APPLIED PC INTERFACING, GRAPHICS AND INTERRUPTS pdf

1: Microprocessor Design/Microprocessors - Wikibooks, open books for an open world

Applied PC Interfacing, Graphics and Interrupts provides an understanding of how the specification of component parts of a PC affects overall system performance, and it also develops an applications-led approach to computer interfacing. In his book, William Buchanan takes a programming language-independent approach to computer interfacing.

Purpose[edit] The network controller implements the electronic circuitry required to communicate using a specific physical layer and data link layer standard such as Ethernet or Wi-Fi. The NIC allows computers to communicate over a computer network, either by using cables or wirelessly. The NIC is both a physical layer and data link layer device, as it provides physical access to a networking medium and, for IEEE and similar networks, provides a low-level addressing system through the use of MAC addresses that are uniquely assigned to network interfaces. The lower right-most card is an early wireless network card, and the central card with partial beige plastic cover is a PSTN modem. Network controllers were originally implemented as expansion cards that plugged into a computer bus. The low cost and ubiquity of the Ethernet standard means that most new computers have a network interface controller built into the motherboard. Newer server motherboards may have multiple network interfaces built-in. The Ethernet capabilities are either integrated into the motherboard chipset or implemented via a low-cost dedicated Ethernet chip. A separate network card is typically no longer required unless additional independent network connections are needed or some non-Ethernet type of network is used. A general trend in computer hardware is towards integrating the various components of systems on a chip , and this is also applied to network interface cards. An Ethernet network controller typically has an 8P8C socket where the network cable is connected. These define a standard receptacle for media-dependent transceivers, so users can easily adapt the network interface to their needs. LEDs adjacent to or integrated into the network connector inform the user of whether the network is connected, and when data activity occurs. The NIC may use one or more of the following techniques to indicate the availability of packets to transfer: Polling is where the CPU examines the status of the peripheral under program control. NICs may use one or more of the following techniques to transfer packet data: This removes load from the CPU but requires more logic on the card. In addition, a packet buffer on the NIC may not be required and latency can be reduced. Performance and advanced functionality[edit] An ATM network interface. That way, taking the application locality into account results in higher overall performance, reduced latency and better hardware utilization, resulting from the higher utilization of CPU caches and fewer required context switches. Accessing local and remote memory without involving the remote CPU. Controlling access to local resources such as control registers and memory. It is primarily used with high-speed network interfaces, such as Gigabit Ethernet and 10 Gigabit Ethernet , for which the processing overhead of the network stack becomes significant. This kind of functionality is usually referred to as user-level networking.

APPLIED PC INTERFACING, GRAPHICS AND INTERRUPTS pdf

2: Interrupts - BIOS Central

EMBED (for www.enganchecubano.com hosted blogs and www.enganchecubano.com item tags).

