

# LOTUS

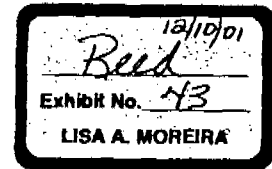
002142

To: Gabriel Baum  
Chuck Digate  
David Gilmour  
Mitch Kapor  
Jim Manzi  
Peter Miller  
Glenn Patsch  
David Reed

From: Ed Belove

Date: July 9, 1985

Subj: Systems software



Attached are my preliminary recommendations as to the strategy and initial tactics Lotus should adopt in the area of systems software.

This analysis and recommendations have concentrated on the IBM arena, since the majority of Lotus sales (both current and projected) are in this area. In addition, Apple's approach with the Macintosh precludes the need (or ability) to add unique systems software.

~~CONFIDENTIAL~~ IBM 0610003018

IBM 7510239530



## LOTUS CONFIDENTIAL

### 1 Summary

Based on the overall corporate strategy, it is in Lotus' interest to control our own destiny in the systems software area as much as possible. Our goal should be complete independence from system software provided by non-hardware vendors. In particular, it is important to distance ourselves from Microsoft as quickly as possible. This approach will enable Lotus to establish a high level of control over the future of its software, particularly allowing us to provide a systems base tailored to our own applications.

We should try to achieve this goal with the next major generation of products. This is based on the assumption that IBM will introduce a multitasking, extended memory operating system which is upwardly compatible with DOS 3.1 within twelve to fifteen months.

Achievement of this goal involves the following:

1. Use of Spotlight technology under DOS 3 to provide the necessary multitasking and data interchange for SLUGGO and Europa. This includes expansion of the background processing capability and addition of support for the data bus technology.
2. In parallel with this, we develop our own windowing environment, in conjunction with Metaphor or another experienced developer, if possible. We base it on the coming IBM system, with DOS 4.0 as the development vehicle and fallback strategy.
3. Determine if VDI 2.0 is the appropriate graphics vehicle and, if not, investigate alternatives.

### 2 Strategy

The strategy for approaching system software is based on the overall corporate strategy, namely providing high value added applications and services to the personal computer user in a variety of business oriented markets.

To do this, we need a stable, controlled base of system software on which to build these applications and services. Until now, this base has been provided by Microsoft through IBM, and has been fairly stable as far as Lotus' application requirements were concerned. However, these requirements have been minimal and constant (primarily simple access to named files).

Now, as we are developing a new generation of applications, their dependence on the facilities offered by the underlying systems software has increased considerably. At the same time, Microsoft is offering increased competition to Lotus in the applications software area and IBM is unwilling to give us any information on their direction in systems software, either with or without Microsoft. In addition, the future direction and cost of both DOS and Windows is unknown to us and completely beyond our control.

For these reasons, our goal should be complete independence from system software provided by non-hardware vendors. In particular, it is important to distance ourselves from Microsoft as quickly as possible, both for the reasons already mentioned as well as to avoid helping to establish MS/Windows and DOS 4.0 as industry standards.

The reason for separating hardware vendors (read - IBM) is that as the line between hardware, BIOS and the operating system becomes blurred, it becomes almost impossible to avoid dependence on the hardware vendor for the base level of software. IBM has proven to be adept at blurring these lines in the mainframe and minicomputer worlds as a way of protecting themselves against plug compatible manufacturers (PCMs) and there is every

reason to believe that they will start to do this in the PC world as well. In fact, the first steps in this direction have been taken with PC/NET, where a large amount of what is traditionally done by software has been included in the NETBIOS ROM.

This approach of integrating system software and hardware is exactly what Apple has done with the Macintosh and its integrated toolbox. It is very difficult to offer a competing operating system or to duplicate the toolbox on another vendor's hardware.

It is important to differentiate between products released directly by Microsoft and those developed by Microsoft and released by IBM. Depending on directly released products leaves Lotus with a high degree of exposure and helps establish Microsoft's primacy with both users and dealers. Supporting IBM products developed by Microsoft may become necessary for the reason outlined above. At the very least, for all the problems of dealing with IBM, they do set the defacto standards and supporting their offerings reinforces an image as a good corporate citizen. This support, however, should not necessarily extend to those layers above the operating system which directly affect our ability to produce competitive applications (e.g. TopView).

