

C.C.O.G.

COLOR COMPUTER OWNERS GROUP

NEWSLETTER JULY 1991

OFFICERS:

PRESIDENT Bernie Patton
VICE-PRESIDENT. Chuck Goodwin
SECRETARY Marcine Glowicki
TREASURER Bob Waite
CORRESPONDING SECRETARY . . Robert Gault
LIBRARIAN Bob Humphrey
BULLETIN BOARD. (313) 292-4713

INDEX:

1. Pres. Comments
2. Meeting Minutes
3. OS-9
8. Viewpoints
10. Beginners Corner
12. Experts Corner

PRESIDENTS COMMENTS:

Wow, do you believe 20 pages of articles. Robert and Tom got a little carried away this month. But, I think its great!!! It may be a little too technical for some but we try to have a little for everyone. So from time to time we should be allowed to go a little overboard one way or another.

One of the advantages of writing this column and editing the newsletter is that I get to read all of the other columns before I write this one. That does help me sometimes with a thought or two when writing this column.

I don't know what article Gus read about the Apple/IBM joint venture. I have seen a lot of information and a lot of mis-information since this venture was announced. I will give you my version of what I have been able to get out of what I have read. The machine will be built around the IBM RISC-6000 chip. RISC = Reduced Instruction Set Chip. RISC chips are used in Workstations and MINI-computers. They are

very fast and very expensive. At least up to now they have been expensive. The machine will have a GUI, Graphical User Interface, built on top of IBM's operating system. The machine will be aimed at high-end CAD users, and at commercial and industrial Multi-media markets. The machine will be rather expensive when first released.

This venture does two things for IBM: 1. A new market for their RISC-6000 chip. And 2. It get Apple off their backs about GUI's (WINDOWS).

This venture does two things for Apple: 1. It gives them a machine to compete with NeXT, Appolo, and Sparc. And 2. It gives them an upward path for Macintosh users. Rumor has it that the supply of CPU's for The Mac is in jeopardy because of a lawsuit filed against Motorola by a Japanese firm.

It will be a long time (2 years after release) before a machine of this type will have any impact on the computer hobbyist. Bernie

COLOR COMPUTER OWNERS GROUP

This newsletter is a periodic publication of the
COLOR COMPUTER OWNERS GROUP
of Metropolitan Detroit.

Copyright for all articles in this publication remains with the original author. Permission is given to reprint for nonprofit publications so long as original article is intact with credit to the Author. Requests can be made for downloading articles thru our BBS from other CoCo Clubs wishing to reprint articles. Contact: Bernie Patton (313) 283-2474 BBS (313) 292-4713

C.C.O.G. Minutes from June 25, 1991 by Marcine Glowicki

The meeting was called to order at 7:18. 18 people were in attendance. A motion was made to accept the minutes from last month's meeting. Motion passed.

OLD BUSINESS--An invitation was made to accept advertising from all advertisers who are still in business and promoting the COCO machine. We will reserve the right to shrink the size of the advertisement submitted but we will not edit the advertisement.

While people were airing their opinions tonight, someone said we should contact Kentucky (RAINBOW MAGAZINE), and complain about the size of the magazine and the lack of material on the disk. Bernie stepped in on this one and referred to his article in the June newsletter.

Frank Hogg was approached by Tandy to purchase ROMS--no word yet.

NEW BUSINESS--Robert Gault informed members that the old gimmi chip can cause computer problems of various sorts. If the chip has a 1986 date this is an older version of the chip. The 1987/88 dated chip

is the improved version. Bernie received a letter from Tenn. asking if the club had any information to share with regards to the new MMI computer. Specifically they wanted comments from anyone who had purchased one of the machines. This information will be compiled into a book that the fellow is compiling on the MMI.

Demonstrations for the evening went into a presentation of RS Basic by Bernie, Bob Gault on OS9 basic and Tom with C. It was interesting that they each worked with the same type program but showed how it could be done using each of the types of programing.

Next month's theme will be discussed on the BBS so voice your opinion and Bernie will take count. Actually this might be a way to try and get more people from the club to use the BBS.

\$17 was collected for the VFW.

See you next month, July 23, 1991.

OS-9 by Tom Napolitano

This article contains minimal answers to the comp.lang.c frequently-asked questions list.

