

Re: Video Mode 13h in windows XP ... impossible?

Source: <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2004-02/0970.html>

From: Beth (*BethStone21_at_hotmail.NOSPICEDHAM.com*)

Date: 02/20/04

Date: Fri, 20 Feb 2004 15:17:59 -0000

The Half A Wannabee wrote:

> *wolfgang kern wrote:*
>> *My (fastest) way is using direct hw-access,*
>> *but this goes a bit deeper and needs hardware specific routines,*
>> *so I can only support a few known cards that way,*
>> *for all 'unknown' stuff I have to use VBE/VESA method.*
>
> *Nice post. I was thinking. Would it be possible to install a driver*
in
> *windows to record anything send to the Nvidia driver and from the*
Nvidia
> *driver to the hardware ? I do not intend to do this, but just for*
> *curriosity. My only question is, would it be possible ? Beth, Wolf.*
Anyone?

Never attempted any such thing so this is just "guessing"...but I can't think of any particular reason why you couldn't "install" your own display device driver, which simply "propogates" things to the real nVidia driver or back from that driver to Windows (and takes a "sneaky peek" at it, recording the stuff :)...there'd be an inherent "slow down" because of this extra "layer" but I can't directly think of any reason why it couldn't work...basically, write a display device driver which simply acts as a "middle man" between Windows and the nVidia driver, recording things but otherwise "propogating" them back and forth so all the graphics still happens as expected...the only thing that might be a problem is anything that requires precise timing but there shouldn't actually be anything that demands such precise timing generally because, like, all cards respond at different speeds and have different driver code (some better or worse than others ;)...

Of course, the big issue here, though, is WHY do this? After all, to write your "display driver" code, you'd get your hands on Microsoft's "display driver" specifications and follow them, right? In which case, why do you need to spy on the nVidia driver? The messages Windows sends it and that it sends back to Windows are, of course, exactly those specified for any Windows display driver...this part of the "interface" in the OS is, in fact, abstract and completely standardised...it has to be because Windows itself uses these

display drivers to draw the Windows desktop and implement DirectX and so on and so forth...this _ISN'T_ the place where the "big mystery" is kept...the bit nVidia – the selfish sods that they are – keep from developers is the interface between the display device driver(s) and the _hardware_...this is the bit which tells us how the actual hardware cards work...and that, yup, nVidia keep from anyone knowing by not publishing specifications, which I'm guessing is your "inspiration" in asking this "can we spy on the driver as it's running?" question...alas, you could do this but that's not actually the bit that's kept hidden... _Microsoft_ define the Windows → display driver interface and the specifications for that are available from Microsoft when you look in the right places (the DDK is a nice place to start...but there are sometimes some additional things for display drivers to do with supporting DirectX properly that you should also look out for any "DirectX DDK" too, which contains "additional" about some of the DirectX specific stuff to supplement the basic stuff in the DDK...I know that DirectX 7 had such an "additional" addressing how to make the accelerator stuff work much better...you just have to rummage around...Microsoft are totally happy to release this stuff because, as I say, their monopoly actually kind of depends on the fact that hardware manufacturers write all the device drivers – if they didn't, then Windows would start to lose its unrivalled hardware support – so MS are actually happy to thrust this into people's faces...they kind of _want_ you (desperately, even ;) to use their DDK and write drivers for all your new hardware so that they can sell Windows on: "Supports all hardware!"...oh, they'll _never_ willingly release the _source code_ of Windows itself, indeed...but all the "interfaces" to actually _use_ Win32, they can't give enough away...creating "MSDN", the "Knowledge base", free downloads of Win32 header files, SDKs, DDKs, etc....on this front, in fact, Microsoft actually _so_ want people to have this information and use it that you'll note that even something "open source" like Linux – even with built-in "man" pages – is actually trying to "play catch up" on giving out the documentation because MS are giving away much more information and support on this stuff...they don't hide this "interfacing" stuff at all...quite the contrary, Microsoft are kind of trying to thrust it all in people's faces – because if you use Win32, then you're _supporting_ Windows (the point Rene doesn't quite get ;) so they are overjoyed at you writing all your software to their proprietary standards and giving _them_ your trade (which includes all _your users_ of your program too, of course ;) rather than Linux or OS/2 or something instead – and, in fact, amazingly, no-one comes close to Microsoft...this is another element of their "monopoly"...they _keep_ the monopoly with the _proprietary standards_ on the hardware device drivers...but I was around when Win3.x was becoming "the" OS and Windows starting pushing all the "alternatives" away as it _gained_ monopoly in the first place...and, this is why people, in fact, go on and on at Linux developers to "make the bloody thing more user-friendly!!!"...they won the monopoly in the first place by simply making Windows the brain-dead simple monster that it is...full of "help files" this and "here, have free SDKs and header files to

alt.lang.asm: Re: Video Mode 13h in windows XP ... impossible?

download" that...it's where Linux does, unfortunately, fall down:
There's no "service" ethic in the Linux community...where MS set-up 24
hour helplines, Linux people pour ridicule on a poor newbie user who
doesn't understand what "fsck" means or how to use it...indeed, IF
Linux wants to "win" the desktops (although, of course, if it doesn't
then that's perfectly acceptable too...it can, of course, choose to
remain a "niche" thing for developers and "power users"), then one
word and attitude needs to develop: a "service" ethic...as a
commercial company, Microsoft kind of get this stuff naturally into
the bargain...the old "the customer is always right...OR, basically,
you make no money!"...Linux gets the "no, the OS should be good_!!"
but doesn't have the "play to the audience" attitude needed to get it
accepted to a wider – yes, mostly Clueless_ – desktop market :)...

Anyway, you could probably do that...but, if you did, you'd wonder why
you bothered ;)

Beth :)