the MONITOR

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 User's Group of Saskatchewan** (CUGS). Meetings are held on the first Wednesday of every month in Miller High School's cafeteria annex, unless otherwise noted. The next meetings will be held on **October 6 and November 3, 1993 from 7:30 to 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** and **shareware** software for inclusion in the **CUGS Disk Library**. These programs are made available to members at $3.00 each (discounted prices when buying bulk). Some programs on the disks are from computer magazines such as *COMPUTE!'s Gazette* and *RUN*. 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 the original program was printed). To the best of our knowledge, all such programs are identified in their listings. Please inform us if you find otherwise.

Another benefit of club membership includes access to our disk copying service, to make backups of copy-protected software (this service is intended for **backups only**).

Any members who own a modem and wish to call our bulletin board will receive increased access to the message and file areas. The board is run off a 128 and 40 megabyte CMD hard drive, and operates at 300, 1200, or 2400 baud, 24 hours a day, 365 days a year.

EDITORIAL

by Tristan Miller

Greetings! With this issue starts my new job as an (the?) editor for the CUGS newsletter. I've tried to keep as close as possible to the original format of *The Monitor*, but then again, I suppose I should be at least *little* bit original.. As you can see, there aren't so many graphics and fonts as there were in previous issues - this can be attributed mainly to the fact that I lack the appropriate software and hardware. I have next to nothing in the line of publishing software on my 64 or 128, but, more importantly, I don't even have a printer. So, you may be asking, how did it come that this newsletter even exists? Well, I'll be quite honest with you. I'm using an IBM.

Yes, I can already hear all the gasps of amazement and disapproval from all of you. Yes, I suppose I really shouldn't have resorted to writing this with WordPerfect on an MS-DOS system instead of borrowing a Commodore-compatible printer and some publishing software for my 64, but it still just wouldn't have cut it. I have no grandiose multi-disk drive setup, nor do I have the benefit of an REU or any other wonderful little goodies someone would definitely want to write a professional-looking newsletter on a 64 with. All I have is the keyboard, the monitor, and the disk drive, which would make the job painfully slow. I believe most of the other editors were fortunate enough to have a much better setup, and so they'd have probably spent far less time and effort putting together the club's newsletter.

By now, you may start thinking I'm just trying to make excuses for not using a Commodore machine to write this newsletter. But I *DO* want to make a point here, and that is that traditionally, articles such as the Editorial and President's Message have often been forums for speeches about why you should hang onto your beloved 64s and 128s and resist the great urge to buy an MS-DOS system. I'm not about to change this ritual entirely, but I would like to take a more logical approach to it. First of all, let me state that Commodores are good at what they're intended to do, and IBM systems are also good at what *they're* intended to do. I can think of far better things to do than type out the same, boring article about how the 64 and 128 can

be just as good as any other system, and that they're still useful. I think that argument has long been exhausted. Yes, you CAN *really* improve your 8-bit system. There are dozens of devices that will speed up your computer tenfold, increase your memory by a hundredfold, and expand your drive space by a thousandfold. With GEOS, a mouse, a hard drive, an accelerator, and a RAMDrive, you can make your computer just as good as any IBM. But doesn't that also mean the IBM is just as good as the 64, too? What it all comes down to is that any computer can probably make any other computer its equal.

Today there is often a lot of "temptation" to go out and buy yourself a brand new 486SX with a few megabytes of memory, a CD-ROM and hard drive, and SuperVGA. All I say is get what you need (or think you'll need in the near future). Obviously, if the only problem you find with your Commodore is the disk access speed, you needn't rush out and buy an expensive $5000 system when you can purchase JiffyDOS or a second-hand REU for a fraction of the price. However, if your needs are much more demanding, you may certainly want to look into getting something a bit better. And it needn't be an IBM/compatible, either. If your main hobby is writing music and you'd like something with a few more sound capabilities, pick yourself up an Amiga 500. But if you do upgrade, don't abandon your Commodore. Regardless how much "better" your new computer will be, the 8-bits still have a lot of potential for home use.

Well, enough of my babbling about computers. I'd just like to thank everyone who contributed to this month's *Monitor* - I believe this is one of the biggest (if not *the* biggest) issues we've had so far. This month, we have several new articles and game reviews, courtesy of Lyndon Soerensen, a column on secret POKEs by Carl Reilly, and several older items I've pulled from previous issues of the newsletter, including Sir Richard's BASIC, a very long-running series on various aspects of BASIC programming, written by our former president, Richard Maze. Extra special thanks go out to Byron Purse for the use of hardware, printing, and graphics.

TAKE MY LAPTOP, PLEASE

by Dan Gutman
from Q-Link, Sept. 1989

Portable laptop computers will help you increase your business productivity. **Bullpucky.**

Laptops, in fact, cut DOWN your productivity. They're the biggest time wasters since the invention of television game shows. The problem is that the instant you pull out a laptop and begin working on it, somebody will inevitably come over to annoy you. Laptops attract interested passersby like lightbulbs attract moths.

Recently I was in the library researching a baseball book I'm writing (it will be out next season). I take notes on my trusty Radio Shack Model 102 instead of on paper, so I can zip them easily into my desktop computer when I get home. It saves alot of time - or so one might think.

I was minding my own business when I see this guy out the corner of my eye, looking intently at my computer.

"I see you have the 102," he says (as if we should be buddies because he knew the model number). "How do you like it?"

That's how it always starts. If a stranger comes up to you and asks you how you like your computer, you should instantly know two things about him.

1. He's boring. People joke about insurance salesmen and accountants, but there's nothing worse than a computer literate bore.

2. The only way to end a conversation with him is to be rude.

The correct course of action would have been to fake chest pains and run screaming out of the library. Unfortunately, I politely told the man how much I like my laptop and how useful it is to me.

Then he says, "Have you got a minute?"

Before I could escape the reference department, he tells me that he's starting his own company and everything will be computerized. In a flash, he whips out a sheet of paper and sketches the floor plan with computers, laser printers, file servers and cables connecting the whole shebang together.

