

EDMONTON
99'er
COMPUTER
USERS'
SOCIETY

The Edmonton 99'er Computer Users' Society
p.o. box 11983
Edmonton, Alberta
CANADA T5J 3L1

99'er ON LINE... is the news letter of the Edmonton 99'er Computer User's Society published ten times a year. Unless otherwise stated, all articles may be republished in other news letters provided that source and author are identified. We will credit authors quoted in 99'er ON LINE.

CORRESPONDENCE: Newsletter editor: Bob Pass, 59 Labelle Cr, St. Albert, Alberta, Canada T8N-2G6. (403)458-7658. All other correspondence should be sent to the address at left.

OFFICERS: president Tom Hall, vice-pres Ken Godbeer, treasurer Jim Mulligan, secretary Roxanne Appelt

DISCLAIMER: Information published in this newsletter is created by and for amateurs; therefore, we cannot guarantee the accuracy or use of presented information.

REGULAR MEETINGS... of the Edmonton 99'er Computer User's Society are held on the second Tuesday of each month in room 849 of the General Services building of the University of Alberta from 7:00 till 10:00 PM and are open to all members in good standing. Non-members may attend their first meeting free of charge.

ADVERTIZING... Commercial space is available in this news letter at the following rates: full page \$20.00, half page \$15.00, 1/4 page \$10.00. Discuss your needs with Jim Mulligan at 467-6021, at the next meeting, or send "photo ready" copy to the P/O Box above. Members may advertise their personal computer related items for free but are asked to limit their ads to about 50 words. Mail your ads to the editor's address or hand it to him at the general meeting; newsletter deadline 3'rd Monday of the month.

MEMBERSHIP FEES: Family 12 months, \$20.00, 6 months, \$15.00. Students 12 months, \$15.00, 6 months, \$10.00. New member initiation, \$20.00.

APRIL MEETING

Despite the Oilers playoff game being on TV, the March meeting was well attended by more than 30 members; not all of us are hockey-holics! Tom Hall managed to borrow a Plato Cartridge which he demonstrated. Apparently this software was very popular in the States before TI folded the 99/4A but this was the first time anyone in our group has seen Plato and the associated computer aided instruction courseware.

Nomination forms for executive positions were received. Due to a mix up during the mailing of last month's newsletter, some of you did not receive nomination forms. If you wish to nominate a person to run for election to the executive, please contact a member of the current executive (see front page for phone numbers) before May 8'th and we will take your nomination by phone. Sorry for the confusion folks.

NEXT MEETING

The next meeting will be Tuesday, May 12'th at 7:15 PM. Same place as usual; General Services Building, U of A campus in room 849.

A reminder to all that your attendance at the June meeting is vital as it is then that we hold elections for the four executive positions of this users' group and vote on any other important business. In this same vein, the nominees will be announced at the May meeting and encouraged to do a little electioneering.



FIRST STEPS INTO c99
DEPT OF DIRTY TRICKS

by Michal Jaegermann

In this article I will show you some things in c99, which you are not supposed to do. Why bother? As I repeated many times (you are probably deeply bored with that) c99 belongs to "small-C" family and, therefore, lacks many features of a full language. In your programming project you may find out that some ways of doing things are not available. Then you have three choices - find detour (often the best way out), abandon the program, or cheat a little bit. The last way out is supposed to be an "advanced" programming, but a limited scope of c99 may force you on that road sooner than you expect. And there are also things like compiler bugs. Compiler are programs and they have bugs too, you know. That is a rationale behind including that into our introductory series. You will find below a short list of most common situations. Admittedly, a border between them, and also between legitimate and "tricky" programming can be blurry, but some kind of classification is handy.

A GOTO DIGRESSION. You may encounter an opinion that a use of "goto" in programming is cheating. Quite often it is, and of the worst possible kind. On the other hand, I do not consider a use of "goto" as a cardinal sin. It happens, although very infrequent, that "goto" is the best and most logical thing to do. But indiscriminate jumps around in a fit of "smarties", which replace a legitimate constructs with a jungle of a spaghetti code, are something truly awful. I do not want to go into a lengthy discussion, since c99, unlike the older brother C, simply does not have "goto" at all. So, like it or not, from that sin you are saved.

