

Erik Stevenson

From: Brad Chase
To: bradsi; davidcol
Subject: RE: win16 apps in a vm
Date: Thursday, January 06, 1994 11:04AM

which of course means that daytona will push this as cool and the fact that os/2 has it and we don't will get brought up even more

From: Brad Silverberg
To: Brad Chase; Brad Silverberg; David Cole
Subject: RE: win16 apps in a vm
Date: Tuesday, January 04, 1994 7:08PM

my sense is that #1 is more what users want: they want protection and preemption. it's also what daytona is doing. did we ever find out what they are really doing and if it suffers the same problems as os/2?

From: davidcol
To: bradc; bradsi
Subject: win16 apps in a vm
Date: Tuesday, January 04, 1994 6:21PM

I've been discretely talking with some people about this since it's one of the things OS/2 can do that we can't with current POR. There are 2 basic routes we can go, the choice depends on why we think people want Win16 apps

in a vm.

1. just copy IBM and run win16 apps in a vm (user's option). This gives preemption and robustness, but compatibility suffers a little. Copy/paste,

dde, ole, multimedia, and display drivers cause problems. Note that only win31 can run in a VM and thus win 3.1 apps, copies of Chicago can't be run

in a VM, we burned our bridges to doing that long ago.

2. preemptively run win16 apps in the Chicago system vm. Current thinking is that we can have a thunk layer (blocking layer actually) for win16 apps.

This is sort of like the parameter validation layer in that it knows when to preempt the app and when not to depending on what DLLs it uses, etc this

works a lot like the thunking layer we have for win32 apps today. The benefit is preemption. Robustness would come from whatever improvements we

made to the system itself.

Either of these could be a premium only thing. I personally think #2 can be

positioned better than #1. #1 is copying IBM, you have to deal with being the clone now. #2 is solving the problem the way the system vendor is suppose to, by fixing the problem. However, if corps want robustness and preemption and are ok with compatibility to suffering, doing it in a VM is better. If the focus is on preemption, then #2 is probably better.

Bradc, I need to drill down on this a little with your guys. Is rogersw the

right guy? I will continue to flush dev work out a bit. Feeling right now

from dev types is that #1 is more work than #2, but I'm not saying for sure
because they've always wanted to do #2 instead of #1, including ralph.