AS IF I CARE! A half an hour later, I was still sitting there, disbelieving, listening to this guy.

"That's great," I finally say, looking at my watch. "I wish you all the luck in the world."

Mind you, this guy didn't know I write about computers for a living. He simply saw me using a laptop and decided that I was the person he should tell the story of his life. If I hadn't had my laptop with me, he wouldn't have even said hello.

And if he *had* known I write a computer column, it would have been WORSE. He probably would have taken me hostage and forced me to listen to everything he knows about local area networks.

This is why laptops don't increase productivity, they REDUCE it.

The ads for laptops always show well-dressed businesspeople busily working in cabs, trains, and planes. Work on an airplane?! Are they out of their minds? When the other passengers see you with a laptop, they act like you're Thomas Edison bringing the first phonograph to the patent office.

Unless you're sitting next to another laptop user on a plane, you're going to get it from both sides.

"Isn't that cute!"

"Hey boy, what'chew do with that thang?"

"Young man, I understand these devices disrupt the pilot's navigational system."
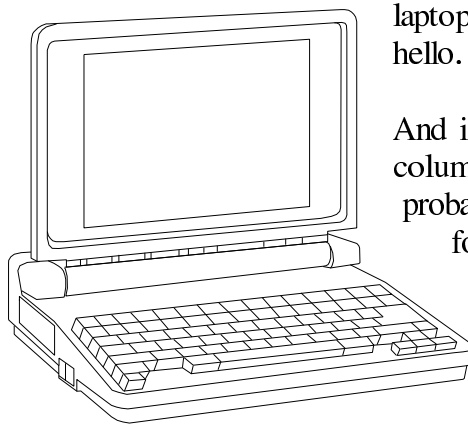
By the time the conversation is finished, you're in Cleveland and you haven't even loaded your word processor yet.

Using a laptop in public is a great conversation starter for lonely people to make new friends. And it's a great way to meet women. But it's a lousy way to get any work done.

CHAMPIONS OF KRYNN

by Lyndon Soerensen

...Quickly, Redmond withdrew his blade from the attacking orc. The stone floor was getting slippery as all around him, his companions battled their own adversaries. Footing was getting treacherous. A yell off to his left - Hortense, cleric and healer of their band, was struggling with a gigantic troll.

"Redmond...!" the cleric gasped between warding off the slashing claws of the troll. "Get the eggs!"

The dragon eggs. The party's goal was to retrieve the silver dragon eggs that were stolen from the good dragons by the minions of Takhisis. They were on the altar ahead of Redmond, wrapped in a dirty burlap sack. The fighter began making his way towards the altar while trying to defend himself from the rest of the ambushing monsters that sought to try and stop the party from completing their mission.

Just a couple of steps away now... Suddenly, his view of his goal was blocked by a massive ogre. "KILL ENEMIES!" it roared. Redmond lifted his blade and swung. His sword connected with the leather leggings wrapping the monster's legs and sliced through them to bite the sinews. The ogre screamed in pain and swung his huge mattock at the fighter, but, blinded with hurt, its aim was poor and Redmond was able to dodge and thrust his blade into the stomach of the ogre, killing it. The monster fell with a crash, and Redmond stepped over the monster to grab the sack. "Lucky the filthy thing didn't crush the eggs," he thought. He grabbed the bag and turned back to give whatever help his companions needed to slay the rest of the enemy force.....

The above scene is just one event in a very detailed Fantasy Role-Playing game by Strategic Simulations Inc. (most known for their simulations of famous war battles). The game is set in the 'DragonLance' world invented by TSR Hobbies (the makers of Advanced Dungeons and Dragons). The plot of the game is very simple: The people of the land of Ansalon have only recently defeated the evil dragons and their armies who sought to set up as ruler, their Dark Queen, Takhisis, and plunge Ansalon into despair and ruin. The goal of the game is to foil this plot by rescuing the eggs of the good dragons that have been stolen by the enemy to be perverted to evil ends.

In order to do this, you must assemble a band of six adventurers and travel the continent to stop the armies from succeeding. You can choose from four different character classes, each having its own strengths and weaknesses. You can also include magic users, whose powers wax and wane with the phases of the moon.

What Strategic Simulations has done is to try to bring the excitement and adventure of AD&D to the computer. They have well succeeded. At each stage in the completions of the various quests you must undertake to reach the final goal, there are hundreds of monsters to fight, many beings that will aid you in your journeys or hinder you. The graphics are well done. The playing screen itself is split into three different areas: in the upper left is what your characters see; in the upper right is their current condition; the bottom half of the screen is reserved for messages and menus.

The game is controlled almost entirely with the joystick, with both horizontal and vertical menus used. In order to make a selection, you just move the joystick to highlight the action(s) you want and press the button. The animation is flowing and the colours make all that you see come alive.

The one thing that will take most of your time when playing this game is combat. Even here, the user interface is well designed. The screen changed to a 3-D close-up view of the combat area, where you can advance characters up to the front lines in order to take the place of those that are wounded. Characters spin around to meet the attacks of monsters behind them, and backstab those monsters that have their backs to them.

Magic-users can cast spells that affect the entire combat field, and you can see the effects as they target enemies. Each spell and arrow loosed is accompanied by sounds as well as the swish and thunk of blades used by your fighters.

Fantasy role-playing games generally take a long time to solve, sometimes several hundred hours. However, when you play this one, you don't think about how many hours a day it will take you to get to a specific goal, you worry about how long it will take you to defeat the monsters guarding the temple. An engrossing game, and well worth the effort to solve it.