Missing types

As you know, c99 deals only with objects of two types, integers and characters, and with pointers to them. Nothing more. So how you can declare a variable which holds a pointer to an address where a pointer to a character is stored? You cannot! Clear and simple. But if you must? Say translating into c99 some C source, where you found declaration

```
char **ppc;
```

You may notice, that in c99 in integer takes two bytes of memory, which happens also to be an amount of storage needed by an address. So you write instead

```
int ppc; /* this was meant to be "char **ppc;" */
```

and hope for the best. Such substitution is highly misleading, machine dependent, fattening and a departure from any reasonable standards of a good coding. You have a little choice with c99, but put an explanatory comment and do not get into a habit! A C compiler on other machine may not accept later assignment of a pointer to an integer variable, or, much much worse, it may accept, but integers and addresses on the other machine may happen to be of a different width.

The other example of a type cheating you may find in my code from the last month. In order to force an unsigned comparison of two numbers I compared two pointers to characters with the same bit pattern. Why pointers to characters instead of pointers to integers? Because integers are stored only at even addresses, so bit patterns which translate to odd numbers are out. C compilers use so called "casts" and sets of casting rules, which tell which types can be converted to what and how, so you may write

```
if ((unsigned int)here < (unsigned int)there)
```

but we, poor souls, have to get away with a clear mind and a clear comment.

One more example you may find in a floating point library distributed with other c99 stuff. A float there is defined as an array of four characters, which in C is equivalent to a pointer to character. This hack has nasty side effects when passing floats to functions. It may spoil your whole debugging session. I do not want to go into lengthy discussions, but be forewarned, that book examples involving floating point may not work with floating point library of c99.

Missing hardware access

This happens on every computer. Your hardware provides some functions which are not supported by a compiler. There is only one recourse. Write missing parts in assembler (or use other suitable compiler, if available) and link with a rest of your program. In c99 you have three choices.

You may write an assembler code, use DEF directive to create an entry point, assemble your file and link it with the rest of your program when loading DIS/FIX 80 files. Some of libraries are created this way and this is a good choice

when you have a set of functions which you will be using over and over again.

Next, you may put together an assembler code produced by the compiler with fragments of code which you created in separate files yourself. Our assembler has a special COPY directive to make it easy.

The third method puts directly pieces of code into a text of c99 program. c99 has a special pair of directives #asm...#endasm just for this purpose. Remember that "hash sign" has to be always in the first column. Everything in between will be just copied directly into compiler output. Do not expect the either sanity or correctness will be checked.

Does not matter which method you are using, you have to know which registers are available and what are conventions used to pass function arguments. All of these is described in your documentation in sections VII. MEMORY UTILIZATION and VIII. STACK USAGE. Check your manual. In short, on function call R14 is stack pointer and points to a location which holds a return address. Below are arguments pushed on stack from left to right (other C compilers often do that in a reverse order). Stack grows towards low memory, so higher on stack means at a lower address.

If, after you carefully read the documentation, you still feel hopelessly lost, then compile the following function, keeping c99 text in an assembler output, and have a look at the resulting code.

```
test (x, y, z)
int x, y;
char z;
{
    x = 1;
    y = 2;
    z = 'a';
    return (x+y);
}
```

Here is also an example of a function which reads a piece of VDP memory into a buffer in CPU memory.

```
#asm
REF VMBR
#endasm
/* the above has to be at the top of a file */

readv (from, to, n) /* return number of bytes actually transferred */
char *from; /* VDP address to read from */
char *to; /* location of a buffer to store read bytes */
int n; /* how many bytes to read */
{
    if (n <= 0) return (0); /* nothing to read - bail out */
    #asm
    MOV@6(R14),R0 VDP address to R0
    MOV@4(R14),R1 buffer address to R1
    MOV@2(R14),R2 how many bytes
    BLW@VMBR read VDP
    #endasm
    return (n); /* tell that you read n bytes */
}
```

Hope that you will find a good use for it.

