

TC-128/64

The Commodore Computer Journal
Issue #35



TC-128/64 - Issue #35

The Commodore Specific Computer Journal since Jan 1986.

Publisher: Parsec, Inc. Editor: John W. Brown

| ARTICLE | AUTHOR | PG |
|---------------------------------------|------------------|-----|
| Cover | | 01 |
| Contents | | 02 |
| Editorial License | John W. Brown | 03 |
| Letters to the Editor | Staff | 06 |
| News, Rumor, & Mayhem | Staff | 06 |
| Dr.Octal's Sharp Operating Tips | | 37 |
| Software Picks | Staff | 42 |
| CLASS(Y) ADS | | n/a |
| Dr.Octal's Q&A Clinic | Staff | 48 |
| | | |
| Epson Stylus 800 Review | Brett Tabke | 07 |
| Fetch It | Mike Gilsdorf | 11 |
| 1571 Write Verify Mode | Dennis J. Jarvis | 17 |
| Learning Machine Language PT2 | Craig T Taylor | 22 |
| A C-65 What? | John W. Brown | 36 |
| CMD Utilities Review | John W. Brown | 31 |
| geoMakeBoot Review | Staff | 41 |
| Font Paint64-128 | Smotherman/Knop | 46 |
| | | |
| Subscription Information | | n/a |
| Legal stuff | | 47 |

Advertisers:n/a

Published, distributed, and copyrighted 1994 with all rights reserved by:
Parsec, Inc. PO Box 111 Salem, MA 01970-0111 USA. ADDRESS
CORRECTIONS SHOULD BE SENT TO THE ABOVE ADDRESS. Parsec is
not responsible for any advertiser's claims, typos, omissions, or misprints. To
contact us: GENIE members can send e-mail to "JBEE", or send a letter by US
mail, no voice calls. All letters acknowledged. We try to answer as many
questions as we can in a following issue. All letters sent become the property of
TC-128/64.

This magazine is dedicated to my mother to whom I gratefully owe all my
successes and failures.

Editorial License

by John Brown

Welcome to our new format for the magazine. What changed between issue 34 and issue 35? Many things, for one, the last widely distributed printed publication supporting the Commodore 64/128, Compute, dropped the Commodore insert and went to a disk only based publication for the Commodore readers. This meant a large drop in circulation to only about 10,000 readers.

Though it was not by much and is cold comfort, at least us C= 8 bitters hung in there. We outlasted all those other computers and that "other" Commodore computer in Compute's pages. Now Compute is all IBM PC clone. I believe there is a version with a Mac insert though I have never seen it.

I guess this signals an end of an era, a mindset, a tradition, the next to last nail in the coffin <GASP>. This pretty much means no major software house will be developing or even consider developing any new games for the C-64 in 1994. Or at least in the USA, since there is no way to hit 50,000+ readers with one ad now.

Our New Format

I decided to bring TC-128/64 monthly, drastically lower the subscription price per issue (1/2), and go to a newer format that will allow up to the minute news and changes.

We have dropped the 3rd class mail rates. The original reason for using 3rd class mail was to lower the cost of mailing TC-128/64 to 4,000+ subscribers who were owed issues from VMI. Unfortunately with 3rd class mail comes true 3rd class treatment. This means, among other things, that the USPS does not deliver (they never "lose" it or "throw" it away ;-) about 3% of the mail for each issue. Multiple this by thousands of users and you have an unending stream of paperwork and complaints.

We are going with a smaller size to eliminate the 30-40 complaints each issue about carriers that folded the issue and companion disk in half, despite the "do not fold" stickers and heavy cardboard mailers.

To go with 1st class mail and hold down end user cost we had to cut down the mailing weight. Which is the main reason we went to this format. The actually printing costs are about 4x higher then the newsprint, but are offset by the lower mailing costs. To mail an issue such as #34 (5.8 oz. min.) by first class mail costs \$2.13+ vs. \$0.98 (3.0 oz.) for this issue.

What people have been saying is they want inexpensive monthly issues, instead of higher cost issues that come out 3-4 times a year. We listened.

A majority of the subscribers have been getting the code 4 subscriptions (1st class with the disk) so we decided to make that the default choice. In these times of dwindling sources of software we realize it is important to bring the newest software to you at an affordable price. \$4 for a disk and a magazine delivered by 1st class mail is both affordable and reasonable.

Most (type-in) programs will be put on the front (commercial) side of the companion disk along with any "value" or "commercial" software. The shareware or public domain programs will be put on the back side of the disk.

Since most subscribers are due to resubscribe between issues 35-38, it was best to make the change now. I expect to be up to speed with the monthly delivery by the end of May for all subscribers. All code 5 and most of the code 3 & 4 subscribers should be monthly before then.

If you receive more than one copy of an issue during 1994, do not worry. We mailed over 30,000 free issues to people in 1993. Because we buy mailing lists from various sources and have found it cheaper not to check the labels against our own database, it is possible for someone to receive 2-3 copies of the same issue if they happen to buy a lot of products from different vendors.

A new magazine, High Tech Entrepreneur

Enclosed you will find a free sample copy of "High Tech Entrepreneur". Since a good percentage of our readers like hardware projects and run businesses on their C-64 or C-128, I thought our readers would be the ideal target market for this type of magazine.

Though, if there was not a copy in your envelope, sorry, we ran out. We could not get enough to cover all of our subscribers. If you decide to subscribe to this magazine, please put "TC-128/64" near the reader response box on the back. If enough people find a magazine such as this of interest, I will consider approaching other people for free offers. Though if you do NOT find the magazine interesting, do NOT subscribe.

There is no sense in wasting HTE's money, time, postage, and paper on something that does not interest you or that you can not use. Myself, I like the ads they carry as much as the articles.

Other good stuff

William E. Gates also publishes a magazine called "Midnight Engineering". I originally subscribed just to get the ads! It is chock full of ads for all sorts of things such as circuit boards, C source code for FAX, "muscle" wires, etc. I soon got hooked on the articles about book on demand printing, starting your own business, patents, the pitfalls of a single owner business, and so on. I still think the best series of articles was on how to protect yourself in the case of a fire, as it applies to your home or home office. That series of articles was worth the subscription price.

Being extremely busy, I never have time to read like I use to. I usually only get subscriptions to magazines that are free to the trade. That way I do not feel bad tossing them into the shredder after reading only a few pages. I subscribe to three magazines, Computer Shopper, Computer Monthly (because I want to advertise in both of the magazines), and Midnight Engineering. This is the only paid magazine subscription I get! In fact, I like it so much I decided to eat the extra postage and include the free sample copy of HTE with TC-128/64. I like Midnight Engineering that much! So, tick off the box for the free sample copy on the reader response card and see for yourself.

Enjoy our new format,

A stylized, handwritten signature in black ink, appearing to be the name 'GENIE' written in a cursive, looped font.

GENIE=JBEE

Letters to the Editor

Your editorial spot on page 3 (issue #33) really hit home. 3 years ago I sold my 64 & "upgraded" to an Amiga 500. I finally unloaded it on someone else just last fall & bought a used 128D w/1581. This is my 6th different computer & I've never had this much fun or enjoyed computing so much before.

Tim George, CA

JBEE: Must be the same person that bought my Amigas!

News, Rumors, and Mayhem!

Hot stuff for 1994, some personal predictions: Pearl Jam, RamLink, 4x4 mini-vans, multi-media, online computer games.

Mice shortage

If you do not know by now, there is a terrible mouse shortage. The mice CMD bought from Europe probably have all been sold by now. The people that bought up all those nifty "Super Earl Mice" are lucky indeed! Rumor has it, the new CMD mouse is not far behind and should be available shortly. CMD posted about a possible track ball, who knows?

GEOVISION INTERNATIONAL and Grady Brown

Grady Brown sent me a letter about GVI being dead, bankrupt, D.O.A. and wanted to know if I would buy GVI. I did mail a reasonable offer for something so buried in debt, even forwarded it to his new address, no answer. Something you might find interesting though. GVI owns the rights to geoMetrix.

The mayhem

Where did GVI get the the mailing list to launch the magazine? Did anyone get a letter informing them geoMetrix had been sold and how their due subscriptions were being honored, or in this case, dishonored? Was it mentioned on the GVI subscription forms or magazine? I never saw it.

A Hacker hacked?

According to a news report from China, Shi Biao, a computer hacker, has been executed as a warning to others contemplating computer crime (think they hanged him). In 1991, supposedly, Biao defrauded the Agricultural Bank of China of around \$200,000 through money transfers. Mayhem (read between the lines): "they missed him in the square with the tank, the first time".

Epson Stylus 800

by Brett Tabke

The Perfect Home Printer?

Over a year ago I embarked on a quest to find a respectable low cost and high quality replacement for my old and well worn MPS-1000 dot matrix printer. I have looked at and studied dozens of printers from dot matrix and ink jets printers to low cost laser printers. What I found was that the quality of high end dot matrix printers was unacceptable and that the initial investment and continuing cost of laser printer consumables were too high, so I centered my search on ink jet printers.

