



CHEATED ... BY A CHEATSHEET!

By BRUCE DUNN

To aid the memory of computer operators who can't remember all of the complex keystroke commands of sophisticated programs, there is a growing trade in "keyboard overlays" which consist of stiff plastic sheets with cutouts for the keys. On these sheets are printed a condensed summary of the operating commands for a particular program. These have sold well for business use, particularly for programs such as Wordstar, which is reputed to have some of the most cryptic commands ever used in a program. I recently purchased such an overlay for Paperclip (C-64 version), put out by "Leroy's Cheatsheets" who also produce overlays for other C-64 programs such as Easy Script, Super Base, The Consultant, and Multiplan.

In advertising for keyboard overlays for other computers, manufacturers frequently note that the overlay is "die cut". Any doubt that I may have had as to whether this is or is not a significant point when buying an overlay was erased when I tried to install the Cheatsheet overlay on my keyboard. The overlay does not have cutouts for the keys -- it merely has lines along which you are supposed to cut to generate the holes. I hunted up a ruler, a razor blade, and a piece of plywood to cut on. I carefully cut along the indicated lines. Tension mounted as I wondered whether I would finish the cutting without slipping and slashing either the Cheatsheet or my fingers with an errant razor stroke. Finally I laid the finished product on the keyboard. IT DIDN'T FIT! Cutting on the indicated lines gives holes which are too small. The overlay snagged the keys on all sides and wouldn't even slip down to the case of the computer. It took two more sessions with the razor blade to get the Cheatsheet holes big enough to accommodate the computer keys. The trimming involved brought the cutting lines within a millimeter or so of the printed information on the sheet -- when cutting you tread a fine line between cutting too little and not having the Cheatsheet fit, and cutting too much and destroying the information that you are trying to get onto your keyboard. Why the overlay is made this way I don't know -- are Commodore 64s smaller in the

U.S.?

Once the overlay was correctly cut, I placed the overlay on the keyboard. IT WOULDN'T LIE FLAT! The overlay is made of flimsy plastic-laminated cardboard. It isn't stiff enough to remain as a flat sheet, but tends to curl and lift at odd corners, getting in the way of your fingers as you type. Neither is it flexible enough to naturally mold itself to the slightly curved surfaces of some parts of the computer. I eventually ended sticking the overlay down with lots of Scotch tape. This is OK, I suppose, if you only need one cheatsheet, but must lead to frustrations if you have to switch overlays to use another program.

Once I got everything strapped down, I proceeded to type some material with Paperclip. The first command I couldn't remember was concerned with how to move a phrase in the text. Immediately the final weakness of this particular keyboard overlay was evident. Paperclip has so many commands available that they won't all fit on the available space of

the overlay. Some of them, including all commands for manipulating columns, blocks, and phrases are printed on the cardboard that you cut out when making the holes for the keys. I eventually ended up sticking these commands on the side of my printer with yet more tape.

Once I got all the commands adequately attached to my computer setup I found the cheatsheet quite useful. While the basic Paperclip commands are logical (CTRL L for load, CTRL S for save, CTRL E for erase etc.), the more complex commands are not easily remembered unless you use them frequently. You may for instance remember that Paperclip supports superscripting, but not immediately remember that the key sequence to start superscripting is EXC %, with superscripting ended by using ESC '. These and related facts are immediately in front of you using the sheet. I find that for a full explanation of how to use the commands however, the manual still must be consulted.

(Continued on page 7)

FROM THE EDITOR'S ~~PEN~~ COMPUTER

Our "lecture" meeting now takes place on the third Wednesday of the month. The location is the Emily Carr College of Art and Design on Granville Island. It's at 1349 Johnston Street, the main drag where the market itself is located. The November meeting, to be held in the College's auditorium, will feature Ken Bell, recently named Computer Personality of the Year by the Pacific Coast Computer Fair. Bell writes two columns on computing for the *Province* and is also that paper's business editor. Be sure and attend what promises to be a lively and informative evening on Wednesday, November 21 at 7 p.m.!

Along with this newsletter at the next couple of meetings, we're distributing free copies of a newspaper called *Input*. They were sent to us from Alberta, where they're distributed free in a large number of computer stores. You can get a subscription to this paper for a mere dollar a year (see the coupon on page four -- of *Input*, that is). How come it's so cheap? Because the paper is supported entirely by adver-

tising, a state to which we aspire as well. Editor and publisher Richard McGuire told me he got some static over the cover story crapping on Commodore's new computers. Seems like a certain computer company tried to pressure certain dealers into not carrying the paper!

Next issue of the newsletter will be in January to coincide with the annual general meeting, to be held the third week of that month. Watch for door prizes and other gimmicks to entice you to this affair. As always, submissions to the newsletter are welcome. They can be submitted on virtually any word processor. One rule: DON'T double space after each sentence, as we were all taught to do in typing class.

Some goofs with the Machine Language book being distributed in the 64 library: The file called ML10A should be renamed ML3C, and ML10B and ML10C should be renamed ML10A and ML10B respectively. ML10D should be deleted, since it's only one line long. Future copies of the disk will have these changes.

A COMPLETE GUIDE TO MACHINE LANGUAGE PROGRAMMING ON THE PET

By HAROLD BROCHMANN

LOOK UP TABLES [7-6]

You will recall from Chapter 1 that BASIC keywords are stored in a program in abbreviated form, using only a single byte known as a TOKEN. This approach saves memory space.

When the program is listed, the keywords are written out in full.

If the tokens in a BASIC program are examined we find that they are always represented by numbers in excess of 127.

The PET maintains a table in ROM in which all the keywords are spelled out in full. This table is to be found at \$C092 in BASIC 2 PETs and \$B0B2 in BASIC 4 PETs. We will now examine this table using the MEMORY function of EXTRAMON. Enter EXTRAMON and:

```
.M C092
.M B0B2 for BASIC 4 PETs.]
```

The display reveals not only the list of keywords but also a lot of error messages as well.

