

Re: Trivia Question

Source: <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2006-01/msg00351.html>

- *From:* "\\o//annabee" <faq@xxxxxxxxxxxxxx>
 - *Date:* Thu, 05 Jan 2006 01:01:08 GMT
-

På 4 Jan 2006 15:18:23 -0800, skrev randyhyde@xxxxxxxxxxxxxx
<randyhyde@xxxxxxxxxxxxxx>:

\\o//annabee wrote:

På 4 Jan 2006 08:14:02 -0800, skrev randyhyde@xxxxxxxxxxxxxx
<randyhyde@xxxxxxxxxxxxxx>:

```
>
> \\o//annabee wrote:
>
>> Who need that? Someone dealing with writing a compiler!
> Or a debugger. Or maybe someone really wants to *change* one of the
> register values on return (and, therefore, needs to poke a value into
> the stack frame where the value for that register is sitting).
>
> Make all the excuses for your ignorance that you like, but, yes,
> assembly programmers *should* know about this stuff.
```

```
I disagree. This is the kind of stuff that is
trivial to enquiry once you
written
about as much code as I have. You know, the stack
may seem strange at
first, but the
fact that it grows to lesser adresses, is not
really THAT hard to figure
out.
```

???

First you say that people don't need to know this. Then you seem to be confused about what the question actually was as what you've posted has nothing to do with the original question.

You misunderstand. By knowing how the stack works, and using RosAsm I can using the incredible RosAsm runtime debugger, inspect the stack, and

Re: Trivia Question

confirm the order of pushed registers.

Its just that I never needed to know.

And that lack of knowledge means that you cannot take advantage of this information when it would be useful. This is what I meant by "ignorance".

But as usual you are wrong. First we have a problem, then we have a solution.

If , for some reason this solution had to involve knowing the order of the pushed registers, then we would have done the steps to find out.

There a million things I never needed to know, and even more that I will never need.

No doubt. But the more you do know, the better the position you'll be in to recognize when that information is *applicable* to your current problem, rather than hacking out a different, and less elegant, solution.

Need to know. As long as I never needed to know, and as long as an elegant solution offer just its elegance, which in the actual cases seen, is usually the complete oposite of fact, while they tend to be rather obscuring solutions, and called elegant just because they obscure the thinking.... I have no need for dead knowledge like that.

If you want to see elegance, take a look at some of the RosAsm sourcecode.

But this doesn't make it impossible to wrote FULL PEs in assembly, quite the oposite.

This has nothing to do with writing PEs. Indeed, the whole concept is independent of the executable file format in which the PUSHA/PUSHAD instruction appears. You are simply over your head on this one and you're making stuff up so that you don't look quite so stupid.

:)) You life story master P. The above sentance is YOUR LIFE STORY.
- Not to say the Your Lie Story.

Re: Trivia Question

Allow me
to give you a hint: when you don't have a clue what you're talking
about, don't post (unless, of course, it's a question). Trying to cover
up your ignorance with additional nonsensical posts just makes you look
worse.

Ah. Getting tired Master PDF? I didnt think you had it in you.
I am only adressing your many strange claims, as best as I can.
This is what this whole "n"g story is about.

You are actually probably scaring beginners by
providing too much info at
once.

???
Where would this be?

Pages and pages of nonsensial, teoretical infos at Webster.
1000 pages? More? Dear god, there used to be 2 chapters on just how to
install HLA!

RosAsm = download, unzip, start. Takes at most 15 seconds.

I guess you've completely missed the whole point of HLA? Leveraging
people's **existing** HLL knowledge to learn assembly language
programming? This was the big improvement over the 16-bit edition,
which **did** blast people with "too much information at once."

The whole point of HLA is imo, to become the greates joke on asm of all
times.

> After all, the
> point of assembly language is **not** to write
HLL-like code in assembly,

The point of assembly is to program the machine.

Re: Trivia Question

Funny, I thought that this was the point of *all* programming languages (HLLs included).

It is. You are its seem correct. One of your occational good shots. :)

Its the native language
(-es)
of the machine.

And what does this have to do with anything?

Well. Itsa bit like me, I suck at english and thus dont get all the nuiances with me all the time. If I spoke your mother toungue, it would be more easy for me to explain myself.
For many complex topics I dont know the words at all, and just have to leave those subjects alone as its very important to have a presise language for communcating certain ideas.

In particular, what does this have to do with the order by which registers are pushed on the stack for PUSHA/PUSHAD? You're going off the deep end here.

We been though this. This is a trivial subject : even Hutch--,.-- could figure it out.

To read more into it than that is not fair.

Sure it is. If people are less productive in assembly than in HLLs, it's *fair* to compare those languages on this basis.

But thats not a fact at all. Its just the oposite.

Assembly usually
comes up short.

