DOWNLOAD PDF PROGRAMMING THE COMMODORE 128

Chapter 1 : Programming Books, part 3: Programming the Commodore 64 | The Reinvigorated Programme

Commodore claimed that a BASIC program written for BASIC (Commodore 64) should also work under BASIC (Commodore in Commodore mode) provided that the program does not POKE to or PEEK from memory locations that have different meanings on C64 and C

None of these were present on the C64 which had only two cursor keys, requiring the use of the Shift key to move the cursor up or left. This alternate arrangement was retained on the , for use under C64 mode. A keypad was requested by many C64 owners who spent long hours entering machine language type-in programs. The two processors cannot run concurrently, thus the C is not a multiprocessing system. The shield was equipped with fingers that contacted the tops of the major chips, ostensibly causing the shield to act as a large heat sink. A combination of poor contact between the shield and the chips, the inherently limited heat conductivity of plastic chip packages, as well as the relatively poor thermal conductivity of the shield itself, resulted in overheating and failure in some cases. The SID sound chip is particularly vulnerable in this respect. The most common remedy is to remove the shield, which Commodore had added late in development in order to comply with FCC radio-frequency regulations. The C has three operating modes. C64 Mode is nearly percent compatible with the earlier computer. Selection of these modes is implemented via the Z80 chip. Based on these conditions, it will switch to the appropriate mode of operation. A sprite editor and machine language monitor were added. The screen-editor part of the Kernal was further improved to support an insert mode and other features accessed through ESC-key combinations, as well as a rudimentary windowing feature, and was relocated to a separate ROM. In column mode the editor takes advantage of VDC features to provide blinking and underlined text, activated through escape codes , in addition to the standard Commodore reverse text. A programmer with both a composite and RGB display can use one of the screens as a "scratchpad" or for rudimentary multiple buffer support. The active display can be switched with ESC-X. The back of the Commodore The VDC chip is largely useless for gaming since it has no sprites or raster interrupts. Two new disk drives were introduced in conjunction with the C A dual-disk model was announced but never produced. Later on, the 3. All of these drives are more reliable than the and promise much better performance via a new "burst mode" feature. The drive also has more on-board RAM than its predecessors, making it possible to open a larger number of files at one time. In addition, the C introduces auto-booting of disk software, a feature standard on most personal computers, but absent from Commodore machines up to that point. If the user switches to C64 mode by typing "GO 64", the drive remains in native mode, but if C64 mode is activated by holding the Commodore key down on power-up, the goes into mode which is necessary for software that performs low-level drive access. These commands are holdovers from the BASIC interpreter intended for a planned but never-produced LCD portable computer and had been intended to exit from the BASIC interpreter and to ignore keyboard input during sensitive program execution, respectively. The diskette was included with the computer, which did not include a disk drive. Software had to be made available on Commodore-specific disks formatted using the GCR encoding scheme. In addition, the cartridges only work on early model C64s from and are incompatible with later units. If one is not detected, control is passed to the and C native mode is started. If this happens, it will default to a gray background with brown text. After the kernal routine is finished executing, control is passed back to the Z C64 mode[edit] Photo from the s showing a C set-up with two disk drives and two monitors displaying the independent and column screens. Many users continued to use the inherited from their C64 system as a second drive. The C64 mode can be accessed in one of three ways: C native-mode cartridges are recognized and started by the kernal polling defined locations in the memory map. C64 mode almost exactly duplicates the features of a hardware C The extended keys of the C keyboard may be read from machine language, although the kernal routines only recognize the keys that exist on the C International models of the C use the unmodified C64 font in both modes, since the second half of the character ROM is instead dedicated to the international font containing such things as accented characters or German umlauts. This memory-mapped register, unused in the C64, determines the system clock rate. Since this register is fully functional in C64 mode, an inadvertent write can scramble the column display by

