Editorial Note: Computers have become an essential tool for physicists. The editor hopes that this article, which was written by a physicist, will be useful for a wide range of readers of our journal.

Computer viruses and methods of combatting them

G. L. Landsberg

Institute of High Energy Physics, Protvino, Moscow Region

(Submitted 3 October 1990)
Usp. Fiz. Nauk **161**, 161–191 (February 1991)

This article examines the current virus situation for personal computers and time-sharing computers. Basic methods of combatting viruses are presented. Specific recommendations are given to eliminate the most widespread viruses. A short description is given of a universal antiviral system, PHENIX, which has been developed.

PREAMBLE

"Computer virus" is a term that, due to the zeal of journalists, is known even by people who are far from computers. An atmosphere of mystery and superstitious terror surrounds these artificially created programs intended to destroy data on magnetic disks, to slow computers, and to interfere with the actions of users. How terrifying and mysterious are viruses? How do they work? What type of arsenal of means to combat them does "computer medicine" offer?

This survey answers these and other questions. It is mainly devoted to viruses on IBM PC compatible computers with the common MS DOS operating system,[1] although there is also information on viruses on other personal computers, and on large computers. The survey is intended for the "rank and file" IBM PC user as well as system programmers. For people who are not well acquainted with the structure of the MS DOS operating system, all information needed to understand this article is presented.

The introduction contains an explanation of some terms which are encountered in the text, as well as a brief description of the operating principles of MS DOS (to the extend they are needed to explain the operating mechanisms of the viruses). In the first section we examine the "anatomy" of virus programs, classify them, and describe the main ways in which they spread. The second section is devoted to various antiviral programs. The third section examines the most widespread viruses in the USSR and gives recommendations on how to "cure" them. The fourth section contains a brief description of the PHENIX universal antivirus system. A separate preprint is devoted to PHENIX.[2]

This survey is an attempt to bring together and to critically analyze the abundant and frequently contradictory information contained in various publications on this subject (articles in specialized and popular journals, preprints, books, descriptions of antiviral programs, etc.).

One should bear in mind that computer virology is a relatively new area of knowledge. It actually developed in 1988, and in the USSR in 1989. At present, standing seminars on viruses are being organized in Kiev (directed by N. N. Bezrukov), and the most interesting results are published in the bulletin Softpanorama. Other seminars are being organized in Moscow as well (directed by L. G. Bunich).

After the first extensive article in the USSR on viruses by A. A. Chizhov,[3] several rather detailed papers appeared, among them the extensive preprint of N. N. Bezrukov[4] and a number of articles by I. Sh. Karasik in the journals Interkomp'yuter[5,6] and Mir PK [PC World (in Russian)].[7] There are numerous foreign publications devoted to viruses. Many of them are special communications in computer journals (for example, Refs. 8–14); however, there are also books which contain a survey of the virus situation.[15,16] Unfortunately, the number of new types of viruses and "strains" of known virus programs is constantly growing; thus, information in such surveys becomes rapidly outdated. In this article attention is focused on the operating principles of viruses and antiviruses, and specific types are examined only as illustrations (except for section 3, which is devoted to the current virus situation in the USSR).

INTRODUCTION

Let us begin with terminology.

Computer virus—an artificially created program which has the capability of covert reproduction in the operating environment of the computer. It reproduces by inserting into executable code (loadable programs, elements of the operating system, compiled text, command files) its own code, possibly a modified copy which retains the ability to reproduce. Usually a virus creates some disruption in the work of the user, destroys data on magnetic media or destroys elements of computer hardware (disk drives, monitors). This is a modification of the definition given by F. Cohen in his article,[17] which is one of the first serious publications on the mathematical study of viruses.

Antivirus—a program intended to locate and destroy a virus on a specific computer. Good antiviruses are capable of eliminating some disruptions created by viruses; for example, an antivirus can restore infected files.

Command interpreter—the part of the operating system that interfaces with the user. In the MS DOS system the standard interpreter is COMMAND.COM; however, there

0038-5670/91/020185-16$01.00

are also other interpreters which are compatible with this operating system for example, 4DOS.COM by J. P. Software).

Protection of files and the core of the operating system—special measures taken to prevent accidental erasure of files, as well as to prevent change of the contents of memory occupied by the operating system. On large computers, much attention is devoted to protection; usually it is very complex and has several levels. The MS DOS operating system was conceptually designed for one user, and has no memory defense at all. There are primitive means of protecting the file system (the so-called Read Only attribute, which does not permit a file to be opened for writing or erasure), however, they are very simple and serve more as a precaution, and not as defense since any program can change this attribute, after which the file can be modified at one's discretion.

Interrupt—a special situation which arises in the work of the computer in which the control is transferred by the so-called interrupt processing program. Hardware interrupts are intended to service the input-output devices, the timer, etc. On a specific computer they are independent of the operating system, because they are implemented through hardware. Usually programs which process these interrupts are contained in ROM (read only memory). On IBM PC computers they are a part of BIOS (Basic Input-Output System). There are also software interrupts which are supported by any operating system. They are actually an interface between the user program and the system. The addresses of entry points in the interrupt processing program on the IBM PC are stored in the low addresses of the RAM (the table of interrupt vectors).

Boot sector, or boot record—the zero sector of the logical disk (partition on the hard disk) or diskettes intended to store the code for initial loading of the operating system. The structure of this sector is described in detail in Refs. 1 and 6.

Master boot sector, or Master boot record, MBR—the first absolute sector of a hard disk on IBM PC compatible computers. This sector contains information about the division of the hard disk into partitions (partitions table), information about which partition is the boot partition (that is, which partition contains the operating system), and the program code, which places the contents of the boot sector of this partition into the RAM and transfers control to the beginning of programs located there. The structure of the master boot sector is examined in detail in Refs. 1 and 6.

Now let us touch upon the operating principles of MS DOS on IBM PC compatible computers. When the computer is turned on, control is transferred to the special program POST (Power On Self Test) which is located in ROM and tests memory, controllers, and other electronic elements. When the tests are successfully completed an attempt is made to load the operating system from the floppy disk in the zero disk drive (logical name A:). When there is no diskette in the drive, control is transferred to the program residing in the main boot sector, which, according to the partitions table, in turn transfers control to the program in the zero sector of the load partition. This boot code loads the operating system itself (which also occurs in several stages, but a description of these stages is beyond the framework of this article). The command interpreter COMMAND.COM, to which control is transferred if loading of the operating system is successful, is not completely loaded into the RAM. There is always only a small part of it in the memory (about 4 kilobytes). To carry out some DOS commands the interpreter is loaded from the disk.

It remains to describe briefly the format of the executable modules of the MS DOS system. The Intel 8086/88 and 80286 microprocessors (which are the basic microprocessors for the IBM PC/XT and IBM PC/AT respectively) have a 16-bit word. The command system of these processors makes it possible to use a 16-bit address (operators and near type operands) which corresponds to 64 kilobytes of memory or one segment. However, it is possible to use a 20-bit address given by a pair of 16-bit words segment and offset.[1)] Here the operators and operands are of the far type, and the volume of addressable memory rises to 1 megabyte. The Intel 80386 microprocessor, which is the basis of the IBM PC/AT-386 or the IBM PC/SuperAT, has a 32-bit address; however, MS DOS uses its capability to emulate a 16-bit work mode; thus, from the point of view of this operating system, all four processors are identical and differ only in speed.[2)]

MS DOS supports two types of executable modules in this microprocessor architecture. One, so-called COM programs (files which contain their code usually have the extension .COM), should have a size of no greater than 64 kilobytes, and thus, usually consist only of near type commands. The means of loading them into memory is extremely simple: the text of the file is located in some segment beginning with the address 100h.[3)] Programs whose length exceeds 64 kilobytes are constructed somewhat differently. They may contain any commands and operators (both near and far) and consist of several relocatable modules. These files usually have the extension .EXE and have a standard heading at the beginning in which the following are found: a table of related modules, the initial values of the counter of the IP command and the stack indicator SP, as well as other maintenance information. When this type of file is loaded (the first two bytes contain the symbols 'MZ' (5Dh, 4Ah), the so-called signature of an EXE module) the operating system adds the appropriate shifts to all far addresses of the related modules, and stores them in a table. Afterwards, the initial values of the SS, SP, CS, and IP registers is set, and control is transferred to the entry point of the executable program. The loading and execution of EXE and COM modules occurs when DOS interrupt 21h is called (this is the main internal interrupt of MS DOS), or more accurately, its subfunction 4Bh. The structure of the heading of an EXE file is described in more detail in Ref. 1.

Before it finishes its work, the program calls interrupt 20h, which resets the value of the vectors of some interrupts (they may be changed while the program is in operation), frees the RAM that the program occupied, and then transfers control to the program (parent program) which requested the execution of the module which just completed its function (usually this is the command interpreter, but it may also be another program, for example, Norton Commander, PCShell, etc.). Another possible means of finishing the work of the program is to leave the body of the program resident in memory (Terminate but Stay Resident, TSR), which is done when interrupt 27h is called. In this case the operating system reserves part of the memory occupied by the module which is being executed, and does not free it when control is transferred to the parent program. This completion mechanism makes it possible to implement interrupt processing procedures which differ from standard DOS and BIOS procedures. It also makes it possible to create a program which becomes active when certain conditions are met (some combination of keys is pressed, at a particular time of day, etc.).

This is all that must be known about the MS DOS operating system to understand the operating mechanism of viruses.

1. What is a virus? This section is completely dedicated to various types of viruses. It contains some historical information, a description of the mechanisms by which viruses are spread and activated, typical manifestations of their activity, indications that computers are infected, and attempts to classify virus programs.