Null Pointers

Q. What is this infamous null pointer, anyway?

A: For each pointer type, there is a special value -- the "null pointer" -- which is distinguishable from all other pointer values and which is not the address of any object.

Q. How do I "get" a null pointer in my programs?

A: A constant 0 in a pointer context is converted into a null pointer at compile time. A "pointer context" is an initialization, assignment, or comparison with one side a variable or expression of pointer type, and (in ANSI standard C) a function argument which has a prototype in scope declaring a certain parameter as being of pointer type. In other contexts (function arguments without prototypes, or in the variable part of variadic function calls) a constant 0 with an appropriate explicit cast is required.

Q. How should NULL be #defined on a machine which uses a nonzero bit pattern as the internal representation of a null pointer?

A: The same as any other machine: as 0 (or (void *)0). (The compiler makes the translation, upon seeing a 0, not the preprocessor.)

Q. Is the abbreviated pointer comparison "if(p)" to test for non-null pointers valid? What if the internal representation for null pointers is nonzero?

A: The construction "if(p)" works, regardless of the internal representation of null pointers, because the compiler essentially rewrites it as "if(p != 0)" and goes on to convert 0 into the correct null pointer.

Q. If "NULL" and "0" are equivalent, which should I use?

A: Either; the distinction is entirely stylistic.

Q. But wouldn't it be better to use NULL (rather than 0) in case the value of NULL changes, perhaps on a machine with nonzero null pointers?

A: No. NULL is, and will always be, 0.

Q. Why is there so much confusion surrounding null pointers? Why do these questions come up so often?

A: The fact that null pointers are represented both in source code, and internally to most machines, as zero invites unwarranted assumptions. The use of a preprocessor macro (NULL) suggests that the value might change later, or on some weird machine.

Q. I'm still confused. I just can't understand all this null pointer stuff.

A: A simple rule is, "Always use '0' or 'NULL' for null pointers, and always cast them when they are used as arguments in function calls."

Arrays and Pointers

Q. I had the definition `char x[6]` in one source file, and in another I declared `extern char *x`. Why didn't it work?

A: The declaration `extern char *x` simply does not match the actual definition. Use `extern char x[]`.

Q. But I heard that `char x[]` was identical to `char *x`.

A: Not at all. Arrays are not pointers.

Q. Why are array and pointer declarations interchangeable as function formal parameters?

A: Since functions can never receive arrays as parameters, any parameter declarations which "look like" arrays are treated by the compiler as if they were pointers.

Q. Someone explained to me that arrays were really just constant pointers.

A: An array name is "constant" in that it cannot be assigned to, but an array is not a pointer.

Q. I came across some "joke" code containing the "expression" `5["abcdef"]`. How can this be legal C?

A: Yes, array subscripting is commutative in C. The array subscripting operation `a[e]` is defined as being equivalent to `*((a)+(e))`.

Q. How do I declare a pointer to an array?

A: Usually, you don't want to. Consider using a pointer to one of the array's elements instead.

Q. How can I dynamically allocate a multidimensional array?

A: It is usually best to allocate an array of pointers, and then initialize each pointer to a dynamically-allocated "row." See the full list for code samples.

Order of Evaluation

Q. Under my compiler, the code `"int i = 7; printf("%d\n", i++ * i++);"` prints 49. Regardless of the order of evaluation, shouldn't it print 56?

A: The operations implied by the postincrement and postdecrement operators `++` and `--` are performed at some time after the operand's former values are yielded and before the end of the expression, but not necessarily immediately after, or before other parts of the expression are evaluated.

Q. But what about the `&&`, `||`, and comma operators?

A: There is a special exception for those operators, (as well as `?:`); left-to-right evaluation is guaranteed.

Memory Allocation

Q. Why doesn't the code `"char *answer; gets(answer);"` work?

A: The pointer variable "answer" has not been set to point to any valid storage. The simplest way to correct this fragment is to use a local

array, instead of a pointer.

Q. I can't get strcat to work. I tried "char *s1 = "Hello, ", *s2 = "world!", *s3 = strcat(s1, s2);" but I got strange results.

A: Again, the problem is that space for the concatenated result is not properly allocated.

