B. The Applications Barrier to Entry

1. Description of the Applications Barrier to Entry

36. Microsoft's dominant market share is protected by the same barrier that helps define the market for Intel-compatible PC operating systems. As explained above, the applications barrier would prevent an aspiring entrant into the relevant market from drawing a significant number of customers away from a dominant incumbent even if the incumbent priced its products substantially above competitive levels for a significant period of time. Because Microsoft's market share is so dominant, the barrier has a similar effect within the market: It prevents Intel-compatible PC operating systems other than Windows from attracting significant consumer demand, and it would continue to do so even if Microsoft held its prices substantially above the competitive level.

37. 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.

38. Software development is characterized by substantial economies of scale. The fixed costs of producing software, including applications, is very high. By contrast, marginal costs are very low. Moreover, the costs of developing software are "sunk" — once expended to develop software, resources so devoted cannot be used for another purpose. The result of economies of scale and sunk costs is that application developers seek to sell as many copies of their applications as possible. An application that is written for one PC operating system will operate on another PC operating system only if it is ported to that system, and porting applications is both time-consuming and expensive. Therefore, application developers tend to write first to the operating system with the most users — Windows. Developers might then port their applications to other operating systems, but only to the extent that the marginal added sales justify the cost of porting. In order to recover that cost, ISVs that do go to the effort of porting frequently set the price of ported applications considerably higher than that of the original versions written for Windows.

39. 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."

40. What for Microsoft is a positive feedback loop is for would-be competitors a vicious cycle. For just as Microsoft's large market share creates incentives for ISVs to develop applications first and foremost for Windows, the small or non-existent market share of an aspiring competitor makes it prohibitively expensive for the aspirant to develop its PC operating system into an acceptable substitute for Windows. To provide a viable substitute for Windows, another PC operating system would need a large and varied enough base of compatible applications to reassure consumers that their interests in variety, choice, and currency would be met to more-or-less the same extent as if they chose Windows. Even if the contender attracted several thousand compatible applications, it would still look like a gamble from the consumer's perspective next to Windows, which supports over 70,000 applications. The amount it would cost an operating system vendor to create that many applications is prohibitively large. Therefore, in order to ensure the availability of a set of applications comparable to that available for Windows, a potential rival would need to induce a very large number of ISVs to write to its operating system.

41. In deciding whether to develop an application for a new operating system, an ISV's first consideration is the number of users it expects the operating system to attract. Out of

this focus arises a collective-action problem: Each ISV realizes that the new operating system could attract a significant number of users if enough ISVs developed applications for it; but few ISVs want to sink resources into developing for the system until it becomes established. Since everyone is waiting for everyone else to bear the risk of early adoption, the new operating system has difficulty attracting enough applications to generate a positive feedback loop. The vendor of a new operating system cannot effectively solve this problem by paying the necessary number of ISVs to write for its operating system, because the cost of doing so would dwarf the expected return.

42. Counteracting the collective-action phenomenon is another known as the "first-mover incentive." For an ISV interested in attracting users, there may be an advantage to offering the first and, for a while, only application in its category that runs on a new PC operating system. The user base of the new system may be small, but every user of that system who wants such an application will be compelled to use the ISV's offering. Moreover, if demand for the new operating system suddenly explodes, the first mover will reap large sales before any competitors arrive. An ISV thus might be drawn to a new PC operating system as a "protected harbor." Once first-movers stake claims to the major categories of applications, however, there is a strong chance that the new operating system could stall; it would not support the most familiar applications, nor the variety and number of applications, that attract large numbers of consumers, and there would no longer exist a first-mover incentive to attract additional ISVs to the important application categories. Although the upstart operating system might find itself with enough applications support to hold a fraction of the market, the collective-action phenomenon would still prevent the system from gaining the kind of positive feedback

momentum that can turn a fringe entrant into a rival that would put competitive pressure on Windows.

