

RE: AMD Dual Core CPU's

Source: <http://linux.derkeiler.com/Mailing-Lists/Ubuntu/2006-01/msg03552.html>

- *From:* "Billy Verreyne \ (JW\)" <VerreyB@xxxxxxxxxxxxxx>
 - *Date:* Tue, 31 Jan 2006 13:02:14 +0200
-

Title: RE: AMD Dual Core CPU's

Sasha Tsykin wrote:

> maybe Microsoft games (as directx is a Microsoft library). In general the
> graphics card is more limiting.

In general? Just how general is Microsoft than owns heavens know what percentage of the desktop market!?! Or are you saying that the chief head on graphic development at Microsoft, a company whose R&D budget is the GNP of a small nation, does not know what he is talking about when it comes to CPU and GPU bottlenecks. And that you know better?

How about backing that statement up with some facts instead of opinion? Show me the money. Or the URLs. From reputable technical authors that state that their development is GPU bound as they are pushing too much from the CPU than what the GPU can handle.

>> Bs! Please show me a –single– game on the Windows market today that
>> does not use DirectX.
>>
> As for a single game that does not use directx, how about doom3. Or quake4.

They –all– use DirectX. Some may use miniGL, or OpenGL as the graphics rendering API and not Direct3D. But they –all– use DirectSound, DirectInput, and even DirectPlay.

DirectX is a suite of software APIs that allows a program near native access to hardware as possible within a multi–processing environment where all access has to go via the kernel (i.e. we cannot talk directly to h/w anymore like we did with DOS games).

Now you're telling me that these games you mention will use the standard Win32 API to read keyboard/mouse/joystick (sharing a message queue with the rest of the applications) and the standard (somewhat pathetic) Win32 sound API? No support for 3D sound. Limited support for the number of sound channels? Poor performance. Etc. Etc.

Hell, I cannot even hook into the DirectInput keyboard queue of a game like America's Army using a debug hook – which I can easily in applications that use the standard Windows keyboard queue.

And then you wonder what I called your statement bs?

> Almost any game with a linux port does not use directx. They use opengl.

Why shift the argument to the type of API and ignore the fundamental issues? Okay, I'll bite. What about Plib? Does Plib attempt to do everything in the main calling thread using sync/blocking calls?

> That's why the developers made linux ports. Because its easy for them
> to port it because linux also supports open–gl.

It is everything except easy to port games from Windows to Linux – even when written in OpenGL. Besides, this has absolutely no bearing on discussion.

> If an application
> is single threaded 300 cups won't make a difference, so stop telling me

RE: AMD Dual Core CPU's

> about how important many cpus are.

Then you say you have actually –read– what I said? Sure does seem like it...

> If you want to prove they're necessary, all you have to do is demonstrate
> that the majority of games are written as multi-threaded. As you can't, because
> they're not, the point is irrelevant.

So did not bother to read what I've said.. <sigh>

Where did I state that the majority of games are multithreaded!? I did not. I stated two things.
1) more and more games are being written as multi threaded and gave two real world examples
2) stated that even a single threaded game on Windows will make use of DirectX and that DirectX itself will run as threaded processes (thus providing an async interface for things like sound and network comms)

Thus there could be benefits in using a dual core CPU as a gaming platform over a single CPU.

Proof?

Simple. Run Windows Task Manager and look at the thread count for DirectX processes when you run one of your games. Run X-Plane (on Linux, OS/X or Windows) and have a look at its thread handle count.

>> You are confusing ASIC's with multi-threading (a software model) with
>> the dual CPU capabilities.
>
> I am not. Unless software supports multi-threading, hardware multi-threading
> will make no difference because it cannot be utilized,

H/w multithreading? ASIC are Application Specific Integrated Circuits. Which means the CPU off-loads stuff to the ASICs to handle. Simplest of examples – an ASIC UART will handle a bunch of stuff (like retries) before telling the CPU that an operation has failed. Thus alleviating the CPU from having to handle that. And this does not matter whether or not the software using the UART ASIC is multi-threaded or not!

> Neither. I just know how to argue a point

You could have fooled me.. Unless you can backup your arguments with some solid technical evidence, this argument of yours is not going anywhere.

> Incidentally, please stop insulting me.

I'm an amenable old fart Sasha... but talking technical bs is the –one– thing that makes me reach for my lead pipe. Don't think I will let technical bs just slide.

--
Billy

~~~~~  
This e-mail and its contents are subject to the Telkom SA Limited  
e-mail legal notice available at  
<http://www.telkom.co.za/TelkomEMailLegalNotice.PDF>  
~~~~~

--
ubuntu-users mailing list
ubuntu-users@xxxxxxxxxxxxxxxxxxxxx
<https://lists.ubuntu.com/mailman/listinfo/ubuntu-users>