DOWNLOAD PDF MEMORY MANAGEMENT AND MULTITASKING BEYOND 640K

Chapter 1 : Formats and Editions of Memory management and multitasking beyond K [calendrierdelascien

*Memory Management and Multitasking Beyond K [Lenny Bailes, John Mueller] on calendrierdelascience.com *FREE* shipping on qualifying offers. This text examines how conventional RAM can be maximized using new DOS 5 features, TSR programs, and various network and device drivers that increase standard memory capabilities.*

It inherited this limit from the bit external address bus of the Intel The scarcity of software compatible with the protected mode no standard DOS applications could run in it meant that the market was still open for another solution. Programs had to be written in a specific way to access expanded memory. A first attempt to use a bank switching technique was made by Tall Tree Systems with their JRAM boards [2] , but these did not catch on. The first publicly available version of EMS, version 3. The final version of EMS, version 4. The companies planned to launch the standard at the Spring COMDEX , with many expansion-card and software companies announcing their support. Thus, entire programs could be switched in and out of the extra RAM. EEMS also added support for two sets of mapping registers. This allowed a primitive form of DOS multitasking. The caveat was, however, that the standard did not specify how many register sets a board should have, so there was great variability between hardware implementations in this respect. EMS functions are accessible through software interrupt 67h. Given the price of RAM during the period, up to several hundred dollars per MiB , and the quality and reputation of the above brand names, an expanded memory board was very expensive. Motherboard chipsets[edit] Later, some motherboard chipsets of Intel -based computers implemented an expanded memory scheme that did not require add-on boards, notably the NEAT chipset. Typically, software switches determined how much memory should be used as expanded memory and how much should be used as extended memory. Device drivers[edit] An expanded-memory board, being a hardware peripheral, needed a software device driver , which exported its services. Such a device driver was called expanded-memory manager. Its name was variable; the previously mentioned boards used REMM. Software emulation[edit] Beginning in , the built-in memory management features of Intel processor freely modeled the address space when running legacy real-mode software, making hardware solutions unnecessary. Expanded memory could be simulated in software. The first software expanded-memory management emulation program was CEMM , available in September as a utility for the Compaq Deskpro Software expanded-memory managers in general offered additional, but closely related functionality. Certain emulation programs, colloquially known as LIMulators, did not rely on motherboard or features at all. This was programmatically easy to implement, but performance was low. This technique was offered by AboveDisk from Above Software and by several shareware programs. Decline[edit] Expanded Memory usage declined in the s. These and similar developments rendered Expanded Memory an obsolete concept.

DOWNLOAD PDF MEMORY MANAGEMENT AND MULTITASKING BEYOND 640K

Chapter 2 : CiNii Books - DOS beyond K

*Memory Management and Multitasking Beyond K/Book and Disk [Lenny Bailes, John Mueller] on calendrierdelascience.com *FREE* shipping on qualifying offers. This simple, straightforward introduction to PC memory and memory management shows how to use graphical interfaces without a large increase in memory.*

