I doubt that's the problem. When I dial MCI on the same machine, I never have a problem. Only dialing into 867-1681. I get a connect message, but then after hitting enter a number of times, nothing happens. So I hang up, redial, and repeat 5-10 times until I get the login message.

No problems at all calling other services.

| >From corpop Mon Nov 26 08:18:44 1990
| To: bradsi
| Subject: linelock
| Cc: tomh
| Date: Mon Nov 26 08:05:33 1990
|
|
| It's possible you are running into a hung modem. Please call us
| at 936-3367 when it happens so we can have a closer look.
|
| Thanks, Corpop.
|
| ----------------------------------
|
| >From joem Sun Nov 25 09:33:02 1990
| To: AllenY GregP corpop
| Subject: linelock
| Date: Sun Nov 25 09:27:59 1990
|
| From bradsi Sat Nov 24 18:50:33 1990
| To: joem
| Subject: linelock
| Date: Sat Nov 24 18:48:07 1990
|
| Why do I have so much trouble getting through linelock
| to get a login message? I typically have to dial 867-1681
| 5-10 times before I get login.
|
|

101
From bens Mon Nov 26 12:21:28 1990
To: bradsi
Cc: cameronm davidw paulma philba richab
Subject: to share or not to share, that is the question
Date: Mon Nov 26 12:19:34 1990

Issue:
- Should MS Apps (and other ISVs) have access to Windows source code?

Complications:
1) If MS Apps have access to the Windows sources, then all ISVs should have access, else we are subject to restraint-of-trade complaints (to say nothing of the morality of the situation).

2) If outside ISVs have access to our sources, then we make it much easier for another company to come along and clone Windows.

I talked to some apps guys on a recent recruiting trip, and they made the following very good point:

Windows is poorly documented

The state machine that is USER.EXE is barely documented in the SDK. Any ISV that wants to write a great Windows app ends up looking at the source code (like our Apps group), unassembling the DLLs, or writing experiment code to divine the actual behavior of the system.

Arguments against giving ISVs access to Windows source code usually boil down to:

An ISV will use some undocumented feature of Windows, or directly access internal data, in such a way that future versions of Windows will be forced to support this bad behavior, restricting MS ability to innovate in Windows.

I claim that letting an ISV looking at the source code is the *best* way to avoid this problem:

1) An ISV that unassembles Windows to figure out its behavior is effectively looking at source code, but without the benefits of source comments. This approach is more work for the ISV, and gives MS no opportunity to guide the ISV. With source code, there are generally comments discussing rationale for the behavior of the system.

2) An ISV that writes test apps to divine Windows behavior is really on thin ice. Either the ISV spends a great deal of effort writing test code to be certain Windows is fully understood, or the ISV may end up making assumptions which are not correct. Since the former approach is a great deal of work, and the ISV is never sure when to stop ("do I really understand how this works now?"), most ISVs will end up in the latter situation. These are the most dangerous apps, since they are most dependant upon the exact behavior of a specific release of Windows.

The key problem is that our documentation does not provide sufficient depth of coverage. The key question is:

How much would it cost to provide sufficient documentation, and is that any different, really, from providing source code?

The key difficulty in writing really great documentation is anticipating all the questions an ISV might have about the behavior of the system. Presumably, there is a level of documentation which is great enough that an ISV would have to perform only a small amount of experimentation.

Solutions:

A. Status Quo
+ Simple
- MS Apps have (unfair) advantage over other ISVs.

B. Make Windows sources available for a fee, with a restrictive licensing agreement (only available to N trusted employees, must be kept on a secure server, no derivative of these sources may be shipped, etc.)
+ All ISVs are equal
- Cloning risk

C. Disallow all ISVs, including MS Apps, from looking at Windows sources (in practice, this would be a very hard thing to do, and certainly runs counter to the spirit of MS).
+ All ISVs are equal
- Difficult to make happen at MS
- Reduces information flow on Windows to ISVs

D. Write great documentation.
+ All ISVs are equal
+ Avoids cloning risk

Conclusion

Given that we want to make Windows programming as attractive as possible, I vote for (B) making the sources available. This gives us a little extra incentive to keep enhancing Windows, so that it does not become a stationary target for cloners, but otherwise benefits the Windows ISV community.

-- bens

102
From ericho Mon Nov 26 12:25:51 1990
To: jamesm
Cc: dos5beta
Subject: Re: Where is DOS 5.077?
Date: Mon Nov 26 12:24:51 1990

We do not have a current release that has been tested well enough to be used in an internal beta. We are in the process of testing a new version and it should be ready later this week. Mail will be going out when it is ready.

If you wish to be a "guinea pig" and install the build we are currently testing for an internal release, email dos5beta and we will add you to our list. If you need disk images though, you will probably have to wait until the internal beta version is ready.

Eric Hough
Dos 5 Beta Support

>From jamesm Mon Nov 26 12:11:32 1990
To: dos5beta
Subject: Where is DOS 5.077?