A linear memory page is a page that is addressed with a selector and an offset in either the real or protected mode. A physical memory page is a page that exists at some actual physical memory location. For example, linear memory location H could be mapped into physical memory location H, or any other location, with the paging unit. This means that an instruction that accesses location H actually accesses location H. Each memory page is 4K bytes long. Paging allows the system software to be placed at any physical address with the paging mechanism. Three components are used in page address translation: EXE, the extended memory manager, uses the paging mechanism to simulate expanded memory in extended memory and to generate upper memory blocks between system ROMs. The Page Directory The page directory contains the location of up to page translation tables. Each page translation table translates a logic address into a physical address. The page directory is stored in the memory and accessed by the page descriptor address register CR3 see Figure 17â€" Control register CR3 holds the base address of the page directory, which starts at any 4K-byte boundary in the memory system. In a virtual mode system, each DOS partition would have its own page directory. The page directory contains up to entries, which are each four bytes long. The page directory itself occupies one 4K-byte memory page. Each entry in the page directory see Figure 17â€"26 translates the leftmost 10 bits of the memory address. This bit portion of the linear address is used to locate different page tables for different page table entries. The page table address A32â€"A12 , stored in a page directory entry, accesses a 4K-byte-long page translation table. To completely translate any linear address into any physical address requires page tables that are each 4K bytes long, plus the page table directory, which is also 4K bytes long. This translation scheme requires up to 4M plus 4K bytes of memory for a full address translation. Only the largest operating systems support this size address translation. Many commonly found operating systems translate only the first 16M bytes of the memory system if paging is enabled. This includes programs such as Windows. This translation requires four entries in the page directory 16 bytes and four complete page tables 16K bytes. The page table directory entry control bits, as illustrated in Figure 17â€"26, each perform the following functions: D Dirty is undefined for page table directory entries by the microprocessor and is provided for use by the operating system. A Accessed is set to a logic 1 whenever the microprocessor accesses the page directory entry. Both bits combine to develop paging priority level protec- tion for level 3, the lowest user level. P Present, if a logic 1, indicates that the entry can be used in address translation. A not present entry can be used for other purposes, such as indicating that the page is currently stored on the disk. The Page Table The page table contains physical page addresses, accessed to translate a linear address into a physical address. Each page table translates a 4M section of the linear memory into 4M of physical memory. The format for the page table entry is the same as for the page directory entry refer to Figure 17â€" The main difference is that the page directory entry contains the physical address of a page table, while the page table entry contains the physical address of a 4K-byte physical page of memory. The other difference is the D dirty bit , which has no function in the page directory entry, but indicates that a page has been written to in a page table entry. Figure 17â€"27 illustrates the paging mechanism in the microprocessor. The paging mechanism functions in the following manner: The 4K-byte long page directory is stored as the physical address located by CR3. This address is often called the root address. One page directory exists in a system at a time. In the virtual mode, each task has its own page directory, allowing different areas of phys- ical memory to be assigned to different virtual tasks. The upper 10 bits of the linear address bits 31â€"22 , as determined by the descriptors described earlier in this chapter or by a real address, are applied to the paging mechanism to select an entry in the page directory. This maps the page directory entry to the leftmost 10 bits of the linear address. The page table is addressed by the entry stored in the page directory. This allows up to 4K page tables in a fully populated and translated system. An entry in the page table is addressed by the next 10 bits of the linear address bits 21â€" The page table entry contains the actual physical address of the 4K-byte memory page. The rightmost 12 bits of the linear address bits 11â€"0 select a location in the memory page. The paging mechanism allows the

physical memory to be assigned to any linear address through the paging mechanism. For example, suppose that linear address H is selected by a program, but this memory location does not exist in the physical memory system. Because this section of physical memory does not exist, the operating system might assign an existing physical memory page such as Hâ€”FFFH to this linear address range. In the address translation process, the leftmost 10 bits of the linear address select page directory entry H located at offset address H in the page directory. Linear address bits 21â€”12 select an entry in this page table that corresponds to a 4K-byte memory page. This first entry contains the physical address of the actual memory page, or Hâ€”FFFH in this example. Take, for example, a typical DOS-based computer system. The memory map for the sys- tem appears in Figure 17â€” Note from the map that there are unused areas of memory, which can be paged to a different location, giving a DOS real mode application program more memory. This section of the memory could be used by DOS, so that the total application-memory area is K instead of K. Software to accomplish this translation and initialize the page table directory, and page tables required to set up memory, are illustrated in Example 17â€”4. Note that this procedure initializes the page table directory, a page table, and loads CR3. It does not switch to protected mode and it does enable paging. Note that paging functions in real mode memory operation.