We see that the first keyword is END, the token for which is \$B0. The second keyword is NEXT which is represented by token \$B1 and so on.

The first three locations contain the codes for the keyword END. The E is stored using the number \$45 (69), while \$4E (79) represents the N.

We would expect that the D which precedes E would be stored with \$44 (68), but as it turns out \$C4 (196) is used instead. In fact, the last letter of every keyword is represented by a number which is \$B0 (128) greater than if that letter was not the last one in a keyword. Take the time to go through the list and verify this.

The code used to store the letters is the PET version of ASCII, also known as PETSCII. The last letter of each keyword is stored using PETSCII+\$B0 (PETSCII+128).

It is our intention to transfer the list of keywords to the screen. But in order to do so we have to convert each number to the appropriate screen code. Let us summarize what we know:

LETTER	PETSCII	PETSCII+128	SCREEN CODE
A	\$41 (65)	\$C1 (193)	\$01 (1)
B	\$42 (66)	\$C2 (194)	\$02 (2)
C	\$43 (67)	\$C3 (195)	\$03 (3)
...
Z	\$5A	\$DA (218)	\$1A (26)

A careful examination of this table will reveal that screen code for each letter can be obtained by subtracting \$40 (64) from the PETSCII code and \$C0 (192) from the PETSCII+128 code.

We will now write a ML program which will transfer the contents of one block of the keyword table to the screen, making appropriate code conversions for each byte. First we set up the indirect pointers in the usual location and then initialize the Y register to zero:

```
033A A9 00 LDA #000
033C 85 5E STA #5E
033E A9 80 LDA #800
0340 85 5F STA #5F
0342 A9 92 LDA #92 ($92 for BASIC 4)
0344 85 60 STA #60
0346 A9 C0 LDA #C0 ($C0 for BASIC 4)
0348 85 61 STA #61
034A A0 00 LDY #000
```

Then we load the accumulator with a byte from the table and jump to the subroutine which performs the code conversion:

```
034C B1 60 LDA (#60),Y
034E 20 57 03 JSR #0357
```

The accumulator which now contains the converted code is stored at the corresponding screen location. The Y register is incremented and tested to see if the entire block has been transferred. If not, the procedure is repeated; otherwise we return from ML to BASIC.

```
0351 91 5E STA (#5E),Y
0353 C8 INY
0354 D0 F6 BNE #034C
0356 60 RTS
```

In the code conversion subroutine, the contents of the accumulator are compared to \$B0 (128) to determine if we are dealing with PETSCII or PETSCII+\$B0. If the results of this comparison are positive we branch to another part of the

subroutine:

```
0357 C9 80 CMP #80
0359 10 FE BPL #035E
```

We now subtract \$40 (64) from the contents of the accumulator using EXCLUSIVE OR (EOR) instruction.

```
035B 49 40 EOR #40
035D 60 RTS
```

Subtracting \$C0 (192) is also accomplished with EOR:

```
035E 49 C0 EOR #008
0360 60 RTS
```

When this ML program has been assembled at \$033A, exit MICROMON and SYS 826. Hopefully, the first block of the keyword table will now appear on the screen.

ASSIGNMENT 7-6a

Write a ML program which will transfer all the keywords and error messages to the screen.

ASSIGNMENT 7-6b

We have seen that we can tell where each keyword in the keyword table ends because the last letter of each keyword is represented by PETSCII+128. It should therefore be possible to select a particular keyword from the table for printing on the screen by starting to search at the beginning of the token table and incrementing the X register every time an end letter is encountered. When the X register reaches a predetermined value one would extract the next keyword.

Write a ML program that selects the tenth keyword and prints it on the screen.

"NEW, IMPROVED" DOS 5.1 WEDGE

Are you tired of the usual colors on your 64 of a blue screen and light blue lettering? There's a nifty little program in Vol. 5, Issue 3 of *Transactor*, which makes the colors whatever you want and produces those colors whenever you hit RUN/STOP and RESTORE.

This short routine can be combined with the boot program for the C-64 WEDGE to kill two birds with one stone. Line 60 sets the color for the screen, which here is 1 (white), and line B0 is for the character color, which is 6 (blue). Choose other colors if these don't meet your fancy.

This program has some disadvantages. You can't change the screen or cursor colors while using it, for example. In order to do so, POKE 681 with the number of the color you want to change the screen to, and POKE 689 with the new number for the cursor.

There's a Wedge program with further improvements in the November/84 COMPUTE!. When this is created, it's stored under the name of DOS 5.1E. If you want to use the following with it, then make sure to add an "E" to the "5.1" in line 105.

```
20 PRINT"(CLEAR)"
30 AD=680
40 FOR1=ADTOAD+15:READX:POKE1,X:
NEXT
50 H1=INT(AD/256):LO=ADAND255
60 C=1
70 POKEAD+1,C
80 C=06
90 POKEAD+9,C
100 POKE770,LO:POKE771,H1
105 PRINT"(CLEAR){3 DOWN}LOAD{GR
Y2}" + CHR$(34) + "DOS 5.1" + CHR$(34)
+ ",8,1":PRINT"{4 DOWN}NEW"
106 PRINT"{2 DOWN}SYS52224"
108 PRINT"(HOME)":POKE631,13:POK
E632,13:POKE633,13:POKE198,3
110 1FA=1THEN SYS65126:A=1
500 DATA109,6,141,32,208,141
510 DATA33,208,169,13,141,134
520 DATA2,76,131,164
```


- A VIEW FROM THE OUTSIDE WORLD -

 Editor's Note: Although the following has nothing whatsoever to do with Commodore, I thought it might be interesting reading for people who are addicted to video games. Harlan Ellison, aside from being a well-known science fiction writer, has written scripts for TV programs and numerous criticisms of the "boob tube."

