

New Projects for 2004

FIGUK magazine:

New Project: the FIG-UK CD

New Project: F11-UK Becomes F12-UK

A Virtual Nondeterministic Machine in Forth

Across the Big Teich

Vierte Dimension 3/2003

Forthwrite 124 — February 2004

(inside front cover – blank)

**Forthwrite
February
2004**

**Issue 124
ISSN 0265-5195**

	events
Forth Gesellschaft 20FG04 20th Annual Conference	22
Nominations for the FIG UK Awards - 2003	23
	news
Forth News	3
Forthcoming in Issue 125	22
Errata	28
	reviews
Across the Big Teich	24
Vierte Dimension 3/2003	26
Forthwrite Index	30
	programming
What Languages Fix- Not!	39
	people
Letters	19
	project
New Project: F11-UK Becomes F12-UK A Virtual Nondeterministic Machine in Forth	4 6
New Project: the FIG-UK CD	20
The FIG-UK CD: call for input	21



Editorial

The first Issue of 2004 sees the start of two exciting new projects for FIG UK.

One of the subjects for discussion at the AGM was the production of a CD of material for members. Jeremy outlines the committee's ideas and Douglas reports on the considerable progress made so far.

FIG UK needs input from all members on this important new project so a competition and a prize is announced in these pages.

With stocks of the first batch of F11-UK single board microcomputer now being exhausted, Jeremy Fowell has some good news about a new development in embedding computing.

Also in this issue, James Boyd concludes his article on virtual nondeterministic machines in Forth. It will be interesting to see what other applications people find for the technique.

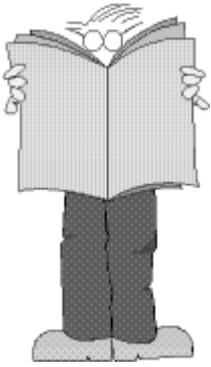
I cannot end this editorial without thanking Douglas Neale and the printers for turning my electronic file for the last issue into paper so smartly. Most of you will have received your copy before Christmas. With things being so busy at that time I feared that it might not arrive before the end of 2003.

If you have any material for publication in the next issue please send it to me by Wednesday 24th March.

Don't forget the monthly IRC session. Our next one is Saturday 6th March 2004 on the IRC server called "IRCNet", channel #FIGUK from 9:00pm UT.

February is a little late to be wishing everyone a Happy New year, but all the same, Good Programming and until next time, sally Forth,

Graeme Dunbar



Forth News

Graeme Dunbar

A roundup of news and events from around the Forth world.

Forth Events

German Forth Conference - 20FG04

The annual conference of the German FIG takes place on the 16th to the 18th of April. This is their 20th annual conference and they extend their welcome to all FIG UK members. For further information see Fred Behringer's report in this issue.

euroFORTH 2004

The likely dates for this euroFORTH conference are the 19th to the 22nd of November 2004, to take place at Castle Dagstuhl in Saarland, Germany. For anyone considering taking part there is a mailing list for delegates at: <http://groups.yahoo.com/group/euroforth/>

FIG UK 25th Anniversary Reunion

FIG UK will be holding its Silver Jubilee this November. Details to follow.

Forth People

FIG UK Webmaster

Jenny Brien, our webmaster has recently undergone major surgery and is presently convalescing. I am sure you will all join the

committee in wishing her a full and speedy recovery.

Jenny reports *she* is doing fine, but her email account is being plagued by spam and asks that anyone emailing her put "FIGUK" in the subject header. She has a dial-up connection, and her spam filter will delete anything it does not recognise as genuine without downloading it.

Forth Resources

Win32Forth

Dave Pochin reports that his Win32Forth Tutorial has been revised and updated. The latest edition (released in December 2003) now covers Win32Forth v6.7001 and can be found at: <http://www.sunterr.demon.co.uk/>

From F11-UK to F12-UK

As reported in the last issue, all the F11-UK microcontroller boards from the first run have been sold. Jeremy Fowell reports on an exciting new development in this issue. Owners of the F11-UK may rest assured that the board will continue to be supported. A new batch of printed circuit boards has been ordered so the F11-UK kit will still be available.

New Project: F11-UK Becomes F12-UK

Jeremy Fowell

As reported in Issue 123 of Forthwrite the first production run of this single board computer kit has sold out, the last one going to Australia. This was a good time to look at the hardware design and see if some improvements should be made. Quite a lot of ideas have been discussed on the F11-UK mailing list and taking these into account three main changes looked important:

1. Increase the number of I/O lines from 20 to maybe 40.
2. Increase the FLASH memory from 32k to 128k bytes. This is already included on the board but only the first 32k is presently used.
3. Add a real-time clock IC powered from the existing RAM back-up battery. This could be used, for example, to wake up the microcontroller every 5 seconds to collect data. This would enable true low-power operation and the board could be powered from a small battery.

And of course it would be good to keep the board size, external connections and price all unchanged. I think it could be done but it gets complicated. (1) and (2) would be best achieved by using some sort of programmable logic device to handle the extra glue logic required. Hard to solder surface mount parts would be needed to keep the size under control. The whole thing begins to lose the simplicity of a minimal system.

After working on this for a while I took a look at the HC12, the upgrade path from the HC11 that Motorola recommends. From the family of different versions the most interesting ones are the new HCS12 parts. Most of the things we need apart from the real-time clock are provided on one chip which simplifies the circuit board quite a bit.

Advantages of the HCS12:

- 256k bytes FLASH memory and 12k bytes RAM.
- Very fast : executes 25 instructions per micro-second (40 ns per instruction)
- Dual 8-input 10-bit analog-to-digital converters.
- Over 60 I/O lines (we will not use them all).
- 2 RS232 serial ports
- Several other serial ports (SPI, IIC, CAN).
- 8 timer inputs.
- 8 PWM timer outputs.

- Same CPU registers as HC11 so will run HC11 code.
- All instructions are 16-bit.
- Fuzzy Logic instructions included.
- Fast 32 x 16-bit signed divide ($0.5\mu s$). [HC11 = $18.5\mu s$ and that's much faster than any of the PIC family].

Disadvantages of the HCS12:

- It is a complex device.
- The top of the range part costs around £20 each (but we save on the cost of external memory ICs etc.).
- Supplied in a surface mount package of the type that will not fit in a low cost socket.

The final argument for changing to the HCS12 is that it could help us reach a wider audience. Motorola has come up with an excellent design here and it seems to be generating quite a bit of interest.

The Proposed F12-UK Board:

- Using mainly surface mount parts would enable the board size to be reduced which would in turn help to offset the higher cost of the new processor.
- Surface mounting of components means that it would be best to supply the board pre-assembled and tested with PygmyHC12 already loaded.
- I/O lines would be buffered with standard logic ICs to protect the expensive CPU.
- The HCS12 processor will be mounted on a small sub-board connected to the main PCB via header pins on a 0.1 inch pitch. This would be a plugable unit which would provide the lowest cost method of replacing the HCS12 should this ever be necessary. A secondary benefit would be the option to plug in a lower cost version of the HCS12.
- The existing DIN41612 edge connector will be retained but with more pins fitted to handle the increased number of I/O lines.
- The price will be as near as possible to the current £49.

Since the HCS12 represents a significant change I have decided to continue making the F11-UK kit available.

So what do you think? Comments would be most welcome either via email to me or the F11-UK mailing list at <http://groups.yahoo.com/group/fig-forth-uk/> .

A Virtual Nondeterministic Machine in Forth

James A. Boyd

In the previous issue, James introduced the technique and outlined his implementation. In this concluding part he discusses the implementation and performance of the software.

Theory of Operation and Usage

Virtual Nondeterministic Machine Theory of Operation

Although the virtual nondeterministic machine uses the same theoretical framework as a nondeterministic machine it's just a simulation. The nondeterministic behavior is simulated by implementing a non-sequential search of the solution space with implicit backtracking if a choice is not "correct". This allows the same expressiveness achieved with a nondeterministic machine and usually requires fewer steps to a solution than a deterministic algorithm requires. The nondeterministic algorithm for the eight queens problem found a solution in only 58 tries compared to 876 tries for the deterministic algorithm.

The implicit backtracking is implemented by saving the machine state (the data and return stacks) as well as a list of valid choices in a history stack each time `choice` is executed. The execution of `failure` forces backtracking to the most recently saved history by restoring the machine state, then removing one of the choices. It's as if `failure` never occurred and the latest occurrence of `choice` simply made a different choice than the one which forced backtracking. If the choices for a particular instance of `choice` are exhausted then the machine state for the previous `choice` is restored. Implicit backtracking occurs as long as there is a valid choice. Exhausting all the choices means there were no choices leading to a valid solution and execution resumes with the word after `failure`. Note that although backtracking is used it is handled by the virtual nondeterministic machine and is nearly transparent. The execution of `success` empties the history buffer to terminate the nondeterministic behavior.

The need to save history data is one disadvantage of simulating the behavior of a nondeterministic machine. An actual nondeterministic machine needs no history data because the choice it makes is always "correct" if such a choice exists. At present such a machine only exists in a theoretical framework.

Normally the machine state consists of the contents of both stacks but some algorithms may need other areas of memory saved as part of the machine state. Therefore, two deferred words have been provided for the code needed to save and restore other data structures `SaveOther` and `RestoreOther`. The fourth example in the file **VNMMISC.F** shows how to use these two words. Note that extra data is restored in the reverse order it is saved.

Data structures which are not part of the machine state must be used with caution. An array such as the queens array used in the N queens problem works fine because items (the queens) are added sequentially. Data structures with items which are shuffled or exchanged must be part of the machine state either by being placed on the stack or explicitly making them a part of the machine state by including code to save and restore them. It would be prudent to limit what is included as part of the machine state to the minimum needed to avoid exhausting room for the history data.

The N queens problem is the main demo for this paper. The problem has a time to solution which is irregular but appears to increase exponentially with problem size. The nondeterministic version of this problem has a time to solution which appears to be polynomial but not as low an order as a theoretical nondeterministic machine.

The best candidates for a virtual nondeterministic machine seem to be problems whose best algorithms have a time which is exponential or polynomial of large degree. The max clique problem and the knapsack problem ^[1] are two problems whose best algorithms are exponential yet have a time to solution which is polynomial for nondeterministic versions of these algorithms. These as well as other nondeterministic algorithms mentioned in Fundamentals of Computer Algorithms ^[1] may work well with a virtual nondeterministic machine.

Virtual Nondeterministic Machine Compared to Nondeterministic Control Words

L. L. Odette gave us nondeterministic control words; an interesting control structure which promised to increase the expressive power of Forth. It also had implicit backtracking with a non-sequential search of the solution space.

The nondeterministic word `nqueens` is very similar in appearance to Odette's `queensoln`. They both have a nondeterministic nature. The big difference between `nqueens` and `queensoln` is the use of a different theoretical framework for a starting place.

Odette's nondeterministic control words are based on the idea of a "coin tossing controller" which randomly chooses one of two branches to take. Odette's coin tossing controller meets the theoretical expectations of such a controller except for not having a true random number generator; therefore it is difficult to see how its performance can be improved.

The virtual nondeterministic machine is based on a nondeterministic machine as described by Horowitz and Sahni. The virtual nondeterministic machine shows some of the promise of a nondeterministic machine yet the need for improvement is clearly visible. Although the virtual nondeterministic machine outperforms Odette's nondeterministic control words its performance can be improved by bringing its behavior closer to the theoretical behavior of a nondeterministic machine.

Both nondeterministic methods have a probabilistic nature. Their nondeterministic nature implies a non-sequential search of the solution space for a particular problem. The virtual nondeterministic machine performs as well as it does because its choices are more random.

randomn is Odette's nondeterministic number generator. **Table 1** is a sample output of choice versus randomn; the code is in the second example in the file **VNMMISC.F**.