switching the CPU over to 2â€”MHz, at which clock rate the VIC-II video processor cannot produce a coherent display. Fortunately, few programs suffer from this flaw. By using the higher clock rate during the vertical blank period, standard video display is maintained while increasing overall execution speed by about 20 percent. Another feature of the memory management unit is to allow relocation of zero page and the stack. Therefore, if the MMU is programmed to access blocks 2 or 3, all that results is a mirror of the RAM in blocks 0 and 1. Called the Commodore D, this new European model features a plastic chassis with a carrying handle on the side, incorporates a disk drive into the main chassis, replaces the built-in keyboard with a detachable one, and adds a cooling fan. The keyboard features two folding legs for changing the typing angle. According to Bil Herd , head of the Hardware Team a. Working to release two models at the same time had increased the risk for on-time delivery and was apparent in that the main PCB has large holes in critical sections to support the CD case and the normal case concurrently. The CDCR mounting provision is for a 60mm fan. A significant improvement introduced with the DCR model was the replacement of the video display controller VDC with the more technically advanced VDC and equipping it with 64 kilobytes of video RAMâ€”the maximum amount addressable by the device. The four-fold increase in video RAM over that installed in the "flat" C made it possible, among other things, to maintain multiple text screens in support of a true windowing system, or generate higher-resolution graphics with a more flexible color palette. Little commercial software took advantage of these possibilities. Market performance[edit] By January Info reported that "All of those rumors about the imminent death of the C may have some basis in fact". Stating that Commodore wanted to divert resources to increasing 64C production and its PC clones, the magazine stated that "The latest word online is that the last C will roll off the lines in December of ". The War Begins from Interstel had separate versions, and took advantage of column display on the C The vast majority of games simply ran in C64 mode. By contrast, many C64 productivity software titles were ported to the C, including the popular PaperClip and Paperback Writer series. The C was certainly a better business machine than the C64, but not really a better gaming machine, and people who wanted business machines bought IBM PC clones almost exclusively by the time the C was released. The main reason that the C still sold fairly well was probably that it was a much better machine for hobbyist programming than the C64, as well as being a natural follow-on model to owners with significant investments in C64 peripherals.

Chapter 2 : Commodore Basic: Programming Techniques | eBay

Enter your mobile number or email address below and we'll send you a link to download the free Kindle App. Then you can start reading Kindle books on your smartphone, tablet, or computer - no Kindle device required.

History[edit] Commodore took the source code of the flat-fee BASIC and further developed it internally for all their other 8-bit home computers. It was not until the Commodore with V7. However, Microsoft had built an easter egg into the version 2 or "upgrade" Commodore Basic that proved its provenance: The easter egg was well-obfuscatedâ€”the message did not show up in any disassembly of the interpreter. This saved manufacturing costs, as the V2 fit into smaller ROMs. This section needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. August This section possibly contains original research. Please improve it by verifying the claims made and adding inline citations. Statements consisting only of original research should be removed. If a line was prefixed with a line number, it was tokenized and stored in program memory. Lines not beginning with a number were executed by pressing the RETURN key whenever the cursor happened to be on the line. It also had the capability of saving named files to any device, including the cassette â€" a popular storage device in the days of the PET, and one that remained in use throughout the lifespan of the 8-bit Commodores as an inexpensive form of mass storage. Most systems only supported filenames on diskette , which made saving multiple files on other devices more difficult. The device would search for the filename by reading data sequentially, ignoring any non-matching filenames. The file system was also supported by a powerful record structure that could be loaded or saved to files. Commodore cassette data was recorded digitally, rather than less expensive and less reliable analog methods used by other manufacturers. Therefore, the specialized Datasette was required rather than a standard tape recorder. Adapters were available that used an analog to digital converter to allow use of a standard recorder, but these cost only a little less than the Datasette. The LOAD command may be used with the optional parameter ,1 which will load a program into the memory address contained in the first two bytes of the file these bytes are discarded and not retained in memory. If the ,1 parameter is not used, the program will load into the start of the BASIC program area, which widely differs between machines. The PET does not support relocatable programs and the LOAD command will always load at the first two bytes contained in the program file. One was to store often-used floating point values in variables rather than using literal values, as interpreting a variable name was faster than interpreting a literal number. Commodore BASIC keywords could be abbreviated by entering first an unshifted keypress, and then a shifted keypress of the next letter. This set the high bit , causing the interpreter to stop reading and parse the statement according to a lookup table. This meant that the statement up to where the high bit was set was accepted as a substitute for typing the entire command out. However, since all BASIC keywords were stored in memory as single byte tokens, this was a convenience for statement entry rather than an optimization. In the default uppercase-only character set, shifted characters appear as a graphics symbol; e. Most such commands were two letters long, but in some cases they were longer. Some commands had no abbreviated form, either due to brevity or ambiguity with other commands. For example, the command, INPUT had no abbreviation because its spelling collided with the separate INPUT keyword, which was located nearer to the beginning of the keyword lookup table. By abbreviating keywords, it was possible to fit more code on a single line line lengths were usually limited to 2 or 4 screen lines, depending on the specific machine. This allowed for a slight saving on the overhead to store otherwise necessary extra program lines, but nothing more. Such long lines could be difficult to edit. The LIST command displayed the entire command keyword - extending the program line beyond the 2 or 4 screen lines which could be entered into program memory. This is in comparison to other implementations of BASIC which typically have dedicated commands to clear the screen or move the cursor. GOTO, and it was common to write programs with no spacing. This feature was added to conserve memory since the tokenizer never removes any space inserted between keywords: Spaces between the line number and program statement are removed by the tokenizer. Program lines can be 80 characters total not counting the line number on most machines, but machines with 40 column text would cause the line to