The promise from ink jet manufacturers is laser quality at half the price, though very few ink jet printers deliver on that promise. Output quality from ink jets runs the full gambit from poor to almost laser quality. After deciding to buy an ink jet printer, I located output samples from the following printers: Epson Stylus 800, Nec Jet Mate 1000, Canon Bubble Jet 200/230, Hewlett Packard Desk Jet 500, Brother HJ-400, IBM Exec Jet II, Epson Stylus 300, and the Texas Instruments Micro Marc.

Inkjet Differences

There are fundamental technological differences among ink jet printers. The majority of ink jets use thermal bubble technology first introduced by Hewlett Packard in the original Desk Jet 500 over 6 years ago. Thermal bubble jets work by heating a droplet of ink until it expands and spews through a nozzle onto the page. A bubble does not burst perfectly, there is always a few drops that fly off in random directions. This same effect causes thermal bubble jet printers to produce random droplets of ink when printing. These stray droplets of ink spray (called satellites) are the main cause of poor output from most thermal bubble jet printers.

Of the printers mentioned above, only the Epson Stylus 800 and IBM Exec Jet II do NOT use thermal bubble technology. The IBM inkjet uses a proprietary system that literally pumps ink onto the page. Epson's Stylus 800 uses a Multi Layer ACTuator Head (MACH) (print head) with 48 piezoelectric cells. When a cell full of ink is charged, the cell changes shape to expel the ink. The IBM and Stylus 800 both consistently produce the highest quality output of any ink jet I have seen - these printers rarely produce satellite spray that most thermal ink jets do.

The Exec Jet uses IBM's own PQET resolution enhancement technology that effectively raises the resolution of the printer to 300x600 dpi. Good output from the IBM could pass for laser output most days. The Epson's 360x360 dpi output was very close to the IBM's; but, a tell-tale satellite here and there still effects the Epson. Please note the Epson Stylus 800 uses the MACH head, but, the cheaper priced Epson Stylus 300 uses a standard thermal bubble jet head with mixed results.

After I had narrowed down my search from all of the above, to the IBM and the Epson, I found out that the Exec Jets \$30 ink cartridges are only rated at 400 pages of print and can not be refilled. The Stylus 800 cartridges are rated at 750 pages with a cost between \$12 and \$20 and can be refilled with bulk ink. Additionally, the Epson is of course, 100% Epson compatible with its ESC/P2 emulation; where as, the IBM is only Desk Jet 500 compatible. So I am now the proud owner of an Epson Stylus 800.

Epson Stylus 800 Features

The ESC/P2 compatibility makes the Stylus 800 work with any driver for the entire line of LQ/SQ/EX/FX/RX/LX Epson printers - the only thing left out was laser printer compatibility. Seven fonts for eight character sets are available including Times Roman, Helvetica San Serif, Courier, Prestige, Script, Roman, and San Serif. The 2 Roman and 2 San Serif fonts can be scaled from 8 to 32 points in 2 point increments, each font can then be double width and height, for a whopping 64 point headline print. Each font also has Italic, Outlined, Shadow, and Outlined Shadow type faces. Also available are 256 additional graphic characters (ding bats) ranging from hearts and stars to foreign and legal characters. The pitches available range from pica and elite to 20 characters per inch (cpi) to compressed proportional spacing.

Using with Software

Output from Pocket Writer, Paper Clip, and the Write Stuff has been excellent. However; even using the 66% reduced Epson driver, output from Geos is extremely poor (I have not tried it with the LQ print system from CMD). Other graphics output from Print Shop and Print Master has also been disappointing.

This is the fault of the low resolution dot matrix printer drivers included with the software, not that of the printer.

Consumables

Buying a factory ink cartridge from Epson costs \$20, with a street price of \$16.50 for single cartridges or \$12.50 in quantities of 6. The Stylus 800 uses soybean oil based ink. This ink is all the rage of newspapers now, because it results in highly desirable low run, and low rub off qualities, plus it is 75% biodegradable for you Ecological fans. This type of ink is a little more expensive than normal ink, but, the cost of factory made cartridges is falling fast.

Even though the manual suggests that you do not refill ink cartridges, I have refilled the original cartridge about 10 times with good results. Refilling the cartridge is as easy as rolling back a label, sticking in a 15cc syringe, and injecting ink into a foam sponge. Several mail order companies sell ink in quantity for about .30-.50 cents an ounce (\$7.50 for a 5 oz bottle). A cartridge is rated at 700-850 pages of text. When refilling with bulk ink, per page ink cost is less than 1/10th of a cent.

Selecting the right kind of paper to put in an ink jet printer is the most important aspect of using them. With paper that is too absorbent, the ink will bleed, with paper that is not absorbent enough, the print will smear easily. I have found that the cheapest standard typing paper and the most expensive laid finish paper works the best. Standard Xerographic or laser printer paper does NOT work well at all - it looks like the ink was put on the paper with an eye dropper. I was fortunate to find 16 pound typing paper for \$10.50 a case (5000 sheets) at a new office supply store. This has made my overall per page cost less-than-laser at 1/5th of a cent. On-a-lark, I also tried some ultra expensive, heavy, 24 pound, gloss laid finish, stationary paper with a 25% cotton fiber content, and the results are the best I have seen to date.

Hardware

The hardware end of the Stylus contains 32K RAM buffer, 32K ROM OS, and a 1k Flash RAM memory for storing power up defaults. The control panel contains font selection, economy mode switches, and paper out/low ink indicators. The printer is virtually silent.

Paper Handling

Although, the paper bin is rated at 100 single sheets of 24 pound paper, I put a 150 sheet pack of 16 pound paper in it without a glitch. There is a rear manual feed for envelopes and legal size paper. The manual feed works fairly well, but the slot uses a guide indicator rather than a guide edge and papers often feed slanted using the manual mode.

The recommended paper for the printer is rated from 14 to 24 pounds, but I have run 60 pound card stock through with good results. If there is anything this printer shines at, it is printing on transparencies or glossy material, it is outstanding with both.

Speed

The head carriage motor is a one speed motor. It does not matter whether you are printing proportional, letter quality, or graphics, the speed from left to right is always constant. With a standard pica 80 character per inch (cpi) line, printing speed is about 150 characters per second (cps), a 12 cpi condensed line works out to 300 cps, and the "real world" average works out to be about 2.3 pages per minute or 140+ pages and hour.

Manual

The manual is a classic Epson 'new user' manual that expects you to run the printer with only pre-packaged software drivers. The manual is well produced; but, it fails to give examples of using ESC/P2 commands. The commands are listed in an appendix, but the cryptic syntax and minimal wording will leave even veteran printer users scratching their heads. After calling the Epson toll-free technical support line (a nice touch); I found that Epson was more than willing to sell me a complete programming ESC/P2 guide for only \$29.95 plus shipping.

Pros & Cons

I have not found many faults with the Stylus 800. With a much lower than expected cost of operation, it has been very easy to overlook any annoyances I may have with it. However; I wish it had laser compatibility, and that the sheet feeder was about 200 sheets bigger, and I often find myself reaching for the none existent paper feed knob (electronic feed only). When the printer needs to put itself through a head cleaning cycle, the print quality is temperamental. The cleaning cycle occurs about every 100 pages, and it frequently works out that pages 95-99 start to show a few smudges before it cleans itself.

With a low (falling fast) street price of \$279 and an incredible low cost of consumables, I have found this printer to be an overall joy with which to work. The quality of text output is very close to a low end laser printer. The assured Epson compatibility, low noise level, and rock solid Epson 2 year parts and labor warranty have made this a very successful quest indeed.

FETCHIT

By Michael Gilsdorf

Load and Execute Programs from any Sub-directory
without Typing Pathnames

Overview

Sub-directories make file management much easier by allowing you to group related files together into their own directory. You really come to appreciate them if you ever had to look through one long directory listing for a particular file, or when you try to determine which files belong to which programs. The larger the drive's capacity, the bigger the problem can become. If you avoided using sub-directories in the past because you found them awkward to use, pathnames easy to forget, or tedious to type, then FETCHIT has come to your rescue. It allows you to retrieve and execute files from anywhere on disk without having to type in a string of directory names, SYS numbers, or RUN commands.

How It Works

FETCHIT runs on either a C128 or a C64 computer; however, C64 users will need JiffyDOS. FETCHIT can be used with any Commodore drive, but is intended primarily for large storage devices that support sub-directories. CMD owners of the newer FD and HD drives will especially appreciate FETCHIT's ability to retrieve files from different partitions and sub-directories nested two or more levels deep.

Up to now, most programs that load and execute multiple programs operate on a menu-driven principal. Typically, you load a menu program and search for the program you want from a list of choices. After you select your program, the menu program then loads and executes the desired program. FETCHIT, on the other hand, eliminates the menu by allowing you to pass it the name of the program directly - much like a command pre-processor permits you to pass it parameters in a command string.