Test of first choice for two nondeterministic words **randomn** and **choice**.

50 RandomnTest

```
9998 9999 9999 9997 10000 10000 9998 10000 9998 9999
10000 10000 10000 9998 10000 9998 10000 10000 9998 9998
9999 9997 9999 10000 10000 10000 10000 9999 10000 10000
9999 10000 10000 10000 9998 9999 10000 9999 10000 10000
9998 9999 10000 9997 10000 9996 10000 10000 9999 9997
```

50 ChoiceTest

```
3737 4547 9179 4726 7880 4659 8727 3934 4200 1362
9125 8719 9417 4501 839 7674 7749 1419 1842 6892
914 6961 7602 9968 8134 3469 1084 7994 6123 787
2761 9566 9104 6261 600 775 8793 4732 69 5219
4410 8545 4139 3987 2673 7543 3276 5676 7803 9071
```

Table 1

The tests consist of displaying the first choice for choice and randomn with an input of 10,000 and clearing the history, or backtracking, data for each iteration. Both tests were run for fifty iterations. The nondeterministic control word randomn is severely biased toward the high end; about half of the time the output was 10,000 and a quarter of the time it was 9999. The output of choice appears quite random and is only limited by the randomness of the number generator used in choice#. The choice is also quite random (as limited by the number generator) for each instance of backtracking.

Times for three N queens algorithms. All times are in milliseconds on a 1.6 GHz machine.							
queens	nqueens			queensoln			deter- ministic
	fast	average	slow	fast	average	slow	
4	0	4	16	0	3	16	0
8	0	11	32	0	9	16	31
10	0	12	16	0	13	16	0
12	0	9	32	0	24	47	16
14	0	21	109	0	66	438	31
16	0	21	47	0	272	1313	250
18	0	39	203	0	1036	3656	1250
20	0	42	500	47	3452	15297	7125
22	15	78	750	297	26403	147437	73922
24	15	48	172	79	24604	143766	19953
26	15	98	485	563	191463	1082562	22047
28	16	121	1125	27672	628442	1713859	188750

Table 2

Table 2 is a table of the average, fastest, and slowest times for the N queen's problem using `nqueens`, `queensoln`, and a deterministic method. **Table 3** is a chart of the average times using a logarithmic scale for the vertical axis. The time to solution for the deterministic `nqueens` problem and `queensoln` appears to increase exponentially with problem size while the time to solution for the nondeterministic `nqueens` appears to increase according to a third or fourth degree polynomial function of problem size. One solution for the 100 queens problem was even found by `nqueens` in about three minutes running on a Commodore 64! Running on Win32Forth, a solution for the 500 queens problem was found in under 18 seconds! An actual nondeterministic machine would have a time to solution for the N queens problem which would increase according to a second degree polynomial function of input size.

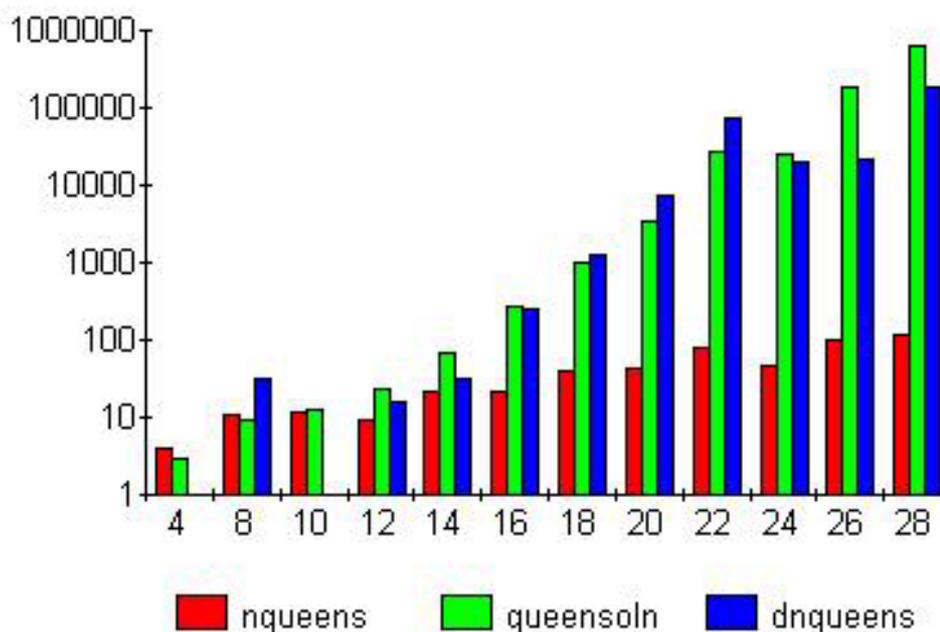


Table 3

Table 4 is a chart of history memory used by `nqueens` and `queensoln`. For all tests, `queensoln` was modified to keep the queens in an array instead of the stack (this actually helped the performance of `queensoln`) and all history data was removed prior to each run. The virtual nondeterministic machine clearly requires less memory than Nondeterministic Control Words. Even though more data is stored by `choice` than `Odette's oneof`, `choice` is executed far less frequently.

Another demo of `Odette's` is list permutation. The same concept can be written to work with a virtual nondeterministic machine instead of nondeterministic control words. The virtual nondeterministic machine version of list permutation is given as the third example in the file **VNMMISC.F**.

One should be careful when trying to implement `Odette's` code on Win32Forth. Win32Forth uses relative addressing but the return stack holds absolute addresses.

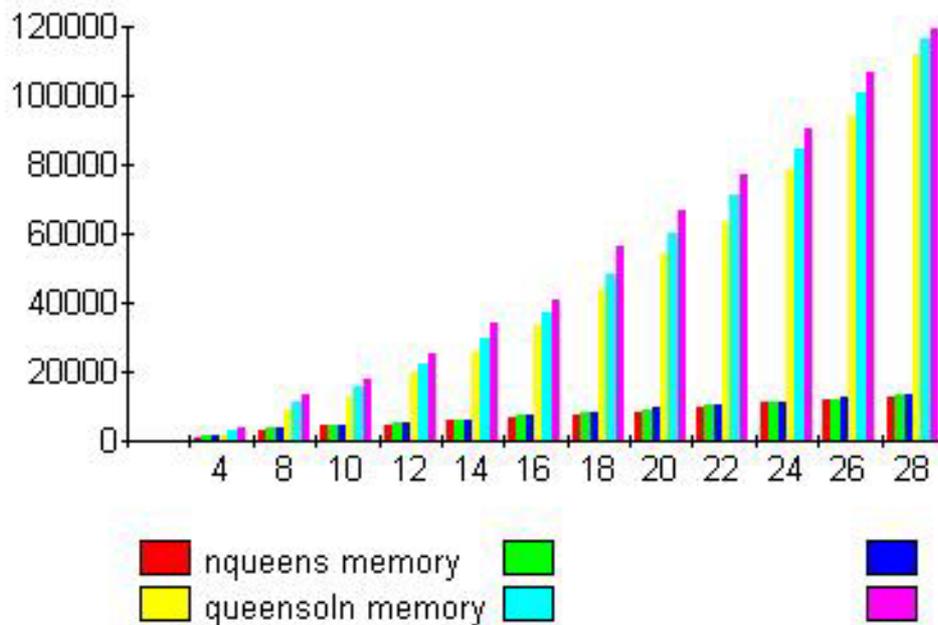


Table 4

Exploiting Implementation

The way the virtual nondeterministic machine is implemented can be exploited to achieve useful results. When a nondeterministic algorithm which has not been terminated with success returns an answer, the algorithm can be forced to search for another answer by typing `failure` at the keyboard or using `failure` in a program which calls the nondeterministic algorithm. The same idea can be used with Odette's Nondeterministic Control Words. This is how the `permute` example works.

We must keep in mind that this would not work with an actual nondeterministic machine. Any occurrence of `failure` before `success` would cause the algorithm to fail. The same technique of using `failure` to force backtracking is used as the first example in **VNMMISC.F** as a simple test of the nondeterministic operations to make sure they work. When `ShuffleTest` is executed the numbers zero through ten will be displayed in random order.

Multitasking

Multitasking with a virtual nondeterministic machine should be handled carefully if at all because of the implicit manipulation of the stacks. The virtual nondeterministic machine should work without modification as long as the nondeterministic task is independent of any other tasks. If there is more than one nondeterministic task then the buffer words need implemented so that each nondeterministic task has its own buffer.

If there is only one buffer then only the task running the nondeterministic algorithm can execute `failure` or the machine state saved with the latest occurrence of `choice` will be

restored to the stacks of the task which executes `failure`. This regrettably precludes running a nondeterministic algorithm in a background task and forcing backtracking to another solution from the console.

Implementation

Virtual Nondeterministic Machine Buffer Interface

Unlike Odette's nondeterministic control words, the history data for the virtual nondeterministic machine is saved in a buffer instead of compiling it on top of the dictionary. Using the standard words `allocate`, `resize`, and `free` the history buffer is only expanded as needed and reclaimed when an algorithm concludes. All the words that would need changed if the buffer implementation is changed have been grouped together in the file **VNMHISTORYBUFFER.F**. Implementation specific areas such as multitasking have been avoided.

Saving and Restoring the Stacks

The words `SaveDataStack`, `SaveReturn`, `RestoreDataStack` and `RestoreReturn` are specific to Win32Forth. Slower versions are provided in the file **VNMSTACKS.F** which are more generic but still require the return stack is used for return addresses. Win32Forth and one of the slow versions require the following words:

- `sp0` and `rp0` are usually two user variables which hold the initialization addresses for both stacks (where to reset the stack pointers).
- `sp!` sets the data stack pointer to the value on the data stack.
- `rp!` sets the return stack pointer to the value on the data stack.
- `sp@` and `rp@` return the address of the data and return stack pointers (top of stack). The phrase `rp@ @` performs the same function as `r@` and the phrase `sp@ @` performs the same function as `dup`.

The Main Virtual Nondeterministic Machine Operations

choice takes a number from the data stack and saves history data with a set from zero to N inclusive and selects one element from that set. The stack depth is first checked by `choice` to make sure there is at least one number on the data stack to specify the size of the set. The CFA of `expand` is saved on the return stack before a copy of the return stack is saved and discarded afterwards. The group size is temporarily stored on the return stack while a copy of the data stack is saved. Finally the set (or group) data is generated and saved by `group` with the first choice left on the data stack.

`group` generates the group data by saving an offset into the yet to be generated group data and the last element of the group. `group` uses the number generator `choice#` to pick an offset into the group data and use that offset to produce the number which would be at that offset. The entire list of possible choices is not generated until `failure` occurs when it is needed.

`expand` decompresses the group data using `generate`, puts the CFA of `(choice)` on top of the saved return stack in the history buffer, and calls `(choice)` to select one element. The history data for the latest occurrence of `choice` is removed by the backtracking. `expand` and `(choice)` preserve the history data by switching the main and auxiliary pointers with the buffer interface word `pswitch` if there are valid untried choices for the latest occurrence of `choice`.

`generate (n --)` places **n** cells on the history stack from zero to **n-1** in ascending order.

failure executes `(failure)` and displays the message “No Solution”. The nondeterministic behavior of `failure` is factored into `(failure)` in case the message is not desired.