This approach to system software will enable Lotus to establish a high level of control over the future of its software, particularly allowing us to provide a systems base tailored to our own applications. In addition, a strong presence in the system software area will keep us in a better competitive position vis a vis IBM and Microsoft. In fact, an aggressive system software stance at this point in time may enable us to influence IBM future direction or possibly even replace or supplement Microsoft as a supplier of system software.

We should try to achieve this goal with our next generation of products. The goal may be difficult to completely achieve, due to resource constraints, public relations problems, lack of control over base hardware design, etc. However, with the exception of the Notes product, we should be able to come very close. Where we can not actually meet the goal, we should strive to create a situation as close as possible. This involves:

- use of generally available software for building blocks (e.g. VDI)
- acceptance of IBM's primacy in the base operating system environment and attempting to maximize our knowledge and control over their future directions
- clean separation of our layers of software to allow rapid response to changes in system software availability or structure

### 3 Product Requirements

There is a range of system software requirements among Lotus' present and planned product line. Following is a summary of my understanding of the current and anticipated requirements.

#### 3.1 1-2-3/Symphony/SOL

- Basic disk file support (DOS 2.0 or later compatibility)
- Interface to Lotus/Intel memory compatible boards

These products interface directly to the hardware for most requirements or have their own driver definitions for common devices. If we develop our own software, we may have to modify these products to support it. In most cases, this would involve new drivers.

**LOTUS**  
002143

## LOTUS CONFIDENTIAL

### 3.2 JAZZ

- MacIntosh toolbox or equivalent

### 3.3 SLUGGO/Europa

- Character oriented window structure (probably built into the product a la Symphony)
- limited multitasking to handle associated accessories (comm., RITA, etc.)
- data bus for interchange among programs, possible networking extensions
- graphics display and/or printer interface
- memory management (optional support for L/I mem spec, protected memory)

### 3.4 Notes/VERTIGO/BASELINE

- sophisticated, graphical window environment
- extensive graphical display/printer interface
- multitasking support
- advanced memory management (protected memory, dynamic linking, shared memory between processes)
- interprocess communication facility

## 4 Tactics

Tactical decisions are driven by our applications strategy, accounting for issues of timing, applications requirements and resource tradeoffs. While we should maintain a focus on the eventual goal, requirements (particularly timing) of our key applications may cause us to diverge from the most direct path. We should recognize where this is happening and try to minimize any duplication of effort or unnecessary digressions.

Until now, the driving forces behind system software investigations have been No Comment and Notes, primarily because these products require more facilities from the system software than any others that we are considering. As we redefine our mainstream products, particularly with the inclusion of SLUGGO, the dependence on the system software facilities will increase.

With the current product strategy, a basic approach to system software can be constructed, based on some assumptions on IBM's future direction. Obviously, this approach may have to be modified as time goes on and we should do everything in our power to determine IBM's plans as soon as possible.

The approach is based on the fact (which I feel is the quite likely) that IBM will introduce a multitasking, extended memory operating system which is upwardly compatible with DOS 3.1 within twelve to fifteen months. This system may come from Microsoft (DOS 4.0 or 5.0) or through an internal IBM development (BIGDOS). Independent of its origins, if it is the next IBM generation, there are strong reasons for Lotus to support it.

The Lotus approach in this scenario would be to use the Integrator under DOS 3 to provide the necessary multitasking and data interchange for SLUGGO and Europa. We have this technology today, and it can be easily adapted to meet the requirements for the next applications generation. Among the extensions that need to be provided are the general purpose data bus and support of extended memory. When the next generation of IBM sup-

ported operating systems comes along, the Integrator will have to be adapted to work with it. This can and should be done in such a way as to minimize the affects on the applications.

In parallel with this, we develop our own graphics and window environments. We develop the environments under DOS 4.0, since it is available today and provides the type of operating system functionality we need. We strive to isolate ourselves as much as possible from the specific implementation by providing a layered, modular interface from the environments to the operating system. Then, when we learn what IBM's plans are, we adapt the environments to their operating system.

If IBM does not produce a next generation operating system (or, their system is delayed so long that it would affect our ability to ship products), we can follow essentially the same scenario as above, but release the environments under DOS 4.0. This leaves us dependent on Microsoft, but to a much more limited and less visible degree than if we did not have our own environment.

