The Flawed Fragmentation Critique of Structural Remedies in the Microsoft Case

Robert J. Levinson
Charles River Associates Incorporated

R. Craig Romaine
Charles River Associates Incorporated

Steven C. Salop
Georgetown University Law Center

January 20, 2000
Abstract

Among the proposed remedies in the Microsoft case are structural remedies that would create competition in operating systems for personal computers. One criticism that has been raised against such remedies is that they would lead to “fragmentation” of the Windows standard. According to this critique, the remedy would lead to a number of radically different and incompatible operating systems. As a result, it would impose expensive porting costs on applications developers, leading to higher costs and less product variety for consumers. This paper examines the “fragmentation” hypothesis and concludes that fears of fragmentation and high porting costs resulting from Windows competition are unwarranted. The competing operating systems created by an effective structural remedy will start from the same code base and run on the same hardware platform, thereby reducing porting costs. In addition, the operating systems competitors will have the incentive to maintain both backward compatibility and compatibility with each other, which will serve to reduce porting costs. Finally, porting costs can be reduced further by cooperation among the operating systems competitors. The cost of porting from one operating system to another will be far smaller than from the Windows operating system running on an Intel microprocessor to a new operating system running on a different microprocessor.
The Flawed Fragmentation Critique of Structural Remedies in the *Microsoft* Case

Robert J. Levinson  
R. Craig Romaine  
Steven C. Salop

I. INTRODUCTION

Judge Jackson’s findings of fact in *Microsoft* demonstrate conclusively that Microsoft has maintained its operating systems monopoly with a variety of anticompetitive conduct.¹ There are two general approaches to remedying this monopoly. The first is to create a set of constraints on Microsoft’s behavior, thereby restraining its ability to injure competitors and consumers. Such conduct remedies would allow Microsoft to continue to operate as a monopoly. The proposed conduct remedies have serious drawbacks. First, Microsoft is unlikely to behave procompetitively following the imposition of conduct remedies. Microsoft has proven adept at circumventing antitrust conduct restrictions and could easily invent new ways to enhance and extend its monopoly power. Second, conduct remedies are highly

---

intrusive and would require ongoing, intensive regulation of Microsoft’s conduct for an extended period of time. It is doubtful that Judge Jackson would want to take on the role of regulating Microsoft’s prices, for example.

An alternative approach is for the Court to create operating systems competition by imposing a structural remedy. Two proposed structural remedies have included (i) the separation of Microsoft's operating systems and applications operations into two separate companies, or (ii) the division of Microsoft into several vertically integrated firms. A hybrid of these approaches would create operating system competition while eliminating Microsoft’s current leverage from its integration. In this hybrid remedy, the Court first would separate Microsoft’s operating systems business from the rest of the company. Then the operating systems entity would be broken up to form several identical Windows competitors (“WinCos”), say three. 2 Each WinCo would be endowed with the same intellectual property rights and would have access to equivalent pools of financial assets and skilled personnel. This hybrid structural remedy would lead to vigorous price and quality competition in operating systems. 3

---

2 In order to protect vertical integration efficiencies, Microsoft could be permitted to add additional products to the operating system. In this case, these products also would be cloned in the competing WinCos.

One criticism of a structural remedy that creates operating system competition is that the competition might quickly “fragment” the operating system standard into a number of radically different and incompatible systems. Under these conditions, it is argued, applications developers would have to perform expensive “ports,” or rewrites, of their software in order to make their existing applications work with these incompatible operating systems. Users of the new operating systems would have to pay the extra cost of the new, ported applications. The critics thus conclude that a breakup of Microsoft’s Windows business is undesirable because it could injure applications developers and result in higher applications prices.

Indeed, the preference that some commentators have expressed for a line-of-business divestiture (rather than one that would create operating system competition) appears to flow from a concern that such competition would lead to a fragmentation of the Windows standard. The line-of-business divestiture does not create competition. Because the fear of fragmentation is unwarranted, as we explain in this paper, the line-of-business divestiture remedy loses appeal.

In fact, this fragmentation criticism actually amounts to an attack of any remedy that causes operating system competition. It is competition, not the structural remedy, that allegedly leads to fragmentation. The fragmentation criticism also applies equally to the entry of competitors like Linux and the Palm OS, two of the operating systems that Microsoft claims prevents it from exercising monopoly power. But if the fragmentation

---

4 For example, see Stan J. Liebowitz, Breaking Windows: Estimating the Cost of Breaking up Microsoft Windows (Association for Competitive Technology and the ASCII Group, Inc., April 30, 1999); Nicholas Economides, “AT&T Breakup for Microsoft Doesn’t Compute,” Star Ledger, November 19, 1999; Robert Samuelson, “Consumer’s Don’t Believe They’ve Been Hurt by Microsoft,” Chicago Tribune, November 19, 1999; SIIA Monograph, supra note 3.
claims are to be believed, these alternative operating systems will never become viable competitors. The costs of porting would deter most applications developers from supporting these operating systems.

This fragmentation criticism assumes that operating systems competition must involve incompatible standards. This point may well be relevant to new operating system competitors like Linux that attempt to compete with a dominant Microsoft Windows. However, the fragmentation criticism does not apply to the hybrid structural remedy. This is because the hybrid structural remedy creates rival, compatible versions of Windows. As a result, porting costs would be minimal under this remedy. At the same time, the remedy would create price and quality competition. Consumers and OEMs would gain the ability to easily switch to an alternative brand of Windows operating system if one brand is too expensive or failure-prone. Applications developers would no longer operate in fear of a dominant Windows monopolist.