`(failure)` has two different behaviours depending on whether there is any history data. If there is history data it sets the auxiliary pointer to the same value as the main pointer, skips the group data, and restores both stacks. The process of restoring the machine state leaves the main pointer where it was just prior to saving the latest chunk of history data effectively removing it. The CFA on the return stack is removed and executed. Program flow resumes where it left off when the machine state was saved and does not resume after `(failure)`. If there is no history data `(failure)` executes `success` to terminate any nondeterministic behavior. Program flow resumes with the word which executed `(failure)`.

success resets `SaveOther` and `RestoreOther` to a no-op and executes `remove` to empty the buffer thereby terminating any nondeterministic behavior. If any data other than the stacks is to be saved as part of the machine state then `SaveOther` and `RestoreOther` must be initialized after any initial success and before the first occurrence of `choice`.

Extensions to the Virtual Nondeterministic Machine

Other words can be added to the virtual nondeterministic machine to extend its functionality. The new words need to set up the history data in an appropriate form for `failure`. The CFA of the word to be executed when `failure` occurs must be placed on the return stack before `SaveReturn` is executed and then removed. A CFA **must** be placed on the return stack even if it is the CFA of `noop` a system no-op. Any parameters not meant to be kept on the data stack (such as group size) must be temporarily removed while `SaveData` is executed. Finally a group is setup either by calling `group` with a size on the stack; by generating a custom group (such as a list of CFAs) and saving the size; or by storing a size of zero for no group. Note that since `failure` does not preserve the history data when it is executed, the word with its CFA saved with the return stack needs to take care of this if `failure` is to restore the same history the next time it is executed.

`suspend` is a handy utility to use in a program instead of aborting when a key is pressed. It saves one instance of history data with a group size of zero; displays the contents of the data stack and aborts. A no-op is saved on the return stack since the history data saved by `suspend` does not need preserved. The program resumes where it was suspended regardless of recursion or stack depth when `failure` is executed. This is handy to use to stop a program to check how it is functioning, check some variables and then resume execution by typing `failure` at the keyboard and it makes a nice example of how to set up the history data for `failure`. It is also quite useful to use `suspend` in a program instead of `abort` just in case a key is pressed by accident.

Further Research

Parallel Processing

Due to the probabilistic nature of the virtual nondeterministic machine the time to solve a given problem will be different each time. As can be seen from **Table 2**, the larger the N queens problem the larger the ratio of the average to fastest solution. By the time the problem size reached fifty queens the fastest fifty queens problem was over four hundred times faster than the average. The same phenomenon would occur if several different computers solved the same problem nondeterministically as long as each random number generator had a different seed value. The fastest times for all the nqueens problems from 8 to 50 was under 17 milliseconds! This can be exploited to program a parallel computer to solve a nondeterministic algorithm by having each processor solve the entire problem but with a different seed value for each random number generator. The first processor to finish would signal the main processor.

Neural Network Number Generator

One possibility to enhance the performance of the virtual nondeterministic machine is to use a neural network to generate the random numbers. `choice`, `success` and `failure` would be modified to send a signal to an output port to train the network. The number generating word could “clock” the network (make it go through one instance of signal propagation) after it gets the network's output. `success` could signal the ending or beginning of a new algorithm and the number of failures for each occurrence of `choice` could be the training signal, the lower the number of failures the better the network's output. It might be possible for a sophisticated enough network to learn the “pattern” of the correct solution and start making better guesses.

References

- [1] Fundamentals of Computer Algorithms by Ellis Horowitz and Sartaj Sahni
- [2] Nondeterministic Control Words in Forth by L. L. Odette Dr. Dobbs Journal September 1983

Appendix: list of files

This is a list of all files included in the two parts of this article.

Published in issue 123

VNM.F	The nondeterministic machine source code
VNMHISTORYBUFFER.F	Words needed by VNM.F
NONSTANDARFD.F	Words needed by VNM.F
NQUEENS.F	The NQueens problem given as an example of the use of VNM.F

Published in this issue

VNMMISC.F	Miscellaneous examples of the use of the nondeterministic words defined in VNM.F.
VNMSTACKS.F	Generic stack words.
VNMDEMOS.F	Additional demonstration examples of the use of the nondeterministic machine words.
BADQUEENS.F	An example of what happens when an erroneous attempt to show partial results is made.
GFORTHWORDS.F	

File: VNMMISC.F (sheet 1 of 2)

Miscellaneous examples of the use of the nondeterministic words defined in VNM.F.

```
\ Miscellaneous examples
\ James Boyd 10/13/01 03:03
```

```
decimal
```

```
\ Example one -- simple test of nondeterministic words
\ The following should display the numbers 0 thru 10 in random order.
```

```
: ShuffleTest ( -- )
  success 10 choice cr . (failure)
  cr ." Test complete." ;
```

File: VNMMISC.F (sheet 2 of 2)

```
\ Example two -- test of choice vs. randomn

: ChoiceTest ( n -- ) success cr
  0 do 10000 choice 8 .r success 16 ?cr loop ;
\ RandomnTest requires L. L. Odettes Nondeterministic Control Words
\ : RandomnTest ( n -- ) cr
\   0 do 0 , 10000 randomn 8 .r new 16 ?cr loop ;

\ Example three -- Odette's permute example using choice

: -roll \ the following has no net stack effect: n roll n -roll
  dup
  begin ?dup while 3 roll >r 1- repeat
  begin ?dup while r> swap 1- repeat ;
: insert
  1+ dup>r dup roll
  swap 1- choice -roll
  r> ;
: permute ( N1 ... Nm m -- N1 ... Nm m )
  success dup if 1- recurse insert then ;
: printpermute ( N1 ... Nm m -- )
  cr 0 ?do 3 .r loop ;
: ptest ( N1 ... Nm -- )
  depth dup>r permute key? if suspend then
  printpermute (failure)
  cr ." No more permutations for" r> 3 .r ." numbers" ;

\ Example four -- Demo how to use SaveOther and RestoreOther
\ three byte arrays

: array create allot ; \ simple byte arrays
20 array small
50 array large
40 array medium
\ m>history and mhistory> take an address and byte count as parameters
: SaveArrays
  small 20 m>history
  medium 40 m>history
  large 50 m>history ;
: RestoreArrays
  large 50 mhistory>
  medium 40 mhistory>
  small 20 mhistory> ;
: IncludeMyArrays ( -- )
  ['] SaveArrays is SaveOther
  ['] RestoreArrays is RestoreOther ;
\ trivial example
: demo ( n n2 -- ) success IncludeMyArrays
  choice swap choice cr .s + 20 = 0= if failure then
  success ;
```

File: VNMSTACKS.F (sheet 1 of 1)

Generic stack words.

```
\ Alternate stack saving and restoring words
\ James Boyd   September 13th, 2003 - 3:25

: -?rdepth  C" rdepth" FIND NIP 0= ;
-?rdepth [IF]

( requires -- rp0 rp@ cell )
: (NestDepth)  ( -- n )  rp0 @ rp@ - cell / ;
: NestDepth  ( -- n )  (NestDepth)  rp0 @ rp@ - cell / - ;
: m-  NestDepth 0 do postpone 1- loop ; immediate
: rdepth  ( -- n )  rp0 @ rp@ - cell / m- ;

[ELSE]

( requires -- rdepth )
: (NestDepth)  ( -- n )  rdepth ;
: NestDepth  ( -- )  (NestDepth) rdepth - ;

[THEN]

: mr>  NestDepth 0 do postpone r> loop ; immediate
: m>r  NestDepth 0 do postpone >r loop ; immediate

: rStackPurge  mr>
  begin rdepth while r> drop repeat m>r ;
: (RestoreReturn)  mr> history>
  begin ?dup while history> >r 1- repeat m>r ;
: RestoreReturn  mr> rStackPurge (RestoreReturn) m>r ;
: SaveReturn  mr> 0
  begin rdepth while r> >history 1+ repeat
  >history main>aux (RestoreReturn) pswitch m>r ;

: depth  ( -- n )  sp@ sp0 @ swap - cell / ;
: StackPurge  begin depth while drop repeat ;
: (RestoreDataStack)  history> 0 ?do history> loop ;
: RestoreDataStack  StackPurge (RestoreDataStack) ;
: SaveDataStack  0
  begin depth 1- while swap >history 1+ repeat
  >history main>aux (RestoreDataStack) pswitch ;
```

File: VNMDEMOS.F (sheet 1 of 1)

Additional demonstration examples of the use of the nondeterministic machine words.

```
\ Some more VNM stuff
\ James Boyd   October 31st, 2003 - 19:15

: sqrt   ( n -- n2 )   s>f fsqrt f>s ;

: ?prime ( n -- f ) \ returns false flag for 0, 1 and negative numbers
  dup 2 < if drop false exit then
  dup sqrt 1+ 2 ?do dup i mod 0= if 0= unloop exit then loop 0<> ;

: ?even  ( n -- f ) \ returns true flag if n is even
  1 and 0= ;

\ Always chooses an even number
: evenDemo ( n -- )
  choice dup ?even
  if      success .
  else    failure drop
  then ;

\ Always chooses an odd number if n > 0
: oddDemo ( n -- )
  choice dup ?even
  if      failure drop
  else    success .
  then ;

\ Always chooses a prime number -- range 0 - n
: primeDemo ( n -- n2 f ) \ returns n2 and a flag -- true if n2 is prime
  choice dup ?prime dup
  if      success
  else    failure
  then ;

\ Always chooses two primes such that the second prime is greater than 3 * the
\ first prime
: prime2Demo ( n -- )
  dup choice dup ?prime 0= if failure 2drop exit then
  swap choice dup ?prime 0= if failure 2drop exit then
  2dup swap 3 * <
  if      failure 2drop
  else    success swap cr 8 .r 8 .r
  then ;

\ Always chooses a number such that when added to n, the result is prime.
: prime3Demo ( n n2 -- n3 flag ) \ returns n3 and flag -- true if n3 is prime
  choice + dup ?prime dup
  if success else failure then ;
```

File: BADQUEENS.F (sheet 1 of 1)

An example of what happens when an erroneous attempt to show partial results is made.

```
\ nqueens with mistake
\ James Boyd   November 11th, 2003 - 3:39
```

```
needs nqueens.f
```

```
Comment:
```

```
    Printing partial results with a nondeterministic machine is a bad idea.
    In the file VNMMISC.F ShuffleTest displays partial results to test the
    virtual nondeterministic machine.
```

```
    In the same file Demo displays the stack to show what the virtual
    nondeterministic machine is doing.
```

```
Comment;
```

```
\ prints partial results, or tries to. Run to see the mess!
\ All the bad version does different is print a partial result
\ before failure with the result that a lot of nonsense is displayed.
```

```
: (badnqueens)  ( n -- )
    ?dup 0= if cr ." No solution for zero queens" exit then
    dup setQueenSize
    1- 0 ( n-1 column )
    begin
        over choice
        dup . \ printing partial results is not a good idea!
        attacks? ( n-1 column queen f )
        if failure 2drop drop
            exit
        then
        addqueen 2dup < ( n-1 column f )
    until ( n-1 n )
    space showqueens drop ;

: badnqueens ( n -- ) success (badnqueens) success ;
```

```
cr cr
8 badnqueens
10 badnqueens
```

File: GFORTHWORDS.F (sheet 1 of 1)

Words need to convert the code to Gforth.

```
\ James Boyd   November 11th, 2003 - 3:39
\
\   To work on Gforth a random number generator needs defined.
\   for the demos to work on Gforth:
\   need to define

: s>f  ( n -- f: f )  s>d  d>f ;
: f>s  ( f: f -- n )  f>d  d>s ;
: ?cr  ( n -- )  drop ; \   Don't know how to do this one. Sorry.

\   need to replace the word  DUP>R  with the two words  DUP >R
```

Letters

The Editor is always pleased to receive correspondence from members of the Forth community and email is undoubtedly the most convenient medium. Recently I received and passed on a request for information about Dave Pochin's Win32Forth tutorial which led to the following exchange. See this issue for news of a forthcoming article by Dave Pochin.

Hi!

Perhaps you can help me out.