1.1. Some history

"...*It was concealed in this very way...from*
what far bounds did the materialism of our
being draw its causality..."
James Joyce, Ulysses

The history of the appearance of viruses is very intricate. There is no official version of their origin, and among historiographers there is a conflict about setting the moment of creation of the first virus programs back further into the past. With this in mind it is reasonable to avoid ambiguous assertions and dwell only on the basic historical landmarks preceding the appearance of viruses.

The possibility of viruses existing was first theoretically proven by the noted American mathematician John von Neumann in 1949 in his *Theory and Organization of Complicated Automata.* He showed that a sufficiently complex automaton could have the capacity for reproduction. Attempts to practically implement self-reproducing automata appeared much later.

In 1959 an article by L. S. Penrose appeared in the journal Scientific American on self-reproducing mechanisms. Based on this article F. G. Stahl created a program for the IBM 650 which modeled a fight for survival among beings that "devoured" nonzero words in machine memory, including each other. After "eating" a certain amount of words, the organism generated an organism like itself (with the possibility of mutations). If during sufficiently long "walks" the being did not "eat" a nonzero word, it would "die of hunger." However, the low speed and small volume of operating memory of the computer did not allow one to obtain interesting results. Details of this program can be found in Ref. 18.

In approximately 1962 an employee of AT&T Bell Laboratories, V. A. Vysotsky, invented the game "Darwin," in which the program-organisms battled for living space in the operating memory of the computer. Here one can trace a link between the virus theme and a biological battle for existence. Articles about this game appeared only ten years later in the journal Software: Practice and Experience. A development of the Darwin idea was the popular game "Core War," which is described in Ref. 19.

In 1982, employees of XEROX created a program called "Worm"[20] which was capable of "creeping" into other computers (joined in a network) with the goal of optimal use of machine resources. It was assumed that at night this program could run on a large number of computers at once, and during the day, when the computers were loaded, it would occupy only the base machine, so as not to disturb the other users.

In 1983, Ken Tompson received an award from the Association for Computer Machinery for his demonstration of a virus code. At the time it was yet not known how much damage the viruses would cause in a few years.

Since the end of 1987 viruses have become widespread on IBM PC computers. By 1990 there were more than 70 of them, and there is a steady trend for growth. The first viruses on the IBM PC compatible computers were Lehigh (detected at Lehigh University in the US) and Jerusalem (detected at Jerusalem University, Israel), of which we speak below. Information can also be found in Refs. 4, 6, 8, and 9.

The combination of the function of reproduction with infection was even observed in the first MS DOS viruses. It must be said that this is not surprising. The point is that even before the appearance of viruses there were programs which were designed to destroy data. The simplest of these (so-called "logic bombs") were created as a method of revenge. For example, an employee who developed an accounting system for a bank installed a check on the presence of his name in the payroll. If his name was absent (he had been fired) some data damage would occur, and repair of this damage could be very costly.

Later, so-called "Trojan programs" appeared, which usually mask their harmful activity with some useful functions. For example, a program which finds bad clusters on disks could actually be slowly creating them; programs which optimize a disk could lead to a gradual loss of files, etc. A detailed list of "Trojan" programs for IBM PC compatible computers known at the time of publication can be found in the August 1988 edition of PC World magazine. More detailed information on the "Trojan War" can be found in Ref. 21.

However, only with the appearance of viruses did individual cases of data damage become widespread. Once a virus has been let loose it can not be finally destroyed, since it is stored and appears again and again from archives, virus "collections" etc. According to some hypotheses the first MS DOS viruses were intended as punishment for illegal copying of software. But, as practice has shown, it is not the direct culprits, but third parties who are harmed. With the appearance and spread of computer networks even users who are not guilty may be harmed.

The problem of protection from computer viruses has acquired such importance that many countries have passed a number of laws which make it illegal to knowingly create and distribute viruses. In the US, bills 55 (Virus Eradication Act) and 287 (Computer Protection Act) were introduced for examination by Congress.[22] These bills impose large fines or prison sentences for a period of up to 15 years for premeditated damage of software. Many computer firms and personal computer user groups are turning to their governments to establish a high priority for research on protection from viruses.[23]

Fortunately for users there is an empirical rule noted in many publications (for example, in Ref. 4). According to this rule, the more literate and elegant the virus program, the less damage it usually causes. Usually, the effect of such viruses is reduced to cunning visual or audio effects ("crumbling" of letters on the screen, the playing of melodies, etc.).

However, even the most inoffensive virus may be easily modified even by a very unskilled programmer so that it will cause destruction. By the way, the largest number of "strains" of known viruses arose in this very way.

Usually, viruses are transmitted by computer games and ...antiviruses. Games are the most frequently illegally exchanged software products, and everyone uses them, even users who do not know how to determine the presence of a virus before the onset of the irreversible consequences of its action.[4)] Antiviruses are designed for work in dangerous proximity to viruses, and so they are frequently infected.

We should again stress the similarity between computer viruses and biological viruses:

—both are capable of multiplying and mutating;

—man is the carrier of both types of viruses;

—a timely appearance of an antidote does not occur for all types of viruses;

—it is impossible to create a universal medicine for all viruses, although one is sooner or later created for each specific virus.

If decisive measures are not taken to fight computer viruses, soon a real fight for existence may begin, not just between individual programs in computer memory, but between *viruses and personal computer users.* To win this fight, one must know "the face of the enemy." To know this, one should read the following section.

1.2. Viruses under the microscope

Before we turn to an examination of viruses for IBM PC compatible computers, and there is no doubt that these viruses exist, let us first talk a little about viruses on other computers.

1.2.1. *Do viruses exist on large computers?*

"To this question we answer positively: Yes, there is no God!"

A. Zinov'ev

On large computers, that is, on computers with time-sharing and multi-user operating systems, true viruses appear extremely rarely today. Many publications on this subject in newspapers and popular journals are usually "exaggerated" sensationalism and evidence of the incompetence of individual journalists, users, and even system programmers. This is indicated by many virus specialists (see Refs. 4 and 24).

The point is that operating systems intended for simultaneous use by a large number of users have rather complex systems to protect files and operating memory, that are intended to provide for the independent work of various programmers. Usually this protection is done through hardware and it is very difficult to circumvent it. Even if a skilled programmer could "break into" the operating system (having found, for example, the password) and introduce a virus, it would be very simple to remove it. Moreover, it is relatively easy to "calculate" who created the virus (for example, by the contents of the backup tapes, the terminal from which attempts to discover the password were made, etc.). There is no doubt that a serious criminal punishment awaits such a hacker.[5)] Thus, the game is not worth the candle.

Apparently, then, the only reliably known case of viral infection of large computers was the precedent-setting infection of about 2000 SUN-3 and VAX computers in the US. These computers were joined in the Internet network, and were infected with the Morris virus[4,24] on November 2, 1988 (by the way, this was before serious federal laws went into effect which deter the distribution of viruses). The virus was created by Robert Morris, Jr., a graduate student of the department of computer science at Cornell University (US). In writing the virus he used an error in the standard software of the UNIX system, which operated on most machines connected in the network. UNIX has so-called "daemon" processes which are not linked with any one user. One of these "daemons" is the program fingerd, which makes it possible to obtain information on other users in a given computer. The virus sent a request to fingerd in the network, however, the "daemon" transmitted too much information, and as a result, it overflowed the buffer and wrote to a memory region where the code of the "daemon" itself was located. Thus, in place of fingerd a simple program was recorded which issued the request to send to it from the computer which called the "daemon" a short (99 line) module written in C, which is standard for UNIX. Then the modified fingerd called the operating system with the command to translate this program and then boot it. The program destroyed all tracks, "hid" in the operating system, and then in turn called and booted special modules which were intended to determine the next candidates for infection. Subtle procedures were used to determine user passwords. When it received the name of the following node in the network, the propagation module sent a request to the "daemon" on this machine, thus insuring the spread of the virus. The total length of all elements of the virus program reached 68 kilobytes. This is the longest known virus code up to 1990.

Due to errors in the program, however, the virus spread too fast, and in several hours more than 1000 computers were infected, and on some of them work actually stopped, because the computers were so occupied with the transmission of the virus to other computers. This made it very difficult for system programmers to exchange information on how to combat the virus, because electronic mail began to function very unreliably. Nonetheless, by the evening of November 3 the virus had been destroyed, and errors which made it possible for the virus to spread were eliminated from the operating systems. Although the virus did not lead to any damage, and only paralyzed users' work for a day and a half, Morris was suspended from the university for two years. The possibility of taking him to court on existing criminal statutes was examined at the time.[6)]

In addition to the Morris virus, there are several other worm programs which try to infiltrate the operating systems using the standard names of directories and passwords. However, they have not become seriously widespread because they are quickly revealed by the large number of unsuccessful attempts to enter the system. Moreover, the passwords of system directories are usually changed periodically, thus, such worms are made more as a joke, and not to infect a large number of computers.

An example of such a program is the worm WNK (Worm Nuclear Killer), which was found on VAX computers with the VAX/VMS operating system at CERN and other European computer centers in 1989–1990. The worm was a batch file written in the DCL command language of the VMS system. When booted the virus program tried to obtain the passwords of standard users (SYSTEM, BACKUP, GAME, etc.) using a special list of about 40 names (names,

names of directories, etc.). When successful, the worm attached itself to batch files in the "opened" directory, thus providing for its propagation. When system directories were infected, the standard configuration file SYLOGIN.COM was changed, and when the user entered the system, instead of the name of the computer he received a message about the presence of the WNK worm on his machine. As far as the author knows, Nuclear Killer is not destructive.