Word Processing

Paperclip (to version E)	Jarrett Currie	757-2391
Paperclip (any version)	Ken Danylczuk	545-0644
Pocket Writer	Barry Bircher	543-8840
Pocket Writer	Real Charron	586-1843
Pocket Writer	Tristan Miller	584-1736
Fontmaster II	Michael Rodgers	728-2595

Spreadsheets

Pocket Planner	Barry Bircher	543-8840
Better Working SS	Ken Danylczuk	545-0644

Databases

Pocket Filer	Barry Bircher	543-8840
Oracle (Consultant)	Ken Danylczuk	545-0644

Communication

DesTerm 2.0	Barry Bircher	543-8840
(Most 64 terminals)	Tristan Miller	584-1736
Pro128Term	Jarrett Currie	757-2391
Library Files	Barry Bircher	543-8840

Music/Sound

(Most)	Ken Danylczuk	545-0644
Stereo Sid Editor	Michael Rodgers	728-2595
Stereo Sid Editor	Tristan Miller	584-1736
Stereo Sid Player	Tristan Miller	584-1736
Enhanced Sid Editor	Tristan Miller	584-1736
Sids in BASIC programs	Tristan Miller	584-1736

Languages

Forth	Ken Danylczuk	545-0644
Pascal	Ken Danylczuk	545-0644
ML (machine language)	Ken Danylczuk	545-0644
ML (machine language)	Barry Bircher	543-8840
BASIC (2.0-7.0, files)	Ken Danylczuk	545-0644

Graphics

Print Shop/Master	Ken Danylczuk	545-0644
Koala Painter/Printer	Ken Danylczuk	545-0644
Doodle!	Tristan Miller	584-1736

Hardware

Disk Drive Maintenance	Ken Danylczuk	545-0644

GEOS

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

THE ONE-LINER TRADING POST
Part One
Compiled by: E. Carl Reilly

Over the past 14 years of Commodore programming and the frustrations because of it, I've managed to fill quite a few notebooks with the tricks of the trade. In the midst of my hysteria, I've compiled a listing of the most usable and efficient POKEs, PEEKs, and one-liners that my numb fingers could type. Some are computer-specific and others are not. If the description of the hack does not mention which computer it is for, then just try it out and hope for the best.

I do hope that you find these hacks useful. Some of them I've created myself and others have come from other programmers of which I cannot remember. So, if you do recognize a hack that's from another author, then please take a moment and bow to another Commodore guru who is out there. And onward we go....

To view the last file accessed by the computer, type:
```
SYS 62753   - C128
SYS 62913   - C64
SYS 63065   - VIC 20
```

To move the cassette buffer to 192 bytes below the bottom of string storage:
```
A=PEEK(51)+256*PEEK(52)-192:POKE179
,A/256:POKE178,A-256*PEEK(179)
```

To prevent a BASIC program from LISTing on a C64:
```
POKE 774,0:POKE 775,141
```

To have the LIST command reset the C64:
```
POKE 774,226:POKE 775,252
```

To have the LIST command RUN a program on a C64:
```
POKE 774,113:POKE 775,168
```

To re-enable the LIST command on a C64:
```
POKE 774,26:POKE775,167
```

To disable line numbers while LISTing a BASIC program:
```
POKE 22,35 - VIC 20/C64
POKE 24,37 - C128
POKE 24,27 - Re-enable the C128 LIST
```

To jiggle the screen on a C64:
```
10 FOR T=0 TO 15:POKE 53270,T:NEXT:
GOTO10
```

Here's a handy one to keep in your pocket. Use this hack to remove ALL levels of subroutine RETURNs:
```
SYS 42352    - C64
SYS 50544    - VIC 20
```

Here is a universal software reset routine that will work on every Commodore machine (that I've come across, anyways):
```
SYS PEEK(65532)+256*PEEK(65533)
```

The C128 has the ability to LIST a program from within a RUNning program. To do this on C64, type:
```
POKE 768,174:POKE 769,167:LIST
XXXX- YYYY:POKE 768,139:POKE
769,227
```
Where XXXX is the starting line number (if any) and YYYY is the ending line number (if any).

To create a quick beep on the C64:
```
POKE 54273,70:POKE 54278,249:POKE
54296,15:POKE 54276,17:POKE
54276,16
```

To change the cursor speed on the C64:
```
POKE 56325,X
```
Where X is a value from 0 - 255 (0 being the fastest and 255 being the slowest)

To have the C64 wait for the SHIFT key to be pressed:
```
WAIT 654,1
```
NOTE: The CTRL, SHIFT, C=, or a combination of these three can be checked through this address or 653.

To have a reliable random number generated (from 0-255):
```
POKE 54287,255:POKE 54290,129:PRINT
PEEK(54299)
```

VIC 20/C64 LIST hider:
`POKE 755,200`

Re-enable LIST:
`POKE 775,167` — C64
`POKE 775,199` — VIC 20

To disable the keyboard on the VIC 20 or the C64:
`POKE 649,0`
To re-enable:
`POKE 649,10`

To enter the monitor on power up of the C128, press the RUN/STOP key while turning on the power. NOTE: If you're resetting the C128 and you use this feature, then the last program in memory will still be there.

To disable the C128 RUN key:
`POKE 4104,0`
To re-enable:
`POKE 4104,9`

To produce a 'no question mark INPUT' on a C64:
`POKE 19,1:INPUT "<text>";A$:POKE 19,0`

To disable the key repeat on a C128:
`POKE 2594,64`
To re-enable:
`POKE 2594,128`

Here's a nifty little hack that'll save you from many a headache and that 'brain oozing out of the left ear' syndrome. A one-liner un-NEW!
`POKE 2050,1:SYS 42291:POKE 45,PEEK(34):POKE 46,PEEK(35):CLR`

To simulate a PRINT@ statement with the C64:
`POKE781,ROW:POKE 782,COL:POKE 783,0:SYS 65520:PRINT"<text>"`

To read the current position of the cursor on C64:
column=`PEEK(212)`:row=`PEEK(214)`
This one is great if you need to print something to the screen and then return to the position where you left from.

To view a hidden message in the C128:
`SYS 32800,123,45,6`

(Note the values in the A, X, & Y registers: 123 45 & 6. Hmmmm....)