Missing feature

You may find yourself in a situation where you would love to have some construct not provided by a compiler. It may be possible to fake that, quite cheaply, with a little help of assembler. A typical situation for c99 will be an array of strings. I will leave for you checking why this is troublesome for c99. But look at the program below - a little bit funny example, but check carefully how C was used. Reprogramming it in BASIC can be instructive.

Editor's note: to save space, the program is listed elsewhere in this issue.

If you will look closer then you will find out that I have not only array of strings, but it was initialized statically. This means that data were not created by a program, but they were ready in place when the program started its execution, loaded from a file. Notice that every string is terminated with a null byte - in a manner which C inspects it. My array 'strings []' holds pointers to strings of characters, therefore it should be really declared as 'char *strings [SIZE];', or equivalently as 'char **strings;'. I cannot do that in c99, so I resigned myself to an array of integers. A size of it is immaterial, since I am going to initialize it in assembler anyway and C does not check array bounds. This declaration is really necessary only for other statements, which have to generate a proper assembler code whenever 'strings' are referenced. Here comes the worst of all tricks. This declaration has to be manually removed from an assembled code in order to be replaced by the "right stuff". Below is a fragment of an

actual compiler output.

```
0005 #int strings [1];
0006 STRING BSS 2
0007 #asm
0008 STRING DATA S0,S1,S2,S3,S4,S5
0009 DATA S6,S7,S8,S9,S10
0010 S0 TEXT 'None '
0011 BYTE 0
0012 S1 TEXT 'One '
0013 BYTE 0
```

Note that I inserted a label STRING (not STRINGS, since c99 truncates all names to six characters) into assembler code. The same label was produced by a compiler in line 0006. This is my defensive programming. In case I would forget to edit a compiler output, then assembler will loudly complain and I would know that something is incorrect. Once you will delete now harmful line 0006 everything will be fine and your program will work as expected.

Conclusion

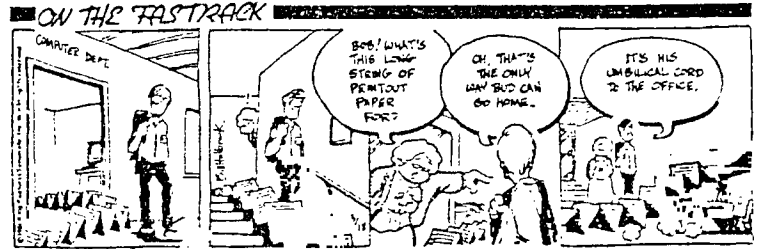
This was the last article in this series. I think that I bored you long enough and also I hope that somebody else has something fascinating to say. Remember, that this was not a replacement for C textbook or c99 documentation. There is no way one can learn a new language in four easy installments. But if you glimpsed a little bit of new understanding or ideas or had some fun, then my goals has been achieved. If you are just starting, then maybe you should try some textbook and some programming for a month or two, and re-read, after that, these articles? Just suggestion... But if you have questions, or you are stuck with your program, then ask me question, write letter to editor, or leave me a message on BBS. If an interest will be high enough, then who knows. Some answers may even appear in this newsletter.

