7 PROTECTED MODE

The air-to-surface battle of 1991 demonstrated once again that among postmodern strategies of appearance, none is as effective as the simulation that there really is software. Until the proof to the contrary on the field of combat—where computers unambiguously revealed themselves as hardware for the destruction of Iraqi hardware (as durable goods are still called in everyday English)—advertising brochures and media conferences spread the fairytale of a development of software that would become increasingly more innocuous and user-friendly, more spiritual and intelligent, until one day in the not so distant future it would effectively lead to German idealism—that is, it would become human.

And that is why software, the billion-dollar enterprise out of one of the cheapest elements on Earth, makes every attempt to prevent the aforenamed humans from ever even coming into contact with the corresponding hardware. With Word 5.0 on a no-name AT 386 and (as one so nicely says) under the operating system Microsoft DOS 3.3, one can write entire essays on precisely these three entities without even suspecting the strategy of appearance. For one writes—the 'under' says it already—as a subject or underling of the Microsoft Corporation.

This worm's-eye view did not always prevail. In the good old days when microprocessor pins were still big enough for simple soldering irons, even literary critics could do whatever they wished with Intel's 8086 Processor. Even standard chips, which at that time still required one hundred thirty-three cycles for the multiplication of a single whole number, could be raised to the processing speed of primitive signal processors through a variety of strategies: not differentiating between RAM and ROM, misusing both of the stack

registers as universal registers, avoiding any interrupts, employing the wait-state for other than its intended purpose, and so on. The silicon chip, which was as stupid as the hobbyist and user, could accommodate all of this because the Von Neumann architecture recognizes no difference between commands and data. In order to get a program to run, the user had to first forget everything pertaining to mathematical elegance or a closed solution that still haunted his mind from his school-days. He even forgot his ten fingers, and translated all decimal numbers that would play a part in the program into monotonous columns of binary digits. As a result, he forgot the immediacy of the task as such and pored over data sheets in order to translate the commands IN, OUT, etc. (already formulated in English, of course), into their op-codes. This is an activity that only Alan Mathison Turing—when he finally had his universal discrete machine of 1936 at his technical disposition one World War later—is said to have preferred over all mnemonic aids and higher programming languages.[1] But once this expulsion of spirit and language was completed, the machine's stupidity equalled that of its user.

To be sure, this so-called machine language ran a million times faster than the pencil with which the user had pieced together the zeros and ones from Intel's data sheets. To be sure, the flip-flops whose infinitely repeated pattern covers the silicon chip took up a million times less space than on paper. But with that, the differences between computers and paper machines, as Turing had renamed humanity,[2] were also already fully accounted for.

Those good old times are gone forever. In the meantime, through the use of keywords like user-interface, user-friendliness or even data protection, the industry has damned humanity to remain human. Possible mutations of this humanity into paper machines are obstructed by multiple malicious tricks. First of all, Microsoft's user data sheets switched over to designating the Assembler grammalogue as the maximum demand or machine approximation that would be granted, meaning that op-codes would no longer be made public.[3] Secondly, the pertinent industry publications "assure us that even under the best circumstances, one would quickly go crazy from programming in machine language."[4] Thirdly, and lastly, these same publications already consider it criminal "to write a procedure for the calculation of sine in Assembler, of all things."[5]

At the risk of having already gone crazy long ago, the only thing one can deduce from all of this is that software has obviously gained in user-friendliness as it more closely approximates the cryptological ideal of the one-way function.[6] The higher and more effortless the programming languages, the more insurmountable the gap between those languages and a hardware that still continues to do all of the work. This is a trend that probably cannot be adequately explained either through technical advances or through the formalities of a theory of models, but rather, like all cryptology, has strategic functions. While on the one hand it remains possible in principle to write user-software or cryptograms with a knowledge of codes or algorithms, on the other and user-friendly concealed hand it is by now impossible to decipher the product specifications of the finished product or even to change these specifications. The users fall victim to a malicious mathematical trick that Hartley, one-time head of Bell Labs, is said to have already instituted while in the slump of old-age—the fact that one can no longer examine the operands of many of the operations.[7] The sum hides the addends, the product the factors, and so forth.