The critics who oppose a structural remedy that creates real operating system competition out of a fear of operating system fragmentation assume that the individual WinCos would be economically irrational. The new Windows competitors will begin with totally compatible products. Although the new Windows companies subsequently could choose to drastically deviate from this standard and create highly incompatible products, they are unlikely to do so. There exists an enormous installed base of Windows-compatible applications and system software programs. Consumers benefit from significant network economies arising from the large installed bases of existing applications. Consumers who switch to an incompatible operating system would lose many of these benefits. Switching to a highly incompatible operating system would
involve scrapping this substantial investment and incurring the cost of replacing all of these applications with new versions. Given these costs, consumers would resist adopting a radically different desktop operating system. If one of the WinCos drastically deviated from the Windows standard, it would risk financial ruin, because consumers might not be willing to suffer the costs of transition. In contrast, a WinCo that adheres to prior standards would stand to gain additional sales. For similar reasons, each of the WinCos would have incentives over the longer term to create new features while retaining a high degree of compatibility with competitors’ versions. As a result, porting costs will remain low.

Fragmentation and porting costs are unlikely to be significant social costs of a structural remedy that creates Windows competition. At the time of the division of the Windows entity, the competing versions of Windows will be perfectly compatible. This means that there will be no porting costs at all initially. Over time, the WinCos will differentiate themselves to some extent, but the resulting porting costs will be minor for a number of reasons. First, the companies all will have strong incentives to remain backward compatible with earlier versions of Windows, which will reduce the scope for incompatibility with each other. Second, because consumers and applications developers value compatibility, the WinCos will have incentives to remain compatible with each other. Unlike the current system, where the Windows monopolist cannot be punished by the market for being incompatible with its much smaller rivals, one of three WinCos likely would be severely punished by applications developers and users if it

---

5 Of course, were consumers to be confronted with the opportunity to switch to an operating system that was so highly superior to justify sustaining the costs of switching, the resulting “fragmentation” would be a good thing.
created a significantly incompatible version. This would also reduce the likelihood of “tipping” back to a monopoly. Third, porting costs also will remain modest because the programs will be starting out from the same code base and run on the same Intel-compatible hardware platform. Combined with the companies’ incentives to maintain backward compatibility, this common starting point would lead to far lower porting costs than would occur for porting between the Windows and the Macintosh OS. Fourth, the WinCos will have the ability as well as the incentive to cooperate to reduce porting costs. The antitrust laws would not prevent this type of cooperation. For all these reasons, the fragmentation critique is much overstated.

The remainder of this article is organized as follows. Section II briefly reviews the economics of switching costs and network effects. Section III explains why significant porting costs will not arise from a structural remedy that creates operating systems competition. We explain why the competing WinCos will sell compatible operating systems in the short run, and why they will have strong incentives in the longer run to maintain cross-platform and backward compatibility, greatly reducing and minimizing porting costs. We discuss why the competitors’ common code base and hardware platform will minimize porting costs. We also explain how cooperative efforts in the industry can work to minimize porting costs. Section IV assesses the claims made by proponents of the fragmentation hypothesis and explains why they are not valid. Section V sets forth the benefits of establishing competition in operating systems, and explains why these benefits are likely to outweigh any increase in porting costs. Section VI concludes that fears of fragmentation and high porting costs are unwarranted.
II. NETWORK EFFECTS AND SWITCHING COSTS

Market forces will give the new Windows companies strong incentives not to fragment the Windows standard. The network benefits and switching costs faced by computer users will create strong disincentives for any of the new Windows companies to abandon its installed base by marketing an incompatible operating system.

A. Network Effects

Network effects involve what economists refer to as demand-side economies of scale. Each user benefits from being part of a large network of users with whom the user can easily exchange information, and from participating in a large network of computers that can run mutually compatible software. For example, the existence of a large installed base of “Wintel” PCs in turn makes it more profitable for applications developers to design Windows-specific software applications. Consumers benefit from this expanded scope for choice. Network effects also can lead the market to

---


7 “Wintel” systems are computers running under Microsoft Windows that employ Intel or Intel-compatible microprocessors. A comparatively trivial number of Windows machines exist that run under non-Intel processors. These include certain Windows NT machines built around MIPS and DEC Alpha CPUs as well as palmtop and handheld computers employing StrongArm or other non-Intel CPUs that run under Windows CE.

8 As of 1999, there were over 70,000 applications available for Microsoft Windows. See, e.g., Microsoft Findings of Fact, at ¶ 40.
overwhelmingly adopt products embodying these standards, such as the Windows
operating system.⁹

In its 1998 Annual Report, Microsoft acknowledges that such network-related
considerations are central to its ability to sell its operating systems:

After all, the way most people truly experience their computing is through
the hardware and the applications they use. Without a rich selection of
hardware and applications that run on our operating systems, there would
be fewer compelling reasons to choose Microsoft.

