will get upset. There is a strong likelihood however that we could
get rid of debug exec2bin and link. That is 60 pages and 15% of the user's
ref-guide

One thing we can get rid of though is the user's guide which is geared at
the user which has never used MS DOS before. That’s 62 pages
and about 15% more of the manual right there.

Finally, as far as GW BASIC is concerned, we need to incorporate a quick
reference of whether we like it or not. I see we will get millions of calls
for the manual.

Susan Johnson

From: mlaw
To: mlawcole
Cc: srik nar; tomle
Subject: DOS 4.0 Retail Upgrade Tree
Date: Thursday, September 22, 1988 12:17PM

Srikam and I created a project for the DOS 4.0 retail upgrade.
We put it on \TROJAN\DOSZ. The project is called 40RETAIL.
-Mike

Susan Johnson

From: blig
To: pascalman; russw; tomle
Cc: phihe
Subject: Dr dos
Date: Thursday, September 22, 1988 12:41PM

You never sent me a response on the question of what things an app
would do that would make it run with MSDOS and not run with DR-DOS.
Is there any version check or api that they fail to have? Is there
feature they have that might get in our way? I am not looking for something
they can't get around. I am looking for something that their current binary
fails on.

This is a fairly urgent question for me and I have received nothing.

Susan Johnson

From: tomle
To: pascalman; russw
Cc: phihe; tomle
Subject: Dr dos
Date: Thursday, September 22, 1988 1:28PM
I am assuming your handling this one based on the info you got from AaronF pascal. Please correct me if I am wrong.

Tom

P.S. I am referring to Bill's question

Suan Johnson

From: robert
To: anthony; davo; me; vch
Cc: ms; steem; neye
Subject: NP faults in device drivers
Date: Thursday, September 22, 1998 1:25PM

After discussing this with Ross Cook, I decided to do some research.
On page 8-91 of the IBM OS/2 Technical Reference, Vol 1, (VerityAccess)
I see the following paragraph:

Once the process has been verified as having the needed access to a specific address location, the device driver doesn't need to request access verification each time it yields the CPU during task-time processing of this process's request. If the process makes a new request, then the device driver must request access verification.

If I were reading this, I would not assume that I had to do any kind of locking of the segment. The next paragraph says:

Note also that, prior to requesting the Lock on user process-supplied addresses, the device driver must verify the user process's access to the memory with the VerityAccess Dev-Hlp call. The device driver must not yield the CPU between the VerityAccess and the Lock, *otherwise the user process could shrink the segment before it has been locked* [emphasis mine]. Once the user access has been verified, the device driver may convert the virtual address to a physical address and lock the memory. The access verification is valid for the duration of the lock.

So the documentation implies that the only reason you'd want to lock a segment would be to prevent a shrink (and presumably a free). There is no mention anywhere in the VerityAccess man page that says anything about having to protect the segment from swapping or discarding.

I suggest someone check our documentation on this.

It may be wrong, but it's documented, albeit badly. A question: can we detect if a device driver was built with 1.0 look, or something similar? We may be able to do something to let 1.0 D0s run unhindered. I asked why this was never seen before, and Ross told me that the EE guys tend to run machines with lots of memory, so they've just never had this happen before.

Based on the above, I'm leaning towards either taking out the message or putting it under strict control. Comments?