Obligatory Stuff

If you have any questions about CUGS please feel free to contact any of the above executive members.

The Monitor is published monthly by the COMMODORE USERS' GROUP OF SASKATCHEWAN (CUGS), Regina, Sask. CUGS meetings are held the FIRST WEDNESDAY of every month (unless otherwise noted) at Miller High School. The next meeting will be held: **December 2, 1992 from 7:30 - 9:30 p.m.**

CUGS is a non-profit organization comprised of C64, 64C, C128, and 128D users interested in sharing ideas, programs, knowledge, problems and solutions with each other. Membership dues are pro-rated, based on a January to December year.

Anyone interested in computing is welcome to attend any meeting. Out of town members are also welcome, but may be charged a small ($5.00) mailing fee for newsletters. Members are encouraged to submit public domain software for inclusion in the CUGS DISK LIBRARY. These programs are made available to members. Any member is entitled to purchase DISKS from our public domain library for a nominal fee. Programs are 'freeware', from computer magazines, or the public domain. Individual members are responsible for deleting any program that he/she is not entitled to by law (you must be the owner of the magazine in which a particular program was printed). To the best of our knowledge, all such programs are identified in their listings. Please let us know if you find otherwise.

Coming in December

CMD Hard Drives
by Tristan Miller

A special CUGS Christmas gift will be given to all members attending in December

Editorial
by Jarrett Currie

It was many years ago, in November, that I allowed my name to be nominated as the Assistant Editor of this newsletter. After my first year, I was elected as Editor, and have enjoyed producing the Newsletter monthly and the Disk Catalogue yearly every since. Unfortunately, the time has come when I can no longer juggle my editorial responsibilities with my work responsibilities. To that end, this will be the first year in several that I won't be running for Editor.

But, the Monitor is not lost. If you are reading this editorial at the general meeting, you have the opportunity to save the Monitor by volunteering for the Editor position. You don't have to be a graphic artist (goodness knows, I'm not), nor do you have to have past experience (I certainly didn't). There are many members in our midst who would be glad to give you encouraging comments, and many who would be glad to give you a hand.

Although I use GeoPublish to produce the Monitor, your choice of tools is not limited to GEOS. You can use the product of your choice, and if you ask loudly enough, I am sure the members of the club will supply you will all the needed tools you require. After all, the Monitor helps to bring all the club's members together, and is an invitation to non-members to give CUGS a try. A small investment in the Monitor can produce many rewards for the club.

I have recently become involved with the Pascal language, and one of the versions that I use is for CP/M. When many people hear the letters CP/M, they chuckle automatically, in the same way they do when they hear the word disco. But, just as thousands of discos were cramped to capacity with

these very same people, so too was CP/M running on thousands of computers, controlling businesses, playing games and managing finances. I have written an article for this Monitor edition describing this CP/M Pascal version, as well as comparing it to Super Pascal 128.

I was introduced to CP/M when I was learning programming, and remembered how confusing it sometimes seemed to be. But, when I began using MS-DOS at work, I was amazed at how similar these two operating systems are, and because of my desire to use the Pascal compiler I had, I decided to learn a little bit more about CP/M on my C128. It became clear very quickly that CP/M is a powerful alternative to CBM's DOS, and contains facilities for advanced computing. If you have had some exposure to MS-DOS, and have the C128 Programmers' Reference Manual, learning CP/M is really a simple matter. In fact, most CP/M programs and utilities contain HELP files that can teach you how to use them just by browsing around. To get the most from CP/M however, you need to have access to CP/M software. There is really only one source for it now – the public domain. Having a CompuServe account is definitely an advantage, although the club does have some CP/M disks already available. So, if you want to give the Pascal compiler a try, don't hesitate to spend some time learning CP/M. You may find the experience exciting.