This cycle of mutual benefit represents the real value of open standards.
For PC makers, IHVs [Independent Hardware Vendors], and ISVs
[Independent Software Vendors], it means new opportunities and open
access to a growing customer base. To customers, it means more
choices at lower prices. For us, it means an enormous pool of talent
focused on products that run on our platforms. Win-win-win.¹⁰

Microsoft “wins” from network effects because consumers are willing to pay for the
benefits of being part of the largest mutually compatible group of computers. These
benefits include: (a) access to the widest variety of applications software; (b) the
greatest likelihood of support from makers of computer peripherals (printers, video
boards, and so on); (c) assurance that they can easily share files with the largest pool of

⁹ See, e.g., Carl Shapiro and Hal Varian, Information Rules (Boston: Harvard Business School Press,
1999) at 176-177. Not all software markets have “tipped” in this fashion. For example, Unix is a
multiprocessing, multitasking operating system of great capability and complexity. For many years it was
the operating system of choice for users of high-end engineering workstation computers. It remains
important as an operating system for server-class machines. Throughout the period in which it has been
available on a commercial basis, Unix has been sold to end-users by computer manufacturers. Each
manufacturer’s workstations are based upon different processor architectures and have unique features.
In their efforts to gain market share -- and because consumers did not have strong reasons to resist,
owing to the absence of a pre-existing, standard set of APIs to which they and software developers had
committed -- each manufacturer has adapted Unix to work only with its specific hardware architectures.
During the 1990s, hardware makers have attempted to create and maintain common Unix standards
under the auspices of the Open Software Foundation (the OSF/1 operating system) and voluntary
standards bodies (e.g., the IEEE POSIX specification). Neither has gained the degree of industry
acceptance sufficient to establish a “standard” Unix. This may be one reason why Microsoft Windows NT
has been gaining market share at the expense of Unix in the market for workstation-class desktop
machines and servers. Linux may be evolving into the “standard” Unix.

users of each application they use, \((d)\) the ability to take advantage of access to the largest pool of experienced users and computer professionals for technical assistance; and \((e)\) the ability to use standard PC hardware.

Judge Jackson describes the network benefits enjoyed by Windows in a similar way:

Consumer demand for Windows enjoys positive network effects. A positive network effect is a phenomenon by which the attractiveness of a product increases with the number of people using it. The fact that there is a multitude of people using Windows makes the product more attractive to consumers. The large installed base attracts corporate customers who want to use an operating system that new employees are already likely to know how to use, and it attracts academic consumers who want to use software that will allow them to share files easily with colleagues at other institutions. The main reason that demand for Windows experiences positive network effects, however, is that the size of Windows’ installed base impels ISVs to write applications first and foremost to Windows, thereby ensuring a large body of applications from which consumers can choose. The large body of applications thus reinforces demand for Windows, augmenting Microsoft’s dominant position and thereby perpetuating ISV incentives to write applications principally for Windows. This self-reinforcing cycle is often referred to as a “positive feedback loop.”

Significant deviations from a common basic standard can substantially reduce these benefits, and can hinder the ability of an operating system competitor to succeed. This can occur even when the alternative operating system might otherwise be preferred. For example, the Apple Macintosh OS is incompatible with inexpensive and ubiquitous Intel-based hardware. Because of the larger installed base of PC users and the broader range of PC software, the Mac OS has been relegated to niche market status. These same considerations will slow the growth of other new operating systems that attempt to compete with Windows.
B. Switching Costs

Computer users that have committed to an operating system standard will invest in applications tailored specifically to that system. The user also invests time and money on training that is specific to the operating system and its compatible applications. Windows has benefited from these user investments.

Unless a new operating system is significantly superior to Windows, a user would be unlikely to be willing to switch to it. The user would have to bear the cost of replacing all of his or her existing applications and the costs of retraining. These costs could amount to hundreds or even thousands of dollars. In addition, some desired applications may not even exist for the new operating system platform, because applications developers would be reluctant to bear the risk of investing in support for a new incompatible operating system with a small installed base. The user also would give up the benefits of being part of a large network of users of compatible systems. As a result, new operating system entrants face significant barriers to entry, barriers that can be increased much further by the exclusionary conduct of a monopolist like Microsoft.

III. ANALYSIS OF FRAGMENTATION AND PORTING COSTS

Suppose that the court mandates the hybrid structural remedy in order to establish operating systems competition among independent Windows competitors. In the short

---

11 Microsoft Findings of Fact, at ¶ 39.
12 Because a mass applications market has not evolved in the Unix marketplace, the “installed base inertia” that is created by switching costs does not approach that of Windows.
run, there will be no porting costs because the competing operating systems will be identical. Because of these network effects and switching costs, high porting costs also are not a serious concern in the longer run. In the longer run, the WinCos will have strong incentives to maintain compatibility. First, they have incentives to retain backward compatibility to take advantage of the large installed base of Windows users and the applications that run on Windows. Second, the WinCos have incentives to maintain compatibility with each other going forward in order to minimize any porting costs that might prevent consumers from adopting their particular operating system. Third, because each WinCo starts with a common code base, and because each runs on the same Intel-compatible hardware, porting costs also are reduced. Finally, supplementary cooperative efforts among the WinCos and other industry participants to set standards may also minimize porting costs.

A. No Short Run Porting Costs

In his discussion of operating systems fragmentation, Professor Liebowitz assumes that the operating systems of the various WinCos would be incompatible with one another, requiring applications developers to extensively rewrite their programs to run on the new platforms. This assumption is contrary to the fundamental logic of a hybrid structural remedy that divides Microsoft into competing operating system companies. That structural remedy would provide each WinCo with full rights to the Windows source code, applications interfaces, and user interfaces. Because each operating systems competitor would start with the ability to market a fully compatible

---

13 Liebowitz, supra at n. 4.
version of the Windows operating systems, porting costs would not be an issue at the time of the breakup.