I can not seem to find the win32forth tutorial by Dave Pochin. It can not be found on the Daemon site, and nor do

I see any other location for it. Would you know where it can be found?

Thanks

Gerald King

Hi Gerald,

Thanks very much for your query. I've been busy revising the site for version 6...

The address is the same as before. www.sunterr.demon.co.uk

But all the old short cuts no longer work.

Just as I got this working, the faithful PC decided to have a battle between my Internet provider and my anti virus software, so I have been off line for nearly a fortnight. All is now sorted. Sorry for the delay.

Please let me know if you still have problems.

And email me any comments or suggestions about the new version.

The page does work, I have just checked it out

Regards.

Dave

New Project: the FIG-UK CD

Jeremy Fowell

This issue of Forthwrite sees the announcement of an important new project, our CD. The idea was the brainchild of Douglas Neale and Chris Jakeman and a topic of some enthusiastic discussion at the recent AGM. Since then Douglas has been hard at work turning the idea into reality, which is no small task.

It is planned to make the CD available to FIG UK members through the web site at: <http://www.fig-uk.org> with payment by PayPal. The exact price has yet to be confirmed but will be fairly low.

Generous Support

Already the content is impressive; the net has been cast far and wide. Stephen Pelc of MPE has generously agreed to the inclusion of a version of VFX Forth (see back cover). There is Tom Zimmer's Win32 Forth, Pygmy and F-PC. gForth and bigForth for Linux are there too. We have Forth Standards documents, articles about Forth, a book list and a list of Web sites. I think you get the idea.

The volume of material means that we would like our members to get involved as well. If you email Douglas at dneale@w58wmorden.demon.co.uk he will send you a list of all the files split into their directories. There are many files of Forth code that need to be examined to check their suitability. Obviously this is time consuming to say the least so if there is an area you are familiar with or interested in please take a look through some of the files. Does the work seem to be well written or is it write-only code? If you have an Amiga machine, for example, perhaps you could run one of the applications and report your findings in about a paragraph.

Also as Douglas has said elsewhere in this issue, if you have ideas please let him know.

```

forthcd_list - Notepad
File Edit Format View Help
Volume in drive D is SLAVE80_1
Volume Serial Number is E499-AA2C

Directory of D:\DOWNLOAD(A-M)\Forth CD

29/10/2003 08:50 <DIR> .
29/10/2003 08:50 <DIR> ..
08/01/2004 13:43      1,970 forth cd.txt
08/01/2004 13:45 <DIR> Introduction
08/01/2004 13:45 <DIR> DOS Forths
08/01/2004 13:45 <DIR> Windows Forths
08/01/2004 13:45 <DIR> Mac Forths
08/01/2004 13:46 <DIR> Linux Forths
08/01/2004 13:46 <DIR> Embedded and PDA Forths
08/01/2004 13:39 <DIR> Forth Standards
08/01/2004 13:46 <DIR> Forthwrite Archives
08/01/2004 13:47 <DIR> Utilities
08/01/2004 13:47 <DIR> Forth on the Internet
08/01/2004 15:52 <DIR> Cross Compilers
08/01/2004 16:21 <DIR> Old Machines
08/01/2004 16:45 <DIR> Applications
08/01/2004 16:58 <DIR> FAQ
08/01/2004 17:01 <DIR> Scientific
08/01/2004 17:30 <DIR> Literature
11/01/2004 13:07 <DIR> Member Stuff
08/01/2004 17:45 <DIR> ForthDimensions
13/01/2004 14:26      0 forthcd_list.txt
                2 File(s)          1,970 bytes

Directory of D:\DOWNLOAD(A-M)\Forth CD\Introduction

08/01/2004 13:45 <DIR> .
08/01/2004 13:45 <DIR> ..
31/03/1997 19:39      558,856 WELCOME.ZIP
31/03/1997 19:37      3,446  README.TXT
11/01/2004 15:33 <DIR> Index to Forth Articles_files

```

We hope this CD will build into a valuable reference work bringing together the best Forth code, documentation and information in one place. If you can, please get involved. Many hands will make lighter work.

The FIG-UK CD: call for input

Douglas Neale

At the last Fig UK AGM I was asked to build a Forth CD which we could sell to our members for a minimal sum. I have spent the intervening time putting together as much stuff as I could garner from our archives and various web sites. At present this amounts to nearly 200 Mb of material made up of various Forths for a wide variety of environments and processors, lots of documentation, including our own back issues from No 105, some Standards etc etc.

CD Material

The headings under which the current material is organised are as follows (in alphabetic order):

Applications, Cross Compilers, DOS Forths, Embedded and PDA forths, FAQ, Forth on the Internet, Forth Standards, ForthDimensions, Forthwrite Archives, Introduction, Linux Forths, Literature, Mac Forths, Member Stuff, Old Machines, Scientific and Windows Forths.

Because our archive PDF files only go back to issue 105, I am looking for suggestions to include articles from earlier issues since Chris Jakeman took over from Gil which you think merit including in this compendium.

We think the CD ought to include a recommended book list, so please tell me which Forth books you would recommend.

If you have ideas, suggestions or contributions to the CD please email or post them to me.

Competition

Finally, just to encourage everyone to join in, there is our usual competition. In this case, we will award a free year's membership to the person who comes up with the best CD label for the CD. The winner will be chosen on the basis of merit and minimal printing time on my HP Ink Jet! The deadline for this competition, and anything you want included is one month from when you receive the next issue.

Looking forward to hearing from you,

Douglas Neale.

(Editor's note: the directory listing itself runs to 37 pages – too big for inclusion in Forthwrite. Douglas asks that if you would like to see what is currently included on the CD then please contact him and he will email you a text file containing the complete directory listing.)

Forth Gesellschaft 20FG04 20th Annual Conference

Fred Behringer

The German FIG will be holding its 20th Annual Conference this year. Fred Behringer, a member of both Forth Gesellschaft and FIG UK, reports on the activities planned for this prestigious event.

The 20th annual conference of Forth Gesellschaft is being held on Friday April 16th to Sunday the 18th at Hotel Schuetzenhof in Burgstaaken on the island of Fehmarn. This is going to be a jubilee meeting with many surprise activities. The conference languages are Forth and German. However, there will hardly be any member of Forth Gesellschaft unable to follow discussions and conversations in English. FIG UK members are welcome. Ask for price reduction if required. Apply to Dr. Ulrich Hoffmann, the conference organiser, for further information or visit Forth Gesellschaft at www.forth-ev.de.

Burgstaaken is the fishery harbour of Burg which in turn is the capital of Fehmarn. Fehmarn is the one and only Baltic Sea island of Slesvic-Holstein, the most northern federal state in Germany. Places not far away are Kiel and Luebeck. There are train and car connections via the Fehmarn-Sund bridge.

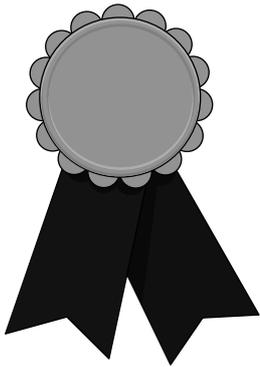
Forthcoming in Issue 125

Simple State Machines by Jenny Brien

In this article, Jenny explains how state machines can be used to parse number input intelligently, this being an example of a process that is easy to specify but hard to program procedurally.

Win32Forth Regions by Dave Pochin

Regions are one of the categories of Microsoft Windows ® API functions. Dave Pochin describes his experiments with Regions in Win32Forth and goes on to develop a demonstration program for manipulating Regions.



Nominations for the FIG UK Awards - 2003

The FIG UK Awards of 2002 were won by Ed Hersom and Howerd Oakford. These awards are given to encourage effort and recognise achievement.

Please take the time to look back over the past year and send in your personal nominations for 2003.

Free membership

To nominate your candidate, send in a note of who, in your opinion, most deserves an award and why. The recipient of each award will receive a place in the FIG UK web-site's Hall Of Fame, a mention in Forthwrite and ***a year's free membership***.

Achievement

The Achievement Award is given to the member who has made the best contribution towards Forth during 2003. The contribution may be a presented paper, a library of code or an idea which inspires others. Whatever form it takes, the contribution must support the goals of FIG UK.

Forthwrite

The Forthwrite Award is given to the member who has made the best contribution to Forthwrite magazine during 2003. The contribution may be judged on quality of writing, tutorial potential, entertainment value or other criteria which the Forthwrite Team deem appropriate.

The awards are judged by the officers of FIG UK. All who are members on 31st Dec. 2003 are eligible (except the judges). Send in your nominations to the Editor or other committee member.

Across the Big Teich

Henry Vinerts

This material was prepared for Vierte Dimension by Henry Vinerts, and printed by kind permission of Forth Gesellschaft (German FIG).

Greetings from California!

As you read this, the Silicon Valley Forth Interest Group will probably no longer be listed on the Web as the Silicon Valley Chapter of the FIG. In the spirit of Forth, the word "chapter" will be factored out and SVFIG will suffice to describe this endangered-species preserve in California. George Perry, the president of FIG, sees no need to dispute the autonomy of SVFIG, the last surviving branch of an organization which just prior to Y2K still sported over a thousand members.

So, I invite you to visit <http://www.forth.org> and take a look further into SVFIG news and activities. Since our webmaster, David Jaffe, diligently posts notes of past monthly meetings, as well as announcements about future ones, and provides links to websites of our speakers, I do not wish to go into great detail about the technical presentations in the meetings (also partly because many times they are "over my head" and partly because often they may be too specific to attract the attention of the average reader). The purpose of my messages -- as I have said before -- is to send them to you whenever I have had the opportunity to witness a live show, with real forthers as the actors. Of course, if I think that I should share with you any personal observations of historical, hysterical, or philosophical value that your editor might deem printable, I'll probably add them to the message.

Silicon Valley FIG Meeting – December 2003

As we might have expected, after the yearly Forth Day and just before the holidays, the attendance at the December 13th meeting was on the skimpy side, peaking around 18 at lunch time. Dr. Tim Duncan, director of our host's (Cogswell College) department of Digital Audio Technology, continued with his lectures and demonstrations of MIDI implementation, filling the morning session. Due to Tim's familiarity with Forth, it is included in the college course catalog between C and Lisp as one of the recommended core programming languages for the bachelor's degree in his department.

After lunch, Dr. Ting's talk about keyboards, Chinese characters, etc, kept the group seated for about an hour, following which they happily fell out (It's an Army expression:

to "fall out" from being at attention.) to a big break of gossip, chit-chat, and watching Kevin Appert steer a browser all over the Web with a projector-equipped laptop.

Silicon Valley FIG Meeting – January 2004

The January 24, 2004, SVFIG meeting was a real "winner." More people were there already in the morning than we had on Forth day in November. Our organizers had done a super job, even advertised the celebrity appearance in the San Jose Mercury News. Who was the celebrity? Perhaps I should first ask how many readers still remember Rafael Deliano's article in the 1/1996 volume of *Vierte Dimension*. And, incidentally, how many have heard of the Canon Cat?

Well, Dwight Elvey brought a Canon Cat, but Jef Raskin came himself, to talk to us about his "THE" project--The Humane Environment. There isn't enough room in my e-mail box to describe the man and all his accomplishments. Even Rafael's three pages in the VD would have to be supplemented. Let me just say that Dr. Raskin may be best known as the creator of the Apple Macintosh project of 20 years ago and more recently as the author of "The Humane Interface" (Addison Wesley, 2000), which has already been translated into seven languages. Although Raskin admits partial responsibility for the "inhuman" GUIs that computer users are expected to live with today, his self-directed mission is to improve the human-computer interface for future generations by replacing inefficient designs and removing bad habits. He started with Forth on the Canon Cat and he intends to return to Forth in his THE project.