EXECUTIVE MEETING MINUTES

submitted by Garth Strawford

☞ Phone blitz to be done for garage sale to be held at November meeting

☞ Earl to be returning officer for elections at November meeting

☞ November door prize: 1350 mouse donated by Tristan

☞ December meeting Door Prize not decided

☞ December presentation: CMD Hard Drives by Tristan Miller

☞ Ken to arrange coffee and snacks for December meeting

☞ To celebrate the Christmas season, a disk will be given away which will consist of one side of music put together by Ken and one side of graphics put together by Stan and Tristan

President's Message

by Barry Bircher

Hello all and welcome again to the CUGS regular meeting. This month's meeting will see our annual club elections take place. I ask that anyone who is willing to help the club out, to please let their name stand for any of the positions available. Namely, we need someone to fill the Editor, the assistant Editor and the Vice President positions. Those positions are important to insure future viability of the club. I have been President now for 4 years and feel my term as President is best left for someone new who can spring up new ideas for running the club. The Vice President position is rather lax in its few responsibilities. The general idea of the VP is to gather experience and to learn how the club activities are run, how to initiate the executive monthly meetings and generally get the feel of how to run the club and be the backup to the President. One or two years down the road, the VP would take over the President position.

Also, the Editor position is open for someone who would like to learn more about desktop publishing. I believe we have one of the better, if not the best, looking newsletters of all the clubs I've seen over the years. Jarrett has done a VERY VERY good job of it. I believe he has learned alot about making newsletters and the knowledge he has gained as Editor is just waiting for a new Assistant Editor. He uses GEOS to make the newsletter and has learned alot about using GeoPublish. But, however the newsletter is actually made is not important and is up to the Editor to decide how he will do it. GeoPublish is one of the better newsletter makers and I urge anyone who wants to learn more about GEOS and making newsletters, to volunteer for this position.
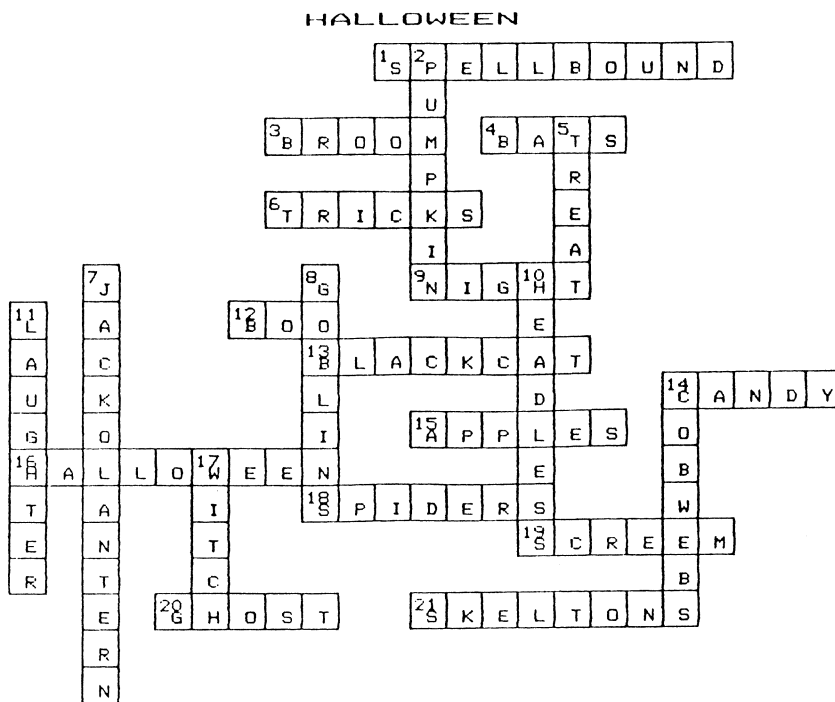
Tonight, we will be having a garage sale of various items the members have brought for sale. If there is something you like, buy it. It will be run just like those weekend garage sale we always see around the neighbourhood.

In other news, CMD has done it again. They have just announced a new drive. This drive is like a 1581 in that it uses a 3.5" disk. The difference is that the FD-4000 can store over 3.2 megs of files and the FD-2000 can store over 1.6 megs, as well as being able to read the regular 1581 formatted disks. This drive is capable of partitioning the drive just like the CMD HD drives, RamLink and RAMDrives do for easier sub-directory handling. Check out the Nov-Dec/92 issue of "RUN" for a review on these drives and the CMD advertisement for the prices.

Talking about drives, the executives has tentatively set up the December meeting presentation to show off the CMD Hard Drives. Tristan Miller has volunteered to do a demo on how easy it is to use the HD and plans on doing a speed comparison between the 1541 and the HD.