wrap around to the next line on the screen, and on the VIC, which had a 22 column display, program lines could occupy as many as four. By using abbreviations such as? The order of execution of Commodore BASIC lines was not determined by line numbering; instead, it followed the order in which the lines were linked in memory. While a program was being entered, BASIC would constantly reorder program lines in memory so that the line numbers and pointers were all in ascending order. However, after a program was entered, manually altering the line numbers and pointers with the POKE commands could allow for out-of-order execution or even give each line the same line number. In the early days, when BASIC was used commercially, this was a software protection technique to discourage casual modification of the program. GOTO and GOSUB statements will search downward from the current line to find a line number if a forward jump is performed, in case of a backwards jump, they will return to the start of the program to begin searching. This will slow down larger programs, so it is preferable to put commonly used subroutines near the start of a program. Most contemporary BASIC implementations used one byte for the characteristic exponent and three bytes for the mantissa. The accuracy of a floating point number using a three-byte mantissa is only about 6. Although Commodore BASIC supports signed integer variables denoted with a percent sign in the range to , in practice they are only used for array variables and serve the function of conserving memory by limiting array elements to two bytes each an array of elements will occupy 10, bytes if declared as a floating point array, but only if declared as an integer array. Denoting any variable as integer simply causes BASIC to convert it back to floating point, slowing down program execution and wasting memory as each percent sign takes one additional byte to store since this also applies to integer arrays, the programmer should avoid using them unless very large arrays are used that would exceed available memory if stored as floating point. SYS clears the screen. As with most other versions of Microsoft BASIC, if an array is not declared with a DIM statement, it is automatically set to ten elements in practice 11 since array elements are counted from 0. Larger arrays must be declared or BASIC will display an error when the program is run and an array cannot be re-dimensioned in a program unless all variables are wiped via a CLR statement. Numeric arrays are automatically filled with zeros when they are created, there may be a momentary delay in program execution if a large array is dimensioned. String variables are represented by tagging the variable name with a dollar sign. Array variables are also considered distinct from simple variables, thus A and A 1 do not refer to the same variable. The size of a string array merely refers to how many strings are stored in the array, not the size of each element, which is allocated dynamically. TI is read-only and cannot be modified; doing so will result in a Syntax Error message. The clock is not a very reliable method of timekeeping since it stops whenever interrupts are turned off done by some kernal routines and accessing the IEC or IEEE port on the PET port will slow the clock update by a few ticks. RND with any number higher than 0 will generate a random number amalgamated from the value included with the RND function and the seed value, which is updated by 1 each time an RND function is executed. RND with a negative number goes to a point in the sequence of the current seed value specified by the number. Garbage collection is automatically invoked any time a FRE function is executed and if there are many string variables and arrays that have been manipulated over the course of a program, clearing them can take more than an hour under the worst conditions. This 1 KB extension to BASIC added a number of disk-related commands, including the ability to read a disk directory without destroying the program in memory. Its features were subsequently incorporated in various third-party extensions, such as the popular Epyx FastLoad cartridge. To load a file to a designated memory location, the filename, drive, and device number would be read by a call: SYS"filename",8; [8] the location would be specified in the X and Y registers: POKE,; [9] and the load routine would be called: It was common practice to increment numbers by some value 5, 10 or to make inserting lines during program editing or debugging easier, but bad planning meant that inserting large sections into a program often required restructuring the entire code. A common technique was to start a program at some low line number with an ON GOSUB jump table , with the body of the program structured into sections starting at a designated line number like , , and so on. If a large section needed to be added, it could just be assigned the next available major line number and inserted to the jump table. In addition, all variables are treated as global variables. Flag variables often needed to be created to perform certain tasks. The even dedicated its second 64k bank to variable storage, allowing values to persist

until a NEW or RUN command was issued. This, along with the advanced screen editor included with Commodore BASIC gave the programming environment a REPL -like feel; programmers could insert and edit program lines at any screen location, interactively building the program. If a programming language was required on these platforms, it had to be loaded separately. This had the effect of overwriting the currently loaded program. Addons like the DOS Wedge overcame this by rendering the directory listing direct to screen memory. Versions and features[edit].