By HARLAN ELLISON
 (from Video Review, Sept/82)

Nothing in this world beyond the first seconds of a baby's birth is innocent. Nothing is precisely what it seems to be. Anything can be a paradigm of life's important lessons.

At first encounter, Parker Brothers' video game cartridge "Star Wars: The Empire Strikes Back" seems to be merely another of the seemingly endless permutations of the callus-producing rage that has swept an entire generation of Urphan-Annie-eyed, overfinanced, leisure-surfited teenagers into the maelstrom of electronic game madness.

But even the botulism bacterium looks innocent at first encounter. And I believe "The Empire Strikes Back" video game is an analogue for the Myth of Sisyphus.

Having never played a video game before, and having stared with creeping horror at the legions of silent, intense kids mesmerized in front of "Pac-Man," "Space Invaders" and "Donkey Kong" machines in Pizza Time Theatre parlors, I greeted the request to review this new cartridge with mixed emotions ranging from fearful curiosity to outright dismay.

I had no reason to think this fad was any more dangerous than stuffing phone-booths, swallowing goldfish, Hoola Hoops or wearing one's hair in imitation of Farrah Fawcett. Yet the vast amount of money being poured into these games, the accumulated years of time lost playing them, the apparent absence of any benefit to the players, had produced in me a *frisson* of concern. In a nation where reading is becoming an arcane art, where TV has become the universal curriculum, where the Lemming-like pursuit of mindless "entertainment" has taken on the noble obsessiveness of a search for the Holy Grail, I thought the inspired exploitation of the Star Wars totem in video-game form could emerge as the most virulent electronic botulism of all.

The Atari console system was rigged to a TV set in my home. I read the simple instruction brochure, and I proceeded to bore my ass off for the next few hours becoming as adept at "The

Empire Strikes Back" as I cared to be.

Kindly refrain from moaning that a 10-year-old can become more proficient at one of these twiddles than I, an adult at least in years, could ever be. Yes, a 10-year-old very likely could beat me 99 out of 100 times. But no 10-year-old I've ever encountered could create a Sistine Chapel fresco, write *Moby Dick* or copulate with any degree of expertise. Mind you, none of those are taught by any video game.

The extremely simpleminded parameters of Parker Brothers' "The Empire Strikes Back" are consistent with virtually all other video games. Destruction is the object. A line of two-dimensional Imperial Walkers plod toward a Rebel power generator on the Ice Planet Hoth (if you can believe those mundane pastel readouts represent an Ice Planet). You, as player, have to blow them up with blasts from the five Snowspeeder aircraft you are given. The object of the game is to destroy as many of the Walkers as you can before they reach the power generator and blow up the entire planet. (It takes 48 direct hits to neutralize a Walker.) Terrific object lesson for kids to learn; invaluable for everyday life in a world where Nuclear Holocaust paranoia already immobilizes us.

There's a lot of hurly-burly. Walkers change color and are weakened as a result of amassed hits; you can crash your Snowspeeder into a Walker; sometimes you acquire The Force and cannot be destroyed ... 32 variations of one- and two-player games.

But here's the bottom line, quoted directly from the rules brochure: "END OF THE GAME: The game ends when the lead Imperial Walker reaches the power generator -- or -- when the last of your Snowspeeders is destroyed!"

In other words, you cannot win.

The game ends when you lose. It may take you 10 minutes or 15 years. The level of your expertise may grow so elevated that the game will have to be concluded by your grandchildren, but ... you cannot win!

In classical Greek mythology we find the familiar legend of Sisyphus, founder and king of Corinth who, because of his avarice and fraudulence, was condemned to the lower world, eternally to roll a great stone to the top of a steep hill, whence it always rolled down to the bottom again. This ghastly punishment, perceived through the ages as a paradigm for the worst eternal fate that could be visited on an errant mortal, is spoken of thus in Webster's *Dictionary of Proper Names:*

"Hence, a Sisyphian task, an unending task on which immense energy is expended with little to show for it."

Hence, to play "The Empire Strikes Back" videogame, costing, with its console, enough to buy a good set of the collected works of Mark Twain, and fostering a solitude of activity that separates the player even more from the real world.

Over and over and over, you roll that great rock up the hill, killing Walkers only to have the rock roll down on you again, only to have faster, cleverer, more destructive Walkers come to life on the screen. And you play, and you play, and in the twilight you find the cobwebs have smothered your imagination, your leg has gone to sleep, your money is gone, your friends have grown up and died; and you are all alone there in the gloaming, with the radiant screen and its two-dimensional electronic death-machines ... firing, firing ... lumbering ... making no progress, and winning no awards.

But does it really matter? Clearly not. Because life -- as viewed by this and other video game *Body Snatchers* -- is a pitiless congeries of rocks being rolled up a steep hill, only to fall back. This is the lesson one learns from "The Empire Strikes Back" -- unless one has the presence of self to become rapidly bored.

That's a helluva recommendation: the best one can hope for is that one yawns before one's soul is snatched.

UPCOMING MEETINGS

Nov. 20 -- Lecture w. Ken Bell
 Dec. 4 -- Workshop
 Dec. 18 -- Lecture meeting
 Jan. 1/85 -- Get serious!!!

Published by The Commodore Computer Club, P.O. Box 91164, West Vancouver, B.C. V7V 3N6. Editor: Mike Wugley (321-8465).

Copyright 1984 by the Commodore Computer Club. Material in this paper may not be reprinted for profit without written permission. Opinions expressed are those of the individual authors, and not necessarily those of The Commodore Computer Club. The name "Commodore" is used with permission of Commodore Business Machines of Canada Ltd.