During this initial period, the competitive weapons available to each operating system competitor would be lower prices and better customer service to consumers and OEMs. The competing WinCos also could rapidly compete by eliminating software defects. Elimination of operating system “bugs” would allow existing applications to more reliably work in concert with the operating system, rendering these more compatible. As a result, applications developers would incur reduced costs of developing workarounds or fixes for their own software to compensate for operating system deficiencies and would benefit from reduced costs of fielding customers’ technical support calls. A WinCo might facilitate these fixes by disclosing APIs and the source code relevant to the problem. The WinCos also might compete by giving OEMs the ability to customize opening screens.

B. Longer Term Incentives and Ability to Maintain Compatibility

The claim that a structural remedy will create significant porting costs relies on the erroneous assumption that the new Windows competitors will create and successfully market highly incompatible operating systems. This assumption is contrary to the logic of network externalities and switching costs. These characteristics of the operating systems market will give the WinCos strong incentives to remain highly compatible.

1. Incentives to Retain Backward Compatibility

The Windows competitors created by the hybrid structural remedy would not have the incentive to alter their versions of Windows to the point where consumers are
unable to run the installed base of existing applications. This is because such a
strategy would create considerable costs and inconvenience for users, application
developers, and the WinCos themselves. As a result, the market would severely
punish a WinCo that was not backward compatible. Because the remedy creates
competition among a number of WinCos, such punishment would be possible.
Backward compatibility, on the other hand, will significantly reduce porting costs for
applications developers.

Consumers prefer operating systems that afford them access to a large pool of
existing applications and relevant information. Consumers would lose this benefit and
incur the costs of new applications and training if they moved to an operating system
that was incompatible with these APIs. Any operating system competitor that chooses
to be incompatible with these APIs would face the difficult, expensive, and time
consuming task of convincing a significant portion of the installed base of Windows
applications users to replace their existing applications and forego the benefits of
remaining with an established standard.

\footnote{As Judge Jackson notes, Consumer interest in a PC operating system derives primarily from the ability of that system to run applications. The consumer wants an operating system that runs not only types of applications that he knows he will want to use, but also those types in which he might develop an interest later. Also, the consumer knows that if he chooses an operating system with enough demand to support multiple applications in each product category, he will be less likely to find himself straitened later by having to use an application whose features disappoint him. Finally, the average user knows that, generally speaking, applications improve through successive versions. He thus wants an operating system for which successive generations of his favorite applications will be released — promptly at that. The fact that a vastly larger number of applications are written for Windows than for other PC operating systems attracts consumers to Windows, because it reassures them that their interests will be met as long as they use Microsoft’s product.}

Microsoft Findings of Fact, at ¶ 37.
In this regard, it is worth noting that over 80 percent of Microsoft’s revenue comes from business customers.\footnote{This figure was provided by Microsoft executive Jeff Raikes in his comments to financial analysts on July 22, 1999. See http://www.microsoft.com/msft/speech/analystmtg99/raikesfam99.htm.} Because of their large investments in existing Windows applications and related training, such customers would be particularly resistant to the idea of switching to an incompatible operating system.\footnote{Sometimes Unix is cited as an example of what would happen to the personal computer industry if a structural remedy were imposed. However, there are significant differences that make Unix a poor comparison. Windows today has hundreds of millions of users, and tens of thousands of commercial applications written to work only under Windows. A Windows user who switches to an incompatible operating system would all at once lose access to this large network of users, information, and applications, and would have to spend considerable sums to obtain access to an inferior network. As a result, Windows users would be highly resistant to the idea of switching to an incompatible operating system. The Unix world, in contrast, does not, and never has had, a monolithic commercial set of technical compatibility standards. Historically, Unix systems from different vendors have been made up of computers with mutually incompatible hardware architectures, for which each vendor tailored its particular “flavor” of Unix.}

A WinCo that elects to sell a backward incompatible operating system also would face the similarly difficult prospect of persuading significant numbers of applications developers to expend the resources necessary to create new applications for its standard.\footnote{Judge Jackson cites the historical record to make this point: IBM’s inability to gain widespread developer support for its OS/2 Warp operating system illustrates how the massive Windows installed base makes it prohibitively costly for a rival operating system to attract enough developer support to challenge Windows. In late 1994, IBM introduced its Intel-compatible OS/2 Warp operating system and spent tens of millions of dollars in an effort to attract ISVs to develop applications for OS/2 and in an attempt to reverse-engineer, or “clone,” part of the Windows API set. Despite these efforts, IBM could obtain neither significant market share nor ISV support for OS/2 Warp. Thus, although at its peak OS/2 ran approximately 2,500 applications and had 10% of the market for Intel-compatible PC operating systems, IBM ultimately determined that the applications barrier prevented effective competition against Windows 95. For that reason, in 1996 IBM stopped trying to convince ISVs to write for OS/2 Warp. IBM now targets the product at a market niche, namely enterprise customers (mainly banks) that are interested in particular types of application that run on OS/2 Warp. The fact that IBM no longer tries to compete with Windows is evidenced by the fact that it prices OS/2 Warp at about two-and-one-half times the price of Windows 98. (Microsoft Findings of Fact, at ¶ 46.)} These difficulties would be compounded by the need to convince OEMs to preload and support this incompatible backward operating system.\footnote{According to Judge Jackson,}
manufacturers were able to choose among operating systems suppliers, they would insist on operating system characteristics that consumers want, including compatibility with existing Windows applications.