APPLIED PC INTERFACING, GRAPHICS AND INTERRUPTS pdf

3: Parallel port - Wikipedia

Microprocessors[edit] Microprocessors are the devices in a computer which make things happen. Microprocessors are capable of performing basic arithmetic operations, moving data from place to place, and making basic decisions based on the quantity of certain values. The components of a PC computer. Part number 3 is the CPU. Types of Processors[edit] Wikipedia has related information at Microprocessor The vast majority of microprocessors can be found in embedded high microcontrollers. Historically, microprocessors and microcontrollers have come in "standard sizes" of 8 bits, 16 bits, 32 bits, and 64 bits. These sizes are common, but that does not mean that other sizes are not available. Some microcontrollers usually specially designed embedded chips can come in other "non-standard" sizes such as 4 bits, 12 bits, 18 bits, or 24 bits. The number of bits represent how much physical memory can be directly addressed by the CPU. In some circumstances, these are different; for instance, many 8 bit microprocessors have an 8 bit data bus and a 16 bit address bus. General purpose microprocessors are typically the kind of CPUs found in desktop computer systems. We will discuss some of the other types of processor units available: General Purpose A general purpose processing unit, typically referred to as a "microprocessor" is a chip that is designed to be integrated into a larger system with peripherals and external RAM. These chips can typically be used with a very wide array of software. Wikipedia has related information at Digital signal processor DSP A Digital Signal Processor, or DSP for short, is a chip that is specifically designed for fast arithmetic operations, especially addition and multiplication. DSPs also have special address generation units that can manage circular buffers, perform bit-reversed addressing, and simultaneously access multiple memory spaces with little to no overhead. They also support zero-overhead looping, and a single-cycle multiply-accumulate instruction. They are not typically more powerful than general purpose microprocessors, but can perform signal processing tasks using far less power as in watts. Embedded Controller Embedded controllers, or "microcontrollers" are microprocessors with additional hardware integrated into a single chip. Programmable State Machines The most simplistic of processors, programmable state machines are a minimalist microprocessor that is designed for very small and simple operations. PSMs typically have very small amount of program ROM available, limited scratch-pad RAM, and they are also typically limited in the type and number of instructions that they can perform. PSMs can either be used stand-alone, or more frequently they are embedded directly into the design of a larger chip. Wikipedia has related information at Graphics processing unit Graphics Processing Units Computer graphics are so complicated that functions to process the visuals of video and game applications have been offloaded to a special type of processor known as a GPU. GPUs typically require specialized hardware to implement matrix multiplications and vector arithmetic. GPUs are typically also highly parallelized, performing shading calculations on multiple pixels and surfaces simultaneously. Types of Use[edit] Microcontrollers and Microprocessors are used for a number of different types of applications. People may be the most familiar with the desktop PC, but the fact is that desktop PCs make up only a small fraction of all microprocessors in use today. We will list here some of the basic uses for microprocessors: Signal Processing Signal processing is an area that demands high performance from microcontroller chips to perform complex mathematical tasks. Signal processing systems typically need to have low latency, and are very deadline driven. An example of a signal processing application is the decoding of digital television and radio signals. Real Time Applications Some tasks need to be performed so quickly that even the slightest delay or inefficiency can be detrimental. These applications are known as "real time systems", and timing is of the utmost importance. An example of a real-time system is the anti-lock braking system ABS controller in modern automobiles. Throughput and Routing Throughput and routing is the use of a processor where data is moved from one particular input to an output, without necessarily requiring any processing. An example is an internet router, that reads in data packets and sends them out on a different port.

APPLIED PC INTERFACING, GRAPHICS AND INTERRUPTS pdf

Sensor monitoring Many processors, especially small embedded processors are used to monitor sensors. The microprocessor will either digitize and filter the sensor signals, or it will read the signals and produce status outputs the sensor is good, the sensor is bad. An example of a sensor monitoring processor is the processor inside an antilock brake system: This processor reads the brake sensor to determine when the brakes have locked up, and then outputs a control signal to activate the rest of the system. General Computing A general purpose processor is like the kind of processor that is typically found inside a desktop PC. Names such as Intel and AMD are typically associated with this type of processor, and this is also the kind of processor that the public is most familiar with. Graphics Processing of digital graphics is an area where specialized processor units are frequently employed. With the advent of digital television, graphics processors are becoming more common. Graphics processors need to be able to perform multiple simultaneous operations. In digital video, for instance, a million pixels or more will need to be processed for every single frame, and a particular signal may have 60 frames per second! To the benefit of graphics processors, the color value of a pixel is typically not dependent on the values of surrounding pixels, and therefore many pixels can typically be computed in parallel.

APPLIED PC INTERFACING, GRAPHICS AND INTERRUPTS pdf

4: THE AND MICROPROCESSORS:THE MEMORY PAGING MECHANISM. ~ microcontrollers

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.