Chapter 3 : blog.quintoapp.comre - New Files

The Commodore , also known as the C, C, C= , or occasionally CBM , is the last 8-bit home computer that was commercially released by Commodore Business Machines (CBM). Introduced in January at the CES in Las Vegas, it appeared three years after its predecessor, the bestselling Commodore

Jack Tramiel opens a typewriter repair shop in the Bronx, New York. Tramiel relocates to Toronto and becames the biggest manufacturer of low cost office furniture in Canada ? Commodore manufactures calculators and digital watches, but gets killed by Texas Instruments. April - Commodore Business Machines Inc. The computer shown is a one-of prototype. During its life, production peaks at 9, units per day. Hi-Toro Incorporated is formed by a group of midwest investors trying to cash in on the video game craze. The name was later changed to Amiga, Incorporated after being confused with the lawn-mower manufacturer, Toro. June - Commodore Business Machines announces the P microcomputer. September - Commodore Business Machines begins shipping the Commodore January - Commodore introduces the SX , the first color portable computer. It incorporates a 5-inch color monitor and one or two 5. May - Commodore ships the Commodore Executive It has a monochrome monitor with column display. They also show the Executive 64, formerly the Commodore SX Commodore debuts the Exactron Stringy Floppy, a high-speed cassette-based data storage device. Tramiel leaves the company and a few months later buys Atari. June - Commodore announces the Commodore It will now feature four built-in programs, not just one. August - Commodore purchases Amiga Corporation. Commodore stops manufacturing the VIC January - Commodore unveils the Commodore Personal Computer. It functions as three computers in one: January - Commodore announces the Disk Drive, for the Commodore Commodore stops production of the Commodore 64 several times during the year, restarting each time based on public demand. January - Commodore announces the Amiga and the Amiga Commodore introduces the Amiga HD and the Amiga January - Commodore announces that 1 million Amiga computers have been sold. It is essentially an Amiga with a Accelerator Board MHz and math coprocessor. June - Commodore ships the Amiga A computer. Commodore announces the Amiga September - Commodore introduces the Amiga December - Commodore introduces the Amiga Commodore International and Commodore Electronics two of the many international components of Commodore Business Machines file for voluntary liquidation. Gateway buys bankrupt Amiga.

Chapter 4 : What is a Commodore ? - Definition from Techopedia

Commodore BASIC Programming Techniques On t.p. the registered trademark symbol "TM" is superscript following Commodore in the title Includes index.

Although a few specifications were not finalized, we found the to be a versatile machine with one of the most powerful BASIC programming languages ever offered in a microcomputer. One of the most imitated trends in personal computing lately has been "integrated software"â€”products that are actually three or more programs in one, like Lotus Now Commodore is introducing a fresh twistâ€”integrated hardware. The deceptively small package contains: A standard Commodore 64 with 64K of Random Access Memory RAM capable of running virtually all existing 64 softwareâ€”estimated at 6, to 10, programs, mostly home and educational. Expandable to K with a RAM disk option, the Commodore also works with all Commodore 64 peripherals as well as a new line of accessories, such as the much faster disk drive. Before the Commodore was announced at the Winter CES, rumors indicated it would simply be an expanded Commodore But the Commodore truly is the near-equivalent of three computers in a single box. Outside, all three computers share the same sleek plastic case and key key-board. Does all this sound confusing? Even when using the Commodore , you can sometimes forget which mode the computer is in. This is especially true of the 64 mode and column mode, which appear virtually identical on screen. Furthermore, some modes let you switch to other modes, but not back again without restarting the machine. And speaking of cold-starts, the Commodore can be switched on in any of its five modes, depending on its state at power-up. Otherwise, it checks the cartridge slot for a Commodore 64 cartridge program. If it finds one, it automatically switches to 64 mode and runs the cartridge. If it finds one, it comes up in mode either 40 or 80 columns and runs the cartridge. If so, it starts up in column mode. Otherwise, it switches to column mode. With so many options, operating the Commodore will take some getting used to. One of the biggest questions about the Commodore is its degree of Commodore 64 compatibility. In fact, Commodore claims the is percent 64 compatible. Our tests showed the had no trouble with a wide range of Commodore 64 programs written in BASIC and machine language. Only one program failed: TurboDisk, a machine language utility which speeds up disk loading by as much as percent, ran fine on a hooked up to a disk drive, but would not work on a connected to the new disk drive. Since the is designed to be much faster than the , it may seem that utilities such as TurboDisk are superfluous anyway. However, keep in mind that the , like the Commodore , is a multimode device. When the is operating in 64 mode, the drive behaves just like a â€”it stores K of data per disk and runs fairly slow. When the computer is switched to mode, the speeds up about percent and becomes a double-sided drive, storing about K per disk. However, the drive we tested would not load our Osborne disk. He also said that production s would be fully compatible, so that programs like TurboDisk should work too. If the final s are not completely compatible, Commodore will run into trouble on another frontâ€”commercial copy protection. Some copy-protection schemes depend on precise timing and certain routines within the disk drive ROMs. Before acquiring a Commodore and drive to run 64 software, it would be a good idea to try loading some commercial disks first to make sure they work. Sprite movement is implemented during a machine-level interrupt, so a few BASIC statements can keep up to eight sprites moving simultaneously while the program performs other tasksâ€”or even stops. To make it easier to define sprites, you can grab any predrawn shape off the screen and store it as sprite data, or design the sprite bit by bit with a built-in sprite editor. Remember that some specifications or syntax rules may have changed by the time the Commodore entered final production. Note that none of these commands adds new capabilities not available with a Commodore 64 and drive; they merely simplify the syntax. Sprite commands not only replace the old-fashioned POKEs, but also offer more options. A number of subcommands let you clear all the sprite data, move a crosshair, turn pixels on and off, and change colors for multicolor sprites. For example, you can define a sprite by first drawing a shape on the screen with various graphics commands, then copy the shape into a string with the SSHAPE command, and finally move the string into the sprite data block with SPRSAV. This is called absolute movement and is like the POKEs used to move sprites to screen positions on the Commodore When the sprite disappears off the screen, it wraps around and reappears at the