Club meetings are normally held: *Workshop*: first Tuesday of the month, 7:00 p.m., Thompson Secondary School cafeteria, 1755 E. 55th Ave. (near Victoria Drive); *Lecture*: third Wednesday of the month, 7:00 p.m., Emily Carr College of Art and Design, 1399 Johnston, Granville Island. These dates and locations are subject to change. For up-to-date information on any changes, please call the club's 24-hour answer phone:

PET-3311 (738-3311)

Club Executive: President -- Jim Bauerle; Vice-President -- Sigurd Steiner; Secretary -- Marvin Steinway; Treasurer -- Hu Rejine; Directors -- Robert de Boer, Guenter Hake, Jim Jorgenson, Terry Jutti, Murray Kopit, Mike Wugley, Elmer Koy, Philip Seligman, Nick Shevchenko, Tony Smith, Arthur Tamer, Al Townsend, Jim Wilcox.

THE COMPUTING CYNIC

By VICTOR VENTI

A long time ago, before I was interested in computers, I collected coins and stamps. Having accumulated a modest collection of pennies from about the last 50 years, I decided to sell them.

I took my treasures to a local coin dealer. He took a look at them and offered me what I thought a miniscule amount -- about 5 cents per coin. I thanked him and left.

I then walked across the street to a competing coin dealer and went through the same routine. He wasn't interested at all.

Figuring that 5 cents per coin was better than nothing, I returned to dealer number one. When I told him I wanted to take him up on his offer, he replied he'd now only offer me face value for the coins. When I asked why, he said something like: "Well, you didn't think my first offer was good enough and went across the street. If that's what you think of me, I don't think much of you either."

Young and naive as I was, I was so freaked out by this turn of events that I took the bus home, dumping my penny collection in the fare box to pay my way.

I was reminded of this merchant and his haughty attitude recently while leafing through some computer magazines. They publish a guide for prospective authors in which is included a line stating: "If you want to submit articles to us, don't send them to anyone else." This policy, in my opinion, stinks.

Let's say I have a business selling mink-lined computer chips. If a dealer I want to sell my product to says, "Don't deal with anybody else if you want to do business with me," he could be probably brought to court for monopolistic practices. Unfortunately, no such aid exists for journalists.

Why do magazines adopt a policy like this? Their rationale is probably: "Look, we're going to take a lot of time deciding whether or not we want to print your article, or whether it's worth editing into shape if you're not a very good writer. It also takes time to draw up contracts and allocate space for your article if it goes into the magazine. If you suddenly decide to sell it to someone else after we've gone to so much trouble, then we're wasting our time and money."

Much of this argument is quite reasonable. However, it doesn't change the fact that journalists are really at the

mercy of these publications. Unless they are extremely wealthy, most writers have to rely on the mail for sending their material, which, with its slow speed, does little to ameliorate matters.

I've heard some interesting tales from writers who have submitted articles. One of them waited six months for a rejection! By the time such an article runs the gamut of two or three publications, it could be a year or more before it sees the light of day. By that time, the topic of the piece could be outdated or even obsolete.

I can't see the reason for a horrendous delay of this length. If someone gives me an article, I can tell after reading it once whether or not it's publishable. If the volume of mail is so great that it contributes to lengthy holdups, then perhaps the publication should consider hiring more staff.

There is obviously no easy solution to this problem of a less-than-free marketplace for computer journalists, but it is something that publications with this kind of policy should

seriously address. There are plenty of talented writers around who, having gone through this grind, will just give up in frustration.

One of the most familiar lines in Commodore programs is "PRESS ANY KEY..." to achieve some desired result like advancing to the next screen, running a game, and so forth. Next time you see this prompt, try pushing the CONTROL key. Or the SHIFT key. Or COMMODORE key. Or RESTORE key. Or even the RUN/STOP key. Notice what happens.

Dumb product award for this issue goes to Cardco's tuners which make your monitor into a TV. Most people already own a device which will perform this same function. It's called a "video cassette recorder"!

Runner-up goes to the versions of Trivial Pursuit which you can play on your computer. I can't see the logic of these games. After you've played them once, surely you know all the correct answers!

PARANOIA DEPARTMENT

 BEING A LIST OF THINGS THAT CAN GO WRONG WITH YOUR DISK DRIVE:

1. SAVE WITH REPLACE: This command fails sometimes but the cause is not really known. It usually happens on disks that have a lot of files and not a great deal of free space. The effect is to chain into other programs or files on the disk.

2. RENAME: This fails sometimes even though the disk system gave the "00,OK,00,00" message. The failure may be due to there being scratched entries on a disk or the number of disk entries is a multiple of 8 (i.e., full block).

3. WRITE PROTECT TAB: Writing to a disk with a write protect tab is attempted. Then a command is later given to read the disk. Even if it has a write protect tab on, at least one write will be made. The solution is to power the disk drive down completely and then start again.

4. SEQUENTIAL FILES: If a sequential file of 254 characters (or any multiple) is written to the disk then an extra carriage return is added to the end of the file.

5. BLOCK ALLOCATE & BLOCK FREE: The best way to use these commands are to convert numbers

into strings and concatenate this to the command before sending the command down the error channel.

6. ILLEGAL TRACK & SECTOR: If illegal track or sector command parameters are given to a block command then partial overlaying of error messages result.

7. BLOCK FREE: If an unallocated block is freed, the block count is automatically incremented by one and thus an incorrect number of blocks free can be generated, i.e. >670. Validate will restore the correct number of blocks.

8. VALIDATE #1: If an error occurs while validation of disk is taking place, then the BAM will be left in an indeterminate state. Re-initialization of the disk is necessary.

9. VALIDATE #2: The validate command frees any sectors allocated for random access.

10. USING ASTERISK AS FILE NAME: * allows access of last LOADED or SAVED program. If the last program was saved with replace, then the * accesses the old version of the program (which has been scratched from the directory).

11. MEMORY READ: The byte returned by a memory read operation isn't accompanied by a carriage return so use GET to access the character.

