# US-5027273-A Method and operating system for executing programs in a multi-mode microprocessor

(1) TECHNICAL FIELD  
(2) This invention relates to improved methods of executing computer programs in a multi-mode microprocessor and improved operating systems for use with such microprocessors.  
(3) BACKGROUND ART  
(4) Newly designed microprocessors may include enlarged memory addressing facilities and revised architecture which result in enhanced capabilities. When such microprocessors are used in new computer systems, they often produce computers which are functionally superior to their predecessors due to these enhanced capabilities. Despite any functional advantages a new computer may have over its predecessors, a computer employing an improved microprocessor may not be a commercial success. Computer programs, sometimes referred to as "software," are microprocessor specific. Therefore, when a computer employing a new microprocessor is introduced into the marketplace, there is generally little or no software which can run on it. Existing software, written for previous microprocessors, is incompatible with the new computer. As a result, sales of such new computers will often be sluggish until consumers see that adequate software is available for the computer. Additionally, consumers with libraries of software for existing computers may be reluctant to purchase new computers which would require them to invest in all new software. This problem is often compounded by the fact that software writers and publishers are reluctant to produce software for a new microprocessor until sales of computers incorporating the microprocessor are sufficient to create a relatively large group of potential purchasers of the software. This "wait and see" attitude on the part of both consumers and software writers can jeopardize the success of a new microprocessor and computers using the microprocessor.  
(5) Designers of new microprocessors sometimes attempt to solve this problem by designing a new microprocessor such that it will operate in two modes. In a first mode, the microprocessor will emulate a prior microprocessor and run existing programs written for the prior microprocessor. In a second mode, the microprocessor will make full use of its enhanced capabilities. Such a design will enable manufacturers of computer systems using the microprocessor to advertise that the entire body of existing programs written for the prior microprocessor will run on their computer, thereby (in theory) stimulating computer sales to a point where software writers will begin to write programs designed to run in the new enhanced mode.  
(6) One such microprocessor is the Intel 80286, which is manufactured by the Intel Corporation of Santa Clara, Calif. The design and operation of the Intel 80286 is described in detail in a publication entitled "iAPX 286 Programmer's Reference Manual Including the iAPX 286 Numeric Supplement," which is available from the Intel Corporation and is hereby incorporated by reference.  
(7) The Intel 80286 (hereinafter "80286") operates in two modes. In a first mode, called the "real mode," the 80286 emulates the architecture of Intel's previous 8086, 8088 microprocessor family, which is used in the IBM PC and compatible computers, for example. Thus, computers which incorporate the 80286 microprocessor, such as the IBM PC/AT, can run existing 8086 programs written for the IBM PC and compatible computers.  
(8) In a second mode, called the "protected mode," the 80286 architecture provides enlarged memory addressing capability, enhanced multi-tasking support features, and a sophisticated protection scheme.  
(9) Although the real mode will run existing 8086 programs, there are limitations associated with the real mode. First, it limits the amount of physical memory which can be addressed to 1 megabyte. (In some computers, such as the IBM AT, the amount of physical memory available for programs has been further reduced to 640K.) Second, the real mode does not provide memory relocation, a desirable feature for multi-tasking. Third, the real mode provides no memory protection scheme, a feature needed for multi-tasking and network environments where user or task interference could be devastating.  
(10) Because of the limitations of the real mode, the 80286 was not designed to allow frequent switching from one mode to the other. The 80286 is initialized in the real mode and can be switched to the protected mode by means of an instruction provided by the 80286. No method or instruction is provided by the 80286 to switch from protected mode to real mode. To return to real mode from protected mode, it is necessary to reset the microprocessor. Thus, the designers of the 80286 contemplated that it would be used in one mode or the other, with real mode operation being kept separate from the protected mode operation, thereby isolating protected mode programs from the unprotected environment of the real mode.  
(11) Unfortunately, such isolation is undesirable from an efficiency standpoint. For efficient operation, the operating system, or "DOS", of a microcomputer incorporating the 80286 should be able to run a mixture of real and protected mode programs in a multi-tasking environment.  
(12) DISCLOSURE OF THE INVENTION  
(13) It is an object of the present invention to provide improved methods of operating a multi-mode microprocessor that will enable a mixture of programs designed to run in the various modes of the microprocessor to be efficiently executed in a multi-tasking environment.  
(14) It is another object of the present invention to provide such methods that will, in alternate preferred embodiments, maximize the capabilities of the individual operating modes.  
(15) It is another object of the present invention to provide such methods that will only require the addition of minimal hardware to existing systems.  