Another virus of this type infected an internal IBM network in 1987. It drew a Christmas tree on the screen and delivered itself to the addresses found in the infected machine.

Thus, except for the Morris virus, only individual, not serious attempts have been made to create a virus for large computers. With the presence of complex defense systems for files and the core of the operating system, and with the enactment of laws in a number of countries which stiffen the criminal penalty for such actions, the further spread of viruses on large computers is not very likely.

1.2.2. *Viruses on personal computers other than the IBM PC.* Actually, the main grounds for the spread of viruses are personal computers with a simple operating system which has hardly any means of defense. And although the first virus programs appeared on IBM PC compatible machines, there are viruses on other computers of this class.

At present it is known that viruses exist for Macintosh computers, one of which infects application programs[25] and is very refined. A representative of Apple announced that the firm planned to conduct an investigation to bring its author up on criminal charges.

Unfortunately, in contrast to multi-user systems, it is usually very difficult to find the author of a virus for personal computers, because he can spread the virus through networks, computer clubs, and many other means while remaining "in the shadows."

There are even viruses for simple personal computers like the Commodore Amiga and the Atari ST. At least three such viruses are known to have been detected in the Soviet Union.

1.2.3. *Viruses on IBM PC compatible computers.* We turn now to the main theme of this survey, viruses on IBM PC compatible computers. In addition to an extremely simple operating system, MS DOS, which is in essence a monitor, virus authors are undoubtedly attracted to the wide distribution of such computers. In fact, IBM announced that they had given this personal computer an open architecture and published all the necessary information for company-users on the hardware and software.[26] This fostered the rapid creation of a large number of application programs (editors, compilers, databases, text processors, games, etc.) and a sharp drop in the price of computers, due to its simultaneous production by a number of small firms. In turn, the presence of a large number of programs unprotected from copying, lead to their widespread exchange.

The uniting of computers into networks only aided in the creation of extremely favorable conditions for the transmission of viruses from one computer to another. The widespread distribution of IBM PC compatible computers among nonprofessionals also leads to the rapid spread of viruses and extends the time before they are detected.

The virtual absence of copyrights on program products in the USSR and the universal "clandestine" exchange of software makes the Soviet personal computer market ideal for the existence of viruses. And the fact that only about a third of the viruses known in the world have been brought to our country only indicates a delay in propagation associated with the great shortage of computers and the high prices charged for them.

Below we will examine all the fine points of the anatomy and life cycle of viruses, and an attempt will be made to classify them and give practical recommendations to detect a virus on a personal computer.

1.2.3-1. *Where do viruses live?* A necessary condition for the spread of a virus is a single execution of its code. In this regard viruses may "live" only in programs (either part of the operating system or application programs).

Possible entry points of viruses into the MS DOS system are shown in Fig. 1. Let us examine these components of the software in more detail.

Master boot sector viruses. These introduce themselves or part of their body into the master boot sector at the location of the program which controls loading of the system. These viruses gain control when the system is booted from the hard disk.

Driver viruses. The drivers of peripherals are infected. Control is obtained any time the system calls the peripheral served by a given driver.

Boot viruses. These place themselves or part of themselves in the boot sector of diskettes and partitions in the location of the program which loads the system. They gain control when the system is started from the hard disk or from a floppy disk.

COM and EXE viruses or general purpose viruses. These attach themselves to executable modules of DOS. They gain control when the infected program is booted.

Batch viruses. This is a rather exotic form for viruses;[7] they are programs written in the command language of MS DOS. Due to their large size and the transparency of the command language these have not been spread practically.

Viruses which infect the initial text of a program. A theoretically possible type of virus which is a program in a high level language which introduces itself into files written in this language. After translation and booting of this program the virus rewrites itself into other files. It has features similar to the Morris virus, transferring part of its code in the form of a program in C. In principle it is possible to introduce the virus into system libraries for high level languages so that control is transferred to it when any standard function is carried out (see Ref. 3). However, these viruses are effectively spread only in a user medium which makes frequent use of compilers. Usually these people are sufficiently skilled in programming to rapidly recognize the presence of a virus and replace the system library with the original library. It is difficult to predict further prospects for this virus, but there have been no announcements about the appearance of such viruses.

Several types of virus programs can "live" in more than one of the places indicated. For example, the Marijuana virus (see below and Ref. 4) infects the boot sector of floppy disks and the master boot sector of Winchester drives.

One should understand that other places where viruses can live on IBM PC computers simply *cannot exist.* Of course, one cannot rule out the penetration of viruses into BIOS, but the manufacturer of such a microcircuit would

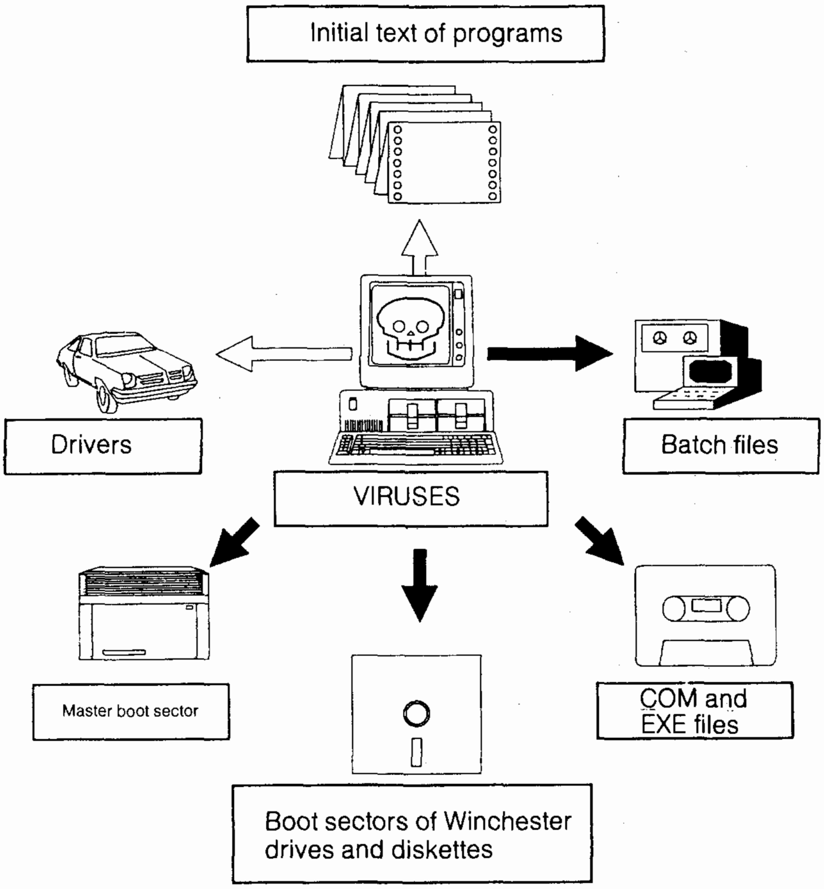

FIG. 1. Habitat for viruses on IBM PC compatible computers. The dark arrows indicate that viruses of this type exist, and the light arrows indicate that viruses do not yet exist or information about them is unreliable.

not last long. There are many stories about viruses living in CMOS memory (battery-powered memory which stores information about the configuration of the computer) but these are legends. Control is never transferred to CMOS memory, and so a virus cannot exist in the CMOS memory alone. It is true that there are viruses which use the CMOS memory of IBM PC/AT-386 and 486 computers (which have a larger CMOS memory than the usual AT) to store part of its code. One can actually observe a strange phenomenon: even after formatting the computer's disks, putting the virus out of commission, the computer refuses to work, and only removal of the contents of the CMOS memory makes it possible to restore the working capability of the computer. The explanation for this "paradox" is very simple. In later models of the series 386 and 486 part of the CMOS memory is assigned to the number of the given computer, which the user cannot even guess. The POST program verifies this number, and in case it does not coincide with the value stored in it, it refuses to load the system. If the virus placed part of its code into CMOS and damaged information on the number, then before the memory is completely expunged (in which case POST automatically boots the SETUP program) the computer will not work.

An announcement has also been made about the existence of rather interesting loader viruses (that is, master boot sector or boot viruses) which wait in the operating memory for the format disk procedure, if it is booted in an infected operating system.[8] Then after the formatting is completed the virus again places itself in the load sector; thus, it is not possible to destroy it in any other way than from an uninfected diskette with an operating system.

1.2.3-2. *The spread of viruses.* All viruses can be arbitrarily divided into two categories: those which remain resident in computer memory after they complete their work, and those which do not. The overwhelming majority of viruses belong to the first category, because the resident part insures fast and effective spreading. Before we examine these viruses, let us briefly touch on the self-reproduction of viruses without a resident part.

1.2.3-2.1. *The spread of viruses without a resident part.* These viruses must place their body in other modules only when the infected program code is working, which limits this class of viruses to general purpose viruses. When an infected program is booted control is transferred to the body of the virus, which finds, according to some algorithm, the next candidate program for infection, and the virus tries to infect it. How does it do this?

All general purpose viruses known at the time of the writing of this article place their body either at the beginning (COM files) or the end (EXE and COM files) of the infected module. The only exception is the Lehigh virus, which will be discussed a little later. Viruses change a program so that control is transferred to their body when the infected program is booted.

In the case of attachment to the beginning of a COM module this occurs automatically: the commands of the virus are executed first, and then the text of the program begins to be carried out. From the beginning it is displaced in the memory by the length of the virus code, so before it ends its work, the virus shifts the program to the low addresses. If the virus is attached to the end of the file, then it must change the code of the initial program so that control is transferred to its body. For a COM file this is usually achieved by replacing its first three bytes[9)] with the operator for unconditional transfer to the beginning of the virus (this command is a near command and occupies 3 bytes).