DAFFY DICT: '42-87

The following originally appeared in the August '85 issue of the HOCU newsletter (from the Milwaukee Area User Group) where it was spotted by David L. Peden of El Cajon California who forwarded it to the Tacoma 99'ers Users' Group newsletter where we spotted it. Vive la grapevine! Your editor couldn't resist adding a few of his own too.

```
AMPS.....Little creepy crawlies
ARC.....Welding invented by Noah
BUS BAR.....Mobile cocktail lounge
CAPACITOR.....One who can hold his liquor
COAXIAL.....2 engineers fired simultaneously
COMMUTATOR.....One who drives to the city each day
CONDENSOR.....Writer for Reader's Digest
DEGAUSS.....To remove a bandage
DETECTOR.....Private eye
DETENT.....A small outdoor canvas shelter
DIODE.....Eulogy in poetic form
ELECTRODE.....Automated highway
FARAD.....Deposed Egyptian ruler
FOURIER.....Superseded by fiveier
GAMMA.....Short for Grandmother
GAUSS.....Singular for geese
.....NIUM.....Solid state pink flower
.....Medical term for pain; coincident with DEGAUSS
IMPEDANCE.....Brat desperately needing bathroom break
INFRA RED.....American spy in Moscow
IGN.....Device to remove wrinkles
JOULE.....Gift from rich uncle
KILOVOLT.....Only volt on the ten most wanted list
LOAD LINES.....Monday morning wash
LOGARITHM.....Dancing on logs
MICROFARAD.....Son of deposed Egyptian ruler
MHC.....Asking for seconds in Alabama
NEGATIVE CHARGE.....Poor credit risk
OHM.....House in England
OUTLET.....Going away party for engineers
```

OVERHEAD.....The boss
 PENTODE.....Frog in captivity
 POWER FACTOR.....Mother-in-law
 QUARTZ.....Unit of measure for booze
 QUALITY CONTROL.....Excuse to return extra parts
 RECEPTACLE.....Round filing cabinet
 REFINANCE.....Historical age of artistic rebirth
 RADIANT DIAGRAM.....Man used by skiers
 SERIES RESISTOR.....TV censor
 SILICON.....Wanna buy a bridge?
 TAIL FORMER.....A good beautician
 TUNER.....Piano repairman
 TUBE CHECKER.....Chemistry student
 UHF.....Sound made by tackled quarterback
 VACUUM.....Product made by Hoover
 VARIABLE MU.....Sounds of many cats.
 VOLT.....What Russians cast in elections
 WATT.....How the French ask for the time
 ZERO.....Production line shut-down



C99 "TRICKS" PROGRAM LISTINGS

by: Michal Jaegermann

```
int strings [1];
#asm
STRING DATA $0,$1,$2,$3,$4,$5
        DATA $6,$7,$8,$9,$10
$0 TEXT 'None '
    BYTE 0
$1 TEXT 'One '
    BYTE 0
$2 TEXT 'Two '
    BYTE 0
$3 TEXT 'Three '
    BYTE 0
$4 TEXT 'Four '
    BYTE 0
$5 TEXT 'Five '
    BYTE 0
$6 TEXT 'Six '
    BYTE 0
$7 TEXT 'Seven '
    BYTE 0
$8 TEXT 'Eight '
    BYTE 0
$9 TEXT 'Nine '
    BYTE 0
$10 TEXT 'Ten '
    BYTE 0
    EVEN
#endasm

char #greenb = "green bottle";
char #margin = "\n          ";
char #longmargin = "\n          ";
```

```
main()
{
    int btls;

    for (btls = 10; btls > 0; btls--)
    {
        putchar (12); /* clear screen */
        locate (7,1);
        if (verse (btls)) break;
    }
    if (btls <= 0)
    {
        puts ("\n\n\n");
        puts (margin);
        puts ("Well, there is no more anymore!\n");
    }
    getchar();
}

verse (btls)
int btls;
{
    line (btls);
    line (btls);
    puts (margin);
    puts ("One ");
    puts (greenb);
    puts (longmargin);
    puts ("* has accident'ly falled!");
    line (btls - 1);
    putchar ('\n');
    puts (margin);
    puts ("-- More? -- ");
    return ('n' == (32 & getchar())); /* to lower case */
}

line (btls)
int btls;
{
    puts (margin);
    puts (strings [btls]);
    puts (greenb);
    if (btls != 1) putchar ('s');
    puts (longmargin);
    puts (" was standing on the wall.");
}
}
```

Local computer project attracts U.S. gov't

Edmonton Journal April 23/87

By RICK PEDERSEN
Journal Staff Writer
Edmonton-based research on a huge computer has caught the attention of both the United States Department of Defence and New York venture capitalists.

Efforts to sell the U.S government on the Myrias Research Corporation computer project help make it a good investment, according to a study done for Canadian Entech Research Corp.