Q. But the manual page for strcat says that it takes two char *'s as arguments. How am I supposed to know to allocate things?

A: In general, when using pointers you always have to consider memory allocation, at least to make sure that the compiler is doing it for you.

Q. Can you use dynamically-allocated memory after you free it?

A: No.

Q. How does free() know how many bytes to free?

A: The malloc/free package remembers the size of each block it allocates and returns.

Q. Is it safe to use calloc's zero-fill guarantee for pointer and floating-point values?

A: calloc(m, n) is essentially equivalent to "p = malloc(m * n); memset(p, 0, m * n);". The zero fill is all-bits-zero, and does not therefore guarantee useful zero values for pointers or floating-point values.

Structures

Q. I heard that structures could be assigned to variables and passed to and from functions, but K&R I says not.

A: These operations are supported by all modern compilers. But unfortunately, not the coco os9 c compiler.

Q. Why can't you compare structs?

A: There is no reasonable way for a compiler to implement struct comparison which is consistent with C's low-level flavor.

Declarations

Q. How do you decide which integer type to use?

A: If you might need large values, use long. If space is very important, use short. Otherwise, use int.

Q. How do I declare an array of pointers to functions returning pointers to functions returning pointers to characters?

A: char *(*(*a[5]))();

Q. How do I initialize a pointer to a function?

A: Use something like "extern int func(); int (*fp)() = func; " .

Boolean Expressions and Variables

Q. What is the right type to use for boolean values in C?

A: C does not provide a standard boolean type, because picking one

involves a space/time tradeoff which is best decided by the programmer.

Q. Isn't #defining TRUE to be 1 dangerous, since any nonzero value is considered "true" in C? What if a built-in boolean or relational operator "returns" something other than 1?

A: It is true (sic) that any nonzero value is considered true in C, but this applies only "on input", i.e. where a boolean value is expected. When a boolean value is generated by a built-in operator, it is guaranteed to be 1 or 0. (This is `_not_` true for some library routines such as `isalpha`.)

Operating System Dependencies

Q. How can I read a single character from the keyboard without waiting for a newline?

A: Contrary to popular belief and many people's wishes, this is not a C-related question. How to do so is a function of the operating system in use.

Stdio

Q. My program's prompts and intermediate output don't always show up on the screen, especially when I pipe the output through another program.

A: It is best to use an explicit `fflush(stdout)` whenever output should definitely be visible.

Q. When I read from the keyboard with `scanf()`, it seems to hang until I type one extra line of input.

A: `scanf()` was designed for free-format input, which is seldom what you want when reading from the keyboard.

Q. How can I recover the file name given an open file descriptor?

A: This problem is, in general, insoluble. It is best to remember the names of open files yourself.

Miscellaneous

Q. What can I safely assume about the initial values of variables which are not explicitly initialized?

A: Variables with "static" duration start out as 0, as if the programmer had initialized them. Variables with "automatic" duration, and dynamically-allocated memory, start out containing garbage (with the exception of `calloc`).

Q. Can someone tell me how to write `itoa`?

A: Just use `sprintf`.

Q. How can I write data files which can be read on other machines with different data formats?

A: The best solution is to use text files.

Q. I seem to be missing the system header file `<sgtty.h>`. Can someone send me a copy?

A: You cannot just pick up a copy of someone else's header file and

expect it to work, since the definitions within header files are frequently system-dependent.

Q. How can I call Fortran (BASIC, Pascal, ADA, lisp) functions from C?
A: The answer is entirely dependent on the machine and the specific calling sequences of the various compilers in use.

Q. How can I make this code more efficient?
A: Efficiency is not important nearly as often as people tend to think it is. Most of the time, by simply paying attention to good algorithm choices, perfectly acceptable results can be achieved.

Q. Are pointers really faster than arrays? Do function calls really slow things down?
A: Precise answers to these and many similar questions depend of course on the processor and compiler in use.

Q. I'm having trouble with a Turbo C program which crashes and says something like "floating point not loaded."
A: Some compilers for small machines, including Turbo C, attempt to leave out floating point support if it looks like it will not be needed. The programmer must occasionally insert a dummy explicit floating-point call to force loading of floating-point support.