Centronics[edit] An Wang , Robert Howard and Prentice Robinson began development of a low-cost printer at Centronics , a subsidiary of Wang Laboratories that produced specialty computer terminals. The printer used the dot matrix printing principle, with a print head consisting of a vertical row of seven metal pins connected to solenoids. When power was applied to the solenoids, the pin was pulled forward to strike the paper and leave a dot. To make a complete character glyph , the print head would receive power to specified pins to create a single vertical pattern, then the print head would move to the right by a small amount, and the process repeated. On their original design, a typical glyph was printed as a matrix seven high and five wide, while the "A" models used a print head with 9 pins and formed glyphs that were 9 by 7. While a serial port does so with the minimum of pins and wires, it requires the device to buffer up the data as it arrives bit by bit and turn it back into multi-bit values. A parallel port makes this simpler; the entire ASCII value is presented on the pins in complete form. In addition to the seven data pins, the system also needed various control pins as well as electrical grounds. Wang happened to have a surplus stock of 20, Amphenol pin micro ribbon connectors that were originally used for one of their early calculators. The interface only required 21 of these pins, the rest were grounded or not connected. The connector has become so closely associated with Centronics that it is now popularly known as the "Centronics connector". The host could then send another character. The host had to carefully watch the BUSY line to ensure it did not feed data to the printer too rapidly, especially given variable-time operations like a paper feed. For example, NCR used the pin micro ribbon connector on both ends of the connection, early VAX systems used a DC connector, Texas Instruments used a pin card edge connector and Data General used a pin micro ribbon connector. In theory, the Centronics port could transfer data as rapidly as 75, characters per second. This was far faster than the printer, which averaged about characters per second, meaning the port spent much of its time idle. To improve performance, printers began incorporating buffers so the host could send them data more rapidly, in bursts. This not only reduced or eliminated delays due to latency waiting for the next character to arrive from the host, but also freed the host to perform other operations without causing a loss of performance. Performance was further improved by using the buffer to store several lines and then printing in both directions, eliminating the delay while the print head returned to the left side of the page. Such changes more than doubled the performance of an otherwise unchanged printer, as was the case on Centronics models like the and Vendors soon released printers compatible with both standard Centronics and the IBM implementation. This was accomplished by allowing the data lines to be written to by devices on either end of the cable, which required the ports on the host to be bidirectional. This feature saw little use, and was removed in later revisions of the hardware. Bi-Tronics[edit] As the printer market expanded, new types of printing mechanisms appeared. While the IBM solution could support this, it was not trivial to implement and was not at that time being supported. Other changes in the handshaking protocols improved performance, reaching , cps to the printer, and about 50, cps back to the host. Two other standards have become more popular for these purposes. ECP offers performance up to 2. The first release in included original Centronics mode "compatibility mode" , nibble and byte modes, as well as a change to the handshaking that was already widely used; the original Centronics implementation called for the BUSY lead to toggle with each change on any line of data busy-by-line , whereas IEEE calls for BUSY to toggle with each received character busy-by-character. This reduces the number of BUSY toggles and the resulting interruptions on both sides. A update standardized the printer status codes. In , the EPP and ECP modes were moved into the standard, as well as several connector and cable styles, and a method for daisy chaining up to eight devices from a single port. Any of these issues might cause no or intermittent printing, missing or repeated characters or garbage printing. Some printer models may have a switch or setting to set busy by character; others may require a handshake adapter. It used a DC connector on

the host side and a 50 pin connector on the printer sideâ€”either a DD sometimes incorrectly referred to as a "DB50" or the block shaped M connector; the M was also referred to as Winchester. The Dataproducts interface was found on many mainframe systems up through the s, and many printer manufacturers offered the Dataproducts interface as an option. A wide variety of devices were eventually designed to operate on a parallel port. Most devices were uni-directional one-way devices, only meant to respond to information sent from the PC. However, some devices such as Zip drives were able to operate in bi-directional mode. Printers also eventually took up the bi-directional system, allowing various status report information to be sent. Before the advent of USB , the parallel interface was adapted to access a number of peripheral devices other than printers. One early use of the parallel port was for dongles used as hardware keys which were supplied with application software as a form of software copy protection. Other uses included optical disc drives such as CD readers and writers, Zip drives , scanners , external modems , gamepads , and joysticks. Some of the earliest portable MP3 players required a parallel port connection for transferring songs to the device. Other devices such as EPROM programmers and hardware controllers could be connected via the parallel port. Interfaces[edit] Most PC-compatible systems in the s and s had one to three ports, with communication interfaces defined like this: Logical parallel port 1: Sometimes, printer ports are jumpered to share an interrupt despite having their own IO addresses i. In some cases, the BIOS supports a fourth printer port as well, but the base address for it differs significantly between vendors. In DOS , the parallel printers could be accessed directly on the command line. BAT file to the printer port. Some operating systems like Multiuser DOS allow to change this fixed assignment by different means. There is even an MS-DOS device in path name vulnerability in Windows 95 and 98, which causes the computer to crash if the user types "C: Microsoft Windows still refers to the ports in this manner in many cases, though this is often fairly hidden. Supports both coax and 10 Base-T.