To SAVE the function key definitions to disk on a C128:
`BSAVE"<filename>",U<drive#>,B0,P4096 TO P4352`
To LOAD, just
`BLOAD "<function keys>",U<drive#>`

To obtain 450 baud on the 1600, 1650, or 1660 modems:
`OPEN5,2,3,CHR$(0)+CHR$(0)+CHR$(12)+CHR$(4)`

And that's not all of them, but I hope that I've supplied you with enough to keep you going for a little while. I've tried to supply you with some hacks that you've never seen before and I hope that I've succeeded in doing so. If you find that one or more of these hacks do not work for you, then please leave me e-mail here on CUGS [BBS]. I'd appreciate the feedback on these tips, and I'd like to know if you'd like more of them.

Here's a quick note to you C128 owners that use the 80-column screen. Just last week a sysop had asked me to supply him with a hack that would turn the VDC off or at least simulate it. He had found that his EBBS system had burned an image onto his screen and he didn't like to keep turning his monitor off and on all the time. Well, the following is what I've come up with. While it does not turn the VDC off, it does blank the screen very effectively and it leaves the C128 running as normal. But the main advantage is that if you have a program that's running and it takes a while for it to complete its task (like de-LZHing a huge file), you can still hear the 'BING' telling you that it's completed its happy job that it does so well! Anyways, this is what I've come up with, and if you can think of a better one, then please let me know.

`POKEDEC("D600"),1:`
`POKEDEC("D601"),XOR(PEEK(DEC("D601")),80): POKEDEC("D600"),6:`
`POKEDEC("D601"),XOR(PEEK(DEC("D601")),25)`

This string of POKEs will toggle the OFF and ON HEX values.

MICRO CITY:
A Review
by Tristan Miller

Anyone who has ever owned a modem in this city has *undoubtedly* heard of Micro City, "Saskatchewan's telecommunity", and "the largest BBS in Western Canada". However, it seems to me that there appears to be a great lack of Commodore users on this system. In light of this fact, I am writing this article to introduce Micro City, and review its good and bad points.

Over the years, Micro City has grown from a small two-line system to a fully functional twenty-line telecommunications service. Among other things, it boasts an online arcade with single- and multi-player games, a large file area containing mainly MS-DOS files, an online dating service, online shopping, and most of all, its online conference room, in which up to twenty users can converse simultaneously.

I decided to buy one of the "credit packs" ($6 plus taxes grants you ten hours of online time) at my local confectionary and see just how good this system really was. All that needs to be done to get on the system is call their new user line at 569-2886, answer a few questions, supply them with a mandatory alias, and you are released into the board. From there, one would simply have to go to the Bank menu and cash in a credit pack. Basically, what they are are little folded cards that are stapled shut. Inside is a combination of numbers and letters that you type in to get your access. Piece of cake.

First, I investigated the online games. They proved to be very exciting, and there is something that would appeal to just about anyone. From Dungeons & Dragons to Star Trek, it would seem that you'd never run out of games to play (though you certainly would run out of time to play them all). Most of the games require an 80 column screen and/or an ANSI terminal, which isn't so much of a problem as long as you have the appropriate terminal program. The rest of the system, though, functioned perfectly in standard ASCII in 40 or 80 columns.

The file area was indeed large, and offered a wide range of local and CD-ROM files for IBM machines. If you own an IBM, you'll find this area quite useful. However, most of you that only own 64s and 128s will find this section of Micro City to be rather worthless.

The shopping mall proved to be nothing more than a collection of cheap advertisements. The same can be said for the online dating service; though it seems like a nice idea, nobody really seems to use it (and please correct me if I'm wrong). It's deteriorated into advertisement - just something to look at.

Micro City also claims it has a "100% private" email system, and stresses that phrase quite a bit. In fact, Clint Krismer, the current sysop, gives you his word that not even he can read your mail, or access any private files you send to someone else. **Hogwash.** I asked him about the circumstances surrounding this statement and he eventually told me that he COULD indeed read your private mail if he really wanted to. Technically, he cannot, as the sysop, access your email, but he can always log on AS you (having access to all your account information, etc.) and read it. Now why he'd want to do this, I have no idea, and I highly doubt he would, but I just thought I'd mention that there *is*, in my opinion, a certain amount of deceit here.

Last stop - the teleconference. This seems to be the facet of Micro City most talked about amongst its users. And indeed, it is everything it was made out to be. The teleconference is beautifully structured. Basically, you type what you have to say, and when you hit return, it is displayed to everyone else who is in the teleconference. But it doesn't stop at that! You can also issue actions, such as "smile Gary", "punch Bob", and "apologize Dave". If, for example, your alias was Genie and you issued the action "wave wizard", your terminal would display something to the extent of "You wave to Wizard!", Wizard's screen would say "Genie is waving to you!", and everyone else's screen would read "Genie is waving to Wizard!". There are dozens of actions, many very humorous, and many very useful.

In addition to the actions, there are also commands, such as the "forget" command, which, when used on a user of your choice, filters out all output generating

from that user - a very useful thing to do if certain people start to get a bit annoying. And, of course, you can always decide to "remember" them again. Other features include live online multi-player blackjack, bingo, and poker, public and private ("whispered") messages, private chatting, and the creation of your own private room, into which you invite only those who you want.

Generally, whenever you enter the teleconference, there will be three to fifteen other users. A wonderful way to have some fun when you're bored, and meet some new people.

And all this for "only" one cent a minute. (you can feel the bad news coming, can't you?) As it turns out, this board is the single most expensive (read "rip-off") pay bulletin board system in the city. Assuming you were to spend about an hour and a half on the system every day (and believe me, this is very possible), at the end of the year you'd've amassed a staggering bill of $330 ($353 with taxes)! This even makes QLink seem cheap. Compare this price with the SAME amount of time on the four other major boards in the city that have much the same features: it's 1.5 times more expensive than Atmospheric Environment, over 6.5 times more expensive than Unibase, over 8 times more costly than DLC-West, and a ridiculous 16 times more than FACT!