In the case of an EXE module it is necessary to change the first bytes of the heading to correct the volume of memory needed by the program as well as the length of the loaded part. Then the virus either changes its point of entry, writing a new initial value of the CS and IP registers in the heading (this value is calculated in the infection process, standard method) or instead of the first bytes of the program text at the point of entry, it writes the command of unconditional transition to its beginning (now it occupies 3 or 5 bytes depending on the type of addressing, because the transfer operator is a far type operator). COM files are lengthened by a fixed number of bytes when the virus attaches itself, but this is not quite the case for EXE modules. When EXE programs are infected the virus usually sets its beginning equal to the limit of a paragraph (16 bytes of memory) when then facilitates its operations to reserve memory and other actions. The lengthening of EXE files is equal to the sum of the base length of the virus and a number from 0 to 15, depending on the initial size of the file.

In all cases the virus remembers where in its body it stored the heading bytes changed in the process of infection or the initial text, in order to restore it before transferral of control to the program itself. The overwhelming majority of viruses are attached to the end of the file. First, this is the only possible means of infecting EXE modules (thus, the virus programs which infect EXE and COM files for economy usually place themselves at the end of the file). Second, even in the case of a COM modules the placement of the virus at the beginning of the file requires rewriting to the disk the entire body of the program (because in the MS DOS system any file should begin with the first byte of a cluster on the disk) which greatly increases the working time of the virus code and makes its presence very noticeable. Moreover, before control is transferred to the program it is also necessary to shift it in memory.

The Lehigh virus, which is a very specific virus, infects only the file of the command interpreter COMMAND. COM, whose structure is well known. Thus, the virus places itself not at the beginning or end of the file, but directly inside it, using the stack region reserved in the text of the program. The length of COMMAND.COM is either completely unchanged or changed by only 20 bytes (there are two varieties of the Lehigh virus), which masks the introduction of the virus. The remainder occurs as follows: the first bytes are changed to transfer control to the body of the virus. No other viruses that place themselves inside an infected program have been observed in the world so far.

After infection attempts are made, the virus returns control to the program itself, which it boots, first restoring the changed bytes of its code or information about the initial values of the registers (according to the content of the old heading, in an EXE module).

In principle, in the process of operation of this virus many files can be infected; however, in practice this does not occur because such actions would be too noticeable. Usually in one boot no more than one file is infected, which is completely sufficient for effective spreading. A typical example of a virus without a resident part is the Restart virus (see Section 3).

Algorithms to search for the next candidate for infection may vary: the first noninfected file in the current subdirectory; in the directories of the DOS path; in the entire disk, etc. Some viruses of this type specifically do not infect the command interpreter, which makes them more difficult to detect.

1.2.3-2.2. *The spreading of boot and master boot sector viruses.* These viruses gain control anytime there is a rebooting of the system from diskette (boot) or from a Winchester drive (master boot sector and boot). In order to spread successfully, these viruses should have a resident part which remains active even after the loading process.

All boot viruses known to 1990 intercept hardware interrupt 13h, with which it gains access to the hard and floppy disks. As soon as an attempt is made to access an uninfected disk, the resident part of the virus carries out the following operations:

—seeks a free sector on this disk;

—saves the master boot or boot programs in this sector;

—writes its body in the place of the master boot or boot programs (in the case of a long virus, only part is written in the load sector, and the remainder of the body is placed either in special maintenance sectors of the disk (the root catalog, etc.), or in arbitrary free sectors marked as bad[10)] for storing data);

—mark the sector with the saved load programs bad, to protect the information in it from erasure or overwriting.

If the disk which is infected in this manner is a bootable disk, then on the first attempt to boot the system with it, the following happens:

—the virus gains control;

—in order to hide, the virus shifts its body to the high addresses of memory;

—disk interrupt 13h is intercepted by the virus;

—the old boot program, which is in a known location on the disk, is loaded into memory;

—control is transferred to it, after which the usual booting of the system occurs.

Some viruses (for example, the Alameda virus, see Refs. 4 and 6) do not bother to mark the sector with the saved master boot sector or the boot record as bad; as a result, after some time it is erased in the next call to disk memory, and the system ceases to load.

Let us examine in more detail the masking of the virus in memory. The fact is that it is not possible for the virus to remain in place because its body will be destroyed after the load program is finished executing. Thus, to remain resident, it usually places itself in the high addresses of memory, after which it corrects the maximum volume accessible to the operating system, decreasing its value by the length of its body (usually 1 to 2 kilobytes). As a result, after the system is loaded MS DOS will assume that the machine has a little less memory, which is usually not noticeable to the user.

Thus, it is sufficient to access an uninfected system diskette in a computer infected with a loader virus, and the diskette will become a virus carrier and infect other machines when it attempts to boot the operating system from the diskette.

Some of these viruses trap a call to the master boot sector or the boot sector, and analyze the state of the registers at the moment interrupt 13h is intercepted. When an attempt is made to read or change the load sector, the virus deceives the system, "palming off" an old saved copy instead of the true sector with the load program. Thus, these viruses cannot be detected by comparing the load sectors with their copies stored in special files, if the operating system of the computer is infected.

In addition to the aforementioned Alameda and Marijuana viruses, other examples of loader viruses are Italian and Pakistani Brain (see below and Refs. 4 and 6). This type of virus is not spread too rapidly, because its only carriers are system diskettes. At the same time, it is not easy to detect them unless special measures are taken, and therefore they are encountered rather frequently.

1.2.3–2.3. *The spreading of general purpose resident viruses.* The mechanism which attaches general purpose viruses with a resident part to files is no different than that described in section 1.2.3-2.1. However, the means of spreading is fundamentally different. When the infected program is booted, the following occurs:

—control is transferred to the body of the virus;

—the virus checks whether the system is infected; if it is not, then the following two things are done:

1) the virus is hidden in memory;

2) some DOS interrupts, including the main interrupt, 21h, are intercepted by the virus;

—elements of the program code damaged by the virus are restored in the program carrier;

—control is transferred to the program.

Thus, a virus is not spread during the time of operation of the infected program. How does it occur?

As soon as a computer with an infected operating system boots some program, the resident part of the virus carries out the following operations:

—it intercepts 21h (subfunction 4Bh)—load to memory and boot program;

—it checks whether the program to be booted is infected, and if not, it attempts to infect it by writing its body to the file with its code (see section 1.2.3-2.1.);

—it transfers control to the standard interrupt 21h of MS DOS (the entry point of which is saved in the body of the virus when the system is infected), to the loading and booting program.

Thus, when any program is booted after the infected program, it too may be infected by the virus (some viruses do not infect everything, only COM modules or files whose length lies within certain limits; others do not infect COMMAND.COM).

Frequently the program booted after the infected program is the command interpreter, which is loaded from disk, thus it quickly becomes infected (that is, if the virus does not leave it alone). After this, even rebooting the system does not save it from infection, because booting the infected COMMAND.COM after the system is started up immediately activates the resident part of the virus.

Verification of the infection of the operating system is usually done to save time, and so that the system does not contain too many resident copies of the virus code. To do this the virus either places a key word in a reserved cell of memory or uses an unsupported MS DOS subfunction of one of the intercepted interrupts for the transmission of a "password."

Masking of a general purpose virus in memory may be done as in loader viruses (see section 1.2.3-2.2.); however, there are other variants. The point is that MS DOS uses a number of memory regions when it is working (DOS buffers) for maintenance. Some viruses skillfully conceal their body in these buffers, changing their length and all references to them. Thus, the Yankee Doodle virus (see Section 3) hides so well that it cannot be detected by even today's best programs that compile a memory map (this will be discussed in Section 2).

Some "smart" viruses intercept other interrupts as well. Yankee Doodle switches over to itself the interrupt used by debuggers to trace programs. If an attempt is made to examine the structure of an infected file in a debugger, the virus tracks it and removes itself from the file. The procedure for disassembling the virus code and creating an antivirus is made more complicated.

Another virus which plays dirty tricks is Dark Avenger (see Section 3), which intercepts the opening and closing of files and infects them if they have the extension .COM or .EXE. It is sufficient to examine such a file with a text editor or the command TYPE to infect it. If an antivirus cures a file in an operating system which is infected in this way, the file will be infected again when it is closed.

Many viruses use timer interrupts to activate themselves at a certain time of day.

1.2.3–3. *Indications that a computer is infected.* Frequently the presence of viruses on a personal computer is easily detected before they are activated. From the material in the previous sections one can indicate the following suspicious happenings on the computer which should alert the experienced user:

—an increase in the size of certain files. It follows from section 1.2.3-2.1. that this is a common manifestation of COM and EXE viruses;

—repeated breakdown and "hanging" of the system when some standard programs are booted. This situation is also very suspicious because viruses which are not written very well incorrectly infect special types of files. For example, if the sum of the length of the virus and the COM file exceeds 64 kilobytes, and this is not checked during infection, as a result, when an attempt is made to boot this program it will not be loaded, and the diagnostic message "Out of memory" will appear. Some viruses incorrectly infect EXE files assuming they are COM modules because programs do not differ in internal format, only in their name extensions. This also leads to hanging when a boot attempt is made. Other viruses (Dark Avenger, Restart) slowly damage files, which may also be the reason for breakdowns in the operation of standard programs. Finally, the Dark Avenger virus scans BIOS in order to determine the entry point of interrupt 13h. The algorithm used to do this leads to "hanging" of a program on some computers:

—the appearance of new defective sectors and clusters on floppy disks and Winchester drives. The point is that bad clusters are noted only when a disk is formatted or when special programs are booted. If either occurs, an "unexplained" appearance of defective clusters usually means the presence of a loader virus in the computer.