This mathematical trick is ideally suited to software. In an era that has long since abandoned the phantoms of creators or authors, but which through copyright passionately holds on to such historical ghosts for strong financial reasons, the trick becomes a goldmine. In any case, the subjects of the Microsoft Corporation did not simply fall from the sky, but first had to be produced like all of their media-historical predecessors—the readers of books, film audiences and TV viewers. The only problem now is how their subjugation can be hidden from the subjects in order that they fall in step with the global triumphal march.

In the domain of the politics of knowledge, the answer follows a proven recipe for the success of modern democracies, while in the technical arena it has changed the hardware of the microprocessors themselves. As concerns the politics of knowledge, perhaps only Siemens' engineers can tell it like it is, as Klaus-Dieter Thies did in the *80186-Handbuch:*

> Today's modern 16-bit micro-processors assume in increasing amounts tasks that are assigned to classic mini-computers in the typical application range. Thus in multi-user systems, it is necessary that the programs and data of individual users are isolated, just as the operating

> system must be protected against user software. In order to give every individual user the possibility of implementing his software independently of the number of other users, and in order to give him the impression that the computer is only there for him, it is essential to allocate the CPU to the individual programs through multi-tasking, which, however, can only remain hidden from the user if the CPU is extremely powerful.[8]

Thus, according to this Siemens version—which also circulates at I I?M-Germany—Intel did not propel the operating-frequencies of ihe 80286 and the 80386 to levels between 12 and 33 megahertz in order that they correspond to the demands of professional users or even to the Pentagon's specifications for electronic warfare,[9] but rather in order to entangle civilian users in an opaque simulation. Multitasking should then, like the hedgehog in the fairy tale, delude the user into believing that only a single hedgehog or process is running, and, above all, that this race or process only benefits a single rabbit or user.* This is the same tune by which the novels and poetry of Goethe's time promised their male, and before that their female, readers that the texts were uniquely addressed to them; it is the same tune by which modern politics presents itself to the public as its absolute antithesis—that is, individuals.

In contrast with traditional simulations which all had an absolute limit in the power or impotence of everyday language, today's electronic simulation—according to which every microprocessor should only be there for a single user—also employs hardware. From the 80286 on, Intel's processors are equipped with a Protected Mode

* [Tr.—The story of the Hare and the Hedgehog *(Der Hase und Der Iget)* is taken from Grimm's fairy tales. A hare and a hedgehog argue who is the fastest runner and decide to wager a piece of gold and a bottle of brandy over a foot race. The hedgehog secretly tells his wife to hide in a furrow at the end of the field which is to serve as the race track. The hare and the male hedgehog start the race together, but just as the hare is approaching the finish line, the hedgehog's wife pops up and calls out, "I'm already here!" The hare insists on a rematch in the other direction. This time the female hedgehog begins the race with him, and as the hare approaches the finish line, her husband pops up and calls out, "I'm already here!" The hare demands still more races to prove his speed, only to be beaten each time by the hedgehog(s). The hare finally drops dead in the middle of the seventy-fourth race, and the hedgehog and his wife take their prizes and return home. *(The Grimms' German Folk Tales,* "The Hare and the Hedgehog," trans. Francis P. Magoun, Jr. and Alexander H. Krappe (Carbondale: Southern Illinois UP, 1960) 604-7.)]

that (in the words of the Siemens engineer) protects the opcnilini system from the users, and through this protection, first allows I 111 users to be deceived. What began as the simple capability ol swiielw ing between the supervisor and the user stack in Moioioln'il 68000[10]—naturally, a secret rival system—is extended to system* wide procedure in the separation of Real Mode and Protected Mode, Different command sets, different address possibilities, different rcg-' ister sets, even different command execution times, henceforth scpu •• rate the wheat from the chaff, the system design from the users. I 'hut it is precisely in the silicon on which the prophets based all tlieu hopes for a microprocessed democracy of the future that the eleinen tary dichotomy of modern media technologies again returns. German civil radio network, for example, was permitted from the <l:iy when the postal service of the Reich could credibly promise the armed forces that the consumer radios of 1923, which were capable of any possible transmission, would never be able to disrupt military - industrial radio communication because an automatic encoding machine—which Turing's proto-computer would only put out of action in World War II—had just been invented.