In the past, we have made up a special Christmas disk containing various related holiday season programs, music and graphic slide shows. This disk will be assembled by the club Librarians, Garth and Stan. A copy will be given away free as a Christmas gift courtesy of CUGS.

Solution to Last Month's Crossword

HALLOWEEN

(Crossword solution grid — answers:)

1 SPELLBOUND
2 PUMPKIN (down)
3 BROOM
4 BATS
5 TREAT (down)
6 TRICKS
7 JACKOLANTERN (down)
8 GHOUL
9 NIGHT
10 HAUNTED (down)
11 LAUGHTER (down)
12 BOO
13 BLACKCAT
14 CANDY / COBWEB (down)
15 APPLES
16 HALLOWEEN
17 WITCH (down)
18 SPIDERS
19 SCREEM
20 GHOST
21 SKELTONS

Head-to-Head Pascal

by Jarrett Currie

Recently, I became frustrated with the limitations of BASIC, and decided to adopt a new programming language. Because of its structured language features, I decided that Pascal would my language of choice. To that end, I decided to acquire a Pascal package that would satisfy my very demanding standards. The two Pascals I chose to compare were JRT CP/M Pascal Version 3.0 by JRT Systems and Super Pascal 128 by Abacus Software. The former is now in the public domain while the latter is extraordinarily difficult to find.

For those of you C128'ers who have not yet tried out your CP/M disk, or for those of you who gave up on CP/M, JRT Pascal may be a good reason to give CP/M a try. CP/M is an old operating system that ran many businesses, and Commodore blessed all C128 owners by supplying CP/M with this machine. It isn't the simplest operating system to use, but once you become familiar with it, it becomes second nature, and it becomes easy to see why it was so popular.

Like GEOS, CP/M isn't useful until you have CP/M programs to run with it. And like GEOS, a session with CP/M usually includes running several programs to accomplish what you want. When I use JRT Pascal, I use the following programs - each of these are either supplied with CP/M or are in the public domain:

VDE266.COM (public domain) - this is an excellent text editor/word processor whose little brother, VDE.COM is on one of the club's disks. Although VDE.COM is vastly superior to the CP/M editor, once you use VDE266.COM, you may very well decide to give up your favorite Commodore word processor!

SEARCH.COM (public domain) - this utility program will search the contents of a group of files for particular text strings. This is useful if you use standard procedures in many of your Pascal programs, and a change to the procedure requires recompiling each of the programs.

KEY.COM (supplied) - this utility program allows you to redefine your keyboard. Without it, only the top row of cursor keys are used with CP/M. After you use KEY.COM, you can redefine the cursor keys to the left of the shift key.

In addition, I set up my CP/M disks to allow for time and date stamping, as well as allowing archiving bits to be set. This greatly facilitates the use of the copy program, PIP.COM. Because I have a RAM expander, I can copy all the files I use into the expander (with a batch file), and then I can use PIP to copy back only the changed files once I am finished with my programming session.

CP/M also uses my CMD Hard Drive, although between the Hard Drive's manual and the little information I had about CP/M, I had a difficult time creating a bootable CP/M partition on the Hard Drive. Luckily I found a program, FORMAT.COM, on CompuServe that can format a variety of CP/M disks.

For whatever reason they decided to do so, Abacus Software recorded Super Pascal 128 on a non-copy protected disk, that while the disk can be easily copied (they encourage you to do so), it cannot be used with the normal CBM DOS. They invented their own DOS routines, so you cannot use the editor of your choice, you cannot use any of the directory/DOS utilities you may have in your collection, nor are you able to use your Hard Drive. Because of the limitations this imposes on the programmer, I would be interested to know their reasons for selecting this type of format for their disks.

JRT Pascal comes with a CP/M formatted manual of 180+ pages on disk. There is no joy in printing this document out, because it was originally formatted with embedded form feeds for a word processor I do not have. Although printing it is a

painful process, having the manual in printed form is a requirement for using the compiler successfully. The manual is also missing a couple of pages; these pages are missing in the archive. The Super Pascal manual comes in little 3-ring binder, making referring to its contents a breeze. Because Super Pascal 128 so closely resembles the 64 version, the Super Pascal 128 manual is really the 64 manual with an included addendum. Both manuals require that you know the Pascal language. Fortunately, the Public Library contains a wealth of information for Pascal, ranging from beginners' to experts' levels.