In the afternoon, Randy Thelen, whom we met on Forth Day, returned with his home-made TTL-based Forth computer MIPPY and drew compliments from the old-timers on his design and presentation. Dr. Ting finished the day with a number of topics related to his recent work in Taiwan and showed us his latest book "Programming Embedded Systems in Forth," which is published by O'Reilly branch in Taipei and is done in Chinese, except for various scatterings of Forth code throughout most of the 365 pages. The primary purpose of the book is to teach Chinese people how to design a CPU. An English version may come to be only if O'Reilly's main office decides to produce it.

In closing I would like to thank Dr. Ulrich Hoffmann for a kind invitation to the 20FG04 Forth Tagung on the Fehmarn Island. The last time I was in that neighborhood, was in Flensburg, as the war ended in 1945. Sorry, folks, I won't be able to attend, but I do wish a good time to everyone who will be there.

Henry



Vierte Dimension 3/2003

Joe Anderson

Joe provides a look at the latest issue of the German FIG magazine.

Editorial.

4

Friederich Prinz

Friederich Prinz regrets that this time Vierte Dimension has turned out so meagre. The membership has again risen slightly and the relationship to Forth-adherents in other countries is as good as never before. All the same, nobody can be found who wants to write articles for Vierte Dimension.

Readers' Letters.

5

Two letters to the editor: VD is good, the content varied (Behringer) - The Forth-Gesellschaft's stand at the Linux conference (Paysan).

HolonForth.

6

Wolf Wejgaard
www@tiredofspam.com

Writing programmes is relatively easy. Incorporating changes is also not much harder, at least in Forth. But keeping the overall view while making changes is a damned difficult job.

Definition of Terms: Compiler/Interpreter.

7

Ewald Pfau
ehp@ear.co.at

Reactions to a letter of the author's in comp.lang.forth.de and answers to it. One person writes: "Interpreting" is enough, "compiling" is modern techno-babble. Answer: some software firms practice deliberate concealment of information. Since the open-source movement the clear division has become a bit shaky. Is an interpreter that can change the compiler on the fly (Forth) still an "interpreter"?

Reviews.

12

Fred Behringer

Fred reviews Vijgeblaadje Issues 37 and 38.

Advertisements.

13

Advertisements for the FIG-UK and the Dutch Forth-Users Group.

Signs of life from the USA.

13

Henry Vinerts
Volvovid@aol.com

Henry has so far written over 50 reports about the meetings of the SVFIG for Vierte Dimension. He's wondering if readers are at all interested and asks for feedback. He is toying with the idea of giving up writing the reports.

Updating the VD title-list.

14

Fred Behringer

direktorium@forth-ev.de

Fred's list of all articles that have ever appeared in Vierte Dimension was published in Issues 2/2002 to 4/2002. It is arranged according to subject-groups and within these by Issue number (date of appearance), and was modelled on Jenny Brien's Forthwrite list. The update presented herewith contains Issues 2/2002 to 2/2003 inclusive.

The LINUX Event.

16

Carsten Strotmann

The Forth-Gesellschaft was represented by a stand at this year's Linux conference in Karlsruhe. Forth and Linux have many things in common, and there are enough Forth systems that run under Linux. The Forth stand was organized by Carsten Strotmann. Also in the party at the stand (for four days) were Bernd Paysan, Ewald Rieger, Thomas Prinz and Holger Petersen.

Presentations were made of the Triceps robot of Ewald and the b16 processor of Bernd.

pOOP in Forth.

17

Manfred Mahlow

pOOP stands for "Prelude-based object-oriented programming". The author presented the Prelude concept for the first time in 1997 at the AGM of the Forth-Gesellschaft and has worked on its further development since then. Object and Methods are combined in a simple manner through implicit context switching. A few new Forth words are enough to provide every existing Forth system with Prelude. Characteristics: encapsulation of objects and methods, single inheritance, operator overloading, early binding, late binding possible. The author provides three pages of examples.

MicroCore Philosophy.

21

Klaus Schleisiek

Talk by the author at the 2003 AGM of the Forth-Gesellschaft. MicroCore is a processor structure that uses Forth as assembler. Among other things, borrowings are made from transputer architecture, by which longer literals are produced through stringing "nibbles" together

Living Forth.

22

Henry Vinerts &
Friederich Prinz

Pessimistic view by Henry, and upbeat reply by Friederich, concerning Henry's reports on the SVFIG meetings.

What is Peg Solitaire?

23

Ewald Rieger

The author discusses the background to the game: historical, winning strategies, evaluation of moves, the mathematics of solution, computerising, time-estimating, heuristics.

Triceps plays Solitaire.

25

Ewald Rieger

The author has introduced his robot "Triceps" in *Vierte Dimension* 2/2003. Triceps can master pick-and-place tasks and is programmed in Forth. In the present article the programme and the robot are developed to play move by move in a game of solitaire; iron balls lie in little recesses on an aluminium board and are moved from point to point by means of a programme-driven electromagnet hanging from a suspension device over the board.

Reviews.

30

Fred Behringer

Fred reviews *Vijgeblaadje* Issue 39.

Errata

Vierte Dimension

In *Forthwrite* Issue 123 Joe Anderson's review of *Vierte Dimension* was incorrectly entitled: "*Vierte Dimension 4/2002*". It should of course have been "*Vierte Dimension 2/2003*".

The Editor apologises to the readers, Joe and the editors of *Vierte Dimension* for this error.

The electronic copy of *Forthwrite* which will be posted on the web in due course will show the correct date.

AGM Report

The Editor also offers his apologies to Jeremy Fowell and Chris Jakeman for publishing the wrong text for the 2003 FIG UK AGM report. Due to an oversight, Chris's minutes were published and not Jeremy's final version of the report. As one would expect of a well-oiled machine like the FIG UK committee, the material content of the two is entirely consistent so it is felt that there is no need to retract Chris's version and print Jeremy's one instead.

– *Editor.*



Dutch Forth Users Group

Reading Dutch is easier than you might think. And as Forth is an international language, reading Dutch code is easier still for a Forth enthusiast. Are you interested? Why not subscribe to

HCC-Forth-gebruikersgroep

For only 10 euros a year (about £6.70), we will send you 5 to 6 copies of our "fig-leaf" broadsheet 'Het Vijgeblaadje'. This includes all our activities, progress reports on software and hardware projects and news of our in-house products.

To join, contact our Chairman:

Willem Ouwerkerk

Boulevard Heuvelink 126

6828 KW Arnhem, The Netherlands

E-Mail: w.ouwerkerk@kader.hobby.nl

The easiest way to pay is to post a 10 euro note direct to Willem.

Forthwrite Index

Jenny Brien maintains a set of three indexes to Forthwrite on the FIG UK web site and updates them with each new issue. These indexes are sorted by date, by author and by subject going back to 1990. The subject index is published in the magazine annually (below), with this year's new entries highlighted.

Back issues of Forthwrite are available from the Library, so this is a good way to catch up on topics of special interest. Copies of Forthwrite may be borrowed, just like the library books, for the cost of the postage, but many people prefer to receive scanned articles by email, or photocopies. Contact the Librarian for details.

If you spot a topic that has not been adequately covered, please drop a line to the Editor.

Forthwrite Subject Index 1990-2003

Subject	Author	Issue	Date	Title
algorithms	Hersom, Ed	68	Oct '92	Advanced course
algorithms	Charlton, Gordon	71	Apr '93	Backwards (psychic programming)
algorithms	Hersom, Ed	71	Apr '93	Trees & splines
algorithms	Hill, Will	72	Jun '93	Solving with Newton-Raphson
algorithms	Payne, John	75	Dec '93	Approximate pattern matching
algorithms	Bennett, Paul	78	Jun '94	Fuzz, fibs and forms
algorithms	Pochin, David	80	Oct '94	First attempts at Fuzzy Logic
algorithms	Bennett, Paul	84	Jun '95	Fractionally angular
algorithms	Charlton, Gordon	84	Jun '95	Easter Sunday
algorithms	Ramsay, Chris	103	Aug '99	Forth and Genetic Programming
algorithms	Jakeman, Chris	119	Jan '03	Word Completion for Quikwriter Project
applications	Green, Roedy	55	Aug '90	Abundance (database)
applications	Brien, Jack	58	Feb '91	Typing tutor (code)
applications	Kendall, Les	58	Feb '91	Terminal emulator for PC (code)
applications	Smith, Graham	58	Feb '91	Logic gates
applications	Grey, Nigel	60	Jun '91	Big Blue on the move IBM CAD (review)
applications	Franin, Julio	67	Aug '92	Torsion measurement system
applications	Stephens, Chris	73	Aug '93	Seven thousand networked micros
applications	Anderson, Joe	97	Jul '98	Forth In Space
applications	Trueblood, Mike	104	Nov '99	Radio Clock
applications	Bennett, Paul	108	Aug '00	Logging on - statistically speaking
applications	Paysan, Bernd	108	Aug '00	A Web-Server in Forth
applications	Kendall, Les	110	Jan '01	XML and Forth
applications	Matthews, John	110	Jan '01	Forth as Preferred Development Environment
applications	Wong, Leo	111	Apr '01	Solving a Riddle
applications	Anderson, Joe	112	Jul '01	Forth for NEAR Spacecraft
applications	Brien, Jenny	112	Jul '01	"Quikwriter" proposal
applications	Fowell, Jeremy	113	Sep '01	"Quikwriter" Project Launch
applications	Brien, Jenny	115	Jan '02	JenX revisited - A Simple XML Parser

applications	Brien, Jenny	116	Apr '02	Flickwriter Project
applications	Paysan, Bernd	118	Sep '02	Competitive Programming
arithmetic	Jakeman, Chris	57	Dec '90	A high-level /MOD (code)
arithmetic	Preston, Philip	58	Feb '91	Multi-cell arithmetic (code)
arithmetic	Filbey, Gil	59	Apr '91	Tutorial
arithmetic	Haley, Andrew	59	Apr '91	Function approx. by Chebyshev series
arithmetic	Filbey, Gil	63	Dec '91	Mixed point arithmetic (tutorial)
arithmetic	Payne, John	63	Dec '91	Fixed point arithmetic (word set)
arithmetic	Filbey, Gil	64	Feb '92	Mixed point arithmetic (tutorial)
arithmetic	Filbey, Gil	65	Apr '92	Mixed point arithmetic (tutorial)
arithmetic	Brown, Jack	68	Oct '92	Floored v symmetric division (tutorial)
arithmetic	Filbey, Gil	70	Feb '93	Floating point
arithmetic	Filbey, Gil	82	Feb '95	Cube roots
arithmetic	Bennett, Paul	91	Feb '97	From the 'Net - Square Roots (code)
arithmetic	Hersom, Ed	97	Jul '98	Quad (Fixed-Point) Arithmetic
arithmetic	Behringer, Fred	106	Apr '00	32-bit GCD without Division
arithmetic	Pochin, Dave	107	Jun '00	Floating Decimal Fudge
arithmetic	Jakeman, Chris	118	Sep '02	Linear Interpolation
arrays	Jakeman, Chris	55	Aug '90	Arrays and records (code)
arrays	Brien, Jack	64	Feb '92	Ways with arrays (code)
assembly	Tanner, P.	88	May '96	Linking machine code modules with Forth
block tools	Filbey, Gil	58	Feb '91	Bits and loading blocks (tutorial)
block tools	Hainsworth, Chris	58	Feb '91	Editing blocks (tutorial)
block tools	Charlton, Gordon	77	Apr '94	One-screen library load (code)
bons mots	Bezemer, Hans	92	Aug '97	Th
bons mots	Eckert, Brad	92	Aug '97	On Off On? Off?
bons mots	Luke, Gary	92	Aug '97	Tally
bons mots	Hersom, Ed	93	Nov '97	NVars [H] [D]
bons mots	Payne, John	93	Nov '97	3rd Swap@ Sgn #>ASCII
bons mots	Wenham, Alan	93	Nov '97	Z
bons mots	Elvey, Dwight	94	Jan '98	Setting bits with MASK
bons mots	Wenham, Alan	94	Jan '98	Printing binary with .SB U1B. U2B.
bons mots	Hoyt, Ben	95	Mar '98	PLACE is to COUNT as ! is to @
bons mots	van Norman, Rick	95	Mar '98	MANY for debugging
bons mots	Wong, Leo	96	May '98	Laying down values with COURSE
concurrency	Charlton, Gordon	62	Oct '91	Co-routine monitors (code)
concurrency	Charlton, Gordon	77	Apr '94	One-screen concurrent Forth (code)
control flow	Charlton, Gordon	53	Apr '90	Universal delimiter (code)
control flow	Brien, Jack	58	Feb '91	Extended ANS structures (F83 code)
control flow	Bennett, Paul	59	Apr '91	High level FOR..NEXT (code)
control flow	Carpenter, R.H.S.	69	Dec '92	Flow-charting method
control flow	Preston, Philip	72	Jun '93	Shortcuts and drop-outs
control flow	Brien, Jack	78	Jun '94	Extending ANSI control structures
control flow	Brien, Jack	84	Jun '95	Portable control structures
control flow	Charlton, Gordon	84	Jun '95	Trouble with DO
control flow	Jakeman, Chris	88	May '96	If and begin - ANS style
database	Filbey, Gil	61	Jun '91	FIG UK database (tutorial)
database	Filbey, Gil	61	Jun '91	FIG UK database (tutorial)
design	Payne, John	57	Dec '90	Simpler Forth (comment)
design	Brien, Jack	62	Oct '91	Return stack ENTER ISNOW and aliasing