opposite end of the screen. This command is useful when you want to send a sprite flying on a predetermined course and speed while your program does other things. Of course, the sprite must be positioned somewhere on the visible screen to begin with. The type parameter lets you detect different kinds of collisions. BUMP type Detects sprite collisions and returns a value corresponding to the sprites involved. Mode 0 is column text default ; 1 is hi-res graphics; 2 is hi-res graphics with a text window; 3 is multicolor graphics; 4 is multicolor graphics with a text window; and 5 is column text RGB only. The text windows are similar to those on Atari and Apple computersâ€"they allow a few lines of text beneath the graphics window on the upper part of the screen. The color parameter defines the color from 1 to The source parameter specifies the color register affectedâ€"0 for the column background, 1 for the graphics mode foreground, 2 for the multicolor graphics mode primary color, 3 for the multicolor graphics mode secondary color, 4 for the column border, 5 for the character color, and 6 for the column background color. BOX 1,10,10,60,60,0,1 draws a filled green box in the upper-left corner of the screen. Draws a dot, line, or figure on the hi-res screens. This defines the default starting point for all the drawing commands. PAINT ,, draws and fills a circle using the default foreground color. The rectangular area of screen data between coordinates corner1 and corner2 is saved in the string variable string. Corner1 and corner2 define the rectangular screen coordinates, and mode specifies how the shape will be plotted. Here are a few examples: SOUND voice,freq,dur,sweep,min,step,wave,width Plays a sound with the selected voice, frequency, and duration. The following parameters are optional. An M in this parameter tells the computer to wait for all voices currently playing to end. This memory is not directly available for programs, though, but is used as a RAM diskâ€"the functions of a disk drive are simulated with the memory chips. This provides faster throughput than a hard disk, but all information is lost when the power is turned off. You need to dump the contents of a RAM disk to a regular disk at the end of each session. The MMU interprets memory addresses even before the microprocessor sees them. It permits you to swap between banks of 64K, but can leave a small portion of memory as common memory. The MMU lets you bank between four 64K banks, and allows multiple banks of K, up to one megabyte of memory. A programmer can set up a series of preset memory configurations and quickly select them by writing to the MMU. The address of the VIC chip can be relocated anywhere within the virtual K memory space. The MMU also controls the fast serial port used with the disk drive and conceivably with other fast peripherals. It determines the clock speed of the , and controls which of the three microprocessors , , Z80A is in control. Maybe someone will find a way to take advantage of this potential multiprocessing capability. It pretends that the K of memory is contiguous and permits five-digit hexadecimal addresses. It makes full use of 80 columns if selected. The monitor works much like 64 Supermon, with commands to assemble, disassemble, fill, go to address, hunt through memory for a hexadecimal string, load, display memory with ASCII equivalents, display registers, save, transfer a block of memory, verify a saved program, exit to BASIC, modify memory, modify registers, and display disk error status. SYS can be followed by four parameters that are transferred into the accumulator, X register, Y register, and status flag register. New Kernal routines support special features of the , including routines for memory management. This reset puts you into the machine language monitor, where you can exit back to BASIC with no harm done. Redefines any of the ten predefined music envelopes for the tune parameter of the PLAY command. The specifies the envelope 0 to 9 , followed by the values for attack, decay, sustain, and release. Wave sets the SID waveform and width selects the width of a pulse waveform. Indeed, a preliminary manuscript for the Commodore System Guide is a stack of single-spaced, typewritten pages two and a half inches thick. There are commands for windowing, switching 64K memory banks, renumbering BASIC programs, deleting ranges of BASIC lines, assigning new definitions to the predefined special function keys, entering the machine language monitor, trapping runtime errors and diverting execution to an error-handling routine at a certain line number, resuming execution after a runtime error, high-lighting errors in BASIC lines, constructing loops without FOR-NEXT, and inserting delay loops. It continues the trend away from low-level instructions such as PEEK and POKEâ€"vestiges of machine languageâ€"and further shields users from intimate contact with the bits and bytes of computer circuitry. Yet, unlike some other personal computers introduced in recent years, the Commodore retains its BASIC as a built-in feature and also provides a machine language monitor for those who want to explore the computer at

every level.