JRT Pascal provides for many features in addition to standard Pascal:

JRTASM   - an assembler package (Z-80)
CRTMAP   - a procedure to allow creating input/output
             screens for records
LETTERS   - a procedure to display/print block letters
INDEX   - allows for indexed file processing (!)
SEARCH   - table searching
PICTURE   - a procedure for formatting numbers (like BASIC's
             PRINT USING, but much more elaborate)
JSTAT   - for statistical programming
JGRAF   - for rudimentary graphing
ACTIVAN   - to analyze the running of your Pascal program

JRT Pascal allows for virtual memory; a program can be written that calls many other routines (which can call many other routines) without any memory conflicts. It will automatically allocate and release storage as needed by the system, and does so intelligently so that a section of code that is often used will not be swapped out of memory. This facility is all done automatically, and allows for the creation of massive programs.

Super Pascal does not extend the standard Pascal language as much as JRT Pascal, although it, too, has some very interesting facilities. Most notable are:

- 80-column VDC graphics support
- A built-in assembler so a Pascal program can have imbedded assembly language programs; the assembler is called when the compiler encounters the assembly language routine
- The use of a small RAM disk (within the normal RAM memory)
- A utility program allowing direct manipulation of disk blocks, as well as other DOS housekeeping utilities

Super Pascal does allow for programming swapping and memory manipulation, although it does so in a relatively awkward, and non-standard way. Except for the CONTINUE command, which causes the start of a new program (without a return to the calling program), transferring to other programs requires a considerable amount of familiarity with memory addresses and layouts. JRT Pascal simply bypasses these difficulties with its automatic memory management. However, although having to manipulate memory under program control can be a disadvantage to the applications programmer, for those programmers with a bit of knowledge of assembly language, this flexibility can be a great benefit.

The one area where Super Pascal shines is the speed at which it can compile a program if it is stored in the RAM disk. Even when using a RAM disk with JRT Pascal, Super Pascal can compile circles around it. Although perhaps not a great selling point, even a small difference in speed can be appreciated when small changes to programs are being made.

Although both of these versions of Pascal are fairly good, they both have distinct disadvantages. JRT Pascal tends to be a bit sluggish, however, that is clearly a problem with CP/M and not the compiler. This sluggishness is also present in VDE266 (the text editor), so setting up the editor correctly, and resetting the responsiveness of the keyboard is very important. VDE266 allows you to use different terminal settings, and includes one for the C128. Also, many of the features you expect to use on the C128 are missing, simply because you are using CP/M: screen windowing, ESC sequences for the editor, simple color changes (they require ESC sequences), and low-res graphics (I still haven't figured out how to change the character set).

Super Pascal also has disadvantages. The only editor available is supplied with the package, and is very similar to the BASIC editor, including line numbers. Although the editor provides for shifting lines around, and renumbering, this is little consolation when writing programs that are often written from the bottom up (unlike BASIC where you tend to write from the top down, Pascal is written by producing code at the bottom of the program, and creating support procedures at the top of the program). Constantly renumbering and shifting is a decided nuisance. Its greatest disadvantage, however, lines in the unorthodox memory management. Pascal is a standardized language that encourages structured programming. This virtue is greatly threatened when the program you create must directly manipulate memory. Lastly, because Super Pascal programs cannot be distributed as-is, but rather must be on a special run-time disk, creating a Pascal program for the public domain is impossible, unless the receiving person has Super Pascal. JRT Pascal, on the other hand, needs only 1 COM file to run the Pascal programs, and of course, the entire Pascal package is in the public domain.

Feature for feature, JRT Pascal wins hands down. And it greatly outshines Super Pascal 128 in cost (JRT Pascal is free). However, if you are more interested in tinkering with your computer's memory, and C128 features are important to you, then Super Pascal is right for you.

I will be creating a CP/M disk containing the Pascal system and documentation and donating it to the club. The cost for me to download this system from commercial BBSs, as well as downloading the archiving programs to dissolve it, is significant. Therefore, if you decide to purchase a copy from the club's library, please remember the effort that people go through to keep the library interesting, and donate programs yourself.