PCs of that era had become much more powerful than, say, an Amiga But the Amiga offered a flat memory address, while a DOS program could only access memory using cumbersome 64 KiB segments. And to add insult to injury, there was this strange KiB memory limitation. No matter how much physical memory you had in your box, the utter most important Conventional Memory was limited to KiB! The truth is of course a little more complicated than that. Thus, in order to access more memory, a bigger address space is split into multiple segments. The way the generates addresses from segments is really peculiar. Instead of using a register to identify a segment and another one to indicate an address inside it, a new segment begins every 16 bytes. As segments are 64 KiB long and there is no memory protection, it means that segments overlap. A huge amount in This address space had to be shared between the RAM, the video memory, and the various peripherals. These two zones should be to be big enough to fulfill their purposeâ€¦ And Microsoft? It never imposed any limitation! So the infamous K memory ceiling was due to IBM! The is of course fully compatible with the And it provides some form of memory protection. Butâ€¦ In order to retain compatibility with the programs and OSes written for the , all these shiny features are only accessible when running in the newly introduced Protected Mode. When operating in Real Mode a. No memory protection and no access to more than 1 MiB of memory! The Real Mode is the default operating mode of the when starting up. Once loaded, the OS can switch to Protected Mode and let the user reap the benefits. Alas MS-DOS, along with its drivers and its existing applications, had been written when no PC came with anything else than a or a Unfortunately, their developers made some bold assumptions on that matter. Modifying MS-DOS to operate fully in Protected Mode would have been an extensive work and would have broken the compatibility with most applications and drivers! A solution might have been to switch back and forth from Protected Mode. But entering Protected Mode on the is a one way ticket: Hacking the way to more memory Fortunately, some rather hacky schemes were devised to allow DOS programs accessing more memory. Otherwise, the PC as a computing platform would have failedâ€¦ Understanding these schemes requires some naming conventions. Conventional Memory is the precious first KiB of addressable memory. And that is why freeing a mere KiB of it was a worthy battle. Upper Memory Area refers to the area that was reserved by IBM for the video adapter and the various peripherals. Note that this area is only reserved and is not always entirely consumed! It is only naturally accessible on latter Intel CPUs. This is the memory pool which access is required for more advanced computing purposes. So the last segment exceeds the 1 MiB memory limit by almost 64 KiB! On a this is a non-problem. Indeed, its 20 pin address bus makes impossible to access more than 1 MiB of memory. All generated addresses that exceed 1 MiB are aliased to the very first addresses. But the is equipped wit a 24 pin address bus. So it can access this memory block, even in Real Mode! This caused compatibility issues with some programs that used the last segment, expecting it to loop back to the first addresses. A hardware assisted bank switching mechanism and an appropriate DOS driver are then used to access any location in extended memory when requiring access to those four segments. It potentially gives the applications access to the whole 16 MiB address space, even in real mode. Is there any drawback? It requires dedicated hardware support on the motherboard and the use of a driver consuming a few KiB of previous conventional memory. And programmers cannot easily manipulate data structures bigger than 16 KiB. Quoting Bill Gates himself: It does not require any dedicated hardware. It also frees the programmer from being caped to 16 KiB of continuous memory. If there are not enough free blocks available for both to be available, the user has to choose which standard he will enable in the boot configuration files. His judgment will be based on the programs he intends to launch. A hidden segment register exists among them, which can be modified to access memory addresses beyond 1 MB, even in real mode. This trick is only used on the as latter CPUs do not require it. Enter the When it designed the , Intel now understood well that and DOS compatibility would be its main asset. Neat features were also added. This opens the way for modern OSes such as Linux. The Virtual mode, a sub-mode of Protected Mode, in which Real Mode program runs in a kind of virtual machine. The monitor itself operates in protected mode so advanced memory usage such as paging and isolation are possible. An instruction to quickly switch between Protected Mode and Real Mode on the fly. XMS is now easily usable on a ! The still relies on a segmented memory model. But being a 32 bit CPU, the segment can be 4 GiB long, giving the same facilities as a flat memory model. But it is still awkward to use. In order to fulfill its role, a DPMI host has to perform mode switching, allocate extended memory, allocate conventional memory, control the interrupt subsystem, communicate with real mode programs, and read or write certain CPU control registers. Although it is quite possible to code a DPMI host on a , most of them require a Thanks again to its fast mode switching and its Virtual mode! In order to avoid switching back to Real Mode too often and save performance, DOS Extenders also handle frequent hardware interrupts, such as those generated by a mouse. With a 32 bit DOS Extender, a is able to run true 32 bit programs and to allocate huge blocks of memory. This also ends the time when the conventional memory pool was the fastest available. Applications built on DOS Extender thus require far less free Conventional Memory, easing the painful process of crafting custom autoexec. A few years later, Windows 95 made those memory headaches a thing of the past as MS-DOS became progressively irrelevant.

DOWNLOAD PDF MEMORY MANAGEMENT AND MULTITASKING BEYOND 640K

Chapter 3 : USA - Multischeme memory management system for computer - Google Patents

Note: Citations are based on reference standards. However, formatting rules can vary widely between applications and fields of interest or study. The specific requirements or preferences of your reviewing publisher, classroom teacher, institution or organization should be applied.