43. The cost to a would-be entrant of inducing ISVs to write applications for its operating system exceeds the cost that Microsoft itself has faced in inducing ISVs to write applications for its operating system products, for Microsoft never confronted a highly penetrated market dominated by a single competitor. Of course, the fact that it is extremely difficult for an efficient would-be rival to accumulate enough applications support to compete with Windows does not mean that sustaining its own applications support is effortless for Microsoft. In fact, if Microsoft stopped investing the hundreds of millions of dollars it spends each year inducing ISVs to write applications for Windows, it might become easier than it currently is for a competitor to develop its own positive feedback loop. But given that Windows today enjoys overwhelmingly more applications support than any other PC operating system, it would still take that competitor years to develop the necessary momentum. Plus, while Microsoft may spend more on platform "evangelization," even in relative terms, than any other PC operating-system vendor, it is not difficult to understand why it is worthwhile for the principal beneficiary of the applications barrier to devote more resources to augmenting it than aspiring rivals are willing to expend in speculative efforts to erode it.

44. Microsoft continually releases "new and improved" versions of its PC operating system. Each time it does, Microsoft must convince ISVs to write applications that take advantage of new APIs, so that existing Windows users will have incentive to buy an upgrade. Since ISVs are usually still earning substantial revenue from applications written for the last version of Windows, Microsoft must convince them to write for the new version. Even if ISVs

are slow to take advantage of the new APIs, though, no applications barrier stands in the way of consumers adopting the new system, for Microsoft ensures that successive versions of Windows retain the ability to run applications developed for earlier versions. In fact, since ISVs know that consumers do not feel locked into their old versions of Windows and that new versions have historically attracted substantial consumer demand, ISVs will generally write to new APIs as long as the interfaces enable attractive, innovative features. Microsoft supplements developers' incentives by extending various 'seals of approval' — visible to consumers, investors, and industry analysts — to those ISVs that promptly develop new versions of their applications adapted to the newest version of Windows. In addition, Microsoft works closely with ISVs to help them adapt their applications to the newest version of the operating system — a process that is in any event far easier than porting an application from one vendor's PC operating system to another's. In sum, despite the substantial resources Microsoft expends inducing ISVs to develop applications for new versions of Windows, the company does not face any obstacles nearly as imposing as the barrier to entry that vendors and would-be vendors of other PC operating systems must overcome.

2. Empirical Evidence of the Applications Barrier to Entry

45. The experiences of IBM and Apple, Microsoft's most significant operating system rivals in the mid- and late 1990s, confirm the strength of the applications barrier to entry.

a. OS/2 Warp

46. 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.

b. The Mac OS

47. The inability of Apple to compete effectively with Windows provides another example of the applications barrier to entry in operation. Although Apple's Mac OS supports more than 12,000 applications, even an inventory of that magnitude is not sufficient to enable Apple to present a significant percentage of users with a viable substitute for Windows. The absence of a large installed base, in turn, reinforces the disparity between the applications made available for the Mac OS and those made available for Windows, further inhibiting Apple's sales. The applications barrier thus prevents the Mac OS from hindering Microsoft's ability to control price, regardless of whether the Mac OS is regarded as being in the relevant market or not.

c. Fringe Operating Systems

48. The applications barrier to entry does not prevent non-Microsoft, Intel-compatible PC operating systems from attracting enough consumer demand and ISV support to survive. It does not even prevent vendors of those products from making a profit. The barrier does, however, prevent the products from drawing a significant percentage of consumers away from Windows.