It's not the price I find so astonishing as the manner in which the system is advertised. In some computer stores, they'll have a little glass-encased paper ad. On it, it promises online games, files, and the other features. The way in which this is presented is absolutely despicable! To the average computer user unfamiliar with modems, it would seem as if this was a really great place and one heck of a deal. It boasts "NO LONG DISTANCE FEES", among other things.

Someone ought to tell them that Micro City is definitely NOT CompuServe! There's just no way the two can compete. They really are desperate to suck people into paying all this money for something they could ordinarily get for free. It would be a great idea if someone posted a BBS list next to that ad; then at least nobody would go about forking over twenty-five bucks a week and think they were getting a fantastic deal with all the files they've downloaded and all the games they've played. Yes, that's the truth. You can get every single one of Micro City's supposedly "unique" features on just about any BBS you call. Sure, not every board has everything Micro City has, but I'd rather call one board for files, one board for games, and one board for online conferencing for free than pay a truckload of money to do it all on Micro City.

My advice: if you can spare the six dollars and you think you may be interested in online conferencing, go ahead and buy a credit pack to see what it's like. If you figure that it's fun enough to spend that much money on, then that's great. But do not, I repeat, do NOT use Micro City for the purpose of files, dating, shopping, or games! For these, I'd recommend a membership on any one of the local non-pay bulletin board systems here, and if you find you need a bit more time, or a few more files, perhaps consider buying a membership on one of the four aforementioned systems. You'll get just as much, and you'll probably cut your spending dramatically.

QLink Network News:
FINDING BASIC STRINGS

by Bob k7 Kober

There is the POINTER command in the C128's BASIC 7.0, that can be used to locate any string's descriptor, and from it, the location of the string in memory. The BASIC 2.0 in the 64, however, doesn't have such a command, so finding the strings is a little more difficult. Herein is an explanation of how and where strings are stored, and a program in BASIC and ML that will actually find them.

Where string data is stored

Strings in the C64 can be stored in two areas. They can be located in the actual BASIC text area, or in the string storage area, located at the top of BASIC program area.

If you have a line in your program like this:

```
10 A$="HELLO":B$="THERE"
```

both the "HELLO" and the "THERE" would be located in the BASIC text area. If you were now to have a line following like this:

```
20 C$=A$+B$
```

the data in C$, "HELLO THERE", would be stored in the string storage area at the top of BASIC memory.

String Descriptions

In order for your program to find these strings, it uses a "descriptor" which is located in the variable storage area located immediately following the end of the BASIC program.

The start of this variable storage area is pointed to by the pointer in VARTAB, located at 45-46 ($2D-$2E), and the end of this area by the pointer in ARYTAB, located at 47-48 ($2F-$30).

The string descriptor is 7 bytes long, and contains the variable name in the first two bytes, the string length in the third byte, and the location of the actual string data in the fourth and fifth byte, in normal LObyte/HIbyte order. (Note that the string length is contained in one byte, and this explains why a string can be no longer than 255 bytes.)

To identify the descriptor as a string, the second byte of the variable name has BIT 7 set. If the string variable is B$, then the first byte is 66 ($42) for the B, and the second is 128 ($80), a zero byte with BIT 7 set.

If the string variable were BK$, then the first byte would still be 66 ($42), but the second byte would be 75 ($4B), for the K, PLUS the 128 ($80), BIT 7 set, or 203 ($CB).

Setting the 7th BIT of these two bytes in different combinations designates different kinds of variables. But, for this article, we are concerned only with strings.

Finding the Descriptors

Since these descriptors contain the address of the string data, all we have to do is find the descriptor. Here's how we do that.

Let's say for this example we use BK$ as the string we want. (Well, I *did* write this article!). Search the variable storage area between the locations pointed to in VARTAB pointer, and the ARYTAB pointer, looking for BK$ which will be two consecutive bytes containing the value 66 ($42) and 203 ($CB).

Once we find these, the byte immediately following the 203 contains the string length, and the two after that, the string's address in memory.

Just multiply the second of these by 256, and add it to the first, and we have the address. i.e. if these two bytes contained 33 and 191, the string would be 191*256+33 or 48899, and go upward from there.

Here's BASIC

Okay, now you know the theory behind finding the strings, here's the actual BASIC program that will do it.

```
10   BK$="HELLO ":GOSUB 30
20   BK$=BK$+"THERE":GOSUB 30:END
30   S=PEEK(46)*256+PEEK(45)
40   E=PEEK(48)*256+PEEK(47)
50   FOR I=5 TO E:A=PEEK(I)
60   IF A=66 THEN X=1:GOTO 80
70     IF  X  THEN  IF  A=203  THEN
J=1+1:I=E
80   NEXT:ON-(J=0)GOTO 130:L=PEEK(J)
90   AD=PEEK(J+2)*256+PEEK(J+1)
100 PRINT:PRINT"STRING LENGTH IS"L
110 PRINT"STRING LOCATION IS"AD
120 RETURN
130 PRINT "VARIABLE BK$ NOT FOUND"
```

This routine will return the lengths and addresses of 2 strings. "HELLO " which is in the BASIC text area, and "HELLO THERE", which is in the string storage area in high memory.

Here's ML

Here it is in monitor format. Notice it uses several of the built in BASIC ROM routines.

```
0334    20 FD AE    JSR $AEFD
0337    20 9E AD    JSR $AD9E
033A    20 A3 B6    JSR $B6A3
033D    85 02       STA $02
033F    98          TYA
0340    20 CD BD    JSR $BDCD
0343    A9 2D       LDA #$2D
0345    20 D2 FF    JSR $FFD2
0348    A9 00       LDA #$00
034A    A6 02       LDX $02
034C    20 CD BD    JSR $BDCD
034F    60          RTS
```

Here is the same thing in the form of a BASIC loader.

```
10 FOR D=820 TO D+27
20 READ Y:POKE D,Y:NEXT
30 DATA 32,253,174,32,158,173
40 DATA 32,163,182,133,2,152
50 DATA 32,205,189,169,45
60 DATA 32,210,255,169,0,166,2
70 DATA 32,205,189,96
```

This routine can be used in two ways:

```
BK$="THIS IS A TEST"
SYS 820,BK$

SYS 820,"THIS IS A TEST"
```

The return will be [the] location and length of the string in this format: XXXXX-L where XXXXX is the address of the first character of the string, and L is the length of the string.