Yet another scenario (much less likely than the above) is that IBM integrates DOS and TopView into their own integrated OS/environment. Assuming that they have added graphics to TopView, we would have two choices. One is to adapt our window environment to run under TopView, taking over a window (or the whole screen) and treating it as a separate logical screen which we then subdivide for our application's usage. The other would be to scrap our environment and rewrite our applications to directly run under TopView. The choice between these would depend on the specific direction that IBM takes TopView and would not have to be made for a long time. In fact, we could put together an approach which combines the two. (First run as a single TopView window and, over time, adapt our applications to run directly under TopView.)

Many other scenarios are possible, but the basic approach should give us the ability to react to a wide range of eventualities. Note that I do not propose development of a Lotus operating system, either by ourselves, or in conjunction with anyone else (e.g. Phoenix). My feeling is that IBM will have to make some strong moves in this direction and that competing with IBM and Microsoft in this arena will be too expensive and will divert too much of Lotus' energies in a direction with low return.

## 5 Actions

### 5.1 Integrator

Continue active development, including definition and implementation of a data bus capability. Target for shipment with or before SLUGGO (Q2 '86). We should concentrate first on the kernel and background execution functions, since the kernel and multitasking will be used by SLUGGO and its associated accessories and must be available for development use as early as possible.

### 5.2 Graphics

The first action here is to determine if VDI is the appropriate choice, since there is no other leading contender. IBM seems to be moving in this direction and GSS says that IBM will adopt VDI 2.0 in a number of product families.

002144  
**LOTUS**

## LOTUS CONFIDENTIAL

We have to decide if the performance and space requirements of the VDI are acceptable. We have to investigate possible business relationships with GSS (they are a potential source of development expertise and talent for the graphics/windows area). We should actively search for other alternatives (MetaWindow?).

We must also consider if this is a strategic area that we will at some point wish to control directly. At the least, we want some close relationship with GSS, possibly even including codevelopment and/or access to their source code. We must avoid the pitfall of lessening our dependence on Microsoft by increasing our dependence on other outside vendors who we do not own or control tightly.

### 5.3 Window Environment

The basic approach towards a window environment outlined in my memo dated June 10 is still valid, with the exception that there is no reason for our window environment to run under MS/Windows. We should determine whether Metaphor is a viable vendor and, if so, how to proceed with them. We should also pursue investigation of other alternatives, including an in house effort.

We must investigate the degree to which our environment must be compatible with MS/Windows, given the development that has already taken place in the Notes and No Comment areas.

Under the scenario I outlined above, a shippable window package would not be needed until early '87, when the first No Comment products ship. This assumes that the FCS of Notes uses MS/Windows and, given the current state of that project, there does not seem to be an alternative.

We should also investigate the availability of character based window managers which might be usable for SLUGGO or Europa. If one is available, it could help reduce the development effort for these products.

### 5.4 Network

An area only lightly touched upon earlier, networking is essential to the Notes product and will become increasingly important to other products as time goes on. For the next generation, the networking provided by the system vendors will be sufficient (remote device and file access, etc.) However, No Comment's products will require more advanced capabilities and it would be useful to be able to expand the data bus concept to include intra-machine communication. The next six months should provide some settling out of the network approaches provided by IBM and alternatives, if any.

### 5.5 Languages and Tools

An issue to resolve is whether we should develop our own language and tools. Just as the operating system and environment are the basic building blocks of the applications, the language and tool environment is the basic building block of systems software. If we do not develop our own language capability, we could be in the position of building system software to compete with Microsoft using their own language compilers. This is not necessarily unacceptable, but we should think out the potential problems and investigate alternative suppliers.

6 Open Questions

1. What if IBM licenses MS/Windows and throws support behind it? (As unlikely as this now appears, it is always possible).
2. Where does this scenario leave Notes? The initial release of Notes will be released under MS/Windows, due to timing constraints. Will it be possible in the future to wean it away from MS/Windows? If so, what will the cost be?
3. What is our strategy in the word processing area and what, if any, are its requirements for system software?

002145  
**LOTUS**

- 6 -

~~CONFIDENTIAL~~ IBM 0610003024