More precisely, when you issue a command to execute a program, FETCHIT is loaded and determines the current device number. Then, it parses the command string to extract the name of the program which it looks up in its data statements. A data statement tells FETCHIT where the program is on disk and how to load and execute it. When the name is found, FETCHIT displays the proper commands on the screen, and then loads and executes the program using

the dynamic keyboard technique. If the name is not found, FETCHIT attempts to load and execute the program by using the last data statement (the default entry).

Data Statements

After you type in FETCHIT, enter a data statement at the end of the listing for each program you want FETCHIT to retrieve. A data statement contains six entries. Commas are used to separate each entry and serve as place holders for any entry not used. Like all data statements, an entry should be enclosed in quotes if it contains commas or colons. Also, use caution when entering spaces. Unlike most other lines of BASIC, spaces not enclosed in quotes can be significant. For FD and HD drives, the entries are defined as follows:

Entry #1: Name.

This entry contains the pseudo name you want to use when retrieving the file. It can be different from the actual filename shown in the directory. Usually you will want to choose a name that is short and easy to remember. For example, if the file stored on disk is named "BOOT MY EDITOR" you can refer to it simply as "EDIT" or just "ED".

Entry #2: Partition.

This entry contains the command to change partitions. If you do not want to change partitions before loading a file, leave this entry blank.

Entry #3: Sub-directory.

This entry contains the command to change sub-directories. You should specify the complete sub-directory path from the root directory to where the file is located. Like entry #2, use this entry only if you want to change sub-directories; otherwise, leave it blank.

Entry #4: Filename.

This entry contains the actual name of the file you want to retrieve. It should appear just like it does in the disk directory; however, wild card characters (*?) are permitted. If you are retrieving a file from a different partition and sub-directory, and are not using entries #2 and #3, then you must precede the filename with the complete path including partition number.

Note: Some applications once they are loaded may produce "?FILE NOT FOUND" errors, or save files in the wrong directory. This generally occurs if the partition or sub-directory is not changed when the file was loaded. If in doubt, it is best to use

entries #2 and #3 to change the partition and sub-directory to where the file is located.

Entry #5: Load.

This entry contains a 0 or 1 which tells FETCHIT how to load the program. A zero (or blank) causes a BASIC (relocatable) load, whereas a one is used for machine language (ML) programs.

Entry #6. Command.

This entry contains the command you want to execute after the program is loaded. If you just want to load the program, or if the program automatically executes after it is loaded (as many ML programs do), then leave this entry blank. Typically, this entry will contain the word RUN, or a SYS number, but other BASIC commands such as LIST can be used too.

As an example, suppose you wish to run a program called "BOOT PACMAN C-64" using the pseudo name "PACMAN". If the file is located in partition 2 with a path "//GAMES/ARCADE", the data statement would look like this:

```
DATA "PACMAN", "CP2", "CD//GAMES/ARCADE", "BOOT PACMAN C-64", , RUN
```

If you only wanted to load PACMAN, but did not want to run it or change the current partition or sub-directory, then the data statement would be changed to:

```
DATA "PACMAN", , , "2//GAMES/ARCADE/:BOOT PACMAN C-64", ,
```

When using a 1581 drive, entry #4 must contain only the filename. Any sub-directory name should be placed in entry #3. However, should you need to change sub-directories twice, you can enter the first command in entry #2, and the second in entry #3 as shown below:

```
DATA "PACMAN", "/0:GAMES", "/0:ARCADE", "BOOT PACMAN C-64", 0, RUN
```

Executing FETCHIT.

FETCHIT requires you to use a single command to load and run a program. The syntax is designed so the filename is passed as a parameter to FETCHIT. For example, if you have JiffyDOS (or a compatible DOS wedge), the syntax is:

Editor's note: Because of character set limitations, please replace the at sign "@" with the up arrow when typing in the file names and program data statements. When using JiffyDos, the up arrow key (for load and run) is contained on Function Key 5.

@/=NAME[, #]

where "/" is the name we will give to FETCHIT and "NAME" is the pseudo name of the program we want to execute. The brackets should not be typed. They are used to enclose optional parameters. The "#" represents a load number (0 or 1) which can override the value in entry #5. This allows you to change the way the program is normally loaded and suppress the execution of any command FETCHIT normally would perform in entry #6. For example:

| | |
|-----------|--|
| @/=NAME | load and execute according to the DATA statement |
| @/=NAME,0 | BASIC load only |
| @/=NAME,1 | ML load only |

Before entering the command you need to make sure you are in the root directory (and default partition). If not, FD and HD drive owners can use the alternate command:

@n//:/=NAME[, #] where n is the default partition number (usually 1)

Using a function key programmed to print the first part of the command will save you some keystrokes. Once you have entered a command, the computer will store it in memory and send the string to the drive. The drive acts upon only the first part of the command string which loads and runs FETCHIT. It ignores any characters after the equal sign. FETCHIT, in-turn, loads and executes the program NAME.

If you are using a C128 without JiffyDOS, the command string is a little longer. However, the unit number can be eliminated if you are using device 8 or the OS Plus utility (see TC-128 issue #32).

RUN"/=NAME[,#]",U8

Notice there are already four data statements in FETCHIT that contain the pseudo names "*", "\$", "/", and "?". These names relate to commands that perform the following functions:

```
@/=*      Load and run first program in directory
@/=*,1    ML load first program in directory
@/= $     Load and list directory
@/= $,0   Load directory
@/= /     Load FETCHIT
@/= ?     Load FETCHIT and list data statements for
          viewing/editing
```

After you have finished entering the data for your programs, make sure the last data statement contains the null name "". Then save FETCHIT in the root directory using "/" as its filename. FD and HD owners should be sure to save it in the default partition that comes up after booting. It also might be a good idea to lock FETCHIT to prevent it from being accidentally scratched.

When you acquire a new program, just issue a @/= ? command and add another data statement to FETCHIT, then save it back to disk. If you have trouble retrieving a file with FETCHIT, double check the data statement. Missing commas and quotes as well as misspellings are the most common types of errors made. Many errors can be found by simply looking at the command when it is displayed on the screen.

Questions or comments? I can be reached on Q-Link under the name "MIKE ALL" or on GENie under "M.GILSDORF1". Until then...Easy DOS it!

Type in FETCHIT using either the TC128 or TC64 Checksum program which can be found elsewhere in this or a previous issue.

```
ba 100 rem fetchit c128/c64 v1.0
ei 105 rem by michael gilsdorf
cg 110 rem copyright (c) dec 93 by parsec inc
ha 115 rem jiffydos syntax: @[1//:]/=name[,#]
lc 120 :
jd 125 x$=",": q$=chr$(34): r$=chr$(13): rr$=r$+r$
pk 130 dv=peek(186): rem current device number
mn 135 x=peek(183): rem length of command string
mb 140 z=peek(187)+256*peek(188)-1: rem filename address
ml 145 :
lc 150 rem find equal sign
ci 155 for q=2 to x: if peek(z+q)<>61 then next q
nk 160 :
jh 165 rem find name
bl 170 if q>=x then 180
fm 175 for n=q+1 to x: z$=chr$(peek(z+n)): if z$<>x$ then n$=n$+z$:
next n
fo 180 if n$="" then print "?missing name error": end
```

```

pd 185 :
gf 190 rem lookup name
mk 195 read a$,b$,c$,d$,e$,f$: if a$="" then a$=n$: d$=n$
ne 200 if a$<n$ goto 195
bo 205 if z$<x$ goto 245
am 210 :
pd 215 rem find load number
cd 220 if n=x then print "?missing load number error": end
ci 225 e$="": f$=""
kp 230 for l=n+1 to x: e$=e$+chr$(peek(z+1)): next l
cf 235 :
co 240 rem display commands
jd 245 print chr$(147);chr$(18);"fetchit c128/c64 v1.0";r$
de 250 :
mn 255 rem change partition/directory commands
nc 260 if b$<>" or c$<>" then print "open 15,";dv;",15";rr$: r=r+1
ad 265 if b$<>" then print "print# 15,";q$;b$;q$;rr$: r=r+1
ce 270 if c$<>" then print "print# 15,";q$;c$;q$;rr$: r=r+1
ei 275 if b$<>" or c$<>" then print "close 15";rr$: r=r+1
fd 280 :
of 285 rem load file and execute commands
ea 290 if d$<>" then print "load ";q$;d$;q$;",";dv;
jo 295 if e$<>" then print ",";e$;
jh 300 if d$<>" then print rr$;rr$: r=r+1
no 305 if f$<>" then print f$: r=r+1
hb 310 :
ep 315 print chr$(19);
hl 320 :
kf 325 rem setup dynamic keyboard
oc 330 kb=631: kc=198: rem c64 keyboard buffer and key count
bf 335 if peek(65533)=255 then kb=842: kc=208: rem c128 addresses
oa 340 for j=0 to r: poke kb+j,13: next: poke kc,r: new
je 345 :
kc 350 rem save "@0:/",dv
jo 355 :
ei 360 rem enter data for each file you want fetched
ki 365 :
cf 370 rem pseudo part sub-directory path load
cb 375 rem name cmd command filename no.
command
lh 380 :
mb 385 data "*" , , , "0:" ,
,run
on 390 data "$" , , , "$" ,
,list
nh 395 data "?" , , , "/" ,
,list 350-
ie 400 data "/" , , , "/" , ,

```


Commodore based the 1571's disk drive on the 154x's memory map. What David did was remove a lot of the code he installed in the 1551 disk drive and splice it into the 1571. This code includes the new Group Code Recording (GCR), the faster formatting code, etc.