It may not ever be necessary for you to find a string's location in normal programming, but should you ever need to, now you know how it is done.

COMPUTE! SAYS AMIGA IS KAPUT!
by Lyndon Soerensen

In a recent surprise move, **COMPUTE!** magazine has dropped coverage of the Amiga. The reasons given for this unexpected development were twofold:

1) A sharp downturn in software sales for the Amiga

2) An unwillingness by manufacturers to develop new software for the machine

What this bodes for the future of the Motorola 68000-based machine can only be speculation. It may well be that this action by a reputed computer magazine may spell the end for the Amiga as an alternative in the 16- and 32-bit computer marketplace.

Another action by **COMPUTE!** was the affirmation of the continuance of the *Gazette* Section dedicated to the Commodore 128 and 64. It appears that the owners of these machines are an extremely loyal group devoted to their 11-year old computers.

To continue to provide the coverage for the 8-bit computers, the editors of **COMPUTE!** are exhorting all subscribers to renew their subscriptions in order that the magazine may provide a service that shows no sign of imitating the fall of the Amiga coverage.

The above events are even more interesting in view of the relative histories of the 8-bit Commodores and the Amiga. Commodore first started marketing the 64 in 1982 as the successor to the tremendously popular VIC-20. With 64K of RAM, plus what was, for its

time, the most advanced sound generator ever in a microcomputer (some have even called it a synthesizer-in-a-chip), the C-64 went on to become even more popular than its predecessor.

The Commodore 128 was a market-smashing alternative to the C-64 when it was introduced in the mid-eighties. Being basically three computers in one (the 128 has a built-in C-64, CP/M, and 128 modes), Commodore decided that it would be the next step up in their computer line.

The acquisition of the Lorraine computer by Commodore in the late-eighties seemed to spell the end for the 64 and 128 as the company's decision to convert to the 68000-based microprocessor seemed to indicate. With superior graphics, stereo sound and much larger internal RAM, the Amiga became the new 'flagship' for Commodore.

SIR RICHARD'S BASIC

by Richard Maze
taken from the Nov. 87 *Monitor*

This article is the start of an ambitious project to examine the various file types that can be used and the programming involved in using each type. In this first article of the series, I will examine the different file types and indicate how these are stored on the diskette. In later articles, I will examine the programming required to use these different files within a computer program. I will use BASIC 2 in all examples. Those of you who have BASIC 4 or BASIC 7 should realize that there are some "shortcuts" available. BASIC 2 will work, however, on all upgrades of BASIC. Hopefully, you will find that it is not that difficult to write programs to access disk files.

There are four different types of files that can be stored on diskettes - PRoGram files, SEQuential data files, RELative data files and USeR (sometimes called 'random access') files.

Let's examine how data is stored on a diskette. A diskette stores data on a number of concentric TRACKS. The 1541 (and 4040) disk drive permits normal access to 35 tracks. Tracks are numbered starting with a 1 on the outside of the diskette and proceeding inward to track 35. Each track is divided into SECTORS (also called BLOCKS). A sector will hold 256 bytes (characters). Because the tracks are of unequal length, there are varying numbers of sectors per track with more sectors on the outer tracks and fewer on the inner tracks. Track 18 is used to keep a record of the diskette contents (directory) and controls access to all files on the diskette.

The directory track has one sector (track 18 sector 0) reserved for the diskette HEADER information. This is the first sector accessed on a read. It contains 2 bytes which tell the track (18) and sector (1) of the first directory sector (block). The next 16 bytes hold the disk name, 2 bytes contain the disk ID, and 2 more contain the disk version and format. 140 bytes are used to hold the BAM (block availability map) which contains information as to what blocks are available on the disk. The remainder are in use.

Each directory block contains the following: the first 2 bytes contain the location of the next directory block; the remaining 254 bytes is subdivided into 8 subdivisions of 30 bytes each with 2 bytes separating each subdivision. Each of these subdivisions contains information about 1 file on the diskette. The information is stored as follows: 1 byte contains the file type; 2 bytes contain a pointer to the track and sector of the first storage location of the file; 16 bytes are reserved for the name of the file; 2 other bytes contain the number of blocks used to store the file. The remainder are either unused or used to control the operation of RELative files.

Remember the 4 basic disk file types?

PRoGram File: - stores a BASIC or assembly language program. This file is normally used to store computer programs. The storage simply involves filling up the 254 bytes of a block (the first 2 are used to link to the next block) and then going on to fill successive blocks until all the program has been saved. A diskette directory will display PRG to indicate a program file. This is marked in a directory on the diskette as hex $82 (decimal 130).

SEQuential Files: In these files, data items are stored one after the other in a long list. Data items are stored

in a manner similar to program files. Because the data may be of different lengths, it must be accessed from the start each time it is used. SEQ in a diskette directory indicates sequential files. They are marked on the diskette by hex $81 (decimal 129).

RELative Data Files: Each record of these is the same length. As a result, data can be accessed individually, a much faster process than reading sequential data. However, there is a greater use of disk storage associated with relative data files. A directory listing will show REL to indicate a relative data file (marked by hex $84 decimal 132).

Random Access Files: These are also called USeR files because they are controlled completely within a program. Individual bytes of data are placed or retrieved directly from a particular sector by program statements. Random access files are the most difficult to set up and use. On a directory listing this file type may be indicated by USR (marked on the directory entry as hex $83 decimal 131). In some cases these files may not have a directory entry at all as information can be put directly on a chosen track and sector. Such a diskette must be used carefully because the information could be destroyed by storing additional files on the diskette.

Next month I will start the examination of each file type and the statements and programming procedures necessary for preparing files using each of the different types.

ANCIENT WISDOM
by Ken Danylczuk
taken from the March 1988 Editorial
(slightly edited)

Been using your computer lately?