The innovation of Intel's Protected Mode consists only in having transferred this logic from the military-industrial realm into that of information itself. The distinction between the two operating levels is not only quantitative—as for example among the varying ranges of operating temperatures for commercial, industrial and military silicon chips (in this indicative ranking)—but more importantly, qualitative because the CPU itself works with priorities, prohibitions, privileges and handicaps of which it constantly keeps a record, though only in Protected Mode of course. That such controls, which themselves require time, are not exactly conducive to the general goal of increasing data output goes without saying. In Protected Mode, the same interrupt requires up to eight times more cycles than in Real Mode, but evidently a high technology can only be passed on to end users and "non-trustworthy" programs (as Intel calls them) under these circumstances, even if and precisely because the signal processing, the military-industrial dimension of computers,[11] is slowed down by bureaucratic data processing. Although there may no longer be any written prohibitory signs that guarantee a power gap, the binary system itself encodes the distinction between

commands and data, what the system permits and what, conversely, is prohibited to user programs. John von Neumann's classic architecture, which made absolutely no distinction between commands and data—unnecessary in an era when all existing computers were still state-secrets—disappears under four consecutively numbered privilege levels. It is for good ironical reason that the only incorruptible German publication in the area of computer applications wrote: "Even if all the talk is of privileges, higher privileged code segments, privilege violations and the like, you are not reading the political manifesto of a former official of the SED,* but rather the explanation of the security concept of the 80386!"[12]

Political manifestos, as the name already indicates, were performed in the dominant sphere of everyday language; for that reason, the privileges with which they took offense are, at least since November, 1989, meaningless. On the other hand, the privilege levels of Intel's so-called 'flagship'—this Cocom-List** transferred to the very heart of the binary system—contributed, even more so than the innundation of Eastern Europe with television, only to the liquidation of politically based privileges. A short essay by Carl Schmitt, *Gesprach iiber die Macht und den Zugang zum Machthaber [A Conversation on Power and Access to the Dictator],* culminated in the thesis that power is reducible to its conditions of access: the antechamber, the office or, more recently, the front office consisting of typewriter, telephone and secretary.[13] With this authority and through this authority, conversations could in fact still be conducted; however, technically implemented privilege levels draw their power precisely from mute efficacy. In order to take advantage of his memory reserves beyond DOS in a sort of posthistorical metaphysics, the 80386-user installs one of the available user-friendly utilities, loads the debugger with a built-in program that still ran yesterday without any problem, and discovers that the new

* [Tr.—Sozialistische Einheitspartei Deutschlands: United Socialist Party of Germany; political party of the former GDR.]

** [Tr.—'Cocom' is the abbreviation for the Coordinating Committee for East-West-Trade-Policy. The 'Cocom-List' is an embargo list drawn up by the majority of NATO members and Japan, which specifies those goods and technologies which because of their strategic military sensitivity are prohibited export to East-block countries.]

installation not only manages memory as promised, but concurrently, and without any warning, has locked all privileged commands.[14] As Mick Jagger already so pointedly formulated it, instead of what he wants, the user always only gets what he needs (according to the industry standard, that is).

As a result, a double shadows the analysis of power systems, thai immense assignment that was Foucault's *legacy*. To begin with, one should attempt to abandon the usual practice of conceiving of power as a function of so-called society, and, conversely, attempt to construct sociology from the chip's architectures. For the present at least, it is a reasonable assumption to analyze the privilege levels of a microprocessor as the reality of precisely that bureaucracy that ordered its design and called for its mass application.[15] It is no coincidence that the separation between supervisor level and user level at Motorola, and Protected Mode and Real Mode at Intel, came about in the same years when the United States of America embarked on the construction of an impermeable two-class system. (One recognizes the Embedded Controller in every improved hotel door-lock in New York.) It is no coincidence that in the 80386, it is precisely the input and output commands that are protected by the highest privilege level—in an empire in which the public views the rest of the world only through the haze of television news, even the thought of foreign policy is a privilege of the government. This is probably also the reason why the latest varieties of systems theory simply deny, at the highest theoretical level, the finding that information systems control input and output. When all is said and done, this would also be a good reason for computer scientists from other countries—that is, somewhere between Japan and Europe—to oppose the US bureaucracy submerged in silicon with other possible bureaucracies. Whether there are better ones is beside the point because they would in any case also have to be bureaucracies; but a competition between different systems and different bureaucracies would as such already allow the subjects of MS-DOS to breathe a little easier.