design	Thomas, Reuben	66	Jun '92	Forth lifestyle
design	Hersom, Ed	68	Oct '92	NVARS
design	Charlton, Gordon	71	Apr '93	Upside down
design	Smart, Mike	74	Oct '93	Computer Shopper Programmer's Challenge
design	Matthews, John	76	Feb '94	On his September lecture
design	Bennett, Paul	79	Aug '94	Taking exception ...
design	Hersom, Ed	79	Aug '94	Simple user interface
design	Flynn, Chris	80	Oct '94	Numerical input
design	Allwright, R.E.	84	Jun '95	Pagination
design	Jakeman, Chris	84	Jun '95	From the 'net
design	Telfer, Graham	88	May '96	The specification method hunt
design	Brien, Jack	100	Jan '99	Working with Wordlists
design	Brien, Jack	102	Jun '99	Handling Literals
design	Telfer, Graham	102	Jun '99	Skeletons - Designing a Recursive Application
design	Telfer, Graham	117	Jul '02	Expanding the Use of the Stack
dynamic data	Charlton, Gordon	53	Apr '90	Dynamic words (code)
dynamic data	Charlton, Gordon	78	Jun '94	Work, rest and play
editing tools	Jakeman, Chris	52	Feb '90	Search and replace 1/2 (code)
editing tools	Jakeman, Chris	53	Apr '90	Search and replace 2/2 (code)
editing tools	Lake, Mike	58	Feb '91	Full screen editor in one screen (code)
editing tools	Brien, Jack	84	Jun '95	Full screen editor
editorial	Hainsworth, Chris	59	Apr '91	Forthtalk and EuroFORML report
editorial	Jakeman, Chris	67	Aug '92	Soapbox - "Do it yourself"
editorial	Payne, John	69	Dec '92	Fat, thin or inflatable?
editorial	Wilson, R.J.	72	Jun '93	Seeing trees in the wood
editorial	Rush, Peter	82	Feb '95	Honeywell Forth Bulletin Board
editorial	Jakeman, Chris	88	May '96	From the 'net - perceptions
editorial	Hersom, Ed	89	Jul '96	Why Forth?
editorial	Jakeman, Chris	90	Nov '96	Sell-by-date
editorial	Jakeman, Chris	91	Feb '97	FIG UK joins the World Wide Web
editorial	Jakeman, Chris	91	Feb '97	Welcome Disk
editorial	Brien, Jack	92	Aug '97	FIG UK Web Site
encryption	Greenwood, Mike	95	Mar '98	File Encryption
exceptions	Charlton, Gordon	59	Apr '91	CATCH and THROW (code)
exceptions	Jakeman, Chris	74	Oct '93	Portable CATCH and QUIT (code)
exceptions	Jakeman, Chris	74	Oct '93	Using CATCH and QUIT (code)
F11-UK	Fowell, Jeremy	122	Sep '03	What's All This Compiler Stuff, Anyway?
F11-UK	Fowell, Jeremy	122	Sep '03	Connecting an LCD to the F11-UK Single Board Computer
FANSI project	Bennett, Paul	54	Jun '90	Time for a new FIG Forth (comment)
FANSI project	Charlton, Gordon	56	Oct '90	High-level /MOD using recursion (code)
FANSI project	Charlton, Gordon	56	Oct '90	High-level multiply (code)
FANSI project	Flynn, Chris	56	Oct '90	Discussion on REQUIRES
FANSI project	Hainsworth, Chris	56	Oct '90	FANSI that (proposal)
FANSI project	Bennett, Paul	57	Dec '90	FANSI environs (proposal)
FANSI project	Flynn, Chris	57	Dec '90	Response to design proposals (comment)
FANSI project	Payne, John	57	Dec '90	Response to design proposals (comment)
FANSI project	Charlton, Gordon	60	Jun '91	FANSI definitions (code)
FANSI project	Charlton, Gordon	61	Jun '91	FANSI bloomers (code)
FANSI project	Payne, John	61	Jun '91	Notes on FANSI (code)

FANSI project	Bennett, Paul	62	Oct '91	Report on FANSI
FANSI project	Charlton, Gordon	63	Dec '91	FANSI vocabularies (proposal)
FANSI project	Brien, Jack	64	Feb '92	FANSI (comment)
FANSI project	Payne, John	64	Feb '92	FANSI (comment)
FANSI project	Preston, Philip	64	Feb '92	FANSI (comment)
FANSI project	Payne, John	69	Dec '92	FANSI QUIT
file tools	Brien, Jack	58	Feb '91	Loading dependant source (code)
file tools	Jakeman, Chris	70	Feb '93	File access, part 1 (code)
file tools	Jakeman, Chris	71	Apr '93	File access, part 2 (code)
file tools	Jakeman, Chris	72	Jun '93	File access, part 3 (code)
file tools	Jakeman, Chris	73	Aug '93	File access, part 4 (code)
file tools	Brien, Jack	86	Oct '95	Hierarchical screen filing
file tools	Wong, Leo	98	Oct '98	ANS File Words for Pygmy Forth
file tools	Behringer, Fred	100	Jan '99	ANS File Words for Turbo Forth - 1
fractions	Charlton, Gordon	52	Feb '90	Vulgar words (code)
fractions	Wilson, R.J.	53	Apr '90	Rational numbers (code)
fractions	Wilson, R.J.	54	Jun '90	Transcendental rationale (code)
fractions	Charlton, Gordon	56	Oct '90	Rational approximation (comment)
futures	Jakeman, Chris	79	Aug '94	Telescript (comment)
futures	Jakeman, Chris	80	Oct '94	Some future directions (editorial)
futures	Jakeman, Chris	90	Nov '96	Forth and Java (comp.lang.forth)
futures	Pelc, Stephen	104	Nov '99	FIG UK - The Next 20 Years
futures	Jakeman, Chris	115	Jan '02	The Semantic Web
graphics	Filbey, Gil	53	Apr '90	Plotting spirals (tutorial)
graphics	Charlton, Gordon	66	Jun '92	Turtle graphics
graphics	Payne, John	67	Aug '92	Flood fill
graphics	Charlton, Gordon	73	Aug '93	Drawing a line
graphics	Charlton, Gordon	74	Oct '93	Not drawing a line
graphics	Payne, John	74	Oct '93	How Bresenham's line drawing alg. works
graphics	Pochin, Dave	109	Nov '00	"BLT is not a Sandwich"
graphics	Pochin, Dave	119	Jan '03	Rectangles in Win32Forth
hardware	Koopman, Philip	56	Oct '90	RTX 4000 (publicity)
hardware	Fowell, Jeremy	67	Aug '92	P20 chip, part 1/2
hardware	Fowell, Jeremy	68	Oct '92	P20 chip, part 2/2
hardware	Bennett, Paul	89	Jul '96	Chuck's chips
hardware	Fowell, Jeremy	100	Jan '99	FIG UK Hardware Project
hardware	Fowell, Jeremy	101	Apr '99	FIG UK Hardware Project - Progress
hardware	Heuvel, Leendert	101	Apr '99	The 'Egel Coursebook
hardware	Fowell, Jeremy	103	Aug '99	FIG UK Hardware Project - Progress
hardware	Fowell, Jeremy	104	Nov '99	FIG UK Hardware Project - Progress
hardware	Fowell, Jeremy	105	Jan '00	F11-UK Hardware Project - Progress
hardware	Fowell, Jeremy	106	Apr '00	F11-UK Hardware Project - Progress
hardware	Fowell, Jeremy	108	Aug '00	F11-UK Hardware Project - Launch
hardware	Jakeman, Chris	110	Jan '01	F11-UK Hardware Project - Progress
hardware	Jakeman, Chris	111	Apr '01	F11-UK Hardware Project - Progress
history	Rather, Elizabeth	83	Apr '95	The evolution of Forth
history	Rather, Elizabeth	87	Dec '95	The Forth approach to operating systems
history	Hainsworth, Chris	100	Jan '99	Forthwrite Issue No. 1 revisited
history	Powell, Bill	100	Jan '99	The Birth of FIG UK
history	Behringer, Fred	104	Nov '99	Swap Dragon