"This connection . . . has resulted in tremendous interest and financial support from the U.S. government, in particular the Department of Defence," says the re-

port from Entech, a Myrias shareholder.

"I can't really comment on that," responded Myrias president Kenneth Gordon, saying he never discusses customers or their business.

But he confirmed that both the United States and Canadian governments have paid Myrias for research and development. "We've performed some work for them."

Myrias announced this week that it had sold an extra \$4-million common shares, to a consortium including provincially-funded Vencap Equities Alberta Ltd.

"Half of that's ours," said Vencap president D.H. Mather.

The rest came from Adler and Co. of New York and Ventures West Technologies of Vancouver. The \$4 million will be used to develop and market the Myrias parallel processing computer system.

The Alberta government advanced Myrias \$1.5 million in January 1985, and in May of 1986 that debt was converted into preferred shares, according to records filed at Alberta Consumer and Corporate Affairs.

Although the Myrias computer is still under development, Gordon said: "We have a very large prototype running."

The system is based on sophisti-

The Myrias system is based on sophisticated software which can link 512 individual computers

cated software which can link 512 individual computers, each with a half megabyte of memory.

"We believe that's probably the biggest in the world," Gordon said. The goal is a relatively inexpensive system with the power to challenge a giant Cray, the standard supercomputer.

Although other researchers are also developing parallel processors, he said Myrias' is the biggest.

"The largest system we know of elsewhere has 256, and most have 32 or less."

The project has been partly funded by a "large" U.S. and Canadian government joint-development contract, he said.

He would not discuss details, but said this type of machine is usually used to simulate experi-

For example: "You can simulate what happens when a car runs into a wall. That's a lot cheaper than running cars into brick walls."

Or a computer like this could help design an aircraft wing, he suggested. Once again, a sophisticated computer model can test wing designs for less money than the traditional wind-tunnel tests.

"We project significant performance and price-performance advantages for this product in its target market."

V.N. Beaumont, Entech's president, was out of the province and could not be reached for comment.

nova

COMPUTERWARE
52 AIRPORT ROAD EDMONTON
ALBERTA T5G 0W7
(403) 452-0372



Texas Instruments Home Computer

TI-99/4A SALES & SERVICES

AXIOM PARALLAX DIRECT-CONNECT
PRINTER INTERFACE - \$119.95
RS232 CARD - \$160.00
THE GENEVE; NOW TAKING ORDERS
INTRODUCTORY OFFER - \$799.95
APCO TURBO XT.....COMPLETE
WITH MSDOS & GWBASIC 3.2 AND
DATATRAN MONITOR - \$1649.95

NOW AVAILABLE: CONSULTING AND
PROGRAMMING SERVICES FOR IBM
SOFTWARE

PROFESSIONAL REPRODUCTION

PROFESSIONAL COPYING & DUPLICATING

- Reports
- Briefs
- Manuals
- Flyers
- Specifications
- Proposals
- Address Labels
- Transparencies
- Letterheads
- Price Lists
- Directories
- Newsletters
- Resumes

Prices Include

COLLATING, 8 1/2" x 11", 8 1/2" x 14", WHITE, COLOURED OR 3 HOLE BOND

10	90
25	1.75
50	2.50
100	4.00
250	9.00
500	16.00
1000	30.00
2500	70.00
5000	130.00

PRICES PER ONE ORIGINAL
ALL ORDERS SUBJECT TO FEDERAL SALES TAX

- SERVICES -

- 2 Sided Copies
- Enlargements
- Reductions
- Transparencies
- Paper Sales
- Folding/Cutting
- Stapling/Padding
- Cerlox Binding
- Laminating

Two Locations to handle all your Professional Copywork & Printing Services
AMPLE FREE PARKING

NORTHERN COPY CENTRE INC.
13212 ST. ALBERT TRAIL
EDMONTON, ALBERTA

455-8961

broadmoor stationers inc.

165 ATHABASCAN AVENUE
SHERWOOD PARK, ALBERTA

464-4343

"We make a Good Impression"