I only wish Commodore had released the 1571 with a serial bus and an IEEE or TED bus which would have increased the overall speed of the drive, if the DOS had been recoded correctly.

To wrap up all of the above, the only drives that are both read AND write compatible are the 4040 with 2031 disk drives, and the 15xx series of drives with each other. Of course, you can not put a 1571 disk in a 15xx series drive and read side 1 since you do not have the read/write head to do it. You can read the same disks with the 15xx, 2030, and 4040, but do not write back to the disk or you may regret it since some pointers for the second side reside on the first side of the 1571 disk.

One of the changes made in the 1571 DOS was the method by which it handles the IRQ's (Interrupt ReQuests). Instead of always jumping to a constant memory location (address), it jumps through a vector the same way that your computer's Kernal jumps through the ICHROUT vector to output a character to the screen, printer, etc. Since this is the first disk drive to have this ability, such programs as this were not possible before without a major effort on the part of the programmer. With this new vectoring system in your 1571 disk drive, we have many options to pursue, such as the one covered in this article.

The 1571's Drive Verification Routine

Whenever one of the current disk drives, from the 2030 through the current 1581, writes a sector to disk, it will then proceed to read that sector back in from the disk to verify it against what it has in the drive's RAM to ensure that the bytes were written correctly.

Generally speaking, you will never see an error occur unless you have a bad diskette, or a hardware failure in the disk drive or serial bus itself. Normally a bad disk is detected during the format process, but they do fail from time to time just by sitting in your disk box.

In either case, you will not see the problem very often. What this program does when installed in the 1571, is to 'wedge' itself into the IRQ vector to allow the program to check to see if the DOS is about to do a VERIFY operation; if it is, I replace it with a SEEK operation. If you do not understand how any of this works, do not worry about it as it is not required to know how the program works to use it.

This program could be considered risky to some degree and it probably is risky, but I have been using this program daily for three years with no problems encountered to date. Do I guarantee this program? Not on your disk I don't. Why, you might ask? Well, to put it simply, there are too many variables involved; from you disk drive having the flu, to disks that are being used as floor mats. It would only take one smudge to wipe a disk. In any account, simply run this program and give it a try by doing some testing on a spare disk. You know the one, it is the one you do not care if your dog or cat eats it, or if Junior uses it as a frisbee!

What I am saying is that I have used this program for quite a while with no problems on several different versions of the 1571 Disk Operating System. But the choice to use this program is up to you to make, not me!

This short program allows you to turn off the write verify operation in your 1571 drive even when it is in the 1541 mode! Currently there are several similar versions of this program running around on the public domain bulletin boards, but several of them contain bugs because they, like this one, sit some where in \$0100, which is the disk drive's stack area. Most of the ones I have seen to date use up too much room on the stack and upon the first DOS error, crash the disk drive! The following program has been tested, tested, and tested some more, with out any crashes!

To use this program all you need is a 1571 disk drive and this program. This program runs on any Commodore computer, from the B-128 to the C-128. To start the program, just type it in or load it. Save a copy of the program before running it if you typed it in. The program will make only one request, that is: "Enter the device number of your 1571 disk drive:". This value can range from 5 through 30 and would be the same number that you would enter for a Basic command such as

```
dload"filename",u(device #)
or
load "filename",(device #).
```

I make several checks to ensure this is really a 1571 disk drive the program will be running on. First I check the IRQ vector at \$FFFE to ensure it contains an \$FE67. This will ensure that it is a 154x or 157x drive. Next I check \$8002 for the letter S to ensure it is a 1571 disk drive. Finally I check \$02A9 which is the IRQ vector for the 1571 disk drive. When I check this vector, I ensure that the MSB of the address contains a value greater than \$80 to ensure that the IRQ is going somewhere in ROM and not to another program such as mine. If all of the above are true, I download the code into the drive and execute it.

Speed example using a 1571:

```
Bsave"test1",d0,u8,b1,p2000 to p50000
(189 disk blocks)
```

```
W/verify : w/o verify
```

```
-----
90sec : 69 sec
```

```
Program name: write.v.bas
```

```
hi 10 rem from tc-128/64 issue #35
eb 20 rem copyright 1994 by parsec inc
fi 30 :
pi 40 printchr$(147)
np 50 printchr$(142)"turn off write verify operation for"
cd 60 print"commodore's 1571 disk drive
po 70 print"(c) 1994 by parsec inc"
ab 80 print"by"
kn 90 printchr$(15)"dennis j. jarvis"chr$(143)
jo 100 :
lg 110 printchr$(7)chr$(15)"press any key to continue"chr$(142)
jd 120 geta$:ifa$<>"then120:rem purge buffer of any previous key
strikes
in 130 geta$:ifa$= ""then130:rem wait for a key to be pressed now
fb 140 print" "
ek 150 input" what is the 1571's device number";a$:dv=val(a$)
ie 160 ifdv<5ordv>30then150
gp 170 print" "
pf 180 open15,dv,15
gh 190 close15:ifsthenprint"device number:"str$(dv)" is not turned
on":end
aj 200 open15,dv,15
jg 210 lsb=254:msb=255:gosub380:rem read the rom irq vector ($fffc)
mf 220 iflsb=103andmsb=254then250
mg 230 print"sorry this program will only work on the 1571 disk
drive"
pj 240 close15:end
```

```
hk 250 lsb=002:msb=128:gosub380:rem check text at $8002 's/w -
      david g. sir...etc
co 260 iflsb<>83ormsb<>47then230
gl 270 lsb=169:msb=002:gosub380:rem make sure irq is pointing to a
      routine in rom
db 280 ifmsb<128thenprint"sorry some one is using the irq
      vector":goto240
lj 290 fori=2to50:read h$:a$=left$(h$,1):b$=right$(h$,1)
lk 300 gosub330:b=a*16:a$=b$:gosub330:c=a+b
nl 310 print#15,"m-w"chr$(i)chr$(1)chr$(1)chr$(c):next
ij 320 print#15,"m-e"chr$(23)chr$(1):print"write verify now turned
      off":end
dl 330 ifa$<"":anda$>"/"thens=48:goto360
ii 340 ifa$>"@"anda$<"g"thens=55:goto360
nn 350 print "invalid hex byte -"h$:stop
hp 360 a=asc(a$)-s
ab 370 return
ol 380 print#15,"m-r"chr$(lsb)chr$(msb)chr$(2):get#15,a$,b$:
      lsb=asc(a$+chr$(0)):msb=asc(b$+chr$(0)):return
ic 390 data 48: :rem pha save acc.
aa 400 data 8a: :rem txa
db 410 data 48: :rem pha save x reg. onto the
      stack
of 420 data a2,05: :rem ldx #5 number of job
      queue's to check
oo 430 data b5,00: :rem lda $00,y get current job out
      of queue
id 440 data c9,a0: :rem cmp #$a0 is it a verify
      command?
eb 450 data d0,04: :rem bne *+15 no then branch
nm 460 data 09,10: :rem ora #$10 replace the verify
      function
cb 470 data 95,00: :rem sta $00,y with a seek command
ca 480 data ca: :rem dex move down to next
      queue
bp 490 data 10,f3: :rem bpl *-11 if any more to do
      branch
fg 500 data 4c,fe,ff: :rem jmp $0000 see text for this
      one
dj 510 :
ff 520 :rem this is where the program is executed when it's in the
      disk drive
ia 530 : [ editor's note - 31 up arrows here ]
hc 540 data 78: :rem sei stop background jobs
je 550 data ad,a9,02: :rem lda $02a9 get the address of
      the
fi 560 data 18: :rem clc current address in
      the irq
lo 570 data 69,03: :rem adc #3 vector and add 3 to
      is then
im 580 data 8d,15,01: :rem sta $0115 save it for us to
      jump to,
```


Since we are using 10 binary digits it should be obvious that since a byte can hold eight "digits" or binary bits that we are going to have to use more than one to handle numbers greater than 255. The number 1024 above can be represented in memory as:

```
1st Byte: (binary)           1 0
2nd Byte: (binary) 0 0 0 0 0 0 0 0
```

or, since the 6502 within the Commodore 64 and 128 likes the digits reversed so that the highest placed digits are last in order, they are usually organized as:

```
1st Byte: (binary) 0 0 0 0 0 0 0 0
2nd Byte: (binary)           0 1
```

We can use all 16 bits in the two bytes that we are considering at the moment, so that opens up as many as (2 to the 16th power) numbers to use, or the values 0 to 65535. Numbers greater than this can be represented by using another byte or bytes. Just keep adding bits on. We will take a look at how to perform certain mathematical operations on these values below.