Of course, as long as the celebration of the triumph of IBM-compatibility continues, the demand is for strategy, more so than sociology. With its move out of front offices and everyday language into the micrometer realm, power has also changed the processes

and the working surfaces. The unequivocal 'no' of an access denied is no longer a given in binary code, simply because the entire hierarchical standard of self-similar program levels—from the highest programming language down to elementary machine code[16]—rests completely flat on the material. In silicon itself there can be, to borrow from Lacan, no other of the other,[17] which is also to say, no protection from the protection. Furthermore, the hidden segment registers that keep a record of all access rights of all programs of a system, must be accessible in order to work. Legible traces even remain when the CPU sets these registers to zero[18] in the case of privilege violations that occur despite all possible and explicit commands. At the level of the machine, then, protection mechanisms have no absolutely protected hiding-place. Because microprocessors must despite everything remain usable to users, that is, communicate with them, Intel's Protected Mode describes a classic power dilemma.

According to the *Programmer's Reference Manual,* even tasks of the operating system are not permitted to enjoy the privilege of freely accessing tasks of a lower privilege level. Because the traffic over the stack runs symmetrically or on a democratic basis—that is, the caller must PUSH the same number of bytes as the program that has been called POPS—the lower privileged task could attempt to maintain control after its completion, and smuggle itself up to the level of the higher task through a technically simulated program return. For that reason, the Intel engineers considered it safer to employ the Boolean concept of gates and to convert to a bureaucratic control of access.

What such prohibitions conclusively demonstrate, however, is only the impossibility of perfect access control. In the good old microprocessor days when the difference between system and application itself resided in the silicon into which it was literally burned—the system in ROM, the application in RAM—there was nothing to assail it. Once the difference has been rendered programmable, however, it is already vulnerable to all sorts of circumventions.

Intel's *Programmer's Reference Manual* repeats approximately 170 times—at each individual 80386 command—the warning that Interrupt 13 is triggered in Real Mode as soon as any part of the command operator exceeds the actual 20-bit address field. In other

words (but still those of the company itself), in Real Mode the 80386 would only run as a faster AT.[19] In the case of contraventions, a draconian sentence is invoked—"all violations of privilege that do not cause a more specific exception" trigger a monstrosity with the name of "General Protection Exception."[20] But neither the 170 repetitions in the manual nor their countless transcriptions in a computer book market—which for its part seems to offer only partial mechanical translations under the names of fictitious authors—make this warning any more true. A single subordinate clause in the manual discloses that any address boundaries in Real Mode are no more and no less than presuppositions programmed into the system start-up. This sentence disappears, of course, from all translations, summaries, popularizations and user handbooks, simply to conceal from the subjects of Microsoft its logical reversal—namely, that presuppositions are easily changed.[21] Instead of the deliberately low default value that the CPU automatically loads into the hidden sections of its segment register at every reversion to Real Mode, programs could also set completely different values. Every 386-AT goes into all four of the possible operating modes with 100 lines of code—into Protected Modes with 32- or 16-bit segment width, but also into Real Modes with the corresponding segment width. As a result, the Real Mode with 3 2-bit segments could produce the most compact and thereby fastest code by far, although there is no mention in the data sheets and manuals of its being even a possibility,[22] to say nothing of real existing operating systems of the 80386.

One hundred lines of Assembler, but only of Assembler, solve the problem of a postmodern metaphysics. At the risk of going crazy, they lead through MS-DOS beyond MS-DOS. Along with the infamous sound barrier at which the operating memory in DOS remains limited to a ridiculous mega-byte, all of the advantages for which Windows is praised dwindle to nothing. In a drastic paradox, it is precisely the most antiquated of all operating systems that provides the trap door out of the operating system. Intel's built-in blockades—which engage immediately in more complex operating systems such as UNIX, and subsequently even pick out those hundred program lines as illegal commands and refuse them—are powerless against stupidity.

