

Re: Is threading the right solution for this challenge?

## Re: Is threading the right solution for this challenge?

---

*Source:* <http://coding.derkeiler.com/Archive/Cobol/comp.lang.cobol/2006-05/msg00114.html>

---

- *From:* "Chris" <[ctaliercio@xxxxxxxxx](mailto:ctaliercio@xxxxxxxxx)>
  - *Date:* 4 May 2006 12:07:01 -0700
- 

Thanks Oliver – I appreciate the feedback.

The problem I am facing is that it is a complex routing that is called in order to refresh the screen. There are several tables in an Oracle database that need to be interrogated, several OS processes that need to be examined, etc. A typical screen "refresh" could take up to 10–15 seconds.

I didn't want the user to have to wait 10–15 seconds between their input and the application's reaction to it. That is why I was leaning toward threads. I understand that there can be some overlapping of memory resources, and some unexpected results if you are not careful in your coding, but I believe MF gives me a way out of that mess with the THREAD-LOCAL-STORAGE section.

What I was envisioning was the main program spawning two threads – one that continually updates the display, and the other that simply accepts/reacts to user input. If the user chose to exit, I could kill the display thread (since it is only inquiring there is no danger there) and have the application exit immediately.

To get the same type behavior in a single program, I'd need to have the loop that is collecting information to update the display continuously interrupt itself to check for user input and react to it. That didn't seem like the most efficient way for me to accomplish what I am looking to do.

Does anyone else have any opinions on this?

Thanks,  
Chris

Oliver Wong wrote:

"Chris" <[ctaliercio@xxxxxxxxx](mailto:ctaliercio@xxxxxxxxx)> wrote in message  
[news:1146500082.954862.255300@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:1146500082.954862.255300@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)

Re: Is threading the right solution for this challenge?

Re: Is threading the right solution for this challenge?

Compiler: MF SE 4.2 SP2

Platform: HP-UX 11i

Challenge: Have a continually refreshing display screen that also responds to user input.

I had been considering using ACCEPT with a TIMEOUT clause and refreshing after each time out, but that does not seem to be the optimal solution here. In theory, this would give me a sluggish application. Either the user would be waiting on the compilation of display data, or the display would become "stale" while processing user input.

I'm thinking that this is a prime candidate for me to tackle my first threaded application. I figure I can run the user input interface in one thread, while running the continually updating output display in another thread.

I don't have much COBOL experience, but the functionality you're referring to is commonly seen in games (e.g. update the screen, check if the user entered any input, and if so process it; either way, update the screen again and repeat). Usually in game development, you want to avoid threads where ever possible (an exception might be heavy duty 3D graphics processing, which I don't think applies to your case), as they tend to just add complexity without bringing too much benefit.

In ACCEPT with TIMEOUT, what happens if the user was in the process of entering data, and then the timeout occurs? Do you get the data that the user half-submitted, or do you get nothing? If it's the former, you could probably use ACCEPT with TIMEOUT, with a very low timeout (e.g. 1/100th of a second?), and just capture 1 character a time, constructing the full string of the user's input manually. If it's the latter, you might want to look into hooking into a library or API written in another language to provide you with the facility to capture characters one a time. See for example the "getch() function with no-delay" at [http://www.mksoftware.com/docs/man3/curs\\_getch.3.asp](http://www.mksoftware.com/docs/man3/curs_getch.3.asp)

Since I have ZERO experience in this arena, I wanted to throw it out there for the masses and see what everyone thought.

In the meantime I'll be wading through the MF documentation on writing threaded applications and try to discern what I can. What is the typical method when designing a threaded application? Do I start by creating the two individual applications and then work toward linking them together, or are there other considerations?

Re: Is threading the right solution for this challenge?

There are many considerations when writing multithreaded code, and it's one of those things that's hard to get right. Code like:

```
MOVE 3 TO A  
DISPLAY A
```

might not yield the output 3, if another thread has stepped in and modified the contents of A in between those two lines. There are techniques and data structures specifically designed for use in multithreaded programming (e.g. mutex, condition variable, locks, etc.), and entire books on writing multithreaded applications within a given programming language.

You might want to start at <http://users.actcom.co.il/~choo/lupg/tutorials/multi-thread/multi-thread.html> and work from there.

– Oliver