OEMs stand to lose in two ways from preloading a Windows-incompatible operating system in the PCs they sell. First, the OEMs would bear significant support costs when buyers discover that the applications they already own do not work on their new PCs. Second, large numbers of consumers might refuse to purchase such computers at all, because they would understand that they would have to endure the costs of new applications and training in order to be productive with these machines.

In contrast, operating system development strategies that retain backward compatibility with the pre-remedy versions of Windows would better match the preferences of consumers, OEMs, applications developers, and the new operating system competitors. Consumers would continue to gain from access to the largest pool of compatible applications and information. OEMs would avoid consumer complaints and support costs. Applications developers would profit by avoiding porting costs.

OEMs are the most important direct customers for operating systems for Intel-compatible PCs. Because competition among OEMs is intense, they pay particularly close attention to consumer demand. OEMs are thus not only important customers in their own right, they are also surrogates for consumers in identifying reasonably-available commercial alternatives to Windows. Without significant exception, all OEMs pre-install Windows on the vast majority of PCs that they sell, and they uniformly are of a mind that there exists no commercially viable alternative to which they could switch in response to a substantial and sustained price increase or its equivalent by Microsoft. For example, in 1995, at a time when IBM still placed hope in OS/2’s ability to rival Windows, the firm nevertheless calculated that its PC company would lose between seventy and ninety percent of its sales volume if failed to load Windows 95 on its PCs. Although a few OEMs have announced their intention to pre-install Linux on some of the computers they ship, none of them plan to install Linux in lieu of Windows on any appreciable number of PC (as opposed to server) systems. For its part, Be is not even attempting to persuade OEMs to install the BeOS on PCs to the exclusion of Windows. (Microsoft Findings of Fact, at ¶ 54.)
Adherence to established standards would allow the new operating system companies to maximize their benefits from access to Microsoft’s Windows source code. Taken together, this means that the WinCos would have very strong incentives to sell operating systems that are compatible with prior versions of Windows.

Operating system vendors subject to such market constraints can nevertheless engage in quality competition by improving and differentiating their products over time. This would be accomplished by eliminating bugs, increasing speed and reducing susceptibility to crashes, and adding new features and APIs to the base Windows operating system while retaining the existing Windows APIs.¹⁹ Because each system remains fully compatible with applications written to the common Windows APIs, such differentiation would not create such significant differences to force applications developers to incur large porting costs.²⁰ In fact, since developers have the option of writing to the pre-existing APIs, they are not forced to incur any porting costs. This differentiation also would continue to allow pre-existing applications to run on the new operating systems.²¹

Because such new features represent small incremental changes to the existing Windows APIs, the costs of adapting applications to use these features would be far

---

¹⁹ This competition also might involve voluntary disclosure of APIs. By disclosing its APIs, one of the WinCos might gain a competitive advantage by making it easier for applications developers to produce higher quality programs at lower cost and increase the magnitude of network effects. Of course, this disclosure may well lead the other WinCos to follow suit as well.

²⁰ Microsoft’s past actions demonstrate the ability and incentive to maintain compatibility with prior versions. For example, Windows 98 added features not supported by the original release of Windows 95, such as support for the FAT32 file system. With the exception of very specialized features of certain disk utilities, however, the transition from Windows 95 to Windows 98 created few, if any, incompatibilities for existing applications. Thus applications developers were not forced to rewrite their code to run on the new system. See also Microsoft Findings of Fact, at ¶ 44.

²¹ Backward compatibility beyond one or two releases might be maintained within a smaller operating system by a middleware plug-in to be used only by those who run such older programs.
smaller than the costs of porting these applications to support an entirely different operating system. Moreover, applications code changes that take advantage of the new operating system features constitute genuine improvements. In addition, as we discuss, the WinCos will have incentives to minimize porting costs as the process occurs.

2. Incentives to Maintain Compatibility and Minimize Porting Costs

Because each of the new operating system companies will have powerful competitive reasons to maintain compatibility with the installed base of applications, they also will have incentives to maintain a high degree of compatibility with each other. Because of their similar code bases, maintaining compatibility will not be difficult. This will tend to minimize porting costs over time.

In the current marketplace, Microsoft controls the dominant operating system. Applications developers wishing to sell to the mass market have no choice but to maintain compatibility with Windows. If Microsoft makes changes to its operating system, developers must follow and support that change. Thus, there are no constraints forcing Microsoft to maintain compatibility with other operating systems.

The situation facing the competing WinCos will be much different. None of them will be dominant. Customers and developers will have a choice. Thus, if one of the

---

22 It is inappropriate to view the costs of these improvements as inefficient porting costs being foisted on consumers and applications developers by antitrust law. Instead, they are the costs of supporting beneficial improvements to Windows. Furthermore, earlier versions of the applications would continue to work as before. Providing applications developers with the opportunity to take advantage of entirely new functionality is very different from forcing applications vendors to rewrite their applications to accommodate changes in the interfaces that allow access to existing functions. The competitive pressure to take advantage of opportunities for such product improvement benefits consumers because it leads to the development of superior applications that better satisfy consumer wants.
WinCos changes its program to make it highly incompatible with the others, developers are not forced to support that change. Developers and users would have the ability to punish a WinCo that makes an unwanted design decision. This discipline will deter the WinCos from creating operating systems that are not backward compatible or that require high porting costs.