APPLIED PC INTERFACING, GRAPHICS AND INTERRUPTS pdf

5: RFID Interfacing with Microcontroller (AT89C51) using serial interrupt

Reviewer: William T. Neumann This basic introduction to the inner workings of many of the systems available on most IBM-compatible personal computers covers virtually the entire spectrum of devices and interfaces available on a PC.

An RFID system consists of a reader device and a transponder. A transponder or tag has a unique serial number which is identified by the reader. RFID tag is applied to products, individuals or animals to identify and track them through this number. The interfacing has been done through AT89C The identification code of the tag is also displayed on a 16x2 LCD. The free source code for the program is available in C. In this topic, the program has been made more efficient by incorporating a serial interrupt. Different RFID tags work on different frequencies. These tags work within a range of 10 cm. When an RFID tag comes in this range, the reader detects it and sends a unique code of the tag serially. This serial code, consisting of 12 bytes, is received by the microcontroller. In the program, Timer1 is configured for serial communication. Refer Timer programming in The baud rate is set to bps for data transmission. The LCD is initialized to display the code. The serial interrupt is triggered on every reception of one byte of data. Since the identification code of transponder consists of 12 bytes, the flag is also generated 12 times to indicate the byte wise transfer of data. Whenever the serial code is generated by the reader, the interrupt is activated and the data is sent to the receiver pin of microcontroller. A serial level converter is required for AT89C51 to receive these serial signals. The circuit connections are as follows: Receiver1 R1 of MAX has been used for the serial communication.

APPLIED PC INTERFACING, GRAPHICS AND INTERRUPTS pdf

6: applied pc interfacing graphics and interrupts | Download eBook pdf, epub, tuebl, mobi

Applied PC Interfacing, Graphics And Interrupts by W. Buchanan. Using good practical examples, this new book introduces the reader to the component parts of a PC, showing how high and low level languages communicate with them, and how computers communicate with peripherals.

Device drivers are modules that can be plugged into an OS to handle a particular device or category of similar devices. Devices connect with the computer via ports, e. A common set of wires connecting multiple devices is termed a bus. Buses include rigid protocols for the types of messages that can be sent across the bus and the procedures for resolving contention issues. One way of communicating with devices is through registers associated with each port. The data-in register is read by the host to get input from the device. The data-out register is written by the host to send output. The status register has bits read by the host to ascertain the status of the device, such as idle, ready for input, busy, error, transaction complete, etc. The control register has bits written by the host to issue commands or to change settings of the device such as parity checking, word length, or full- versus half-duplex operation. For example, graphics cards still use registers for control information such as setting the video mode. The host repeatedly checks the busy bit on the device until it becomes clear. The host sets the command ready bit in the command register to notify the device of the pending command. When the device controller sees the command-ready bit set, it first sets the busy bit. Then the device controller reads the command register, sees the write bit set, reads the byte of data from the data-out register, and outputs the byte of data. The device controller then clears the error bit in the status register, the command-ready bit, and finally clears the busy bit, signaling the completion of the operation. Polling can be very fast and efficient, if both the device and the controller are fast and if there is significant data to transfer. It becomes inefficient, however, if the host must wait a long time in the busy loop waiting for the device, or if frequent checks need to be made for data that is infrequently there. The CPU has an interrupt-request line that is sensed after every instruction. The CPU then performs a state save, and transfers control to the interrupt handler routine at a fixed address in memory. The CPU catches the interrupt and dispatches the interrupt handler. The interrupt handler determines the cause of the interrupt, performs the necessary processing, performs a state restore, and executes a return from interrupt instruction to return control to the CPU. The interrupt handler clears the interrupt by servicing the device. Note that the state restored does not need to be the same state as the one that was saved when the interrupt went off. See below for an example involving time-slicing. The need to defer interrupt handling during critical processing, The need to determine which interrupt handler to invoke, without having to poll all devices to see which one needs attention, and The need for multi-level interrupts, so the system can differentiate between high- and low-priority interrupts for proper response. These issues are handled in modern computer architectures with interrupt-controller hardware. Most CPUs now have two interrupt-request lines: One that is non-maskable for critical error conditions and one that is maskable, that the CPU can temporarily ignore during critical processing. The interrupt mechanism accepts an address, which is usually one of a small set of numbers for an offset into a table called the interrupt vector. The number of possible interrupt handlers still exceeds the range of defined interrupt numbers, so multiple handlers can be interrupt chained. Effectively the addresses held in the interrupt vectors are the head pointers for linked-lists of interrupt handlers. Interrupts 0 to 31 are non-maskable and reserved for serious hardware and other errors. Modern interrupt hardware also supports interrupt priority levels, allowing systems to mask off only lower-priority interrupts while servicing a high-priority interrupt, or conversely to allow a high-priority signal to interrupt the processing of a low-priority one. At boot time the system determines which devices are present, and loads the appropriate handler addresses into the interrupt table. During operation, devices signal errors or the completion of commands via interrupts. Exceptions, such as dividing by zero, invalid memory accesses, or attempts to access kernel mode instructions can be signaled via interrupts. Time slicing and context switches can also be implemented using the interrupt mechanism. The scheduler sets a hardware timer before transferring control over to a user process. When the timer raises the interrupt request line, the CPU performs a state-save, and transfers control over to the proper interrupt handler, which in turn