-- from Compuserve

HARDWARE AND SOFTWARE

Reviewed by MIKE QUIGLEY

APROSPAND-64

An expansion board is required equipment for the VIC-20 if you want to get serious about computing, especially as far as extra memory is concerned. The 64, on the other hand, has plenty of memory, so why would anyone need a board for that computer?

Most 64 cartridges are of the auto-start variety, which means that when they're not in use, they have to be unplugged from the computer. An expansion board provides an alternative solution to this problem, thereby lessening wear on both cartridges and the expansion port. As well, if you have more than one cartridge, a board not only allows you to switch them in and out with ease, but provides a place to keep them all organized. There are also a few companies making cartridges which run in conjunction with each other (for example, Cardco's Write Now!, Spell Now!, File Now!, etc.) and an expansion board provides the only way of accessing them easily or all at once.

The Aprospand-64 from Apropos Technology provides a simple solution to all these needs. It offers four cartridge slots, each with a large, easily maneuvered on/off switch. The board is supplied with a fuse to prevent damage to your computer. There is also a reset switch of the momentary contact variety, which often relieves you of the tedium of turning your computer on and off. (It should be noted, however, that due to the 64's memory construction, the reset switch will not always eliminate material which is in the area usually reserved for cartridge programs.)

The board, as expected, plugs into the expansion port of your 64. The circuit board itself is sandwiched into a vinyl case which has been epoxied together. It is possible to split this case open if you want to make some modifications to the board or remove it entirely if you want to install some legs on the board (these are not supplied).

There are some limitations on the use of the Aprospand-64. The major one is that the maximum power to all the cartridges used should not exceed 1/2 amp. One suggested example of this happening is with 80-column display cartridges which require extra power. If the computer is unable to provide the necessary power in this case, it is recommended that a separate power supply for the board be purchased.

The reset switch of the Aprospand-64 is used to directly connect pin C (RESET) and A

(GND) of the expansion connector. There are theories in some circles that extra resistance should be inserted in such a line to prevent damage to the computer. When I asked Apropos about this, the technician dismissed such theories as "hearsay" and said that they had never received any complaints about their boards. Still, the rumours persist, just like the controversy over the 1541's "save and replace" command.

Another factor that should be noted is that there is slightly less than 7/16" clearance between the edge of each expansion slot and its accompanying switch. I've been informed that there are a couple of cartridges for the 64 which might not be able to accommodate this tolerance.

These limitations aside, Aprospand-64 is excellent value for the money, priced somewhat lower than the competing Cardco board. There is no mention of any warranty or user support in the enclosed operating manual (a single sheet of paper), but the company has a 10-day guarantee of "satisfaction or your money refunded."

(Aprospand-64. From Apropos Technology, 1072-A Avenida Acaso, Camarillo, CA 93010. Price: \$39.95 U.S.)

* * * * *

MR. TESTER

Most 64 owners have endured problems with their computers which had them shuddering at the idea of not only parting with their favorite toy, but also paying a costly repair charge to restore it to its former glory. One product which purports to be an anodyne for these fears is Mr. Tester from Micro-W. Distributing, touted as "a complete diagnostic for your Commodore 64 and 1541 drive."

Like the claims of Mark Twain's death, this statement is somewhat exaggerated. The program, which is on a copy-protected disk (but which doesn't cause the usual knock-knock), contains 9 tests, many of can be found for free, albeit in less fancy versions, in user group libraries.

These include a test for joystick operation in both "parts" (ports?), a video test to see if the monitor's colors are OK and whether the screen is properly centred, and a test for the computer's RAM memory to see if there are any bad chips. There is also an improved, speedy version of Check Disk (found on the 1541 test disk), and a "comprehensive printer test", which I couldn't make function because my interface wouldn't handle all the characters. In fact, I couldn't

even boot up Mr. Tester when my printer was connected. There is a warning for problems of this nature in the program's manual.

The test for cassette record and playback wouldn't work either with my system. When this was completed, a line reading "STRING TOO LONG -- ERROR IN 4230" appeared on screen, suggesting there was something wrong with the program.

A test for the SID chip consists of playing a major scale in each voice for 4 octaves. (What about the other 3?) This tells you only if the SID chip is functioning and gives some idea as to the quality of the sound. The keyboard test gives a diagram on screen indicating when individual keys are pressed, but doesn't work for the RESTORE key.

About the only part of Mr. Tester I found interesting was the "1541 speed/alignment test". It is divided into two parts, the first of which shows if the speed of your drive is correct. The second checks to make sure the drive head is aligned correctly. To do this, the 1541 makes its usual clattering noise like when NEWing a disk (one of the main reasons drives go out of alignment!) and then seeks out a specific track which you input. Since the tracks are relative in location to the hypothetical track zero (which the drive is seeking when the racket occurs), it's possible a disk recently created with a misaligned drive could give a reading showing that the alignment was correct. This doesn't seem to be much of a match for the more accurate method of using an oscilloscope and cats-eye alignment disk.

On the whole, Mr. Tester performed well when I ran its various sections, aside from the cassette and printer ones. That is to say, it found nothing wrong with my computer system. It does tend to check for "obvious" problems and tends to ignore other important ones like faults with the power supply. As such, it's no match for the investigative efforts of a good technician.

(Micro-W. Distributing, Inc. P.O. Box 113, Pompton Plains, NJ 07444. \$29.95)

* * * * *

LOADSTAR

Loadstar is neat!

And what is Loadstar, you ask?

It's a disk-based magazine emanating from the southern United States which is published every month. It arrives in the form of two disks. One of these contains programs, articles and

(Continued on next page)

(Continued from previous page)

reviews. The other is for feedback to the magazine, which pays return postage. (For purposes of this review, I was not supplied with the "feedback" disk, and thus can't say how it operates.)