We have seen how we can represent numbers greater than 255 by using more than one byte to represent the number. Let us see how we can represent negative numbers.

We have to have some way of determining whether the value is negative or not, some sort of "+" or "-" to be stored in one of the bits that hold the number. This is usually represented in the highest value binary bit, in a single byte value this would be the 7th bit:

```
  7   6   5   4   3   2   1   0
sign #   #   #   #   #   #   #   #
```

In a multi-byte number it is also in the highest bit position:

```
15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
sign #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #
```

Obviously we can not just put a "+" or "-" in a bit. A bit can only hold the values 0 and 1 so we will use a value of 0 in the highest valued binary bit to represent positive and a "-" to represent negative.

Using this notation the number -1 could be represented as:

```
sign6 5 4 3 2 1 0
  1 0 0 0 0 0 0 1 = negative (1*1) = -1
```

Yet, if you examine what we have done so far you can see that we will run into problems with 0. We currently have a negative and a positive zero. If we do any comparisons then we will have problems. Also, during addition and subtraction we will have to check the signs of each of the values and adjust accordingly.

If we add one to the number then we can avoid the problems that currently exist with the zero. If we change all the 1's to 0's and all the 0's to 1's when representing the number as a negative it will allow us to add and subtract with relative ease:

```
7 6 5 4 3 2 1 0 (negate) (add 1)
0 0 0 0 0 0 0 1 -> 1 1 1 1 1 1 1 0 -> 1 1 1 1 1 1 1 1 (-1)
0 0 0 0 0 0 0 1 (let's keep as positive so no change)
+ 0 0 0 0 0 0 1 0 (+2)
-----
0 0 0 0 0 0 0 1 (+1)
```

Note that we threw away the left most bit that got carried all the way when adding the 1's to get the correct result. This is always necessary. Other examples are below. Try coming up with some numbers on your own and trying similar operations and make sure you are familiar with the process before continuing.

```
0 0 0 0 1 1 1 1 { 15}          0 1 0 1 0 1 0 1 { 85}
1 1 1 1 0 0 0 1 {-15}         1 1 1 0 0 1 1 1 {-25}
-----
0 0 0 0 0 0 0 0 { 0}          0 0 1 1 1 1 0 0 { 60}
```

Addition and Subtraction

The Commodore 64 and 128's 6502 processor can only work with eight bits at any given time. To add the numbers 2 and 3 you might have the following code:

```
clc ; The 6502 CPU requires the carry always cleared
lda #2 ; Put a value of 2 in the .A register
adc #3 ; Then add the value of 3 and the carry (which
;is 0)
;(Add with Carry)
```

Note that you have to clear the carry. Failing to do so may result in accidentally having a value one more than what you wanted depending on previous calculations. Those previous calculations and using the result of the carry flag can be helpful in dealing with numbers that are contained in more than just one byte. Note that the number 56 plus 7 is equal to $6+7 = 1$ and a carry, then considering that $5 + 0 +$ the previous carry = 6, so that $56+7 = 61$. An example: we evaluate the one's digits, then the ten's digits taking into account the carry. We do the same with binary numbers, starting with the lowest valued byte and moving to the highest valued. Let us add the numbers \$1234 and \$34FF:

```
clc
lda #$34      ; low byte value of 1st number.
adc #$FF      ; add low byte value of 2nd number.
sta result+1  ; Remember that numbers stored High byte, Low byte.
lda #$12      ; Note that we kept the value of the carry...
adc #$34      ; Add high byte + the carry.
sta result    ; and result will hold the answer.
```

If more than two bytes were used then it would continue to the next higher byte value:

```
lda #next value
adc #next 2nd value
sta result+offset
lda #even higher...
adc #even higher 2nd...
```

Subtraction is done similarly but with one difference. You must always set the carry instead of clearing it when dealing with single byte values. To subtract \$34 from \$ff we use:

```
sec          ; Set the carry.
lda #$ff    ; Load initial value
sbc #$34    ; and subtract.
```

Multi-byte subtraction is done almost identically to multi-byte addition with the exception of using SBC instead of ADC and setting the carry instead of clearing it.

Shifting and Rotating

Note that if we have the binary value % 0000 0001 that has a decimal value of 1 (broken up into 4's to make it easier to "read") and we shift all the digits one place to the left then it would be: % 0000 0010 which has a decimal value of 2. If we take 3 (% 0000 0011) and shift it one place to the left then we end up with a result of 6 (% 0000 0110). Shifting one place to the left results in a multiplication of 2.

This can also be seen when dealing with decimal numbers. If you shift a decimal number (which is a base 10 number as opposed to binary which is base 2) then you will end up multiplying the number by 10. 63 shifted left one time is $63 * 10$. As a rule, shifting any number to the left that is in a given base will multiply the number by that base. Shifting a hexadecimal number to the left will give a result of multiplying the hexadecimal number by 16. (\$01 shift left = \$10 = $1 * 16$).

Does the opposite hold true? Can we take a binary number and shift it to the left to obtain a result that is the number divided by two? With decimal numbers we can do that and obtain the number divided by 10. With binary numbers the same holds true. We can take a number such as %0000 0110 (6) and shift it to the right to give a result of 3 - %0000 0011.

Multiplication and Division

With multiplication and division things become much more complex, the 6502 CPU does not support any form of multiplication and division other than the simple rotations that we saw above.

When you go to multiply $3 * 8$ most people use the multiplication tables that were learned by rote back in grade school. Numbers such as $123 * 45$ are done by long multiplication:

$$\begin{array}{r} 123 \\ * 45 \\ \hline 615 \\ + 492 \\ \hline = 5535 \end{array}$$

Binary numbers can be done the same way:

Let us multiply 3 (%0000 0011) by 5 (% 0000 0101) in binary:

```
% 0011 (3) Note that I am just representing the
* % 0101 (5) lower 4 bits in each binary number.
-----
  0011
 0000
+ 0011
-----
% 1111 (15)
```

Okay, we know how we can do it, how can we make the computer do it?

Let us take a look at the problem we had above

$\%0011 * \%0101 = \%1111.$

If we really look closely at what we did above we can see that it was

$\%0011 + \%1100.$

The first number got rotated and added only when the bits in the other value were one. So that we could have written the above as

$\%0011 * \%0101 = \%0011 (\%0001 * \%0011) + \%1100 (\%0100 * \%00001) = \%1111.$

The basic algorithm is as follows:

```
set sum to 0
+--> rotate number 2 to the right.
      if we rotated a 1 bit off, then add number1
          to sum.
      if number2 = 0 then sum = result (end)
      rotate number1 to the left
```

+-----+

Division is the opposite of this process. Binary division done "long-ways" is done the same way that it is done using decimal numbers - dividing 3 (%0011) into 9 (%1001) is :

```
          1 1      -> %0011 = 3. -> Quotient
-----
% 11     1 0 0 1  <- Dividend
Divisor  - 1 1
-----
          1 1
         - 1 1
         ---
          0
```

The basic algorithm is as follows:

```
Set Quotient = 0
Find initial length of Dividend
Shift divisor left LENGTH times
+--> Trial subtract divisor from dividend
Successful? (ie: > 0)
      Dividend = Dividend - Divisor
      Rotate 1 into Quotient (left shift)
NonSuccessful?
      Rotate 0 into Quotient (left shift)
Rotate Divisor right
= 0?
+--- no
```

Try working some examples out on paper, these routines really do work. And they are the exact same steps you go through (they may not look like it because they are in binary and we are not doing it in chart form) do it the long way by hand.

Square Root

Square roots present a problem on computers. While we are not focusing on floating points in this article (we will cover how they are represented but save the discussion of actual code for another column) it would be nice if we could find the square root of say a number like 25 and what fraction of 25 is left.

Some while back I was looking through a math book and it gave a unique and interesting way of arriving at prime numbers. I have attempted to find it again, but I am still looking for the book over a year now. This algorithm does work and I expect it has something to do with convergent series but have never had the time nor the inclination to actually prove it.

Basically, it states:

The square root of a number, n, may be found by counting the number of successive odd numbers that can be subtracted from n.

So if we have 25 then it follows:

$$25, \underset{1}{25-1} = 24, \underset{2}{24-3} = 21, \underset{3}{21-5} = 16, \underset{4}{16-7} = 9, \underset{5}{9-9} = 0 \text{ odd #'s}$$

If we try numbers like 17 we will find that we get 4 and a remainder of 1 to represent $4^2 + 1 = 17$.

$$17, \underset{1}{17-1} = 16, \underset{2}{16-3} = 13, \underset{3}{13-5} = 8, \underset{4}{8-7} = 1, \underset{5}{1-9} \text{ xxxx and the remainder of } 1.$$

Because the inverse of the square root is an exponential function we are blessed with being able to know that finding the square root of a number such as 65536 can be found in 256 subtractions. That may seem like a lot of subtractions but 256 subtractions can be done very quickly in machine language.