(16) It is another object of this invention to provide an improved operating system for computers using multi-mode microprocessors.  
(17) It is another object of the present invention to provide, in alternate preferred embodiments, an improved system design for use with multi-mode microprocessors having a protected mode and an unprotected mode that will provide enhanced protection when operating in the unprotected mode.  
(18) It is another object of this invention to provide preferred embodiments of such methods which will enhance the multi-tasking capability of microprocessors such as the Intel 80286.  
(19) It is another object of the present invention to provide preferred methods designed for use with computers such as the IBM PC/AT which utilize the Intel 80286 which will optimize the operation of such computers in a multi-tasking, mode switching environment.  
(20) These and other objects of the invention, which will become more apparent as the invention is described more fully below, are obtained by providing an improved method of switching modes to execute a mixture of programs in a multi-mode microprocessor. In preferred embodiments of the present invention designed for use with microprocessors which must be reset to switch from some modes to others, mode switching is preferably performed by activating the reset hardware in as efficient a manner as possible. After the microprocessor is reset, improved boot-up software will determine whether the reset was triggered under software control (indicating a mode switching reset), in which case the normal initialization routines of the boot-up software are bypassed. During reset procedures, special provisions are preferably made to handle direct memory access and interrupts.  
(21) Preferred embodiments of the present invention preferably include at least portions of the operating system, including device drivers and interrupt service routines, that can be executed in all modes. For microprocessors wherein a common method of memory addressing is not used in all modes, alternate preferred embodiments of the present invention provide an improved method of selecting the base addresses for the operating system subroutines to enable multi-mode addressing. A preferred embodiment designed for use with the Intel 80286 and microprocessors with similar architecture includes the steps of selecting real memory segment base values that are in a format compatible with the protected mode mapping architecture and configuring the protected mode descriptor tables to produce a resulting base address identical to that obtained in real mode. Device drivers, interrupt service routines, and portions of the operating system that are frequently used in both modes are thus placed in real memory at locations selected in this manner.  
(22) Preferred embodiments for use with the Intel 80286 microprocessor also preferably include an operating system subroutine that will examine the address of the I/O location designated by a device driver and produce a 32-bit (segment:offset) pointer which will address the desired memory location in the current mode. When the system is in protected mode, the subroutine will program the GDT or LDT to achieve this result. When the system is in real mode, the subroutine will preferably generate real mode addresses using internal diagnostic instructions to cause the 80286 to address memory locations above 1 megabyte although in real mode. Alternately, information intended for storage at memory addresses above 1 megabyte can be temporarily stored in a buffer at addresses below 1 megabyte while the system is in real mode, and then transferred to the desired memory location above 1 megabyte when the system switches to protected mode.  
(23) Preferred embodiments for use with the Intel 80286 microprocessor also preferably include steps which are designed to eliminate compatibility problems between 8086 programs and the 80286. Depending upon the nature of the system, software modifications or the addition of an auxiliary hardware element to disable the effect of address line A20 are provided.  
(24) Preferred embodiments include means for handling existing real mode programs which store the address of their own interrupt handling routines into the hardware interrupt vector table. Special code enables the DOS to mode switch to real mode as required by such interrupt handler routines and switch back to protected mode to continue execution of the interrupted program.  
(25) Alternate preferred embodiments designed for use with programs that hook interrupt vectors include means for eliminating the problems caused by such programs in a multi-tasking environment. The DOS includes a dispatcher that monitors the hardware vector table to detect hooks by application programs and transfers control to the interrupt handler routines of such application programs at appropriate times. In one preferred embodiment, interrupt vectors are moved to new locations within the hardware interrupt table to facilitate operation of the dispatcher.  
(26) Alternate preferred embodiments may also include techniques for enlarging the amount of memory available to programs in the real mode. First, 64K of the DOS is positioned in memory at location 1 megabyte. Additionally, portions of the DOS which are comparatively large and infrequently used or relatively slow are placed in memory above 1 megabyte and used only in protected mode. The mode switching techniques of the present invention enable such DOS code to be accessed by real mode programs by switching into and out of protected mode as necessary to perform the requested operation.  
(27) Auxiliary protection hardware may also be provided in alternate preferred embodiments to provide enhanced protection when running real mode programs. I/O masking hardware can be provided to check each I/O operation attempted by the CPU against a list of valid I/O addresses. Memory protection hardware can also be provided to check each memory operation attempted by the CPU against a list of authorized addresses stored in an auxiliary RAM.