The disk with the goodies on it is the epitome of "user-friendly". You can jump from one place to another in the menus or load programs just by using the joystick (or the keyboard if you're less lazy). Side one of the disk contains programs, the other articles and reviews. This double-sidedness is frowned upon in some circles, but the disk is not protected, so you can copy the back side to another disk if you're paranoid.

I received the first Loadstar for review. Side one contains three sections. The Cover Page is nothing more than a title screen, but it contains a mind-soothing bass hum which sounds great when fed into your stereo system! Paperless Pages contains an article on SOFTCON, a software convention held in New Orleans, and the first of what promises to be an excellent series on Commodore's Disk Operating System and its idiosyncracies. These articles appear on the screen like a continuous typewritten page of copy, and you can scroll through them forwards or backwards. You can also dump them out to your printer, and the text will appear in one or two columns across the page!

The third part of side one is called Influx, which is contributions from readers, presumably obtained with the feedback disk mentioned above. My issue contained articles on How to Start a User Group, reviews of Archon-Chess, Pogo Joe and Enchanter, Hints and Tips, User Requests, and further reviews of Software and Hardware.

While reading either Paperless Pages or Influx, a tune plays in the background. If you don't like it, you can turn it off.

Side two of this first disk contains several games, utilities, aids, and "novelties." The games aren't going to make you throw away your copy of Beachhead or Zaxxon, but they are clever and mostly of the strategic variety. Color Lister allows you to put an end to monochrome listings, while Disk Scanner, Sector Corrector, File Scanner and BAM Map 64 are all useful disk-oriented programs. There are two humorous entries -- Blip is the Blap of Bleep is a random aphorism generator, while How Smart Are You? will, as its description goes, "not answer the question." The menu for this side is especially clever. You can either select "ABOUT", which loads in another "paperless page" telling you, not surprisingly, what each program is about, or "RUN" which

actually runs it.

About the only thing I don't like about Loadstar is that if you decide to stop in the middle of a program, the only way you can get back to the menu is by typing LOAD "STAR",8, which is how you boot up programs on each side in the first place. (Most of the programs will go back to the menu when they are finished.) But this is a picayune quibble.

Loadstar costs \$9.95 U.S. per individual issue, or \$69.95 U.S. for a year's subscription. On the whole, it's intelligently designed, easy to use, and a real bargain at the yearly rate.

(Loadstar, 3811 St. Vincent, Shreveport, Louisiana 71108)

* * * * *

THE SOFTWARE PROTECTION HANDBOOK

According to its ads, *The Software Protection Handbook* from PSIDAC will help you "blow the locks off protected software!". At the same time, it claims it "does not condone piracy."

This latter claim is a bit hard to take after reading the first page. The book was originally to be called "The Software Pirate's Handbook II." However, its name was changed because "several advertisers flatly refused to advertise a book of that title in spite of the fact that the book was never meant to encourage piracy in any form. (The use of 'Pirate' was intended as a light-hearted reference to any copying process, and to inspire a certain tendency [sic] of humankind; the attraction to things mysterious or secret." Sure, sure... On the second last page of their 212-page book, PSIDAC advertises "Software Pirates' T-shirt White Skull and Crossbones on jet-black Shirt." This seeming tongue-in-cheek attitude reminds me of people who sold alcohol during Prohibition for "medicinal purposes only".

The book, co-authored by

David Thom and Vic Numbers (yes, that's correct) does live up to the first claim above, even though it admits that software protection schemes (and the subsequent methods of breaking them) become obsolete as fast as they are created. Methods on how to break into disk, tape and cartridge programs are all covered. The last two can be done with the help of hardware which PSIDAC will be glad to sell you at additional cost. There are numerous programs listed throughout the book for examining disks, checking for and creating errors, duplicating disks and individual files, creating autoboots, and so forth. All these programs can be purchased on disk from PSIDAC, again for more money. (I was not supplied with the disk and didn't try any of the programs for purposes of this review.)

Probably the most interesting part of the book is its first chapter, which examines the legal aspects of copying software, and in doing so, gives new dimensions to the phrase "rambling discourse." It does have a few points which I agree with, such as the fact that a person should be able to back up and customize software for his own purposes, and that software is grossly overpriced. However, what is one to think of opinions like these when placed in the context of fuzzy thinking like the following passage: "Copying for sale, distribution or other non-personal uses is Piracy. ... Loaning your original to another person for temporary use is not piracy. ... However, copying an original you do not own is unethical."

The book gives the appearance of being well-made with a plastic spiral-type binding and glossy cover. The typography inside leaves a great deal to be desired, however, using an IBM style of typewriter and listings from what seems to be a Commodore printer. The book is full of unbearable illiteracies on practically every page, ranging from three-letter words like "its" on up.

(*The Software Protection Handbook*, from PSIDAC, 7326 N. Atlantic, Portland, Oregon 97217. Price: \$19.95; disk of programs \$16.95 or \$29.95 for the two purchased together.)

—VIC LIBRARY RUNNING LOW ON GAS—

Although the VIC library contains 24 tapes and 15 disks, that's probably all it's going to contain. No more disks and tapes are contemplated in the immediate future for the reason that no one is interested in contributing original programs. Most of the recent efforts, consisting of programs typed in from magazines, have been contributed by two people -- the VIC librarian and the previous VIC librarian -- along with a couple of disks obtained from TPUG.

The Compute Gazette disks are currently available through Glenn Hazlewood, the 64 librarian, and both RUN and AHQY magazines are offering programs on disk. This may not be much consolation to those people who have only a Datasette.

Still, there is a good selection of programs in the tapes and disks which are available. To borrow any of these (for a refundable \$5 deposit), see Mike Buckley at any of the club's regular meetings.