THE WRITE DISK

by Barry Bircher

Last month you may recall a discussion about opening and writing to disk files. The last issue explained how a simple program is used to read and write to disk files. It was the bare bones, no frills, nothing but the grits and gusto, brute force routine to get the job done. This month, I want to build on that program and add in some disk error catching routines. When a disk error occurs, this program will communicate to you what the disk drive's problem is.

What would happen if you were to attempt to RUN the program previously discussed more than once (see October, 1992 issue of the Monitor)? Once a file of a name has already been created on the disk, the DOS will not allow you to make another with the same name. The program will attempt to open up another file with the same name, and that is not normally allowed (I say "normally" to satisfy you disk hackers/jockeys who say they have techie ways of doing it, by not by using normal means). In Commodore DOS, each file on a disk must have it's own unique name and will not save 2 files with the same name. After all, how else would it know what file you wanted if 2 files have the same name? If you attempt to save a file with the same name, the disk drive will sit there blinking it's little light as if to say,

"HUH, no can do, ya user, pick another name!"

If you wish to overwrite the file of the same name with new information, you could add in "@0:" in front of the filename. This is formally called, "SAVE WITH REPLACE"

eg. 10 open 2,8,2,"@0:my first file"

This is telling the drive that you want to save a file with the same name and delete the old information and put in the new. The drive will save the file as it did in the past, but now, when the CLOSE 2 command is given, the drive will update the disk directory with that same filename and delete (scratch) the old file. There has been some debate over the integrity of that function, as there have been cases were the file simply gets corrupted and is lost in never-never land. I personally have not experienced this problem to any great extent worth noting (due to the fact that I have the new version ROMS in my machine). The official/recommended method to avoid this potential disaster is to first scratch the old file, then save the new one. The command "OPEN 15,8,15,"s0:filename":CLOSE15" will delete a file called "filename" from the disk.

In the above case, if you were to run the program again for the second time, the disk error light will start to flash. You may already know what the problem was because you are reading this and were hopefully paying attention and realized what had happened. But, what would you do if the disk light was flashing and you didn't have the foggiest idea what the heck is wrong with the drive? You would know that the drive had encountered an error. But what error? Is there a way to talk to the drive? YES there is.

An additional couple of lines is in order to help read the disk error channel. Our modified program thus far should read:

```
5 OPEN 15,8,15
10 OPEN 2,8,2,"MY FIRST FILE,S,W":GOSUB1000
20 PRINT#2,"THIS IS MY FIRST DATA FILE!"
30 CLOSE2
40 OPEN 2,8,2,"MY FIRST FILE,S,R":GOSUB 1000
50 INPUT#2,A$
60 CLOSE2:CLOSE15
70 REM A$ NOW CONTAINS THE STRING "THIS IS MY FIRST DATA
FILE!"
1000 INPUT#15,DS,DS$,TR,SC
1010 IF DS <>0 THEN PRINT DS,DS$,TR,SC
1020 RETURN
```

Line 5
Opens up a special channel #15 to the drive. It is the second 15 in the statement that is the important one; the first could be just about any number as long as you use it throughout the program whenever you are talking to the disk drive.

Lines 10 and 40
Added in a GOSUB 1000 to our error checking routine explained below.

Line 60
CLOSE both files when done.

Line 1000
Is our new disk error checking routine. It grabs information from the command channel. DS contains an error number, DS$ contains the error text, TR and SC is the track and sector the error occurred on.

Line 1010
This line simply checks to see if the error number is anything else but 0 (zero). Zero indicates an "OK-no problem". If there is no problem, then it's fair to say, "no news is good news". If DS is greater than 0, then there is some sort of problem, so let's print the information we got from the drive so we can see it on the screen and figure out from there what the problem was with the drive. Each error type on the drive is associated with an error number and an error description. We could check for a specific error if we wanted to, but anything other than error number 0 is generally bad news and means the previous command has not been executed.

Line 1020
Sends us back to the calling routine. Since this is a routine that you could use elsewhere in the program, why not just make it a subroutine and be able to call it from anywhere in the program just by saying GOSUB 1000? If there is an error, the routine will print it out.

Commodore 128 users have it much easier when programming in 128 mode. The BASIC in the 128 already checks the error channel and assigns the error to DS and DS$; all the programmer has to do is check DS or DS$. They perform the same thing as lines 1000-1020.