runs the scheduler. The scheduler does a state-restore of a different process before resetting the timer and issuing the return-from-interrupt instruction. System calls are implemented via software interrupts, a. The system does a state save and then calls on the proper interrupt handler to process the request in kernel mode. Software interrupts generally have low priority, as they are not as urgent as devices with limited buffering space. Interrupts are also used to control kernel operations, and to schedule activities for optimal performance. For example, the completion of a disk read operation involves two interrupts: A high-priority interrupt acknowledges the device completion, and issues the next disk request so that the hardware does not sit idle. A lower-priority interrupt transfers the data from the kernel memory space to the user space, and then transfers the process from the waiting queue to the ready queue. The Solaris OS uses a multi-threaded kernel and priority threads to assign different threads to different interrupt handlers. This allows for the "simultaneous" handling of multiple interrupts, and the assurance that high-priority interrupts will take precedence over low-priority ones and over user processes. The host issues a command to the DMA controller, indicating the location where the data is located, the location where the data is to be transferred to, and the number of bytes of data to transfer. While the DMA transfer is going on the CPU does not have access to the PCI bus including main memory , but it does have access to its internal registers and primary and secondary caches. DMA can be done in terms of either physical addresses or virtual addresses that are mapped to physical addresses. The latter approach is known as Direct Virtual Memory Access, DVMA, and allows direct data transfer from one memory-mapped device to another without using the main memory chips. Direct DMA access by user processes can speed up operations, but is generally forbidden by modern systems for security and protection reasons. DMA is a kernel-mode operation. Devices differ on many different dimensions, as outlined in Figure A few devices are special, such as time-of-day clock and the system timer. Most OSes also have an escape, or back door, which allows applications to send commands directly to device drivers if needed. Ioctl takes three arguments - The file descriptor for the device driver being accessed, an integer indicating the desired function to be performed, and an address used for communicating or transferring additional information. Operations supported include read , write , and seek. Rather than reading in the entire file, it is mapped to a range of memory addresses, and then paged into memory as needed using the virtual memory system. Access to the file is then accomplished through normal memory accesses, rather than through read and write system calls. This approach is commonly used for executable program code. Character devices are accessed one byte at a time, and are indicated by a "c" in UNIX long listings. Supported operations include get and put , with more advanced functionality such as reading an entire line supported by higher-level library routines. One common and popular interface is the socket interface, which acts like a cable or pipeline connecting two networked entities. Data can be put into the socket at one end, and read out sequentially at the other end. Sockets are normally full-duplex, allowing for bi-directional data transfer. The select system call allows servers or other applications to identify sockets which have data waiting, without having to poll all available sockets. Get the current time of day. Get the elapsed time system or wall clock since a previous event. Set a timer to trigger event X at time T. Unfortunately time operations are not standard across all systems. A programmable interrupt timer, PIT can be used to trigger operations and to measure elapsed time. It can be set to trigger an interrupt at a specific future time, or to trigger interrupts periodically on a regular basis. The scheduler uses a PIT to trigger interrupts for ending time slices. The disk system may use a PIT to schedule periodic maintenance cleanup, such as flushing buffers to disk. Networks use PIT to abort or repeat operations that are taking too long to complete. More timers than actually exist can be simulated by maintaining an ordered list of timer events, and setting the physical timer to go off when the next scheduled event should occur. On most systems the system clock is implemented by counting interrupts generated by the PIT. Unfortunately this is limited in its resolution to the interrupt frequency of the PIT, and may be subject to some drift over time. An alternate approach is to provide direct access to a high frequency hardware counter, which provides much higher resolution and accuracy, but which does not support interrupts. This allows the process to check for available data without getting hung completely if it is not there. Priorities can also play a part in request scheduling. The classic example is the scheduling of disk accesses, as discussed in detail in chapter Buffering and caching can also help, and can allow for more flexible scheduling options. On systems with many devices,