—an increase in the load time of the system or programs, and computer slowness are also characteristic indications of the presence of viruses.

1.2.3-4. *Activation of viruses.* The harm that viruses cause can be categorized as follows:

-"*harmless*," that is, they have no side effects other than spreading (Vacsina, Micro88; see Ref. 4 and Section 3).

-*amusing*, that is,they have no destructive effect, and only create visual or audio effects (Chucha,[27] Yankee Doodle, Falling Letters, Marijuana, see Section 3), for example the appearance of inscriptions, the playing of melodies.

—"*fighting*," that is viruses intended to disrupt the file system (Alameda, Pakistani Brain, Lehigh), to change individual files [Restart, Dark Avenger, dBASE[11)] (Ref. 9)], as well as put individual elements of computer hardware out of commission. There is information on the existence of a virus which burns out the screens of monochrome monitors using the properties of their control circuit. In principle, one can also create a virus which puts the floppy disk drive out of commission [by setting the maximum head speed and periodically moving it from the first track to the last when the user is not nearby (for example, when no one accesses the keyboard during a prolonged period of time)].

The activation of fighting and amusing viruses occurs under very different conditions. Here everything depends on the imagination of the author. There are several groups of activation conditions:

—according to a timer at a specific time of day (Yankee Doodle);

—depending on the readings of system clocks (Falling Letters, Jerusalem);

—periodically, once every several boots of infected programs (Dark Avenger, Restart);

—after a certain number of files are infected (Lehigh).

There are also more complex logical conditions for virus activation (Italian).

One should bear in mind that even "harmless" viruses are harmful.

—They increase the size of files and the load time of the programs.

—They can easily be transformed into fighting viruses.

—Even the most sophisticated viruses, saying nothing of low quality ones, can lead, under certain conditions, to "hanging" of the system.

1.2.3-5. *Attempts at classification.* A large number of viruses have several common names given them by the first researchers, frequently independent of each other. Thus, the Jerusalem virus has at least five very commonly used names (Israeli, Black Friday, Friday the 13th, Time, Black Hole). This variety of names makes it difficult to cure viruses because frequently it is not understood for which virus a specific antivirus is intended.

The situation is somewhat reminiscent of the situation in physics with elementary particles, when along with the commonly used names (like h-meson) the designations of the Particle Data Group are being earnestly cultivated (f_4 meson), which frequently leads to confusion.

At present the classification of viruses is probably not confused, and is most similar to the classification made by Jorge Luis Borges in his story "Analytical language of John Wilkins."[28] By the way, this example is used in L. B. Okun's book *The Physics of Elementary Particles.*[29]

It makes sense to create a single system of notation for viruses which makes it possible to identify them easily through external indications. One of the first attempts of this type was undertaken by N. N. Bezrukov.[4,30] According to his classification, each virus is designated by a certain combination of letters and numbers. The notation includes three elements:

—a classification code which contains the main characteristics of the virus, characteristics sufficient to identify it;

—a descriptor, which is a formalized list of its properties;

—a signature, that is, a line for a contextual search of the body of the virus in an infected file.

The descriptor and signature may be useful to the authors of antiviral programs,[12)] but even a person who is not a specialist in this field can use the classification code to determine the type of virus.

Let us now discuss in more detail the classification code for viruses. It consists of an alphabetic prefix, a numerical root, and a suffix.

The prefix is one or several letters which indicate the type of virus. Thus, general purpose viruses which infect COM files have the letter C as one of the letters of the prefix. If they also infect EXE modules, then the prefix is CE. For viruses with a resident part the additional letter R appears in the prefix. For loader viruses the prefixes B, D, and M are used or combinations of these letters, depending on whether they infect the boot sector of hard disks (B), floppy disks (D), or the master boot sector (M).

The numerical root indicates the characteristic length of the virus. The viruses which infect EXE files do not lengthen all files identically, so a certain base value is used (that is, the change in the length of a COM or EXE file leveled to the paragraph boundary).

An optional suffix may indicate the number or properties of a "strain" of a given virus which cannot be distinguished from the root and prefix.

For example, the Falling Letters virus has two varieties 1701 and 1704 bytes long, it infects only COM files, and has a resident part. Thus, the two strains of this virus are designated RC-1701 and RC-1704. There is also another variety of RC-1704 which formats disks. It makes sense to designate it RC-1704F. Usually viruses which differ only in their suffix are cured by the same antivirus, thus, in principle, it is not so important for a cure.

The loader virus Italian occupies two sectors on the disk, and according to this classification, it has the code RBD-1024; the Marijuana virus is designated RDM-512.

2. COMPUTER PHARMACOLOGY

Now we know quite a bit about viruses. However, it is better not only to know them well, but also to fight them. Thus, this section of the article is devoted entirely to various antiviruses and other means of protection from virus programs. The wide spreading of viruses has led to the creation of many program products of this type for hobbyist programmers and serious firms. The situation is reminiscent of the famous rule about the competition of a missile and armor.

Before we begin our discussion of antiviruses, we wish to caution that using them without thinking could lead to the same type of harm as using all home remedies at the first signs of a head cold. When antiviruses are used incorrectly, they are barely able to help cure the computer. The reasons for this are explained in the following two sections.

2.1. On rules of good form in the creation of antiviruses

"Do no harm"
(*From the Hippocratic Oath*)

At present, several hundred antiviral programs are known in the world. Many of them partially or completely duplicate each other. It is not always possible to understand which antivirus fights which virus because the documentation for such programs is usually absent or written in uncommon languages (Italian, German, Polish, Russian, etc.) Viruses do not acknowledge national borders, so we must fight them side by side. Thus, in our opinion, one of the rules of good form in the creation of antiviruses is that all the diagnostics and documentation should be written in the commonly accepted standard for scientific communications, English. No one has thought to translate FORTRAN or C into Russian. This is exactly why, to cure viruses, one need not know English well to understand the diagnostic messages. This is why all the names of viruses in this article are given only in English (except for the virus Chucha which uses Cyrillic).

The author calls for the simply ruthless erasure of antivirus programs which do not have sufficiently understandable instructions for use (either in the form of a separate file or within the antivirus itself).

The point is that using these programs hardly does any appreciable good, and the harm can be great. Actually, programs which do not have good documentation for use are frequently written by amateurs (as experienced programmers know, a program without a description is not a program product). These do-it-yourself programs are usually created carelessly and have a very low quality. This leads to the situation that after these programs work, if they identify the virus, a large number of files are irreversibly damaged. Usually this occurs because the authors of the antivirus programs choose the context line for the detection of the virus inaccurately, and as a result several widely differing versions of the virus are treated by the program using a single principle, and this leads to irreversible losses.

Another common error made by the authors of such programs is that their effect is checked only on some standard EXE files which usually have a length of a multiple of 16 bytes. If an attempt is made to cure EXE files of another length, then usually the virus is not totally removed from the file, and sometimes the converse is true, several bytes of the code of the program itself are removed with the virus. The same outcome can be obtained in curing modules which contain information for debugging or programs intended for simultaneous use under MS DOS and OS/2 (the majority of new products by Microsoft). Of course not all viruses allow absolutely accurate restoration of the files which they infect, but if this is possible, sufficient time should be devoted to searching for nuances in the infection of various types of files.

One should recall that when a new medicine is invented, before it is mass-produced it must pass through lengthy clinical testing. Even the fact that a computer is infected with a virus which must be promptly removed is no justification for writing and copying such unfinished programs, because they spread too fast and control over them is lost instantaneously.[13]

The creation of antiviruses should be governed by the main principle "do no harm." *It is better to have an infected working program than an irreversibly damaged, but virus-free program.* In exactly the same way one must select antiviral programs for use on one's computer.

2.2. On the harm of treating yourself

There is a widespread practice among users of booting a whole series of antiviral programs (ten or more) when the first signs of computer breakdown appear. Frequently these breakdowns are caused by errors of the user himself, or the use of low-quality (in the majority of cases, homemade) software.

Without mentioning that this procedure takes a great deal of time, it may simply lead to infection of the entire computer and the loss of a large number of files. Where is the danger?

—Antiviruses work in close contact with viruses, thus, they are themselves frequently infected. When ten or so such programs are used one after another, usually programs obtained from friends or colleagues, the probability of infection of an uninfected computer becomes very high.

—Many antiviruses incorrectly treat "strains" of the virus for which it was intended (see section 2.1.) which leads to the loss of files.

—When a computer is infected by a "serious" virus, for example, Dark Avenger, the booting of a series of antiviruses leads only to infection of all executable files in the computer as well as the appearance of a large number of damaged sectors which can be found in executable files and data files.

As in medicine, when the presence of a virus in a computer is suspected, it is best to turn to a specialist. In section 2.4. we will give some simple rules, which, if followed, will almost certainly protect the computer from becoming infected.

2.3. What types of antiviruses are there?

"If the enemy doesn't surrender,
he is to be annihilated" *M. Gorkiĭ*

All antiviral programs can be divided into several classes, and these are shown in Fig. 2. The figure also includes the means of combatting viruses and auxiliary programs which make it easier to identify viruses and help protect files from infection. Let us examine these in more detail.

Indicators. These are programs which search for code sequences that are characteristic of various viruses in the text of infected modules. The most well known programs of this type are SCAN (copyright 1990, McAfee Associates) and VIRSCAN (copyright 1990, IBM).