Technical writer, fiction author, system administrator, web and embedded developer, and philosopher. History is a bit murky on this. For the sake of comparison, the calculator in Windows 10 uses Stranger Things Would you believe me if I told you there was an active community that still used and developed software for this antiquated platform? Your first response might be: Industrial control programs, point of sale systems, and scientific tools are just some of the categories of applications still running under DOS as we speak. Over the years, most of these applications have been ported to other systems or rewritten entirely, but a few remain. A real-time operating system is principally characterized by predictable latency in software and hardware requests. Since DOS has a minimal API with no inherent multitasking, the stability of latency across operating system calls is remarkably stable. Nostalgic Video Games Modern video games are amazing. VR headsets are driving this forward in an incredible way, enabling players to become fully immersed in game environments. Fueled mostly by nostalgia and programming curiosity, developers are writing games that run on old systems, including PCs that originally came with DOS. David Murray , better known on YouTube as The 8-bit Guy , has well over half a million subscribers who tune in each week to dive deep into the land of nostalgic computer hardware. He even wrote a real-time strategy game for the Commodore 64 titled PlanetX2. The project was such a success that he sold out of his physical stock and is planning a sequel for the DOS platform. You may even have a computer in your closet or basement that can boot it right now. First, a word about licensing. Even if you do legitimately own a copy and I suspect many of you do , installing it, even on a virtual machine, can prove tricky. Within a few weeks, Pat Villani and Tim Norman joined the project. In just a few short months, version 0. You might be surprised at how little RAM and disk space allocations are presented. Installing FreeDOS is as simple as booting from the ISO in your virtual machine you will be prompted either at boot or during virtual machine creation for the ISO file and following the prompts. The default option, Jemmex with no EMS is just fine for a standard boot, especially for development purposes. Your FreeDOS system is technically ready to go, albeit bare. To use it, run: You have two options here â€" install all of the development tools, or just install everything. To navigate the categories and packages, use the arrow keys. Whichever option you pick, make sure that the development category is selected.

Chapter 4 : Windows x - Wikipedia

Study Aid for Super Learning and Memory: Alpha Binaural Beats for Study, Focus, Memory.