At the same time, innovation can continue to advance without creating a high degree of incompatibility or porting costs. This can be illustrated by a simple hypothetical example. Suppose there are three operating system competitors, A, B, and C. Suppose that operating system C is an exact clone of Windows 98, with a set of APIs that are identical to the Windows 98 APIs. Suppose that A and B each include all of the Windows 98 APIs, but also a few additional, new APIs. Further, suppose that A's and B's sets of new APIs differ. In this hypothetical, applications written to work with Windows 98 (i.e., with calls only to the original Windows 98 APIs) will be fully compatible with all three operating systems.

New applications that employ the extensions introduced by A or B are not automatically fully incompatible with the other operating systems. However, applications developers can maintain cross-platform compatibility by writing the installation routines of their applications so that they detect which version of Windows is running. The installation program then could load application components that support the appropriate enhanced operating system, or could configure the application to run using only the common API subset (in our example, those of system C). This would allow the application to run on all three of the operating systems.
Applications developers that choose to support proprietary extensions to the core Windows standard would face relatively minor costs of porting. This is because the bulk of the application program code that relies upon standard Windows APIs would not have to be rewritten. Only those few routines that benefit from the use of proprietary extensions would be affected. Because developers would remain free to write code that took advantage only of the core Windows APIs (and thus runs on all 3 Windows versions), these costs would be incurred only if the benefits of new APIs were greater than these small costs. In addition, the WinCos would have the incentive to make this process as simple as possible for the application developers. Otherwise, the application developers might not support the new APIs, or might promote the competing WinCo’s systems.

Over time, the best and most popular proprietary extensions created by the Windows competitors will appeal to applications developers who perceive commercial advantage in supporting these features. The competing WinCos will have incentives to imitate these most successful features and their associated APIs. This would allow each to support the widest base of application functionality and to attract (or keep) customers. The ultimate effect of such efforts to expand the extent of cross-operating-system compatibility will be to expand the “core” Windows standard itself.

Such technological growth within the Windows standard likely would occur at least as rapidly as under the Microsoft monopoly and clearly would be more responsive to the desires of applications developers, consumers, and OEMs. This is because the changes to the Windows standard would be propelled by the forces of competition rather than by the decisions of a monopolist bent on maintaining its monopoly with
exclusionary conduct. At the same time, developers could avoid incremental
development and marketing costs by designing applications that use only the common,
industry-standard Windows APIs.

3. *Common Code Base and Hardware Platform*

These porting costs also would be modest because all the WinCos would be
starting from the same code base and hardware platform. The common code and
programming protocols would make it easier and cheaper for applications developers to
do any rewriting of the code that might be necessary. In addition, porting costs are
lowered because all the WinCo operating systems are designed to run on the same
Intel-compatible microprocessor. This common hardware platform reduces the costs to
lower levels than the cost of porting Windows programs on Intel microprocessors to
(say) the Solaris OS on Sun or the Mac OS on Motorola microprocessors.\(^{23}\) Because
these other operating systems run on different microprocessors, ports require the use of
different development tools and separate testing and performance tuning. These other
operating systems also require different binary executable code for each different
microprocessor, potentially requiring other logistical costs.

4. *Supplementary Cooperative Efforts to Minimize Porting Costs*

The new operating systems competitors also may cooperate to maintain
compatibility. Such supplementary cooperation may take the form of a formal standard-
setting process, administered by either an impartial industry group or a computer

---

\(^{23}\) Even Professor Liebowitz acknowledges this point, stating that “the difference between the Macintosh
and Windows will undoubtedly be greater than the differences between the various flavors of Windows,
overstating the porting costs.” Liebowitz, *supra* note 4 at 5.
industry group consisting of interested applications, operating systems, and hardware producers.24 Such a group may formally adopt existing improvements to the Windows standard as "industry standard," making the relevant APIs available to all market participants, or may go further by working to create and implement new Windows-standard APIs and functionality by consensus.

Professor Liebowitz suggests that antitrust laws would prevent the WinCo competitors from cooperating to minimize porting costs.25 This view is mistaken. The large number of software industry standards bodies demonstrates that it is possible to coordinate industry standards among software competitors without running afoul of the antitrust laws. Indeed, the AT&T decree required cooperative standard setting.26

With a small number of WinCo competitors, a formal standards body might be unnecessary to coordinate this standard-setting and exchanges of technical information. The new companies may, for example, cooperate with each other and with specialists in cross-platform compatibility technologies to adopt common API standards or to create "bridges" that allow their operating systems to run code written to take advantage of competitors' proprietary extensions. Bridging technologies could also be developed in order to mitigate the effects of whatever minor incompatibilities that might arise.

---

24 This occurs in the industry today. For example, Internet standards -- such as HTTP, XML, and SSL -- are developed and/or overseen by groups such as the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C).