Chapter 5 : Commodore Machine Language, Part 1

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.

The Commodore ended up being wedged between the already hugely successful Commodore 64 and the soon to be huge seller, the Commodore Amiga. So let us take a look at a machine that never quite got the recognition it deserved The Commodore Machine This machine was marketed as the successor to the Commodore 64 which had already succeeded the Vic 20 , it was compatible with all of the existing software including those classic games! One of the main specialities of this machine was the additional Z80 CPU and Video chip, which meant that you could attach two monitors to the computer. You felt like rather special if you used TWO monitors back in ! One screen for playing games, the other for your homework! Thanks to the Z80 chip, not only could you run 2 monitors, you could also select which mode to run the C under during boot up. The downside was that the 2 processors could not run concurrently, so it was not a true multiprocessing system. A multi-mode machine that was truly clever stuff from CBM. What does a Commodore look like? The lack of a numeric keypad was also an issue with some office suite software. Also criticized was the lack of a hardware reset button, which was an essential device when developing assembly language routines. A numeric keypad was also added to the keyboard which was another great feature. It also incorporated a sprite editor very useful when developing arcade games and a machine language monitor. The screen editor was further improved and an all important reset button was added to the system. All of these were brilliant features. Three new disk drives were introduced in conjunction with the C, the , , and 3. You gotta love that technical jargon. The C64 mode was accessed in a number of ways: A little bit of trivia there for you. The Commodore D The D model: The D was the direct successor of the Commodore It had exactly the same characteristics as the C apart from its external case which contained the Commodore floppy disk unit. It did have a more professional look to it, akin to a desktop PC, but also lost some of that well known Commodore appeal. It was also hampered by the release of Bit machines such as the Amiga and the Atari ST which quickly built a large library of text adventures and classic games. A lot of users bypassed the C models and wanted to play the new and exciting Amiga games. It still rankles me to this day that a company such as Commodore that manufactured great home computers is no longer with us. From the early years of the PET right through to the Amiga they were always a major force in the market, and in some cases market leaders. With better marketing Commodore could still have been a major player in the market today

Chapter 6 : Commodore C64 Books : Free Texts : Free Download, Borrow and Streaming : Internet Archive

Commodore The last of Commodore's 8-bit machines (the C) was launched at the Las Vegas Consumer Electronic Show in It was presented as a competitor to the Apple Macintosh and the IBM PC, but things never really panned out this way.

Programs I have implemented for the Commodore Power Assembler version 8. If you know why, please tell me! Bug in Power Assembler version 8. When I assembled using the. BAS pseudo-op, I wanted some data defined using. It should be possible to do this using. Double-Ass another assembler for the C including brief documentation, this one I have not tried myself but it seems less powerful than Power Assembler Memory management Both the Commodore 64 and the Commodore have a bit address bus and can therefore address 64 kB of memory at the same time. The Commodore has more sophisticated memory management than the Commodore Using the MMU, it is possible to set four different pre-set memory configurations. If you try changing one of these three pre-set memory configurations, BASIC may no longer work correctly you may end up in the machine code monitor. The Commodore has two RAM banks bank 0 and bank 1 , each consisting of 64 kB RAM, and the memory configuration also defines which of the two RAM banks that should be switched in the other one is switched out. If memory is shared, it can be defined how much memory that is shared and if memory should be shared at the bottom or at the top of the banks or both. Sharing memory means that if bank 1 is switched in and a memory location in a shared area is accessed, it is the memory location in bank 0 that is accessed instead of in bank 1. This means that if the software tries to access a memory location in these two pages, the access is redirected to a memory location where the lower 8 address bits are the same but the higher 8 address bits are set to another value that has been defined by the user in a register in the MMU. Page 0 contains a lot of variables used by the operating system and page 1 contains the processor stack so the possibility to redirect these pages makes it possible to quickly switch between different sets of variable values and processor stacks. Of course, the memory map for the Commodore is different from that for the Commodore On the C, there are some areas that are particularly suitable for machine code programs. These are some of the major differences in the memory map. These two extra registers are used for new keys on the keyboard more columns in the keyboard matrix and 2 MHz mode. These features are described at other places on this page. It might seem a bit strange that the 2 MHz register is located in the VIC chip considering that the VIC chip does not support 2 MHz mode it results in garbage on the screen but this is the way it is. The two versions of the VIC chip are not interchangeable. With the VIC chip, you can see vertical lines going through the middle of each of the columns of the screen. When I first used a C, I thought that there was something wrong with it but I then found out that this bug exists on all Cs. However, it is more visible on some Cs than on others. This depends on the revision of the VIC chip. The bug also to a certain degree exists with the VIC chip in the C64 but there it is much less visible. Another bug in the VIC chip is that when using raster interrupts, there can sometimes be lots of white dots flashing around. This is visible in some games. The just mentioned bugs in the VIC chip can be pretty annoying and therefore if you want to play C64 games a lot, I would recommend using a real Commodore 64 rather than a Commodore in Commodore 64 mode. So, what about graphics features? The answer to this question was long believed to be no. It was found that as long as this bit is set, one raster line is skipped per clock cycle. The test bit can also be used for more advanced techniques as described in the following sections. On PAL systems, skipping an odd number of raster lines not only increases the screen refresh frequency but also leads to new colours in addition to the 16 usual ones. This is used in the C demo "Risen from Oblivion". It is impressive to see the frogs in lots of different green colours in that demo. It is the best C demo there is and one of the very few there is as well. In , it was found that skipping raster lines makes it possible to accomplish interlace mode, which doubles the vertical resolution, although it causes some flickering. The test bit tricks all have in common that they only work with real Cs, i. The VIC part of "Risen from Oblivion" for example works with most Commodore monitors I have however seen one person reporting that it did not work with his monitor. VIC graphics and character programming One difference between the Commodore 64 and the Commodore when it comes to