The memory management system includes a conventional memory handler implemented in hardware for managing memory addresses below a fixed limit, for example four gigabytes; and a plurality of memory handlers implemented in software for managing memory addresses greater than four gigabytes. A programmable memory range detectors is associated with each software implemented memory handler. Memory handlers are selected by addressing the different memory address ranges programmed into the memory range detectors. The memory range detectors associated with the software implemented memory handlers are also prioritized so that lower priority memory range detectors are disabled when a higher priority memory range detector receives an address included within its memory range, thereby resolving conflicts which may otherwise occur when address ranges overlap. The range of addresses associated with the lowest priority memory range detector may be defined to include all memory addresses, thereby establishing the lowest priority memory range detector and its associated software implemented memory handler as defaults for memory addresses greater than four gigabytes. Description The present invention relates to a method and system for managing system memory within a computer system and, more particularly, to a system employing multiple memory management techniques for managing different memory address ranges within system memory. However, most computer processes or programs utilize; a logical or virtual addressing system to address system memory. A logical address is an address expressed as a location relative to a beginning or base location for a process, thus a process does not require use of the same physical memory addresses each time it executes. The size of the logical address space can be smaller than, equal to, or larger than the size of the physical address space. Logical memory having an address space which can be resigned to or translated physical memory is known as virtual memory. Most microprocessors in use today support virtual addressing and also provide the capability of developing multitasking, multiuser, and distributed processing systems. These advanced features are supported through a hardware device known as a memory management unit, abbreviated as MMU. Memory management units generally provide the following functionality: Support for dynamic memory allocation, providing for the efficient management of the physical address space; Support for virtual memory implementation, providing for the management of the virtual address space; Mapping of virtual addresses to physical addresses; Memory protection and task security in some cases ; Processing of page faults or bus fault exceptions, wherein addressed code or data not currently resident in main memory is called into memory and the associated page table is updated to allow task execution to complete; and Supporting sharing of code and data in main memory by multiple processes. As indicated above, one of the primary functions of an MMU is to receive virtual addresses and map them into physical addresses. Several mapping techniques are commonly employed, such as paging, segmentation, and segmented paging. Paging, possibly the most widely supported mapping technique, involves the partitioning of memory into equal fixed-size chunks known as frames, and the division of each process into the same size chunks known as pages. Pages are allocated to frames which are distributed throughout the memory. In order to keep track of the physical location of each memory page, the operating system maintains a page table. The page table includes a number of entries, referred to as page table entries, each of which shows the frame location for each page of a process. Each process operates with logical addresses which include a page number and relative address within the page. In order to access a memory page, the page number is used to address an entry in the page table in order to obtain the frame location. In a virtual memory implementation, it is not necessary that an entire task be placed into main memory for execution. Generally, only a small number of pages or segments discussed below for a task need be resident in main memory at any moment. When the system encounters a portion of a task not resident in memory, the mapping mechanism will generate a high priority exception referred to as a page fault, bus error exception or segment-not-present exception. The operating system is responsive to this page fault to find the required page or segment in secondary storage and move the page or segment into main memory. As task execution progresses, additional pages or segments will be called into main memory as required, replacing pages or segments not currently needed, to permit the task to complete. In a paged mapping system, the operating system must maintain a page table for each active task or process. One disadvantage of a paged mapping system is that the size of the page tables can become quite large. When multiple processes are active, a significant amount of main memory may be occupied by page tables. Segmentation is another way in which memory can be subdivided. A segment comprises a block of contiguous locations within the logical address space, which is placed in its whole into main memory when required during task execution. Segments are of large size than data pages; but like data pages, segments are located by means of a entries in a segment table, the segment table entries having data access bits associated therewith. Because of the larger size of segments versus pages, and further because the size of segments may vary, memory fragmentation problems can occur. Fragmentation results form the swapping of segments into an out off memory creates large gaps in system memory between areas occupied by active segments, the gaps not containing enough contiguous locations to accommodate all segments which may be called into main memory. Although paging alone, and segmentation alone, each have weaknesses, the two concepts can be combined to produce a mapping system which minimizes these weaknesses while obtaining some of the advantages of both paging and segmentation. Main memory is also partitioned into page frames of equal size. Utilizing this scheme, the pages of a segment placed into main memory need not occupy contiguous locations within main memory. A segmented-paging scheme requires the operating system to maintain both a segment table