Thus a machine can simultaneously do less and more than its data sheets admit. It is not only this aforesaid trick that expands the addressable storage capacity from one mega-byte to four giga-bytes—a 4,000 percent increase—but in addition, the 80386 has at least two "undocumented commands" that the data sheet intentionally keeps secret,[23] and in the 32-bit Real Mode, at least one operation that it disregards without any intention whatsoever. Such chaos does not reign at the elevated level of information science, where the computability of Finite State Machines and their ability to predict is argued over in general, but rather at the modest level of the engineer's empiricism. Only because, as Morgenstern said, "nothing can be that is not allowed to be," mere presuppositions are sold to the users as absolutes. In similar fashion, the Reichspost at one time was careful to insure that only detector equipment and no tube equipment would be sold to civilian radio consumers of the early '20s—if this were not the case, the listeners would have been able to transmit and thereby disrupt military-industrial radio traffic.

In other words, information science appears to be confronted with internal information obstacles. Information science must refer to the actual domain of code, even if the theory could generate completely different models (and should). And despite the will and belief of the code's developers, decodings are just as possible as they are rare. Long after the end of the print monopoly and authorship, the phantom of humanity apparently makes sure that mere opinions or even assertions of protection will continue to be recorded, as opposed to actually cracking the codes. A systems program must be created precisely to this end—to be used by programmers, to begin with, but in principle for machines as well. Just as it is possible, and in the meantime also realizable, for programs generated by chance to compete with each other according to purely Darwinian rules, the empirical control characteristics of the machines would first have to be deciphered and then contrasted with their data sheets.

At least to the literary critic, it appears that this, as it were, military-strategic field of information science has a big future before it. Specifically, it could proceed on a strictly technical plane according to methods similar to those which Foucault's discourse analysis

proposed for utterances and texts. Rather than investigating the meaning of a sign-chain as interpretation or investigating the rules of a sign-chain as grammar, discourse analysis is quite simply concerned with sign-chains inasfar as they exist and do not, on the contrary, not exist. Whether meanings are merely a pedagogical-philosophical fiction, and whether grammar rules comprehend completely and arc completely comprehensible, is beside the point. But that the two words 'grammar' and 'rule' are connected in a single verbal expression is and remains a fact.

Johannes Lohmann, the renowned language scholar and Indo-Germanist, already proposed thirty years ago that one look for the historical condition of possibility of programming languages in the simple fact that they exist in English, and furthermore that they exist only in English verbs such as 'Read' and 'Write,' that is, verbs which in distinction to the Latin *amo amas amat* and so forth have shed all conjugation forms. According to Lohmann, these context-free word blocks may well stem from the historically unique confusion of Norman and Saxon in old England, but that is no hindrance to their being translated into context-free mnemonics and ultimately into computer op-code. As everyone knows, the endless litany of 'read' and 'write,' 'move' and 'load' is called Assembler.

A discourse analysis whose elements are obviously not only words but also codes, would, of course, level the sacred distinction between everyday languages and formal languages. In light of the wonderful "orthogonality" that, for example, Motorola's processor series flaunts since the 68000, that would be heresy. The history of Protected Mode as a half-compatible, half-incompatible extrapolation of good old standards could, however, teach us that codes are subject to the same opacity as everyday languages. As everyone knows, in the 8086 there were more than a few commands that were synonymous with other commands and that were only obviated through operating speed. It made a significant temporal difference whether a universal register or an accumulator wrote its contents into memory. However, Intel's new generation de-optimized precisely this speed advantage, while still permitting the synonymous commands to survive for compatibility reasons. Thus the code has achieved a redundancy that everyday language already boasted in Frege's wonderful example of "evening star" and "morning star."