graphics is that the C has built-in support for split-screen mode. When split-screen mode is enabled, it means that there is a vertical division of the screen into two different areas. One of the areas uses bit-map mode or multi-colour bit-map mode while the other area uses standard character mode. This is possible to do also on the C64 but there is no built-in support for it so you need to write your own raster-interrupt code to do it there. The screen editor which is a part of the Kernal in the Commodore is raster interrupt-driven in order to make split-screen mode possible. This is a difference compared to the Commodore 64 where the Kernal uses timer interrupts instead of raster interrupts. The C screen editor reads the value of so called shadow registers to update actual registers when raster interrupts occur. If they are written to directly the written value will just be overwritten by the screen editor when the next raster interrupt occurs. Instead, shadow registers should be written to. Let us for example assume that we have a split-screen with standard or multi-colour bit-map mode at the top of the screen and standard character mode at the bottom of the screen. Then, we will get two raster interrupts per screen update. We will get the "normal" interrupt at the top of the screen that we always get but since we are in split-screen mode, the raster compare register is reprogrammed when the interrupt occurs so that we will also get an interrupt further down on the screen where we want the split to occur. As mentioned in the previous paragraph, there is one "normal" interrupt occuring once per screen update and if split-screen mode is enabled, a second interrupt occurs as well further down on the screen during each screen update. However, it is only during the "normal" interrupt that the Kernal operations occuring at an IRQ are done e. If that bit is equal to 1, it is a "normal" interrupt, otherwise not. Then, you can program graphics in exactly the same way as on the Commodore 64, i. The registers have the same addresses as on the Commodore This makes it possible to quickly change the colours of a whole screen. I have come up with this idea for improvement myself. Note that this feature is only possible in C mode. The Commodore does not have this limitation. The C64 does not have pre-defined values for the sprite pointers. The sprite pointers are in the C64 and the C always located as the last 8 bytes of the 1 kB chunk of screen memory. In contrast to what many people believe, most Commodore s contain the older version of the SID chip called My two Cs both contain that version and also other people I have been in contact with who have checked the SID chip version of their Cs have that version. The only C model I know of that contains the newer version called is the Commodore DCR the metal case model that was mainly sold in North America. However, it is possible that late manufactured versions of other C models might also contain the version although I have not encountered any so far. The two versions of the SID chip are not interchangeable without adding or removing other electrical components due to different voltages. Kernal The same 39 Kernal calls as on the Commodore 64 are also available on the Commodore A few of these calls differ from the calls made on the Commodore According to "Das C Buch", it was necessary to make these changes considering some special features of the C, e. In addition to the 39 Kernal calls that also exist on the Commodore 64, there are 19 new Kernal calls that are Commodore specific. There are for example calls for accessing or jumping to memory locations in a selectable RAM bank the C has two RAM banks each consisting of 64 kB as mentioned before , for going to C64 mode, for booting from an autostart floppy, for switching between column and column modes, for programming function keys and for outputting a string of data. Some of the screen editor routines are called when the Esc key followed by another key is pressed but the routines can also be called directly from an assembly program. The routines are not described in "Das C Buch" but there are various other books that describe at least some of the routines. RS The main difference between programming the User Port RS interface device number 2 on the C compared to the C64 is that on the C a fixed range of memory addresses is always reserved for the byte output and input buffers. On the C64, on the other hand, the buffers are allocated in the end of BASIC text memory when an RS channel is opened and de-allocated when it is closed. This is not the case on the C The memory addresses used for RS system variables in zero-page are the same on the C as on the C However, the RS system variables that are not placed in zero-page e. However, when you use more than one raster interrupt per screen update you probably only want to jump to the Kernal interrupt routine once per screen update. Otherwise, the parts of the Kernal that are handled at interrupt level will be handled too often resulting in strange effects such as the cursor blinking faster than usual. Therefore, there is sometimes a need to be able to return from an interrupt without jumping to the Kernal interrupt routine. The following code should be used