as well as a page table, and two levels of translations are required to compute a physical address. It should be noted that many improvements and modifications to the mapping schemes summarized above are possible. Unique mapping schemes may be developed to produce optimal performance for specific applications or systems. Many of these mapping schemes can be implemented in software as well as in MMU hardware. It should also be noted that the MMUs incorporated into or utilized with many popular microprocessors which support virtual addressing limit the virtual and physical address space to four gigabytes. Thus a memory management system supporting virtual address spaces greater than four gigabytes, and providing optimal memory performance for differing applications is desired. It is another object of the present invention to provide such a memory management system which provides multiple memory managers or handlers for different virtual address ranges identified by a user. It is yet another object of the present invention to provide a new and useful multischeme computer memory management system combining conventional hardware based memory management with additional software based memory handlers. SUMMARY OF THE INVENTION There is provided, in accordance with the present invention, a multischeme memory management system for a computer systems comprising a plurality of memory handlers; and a plurality of memory range detectors, a memory range detector corresponding to each one of the plurality of memory handlers, each one of the memory range detectors being connected to receive memory addresses the computer system. Thus, memory handlers are selected by addressing specified memory address ranges. The described embodiment includes a conventional memory handler implemented in hardware for managing memory addresses below a fixed limit, for example four gigabytes; and a plurality of memory handlers implemented in software for managing memory addresses greater than four gigabytes. The memory range detectors associated with the software implemented memory handlers are programmable to be responsive to different memory ranges specified by a computer system user. The range of addresses associated with the lowest priority memory range detector may be defined to include all memory addresses, thereby establishing the lowest priority memory range detector and its associated software implemented memory handler as defaults for memory addresses above the fixed limit. The above and other objects, features, and advantages of the present invention will become apparent from the following description and the attached drawings. The computer system is part of a multiprogramming system in which the part of memory allocated to users is subdivided to accommodate multiple processes. Main memory is partitioned into fixed length divisions known as data pages In the embodiment shown, each page consists of 4K bytes of data. The address of each page is stored in respective entries in a page table which also resides in main memory Additionally, memory includes a segment table for accommodating segmented paging memory mapping techniques. Memory management unit includes an address translation cache , a segment base register , a status register , an error bit processing circuit , and a microcontroller A first address bus connects memory management unit to microprocessor , while a second address bus connects memory management unit to memory Error bit processing circuit is connected to main memory through address bus and receives certain hardware control bits therefrom. Status register is connected to microprocessor by a control line and provides a bus error signal over line to microprocessor Status register is also connected to microprocessor by bus over which the microprocessor may read the contents of register The exemplary system as described to this point is a conventional system employing a MMU interposed between the system processor and main memory The construction and operation of the system and MMU is well understood by those skilled in microprocessor based system technologies. The logic shown in the simple block diagram of FIG. Conventional hardware MMU is retained for support of virtual addresses up to four gigabytes. The logic includes sixteen memory handlers through , implemented in software, in addition to conventional hardware handler Sixteen programmable range detectors through , corresponding to memory handlers through , respectively, are connected to receive virtual addresses from address bus In the embodiment shown, range detector is fixed to detect all of memory, and points to a hard-coded address for memory handler Detector and handler thereby provide a known memory manager default. The remaining range detectors through are programmable and prioritized. This method of prioritizing detectors will allow multiple detections of a single range of addresses, but will result in only one handler selection. An additional range detector is connected to address bus for detection of virtual address within the conventional four gigabyte range. Conventional hardware MMU is enabled, and the remaining detectors through and their corresponding handlers through are disabled, for virtual addresses within the four gigabyte range. Conventional hardware MMU and range detectors through are individually enabled through use of a control register It can thus be seen that there has been provided by the present invention a new and useful multischeme memory management system for large memory computer systems, combining traditional hardware based memory management with a versatile software based memory management scheme. The invention is intended for use with large memory systems, eliminating the need for the large tables required by traditional memory managers. Unique memory handlers are associated with different virtual address ranges identified by a user. Memory handlers are selected by detecting which of several possible ranges of memory an address is included in. Although the presently preferred embodiment of the invention has been described, it will be understood that various changes may be made within the scope of the appended claims. A memory management system for a computer system, comprising: The memory management system in accordance with claim 1, further comprising: The memory management system in accordance with claim 2, wherein: The memory management system in accordance with claim 2, further comprising:

Chapter 5 : Full text of "DOS beyond K"

Auto Suggestions are available once you type at least 3 letters. Use up arrow (for mozilla firefox browser alt+up arrow) and down arrow (for mozilla firefox browser alt+down arrow) to review and enter to select.

A memory management system for a computer system, comprising: The memory management system in accordance with claim 1, further comprising: The memory management system in accordance with claim 2, wherein: The memory management system in accordance with claim 2, further comprising: The present invention relates to a method and system for managing system memory within a computer system and, more particularly, to a system employing multiple memory management techniques for managing different memory address ranges within system memory. However, most computer processes or programs utilize; a logical or virtual addressing system to address system memory. A logical address is an address expressed as a location relative to a beginning or base location for a process, thus a process does not require use of the same physical memory addresses each time it executes. The size of the logical address space can be smaller than, equal to, or larger than the size of the physical address space. Logical memory having an address space which can be resigned to or translated physical memory is known as virtual memory. Most microprocessors in use today support virtual addressing and also provide the capability of developing multitasking, multiuser, and distributed processing systems. These advanced features are supported through a hardware device known as a memory management unit, abbreviated as MMU. Memory management units generally provide the following functionality: Support for dynamic memory allocation, providing for the efficient management of the physical address space; Support for virtual memory implementation, providing for the management of the virtual address space; Mapping of virtual addresses to physical addresses; Memory protection and task security in some cases ; Processing of page faults or bus fault exceptions, wherein addressed code or data not currently resident in main memory is called into memory and the associated page table is updated to allow task execution to complete; and Supporting sharing of code and data in main memory by multiple processes. As indicated above, one of the primary functions of an MMU is to receive virtual addresses and map them into physical addresses. Several mapping techniques are commonly employed, such as paging, segmentation, and segmented paging. Paging, possibly the most widely supported mapping technique, involves the partitioning of memory into equal fixed-size chunks known as frames, and the division of each process into the same size chunks known as pages. Pages are allocated to frames which are distributed throughout the memory. In order to keep track of the physical location of each memory page, the operating system maintains a page table. The page table includes a number of entries, referred to as page table entries, each of which shows the frame location for each page of a process. Each process operates with logical addresses which include a page number and relative address within the page. In order to access a memory page, the page number is used to address an entry in the page table in order to obtain the frame location. In a virtual memory implementation, it is not necessary that an entire task be placed into main memory for execution. Generally, only a small number of pages or segments discussed below for a task need be resident in main memory at any moment. When the system encounters a portion of a task not resident in memory, the mapping mechanism will generate a high priority exception referred to as a page fault, bus error exception or segment-not-present exception. The operating system is responsive to this page fault to find the required page or segment in secondary storage and move the page or segment into main memory. As task execution progresses, additional pages or segments will be called into main memory as required, replacing pages or segments not currently needed, to permit the task to complete. In a paged mapping system, the operating system must maintain a page table for each active task or process. One disadvantage of a paged mapping system is that the size of the page tables can become quite large. When multiple processes are active, a significant amount of main memory may be occupied by page tables. Segmentation is another way in which memory can be subdivided. A segment comprises a block of contiguous locations within the logical address space, which is placed in its whole into main memory when required during task execution. Segments are of large size than data pages; but like data pages, segments are located by means of a entries in a segment table, the segment table entries having data access bits associated therewith. Because of the larger size of segments versus pages, and further because the size of segments may vary, memory fragmentation problems can occur. Fragmentation results form the swapping of segments into an out off memory creates large gaps in system memory between areas occupied by active segments, the gaps not containing enough contiguous locations to accommodate all segments which may be called into main memory. Although paging alone, and segmentation alone, each have weaknesses, the two concepts can be combined to produce a mapping system which minimizes these weaknesses while obtaining some of the advantages of both paging and segmentation. Main memory is also partitioned into page frames of equal size. Utilizing this scheme, the pages of a segment placed into main memory need not occupy contiguous locations within main memory. A segmented-paging scheme requires the operating system to maintain both a segment table as well as a page table, and two levels of translations are required to compute a physical address. It should be noted that many improvements and modifications to the mapping schemes summarized above are possible. Unique mapping schemes may be developed to produce optimal performance for specific applications or systems. Many of these mapping schemes