16 '10 at 16:09

@Mitch, and if someone excerpts the bit that's relevant to RTOS rewords it and makes it specific to the "disciplines that must be stuck to" bit of the question, then hopefully the first google hit will become this question and thus SO .. which we all want, right? =) – Rob Feb 16 '10 at 16:10

@Rob: maybe. It's hard to say..... – Mitch Wheat Feb 16 '10 at 16:12

..it somewhat depends on the direction SO goes, it's business model, whether it will 'have it's day'...etc. – Mitch Wheat Feb 16 '10 at 16:14

1 also, given SO's not particularly good search ability, a Wiki still seems like a perfectly valid repository of knowledge... – Mitch Wheat Feb 16 '10 at 16:16

7 Answers

In general, a re-entrant block of code is one that can be entered by another actor before an earlier invocation has finished, without affecting the path that the first actor would have taken through the code. That is, it is possible to *re-enter* the code while it's already running and still produce correct results.

In most cases, the "actors" are threads of the same process, but the concepts of **thread safety** and re-entrant are subtly different: not every thread-safe block is re-entrant, but every re-entrant block is thread-safe. That is, re-entrancy is a stronger property than thread safety. Here's a **good example** from Raymond Chen of how a block of code might be thread-safe but not re-entrant.

There's a special case when the code is recursive: the same actor is calling into the code before its own invocation is finished, as Marc Gravell points out. All correct recursive blocks are re-entrant; of course, not every re-entrant block is recursive.

edited Apr 26 '12 at 18:41

answered Feb 16 '10 at 16:11



John Feminella

128k 25 223 280

thanks for all the replies, however, i can only mark 1 reply as the answer. sorry for that.. :D – smwikipedia Feb 22 '10 at 5:56

Not all recursive blocks are re-entrant. On some 8x51 compilers, local variables are stored at static addresses. Routines that can be called recursively have an "in use" flag; if the routine enters when the "in use" flag is set, it will copy its variables to a stack and clear the flag. Before the routine makes a call that could recurse, it sets the "in-use" flag; once that call returns, if the flag is not set, the routine re-loads its variables off the stack. Routines coded this way support recursion, but are not re-entrant. – supercat May 15 '12 at 15:38

1 "...but every re-entrant block is thread-safe." Would you mind taking a look at the [wikipedia entry about](http://en.wikipedia.org/wiki/Thread-safe)

[reentrancy](#)? Because there's an example of a function which is re-entrant but not thread safe. Is it possible that the editor made a mistake?. Thank you. – [Hernán Erasmó](#) Jul 7 '14 at 11:47

Same question as @HernánErasmó. The more I read, the more it seems that reentrant and thread-safe *can* be exclusive; i.e., exclusively one or the other. See also this Qt article which says exactly the opposite of what you (@John) say: "Hence, a thread-safe function is always reentrant, but a reentrant function is not always thread-safe." qt-project.org/doc/qt-4.8/threads-reentrancy.html – [Hawkeye Parker](#) Aug 6 '14 at 22:53

\$ git push origin **stackoverflowcareers**

 Add repos

John Feminella's answer says:

a re-entrant block of code is one that can be entered by another actor before an earlier invocation has finished. That is, it is possible to re-enter the code while it's already running.

But that is also true of non-re-entrant block of code. If the block of code has been written without regard to this issue, it will still be possible for a second actor to enter it simultaneously.

The issue is what effect this has on the results of either invocation. So more accurately: a re-entrant block is one that can be entered by another actor before an earlier invocation has finished, *without changing the outcome* of either invocation.

Neither invocation should be able to detect the "presence" of the other.

answered Feb 16 '10 at 16:24



[Daniel Earwicker](#)

67k 19 129 221

1 I didn't notice this answer until just now! Thanks for the clarification; you are right that I could have been more specific. I'll amend accordingly. – [John Feminella](#) Apr 26 '12 at 18:41

Virtually any kind of recursive code could be classified as reentrant (i.e. you can call back into the same method without having finished it), but this is used *in particular* when talking about locks, mutex, semaphores etc. For example, a lock is re-entrant if *once you have the lock* you can successfully "lock" the code again (i.e. you don't deadlock yourself) - for example:

```
public void AddIfNecessary(string s) {
    lock(syncObj) {
        if(!Contains(s)) Add(s);
    }
}

public void Add(string s) {
    lock(syncObj) {
        list.Add(s);
    }
}

public bool Contains(string s) {
    lock(syncObj) {
        return list.Contains(s);
    }
}
```

Here the fact that the lock is re-entrant means we can call `Contains` and `Add` without worrying that we already have the "exclusive" lock, making the code simpler. Internally, a counter is used rather than a simple "in use" flag.

answered Feb 16 '10 at 16:12



[Marc Gravell](#) ♦

519k 104 1396 191!

re entrant code is when the pages shares a common resource and the resource should not be changed or manipulated.then this resource is known as re entrant code or pure code

answered Nov 5 '13 at 13:16



Sreekamalesh

21 1

Can another thread call the code while a first thread is in the middle of running it? If the code yields to a callback function, can the callback function itself call the code before the first runthrough has completed?

If the code uses global vars that aren't locked down, or has its own static vars that it doesn't take special precautions with, any of those scenarios might break it.

answered Feb 16 '10 at 16:12



dubiousjim

2,490 1 13 25

A computer program is called reentrant if it can be interrupted in the middle of its execution and then safely called again before its previous invocations complete execution. The interruption could be caused by an internal action such as a jump or call, or by an external action such as a hardware interrupt or signal. Once the reentered invocation completes, the previous invocations will resume correct execution.

answered Jun 8 '12 at 17:37



ANIK SINGH

1

Add reference to Wikipedia? [en.wikipedia.org/wiki/Reentrant_\(subroutine\)](http://en.wikipedia.org/wiki/Reentrant_(subroutine)) – greeness Oct 22 '12 at 2:06

Code that can be called by different threads running in parallel. So, the code:

1. **can have** local variables (allocated on each thread's stack)
2. **should have** guarded global and static variables, since the threads will be sharing them and there will be a race condition here.

answered Mar 21 '14 at 17:04



brokenfoot

3,906 1 7 23