There are other ways of going about calculating the square root, which imitate the "real-life" solution. Namely, by trying to divide a 2, then trying to divide a 3, a 5, etc... and then recomposing these numbers. But for our purposes here these should serve. Also, this may be slightly faster for some numbers as the division takes a series of steps "n" or less long for each factor where ours will take at most the square root of the number.

Floating Point Numbers - Overview

I am only presenting an overview here as full coverage on floating point numbers should be given an entire column to do sufficient justice on them. Here, I will simply try to give you an inkling on how they are represented.

The decimal number 10.34 is the same as 10.34×10^0 (power), or 1.034×10^1 (1st power), or $.1034 \times 10^2$ (2nd power). If we want to add, say 34 to this number how would we go about it? Normally you move the 34 over and under the 10 and then add the 10 and 34 together and treat the 34 as the equivalent of 34.00. However, let us make 34 have a multiplier of 10 (to the 2nd power) on it. Which is the same thing as moving it two places: 34×10^0 , 3.4×10^1 (1st power), $.34 \times 10^2$ (2nd power).

If we add $.1034 \times 10^{(2\text{nd power})}$ and $.34 \times 10^{(2\text{nd power})}$, because they have the same exponent, then we end up with $.4434 \times 10^{(2\text{nd power})}$ which converted back is 44.34 which is the same as $10.34 + 34.00$.

A convoluted way of going about it right? The process where we shifted each number is called "Normalization". We are doing the equivalent of being able to write them one under the other. The placement of 10.34 and 34 in the example above in normal terms looks like:

$$\begin{array}{r} 10.34 \\ + 34 \\ \hline 44.34 \end{array}$$

You knew where to put the decimal point because of the decimal. In floating point representation in computers, all numbers are assumed to have an implicit decimal point to their immediate left.

The floating point numbers are held as two different values: The "MANTISSA" which is the number with an implicit decimal point at the far left. So the mantissa of 10.34 would be 1034. The mantissa of 3.14159 would be 314159. The other part consists of the EXPONENT which contains the power that the number needs to be raised to have the actual representation of that number.

So, to add numbers that consist of floating point, you first have to normalize them. Basically shifting the number and mantissa so that they have the same exponent as the number you wish to add it to.

Then a normal addition is performed and the number is given the exponent that each number that was added has. Note that we always shift the exponent up, never down. If you shift it up, then you rotate the mantissa to the right (to keep the number the same: Example: $3.4 \times 10^{(1\text{st power})} = .34 \times 10^{(0\text{ power})}$). Since you can only hold so many digits it is much better to represent $3.4 \times 10^{(1\text{st power})}$ as $.34 \times 10^{(2\text{nd power})}$ and lose a fractional digit than to lose the leading 3. Obviously you will hold more than just one digit after the fraction but this is intended as an example).

In the next article, we will take a look at these routines and see how by using basic for the plotting, and machine language for the calculations, fractals can be generated.

Craig can be contacted as C.TAYLOR37 on GENie

CMD Utilities

A review by John W. Brown

The "CMD Utilities" package is a disk of utilities written to make using CMD hardware devices easier. The disk will be of interest to people that already own the CMD hardware and want to upgrade the earlier versions of their software tool kits.

The package itself consists of a double sided 5.25 diskette. One side contains the utilities and the other side contains the documentation for the programs. The documentation is in the form of Pet ASCII text files formatted for eighty columns screens. If you want printed documentation, it is \$7 (ouch) extra. Most of the programs on the disk are easy to use and to run. Most of the programs will run in either C-64 or C-128 mode.

This disk will appeal to people that:

- 1) Already own CMD hardware devices.
or
- 2) Have more than one disk drive.
or
- 3) Have no access to User Groups.
or
- 4) Do not own modems.

If you have a single disk drive and have access to public domain programs, you do not need this package. If you own CMD hardware devices and want to update your utilities that came bundled with the purchase, then you might consider this disk. Since the disk contains many small utilities I will list each program on the disk and follow it by a comment.

MENUETTE 64

This is the star program on the disk. Its main appeal is that it cushions the user from the DOS system and removes the need to type partition and disk commands. Though, you still have to have at least a scant knowledge of how your hard drive works and how it is set up, to properly use the program.

The default user input is set for the keyboard. You can also set it up to use the joystick or a C-1351 mouse. Input and screen movement using any of the input

options is smooth and fast. The menu options are laid out in a logical manner, are easy to figure out, and are easy to use, without reading the on disk documentation. Though for the real novice, it would have been nice to have in the sparse one sheet flyer, a mention about which keys work what (IE: the space bar key selects items) to get people going. All and all a very nicely done menu program and no complains about it.

FCOPY+

This is an update of the earlier FCOPY program and needs at least two disk drives to work properly. A well written program that gets co-billing on the disk with Menuette. FCOPY is all I use for file copying, but, if I did not own any CMD hardware, I probably would still be using Fast Hack'em 128 or Maverick 64 because they are faster on non-CMD devices and support single device file copying.

MCOPY

This program allows you to copy whole disks between similar devices or between one source device and a target emulation partition on your CMD HD, Ramlink, or FD series. You might consider this the whole disk nibbler.

MCOMPARE

Its main use is for verifying MCOPYed partitions against the source disk you just copied.

BCOPY

One of the outstanding programs on this disk. The people that will find this program attractive are the ones with CMD hardware with large native mode partitions that want to back them up on a regular basis without file copying or compression. Backing up large native mode partitions to 1581 or FD-4000 disks is a snap and takes a relatively short amount of time.

FIND

Used to locate files where ever they are on your mass storage device. It offers both screen and printer (pet ascii) output. The idea sounds nice, but in reality, the program is too slow and I never use it.

SUBPARTITION AID

You can use this program to make C-1581 type sub-partitions on C-1581 disks or in C-1581 emulation partitions on your mass storage device. No real need for this program if you own a piece of CMD hardware since CMD directories and partitions are superior to the C-1581 sub-partitions.

CMD DIR SORT 1.5

With DEDIT now available, on the same disk, this program is fairly useless to most people since it is outdated. Though the program might appeal to a programmer who wants to know "how it is done" when dealing with CMD devices and directory structures.

CONVERT 41<>71

This allows you to make any 1541 disk that was MCOPYed to a CMD 1571 emulation partition on a CMD HD or CMD Ramlink, into a 1571 type disk so you can take advantage of the increased disk space on the larger storage device. I have never used it because I have never had a need for it. The biggest advantage is probably using it on mildly copy protected applications or on older programs that look for data located in certain tracks and sectors on a C-1541 disk.

FOLLOW LINKS

This program allows you to follow the track and sector links of a file and if you find a bad track/sector link in that file, to remove that file from the disk directory. This is handy when you have a bad file, that somehow got corrupted, that will not let the VALIDATE command clean the directory properly. This usually happens when a file get corrupted while it is being written to. When this happens the track and sector pointers are not correct that link the file together and contain illegal or recursive track and sector pointers. This file allows you to remove that bad file so you can salvage the remaining programs in the directory. All and all a pretty useful program that you would want in your toolbox "just in case".

ZAPREU

This program is used to browse through the memory of your REU (17xx) or to fill the memory with a numerical value.

ZAPDACC

Used to destroy everything inside a Direct ACCess partition, which can be handy at times using a Ramlink.

FOREIGN CREATOR

Its main use is to create large partitions to be used with other computers connected to your CMD hard drive. Never had a need and never used it.

HD POWER TOOLS

This program allows you to edit your CMD partition table and data. To use this utility you have to know what you are doing when it comes to how CMD hard drives are set up and how they run. This program is good for the person that likes to "hack" around. Otherwise you should never use or need this program.

REBUILD PDIR

If your hard drive operating system crashes, from using something like HD Power Tools, this utility helps you rebuild your partitions. I have never used it since it would mean taking a chance on destroying all the data on my hard drive. The utility to use as a last resort when you or the software you are using, does something real dumb.

DOCPRINT

Used to print out the on disk documentation.

CMD TIME

This is an auto-executing file for GEOS that you place on your GEOS boot disk. While booting GEOS, it searches for a CMD hardware device (HD.RL.FD.) with a real-time clock module installed and uses that time and date information to set the GEOS time and date clock. This only works with GEOS V2.0. This is a useful utility if you do not have gateWay.

CMD MOVE

This is for use under GEOS using CMD hardware devices. It allows you to change partitions on CMD devices and to copy files from one partition to another. I have never used this utility because I do not like it and always find another way to do the file copying. Enough said about that.

DEDIT.64 V7.1 and DEDIT.128 V7.1

On side two are the "dedit" (Disk EDIT) programs by Jerome Yoner. One for the C-64 and one for the C-128. They are directory sort programs that are good shareware programs. The cost to register DEDIT is \$5. DEDIT sorts and works on all kinds of directories and devices, including the whole CMD line. Good programs and well worth the low shareware registration price.