25 Liebowitz at 7.

26 The AT&T decree required the divested companies to fund the Bellcore organization in part to "set the standards which will permit telecommunications to continue to operate in an engineering sense as one national network," and thus to serve as "an important check against deterioration and fragmentation of the existing telephone system." United States v. Western Electric Co., 569 F.Supp.1057, 1118 (D.D.C. 1983).
IV. PROFESSOR LIEBOWITZ’S OVERSTATED PORTING COSTS

Professor Liebowitz argues that porting costs would be quite large. However, his estimates are based on invalid assumptions that lead them to be biased upwards. He also ignores the benefits to consumers, OEMs and applications developers of operating system competition.

A. Invalid Assumptions

Professor Liebowitz claims that applications producers would be subjected to considerable costs of porting as the result of a structural remedy that creates operating system competition. His empirical support of these claims is quite weak. Much of his discussion is anecdotal in nature, referring to a small survey of software producers and press reports regarding very different porting situations. In fact, Professor Liebowitz concedes that his survey data could not be used to predict "with any great precision what the actual costs of porting will be."27

Professor Liebowitz’s references are not applicable to the hybrid structural remedy for several reasons. First, these references deal with operating systems that have much smaller installed bases of applications and users, such as the Linux and Apple Macintosh operating systems. As described above, a major reason why the Windows standard would not become fragmented is the large scale of its installed base. Second, the costs of porting between different versions of Windows will be far smaller than the

27 Liebowitz at 7. One reason why these estimates are unreliable is the small number of respondents. Given the different software businesses engaged in by his interview respondents, it is impossible to judge whether their estimates were representative of the beliefs of the industry as a whole. Professor Liebowitz also does not reveal the questions put to his interviewees. The nature and tone of the questions can bias the results.
costs of porting applications from the Windows platform to the Macintosh because of the common code base and hardware.\textsuperscript{28} As discussed above, the Mac and Windows operating systems differ greatly in their design and API sets. In addition, these operating systems run on radically different hardware. Porting applications between these disparate environments generally would require wholesale rewrites of source code. The costs of porting applications between such different architectures would greatly exceed the costs of in adapting existing Windows applications to run under versions of Windows that incorporate incremental modifications and a common set of core APIs.

Professor Liebowitz assumes that the introduction of each new operating system causes the percentage of applications developers' revenues that are absorbed by R&D and technical support costs to rise by a factor of one-quarter. He also assumes that the percentage of applications developers' revenues that are absorbed by certain

\textsuperscript{28} In fact, Professor Liebowitz both concedes and ignores this point in his analysis. In particular, Professor Liebowitz discusses the importance of porting costs by citing Intuit's costs of supporting and upgrading the Macintosh version of its Quicken financial program. He estimates these as 10\% of total revenues. It is odd that Professor Liebowitz would stress this estimate in his paper, because he also concedes that this figure "cannot be taken as typical" of the outcome of a structural remedy because "the difference between the Macintosh and Windows will undoubtedly be greater than the differences between the various flavors of Windows, overstating the porting costs." \textit{Id.} at 5.

Professor Liebowitz's estimate is likely to significantly overstate the costs of porting in a market with multiple, competing versions of Windows. The Macintosh and Windows operating systems differ radically. The source code for the Windows version of an application therefore will largely or completely differ from that of the Macintosh version. As already explained, applications written to run under one version of Windows will run under another with little or no modification, meaning that porting costs will be minimal or nonexistent. In addition, because the Macintosh and Windows operating systems run under different microprocessors, ports of applications between these platforms require the use of different development tools and separate performance tuning on each CPU. This would not be an issue in a competitive Windows market because all the Windows operating systems will run on the Intel platform. Lastly, the costs of upgrading and supporting Quicken (or any other pre-existing Mac application) for the Macintosh platform cannot be attributed entirely to porting costs. The costs of porting associated with Quicken for the Mac at most would be the costs of adapting innovations first implemented in the Windows version for use on the Macintosh. Other Macintosh-related development and support issues, such as those relating to bug fixes and new features not already present in the Windows version, are not porting costs.
marketing-related expenses each rise by 5% as each new operating system is introduced. Professor Liebowitz employs these and other ad hoc assumptions to create his estimates of the porting costs. On this basis, he asserts that developers’ costs would rise on average by 6.46% of revenues. However, because these estimates are based upon unreliable data, policymakers cannot rely upon them.

B. Biased Estimates

In Section III, we explained why significant fragmentation is unlikely to result from a structural remedy that creates Windows competition. If Professor Liebowitz’ survey respondents were told to assume that significant fragmentation would occur, for example, on the scale of differences between Windows and the Macintosh operating system, then they likely would respond with unrealistically high estimates of porting costs. Professor Liebowitz’ paper hints that respondents may have been told to assume just this.

29 Liebowitz at 8-11. His analysis begins with two sets of third-party data. From a 1997 KPMG Peat Marwick report, Professor Liebowitz obtained estimates of software publishers’ average costs, as a fraction of revenues, arising from R&D, Sales & Marketing, Technical and Administrative, and other activities. This data was broken down into three software company size categories: Under $5 million; from $5 to $50 million; and larger than $50 million. Using this data he constructed estimated weighted industry-average percentages of the total revenues attributable to each cost category. IDC forecasts of revenues for software for 32-bit Windows systems then were used to construct revenue forecasts for 32-bit Windows software other than operating systems for the three years ending in 2002.