history	Brien, Jack	104	Nov '99	FIG UK - The Last 20 Years
history	Jakeman, Chris	105	Jan '00	Did you Know? - EasyWriter
history	Jakeman, Chris	106	Apr '00	From the 'Net, Forth for Novell
history	Crook, Neal	107	Jun '00	The Canon Cat
history	Jakeman, Chris	107	Jun '00	Did you Know? - Forth OS
history	Jakeman, Chris	108	Aug '00	Computer Conservation
history	Jakeman, Chris	108	Aug '00	Did you Know? - Forth v C
history	Jakeman, Chris	109	Nov '00	Did you Know? - Open Firmware
history	Jakeman, Chris	113	Sep '01	Did you Know? - smart cards
history	Jakeman, Chris	114	Nov '01	Did you Know? - large Forth projects
history	Jakeman, Chris	116	Apr '02	Did you Know? - Forth Help Nobel Prize Winners
history	Moore, Charles	118	Sep '02	Forth - The Early Years
humour	Payne, John	57	Dec '90	A program that works the French way
humour	Smith, Graham	84	Jun '95	Book titles
humour	Jakeman, Chris	88	May '96	From the 'net - a drinking song
humour	Allwright, Ray	96	May '98	A Story of Cowboys
humour	Gassanenko, Michael	115	Jan '02	From the 'Net - the non-English view
humour	anon	122	Sep '03	What Languages Fix
humour	anon	123	Dec '03	What Languages Fix - Not!
interfacing	Robinson, Dave	61	Jun '91	Mouse handling (F83 code)
interfacing	Bennett, Paul	96	May '98	Reading the World - 1
interfacing	Bennett, Paul	97	Jul '98	Reading the World - 2
interfacing	Bennett, Paul	98	Oct '98	Writing the World - 1
interfacing	Bennett, Paul	100	Jan '99	Writing the World - 2
internals	Hainsworth, Chris	52	Feb '90	Kiss and run (exploring F-PC)
internals	Charlton, Gordon	58	Feb '91	A replacement for DO .. LOOP (code)
internals	Flynn, Chris	60	Jun '91	Forth engine on 68000
internals	Bennett, Paul	68	Oct '92	Top-down development of a Forth system
internals	Bennett, Paul	71	Apr '93	QUIT, the story continues...
internals	Preston, Philip	75	Dec '93	RatForth - ANS on F83
internals	Preston, Philip	76	Feb '94	Ratforth revised etc.
internals	Preston, Philip	78	Jun '94	Redefining colon
internals	Preston, Philip	80	Oct '94	Simulating EVALUATE
internals	Preston, Philip	86	Oct '95	Variables, values & locals
internals	Wenham, Alan	87	Dec '95	Meandering Forth
internals	Brien, Jack	92	Aug '97	Building a new inner interpreter
internals	Allwright, Ray	95	Mar '98	From the 'Net - Minimal word sets
internals	Allwright, Ray	101	Apr '99	From the 'Net - Turnkey Apps and Docs
internals	Tasgal, John	106	Apr '00	An Introduction to Machine Forth
internals	Brien, Jenny	113	Sep '01	Treating Data as Source
interpreting	Jakeman, Chris	86	Oct '95	From the 'net - text interpreter
interpreting	Brien, Jack	90	Nov '96	Implementing an outer interpreter
interview	Moore, Charles	102	Jun '99	1xForth
interview	Lawless, Jim	114	Nov '01	An interview with Tom Zimmer
interview	Morrison, George	114	Nov '01	Charles Moore interview on Slashdot
interview	Culver, Barry	119	Jan '03	An interview with Barry Culver
library	Hainsworth, Sylvia	59	Apr '91	FIG UK library bulletin
library	Jakeman, Chris	90	Nov '96	Library assets
library	Hainsworth, Sylvia	96	May '98	Purchases and current publications

logic	Behringer, Fred	112	Jul '01	Arithmetized Logic in Forth
MCFAs	Brien, Jack	55	Aug '90	Comment
objects	Jakeman, Chris	81	Dec '94	Objects and so forth
objects	Jakeman, Chris	99	Nov '98	OOF - A Minimal Approach
objects	Prinz, Friederich	100	Jan '99	Counting Fruits the Classic Way
objects	Jakeman, Chris	115	Jan '02	A Safer Mini-OOF
performance	Jakeman, Chris	94	Jan '98	From the 'Net - Speed Demons
permutations	Charlton, Gordon	52	Feb '90	Permutations, a new algorithm (code)
permutations	Hersom, Ed	62	Oct '91	Permutations (code)
permutations	Hersom, Ed	65	Apr '92	Permutations/combinations
permutations	Baden, Wil	109	Nov '00	Permutation by Transposition Sequence ACM 115A
permutations	Jakeman, Chris	109	Nov '00	Simple Forth Permutations
permutations	Behringer, Fred	111	Apr '01	Generating Combinations
presentation	Brien, Jack	52	Feb '90	Locals and more (discussion)
presentation	Matthews, Keith	57	Dec '90	Stack diagrams (explored)
presentation	Brien, Jack	58	Feb '91	GIST for indexing source (code)
presentation	Bennett, Paul	60	Jun '91	Manual documentation (code)
presentation	Charlton, Gordon	75	Dec '93	StackFlow
presentation	Brien, Jack	80	Oct '94	Readable Forth
presentation	Tanner, P.H.	81	Dec '94	Post indentation
presentation	Charlton, Gordon	91	Feb '97	From the 'Net - StackFlow
probability	Filbey, Gil	57	Dec '90	Probability of common birthdays
probability	Filbey, Gil	57	Dec '90	Random thoughts on probability
probability	Payne, John	57	Dec '90	Probability of common birthdays
programming	Brien, Jenny & Jakeman, Chris	119	Jan '03	Using Wordlists for Many[
programming	Jakeman, Chris	119	Jan '03	From the 'Net
programming	Jakeman, Chris	120	Mar '03	Sorting a List
programming	Boyd, James A	123	Dec '03	A Virtual Nondeterministic Machine in Forth
publications	Haley, Andrew	63	Dec '91	FORML 87, 88 & 89 (review)
puzzles	Hainsworth, Chris	54	Jun '90	Forth brain teasers
puzzles	Charlton, Gordon	57	Dec '90	Name that word
puzzles	Charlton, Gordon	58	Feb '91	Puzzle answers (code)
puzzles	Filbey, Gil	68	Oct '92	Tethered goat puzzle, part 1/2
puzzles	Filbey, Gil	68	Oct '92	Tethered goat puzzle, part 2/2
random nos.	Filbey, Gil	72	Jun '93	Visualising random numbers on Apple II
random nos.	Jakeman, Chris	72	Jun '93	Random numbers
random nos.	Filbey, Gil	73	Aug '93	Testing for randomness
random nos.	Payne, John	73	Aug '93	More on random numbers
review	Charlton, Gordon	80	Oct '94	Riding the wild 'net
review	Charlton, Gordon	82	Feb '95	Report from EuroForth '94
review	Bennett, Paul	93	Nov '97	EuroForth '97 Conference
review	Wenham, Alan	94	Jan '98	Vierte Dimension
review	Fowell, Jeremy	96	May '98	Forth Programmers' Handbook
review	Jakeman, Chris	96	May '98	Genetix - The Inside Story
review	Payne, John	97	Jul '98	FORML Proceedings 94 & 95
review	Flynn, Chris	98	Oct '98	A Hard Day Garbage Collecting
review	Jakeman, Chris	98	Oct '98	jeForth
review	Bennett, Paul	99	Nov '98	euroForth '98 Conference
review	Wenham, Alan	100	Jan '99	Vierte Dimension

review	Anderson, Joe	102	Jun '99	Forth for Virtual Reality
review	Wenham, Alan	104	Nov '99	Vierte Dimension
review	Jakeman, Chris	105	Jan '00	FIG UK 20th Anniversary Reunion
review	Wenham, Alan	105	Jan '00	Vierte Dimension 4/99
review	de Ceballos, Federico	106	Apr '00	21st FORML Conference
review	Wenham, Alan	106	Apr '00	Vierte Dimension 1/00
review	Wenham, Alan	107	Jun '00	Vierte Dimension 2/00
review	Jakeman, Chris	108	Aug '00	euroForth '99 Conference
review	Jakeman, Chris	109	Nov '00	Forth in the UK
review	Wenham, Alan	109	Nov '00	Vierte Dimension 3/00
review	Ives, Robert	110	Jan '01	"Forth Application Techniques"
review	Oakford, Howerd	110	Jan '01	euroFORTH 2000 Conference report
review	Wenham, Alan	110	Jan '01	Vierte Dimension 4/00
review	Abrahams, David	112	Jul '01	"Extreme Mindstorms" book
review	Bennett, Paul	112	Jul '01	3 Free Forths and an OS too!
review	Jakeman, Chris	112	Jul '01	Gesellschaft 2001 Conference report
review	Wenham, Alan	113	Sep '01	Vierte Dimension 2/01
review	Wenham, Alan	114	Nov '01	Vierte Dimension 3/01
review	Vinerts, Henry	115	Jan '02	Across the Big Teich
review	Jakeman, Chris	116	Apr '02	From the 'Net
review	Oakford, Howerd	116	Apr '02	euroFORTH 2001 Conference report
review	Vinerts, Henry	116	Apr '02	Across the Big Teich
review	Wenham, Alan	116	Apr '02	Vierte Dimension 4/01
review	Behringer, Fred	117	Jul '02	German FIG Annual Conference
review	Fennema, Boris	117	Jul '02	"Write Your Own Programming Language Using C++"
review	Fennema, Boris	117	Jul '02	"The Practice of Programming"
review	Moore, Charles	117	Jul '02	An Interview with Chuck Moore
review	Vinerts, Henry	117	Jul '02	Across the Big Teich
review	Wenham, Alan	117	Jul '02	Vierte Dimension 1/02
review	Rodriguez, Brad	118	Sep '02	Choosing Forth
review	Vinerts, Henry	118	Sep '02	Across the Big Teich
review	Anderson, Joe	119	Jan '03	Vierte Dimension 3/2002
review	Jakeman, Chris	119	Jan '03	AGM Report
review	Stoddart, Bill	119	Jan '03	euroFORTH Conference Report
review	Vinerts, Henry	119	Jan '03	Across the Big Teich
review	anon	120	Mar '03	nnCron
review	anon	120	Mar '03	F11-UK FIG Hardware Project
review	Jakeman, Chris & Powell, Bill	120	Mar '03	Forth and the Neuron Chip
review	Vinerts, Henry	120	Mar '03	Across the Big Teich
review	Anderson, Joe	121	Jul '03	Vierte Dimension 4/2002
review	anon	121	Jul '03	Forth Archive
review	Jakeman, Chris	121	Jul '03	From the 'Net
review	Vinerts, Henry	121	Jul '03	Across the Big Teich
review	Anderson, Joe	122	Sep '03	Vierte Dimension 1/2003
review	Vinerts, Henry	122	Sep '03	Across the Big Teich
review	Anderson, Joe	123	Dec '03	Vierte Dimension
review	Jakeman, Chris	123	Dec '03	AGM Report

review	Oakford, Howerd	123	Dec '03	EuroForth 2003 - The Report
review	Vinerts, Henry	123	Dec '03	Across the Big Teich
roots	Wilson, R.J.	55	Aug '90	Root of rational numbers (code)
roots	Charlton, Gordon	56	Oct '90	Square root (code)
roots	Trapp, John	58	Feb '91	Square-roots for double/floating point
roots	Brien, Jack	93	Nov '97	From the Net - More on square roots
roots	Behringer, Fred	95	Mar '98	Square roots once more
roots	Behringer, Fred	96	May '98	Cubic roots without division
roots	Jakeman, Chris	106	Apr '00	Cube Roots Again
roots	Jakeman, Chris	106	Apr '00	From the 'Net - Cube Roots
roots	Jakeman, Chris	107	Jun '00	From the 'Net, Cube Roots
searching	Charlton, Gordon	57	Dec '90	A faster string search (code)
searching	Charlton, Gordon	62	Oct '91	A binary search (code)
searching	Hersom, Ed	63	Dec '91	Recursive BINSEARCH (code)
searching	Charlton, Gordon	70	Feb '93	Shift-AND string search (code)
searching	Charlton, Gordon	76	Feb '94	Best string search (code)
searching	Jakeman, Chris	84	Jun '95	Linear search
sets	Charlton, Gordon	54	Jun '90	Set manipulation (code)
sorting	Charlton, Gordon	55	Aug '90	Radix, an extravagant sort (code)
sorting	Charlton, Gordon	56	Oct '90	Sorting strings with qsort (code)
sorting	Charlton, Gordon	62	Oct '91	Heapsort (code)
stacks	Preston, Philip	69	Dec '92	Stacking fillers - stacks & write-only
stacks	Charlton, Gordon	77	Apr '94	Stacrobaticus exotica
stacks	Filbey, Gil	79	Aug '94	Stacks (tutorial)
stacks	Jakeman, Chris	85	Aug '95	Stack manipulation
stacks	Joseph, Neville	86	Oct '95	Stack manipulation
stacks	Barr, Stan	87	Dec '95	A third stack
stacks	Hersom, Ed	93	Nov '97	Multi-precision Stack Operators
standards	Jakeman, Chris	60	Jun '91	Portable code (code)
state machines	Charlton, Gordon	56	Oct '90	Variables for state machines (code)
state machines	Dunbar, Graeme	97	Jul '98	Finite State Machines 1/3
state machines	Dunbar, Graeme	98	Oct '98	Finite State Machines 2/3
state machines	Dunbar, Graeme	103	Aug '99	Finite State Machines 3a
strings	Leibniz, David	58	Feb '91	String stack routine (code)
strings	MacLean, Ruaridh	58	Feb '91	High level DIGIT (tutorial)
strings	Charlton, Gordon	59	Apr '91	A string pattern matcher (code)
strings	Payne, John	65	Apr '92	Text processing
strings	Preston, Philip	68	Oct '92	TACK and BLOCKL
strings	Charlton, Gordon	71	Apr '93	ANSI and portability - STRLIT (code)
strings	Brien, Jack	72	Jun '93	Comment on Blockl & Tack
strings	Charlton, Gordon	72	Jun '93	Similarity
strings	Jakeman, Chris	87	Dec '95	From the 'net - please
strings	Brien, Jack	89	Jul '96	String handling
strings	Jakeman, Chris	91	Feb '97	Pattern matching - 1/3 (tutorial)
strings	Jakeman, Chris	92	Aug '97	Pattern matching - 2/3 (FoSM with Forth)
strings	Jakeman, Chris	93	Nov '97	Pattern matching 3/3 (Rex)
strings	Borrell, Richard	95	Mar '98	Deferred Language Translation
strings	Oakford, Howerd	99	Nov '98	Multiple Language Programs Made Easy
structures	Brien, Jack	94	Jan '98	Building Forth Structures
systems	Green, Roedy	55	Aug '90	BBL Forth (review)