To Sum It Up

There is nothing really new or spectacular about most of these programs, save Menuette. They are just good basic utilities that you might need if you are using CMD hardware. Most are upgrades to older versions of the programs found on the utility disk that comes with every CMD hardware device.

The unique "value" program is Menuette 64 though for a seasoned user it does not provide much of a benefit. Flitting around your hard drive is much easier and quicker doing it from DOS or from within programs that provide disk command access.

For the person or novice that does not know how to send disk commands or hates doing it, Menuette would probably be worth the price of the package if you need that kind of insulation from your hardware.

Bang for the buck

When I got the utility disk with a single sheet flyer and saw what was on the disk, I thought "\$9.95 and packed with utilities, good deal". I am of the mindset that software that costs more than \$9.95 and certainly anything over \$19.95 practically demands some sort of manual. I am a little taken back by "commercial" software that costs \$24.95, comes with no printed documentation save a flyer, comes with shareware on the back of the disk as part of the "value added" to the package, has print it yourself "docs", and then tells you to cough up another \$7 for the printed manual. Since the "docs" are formatted to 80 columns, viewing them from Menuette does not provide the best of reading experiences.

All I could think of was a person that saved up enough money to buy a CMD hard drive instead of a printer, buying the disk, forking out \$24.95, plus the \$5 shareware fee, plus another \$7 for the manual.

If this disk was \$9.95 with no printed documentation, then it would have gotten a "A" for the value, but for \$24.95, plus another \$5 to register the shareware program, plus the absence of any printed documentation ...

RATING: B

Price \$24.95

Creative Micro Designs

PO Box 646

E Longmeadow MA 01028

1-800-638-3263 (order), 1-413-525-0023 (support), 1-413-525-0148 (bbs)

A C-65 what?

By John W. Brown

People have been asking what the C-65 computer is. As I understand it, this is the same model that was suppose to be shown at a computer fair in England, during the Fall of 1990. The C-65s being offered for sale by The Grapevine Group Inc. are basically pre-production models that were originally meant for the developers working on the machines. We do not want to mention names and get into a JAM, but anyone that was somebody in those heady C-128 days worked on this model.

There are no service manuals, replacement chips, owner's manuals, etc. If it breaks down outside of the 90 day warranty period you have a nice door stop.

The C-65 is set up for the PAL display they use in Europe. On a regular NSTC monitor like those used in the USA and Canada, you will only be able to display B&W on the forty column screen. Because of the very limited supply, I have decided just to list some of the features and not to do a complete review since you probably can not buy them now. Even if you could, you would have no way to use it unless you were online and had access to messages being posted about it as people "hack" away at it.

Features:

Regular C-64 mode and a C-65 mode.
A CPU CSG65CE02, running at 3.54MHz
RAM 128K expandable to 8 Megabytes
Regular C-64 video modes

C-65 video modes that includes 80x25 text, true bitplane graphics, screen resolution up to 1280x400 pixels along with a 256-color palette with 16 intensity levels per primary color.

Stereo SID chips

A built-in C-1581 disk drive.

Built-in DMA/blitter chip

Built-in 6511-type UART

All the normal C-64 ports parallel, expansion, serial

A new expansion port, accessible by DMA and the video chips.

Dr. Octal's

Sharp Operating Tips <tm>

Tip &0017

USING TIME

From: Noel Nyman

You can not reset the reserved variable TI directly. But you can make TI=0 indirectly. Just use TI\$="000000"

You must use all six zeros. By using non-zero values for TI\$, you can set TI to any value. TI will be sixty times the number you use for TI\$. Just remember that TI\$ cannot exceed "235959". If you use a larger value, the computer will reset it to "000000".

Tip &0020

RELATIVE FILES

From: Noel Nyman

When creating a database that uses "relative" files, do not use records lengths 42, 58, or 63. These correspond to the ASCII codes for "*", ".", and "?". Those are special characters to the drive. If you use them as relative record lengths, you will get a "syntax error".

Although you may not ever create relative files in programs of your own, many Commodore databases use relative files. Be sure the total length of all fields in any database you create is not one of the three illegal lengths. Some programs, such as QUICKPRO+II, do not check for the illegal lengths and may let you enter all your data before you realize that no file has been created to hold it.

Tip &0021

SAVING TIMES

From: Noel Nyman

To save multiple versions of a file, such as when writing a large BASIC program, try

```
SAVE "0:FILENAME"+TI$,8
```

Each file name will have TI\$, six numbers representing the elapsed time since the computer was turned on appended to it. Be sure the filename is ten characters or less.

Tip &0022

FILE NOT FOUND

From: Noel Nyman

If you get a "file not found" error when a boot program tries to LOAD another program, and you can not list the boot program to find what file is missing, just type SYS 62913 (C64 only) and the file name looked for will be printed on the screen.

Tip &0023

BEING RESTORED

From: Noel Nyman

If you press <RESTORE> gently, it may not register with the computer. A tap (not necessarily a hard one) is required.

Tip &0024

KEYBOARD-JOYSTICKS

From: Noel Nyman

If you want to play a joystick game, but do not have a joystick handy, you can try these keyboard equivalents (hope you have good dexterity).

| | JOYSTICK1 | JOYSTICK2 |
|-------|--------------|-----------|
| fire | space | space+m |
| up | 1 | space+f1 |
| down | <left arrow> | space+z |
| left | control | space+c |
| right | z | space+b |

Tip &0025

DATA CASSETTE

From: Noel Nyman

If you use a Datassette, cut the metal braid off the plug and throw it away! This was required by the FCC only for Commodore Business computers, and can cause serious damage to a VIC/C64/C128 if it comes into contact with the user port.

Tip &0026

POWER DOWN

From: Noel Nyman

Do not plug anything into your computer with the power turned on! Do not unplug anything from your computer with the power turned on! This is a common cause of hardware problems in Commodore computers, especially the joystick ports.

Tip &0027

JOYSTICK PORT STATIC

From: Noel Nyman

Do not let your fingers come close to the exposed joystick port pins when using the power switch or the reset switch on a C128. The computer chips can be easily damaged by static charges jumping from your fingers to the exposed pins. To avoid the problem either leave two joysticks (or other devices) plugged into the ports, or cover the ports with a small cardboard shield when not using them.

Tip &0030

DISK DRIVES

From: Noel Nyman

Keep the disk drive(s) away from the left side of your tv set or monitor. There are strong magnetic fields on that side, which can affect the drive operation. (note from Editor - my small amber monitors wrecks havoc with my 1581s no matter which side it is on. If you are having read and write errors, try moving the disk drive)

Tip &0031

JOYSTICK

From: Noel Nyman

If you suddenly get strange characters when typing on your keyboard, check to see if you have a joystick plugged in. The computer uses the same chips to read both joysticks and the keyboard. If you push on a joystick or fire button while typing, you can get weird results. Sometimes the joystick may be stuck or you might have left the auto-fire button on.

Tip &0032

Ramlink Tips

From: John W. Brown

Put electrical tape over the battery terminals to secure them, especially if you are moving the Ramlink and do not want to lose the contents. Most importantly, this prevents loose metal objects such as paperclips or pens from accidentally shorting the terminals out.

Tip &0033

Ramlink Tips

From: John W. Brown

Whenever you power down your system to unplug Ramlink or to work on your computer, make sure before you turn the power off, that the power connector for the battery is plugged snugly into the Ramlink (usually it "clicks" in). Many times the other cables, such as the serial cable, can unknowingly pull the battery cable loose from the Ramlink.

Tip &0034

CMD Hard drive

From: John W. Brown

Though my reconditioned CMD hard drive works well, it makes a slight humming or whine, which would not be too bad except it sits on a metal shelf that amplifies it many times over. My solution was to take an old mouse pad, that had a thick foam backing and place it underneath the hard drive. Just make sure your mouse pad is smooth and will not block the air intake vents at the bottom of the drive.

Tip &0035

Killing C-64 mode

From: John W. Brown

I am sure if you have a C-128 and have run any type of C-64 demo or game, you have been frustrated at one time or another. Why? Because when you reset your C-128 with the reset button and go back to C-64 mode, the C-64 program is still there either running or locking up your C-64 mode. This is especially true with the Gary Kitchen Maker games. My solution, besides trying to reach for the on/off button? Reset your C-128, enter the machine language monitor and then type this line: F 01000 0d600 ff

This is usually enough to wipe all pointers and programs from the C-64 memory area and will allow you to run in C-64 mode without having to shut your computer off.

Tip &0036

POST - ITs

From: John W. Brown

If you read this magazine, chances are you have opened your computer more than once. If you have an older brown case C-64, this means the posts where the screws go in to hold the case halves together have cracked or broken. My solution was to glue the posts back together as best as possible and then put some shrink wrap tubing around the posts. By using multiple layers of shrink wrap tubing, shrunk one at a time around the posts, you can create a very strong grip on the broken posts. This can only be broken by having a gorilla with a screw driver tighten the case. You can pick up shrink wrap tubing suitable for the job at Radio Shack.

geoMakeBoot

by Staff

geoMakeBoot is a program that enables the C-64 or C-128 owner to make a GEOS boot disk that will run from practically any type of disk drive, as long as they are device numbers 8 or 9. This includes 1541, 1571, 1581, CMD Hard Drive and CMD RamLink owners. If you have not bought gateWay yet, are tired of being able to boot GEOS only from your 1541/71 disk drive, even though you have a C-1581 (or CMD hardware) attached to your system as device #9, then this software is for you.