TIPS ON USING PAPERCLIP

By BRUCE DUNN

If you are using Paperclip version B or C: don't. Take a moment and get a copy of Paperclip D or whatever is the latest version. Be sure to get the descriptive information you need to update your manual (B to C changes are described in file "new paperclip" on C disks, and C to D changes are described in file "64d notes" on D disks). Each update adds new features, and eliminates bugs. There is only one major inconsistency between the various versions of the program. When you underline or otherwise use special print features with the earlier Paperclip versions, the underlining or whatever automatically is cancelled at the next "return" (left arrow) symbol. Paperclip D, however, continues underlining even past a return symbol, until it hits an underline stop character. You may have documents on file written with Paperclip B or C in which you have underlined titles, etc. by starting the underlining function, then letting the program shut off the underlining at the next return symbol. If you try to print out these using Paperclip D, you will get entire documents underlined. To avoid this, you merely have to insert the underline stop character where appropriate. The same holds true for some other special print modes.

Go to the trouble of customizing Paperclip so that it is easy to use (see "Setting Up Paperclip", September newsletter). If you are using a Cardco interface, set up the printer file so that it automatically locks the interface in the no-ASCII translation mode when it prints. Embed the printer file in the program. Then use the defaults program to select the screen colors you like, and if you use the 80 column display feature, set up the program so that it gives you 80 columns automatically without having to press the f8 key.

Disk space is relatively cheap. Since Paperclip can be copied using normal BASIC commands, put a copy of it on each disk that you use for word processing. This means that you have to deal with only one disk each time you want to do word processing. Also, since disk space is relatively cheap, it does not pay to erase files. Instead, keep the files as a permanent record of what you have written.

If you do much word processing, consider keeping separate disks for the different varieties of material that you type (personal letters, business

letters, notices etc.). This makes it much easier to keep track of the documents that you have typed.

Paperclip has a fabulous "screen read" feature that will save you endless trouble (manual, page 3.7). When the document that you are working on is going to be saved, move the cursor to the first letter of the document name that you have typed after "file=" in the comment line at the top of the document. Type CTRL S to save, and when the program asks for the file name, instead of typing out the name, hit the run/stop button. The name of the document will be read from the screen. This may sound trivial, but it is important when you are modifying a document that you already have on a disk. If you wish to resave the modified document over top of the old version on the disk, you must use exactly the same name, with the same spacing and abbreviations. Using the "file=" comment and the screen read feature each time you save ensures that you will get the name of the document right.

When you want to load a document, the screen read feature is also very handy. Usually by the time you have a few documents on a disk you can't remember exactly what you called a particular document. To

load a document that you can't quite remember the name of, type CTRL 0 (zero) to get the disk directory. When you recognize the name of what you want to load, press the CTRL key. The screen cursor will reappear, and you can position it over the first letter of the name of the desired file. Then press CTRL L for load, and run/stop to read the name of the file directly from the screen.

CHEATSHEET

(Continued from page 1)

Although at one time I have read the complete manual, I was somewhat surprised to spot on the cheatsheet several little used commands that I had forgotten existed. One of these was quite helpful, offering an easier way to do some block manipulations than the way I had been performing them. In this respect the cheatsheet appears to be very complete, even to the extent of including commands for the optional Spellpack for Paperclip.

Overall, my reaction to Leroy's cheatsheet for Paperclip is that it can be very useful to someone using the more exotic features of the wordprocessor. However, anyone wanting to buy one must be prepared to spend some time and frustration installing it on their computer, and probably must resign themselves to taping it to the computer to keep it from snagging keys and fingers.

CASSETTE USE HINTS

LOAD ERROR...a programmer's nightmare when you've just LOADED a program from your Commodore Datasette and this glaring message blinks up on your screen. What did you do wrong? Was it your program, the Datasette, sunspots, poltergeists, or what? There are several techniques which can help you load program tapes more reliably. Here are a few of them:

1. TIGHTENING NEW TAPES: If you're using a new tape, either a blank tape or a pre-recorded tape, it's always a good idea to FAST FORWARD the entire tape and then REWIND it before using it. This tightens the tape on the spool and reduces the possibility of load errors due to loose tape in the cassette.

2. POSITIONING THE DATASETTE: Placing the Datasette too close to your television set may expose it to interference which can produce LOAD errors. Try placing the Datasette away from the television set, and avoid coiling the Datasette cord. Commodore tests show that Datasette placed on top of a television set produces more LOAD errors than Datasets positioned away from the set

with the cord fully extended.

3. SAVE:SAVE: When SAVEing programs on tape, it's always a good idea to save it twice. You can do this in one step. For example, if your program name is "MAGICIAN," you should save it twice by typing: SAVE 'MAGICIAN': SAVE 'MAGICIAN' and hit RETURN. These are actually two identical commands separated by a colon. After the first MAGICIAN is SAVED, the second command is read and the program is recorded again. Pre-recorded tapes are often recorded more than once so if one program is damaged or if you get a LOAD error, you can type LOAD again and get the next program on the tape.

4. CLEANING AND REALIGNING TAPE HEADS: Like any tape recorder, the Datasette requires servicing to clean and demagnetize the tape heads and to adjust the head alignment. If your Datasette has been in use for some time and it is producing errors, it may require servicing.

These techniques should improve your tape loading techniques...whether you're using a VIC, 64, PET or CBM with your Commodore DATASETTE tape recorder.

STOP DESTROYING YOUR DISK DRIVE ... USE

By MR. MIKE

Tired of subjecting your 1541 to unnecessary stress every time it loads one of these overly protected commercial programs? Yes, every time your drive rattles like when NEWing a disk, certain critical components are becoming unglued ever so slightly, leading up to the dreaded condition called "misalignment" and an expensive trip to the nearest repair depot.

Up until recently, the only way that errors on disks which cause the knock-knock which is no joke could be fixed was to go into the programs and change the machine language. In the last while, a one-liner has appeared in several publications which tells the drive to ignore the slam-bam-thank-you-ma'am. Only hassle is you have to type this item in (as well as remember it) every time you want to use it.