separate request queues are often kept for each device: Speed differences between two devices. A slow device may write data into a buffer, and when the buffer is full, the entire buffer is sent to the fast device all at once. So that the slow device still has somewhere to write while this is going on, a second buffer is used, and the two buffers alternate as each becomes full. This is known as double buffering. This prevents the user from ever seeing any half-finished screen images. Data transfer size differences. Buffers are used in particular in networking systems to break messages up into smaller packets for transfer, and then for re-assembly at the receiving side. To support copy semantics. Now the application can change their copy of the data, but the data which eventually gets written out to disk is the version of the data at the time the write request was made. Buffering and caching are very similar, except that a buffer may hold the only copy of a given data item, whereas a cache is just a duplicate copy of some other data stored elsewhere. Buffering and caching go hand-in-hand, and often the same storage space may be used for both purposes. For example, after a buffer is written to disk, then the copy in memory can be used as a cached copy, until that buffer is needed for other purposes.

7: Pc Interrupts | Download eBook PDF/EPUB

"by William Buchanan | PB | Acceptable - Applied PC Interfacing, Graphics and Interrupts by William Buchanan A readable copy. All pages are intact, and the cover is intact. Pages can include considerable notes-in pen or highlighter-but the notes cannot obscure the text.

8: Network interface controller - Wikipedia

author's successful Applied PC Interfacing, Graphics and Interrupts this book covers all the different aspects of computer systems from low-level hardware (such as processors and interface devices) to high-level software (such as Win32 programming).

9: Operating Systems: I/O Systems

From the Publisher: author's successful Applied PC Interfacing, Graphics and Interrupts this book covers all the different aspects of computer systems from low-level hardware (such as processors and interface devices) to high-level software (such as Win32 programming).

APPLIED PC INTERFACING, GRAPHICS AND INTERRUPTS pdf

Jana (A Tale of Decadence) Hamilton, E. Requiem. Illinois trial evidence Painter X for Photographers Dungeons and dragons 4th edition underdark Cesaire poetry and knowledge Animals with Novel Genes The Queen Who Saved Her People (Arch Books, set of 6) Gladly woulde he learn analog short story Reaching the Voter Our faithful provider Once upon a dream lana del rey piano Bakery shop business plan Master And Lord Of Life Emerging Stock Markets Factbook, 1991 Proceedings of the Twenty-Fourth General Assembly Rosslyn Chapel revealed Life orientation grade 11 exam papers and memos Cesar Chavez Commonwealth Club Address Black and white : a myth GOOD GOODIES (Fawcett Crest Non-Fiction) A historical sketch of the Congregational churches in Massachusetts What is a wormhole? The Original Teachings of Jesus Urban theology as public theology The Art of Spiritual Direction 3. Equipment Set-up and Control Media gratifications research Why was CERCLA enacted? Fascinating world of the nightshades Toyota voxy owners manual Peer-to-peer communication Water wars and the Ganges I was a teenage movie maker First Middle English primer Suspension of clause 2, rule XXIX. A Select Bibliography on the Sociology of Language (Studia composita) Powerful public speaking A place not a place EasTT-West Corridor . 38