Re: what's a callback?

Source: http://sci.tech−archive.net/Archive/sci.electronics.design/2004−12/4325.html

---

From: Active8 (*reply2group_at_ndbbm.net*)
Date: 12/21/04

Date: Tue, 21 Dec 2004 12:10:32 −0500

On Mon, 20 Dec 2004 08:17:30 −0800, John Larkin wrote:

> On Mon, 20 Dec 2004 10:12:58 GMT, "Anthony Fremont"
> <spam@anywhere.com> wrote:
>
>>John Larkin wrote:
>>> On Sun, 19 Dec 2004 21:50:21 GMT, "Anthony Fremont"
>>> <spam@anywhere.com> wrote:
>>>
>>>>
>>>> "Fred Bloggs" <nospam@nospam.com> wrote in message
>>>>
>>>>> No− that is called "setting a flag to remind you to finish some
*>>>>> chores" and is an example of *re−entrant code*:−) Why in the world*
>>>>> would that be called "callback" unless you collect useless jargon?
>>>>
>>>> That's certainly the most unique definition of re−entrancy that I've
>>>> seen.
>>>>
>>>> When I think of re−entrant code, I think of code that has no local
>>>> variable storage associated to it. I also think of code that can
>>>> call itself recursively or be executed in several threads across
>>>> multiple processors concurrently with only one copy in memory.
>>>
>>>
>>> AKA "pure" code.
>>
>>Well.....I guess if it were to be really pure code, all addresses would
>>have to be relative to the instruction counter. ;−) We used to call
>>this floatable code as it could be just plunked into memory anywhere and
>>executed.
>
> My (perhaps non−professional−programmer) definition of that is
> "relocatable" or "PIC" (position independent) code.

Right. As opposed to "absolute", IIRC.
>
> "Pure" code is code that has no associated statics and if of course

> *not self−modifying, so that it can be executed by multiple threads*
> *without hassle.*
>

"Static" referes to local variables that retain their values between calls. Static linking of a function (declared in a class declaration) means that all instances of that class use the same function at the same location.

My def?

Reentrant code means that a program can have more than one thread executing concurrently.

http://vergil.chemistry.gatech.edu/resources/programming/threads.html

Eh? Kinda general really.

If you do something to modify a variable and another thread executes the function it may or may not get a valid value. First, "concurrently". It's an illusion for a single processor system. They get a time slice. But when the threads pause, the code where execution left off is reentered.

```
--
Best Regards,
Mike
```