But what are computers for? Aren't they for saving us from performing such menial tasks? The answer to this skill-testing question is "YES", and the result is "Antiknock", a short program utilizing the famous "dynamic keyboard" of your computer where instructions are printed on the screen and then executed.

It's quite common to see instructions of a one-line nature printed in magazines to be typed on screen in "direct mode" (with no line numbers) before a program is LOADED and RUN. These often have the effect of changing memory locations, especially on the VIC 20, where creating programmable characters with expanded memory is a real hassle.

The dynamic keyboard prints these instructions on screen, then executes them by taking information stored in the keyboard buffer (a place where letters are normally stored when you're typing fast, so you can actually type ahead of what's appearing on screen). The cursor is sent to the top of the screen with the "HOME" key, and then a series of character 13s (CHR\$(13)) -- which is the code for "RETURN" are dumped from the buffer, causing the cursor to land on each instruction and execute it. An example of this is found in the program "H1-RES EDITOR" in the 64 library where there are no less than three consecutive examples of the dynamic keyboard.

Antiknock sends a command to the DOS (Disk Operating System) telling it to stop wasting time trying to read errors (which cause the drive to freak out, hence the banging). It has been tried, and worked successfully, with the following commercial software: Master Composer (Access), Synth-64 (Abacus), Songwriter (Scarborough), Creative

Writer (Creative Software), and Raid on Bungling Bay (Broderbund -- two errors). It also worked with a program which checks for errors and usually knocks on every single sector of an

errored track (a real pain in the drive). It will not stop the rattling when trying to format a disk.

So use it, and feel instant relief!

```

10 POKE53281,0:POKE53280,0
20 PRINT"(CLEAR)"
30 PRINT"{YEL}<REV>ANT1KNOCK<RVD
FF)"
40 PRINT"<DOWN><CYN>THIS PROGRAM
  ALLOWS YOU TO LOAD
50 PRINT"COPY-PROTECTED SOFTWARE
  WITHOUT
60 PRINT"HAVING THE DISK DRIVE M
  AKE RATTLING
70 PRINT"NOISES CAUSED BY INTENT
  IONAL ERRORS

80 PRINT"PLACED ON THE DISK (IN
  MOST CASES).
90 PRINT"<DOWN>FIRST OF ALL, INS
  ERT PROGRAM DISK!
100 PRINT"<DOWN><WHT>1) <CYN>LOA
  D FIRST PROGRAM ON DISK
110 PRINT"<3 SPC>(EQUIVALENT TO
  <PUR>LOAD"+CHR$(34)"0:*"+CHR$(34
  )+" ,8,1<CYN>
120 PRINT"<DOWN><WHT>2) <CYN>LOA
  D SPECIFIC FILE NAME

```

```

130 INPUT"<DOWN><REV>(LTGRN)YOUR
  CHOICE<RVOFF>";NM
140 IFNM<10RNM>2THENPRINT"<2 UP>
  ";GOTO130
150 IFNM=1THEN170
160 IFNM=2THEN180
170 LO$="0:*"+CHR$(34)+" ,8,3":G0
  T0200
180 INPUT"<DOWN><LTBLU>FILE NAME
  ";NA$
190 LO$=NA$+CHR$(34)+" ,8,3"
200 PRINT"<DOWN><REV>(RED)ARE YO
  U SURE?<SPC>(Y/N)<WHT>
210 GETA$:IFA$=""THEN210
220 IFA$="N"THENRUN
230 PRINT"<CLEAR><3 DOWN>OPEN15,
  8,15:PRINT#15,"+CHR$(34)+"M-W"+C
  HR$(34)+" ";
240 PRINT"CHR$(106)CHR$(0)CHR$(1
  )CHR$(133):CLOSE15"
250 PRINT"<2 DOWN>LOAD"+CHR$(34)
  +LO$:PRINT"<4 DOWN>RUN"
260 PRINT"<HOME>":POKE631,13:POK
  E632,13:POKE633,13:POKE198,3:NEW

```

CRUNCHING...

You can pack more instructions and power into your BASIC programs by making each program as short as possible.

Crunching programs lets you squeeze the maximum possible number of instructions into your program. It also helps you reduce the size of programs which might not otherwise run in a given size; and if you're writing a program which requires the input of data such as inventory items, numbers or text, a short program will leave more memory space free to hold data.

ABBREVIATING WORDS

On Commodore machines BASIC commands can be abbreviated. For example, POKE can be abbreviated by P<SHIFT>O. The most frequently used abbreviation is the question mark (?) which is an abbreviation for the PRINT command. However, if you LIST a program that has abbreviations, the computer will automatically print out the listing with the full-length keywords. If any program exceeds 80 characters with the keywords unabbreviated, and you want to change it, you will have to re-enter that line with the abbreviations before saving the program. SAVING a program incorporates the keywords without inflating any lines because BASIC keywords are tokenized by the computer.

SHORTENING PROGRAM LINE #'S

Most programmers start their programs at line 100 and number each line in intervals of 10. This allows extra lines of instruction to be added as the program is developed. One means of crunching the program after it is completed is to change the line numbers to the lowest numbers possible because longer line numbers take more memory than shorter numbers.

PUTTING MORE CODE ON A LINE

You can put more than one instruction on each numbered line in your program by separating them by a colon. The only limitation is that all the instructions on each line, including colons, should not exceed the standard 80 character line length. Here is an example of two programs before and after crunching:

Before Crunching

```

10 PRINT 'HELLO...'
20 FOR T=1TO500:NEXT
30 PRINT 'HELLO, AGAIN...'
40 GOTO10

```

After Crunching

```

10 PRINT 'HELLO...':FOR T=1
  TO 500:NEXT:PRINT 'HELLO,
  AGAIN...':GOTO10

```