can be implemented in software as well as in MMU hardware. It should also be noted that the MMUs incorporated into or utilized with many popular microprocessors which support virtual addressing limit the virtual and physical address space to four gigabytes. Thus a memory management system supporting virtual address spaces greater than four gigabytes, and providing optimal memory performance for differing applications is desired. It is another object of the present invention to provide such a memory management system which provides multiple memory managers or handlers for different virtual address ranges identified by a user. It is yet another object of the present invention to provide a new and useful multischeme computer memory management system combining conventional hardware based memory management with additional software based memory handlers. SUMMARY OF THE INVENTION There is provided, in accordance with the present invention, a multischeme memory management system for a computer systems comprising a plurality of memory handlers; and a plurality of memory range detectors, a memory range detector corresponding to each one of the plurality of memory handlers, each one of the memory range detectors being connected to receive memory addresses the computer system. Thus, memory handlers are selected by addressing specified memory address ranges. The described embodiment includes a conventional memory handler implemented in hardware for managing memory addresses below a fixed limit, for example four gigabytes; and a plurality of memory handlers implemented in software for managing memory addresses greater than four gigabytes. The memory range detectors associated with the software implemented memory handlers are programmable to be responsive to different memory ranges specified by a computer system user. The memory range detectors associated with the software implemented memory handlers are also prioritized so that lower priority memory range detectors are disabled when a higher priority memory range detector receives an address included within its memory range, thereby resolving conflicts which may otherwise occur when address ranges overlap. The range of addresses associated with the lowest priority memory range detector may be defined to include all memory addresses, thereby establishing the lowest priority memory range detector and its associated software implemented memory handler as defaults for memory addresses above the fixed limit. The above and other objects, features, and advantages of the present invention will become apparent from the following description and the attached drawings. The computer system is part of a multiprogramming system in which the part of memory allocated to users is subdivided to accommodate multiple processes. Main memory is partitioned into fixed length divisions known as data pages In the embodiment shown, each page consists of 4K bytes of data. The address of each page is stored in respective entries in a page table which also resides in main memory Additionally, memory includes a segment table for accommodating segmented paging memory mapping techniques. Memory management unit includes an address translation cache , a segment base register , a status register , an error bit processing circuit , and a microcontroller A first address bus connects memory management unit to microprocessor , while a second address bus connects memory management unit to memory Error bit processing circuit is connected to main memory through address bus and receives certain hardware control bits therefrom. Status register is connected to microprocessor by a control line and provides a bus error signal over line to microprocessor Status register is also connected to microprocessor by bus over which the microprocessor may read the contents of register The exemplary system as described to this point is a conventional system employing a MMU interposed between the system processor and main memory The construction and operation of the system and MMU is well understood by those skilled in microprocessor based system technologies. The logic shown in the simple block diagram of FIG. Conventional hardware MMU is retained for support of virtual addresses up to four gigabytes. The logic includes sixteen memory handlers through , implemented in software, in addition to conventional hardware handler Sixteen programmable range detectors through , corresponding to memory handlers through , respectively, are connected to receive virtual addresses from address bus In the embodiment shown, range detector is fixed to detect all of memory, and points to a hard-coded address for memory handler Detector and handler thereby provide a known memory manager default. The remaining range detectors through are programmable and prioritized. This method of prioritizing detectors will allow multiple detections of a single range of addresses, but will result in only one handler selection. An additional range detector is connected to address bus for detection of virtual address within the conventional four gigabyte range. Conventional hardware MMU is enabled, and the remaining detectors through and their corresponding handlers through are disabled, for virtual addresses within the four gigabyte range. Conventional hardware MMU and range detectors through are individually enabled through use of a control register It can thus be seen that there has been provided by the present invention a new and useful multischeme memory management system for large memory computer systems, combining traditional hardware based memory management with a versatile software based memory management scheme. The invention is intended for use with large memory systems, eliminating the need for the large tables required by traditional memory managers. Unique memory handlers are associated with different virtual address ranges identified by a user. Memory handlers are selected by detecting which of several possible ranges of memory an address is included in. Although the presently preferred embodiment of the invention has been described, it will be understood that various changes may be made within the scope of the appended claims.