Installation is easy enough. Format a new disk, copy over the needed GEOS files from your old boot disk to the new boot disk. The files needed are GEOS, desktop, configure, the input and printer drivers, and the geoMakeBoot program. Run the geoMakeBoot program, wait for it to create geoBoot on your new boot disk and you are all done.

Any C-128 owner thinking about purchasing Maverick 64 just to get the ability to create GEOS boot disks on their 1581 disk drive will save a lot of money by buying this utility!

geoMakeBoot does come with one page of badly xeroxed and barely readable documentation, which is enough to get the job done.

RATING: A

Price \$12.95

Creative Micro Designs

PO Box 646

E Longmeadow MA 01028

1-800-638-3263 (order), 1-413-525-0023 (support), 1-413-525-0148 (bbs)

Software Picks

by staff

This column is where we spotlight exceptional files, either public domain, shareware, or "demos", from both the past and present. They range from the simple basic program to the full blown application. These programs are located on the back side of the TC-128/64 companion disk. It is (ONLY) the back side of the companion disk that is okay to copy for a User Group disk library or to pass around to friends. Please remember the front side of the TC-128/64 companion disk is commercial software and may not be duplicated except for archival purposes.

GENie file #: 10862

Filename: Irr.sfx

Program : Little Read Reader 128

CBM BLKS: 154

Computer: C-128

Author : Craig Bruce

Type : public domain

This is the "Little Red Reader 128", a public domain program that lets you read IBM formatted disks on either a 1571/81 and then write the files to a second disk drive of any type. What is exceptional about this file is it includes complete documentation and the source code so it could be easily modified to read the high density disks of IBM clones.

GENie file #: 13789

Filename: spv2demo.sfx

CBM BLKS: 190

Computer: C-128

Author : Brett Tabke

Type : product demo

This is a "demo" file of Spray Paint 128 that was favorably reviewed in a previous TC-128/64 issue and advertised in this issue.

GENie file #: 14135

Filename: ultrav7.0.sda

CBM BLKS: 85

Computer: C-64

Author : Bob Kober

type : public domain

Our favorite sequential text file reader for the Commodore 64. An oldie but goodie!

GENie file #: 14283

Filename: epson8pincolor

CBM BLKS: 9

Computer: GEOS 64/128

Author : George H. Wells, Jr.

Type : public domain

A color printer driver for Epson 9 pin color printers.

GENie file #: 14249

Filename: epson24pincolor

CBM BLKS: 8

Computer: GEOS 64/128

Author : George H. Wells, Jr.

Type : public domain

A subscriber recently asked if there were any color printer drivers for GEOS

that would work with his 24 pin
Panasonic printer. I said there were
not, I stand corrected, here is the
printer driver!

GENie file #: 14153

Filename: bubblejet-lq.cvt

CBM BLKS: 8

Computer: GEOS 64/128

Author : Lloyd Hayes

Type : public domain

People have been asking about Canon
BJ-200 printer drivers so ...

GENie file #: 11509

Filename: zed077.sfx

CBM BLKS: 89

Computer: C-128

Author : Craig Bruce

Type : public domain

A full featured text editor for the
C-128 that supports your REU, from the
C-1750 (512K) all the way up to 8 mega-
bytes for the expanded REUs. It can
handle 100K on an unexpanded C-128!
Finally a way to handle those whopping
text files without splitting them. It
can handle text files in sequential and
program form, can read Speedscript, Pet
Ascii, and regular ASCII text files. It
has many text editing commands and also
doubles as a handy Pet<->ASCII text
converter and word processor.

GENie file #: 13842

Filename: bigclipper.cvt

CBM BLKS: 29

Computer: GEOS 64/128

Author : Nick Vrtis

Type : shareware \$10

Finally, a way to stuff large pieces of geoPaint pictures into your Photo Albums without any hassles. Runs on the C-64 or C-128 in forty columns.

GENie file #: 10470

Filename: world war

CBM BLKS: 80

Computer: C-64

Author : Darren Duffy

Type : public domain

This game allows 1-4 players. If you pick 1 player, you play against the computer. 9 difficulty levels and nine land masses to conquer. This game can be compiled for greater playing speed and also ported over to the C-128 and probably the +/4 by changing just the joystick values. An addicting game.

Font Paint64-128

"Font Paint64-128" was written by Kent Smotherman for Parsec, Inc. and adapted for the C-128's 80 column screen by Robert Knop. "Font Paint64-128" can be found on the commercial (first side) side of the companion disk for TC-128/64 #35. "Font Paint64-128" will "paint" fonts to a geoPaint file.

Coupled with geoPager 64 or geoPager 128 from previous TC-128/64 issues, you can print a "book" containing all your font images. "Font Paint64-128" prints all the important font information above the images of each individual font and correctly handles "mega" or "supra" fonts.

Looking up and choosing fonts has never been easier and this program will be an invaluable tool for the avid GEOS user with 100s of fonts.

More about the program

- o FP64-128 fully supports three drives
- o The "font drive" is the disk drive where the fonts can be found and the "paint drive" is where the geoPaint images will go.
- o If all of the fonts for the current source disk will not fit into a single geoPaint picture, FP64-128 will automatically print the disk name and page number at the top of each successive geoPaint picture for that disk.
- o If you want to change the default disk name and page number, move the cursor all the way to the left hand side with the inst/del key and then type in the disk name. Remember to use 12 characters or less for the disk name.

This application works with GEOS 64 or GEOS 128 in forty or eighty columns.

LEGAL NOTICES -

Twin Cities 128/64, the magazine, or companion disk, may NOT be copied in whole or in part for ANY reason. TC-128/64's companion disk is commercial software and is only for individual use. It may NOT be put into a User Group's "commercial disk collection" where it can be copied freely, loaned, or rented. If it is, the group and officers will be held liable for all attorney fees and damages. Please note though that the Public Domain and/or shareware disk that comes on the back side of the companion disk CAN be freely distributed.

SOFTWARE NOTICE, RIGHT TO USE

The software, hardware, and routines published in this magazine can be used free of charge only if ALL of the following conditions are met:

- 1)The program is copyrighted but freely distributable and you were a subscriber when the issue was first published.
- 2)You have to give a written notice on your first screen or title screen, where this type of phrase can be clearly noted by the user (an example) "Sound Routines from Twin Cities 128/64 - issue #32".
- 3)You have to send us a copy of the program on a disk or upload the program to our library on GENie. Do not send it by e-mail!
- 4)If there are any kind of charges for the program, either as commercial or shareware software, or if it is a "demo" for a company, contact us FIRST before releasing the software/hardware so we can talk about the licensing fee. This usually will be something small, such as copy of the finished product. If we find out after the fact it will cost you *MUCH* more. Only written releases from Parsec through the U.S. mail with our company stamp imprinted on the contract will be considered valid.
- 5)These routines may not be uploaded to any network.
- 6)These routines may NOT put into ANY disk library collection individual use only - no exceptions!

C-128, C-128D, C-64, CBM, and other names of Commodore equipment are trademarks of Commodore Business Machines. GEOS 64/128 are trademarks of GeoWorks. CMD, HD, RL, FD are trademarks of Creative Micro Designs. All other trademarks or servicemarks mentioned in this magazine belong to their respective owners and are mentioned for their benefit or for editorial purposes. Litewair, Lweir, RUR U2, Software Light Years Ahead of the Rest, Twin Cities 128/64, and TC-128/64 are trademarks of Parsec, Inc.

Dr. Octal's Q&A Clinic

Q:The review of gateWay V2.51 was good but I have heard there are some buglets, is this true? William Kempert Ont, Canada

A:There are always some "buglets" in any program, though software developers like to refer to them as undocumented features : -) The newest gateWay is pretty solid and I do not believe there are any major bugs left to impede your enjoyment of the software.

Q:From a technical point of view, could the C-128 keyboard be disconnected and the C-128D keyboard be plugged in? This way I could put the C-128 with my Ramlink and all the cables, mouse wire, and 1571 drive into a box and set the monitor on top. This would allow easy moving of the computer. William Kempert, ONT Canada

A:It most certainly can be done. I have seen this done and once someone on Q-link sent me a picture of their C-128 "clone" sitting inside of an PC XT case.

Q:I have seen the fan article from a previous TC-128 and I was wondering if you could recommend an industry standard model of a fan that I could find at my local hardware jobber. (Various)

A: TRW fan D24-B08a 04w4-000 12VDC 0.19a I got this number off a fan in a used C-128D I bought, so I know it works. TRW is about as standard as you can get!