Indicator-phages. In addition to determining the virus from a characteristic element of code, they attempt to cure the file, more or less successfully depending on the quality of the indicator-phage. The best phages are capable of neutralizing the resident part of viruses, making it possible to continue work after the cure without rebooting the system. There are universal phages capable of curing a number of viruses [CLEAN57 (McAfee Associates), VR (SiP), AID-

STEST (D. Lozinskiĭ), ANTI-KOT (O. Kotik), etc.] and specialized phages designed for a specific type of virus and, possibly, its "strains" [DISARM (J. Blach & M. Weiner–Falling Letters 1701), DR-NO (H. Leeb-Restart), PASTER (G. Landsberg—Falling Letters 1701/1704)]. There is an empirical rule: the more types of viruses a program cures, the less well it does it. Thus, VR and AIDSTEST damage files infected with the 1704 virus of Falling Letters; CLEAN57 cannot cope with the majority of cases of Yankee Doodle, etc. It is reasonable to use universal indicators or phages in indicator mode, and then cure the detected virus with a good specialized phage.

Vaccines. A verification portion is included in the code of programs which compares the checksum, length, and fragments of code with saved values. Sometimes they can restore infected programs. Usually they act like a general purpose virus, attaching themselves to the protected program and checking it when it is loaded into memory before execution. The advantage of vaccines is that they may recognize new types of viruses; their drawbacks are that it is impossible to cure an infected file if it was not vaccinated beforehand, it increases the loading time of the program, it cannot cure Dark Avenger type viruses, which reinfect the file when it is closed, etc. Examples of vaccine programs are: STAMPER (A. Chizhov) and PROTECT (D. Stefankov). A somewhat different approach was used in the universal vaccine PHENIX (G. Landsberg) (see Section 4) which made it possible to eliminate the majority of drawbacks inherent in typical vaccines.

Programs which verify the checksums and the state of the file system. The operating principle of these programs is the writing of the checksums of files or other information in a special database with subsequent comparison of this information to the current state of the system. This method makes it possible to reveal virtually any type of virus (especially if the checksum is calculated using several methods). The drawbacks of such programs are the length of the procedure for calculating the checksums and the need to renew the database constantly when new versions of programs appear with the same names, as well as when there are changes in the file structure (the creation of new subdirectories, copying, renaming, erasure of files, etc.). Examples of such programs are CRCDOS (R. Faith), SENTRY (McAfee Associates), and Vaccine 1.3 (Art Hill).

Monitors. Monitors trap suspicious events on a computer (an attempt to open EXE or COM files for writing, the interception of several interrupts, etc.). When these events appear they ask the permission of the user for their execution. Examples of such programs are VIRBLK (M. Fitz) and ANTI4US (E. Lantung). The monitor is still powerless against viruses which use direct access to the disk (for example, Yankee Doodle). McAfee Associates has produced a symbiosis between an indicator and a monitor (SCANRES; the latest versions are called VSHIELD) which is a resident program which checks all files being loaded for the presence of characteristic elements of a large number of viruses. In addition to program monitors there are also hardware monitors, which are usually implemented as a BIOS expansion which intercepts the 13h interrupt and traps any attempt to write to the loading sectors (for example, Ref. 31). They are intended to fight boot viruses. However, the author feels that it will be useless to fight new monitor viruses without an indicator, because nothing stops smart viruses from using for its actions the entry point of the 13h interrupt for stan-

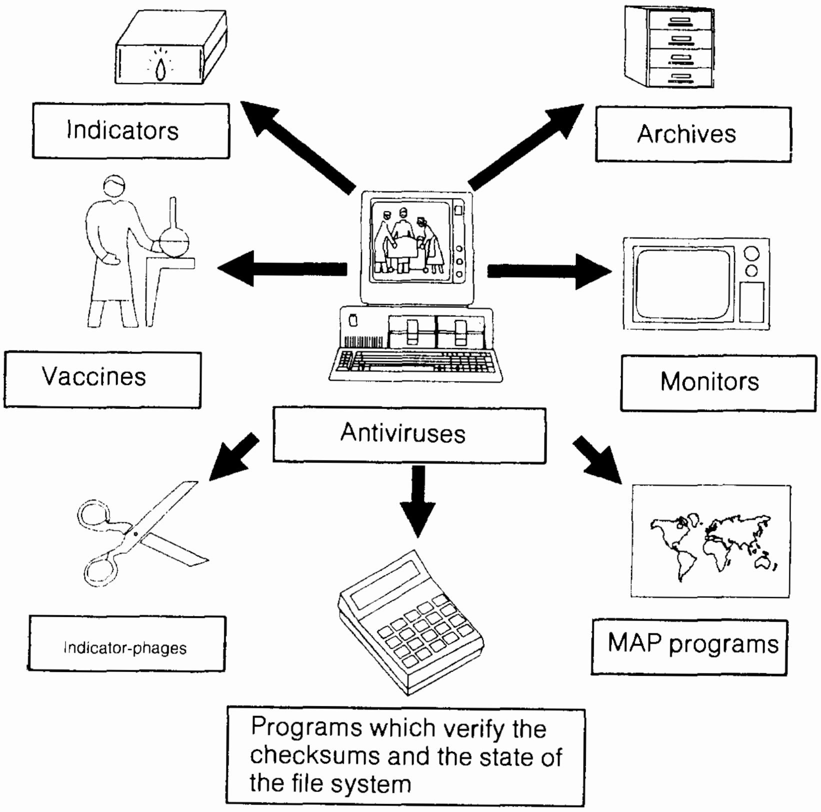

FIG. 2. Types of computer viruses.

dard BIOS (which is not so difficult to find with simple syntactic analysis). Thereby all monitors of this type are "out of the game" because a virus will not use even an explicit call to interrupt 13h.

MAP programs. These programs are intended to construct a computer memory map with an indication of the interrupts intercepted by various programs. One can sometimes detect the presence of a virus using these maps, but basically they should be seen as an auxiliary tool for specialists who fight viruses. Examples of these programs are VTSR (Golden Bow) and PCMAP (D. Stefankov). Another type of MAP-programs shows a map of the disk which indicates bad sectors. Such utilities may reveal the presence of loader viruses. VMAP (Golden Bow) is one of these utilities.

Archives. These programs are intended to archive or backup data. On the one hand this is a reliable means of saving a program from a virus; on the other hand, if the virus enters the archive, it constantly arises again and again, and it is very difficult to completely annihilate it. The best archives are made by PKWARE Inc. (PKARC, PKZIP), and the best backup programs are made by Central Point Software (PCBACKUP).

Additional information about antiviral programs can be found in Refs. 16, 32, and 33, as well as in R. Roberts' book *Computer Viruses.*

2.4. How can we protect ourselves from viruses?

Some personal computer users are so intimidated by the presence of viruses that they will not even update their software or use other people's programs, etc. Naturally isolation is also a way out, but it also usually does not lead to the desired result, if only your computer is not locked in a safe and the opening of the disk drive is not stopped up with epoxy.

It is much simpler and safer to understand how one can be infected with a virus, and take a number of prophylactic measures before loading new software into the computer. What type of measures?

If the new software you obtain is not on the original write-protected disks from the firm, then before you load it into the computer, check the diskette with a good virus indicator, for example, the SCAN program, preferably the latest version. If some of the programs are in archives on the diskette, one must first unpack them into some directory of your hard disk (usually there is no room on a diskette for such an operation), for example, into the directory \TMP. If the indicator does not detect a virus on the diskette and in the archives, then it is likely that you have an uninfected program product. For insurance (what if you suddenly get a yet unknown virus?) one can do the following

If the diskette contains an operating system and the intent is to load the computer with it:

—check it for the presence of clusters marked as bad (if there are such clusters, this is very suspicious);

—immediately after loading, access your hard disk, for example, with the command COPY (but do not load programs from it);

—after this, quickly reload the system from a diskette which is known to be good, which is write-protected, and which contains, in addition to the operating system, a copy of the boot sector and master boot sector for your Winchester drive. One can obtain these files using special programs (BOOTCHECK) from McConachie Associates, PHENIX by G. Landsberg, etc.), or using the well-known Norton Utilities program. After reloading compare the contents of these files with the condition of the Winchester drive. If they coincide, then there is no loader virus on the diskette. If they do not coincide, the diskette is infected, and one must immediately restore the damaged sectors of the Winchester drive using these same programs.

Even if the diskette is infected by a loader virus, the files on it can still be used if, of course, one is not required to load them with this diskette. Frequently the diskette can be cured of the loader virus with the SYS command, which transfers a "clean" operating system and reformats the loading sector (in this case the version of the old and new systems should coincide!).

If the diskette contains executable files (diskettes with an operating system always have at least one such file, the command interpreter):

—load from your hard disk without using shell programs (like Norton Commander or PCShell) and boot the executable files on the diskettes one after another. Carefully track the length of your command interpreter. If it has not changed, try to boot several standard DOS programs which can be easily restored if they are lost (MORE, ASSIGN, ATTRIB, as well as the command interpreter itself, COMMAND.COM, etc.). If their lengths also are not changed, it is very likely that you are dealing with uninfected programs and can work with them. *At the first indications of change in the length of the booted files* immediately erase all files on the Winchester drive which you managed to load. Finally you should erase the command interpreter. After this, reboot with a system diskette and restore the erased files.

A diskette with infected files or an infected boot sector should be sent to the indicator manufacturer so that they can take into account the presence of a new virus in their next versions of the program. If you know people who are actively creating antiviruses, you should also send a copy of the diskette to them.

Guided by these simple requirements you are almost guaranteed not to have a virus infiltrate your computer.

3. VIRUSES IN THE USSR

As noted above, by 1990 in the USSR about 25 viruses were known of approximately 70 registered in the world at that time. This section is devoted to the most widespread viruses in our country, and contains a brief description of their characteristics, indications of infection, and recommendations on how to cure infected computers.