for this: For the Commodore 64, these lines should not be present. What the two red lines do is to load a byte from the stack and then store it in the MMU configuration register. This byte was read from the MMU configuration register and stored on the stack by the Kernal when the interrupt occurred. I have found that returning from an interrupt like explained above does not work together with BASIC you end up in the machine code monitor or the computer freezes so do this only when you are programming in assembly! However, as written above, this leads to strange effects if you have more than one interrupt per screen update sprites will move too quickly, music will play too quickly, the cursor will blink too quickly etc. If you know how to solve this, please tell me! The coding of which key that has been pressed is not the same for the Commodore as for the Commodore The value for "no key pressed" also differs. Furthermore, the Commodore has more keys than the Commodore The extra keys on the Commodore keyboard can be used also in Commodore 64 mode as is shown in this assembly code example. On the Commodore , it is possible to change the key definitions, i. Then, modify entries in the table s and change the vector s to point to the table s in RAM. Otherwise, the Kernal will just overwrite the values you have written to the vectors at the next interrupt. Since both of these books are a bit erroneous and contradictory to each other regarding this functionality, I have also had to experiment myself to see how it really works.

Chapter 7 : Downloads - Everything Commodore

Commodore Programmer's Reference Guide Memory Management Unit Programming Information While there is a complete transcription of the Commodore

This series of articles discusses programming the computer in machine language in mode. These articles are directed especially at programmers who need to make the transition from 64 machine language to ML programming. Ground Rules Here are two simple ground rules to keep you out of trouble on the The lower area contains critical system vectors and subroutines; if you change their contents, the system will crash. The is capable of seeing its memory as 16 different banks numbered The term banks is somewhat misleading, since a bank does not represent a separate 64K block of memory. The bank number determines what the sees within various areas. In fact, there are possible memory configurations. Most of these, however, are of little or no use. Commodore has chosen 16 configurations which seem most useful, named the different configurations banks, and identified them with numbers from Figure 1 shows the configuration for bank More about the rest later. Thus, your favorite 64 POKEs to make sound effects and so forth work exactly the same in mode. After all, bank 0 gives you access to all the memory in RAM 0. Figure 2 shows the bank 0 configuration. The computer has lots of memory, but no way to communicate with the outside world. Stay in bank This ensures that your program will work even if some other program has left the machine configured for a different bank. As a courtesy to other programmers and users in general , programs that use other configurations should end by returning the machine to the default bank. Note that there are several unused memory areas available for program storage. Figure 3 also reveals other unused or little-used memory zones. Your System Guide contains additional information. Simply tack them onto the end of the SYS command, separated by commas. The file loads into the same memory area from which it was saved, and BASIC continues with the next command. The DEC function converts a hexadecimal string into a decimal number. It counts the number of 1 bits in any eight-bit number and prints them out in a table. You may not be excited to learn that the number 14 binary contains three 1 bits, while the number 16 binary contains only one, but the program does demonstrate how to pass information from BASIC to machine language and back again. It takes a value from the accumulator A register , counts the 1 bits in the value, and places the result in the X register. A simple additive checksum detects most typing errors. If not, the ML code is safely planted in memory and we can proceed to the job of bit counting. When the machine language program begins to run, the A register will contain that value. The A register goes into variable S and the X register goes into T. Now T contains the bit count. We print the value of J and the bit count T, then go back to do it again.

Chapter 8 : COMMODORE SYSTEM MANUAL Pdf Download.

The Commodore A Hands-On Report. Tom R. Halfhill, Editor. Commodore's new three-in-one machine, the Commodore Personal Computer, should be hitting store shelves in June.

Chapter 9 : COMMODORE PROGRAMMER'S REFERENCE MANUAL Pdf Download.

Learn to Program the Commodore All By Yourself! (used). This is more of a study guide but looks like an excellent course guide. $