

## Re: help: why dosent this code work?

**Source:** <http://linux.derkeiler.com/Newsgroups/comp.os.linux.misc/2005-02/1724.html>

---

**From:** Peter T. Breuer (*ptb\_at\_lab.it.uc3m.es*)

**Date:** 02/20/05

Date: Sun, 20 Feb 2005 09:42:40 +0100

vesper\_carmen@yahoo.com wrote:

> *This code is supposed to print "hello world" character by character*  
> *with an interval of 40 ms between it. But it goes into some kind of*  
> *infinite loop :(*

I suggest you debug it, then. Get it down to ten or max twenty lines of code and somebody may be inclined to help you. And indenting reasonably may help.

```
char *h = "Hello World\n";
int main ()
{
    printf ("\n In main \n");
    //initialize mutex here?
```

Somewhere at least. Shume mutex shumwhere?

```
    init_timer (40000);
```

Does that launch the thread? Yes it does. And runs it (`pthread_create`). Does it really run? Have you checked with `printf`? As far as I could see it is stuck waiting on a semaphore that I never saw anyone signal.

```
    while (*h++) {
```

Well, isn't that all wrong for a start? You will count through the characters in `h`. Shrug.

```
        while (!stopped) ;
```

And that appears to be an infinite busy loop. I suppose you are hoping that a thread will change it? But when? What does the scheduler think is the time to let the other thread in? Is this where you are stuck?

```
        stopped = 1;
```

comp.os.linux.misc: Re: help: why doesnt this code work?

But then why do you set stopped to 1, when it is already that, since it just exited the busy loop?

```
init_timer (40000);
```

Well, you don't seem to trust do while.

```
    }  
  
    printf ("\n %d \n", &delaycounter);  
    //sleep();  
  
}
```

Go and debug your code – first find where you are stuck. Then find WHY. I would imagine that you are stuck in the busy while(!stopped) loop.

If you don't feel like debugging it, explain it. There are at least two threads, and one of those threads seems to be prepared to create more threads, so you cannot expect people to know what you have in mind for the code to do.

If you don't feel like explaining it, reduce it to smething that need no explanation.

I would guess that you are all mixed up about who gets signals. But then I'm not going to do your debugging for you!

Peter