Huh? A two year beginner can really near damn blow you away.

Re: Trivia Question

But again, please explain what this has to do with PUSHA/PUSHAD. Sounds like you're trying to change the subject to avoid attention being drawn to your ignorance about these assembly language statements.

:)))))))))) Comming from you, thats not only funny, its sick!

```
> but to take advantage of what the machine has to offer. Now, knowing
> the exact sequence of the registers on the stack isn't something that
> every programmer ought to have memorized (it's easy enough to look up
> when you need it, or you can do something like #include( "x86.hhf" );
> and get symbolic equates for these offsets), but a decent assembly
> programmer *should* understand why knowing this information may be
> useful.
```

Information are rarely useful until needed.

And that information remains useless if you don't know about it. That's why its a good idea to know things like how PUSHA/PUSHAD work.

Redundant argumentation. Allready attended.

No matter how many manuals,
how many infos you memorized, this will only be a small help at writing
programs.

The true hacker's motto. What can I say? You do realize that you're
preaching that ignorance is better than knowledge, right?

No, I preach that its better to know HOW to apply and seek knowledge than
to stuff everything without sensurship into the old hat.

Its not like your memory is the one that moves and inspires you to do
programming,

Been taking too many drugs, lately?

Re: Trivia Question

Well, no, but feel free to explain how your memory inspires you. Oh I forgot, You wrote an assembler back in what was it 1986? , that could do everything the RosAsm can do. Sorry. Which I had such an assembler when starting with Delphi.

It is your imagination. Then the language becomes a tool to make that thought a running application. After working with the needed instructions some times, repeatedly, they tend to stick around in your memory.

Ah, I see, you're following the "Betov philosophy" of only needed to use 20 or so instructions in an application.

Its a bit more, maybe 50. I havent counted.

You're **not** thinking in assembly when you do this.

Of course not. I am thinking in terms of how I can make my apps do what I want.

And you're **not** taking advantage of the machine when you do this.

Nope. My head and thoughts are 100% true blue human.

And if you don't study the instruction set and learn what the majority of the instructions do (and their relative timings), how do you know when to use, say, LODSB and STOSB versus a MOV and INC/DEC instruction?

I watch other apps, step through the debugger, read about just those instructions that I can see right in front of me, running, and try to make use of them in my own apps.

And when you ever discover something that you cannot formulate, you may take a look at the instruction set for clues. Then, by and by, you build the needed base to

Re: Trivia Question

do what you want.

Well, from the code of your's that I've seen, what normally happens is that you hack out a solution with a sequence of instructions rather than taking the time to learn about the one instruction that does it all. Don't feel bad, this is human nature (and your mentor Rene is a classic example of someone who follows this philosophy).

But seeing as RosAsm is the better assembler, and that HLA is just a joke
:> I may prefer Renè philosphy.

> Hopefully, in a few more years, you'll discover why it just
> might be important to know the order of the registers pushed on the
> stack by pusha/pushad.

If the situation where important, it will be
important. Sure. But for more
then 2megas
of written sourcecode, between all my apps, this
was never needed.

What you mean to say is that you've never recognized when this would
have been useful.

No, it means I found solutions that wore good enough and that didnt
require it.

So thats why, it is useless info to throw at a beginner, unless he asks
for it.

Wrong. Most beginners don't even know the question to ask.

To some extent this is correct. But to read a whole book the size of
PDU, is certainly not needed at first.

The fact
that Chuck's little quiz pointed out your ignorance on this subject
makes that abundantly clear in your case.

My ignorance has been debated before. I have no problems saying it like
it is
I know 3 eflags, about maybe 30-50 instructions and thats basically it.
But since I use RosAsm, I can pop out applications easy as that.

Re: Trivia Question

Its just 4 clicks to create an application with RosAsm.
To expand it from there is not incrediby difficult.

Beginners need to herded in
the right direction.

Away from PDU.

Then their education is far more efficient than if
they simply "wander around in the desert for 40 years" as you seem to
be doing.

10 years blindfolded, using Delphi, 2 years with RosAsm.

You do realize, don't you, that most people become *expert*
level assembly programmers when doing it constant for two years.

How dumb can you be to think I did it constantly for two years ?

The fact that you're still calling yourself a "beginner" after all this
time (and the ignorance displayed in many of your posts confirm your
opinion of yourself) suggests that the educational approach you're
using could stand for some improvement.

Looking at HLA, its safe to assume you competance to be equal to &NULL
HLA nomatter what you claim, is a very very talentless application.

Then again, if you programmed
in Delphi for ten years and still couldn't figure out how to do a
linked list, I suspect than *any* educational approach isn't going to
work well with you.

:)) I just love it when you repeat that Prototype9+ :)

Cheers,
Randy Hyde

Re: Trivia Question