Chapter 6 : Expanded memory - Wikipedia

Title / Author Type Language Date / Edition Publication; 1. Memory management and multitasking beyond K: 1.

The following presents books about DOS utilities such as compression, data recovery, memory management, among other topics. Realise that books in other categories may contain these topics as well. Each aspect of likely problems is carefully discussed and laid out for the reader. The under-workings of DOS are discussed and how they relate to the error messages regarding file storage and other operations. Then a detailed troubleshooting step-by-step is given to assist the reader. The books finishes up with a list of DOS viruses many of which still lurk on discs and the Internet , suggested software, and a glossary. A must book for anyone that wants to know more detail on how DOS works with files and how malware and mal-users! There are also sections on batch techniques and DOS command tips, plus a wealth of utilities on the accompanying disc that are still very useful to the DOS afficiando. Although this is diluted by modern DOS versions and by myriads of utilities for DOS now available, Norton Utilities are still very usable in the 21st century. Preparation of a hard drive is discussed including partitioning, formatting and installing DOS. Then it goes on to talk about files, directories and paths before discussing DOS batch files and menu systems. The book finishes out with disc security and optimisation. It covers hard drive basics, directory organisation with sample structures and file management. Also covered are backups and drive maintenance, and it has a great little section on batch files. Also included is a list of shareware and freeware - much of which should still be available today via the Internet. This edition details the workings of the PC in great death. As such, it is aimed at the advanced user, although intermediates would still get much out of it. A must read for anyone that wants to know more about his PC and the role it plays. Iomega Drivers for DOS by? EN A manual for those using Bernouli drives specifically, and for anyone using zip cartridge drives in general, this booklet is a handy reference. It shows how to set up an Iomega disc system, how to format the discs, how to make them bootable, and how to integrate with DOS. Instructions are given for incorporating the system into a networked setup. He spells out many methods for optimising memory both automatically and manually and then discusses RAM drives and disc caching. A chapter is devoted to after-market managers, and then setup scenarios are discussed. A glossary rounds out this very useful book. After-market managers are discussed along with task switching and multitasking in the DOS environment. A disc is included. An older book, but one that still holds valuable information for those wishing to recover data. It discusses bad file and directory entries, tree structure problems, and partition and FAT problems. Each is given suggested solutions using Norton Utilities. Each utility is listed in alphabetical order with a one-to-five-page explanation and switches usage. A nice, fast review for users of Norton Utilities. Then each utility is listed by alphabetical section with an explanation regarding difference between the 4. A how-to format of using that particular utility for various purposes is then laid out. Plain instructions and screen shots aid the user in implementing very some powerful commands. Although written for initial versions of the utilities package, it still provides lots of useful information that can be projected to later versions. Even though improved compression methods exist, PKZIP or its equivalents are ingrained in the culture of computers. It is so dependable and accurate that I use PKZIP for all my backups, file transfers among removable media, Internet webpage file updates, and all e-mail attachments consisting of multiple files. A manual from the manufacturer, this book is included here because so few publications exist on this de-facto standard for data compression. Some of the switches options could be better explained, but anyone that wants to use or gain more power via PKZIP should have this book. This has something for any DOS user regardless of skill level. It also includes computer basics, with many tips for better operation and suggestions for saving money. Another must-have book for your DOS library. A manual from Quarterdeck, the makers of the popular memory management software. This details the installation, setting up, usage and diagnosis of QEMM version 8. This book is recommended for anyone using QEMM, even other versions, because what is here can teach the user a wealth about memory managing his DOS setup to obtain its

DOWNLOAD PDF MEMORY MANAGEMENT AND MULTITASKING BEYOND 640K

maximum performance. Productivity gains often come from your ability to work faster with your computer, not from the electronics. At the least, the information given is an excellent background into how computers and their components work and how they all interact. A final section discusses in detail some add-on software that will boost performance for those using older systems. Although only covering up to Version 7, this is still a very valuable publication even for users of later versions. This one is more general than Using QEMM see farther back , but it describes the various types of memory and memory utilities - both those that come with DOS and after-market ones. Then he moves on to how to use those utilities to get the best performance from your system. This is a recommended book for an all-round overview of memory for the beginner, but a more in-depth look for the intermediate or advanced user. An included disk has eight popular shareware products for DOS and Windows. Zipping for Beginners by? These include multiple-file zipping, spanning discs, self-extracting archives, and viewing the compressed file, plus all their relevant unzipping methods. A disc is included with shareware versions of each program. Be sure to check for links to some of the programs mentioned in this publications reference.

Chapter 7 : k Really is Enough for Anyone â€“ Sourcerer Blog

Lenny Bailes is the author of Memory Management and Multitasking Beyond k (avg rating, 0 ratings, 0 reviews, published), Windows 98 Revealed.

Chapter 8 : Lenny Bailes (Author of Memory Management and Multitasking Beyond k)

This is a great book for hobbyists. This book describes PC hardware, memory management (including QEMM), and operating system setup. Operating systems covered include DOS, Windows , OS/2, Desqview and touches on Desqview-X. Describes how memory works, and how these operating systems use memory.

Chapter 9 : DOS beyond K : Forney, James : Free Download, Borrow, and Streaming : Internet Archive

First General Announcement Http://wwwfbogov/, we would like to show you a description here but the site wont allow us. Inghroidahogov (j1/manpower & personnel), federal human resources.