Professor Liebowitz multiplies the percentage of revenues accounted for by each cost category (as obtained using the 1997 KPMG data) by his assumed percentage increase factors to obtain estimated changes in the percentage of revenues that would be accounted for by these cost categories following the introduction of each new Windows version. Summing these leads to Professor Liebowitz’ claim that each new operating system would cause applications developers’ costs to increase, on average, by 6.46% of revenues. He then multiplies this value by the IDC revenue forecasts for 2000-2002 to obtain his estimate of the dollar costs of porting resulting from the introduction of each new operating system.

30 Id. at 11 and at Section VII (Appendix).

31 Id. at 7 As Professor Liebowitz states, “We should note that there is little question that the new competing versions of Windows will have to diverge over time. If the various versions of Windows truly compete with one another, as the DOJ hopes, they will have separate R&D efforts and will be making
As discussed above, however, the likely degree of differentiation would not impose significant porting costs on developers or consumers.\textsuperscript{32} These proprietary extensions can be added while maintaining complete compatibility with the complete set of existing Windows APIs. Under these conditions, developers would be free to market their existing programs as compatible with all of the competing Windows implementations and produce new versions that run on each of them without any significant porting costs.

Professor Liebowitz instead assumes that the new operating system competitors will opt out of the Windows standard, forcing consumers to jettison their existing investments in applications software and coercing applications developers to rewrite their products. In reality, the costs of switching and the benefits of network effects create tremendous incentives for consumers to prefer a compatible operating system and for the new operating system companies to retain compatibility with the installed base and each other. Therefore, such conduct by competing WinCos would be contrary to their own incentives as well as the interests of consumers, OEMs and developers.

V. BENEFITS OF COMPETITION VERSUS PORTING COSTS

As we have discussed, each Windows operating system competitor will have strong incentives to maintain a high degree of compatibility with the installed base of

\textsuperscript{32} See Section III.B.2, \textit{supra}.
applications and with each other. This means that applications developers will not be put in the position of having to bear high porting costs to support the operating systems offered by the WinCos. Applications developers may elect to improve their products by incrementally adapting their applications to suit the proprietary extensions of individual operating systems. However, the costs of these incremental adaptations clearly would be far smaller than those required to port an entire application’s code to an entirely alien operating system.

These changes also will benefit users by providing variety and improvements that users value. Professor Liebowitz’ attack on the structural remedy is really a more general assault against competition in software markets. Our view is rather different. Competition gives consumers increased choices and spurs innovation. The competition made possible by a structural remedy also would result in lower operating system prices, increased operating systems output, and increased sales of personal computers. Compared with the prospect of such gains to society, the issue of these minor porting costs pales in insignificance.

From this perspective, it makes no sense to argue that a structural remedy should be avoided in the Microsoft case because of concerns related to the fragmentation of the Windows standard. As Judge Jackson detailed in his Findings of Fact, Microsoft pursued a variety of anticompetitive exclusionary conduct with the express purpose and effect of protecting its operating systems monopoly. Given the network-related advantages that it enjoyed while it anticompetitively sought to neutralize Netscape and the competitive threat posed by new technologies such as Java, it is clear that the intent and effect of Microsoft’s conduct has been to destroy competition and competitors
whenever their new technologies or business models suggest a potential long-term threat to Microsoft’s market dominance. A structural remedy that creates Windows competitors would eliminate the ability of any one Windows company to continue to distort the progress and competitiveness of the computer industry in this way.

Over time, the competitive Windows companies might differentiate their products to some extent at the margin, leading some consumers to alter their choices and bear some switching costs. It also might create some minor porting costs for applications developers. However, the costs to society of this will be minimal. In contrast, the elimination of the costs of today’s Microsoft monopoly would benefit society greatly.

VI. CONCLUSIONS

We have explained why fears of fragmentation and high porting costs resulting from Windows competition are unwarranted. The competing operating systems created by the structural remedy will start from the same code base and run on the same hardware platform, thereby reducing porting costs. In addition, the WinCos will have the incentive to maintain backward compatibility and compatibility with each other, which will serve to minimize porting costs. Finally, porting costs can be reduced further by cooperation among the WinCos. The cost of porting from one WinCo to another will be far smaller than from the Windows operating system running on an Intel microprocessor to a new operating system (such as the MacIntosh) running on a different microprocessor. Meanwhile consumers and applications developers will benefit from real operating systems competition.
Finally, it is also important to note that the argument that increased competition is bad is antithetical to the premises of the antitrust laws, which rely on competition, not entrenched monopoly, to constrain prices and spark innovation that benefits consumers. As stated by the Supreme Court, the antitrust laws have “foreclose[d] the argument that because of the special characteristics of a particular industry, monopolistic arrangements will better promote trade and commerce than competition.”33 In this regard, the fragmentation critique of operating system competition amounts to a “frontal assault on the basic policy of the Sherman Act.”34 Under the antitrust laws, the market, not a monopolist, should decide whether the benefits of price and quality competition are more attractive than the carefully rationed future planned by a monopolist.35

---

33 National Society of Professional Engineers v. United States, 435 U.S. at 690.
34 Id. at 680.
35 As Professors Areeda and Hovencamp put it, “the compatibility problem is less in the case of nondominant firms developing a common software code than in the case of the dominant firm,” because a “nondominant firm has every incentive to maintain compatibility” to prevent the “immediate loss of sales” that “departing from the standard entails.” Areeda & Hovenkamp, Antitrust Law, ¶ 704.3, at 214 (1999 supp.).