I have... to update my disk catalog, write several documents (including this one), update my household inventory, prepare my 1987 tax return (thanks, Earl), prepare a graphic/sound display for a presentation I'll be making next month, play a few games, update my budget (home AND office), organize an equipment inventory for a friend, and prepare AND translate a track and field program for school use! (Puff, puff, puff .... whew!)

So, now that my HUMILITY index has been established, exactly what IS my point?

"Well, Sonny..." Here the wizened old man, stooped and bent over his cane, rubbed his stubbed chin as he reminisced, "I re-call how in them old days, when thar warn't so much spectacler software, us users atchually had to WRIT OUR OWN BY OUR OWNSELF!!"

"Gawsh!" breathed the young wide-eyed innocents gathered at his feet, clutching their 64s and "SKATE OR DIE" game disks to their bosoms. "You mean you actually writ ... uh, wrote programs that made the computer WORK?!!"

The old fellow shifted his gaze down into their innocent, wondering faces. "Yup, an' they even worked, an' if they didn't work like we wanted, WE CHANGED 'EM; MADE 'EM WORK PROPER-LIKE!"

"WOW, no kiddin'?" they murmured in awe as the camera pans away from the scene to focus on the setting sun on the horizon!

Ahem! Sorry, I got carried away. But it's true, there was a time when WE (the users) CONTROLLED AND USED THE COMPUTER. NOW, too often, we're simply CONSUMERS of computer software. I often make comparisons to music, because that's where my heart lies. I know so many people who CONSUME musical products and I KNOW HOW MUCH THEY'RE MISSING BY NOT KNOWING how to create the music themselves. I know that mass marketing is turning everyone it can into computer product CONSUMERS; I'm one myself - and it's a nice and comfortable thing to be - no fuss, muss, or bother as long as you can find a program that DOES what you want, or one which ALMOST does what you want. (And that's NOT as easy as you might believe.)

Now, I'm no fool, and my ma didn't raise no dummy. I don't believe that everyone can be a super-programmer. Nor do I believe that everyone would ENJOY programming, BUT the only way to really make full use of your machine AND your software is to learn a

bit about making the machine YOUR tool following YOUR instructions. Knowing how one goes about making your computer store and sort information will help you select software to suit your needs more effectively.

One more comment on programming. I know that the vast majority of computer users (including our club members) spend most of their time using OTHER people's software, but that's no sin - like I said, I do it myself! But sometimes we spend a lot for software, and invest a lot of time LEARNING to use the software, when a simple, self-made program would do what we want, could change to suit our own changing circumstances, cost little (especially if programmed in BASIC) and take no time to learn! A case in point: the extent of most people's need for a home inventory program is so minimal that writing your own (with the knowledge gained from reading Richard's or other columns in this noble journal) or altering a FREE public domain program to meet your needs can be a doubly rewarding adventure.

So that's my soap-box for this month. I hope you'll consider the beast you sit before so much each month. Learn about it and its inner workings! You'll only learn to love it more, 'specially if it's a 64! (Hey, there's a poet inside me!)

See you USERS next month!

LONG DISTANCE VOYAGER
The Trials and tribulations of BBSing Long-Distance
By Lyndon Soerensen

Lately, I have been calling bulletin boards that aren't around town. These boards are either in another city in the province, or are entirely out of the province and even out of the country altogether.

While most of the BBSs I call are Commodore-supported, there are also a few I contact that are non-Commodore. I restrict myself to calling only a few a night, since contacting a board and going through all new messages may take several precious minutes. There are many advantages and some disadvantages to calling boards long-distance.

The advantages lie mainly in meeting new people that otherwise you would never get to talk to. Most people you meet will always have something interesting to say, even though you are separated by, in some cases, several thousand miles. The power of personal communication is not lost and may even be enhanced when talking long-distance. Another advantage is the acquisition of programs ("YEAH!" I can hear some people say. "He's getting to something *important* finally!"). There are a *huge* number of new files on some boards, as most will be using a hard drive to operate the BBS. It's possible on some calls to entirely fill up an entire disk with new downloads, and start on the other side before you've gotten through even the first file listings! The number of new files you can get is enormous. With boards that use hard-drives of 10, 20, and even 40 megabytes, it's clear that you'll soon be wallowing in new software. And of course, there are always on-line games to play, from Barren Realms Elite to Poker, there are games to satisfy even the most demanding game player.

The biggest (and probably the worst) disadvantage is the cost of a long distance call. Depending on the distance from your city to the other, the cost could vary from $3 a call up to $10-$12 a call. Of course, the most obvious way to reduce phone charges is to call after midnight. When you call after midnight, you make a substantial saving on your bill and what might be a $15 call in daytime might only be a $5 call at 2 AM. Another way to reduce the amount of charges on your phone bill is to buffer-capture text. Most terminals will support the collection of text into a buffer, which is basically just an area of memory set aside in your computer to store what is onscreen.

Depending on the size of your buffer, it's possible that you could save an entire massage area filled with new messages. When you have what you want saved, you can then log off the BBS, look at what captures your eye, and discard the rest. A real time-saver. Obviously, the higher baud rate of the modem you use, the faster you will be able to store text and download

files. Probably the minimum baud rate that would be useful would be 2400 baud. Of course, anything faster is a bonus, but the minimum rate should be at least 2400 baud in order to strike a balance between on-line time and phone charges.

There is another problem when calling long distance: some BBSs have several logon screens that you must page through in order just to get to the main menus. It's possible to waste 3 or 4 minutes of time just to do one simple thing (of course, most of these screens are there for the local users, but it's still an annoyance when you have to go through 4 pages of community notes in order to get the new messages if the BLAB-IT message base). Sometimes it's possible to abort these screens with a key-press or two, but there are occasions when all you can do is just suffer through them, and thus flatten your wallet.

Despite the problems, calling BBSs long distance can be a very enjoyable experience, well worth the cost.