As history has shown, this redundancy can only increase if machine codes are to remain compatible over the generations. In contrast to everyday languages—and especially to German where there are neither (to deliver concurrently two autonomous examples) restrictions on the length of a word [Wortlangenbegrenzungen] nor restrictions on the length of word combinations [Wortkombinationslangenbegrenzungen]—all elements of a command set are of a finite length and for that reason also of a countable quantity. As a result, there would be no more room for the extended commands of the 80386, for example, if there were no authorization for over-length. Under these circumstances, the codes begin to grow wild, no matter how economical or orthogonal their first design may have been. The silver chip-surface, concurrently the model and the main field of application of all topological optimizations, loses its mathematical transparence—it becomes a Babylonian tower in which the ruins of towers that have already been demolished remain built-in. Protected Mode as both the enemy and co-existent partner of a Real Mode that has already been superceded technically for some time is computer history *on chip.* And David Hilbert's dreamlike program to clear out the opacity of everyday language once and for all through formalization is undone not only at the clear, axiomatic level of Godel or Turing, but already by the empiricism of the engineers. Codes with compatibility problems begin to grow wild and to adopt the same opacity of everyday languages that have made people their subjects for thousands of years. The wonderful term source code becomes literal truth.

To be sure, a discourse analysis can neither tame nor debug such wild growth. But it is quite possibly more efficient simply to count on its happening. Turing's old idea of allowing the machines themselves to roll out their code may well have already secretly come true. Precisely because "the complex function of highly integrated circuits (aside from memory-ICs) can no longer, as in the case of a simple, logical connection, be checked by testing all of the possible input signal combinations,"[24] tests that are independent of the producer are in order. Resistances—recently raised to a system by US patent law—should not prevent the publication of unedited test results, patches and detour techniques, of which there is no comment in the official documentation. That would be information about information science, whether for peace or not.

Hugo von Hofmannsthal once ascribed the ability to read "whai has never been written" to the "wonderful being" called Man. Similar crypto-analyses must become universal and mechanical in the chaos of codes that begins with the world-historical dismissal of everyday language in favor of a universal discrete machine.

Translated by Stefanie Harris

Protected Mode

1. Cf. Andrew Hodges, *Alan Turing: The Enigma* (New York: Simon and Schuster, 1983), 399.
2. Cf. Alan M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society* (42: Jan. 1937), 230-265.
3. Cf. Microsoft Corporation, *Macro Assembler 5.1, Reference* (1987)115: "This section provides an alphabetic reference to instructions of the 8087, 80287, and 80387 coprocessors. The format is the same as for the processor instructions except that encodings are not provided."
4. TOOL Praxis, *Assembler-Programming aufdem PC. Ausgabe I* (1989: Wiirzburg), 9.
5. TOOL Praxis, 39.
6. Regarding one-way functions in mathematics and cryptology, cf. Patrick Horster, *Kryptologie: eine Anwendung der Zahlentheorie und Kmnplexitdtatheorie* (Mannheim-Wien-Ziirich: 1982/1985), 23-27.
7. Personal communication of Hartley to Friedrich-Wilhelm Hagemeyer, Berlin.
8. Klaus-Dieter Thies, *Das 80186-Handbucb* (Diisseldorf-Berkeley-Paris: 1986), 319.
9. For details of the Pentagon specification pamphlet, cf. D. Curtis Schleher, *Introduction to Electronic Warfare* (Norwood/MA: 1986).

