

Re: 8ms Timer for serial port access

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.development.apps/2003-10/0588.html>

From: Jens Schumacher (jens.schumache_at_gmx.net)

Date: 10/29/03

Date: Tue, 28 Oct 2003 19:41:48 -0500

Hello,

- > *You could loop on a blocking read. You could also use 'poll' to wait up*
- > *to a certain amount of time. The key is that each time you call 'read', you*
- > **must* read all the data that's available before sleeping.*

When I specify the blocking in the open function, read returns as late as there are 24 bytes available, is this right?

- > *Do not try to teach the kernel that 24 bytes means something special to*
- > *you. It doesn't care. If you get 8 bytes, wonderful. You'll get 16 more*
- > *later. If you get 36 bytes, fine, process 24 of them now and save 12 for the*
- > *next time you get some data.*

I meet some Problems when I tried this. I tried to read the data and checked how many bytes were read. When I had less than 24 I read again and asked for the missing bytes. But somehow some bits were corrupt and gave me wrong numbers.

This is why I checked the bytes in the buffer first and when there are 24 bytes I read them.

- > *No, you don't. You read the data when it's ready, not at some magic*
- > *time. You can use 'poll' to tell when there's data ready or you can use a*
- > *blocking read. Here's what you're doing:*
- >
- > *1) Wait X time*
- > *2) Read Y bytes*
- > *3) Repeat.*
- >
- > *Here's what you should do:*
- >
- > *1) Wait X time or block in 'poll' until data is available.*
- > *2) Read however many bytes there are.*
- > *3) Repeat.*

1) Sounds reasonable to me...but how to block it. Sorry maybe a stupid question.

2&3) I meet a problem described above.

>> *I just need the data from the serial when a whole frame arrives every 8ms*
>> *and need to process the data directly. I never thought this would be a*
>> *problem on a modern system.*
>
> *It's not a problem, you'd just trying to teach the kernel that 8*
> *milliseconds and 24 bytes are special. It doesn't care and so won't do what*
> *you want it to do. Just read however much data it has whenever it has it and*
> *you'll be fine.*
>
> *You can even do this:*
>
> *1) Wait 8 milliseconds (though it will probably really be ten)*
> *2) Read as many bytes as are ready without blocking.*
> *3) Process as many complete frames as you now have saving any leftover*
> *bytes for the next pass.*

Then I will have another delay before I can process the data. I try to avoid any delay like this.

>
> *DS*
>
>