For the adventurous, some BBS numbers to try:

(all are C= support)

S.C.U.G. BBS   Brighton BBS
Saskatoon, SK   Belleville,ON   1-374-0703
      1-613-475-4668

Port Commodore BBS
Spanish Fork, UT
1-801-423-2743

PRESEDENTIAL PARLEZ

Hello Commodore 8 bitters. Welcome to the CUGS *Monitor*. What you see before you is the first initial newsletter with Tristan Miller in the editor's chair. After looking at a draft copy that Tristan showed us at the executive meeting on September 13, I was very impressed with his work. Very good for first try. I understand that Byron Purse was also largely responsible for the newsletter efforts as well. Thanks, guys, and good work.

This meeting we decided to do some interesting things with SIDS. Tristan has volunteered to show us what can be done with them and how to edit and change them to suit you, the user.

The remaining CUGS general meeting dates till the end of 1993 are October 6, November 3 and December 1. The next Executive meeting is tentatively scheduled for October 18.

The October meeting is just one month away from the CUGS annual election night. November 3 is the day you want to keep in mind and be sure to attend. The executive people are the ones who keep the CUGS club flowing and moving along. We have several people who have volunteered to fill in some of the available positions. Some are moving up and some are maintaining their positions. I, for one, will be stepping down as president of CUGS after four years of service. Tristan Miller has indicated he is willing to fill that position as well as looking after the CUGS BBS. Ross Parker, the secretary/treasurer, has indicated he is stepping down as well. Any and all positions should be filled in order that the CUGS activities will continue smoothly to help the Commodore 8-bit computers to survive and be fun. The executive positions that are available are:

President
Vice President
Secretary/Treasurer
Editor
Assistant Editor
64 Librarian
128 Librarian
Members at Large

If you wish to fill any of these positions, please call me before the meeting or before the November meeting starts. These positions do not require you to be a expert in any given field, just the desire to help, to learn, to teach, to help these 8 bit machines.

Till next time, have a good one!

System Name		Number	BPS	Protocol	SW	CPU	Nets	Flags
Adult Superboard		789-8682	14.4	3b,4b	PB	P	F	1$
Atmospheric Environ.		780-6049	9600	3,M	WC	P		$
Beach House		729-4185	2400	None	Vi	P	FV	Q
Crystal Visions		586-6790	2400	None	RA	P	F	
C.U.G.S.		543-7683	2400	None	EB	C		
DataForce		585-1958	14.4	H,4b	RA	P	F	Q1
DLC-West <Hi Speed>		352-9390	16.8	H,3b,4b	--	P	F	$
DLC-West		352-9378	2400	None	--	P	F	L$
Double Check		525-0807	16.8	H,3b,4b	SN	A	F	Q1
Extreme Outer Limits		545-0417	19.2	Z,3b,4b	WC	P	FI	Q
FACT		924-8776	2400	None	WC	P	F	LQ$
Fernando's Retreat		585-0298	9600	H	Op	P	F	
Fourth Floor		352-0472	14.4	3b,4b	Sy	P	F	Q
Grand Ole Opry		924-2983	2400	None	RA	P		
Green Zone		789-7652	14.4	H,3b,4b	Mx	O	F	Q1
Holistic		789-9909	14.4	3b,4b	TB	P	F	Q
IEEE South Sask.		586-1939	9600	3	RA	P	F	B
Impossible Missions		569-9705	2400	None	*Su*	P	FW	
Intimate Encounters 1	*	789-3854	2400	None	RA	P		
Intimate Encounters 2	*	789-3856	2400	None	RA	P		
Micro City <Hi Speed>		791-3388	14.4	3b,4b	Ma	P	JR	$
Micro City		757-0088	2400	None	Ma	P	JR	$
MEBBS II		775-1437	14.4	3b,4b	TA	A	F	Q
Missing Link I		775-1511	14.4	H,3,4b	PB	P		
Missing Link II		775-1512	14.4	H,3,4b	PB	P		
No Quarter		584-7428	14.4	3b,4b	TR	P	F	Q
Pool Hall I		586-8490	9600	H,4b	PB	P		1
Pool Hall II		586-0922	14.4	H,4b	PB	P		1
Q-COMM Data Services		584-2916	2400	None	LN	P		
Reflex Point		586-9571	9600	3	WC	P	AF	Q
Regina_FIDO		777-4493	9600	3,4b	TB	P	F	LQ
Ronchy's Pleasure Dome		949-8486	16.8	H,3b,4b	*RA*	P	BF	B1$
Sage's Desk		545-2943	14.4	3b,4b	DL	A	F	Q
Scout's Own		777-2998	9600	3	RA	P	F	B
Shadowland		789-1899	14.4	3b,4b	RA	P	F	
Shareware Superboard		789-8690	14.4	3b,4b	PB	P	F	1
Skywatch		569-0581	14.4	3b,4b	TR	P	F	Q
Snake Pit		924-0773	14.4	3b,4b	Su	P	FIR	B
Tower of High Sorcery		545-0801	14.4	3b,4b	Vi	P	FI	
TTL Computer Concepts		522-3233	16.8	H,3b,4b	RA	P		B
Unibase		789-0709	9600	3,4b	--	U	U	$
USS Galifrey		949-6032	14.4	H,3b,4b	RA	P	FT	BQ1
VorteX		584-7062	2400	None	RA	P		1
Wedge Net BBS		352-3434	14.4	3b,4b	*WC*	P	FDW	Q1$

Total Systems: 40
Systems with a * before phone number are NEW entries

CPU: A=AMIGA C=C128 O=OS/2 P=MS-DOS U=UNIX   NETWORKS: A=ADULTNET F=FIDONET R=RIME
V=VIRTUALNET B=ROBONET I=INTERWAR-NET T=TREKNET W=WORLDNET D=ADANET J=MAJORNET U=USENET
FLAGS: B=BLUEWAVE Q=QWK L=ROTARY SWITCH LINES 1=1200 BPS MIN 2=2400 BPS MIN 7=7,E,1 MODEM
SETTINGS 9=9600 BPS MIN UPDATES SENT TO ALLAN FINKAS ON GALIFREY