3.1. The Restart virus

One of the first viruses registered in the USSR. There are a number of other names for this virus, including Vienna virus and Time Bomb. According to N. N. Bezrukov's classification (see section 1.2.3–5) this virus has the code C-648, that is, it infects only COM files, lengthening them by 648 bytes, and does not remain resident. The virus places its body at the end of the infected program. When it is booted it seeks the next candidate program for infection in the current directory, and in the directories of the DOS path. There is a probability of 1/8 that it will not infect the program it finds and damage it (by writing in place of the first five bytes of the file the command for unconditional transfer to the address of the reloader FFFFh: 0000h).[14] Booting this program is analogous to pressing Ctrl-Alt-Del and leads to rebooting of the system. If the file which is damaged in this way is booted from AUTOEXEC.BAT, the rebooting process is repeated. As an indicator of file infection the virus uses the "nonphysical" value of the seconds field (62 seconds) in the creation date. There are a number of antiviral phages which cure infected (but not damaged) files, among them SERUM3 (M. Fitz, H. Veit, ANTI-KOT (O. Kotik, and AIDSTEST (D. Lozinskiĭ).

3.2. The Micro 88 virus

This is a somewhat improved version of the Restart virus. It infects only COM files, increasing their length by 534 bytes. Its classification code is C-534. In contrast to the Restart virus, it does not damage files, it only infects them. Infected modules are marked by setting the month in the date of creation equal to 13. Antiviral phages are ANTI-KOT (O. Kotik) and AIDSTEST (D. Lozinskiĭ).

3.3. The Jerusalem virus

There are a number of other names for this virus (see Section 1.2.3–5).

It infects EXE and COM files, lengthening them by 1808 bytes. The virus remains resident in the memory of the computer, intercepting interrupts 21h and 08h (timer interrupts). The Jerusalem virus is classified as RCE-1808. When COM files are infected the virus code is written at the beginning of the file, and five bytes are appended to the end of the file. The five bytes contain the symbols for "MsDos" (used to recognize files which have already been infected). When EXE modules are infected the virus places itself at the end of the file; however, it does not append the key word. Thus, the EXE modules can be infected multiple times, "swelling" them to a very large size. Some period of time after the computer begins work, the virus slows it severalfold, using the idle cycle to process the timer interrupt. Moreover, in the lower left hand corner of the screen a black square appears. If the system date is set to the 13th hour on a Friday, then instead of infecting it the virus erases the loaded program. Jerusalem distinguishes EXE and COM files not by internal format but by name, so it makes modules with the wrong name extension unworkable. Antiviral phages are ANTI-KOT (O. Kotik) and AIDSTEST (D. Lozinskiĭ).

3.4. The Falling Letters virus

There are at least two varieties of this virus 1701 and 1704 bytes long. It infects only COM files, and remains resident. The two "strains" of the virus are classified as RC-1701 and RC-1704. It is activated on computers without internal clocks or on machines with clocks if the system date is between October and December 1988. The external manifestations are "shedding" of random letters on the screen, accompanied by the characteristic sound of a drop. At first this amuses the user, but then it makes work on the computer impossible because the "shedding" occurs more frequently each time, and control is taken away from the user until the last letter "falls." There are several modifications of the virus which are activated by different indicators (the author knows of a variety of RC-1704 which manifests itself during even months). There is also a modification RC-1704F which formats the disk. A typical error of the majority of antiviral phages [AIDSTEST (D. Lozinskiĭ, VR(SiP)] is that they treat RC-1701 and RC-1704 using the same algorithm, which leads to irreversible damage to files infected by one of the two types of this virus. The author knows of two programs which are free of this drawback: CLEAN (McAfee Associates) and PASTER (G. Landsberg). The indicator-phage of PASTER is capable of neutralizing the resident part of the virus, even if several resident programs were loaded after it. This makes it possible to work on a cured computer without rebooting.

3.5. The Yankee Doodle virus

This has another common name, the Five o'clock virus. The viruses belong to a series of several similar programs which play the tune Yankee Doodle Dandy under certain conditions. The author knows of two varieties of this virus which lengthen COM files by 1805 (RCE-1805) and 2885 (RCE-2885) bytes. The first plays the tune after Ctrl-Alt-Del is pressed (reboot the system), the second at time 16:59:53. These versions of the virus do not damage the file system. There is information about the existence of at least three other varieties of the Yankee Doodle virus, and the latest of these damages files. The viruses are written very coherently; when they were created measures were taken to neutralize monitor programs like VIRBLK and ANTI4US (see Section 2.3.). There is also protection from examination of infected files by debuggers: when they are used the RCE-2885 virus removes its body from the file. There are several antiviral-phages which cure files infected by this series of viruses: AIDSTEST (D. Lozinskiĭ, 4 varieties). VR (SiP, 3 varieties), SHOPEN (G. Landsberg, RCE-2885). The SHOPEN program is capable of neutralizing the resident part of the virus, even when other resident programs are present, which makes it possible to work on a cured computer without rebooting the system.

3.6. The Vacsina virus

This virus has the classification code RCE-1339. When COM files are infected they are lengthened by 1339 bytes; when EXE modules are infected (the virus attaches itself only to EXE modules with a length less than 64 kilobytes) they are first lengthened by 132 bytes,writing at the beginning the command to transfer to the body of the virus, which occupies itself only with the transmission of control to the program itself. After this the file virtually ceases to be an EXE module and may be infected again by the Vacsina virus, but as a COM module. It is not destructive. Antiviral phages which cure files of the Vacsina virus are ANTI-KOT (O. Kotik) and AIDSTEST (D. Lozinskiĭ).

3.7. The Dark Avenger virus

This virus is classified as RCE-1800. Other names of this virus are Sofia and Eddie. Dark Avenger spreads extremely rapidly because it tracks not only the booting of programs, but the opening of files by the programs for reading and writing, as well as closure of the files. In this regard the virus cannot be cured without neutralization of the resident part. One time in 16 boots of the infected programs (the counter is stored in one of the inactive bytes of the boot sector) it destroys a relatively random sector of the disk, placing there the contents of part of the RAM, beginning with the phrase "Eddie lives...somewhere in time!" Thus, the virus can also damage data files. The virus does not permit exact restoration of the length of infected EXE modules if they have a nonstandard heading which is used for simultaneous booting under DOS and OS/2. The majority of antiviral phages [VR (SiP) and CLEAN (McAfee Associates)] incorrectly restore the length of the EXE modules (the cured file becomes somewhat longer than the original) even when this is possible. The indicator-phage SOFIA (G. Landsberg) restores the length correctly where this can be done. When an infected OS/2 module is detected it announces that this file may be cured incorrectly and it should be replaced with the original from the firm. The SOFIA program is also capable of completely neutralizing the resident part of the virus even when there are other resident programs, which makes it possible to work on the computer after it is cured without rebooting the system.

3.8. The Italian virus

This has the classification code RBD-1024. Other names are Italian Bouncing and Ball. This loader virus is not destructive. Under certain conditions a dot begins to move on the screen, bouncing off its edges and off certain symbols. Like the majority of boot viruses, it can be removed with the command SYS, as well as BOOTCHEK (McConachie Associates) and AIDSTEST (D. Lozinskiĭ).

3.9. The Marijuana virus

This virus has the classification code RBM-512. Another name is Stone. This virus is not destructive either. On floppy disks it infects the boot sector, on hard disks, the partitions table. There is a probability of 1/8 that when it is loaded the legend "Your PC is now stoned" will appear on the screen. The initial loading sector is placed in the last sector of the main directory on floppy disks or in the seventh absolute sector of a Winchester drive (which is usually empty). There have been communications about the appearance of a new strain of this virus which uses Cyrillic (obviously, general russification of western software products has also touched the world of viruses...). It can be cured in the same ways as the Italian virus (see Section 3.8.).

4. UNIVERSAL PHENIX ANTIVIRAL SYSTEM

This section briefly examines the PHENIX universal antiviral system, which was developed by the author for comprehensive protection from viruses. It is described in more detail in a separate publication[2] which also includes an explanation of the operating principles of the program and instructions on how to use it.

From Section 2 it is clear that the overwhelming majority of viruses change only a small fragment of the program code of infected modules. Consequently, if information is saved on the length of the program, the date of creation of the file, and an element of programming code near the entry point (and part of the heading in the case of EXE files), then one can frequently restore a file infected by a general purpose virus, even if the virus is new and as yet unknown. Actually, the length of the infected file may indicate the size of the virus, and a fragment of the initial code can indicate whether it is at the beginning or end of a file. By saving information on a file which is known to be uninfected (for example, one drawn from distributions by the firm) one can not only detect the presence of a virus in it, but usually remove it.

The vaccines which were created earlier (see Section 2.3.) acted like general purpose viruses without a resident part, attaching themselves to a file and checking it for infection before control was transferred to the program. This method has a number of drawbacks. First, the files are lengthened by several kilobytes, which decreases free space on the disk. Second, the time to load the program into memory is lengthened (or more accurately, the time from the moment the command is given to start the program to the beginning of the execution of its body). Third, not all COM files may be protected in this manner because their size, including the vaccine program, cannot exceed 64 kilobytes. Fourth, when there is Dark Avenger type virus on the computer which tracks the opening and closing of files, it is impossible to effect a cure with this type of vaccine because the cured file is reinfected when it is closed. Finally, the simplicity of such programs, which is dictated by economizing on the length of the vaccine code, in principle makes it possible for virus authors to surmount the protection easily. The result has been that vaccine programs have not become widespread.