10. Cf. Gerry Kane, *68000 Microprocessor Book* (Berkeley: 1981), 8ff.
11. Cf. Michael Lowe, "VHSIC: Ultraschnelle Schaltkreise frisch vom Band ins Pentagon," *Militarisierte Informatik,* eds. Joachim Bickenbach, Reinhard **Keil**-Slawik, Michael Lowe and Rudolf Wilhelm (Miinster: 1985), 64.
12. Harald Albrecht, *MSDOS in a box. Teil 1* (1990), 258.
13. Cf. Carl Schmitt, *Gesprach iiber die Macht und den Zugang zum Machthaber* (Pfullingen: 1954).
14. This occurred with a QEMM 386 from Quarterdeck during an attempt to execute the commands LGDT and SGDT within CODEVIEW.
15. For evidence from an authorized source, namely the Intel Corporation, cf. Sharad Gandhi, "Die Intel Architektur & RISC," in *Pentium-Prozessor. Die niichste Generation der Intel Architecktur,* eds. Johann Wiesbock, Bernhard Wopperer, Gerold Wurthmann (Haar: 1993), 80: "The primary benefit of a faster CPU, which the RISC-Processors previously had, will in time become more of a secondary quality. Let me put forward an analogy. The core of the CPU behaves like an entrepeneur who starts a new, innovative company. If the company is small, it flourishes according to the ideas and inspirations of the entrepeneur. With increasing size, the company's success will be increasingly determined through peripheral functions such as sales, marketing, publicity, finances, accounting, and so on, functions which usually have no connection to the company's product. Large companies compete with each other on the basis of their peripheral functions. In the future, the performance share of the (mathematical) core will continue to decrease." This is a strangely ambiguous estimation of success which would also permit totally different endings and thoughts on still newer innovative companies. If on the chip surface of the Pentium-Processor, only 8% in calculating performance is opposed to 92% in self-representation of the company (Ghandi 79), then high-technology comes dangerously close to the ratio between brain and body weight in dinosaurs.
16. Cf. Arne Schapers, *Turbo Pascal 5.0* (Bonn-Reading/MA-Menlo Park/CA-New York-Don Mills-Wokingham-Amsterdam-Sydney-Singapore-Tokyo-Madrid-San Juan: 1989), 1: " 'The processor's address space is organized through the architecture of the system into areas of RAM and ROM, whereby the RAM becomes subdivided by the operating system into storage blocks for the recording of programs. One program consists of individual segments, a portion of which contain procedures and functions which for their part can locally define further routines.' One is almost reminded of a painting by Escher (or even more to the spirit of the age, by Apfelmannchen)."
17. Cf. Jacques Lacan, *Ecrits,* trans. Alan Sheridan (New York: W.W. Norton & Company, 1977), 316.
18. Cf. Intel Corporation, *80386 Programmer's Reference Manual,* Chap. 17 (Santa Clara/CA: 1986), 145: "The DS, ES, FS, and GS segment registers can be set to 0 by the RET instruction during an interlevel transfer. If these registers refer to segments that cannot be used by the new privilege level, they are set to 0 to prevent unauthorized access from the new privilege level." In the code segment GS, such sledge-hammer protection tactics are eliminated in order not to block the entire system.
19. Cf. Intel Corporation, Chap. 14, 1.
20. Intel Corporation, Chap. 9, 16.

21. Cf. Harald Albrecht, *Grenzenlos. Vier Gigabyte im Real Mode der 80386 adressieren,* Vol. 1 (1990), 212: "In the 80386, the segment boundary of 64 K-Bytes is not as firmly set as it appears, for example, in Intel's documentation over the 386 DX. If one persistently follows the necessary steps for the return of the 803 86 out of Protected Mode into Real Mode, then suddenly the entire address space of the 4 G-Bytes of Real Mode is opened up (whereby the smirk of the Motorola fans noticeably diminishes in width)." The following is with extensive thanks to this really inspired suggestion.

22. One exception is made, though tellingly without any commentary, in Klaus-Dieter Thies, *PC XT AT Numerik Buch. Hochgenaue Gleitpiinkt-Arithmetik mit 8081. .. 80281.. . 80387 . . . Nutzungmathematischer Bibliotheksfunktionen in Assembler' und 'C,'* " (Miinchen: 1989), 638. Edmund Strauss, on the other hand, though he (according to the Preface of the 80386 architect, Robert Childs) "has seen the full range of system issues and devised many practical solutions during his work for Intel," completes the work of art by keeping silent about the non-documented room for play, over the course of an entire authoritative handbook. Cf. Edmund Strauss, *80386 Technical Reference. The guide for getting the most from Intel's 80386* (New York: 1987).

23. Cf. Andrew Stiller, *Bitter fur 32-Bitter,* Heft 8 (1990) 202. For details on the command LOAD ALL (including the dubious assertion that only the 802 86 accepts it), cf. Norbert Juffa and Peter Siering, *Wege iiber die Mauer. Loadall—Extended Memory in Real Mode des 80286,* Heft 11 (1990), 362-6.

24. Lowe, 70.