49. As discussed above, Be markets an Intel-compatible PC operating system, called BeOS, that is specially suited to support multimedia functions. The operating system survives on a relatively minuscule number of applications (approximately 1,000) and a user base which, at around 750,000, is trivial compared to the number of Windows users. One of the reasons the BeOS can even attract that many users despite its small base of applications is that it advertises itself as a complement to, rather than as a substitute for, Windows. Although the BeOS could run an Intel-compatible PC system without Windows, it is almost always loaded on a system along with Windows. What is more, when these dual-loaded PC systems are turned on, Windows automatically boots; the user must then take affirmative steps to invoke the BeOS. While this scheme allows the BeOS to occupy a niche in the market, it does not place the product on a trajectory to replace Windows on a significant number of PCs. The special multimedia support provided by the BeOS may, for a small number of users, outweigh the disadvantages of maintaining two large, complex operating systems on one PC. Of that group, however, it is likely that only a tiny number of users will find that support so attractive that they would be willing to forego Windows, and its huge base of compatible applications, altogether.

50. The experience of the Linux operating system, a version of which runs on Intel-compatible PCs, similarly fails to refute the existence of an applications barrier to entry. Linux is an "open source" operating system that was created, and is continuously updated, by a global network of software developers who contribute their labor for free. Although Linux has between ten and fifteen million users, the majority of them use the operating system to run servers, not PCs. Several ISVs have announced their development of (or plans to develop) Linux versions of their applications. To date, though, legions of ISVs have not followed the lead of these first movers. Similarly, consumers have by and large shown little inclination to abandon Windows, with its reliable developer support, in favor of an operating system whose future in the PC realm is unclear. By itself, Linux's open-source development model shows no signs of liberating that operating system from the cycle of consumer preferences and developer incentives that, when fueled by Windows' enormous reservoir of applications, prevents non-Microsoft operating systems from competing.

3. Open-Source Applications Development

51. Since application developers working under an open-source model are not looking to recoup their investment and make a profit by selling copies of their finished products, they are free from the imperative that compels proprietary developers to concentrate their efforts on Windows. In theory, then, open-source developers are at least as likely to develop applications for a non-Microsoft operating system as they are to write Windows-compatible applications. In fact, they may be disposed ideologically to focus their efforts on open-source platforms like Linux. Fortunately for Microsoft, however, there are only so many developers in the world willing to devote their talents to writing, testing, and debugging software pro bono publico. A

small corps may be willing to concentrate its efforts on popular applications, such as browsers and office productivity applications, that are of value to most users. It is unlikely, though, that a sufficient number of open-source developers will commit to developing and continually updating the large variety of applications that an operating system would need to attract in order to present a significant number of users with a viable alternative to Windows. In practice, then, the open-source model of applications development may increase the base of applications that run on non-Microsoft PC operating systems, but it cannot dissolve the barrier that prevents such operating systems from challenging Windows.

4. Cloning the 32-Bit Windows APIs

52. Theoretically, the developer of a non-Microsoft, Intel-compatible PC operating system could circumvent the applications barrier to entry by cloning the APIs exposed by the 32-bit versions of Windows (Windows 9x and Windows NT). Applications written for Windows would then also run on the rival system, and consumers could use the rival system confident in that knowledge. Translating this theory into practice is virtually impossible, however. First of all, cloning the thousands of APIs already exposed by Windows would be an enormously expensive undertaking. More daunting is the fact that Microsoft continually adds APIs to Windows through updates and new versions. By the time a rival finished cloning the APIs currently in existence, Windows would have exposed a multitude of new ones. Since the rival would never catch up, it would never be able to assure consumers that its operating system would run all of the applications written for Windows. IBM discovered this to its dismay in the mid-1990s when it failed, despite a massive investment, to clone a sufficiently large part of the 32-bit Windows APIs. In short, attempting to clone the 32-bit Windows APIs is such an expensive,

uncertain undertaking that it fails to present a practical option for a would-be competitor to Windows.

C. Viable Alternatives to Windows

53. That Microsoft's market share and the applications barrier to entry together endow the company with monopoly power in the market for Intel-compatible PC operating systems is directly evidenced by the sustained absence of realistic commercial alternatives to Microsoft's PC operating-system products.

54. 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.