Q. This program crashes before it even runs!
A: Look for very large, local arrays.

Q. How do you pronounce "char"?
A: Like the English words "char," "care," or "car" (your choice).

Abbrided from a text file by:

Steve Summit

Quality OS/9 Level 2 Software
from
ColorSystems Games!

Specialty Programs	Games!
WFSHcl	Variations of Solitaire
An OS/9 Level 2 Word Processing Shell.	Include five separate game programs, five variations of Solitaire.
Features:	Includes the following Variations:
<i>Pull-Down Menus</i>	Pyramid
<i>User Customizable</i>	Klondike
<i>Works with most any Editor and Text Formatter (Not Included)</i>	Spider
<i>Requires Multi-File</i>	Poker
Complete with documentation for only \$22.00	Canfield
	Complete package for only \$34.95 <i>Multi-File is NOT required!</i>
MYBanner	Game Pack Special!
A Banner Generation Program for the Multi-View Environment	Contains ALL of the following:
Features:	CoCobello
<i>Pull-Down Menus</i>	CoCoYahzee
<i>On-Screen Banner Previewing</i>	KnightsBridge
<i>Image Library and Editor</i>	Minifield
<i>Requires Multi-File and an 80 Column Monitor</i>	Sea Battle
Only \$20.00	All five games for only \$34.95 <i>Multi-File is NOT required!</i> <i>CoCoYahzee requires an 80 Column Monitor</i>

To Order Send Your Check or Money Order to:
ColorSystems
P.O. Box 540
Castle Hayne, NC
(919) 675-1706 (Voice)
(919) 675-1847 (BBS)

*No Shipping Charges to the Continental US
Add \$3 for Canada, Alaska, Hawaii and Mexico
Add \$5 for other Oversea Areas*

North Carolina Residents Please Include 5% Sales Tax.
For Complete Product Descriptions Send For Our FREE Catalog!

"Viewpoints" by Gus Korte"

Recently I completed a program in RSDOS that enhances SCRIPSIT word processing software so that I can use it do bold type printing. Between the helpful suggestions received from CCOG club members on the club electronic bulletin board system (BBS) and what I could extract from my printer manuals, I was able to eventually run the program successfully. I am only an amateur at programming. However the program works and if I can do this so can you other hobbyists who are "green" at programming. Try writing some programs to accomplish processing data for your own purposes. I think you'll enjoy it. As you no doubt know, MAX10 software can already do bold and other fancy printing but the printing is at a much slower rate. So I use SCRIPSIT for my rapid printing jobs and include the bold print option for the more important output. However for the really fancy printing, usually for master copies from which to duplicate more copies, I use MAX10. If any of you who own SCRIPSIT are interested in bold printing your results, let me know and I can make my program available to you.

One of the good reasons to contact the forum messages on the electronic bulletin board system (BBS) the CCOG club supports, is to keep yourself current on the latest information about our hobby. Recently I noticed a message on the BBS about a FD502 disk drive on sale at a nearby Radio Shack store (located near 5 Mile and Merriman Roads in Livonia) for a very low price. I quickly went over there and

bought that bargain before it was gone. Now I have to determine how to hook it to my COCO3 through a multipak or other type of system when one becomes available. This again indicates that Radio Shack has some good sale bargains available at this time for all you COCO users. This Radio Shack store, mentioned previously, also has a lot of COCO software and manuals at sale prices so you may be interested in seeing what they have to enhance your hobby. Having a multiple disk drive system is a necessity to get full use of the BASIC09 software and related products, as club member Bob Gault has indicated previously.

I have learned a lot about BASIC09 from reading the manual on the subject which I also obtained at a sale bargain price at Radio Shack. As I get into more of this subject, I can see how much better this type of BASIC is than RSDOS BASIC. That's why many club members feel that OS9 is the main future for all COCO users. So if you can afford to go this route do so since it will make your COCO hobby more enjoyable.

Have you noticed the newspaper report that the Apple and IBM computer companies have decided to cooperate in making simpler-to-use computers? They also want to standardize them too as a cooperative effort. I assume this will also include the computer software languages. I wonder how this will relate to the so-called COCO4 type of computers now developing and even the OS9 system. Does this mean that all future computers will become IBM-Apple compatible in