

Aug 29 13:09 1984 Page 1

From iv Wed Aug 29 11:47:49 1984
To: uhclcm
Sent: Wed Aug 29 1984, 11:47:44 CDT
Subject: Serial Status [Forwarded]

==== Start Forwarded Message =====
>From uhclcm Wed Aug 29 11:32:35 1984
To: ron iv bc foggy/bc gordon tbul
Sent: Wed Aug 29 1984, 11:32:32 CDT
Subject: Serial Status

Ah, there's bad news tonight. I have come across several things that are causing serial to be the problem it is.

Currently we have a scheme that regulates how often the system is going to interrupt the 68k. The task runs once a second and asks how many characters were received in total during the last second. If less than 16 arrived, it will interrupt the 68k on the arrival of each. If less than 50 arrived, it will interrupt the 68k when 16 have been saved up. If more than 50, it waits until 32 are saved up. All of this is covered by a trap which will flush the queue once a second. This was meant to take care of the case where the alarm level was high and the input rate suddenly dropped to one or two chars per second.

It turns out that this scheme has a built-in flaw. It is a lump sum of all serial traffic. Subsequently, if UICP or some high speed serial traffic is running, the alarm level would jump to 16 or 50 and the guy in the interactive editor would see bursts of data.

This is really bad in cv/cu, where each character you type counts twice. Once as it goes in on the keyboard and once when it comes back to be displayed on your terminal. You only have to type 8 characters in a second to get above the fifo alarm level all by yourself. Pressing <Tab>, which is echoed as several characters, promptly puts the fifo alarm up high.

The solution to this would be to have a separate queue for each input channel. This would give each channel its own alarm level (if needed) and the data would be already broken down by device, which should ease the sorting task on the 68k side.

Another problem has shown itself and explains why cv/cu loses so many characters. When cu is running, you will rarely see more than 4 characters in the fifo, even though the alarm level has been set up to a 50 character level. This is because the 68k drains the fifo even when the r20 did not request that action. The 68k drains the fifo on any reference to the serial hardware. In the case of cu, it receives a character, and then echoes it to the console. When that character is transmitted, the receive queue is emptied again. This defeats the purpose of the alarm mechanism at a time when it is most needed. I am considering "tricking" the 68k into thinking there are no characters in the fifo until the r20 reaches the alarm level or the timeout point. However, this will not solve the problem of fifo alarm-level calculation.

It has also been found that when the console scrolls, we lose one character and sometimes two from the SIOs. This is interesting since the sios can store three characters on board, giving us just over 3msac to service

Aug 29 13:09 1984 Page 2

the pending interrupts. The period of time that the screen has interrupts disabled is for about 25 bytes of code, which should be executable in less than 100 usec. Tom has suggested that some other routine is entering screen with interrupts disabled, so that is another area I will check.

Any suggestions or comments on my multi-buffer proposal are welcome. I suspect that sort of change will be so big that it will get killed by management. Oh well.

Frank

===== End Forwarded Message =====

Cheers,

IV (aka John Elliott IV)
Tandy Systems Software; Fort Worth, TX
... (convex!ctvax,microsoft)!trsvax!iv

From: ron Wed Aug 29 12:58:01 1984
To: ubriem
Subject: Serial Status
cc: thul iv bc gordon
Date: Wed Aug 29 12:57:56 1984

- 1) Don't propose it, JUST DO IT. By the time management gets around to finding out about it, it will be too late to cancel it and we will have improved the system a great deal. This will help the sales of the 16c (we can say).
- 2) If a multi-buffer scheme will help or increase the throughput of the serial drivers, do it. If you need help, on either the x80 or 68k side, I'll be happy to throw my two cents in on off hours (which I sometimes have plenty of). I, for one, would like to see us come out with something that is nice and actually works! I can feel very sorry for the customers that decides to get a 16/12, finds that he can also add a few more channels for his secretaries to use, and then discovers that he can only run 300 baud. I realize with our present hardware we are somewhat limited, but it would be nice if our software were capable of 19.2 even though the hardware isn't.

I guess what I'm trying to say is don't waste time talking and discussing a change with management. Most every time we've suggested some change to management, they've replied that it would take too long or involved too much work, only to find out later that we did have time and/or it wouldn't have taken much work. Most of this was due to managements lack of understanding how much a programmer can do in a given time, or the suggestion "sounded" overly complicated to them. If you see a way to make serial run faster, send us mail saying "hey, I found this neat way to speed up serial", and then go do it.

Aug 29 13:29 1984 Page 4

From iv Wed Aug 29 13:04:09 1984
To: ron uhcleam
Sent: Wed Aug 29 1984, 13:04:02 EDT
Subject: Re: Serial Status
Cc: thui bc foggy@bc.gordon iv

I agree with ron. Let's do it. If there's anything you'll need in the way of 68000 changes, let me know. If I have to, I'll do them at home!

We'll get XENIX faster if it kills us...

IV (aka John Elliott IV)