OK, (I hear you saying), you can read information from the drive, but can you talk back to it? Yes, in fact I have given one example previously of talking to the drive when I explained the SCRATCH command.

OPEN 15,8,15,"s0:filename":CLOSE15

or in the case that channel 15 has already been opened, just type:

PRINT#15,"s0:filename"

After a scratch command is given, you may want to GOSUB 1000 to check if there is a disk error (like FILE NOT FOUND). There are several commands that we can communicate to the drive, the most notable are as follows (assuming channel 15 has been opened already by entering OPEN15,8,15):

PRINT#15,"S0:FILENAME" - erases/scratches the file
PRINT#15,"N0:DISKNAME,ID" - formats a disk as "diskname" with and id of "id"
PRINT#15,"R0:NEWNAME=OLDNAME" - renames the "oldname" file to "newname"
PRINT#15,"U0>M0" - changes a 1571 drive to 1541 mode.
PRINT#15,"U0>M1" - changes a 1571 drive to 1571 mode.
PRINT#15,"C0:DUPLICATE=ORIGINAL" - makes a copy of a file

There are several other commands to give the drive, but they are listed in the users' manual in the section on the disk drive, so I will not repeat those here. You may be wondering why there is a '0' after the letter of the command. I put it there to maintain a consistency in the disk commands. Commodore originally had Dual Drive disk devices when the early systems came out (eg 4040, 8050 etc).

One confusing thing that occurs when talking about "device" and "drive" is that alot of people use the two terms interchangeably. People talk (myself included) about drive 8, or drive 9, when they actually mean device 8 and device 9. The drive number is actually either 0, or 1, on device 8, or 9. A dual drive would have a device number (like 8) but there had to be a way of distinguishing between the two drives, hence the drive number. The '0' refers to drive 0 and '1' refers to the second drive, so that the commands could be directed to one or the other drive on a device #. On single drive systems (like most people have) the '0' seems redundant, but is recognized as a valid drive number. Creative Micro Designs has brought out new drive systems that extends the use of those numbers to select partitions. I urge you, especially you programmers, to learn to use those numbers and put them in even though it seems redundant on your system. The CMD drives can be thought of as multi-drive devices and are able to recognize not only drive 0 or 1, but also 2, 3, 4, 5 on up to 254.

By printing or getting from channel #15 we can communicate to and from the drive.

Well, that's about it for space for this month's article. Next month I will take this program a little bit further and talk about using the disk STATUS byte.

EXPERTS LIST

The following members have agreed to be experts in some area of Commodore computing. If you have some expertise that may be of some assistance to other club members, please consider allowing your name to be listed here.

Wordprocessing

Paperclip (to version E)	Jarrett Currie	757 2391
Paperclip (any version)	Ken Danylczuk	545 8644
Pocket Writer	Barry Bircher	543 8840
Pocket Writer	Real Charron	586 1843

Spreadsheets

Pocket Planner	Barry Bircher	543 8840
Better Working SS	Ken Danylczuk	545 8644

Databases

Pocket Filer	Barry Bircher	543 8840
Oracle (Consultant)	Ken Danylczuk	545 8644

Communication

Desterm 2.0	Barry Bircher	543 8840
Dialogue 128	Jarrett Currie	757 2391
Library files	Barry Bircher	543 8840

Music/Sound

(Most)	Ken Danylczuk	545 8644

Languages

Forth	Ken Danylczuk	545 8644
Pascal	Ken Danylczuk	545 8644
ML (machine language)	Ken Danylczuk	545 8644
ML (machine language)	Barry Bircher	543 8840
BASIC (2.0-7.0, files)	Ken Danylczuk	545 8644

Graphics

Print Shop/Master	Ken Danylczuk	545 8644
Koala Painter/Printer	Ken Danylczuk	545 8644

Hardware

Disk Drive Maintenance	Ken Danylczuk	545 8644

GEOS

GEOS 64	Jarrett Currie	757 2391
GEOPublish 64	Jarrett Currie	757 2391
GEOS 128	Barry Bircher	543 8840

CP/M

CP/M 128	Jarrett Currie	757 2391
Archiving Programs	Jarrett Currie	757 2391