systems	Bennett, Paul	64	Feb '92	Pygmy Forth (review)
systems	Tanner, Philip	65	Apr '92	As in a glass darkly
systems	Hersom, Ed	70	Feb '93	Pocket Forth (review)
systems	Tanner, P.H.	72	Jun '93	URForth (review)
systems	Payne, John	82	Feb '95	A 32-bit Forth for Windows (review)
systems	Stephens, Chris	82	Feb '95	Forth for the Transputer (review)
systems	Behringer, Fred	92	Aug '97	Forth for the Transputer
systems	Worthington, Thom.	94	Jan '98	Aztec - A Forth For Windows '95
systems	Besemer, Hans	96	May '98	4th - The Alternative Compiler
systems	Jakeman, Chris	100	Jan '99	Web Forth Project
systems	Lancaster, Garry	101	Apr '99	Forth for the Z88
systems	Jakeman, Chris	102	Jun '99	Web Forth Project
systems	Ouwerkerk, Willem	103	Aug '99	ByteForth for MCS51 cpu's
systems	Tasgal, John	107	Jun '00	An Introduction to Color Forth
systems	Tasgal, John	107	Jun '00	The BMP Example
systems	Zimmer, Tom	113	Sep '01	4-bit Forth
systems	Eckert, Brad	114	Nov '01	Tiny Open Firmware
systems	Myneni, Krishna	116	Apr '02	Special Features of kForth 1/2
systems	Myneni, Krishna	117	Jul '02	Special Features of kForth 2/2
tools	Jakeman, Chris	54	Jun '90	Patch programming aid (code)
tools	Jakeman, Chris	56	Oct '90	Run-time operators (code)
tools	Preston, Philip	63	Dec '91	ALIAS ALIAS ALIAS (F83 code)
tools	Jakeman, Chris	69	Dec '92	Also and -Also (code)
tools	Charlton, Gordon	71	Apr '93	Wrong way round!
tools	Bennett, Paul	72	Jun '93	+MOD! (LOG?) and commenting words
tools	Brien, Jack	74	Oct '93	Utilities for F83 on Amstrad PCW
tools	Jakeman, Chris	75	Dec '93	Shell (code)
tools	Bennett, Paul	76	Feb '94	Spooling and browsing
tools	Jakeman, Chris	76	Feb '94	.Call and Assert (code)
tools	Jakeman, Chris	77	Apr '94	Check (code)
tools	Flynn, Chris	78	Jun '94	Conditional compilation
tools	Preston, Philip	79	Aug '94	More fun with EVALUATE
tools	Charlton, Gordon	81	Dec '94	16-bit cyclic redundancy checksums
tools	Franin, Julio	82	Feb '95	MC51 Forth debugging
tools	Smith, Graham	84	Jun '95	MARK
tools	Jakeman, Chris	85	Aug '95	Limit variables (code)
tools	Abrahams, David	86	Oct '95	General purpose utilities for F-PC
tools	Stott, Barrie	91	Feb '97	Stack checking (code)
tools	Jakeman, Chris	102	Jun '99	From the 'Net - Iterative Interpretation
tools	Wong, Leo	118	Sep '02	Iteration with Many:
tutorial	Charlton, Gordon	65	Apr '92	Two geese and a car
tutorial	Brown, Jack	66	Jun '92	An indefinite loop example
tutorial	Filbey, Gil	69	Dec '92	Escape codes and printing
tutorial	Filbey, Gil	70	Feb '93	A conjuring trick
tutorial	Hainsworth, Chris	70	Feb '93	Shallow end
tutorial	Filbey, Gil	71	Apr '93	Some old words revisited
tutorial	Filbey, Gil	74	Oct '93	Floating point
tutorial	Charlton, Gordon	75	Dec '93	Create .. does> ..
tutorial	Filbey, Gil	75	Dec '93	Postfix
tutorial	Filbey, Gil	76	Feb '94	Editorial & Tu

tutorial	Filbey, Gil	81	Dec '94	Floating point
tutorial	Filbey, Gil	85	Aug '95	Immediacy
tutorial	Filbey, Gil	86	Oct '95	Editorial
tutorial	Telfer, Graham	97	Jul '98	Wondrous Numbers
tutorial	Jakeman, Chris	99	Nov '98	jeForth Project
tutorial	Pochin, Dave	100	Jan '99	Forth for the New Year
tutorial	Pochin, Dave	100	Jan '99	Guide to Getting Started
tutorial	Pochin, Dave	101	Apr '99	Getting Stuck Into Win32Forth
tutorial	Pochin, Dave	103	Aug '99	Figuring it out with Win32Forth
tutorial	Jakeman, Chris	104	Nov '99	Clock Challenge
tutorial	Jakeman, Chris	105	Jan '00	Clock Challenge - 2nd installment
tutorial	Pochin, Dave	105	Jan '00	"See Win32Forth scroll the Window"
tutorial	Brien, Jack	106	Apr '00	All you need to know about STATE, IMMEDIATE and POSTPONE
tutorial	Pochin, Dave	111	Apr '01	Six Easy Fonts
tutorial	Noble, Julian	113	Sep '01	A Call to Assembly 1/3
tutorial	Pochin, Dave	113	Sep '01	Win32Forth Fonts
tutorial	Noble, Julian	114	Nov '01	A Call to Assembly 2/3
tutorial	Noble, Julian	115	Jan '02	A Call to Assembly 3/3
tutorial	Pochin, Dave	115	Jan '02	The End of the Line
tutorial	Telfer, Graham	116	Apr '02	Seven Times Five Equals Eleven
vectoring	Charlton, Gordon	56	Oct '90	Resolving forward references (code)
vectoring	Jakeman, Chris	58	Feb '91	Deferred words (code)
vectoring	Preston, Philip	59	Apr '91	Forgettable vectors and smart compiling
vectoring	Bennett, Paul	68	Oct '92	Vectoring with DOER and MAKE
vectoring	Allwright, Ray	93	Nov '97	From the Net - Defer and Is

What Languages Fix- Not!

The Editor's in tray is still decidedly empty on this topic...

Note that the original article in Issue 122 has been listed under "Humour" in the Forthwrite Index (see above), so nothing too deep or philosophical is expected by way of contributions from readers. A deep and philosophical treatise would be just as welcome, however!

Taking the evidence (or lack of it) in a Bayesian sense, one might be led to either of the following conclusions:

- (a) Forth fixes none of the problems exhibited by other languages, or,
- (b) Forth's superiority is so self-evident that no comment is necessary.

So let's see some input.

– *Editor.*



FIG UK Contacts and Information

Chairman	Jeremy Fowell,	11 Hitches Lane, EDGEBASTON B15 2LS 0121 440 1809 jeremy.fowell@btinternet.com
Secretary	Doug Neale,	58 Woodland Way, MORDEN SM4 4DS 020 8542 2747 dneale@w58wmorden.demon.co.uk
Editor (temporary)	Graeme Dunbar,	School of Engineering, The Robert Gordon University, Schoolhill, ABERDEEN AB10 1FR 01224 262415 g.r.a.dunbar@rgu.ac.uk
Treasurer	Neville Joseph,	Marlowe House, Hale Road, WENDOVER HP22 6NE 01296 62 3167 naj@najoseph.demon.co.uk
Webmaster	Jenny Brien,	Windy Hill, Drumkeen, BALLINAMALLARD, Co. Fermanagh BT94 2HJ 02866 388 253 webmaster@figuk.plus.com
Librarian	Graeme Dunbar,	School of Engineering, The Robert Gordon University, Schoolhill, ABERDEEN AB10 1FR 01224 262415 g.r.a.dunbar@rgu.ac.uk

Membership enquiries, renewals and changes of address to Doug.

Technical enquiries and anything for publication to Graeme.

Borrowing requests for books, magazines and proceedings to Graeme.

FIG UK Web Site

For indexes to Forthwrite, the FIG UK Library and much more, see <http://www.fig-uk.org>

FIG UK Membership

Payment entitles you to 6 issues of Forthwrite magazine and our membership services for that period (about a

year). Fees are:

National and international	£12
International served by airmail	£22
Corporate	£36 (3 copies of each issue)

Forthwrite Deliveries

Your membership number appears on your envelope label. Please quote it in correspondence to us. Look out for the message "SUBS NOW DUE" on your sixth and last issue and please complete the renewal form enclosed. Overseas members can opt to pay the higher price for airmail delivery.

Copyright

Copyright of each individual article rests with its author. Publication implies permission for FIG UK to reproduce the material in a variety of forms and media including through the Internet.

FIG UK Services to Members



- Magazine** Forthwrite is our regular magazine, which has been in publication for over 100 issues. Most of the contributions come from our own members and Graeme Dunbar, the Editor, is always ready to assist new authors wishing to share their experiences of the Forth world.
- Library** Our library provides a service unmatched by any other FIG chapter. Not only are all the major books available, but also conference proceedings, back-issues of Forthwrite and also of the magazine of International FIG, Forth Dimensions. The price of a loan is simply the cost of postage out and back.
- Web Site** Jenny Brien maintains our web site at <http://www.fig-uk.org>. She publishes details of FIG UK projects, a regularly-updated Forth News report, indexes to the Forthwrite magazine and the library as well as specialist contributions such as “Build Your Own Forth” and links to other sites. Don’t forget to check out the “FIG UK Hall of Fame”.
- IRC** Software for accessing Internet Relay Chat is free and easy to use. FIG UK members (and a few others too) get together on the #FIG UK channel every month. Check Forthwrite for details.
- Members** The members are our greatest asset. If you have a problem, don’t struggle in silence - someone will always be able to help. Do consider joining one of our joint projects. Undertaken by informal groups of members, these are very successful and an excellent way to gain both experience and good friends.
- Beyond the UK** FIG UK has links with International FIG, the German Forth-Gesellschaft and the Dutch Forth Users Group. Some of our members have multiple memberships and we report progress and special events. FIG UK has attracted a core of overseas members; please ask if you want an accelerated postal delivery for your Forthwrite.

Copy deadlines:

Issue 125: 24 March 2004

Issue 126: 2 June 2004

All material for publication to the Editor by email or post by that date please. Plain text, MS Word or Rich Text format preferred.

Back cover (Advert)