Thus, to remove these drawbacks in vaccines, PHENIX uses another principle: It only stores records about the state of a file or program, tracking the "health" of the computer in a separate file. This program is loaded periodically or when the machine exhibits strange behavior.

The question arises of where such information can be stored (we will call this the *protection record*). The use of a database for this purpose, as is done in programs which calculate checksums, is inconvenient for the reasons cited in Section 2.3.: frequent change in the file system would require constant updating of the database with the active participation of the user in this process, because he would continually have to answer the question of whether the file coincides in internal structure with the protection record, is simply a new version with the same name, or if it is infected with a virus and must be cured. In this regard PHENIX uses a fundamentally new approach: *the protection record should be stored in the file itself.* This is done either by placing it at the end of the file or by *introducing the record into an unused portion of the protected program* (a stack region, an unused portion of the heading of an EXE module, etc.).

Thus, the presence of a short (about 40 bytes) protection record has no effect on the loading speed of the module, and when it is introduced into the body of the program has no effect on its length either, making it possible to restore a file infected by virtually any general purpose virus. According to the author's estimates, the PHENIX system is capable of neutralizing more than 50 of the viruses which exist today.

To reveal and eliminate loader viruses, at the time the file system is vaccinated also the load sectors and partition tables are saved. Beginning with version 2.0, when the PHENIX program is installed on a specific computer, it performs a syntactic analysis of the BIOS code in order to determine the entry points of the 13h interrupt, which provides a unique capability of detecting and destroying boot viruses *in an infected operating system* (see section 1.2.3–2.2. on the interception of interrupt 13h by loader viruses).

The PHENIX program takes serious measures to encode its code and the contents of the protection record and files with information about load sectors. Several types of checksums are used to verify the correct restoration of infected files.

The program is capable of neutralizing several types of damage created by viruses (for example, it can restore files damaged by the Restart virus; see section 3.1.) and it can remove defects in files which are incorrectly cured by bad antiviral phages. There are no problems with curing multiple "strains" of existing viruses when there are two or more viruses present in the computer simultaneously.

The PHENIX system has a well-developed menu-based user interface which makes it possible to select the correct working configuration. There is also a full processing system for errors, including disk exchange errors.

The vaccine protects executable modules, overlays, drivers, etc. The protection record may be removed from the protected file at any time. It is also possible to mark individual files in order to admit them for protection and scanning. This is useful for SCAN type programs (McAfee Associates), which verify the checksum during loading. Information on marked files is also contained in the file itself using several fine points of the MS DOS operating system associated with the storage of the creation date of the file.

The program is at present one of the most powerful means of detecting and destroying viruses, and when used regularly reliably protects the computer from virus infiltration.

CONCLUSION

Is there a reason for optimism? Viruses are spreading on IBM PC compatible computers at a record pace. The number of new types of viruses is rising steadily and faster. Is there a reason for optimism, or will viruses rapidly make work with simple operating systems impossible?

In the author's opinion, the situation is not hopeless. The existing protection methods, which are already engaged in a battle with viruses, has made the creation of new types of viruses which are not noticeable to antiviral programs much more difficult. At present the writing of such a virus is a task which is an order of magnitude more complex than it was three years ago. One can hope that sore losers (those who, in the opinion of psychologists, create the most destructive viruses) will not be able to overcome this barrier. Virus builders, naturally, will remain. One can hope that these people will be truly talented programmers who are seeking elegant virus programs for their own satisfaction, and who do not have the goal of destroying data and creating disruptions for other users... .

Along with a definite "global" optimism one should bear in mind that many viruses have not yet appeared in the USSR. Let us have no illusions. These viruses are certain to come, just like the virus programs created in the Soviet Union.[15] And we should be ready for their appearance.

In conclusion I would like to mention several sources of information used in the writing of this survey which are not indicated in the bibliography. Some information on the structure of viruses is contained in the instructions for the phage programs ANTI-KOT (O. Kotik), AIDSTEST (D. Lozinskiĭ) and SCAN (McAfee Associates). Information on the worm program WNK was graciously provided by M. Ikeda. The author would like to thank all these people.

[1] Address = (segment $\ll 4$) + offset, where $a \ll n$ is the shift of word a to the left by n bytes, which is equivalent to multiplying it by 2^n.

[2] The Intel 80286 and 80386 processors have an expanded command set, but the majority of application programs designed for work on any IBM PC computer do not use this capability.

[3] Hereinafter the symbol h after a number will indicate a hexadecimal counting system (100h = 256).

[4] This is reminiscent of the behavior of a patient who does not come to the doctor's office until the onset of a serious phase of illness and infects others.

[5] A hacker is a person who engages in "computer hooliganism," that is, he attempts to use computers illegally (selecting passwords, giving himself enhanced priority, "breaking into" protected systems, etc.).

[6] When the author was writing this survey there was an announcement that Morris had been fined a large sum of money.

[7] At present this is the only such virus known, and it is described in Ref. 15.

[8] There are several such "elusive" virus programs which behave, at first glance, very strangely and inexplicably. The presence of such "smart" viruses has generated a number of legends about the elusiveness and supernatural nature of computer bacilli. In fact there is nothing inexplicable about their behavior and there cannot be, because all viruses are, all in all, programs, although it is true that they are sometimes very coherently written programs.

[9] Some viruses, for example, Yankee Doodle (see Section 3) change a large number of bytes in the beginning of COM files, and some maintenance information is stored in their place.

[10] Virtually all disk controllers (except possible the very first ones) have the capability of marking as bad defective sectors on disks. These defects may arise due to scratches, and inhomogeneous magnetic layer, etc. The operating systems then do not use these sectors.

[11] This virus switches pairs of numbers in files with the extension .DBF (used by databases). It is usually very difficult to detect errors until, for example, someone receives 0100 dollars instead of 1000.

[12] The author has doubts about the usefulness of signatures for creating antiviruses, because the signature will be immediately changed by the creators of "strains."

[13] The author had imprudently cured one computer of the Yankee Doodle virus with a test version (V 1.0) of his SHOPEN antivirus. Afterwards, despite the request not to distribute this imperfect copy, in the course of half a year he was forced to keep replacing it with a high-quality version (V 1.7) on many personal computers at the Institute for High Energy Physics.

[14] The expression $a{:}b$ means a far address with segment = a, offset = b.

[15] One domestic virus, Chucha, was already mentioned in section 1.2.3–4. Apparently, this is the first Soviet virus program. It is harmless. Under certain conditions the saying "Khochu chuchu" [I want a cookie] appears on the screen. After this the computer refuses to work until the user types the word "chucha" [cookie]. [Transl. note: There is an American version of this virus named Cookie Monster, after the Sesame Street character. It works the same way as Chucha].

[1] *MS DOS 3.30 Technical Reference,* Microsoft Press 1984–1988.

[2] G. L. Landsberg, Institute of High Energy Physics Preprint 90-122, (in Russian), Protvino, 1990.

[3] A. A. Chizhov, *In the world of personal computers* (in Russian) 1, 121 (1988).

[4] N. N. Bezrukov, *Computer Virology, Part 1* (in Russian), KNIGA, Kiev, 1989.

[5] I. Sh. Karasik, Interkomp'yuter No. 2, 14 (1989).

[6] I. Sh. Karasik, Interkomp'yuter No. 1, 39 (1990).

[7] I. Sh. Karasik, *PC World* (in Russian) No. 3, 127 (1989).

[8] R. M. Greenberg, Byte **14(6)**, 275 (1989).

[9] J. McAfee, Datamation **35(4)**, 29 (1989).

[10] P. J. Denning, Am. Sci. **76**, 236 (1988).

[11] M. Eliot, Science **240(4849)**, 133 (1988).

[12] F. Cohen, Computers and Security **7**, 167 (1988).

[13] V. Bonchev, *Komp'yuter za vas* (in Bulgarian) **3-4**, 8 (1989).

[14] "New viral strains take hold," Computing 4 (December 15, 1988).

[15] *Computer Viruses: A High-Tech Disease,* MI 1988.

[16] J. McAfee and C. Haykes, *Computer Viruses, Worms, Data-Diddlers, Killer Programs and Other Threats to the System: What They Are, How They Work and How to Defend Your PC or Mainframe*, St. Martin, N.Y., 1989.

[17] F. Cohen, Computers and Security **8(8)** (1989).

[18] A. K. Dewdney, Scientific American **252(3)**, 14 (1985).

[19] A. K., Dewdney, *ibid.* **250(5)**.

[20] J. F. Shoch and J. A. Hupp, Commun. ACM **25**, 172 (1982).

[21] A. Solomon, PC World **11**, 166 (1988).

[22] D. Crowford, Commun ACM **32**, 780 (1989).

[23] Byte **14(9)**, 19 (1989).

[24] A. K. Dewdney, *In the World of Science* (in Russian) **5**, 82 (1989).
[25] In the World of Personal Computers (in Russian) **1**, 122 (1988).
[26] *Technical Reference for the IBM Personal Computer XT*. Part Number 6936763.
[27] *Chemistry and life* (in Russian) 7 (1989).
[28] Jorge Luis Borges, *Prose of Various Years* (in Russian) Raduga, M. 1984.
[29] L. B. Okun', *The Physics of Elementary Particles* (in Russian) Nauka, M. 1988, pp. 177–178.
[30] N. N. Bezrukov, Interkomp'yuter **2**, 40 (1990).
[31] M. V. Chizhov, Joint Institute for Nuclear Research Preprint R11-90-313, (in Russian) Dubna 1990.
[32] I. Sh. Karasik, Interkomp'yuter **2**, 40 (1990).
[33] PC Magazine, 193 (April 1989).

Translated by C. Gallant