Subject: Re: Introducing FL
Posted by biophys on Sat, 01 Apr 2006 19:23:55 GMT
View Forum Message <> Reply to Message

good job, Lajos! It does run as fast as idl6.2 with some of my old code. i've noticed backingstore problems with my linux machine. device, retain=2 does not work. i'd like to know how to set working dir/search path/start_up file.

cheers,
bp

side by side comparison of time_test2(incomplete, see below):

############ IDL ############
IDL> time_test2
% Compiled module: TIME_TEST2.
% Compiled module: TIME_TEST.
|TIME_TEST2 performance for IDL 6.2:
| OS_FAMILY=unix, OS=linux, ARCH=x86
| Sat Apr 1 11:11:07 2006
  1 0.0301080 Empty For loop, 2000000 times
  2 0.0234098 Call empty procedure (1 param) 100,000 times
  3 0.0129421 Add 100,000 integer scalars and store
  4 0.0131989 25,000 scalar loops each of 5 ops, 2 =, 1 if)
  5 0.00566292 Mult 512 by 512 byte by constant and store, 10 times
  6 0.0338290 Shift 512 by 512 byte and store, 100 times
  7 0.0265679 Add constant to 512 x 512 byte array and store, 50 times
  8 0.0145931 Add two 512 by 512 byte images and store, 30 times
  9 0.0347710 Mult 512 by 512 floating by constant and store, 30 times
 10 0.0317931 Add constant to 512 x 512 floating and store, 40 times
 11 0.0483379 Add two 512 by 512 floating images and store, 30 times
 12 0.00934505 Generate 225000 random numbers
 13 0.0171521 Invert a 150 by 150 random matrix
 14 0.00771499 LU Decomposition of a 150 by 150 random matrix
 15 0.0133338 Transpose 256 x 256 byte, FOR loops
 16 0.0229912 Transpose 256 x 256 byte, row and column ops x 10
 17 0.00440502 Transpose 256 x 256 byte, TRANSPOSE function x 10
 18 0.0311699 Log of 100,000 numbers, FOR loop
 19 0.00604892 Log of 100,000 numbers, vector ops
 20 0.0194640 131072 point forward plus inverse FFT
 21 0.0228901 Smooth 512 by 512 byte array, 5x5 boxcar, 10

```
times
      22   0.00685596 Smooth 512 by 512 floating array, 5x5 boxcar, 2
times
% Compiled module: FILEPATH.
      23    0.0223649 Write and read 512 by 512 byte array x 20
     0.458950=Total Time,      0.016516175=Geometric mean,      23
tests.
############ IDL ############


########## FL #################
FL> cd, "/usr/local/rsi/idl/lib"
FL> time_test2
% Compiled routine: TIME_TEST2
% Compiled routine: TIME_TEST_TIMER
% Compiled routine: TIME_TEST_INIT
% Compiled routine: TIME_COMPARE
% Compiled routine: TIME_TEST_RESET
% Compiled routine: TIME_TEST_DUMMY
% Compiled routine: GRAPHICS_TIMES2_INTERNAL
% Compiled routine: GRAPHICS_TIMES3_INTERNAL
% Compiled routine: TIME_TEST2_INTERNAL
% Compiled routine: TIME_TEST3_INTERNAL
% Compiled routine: GRAPHICS_TIMES_INTERNAL
% Compiled routine: TIME_TEST
% Warning: N_PARAMS: extra arguments ignored
|TIME_TEST2 performance for IDL 0.6:
|       OS_FAMILY=unix, OS=linux, ARCH=x86
|       Sat Apr  1 11:10:29 2006

       1    0.0307741 Empty For loop,     2000000 times
       2    0.0128050 Call empty procedure (1 param) 100,000 times
       3   0.00270391 Add 100,000 integer scalars and store
       4    0.0155468 25,000 scalar loops each of 5 ops, 2 =, 1 if)
       5   0.00199509 Mult 512 by 512 byte by constant and store, 10
times
       6    0.0217750 Shift 512 by 512 byte and store, 100 times
       7   0.00250912 Add constant to 512 x 512 byte array and store,
50 times
       8   0.00150394 Add two 512 by 512 byte images and store, 30
times
       9    0.0279901 Mult 512 by 512 floating by constant and store,
30 times
      10    0.0378730 Add constant to 512 x 512 floating and store, 40
times
      11    0.0412631 Add two 512 by 512 floating images and store, 30
times
      12   0.00127006 Generate 225000 random numbers
```

```
      13   0.00448203 Invert a 150 by 150 random matrix
      14    0.0115409 LU Decomposition of a 150 by 150 random matrix
      15    0.0201399 Transpose 256 x 256 byte, FOR loops
      16   0.00537109 Transpose 256 x 256 byte, row and column ops x 10
      17   0.00193810 Transpose 256 x 256 byte, TRANSPOSE function x 10
      18    0.0396800 Log of 100,000 numbers, FOR loop
      19   0.00554919 Log of 100,000 numbers, vector ops
      20    0.0392570       131072 point forward plus inverse FFT
      21    0.0368030 Smooth 512 by 512 byte array, 5x5 boxcar, 10
times
      22   0.00418997 Smooth 512 by 512 floating array, 5x5 boxcar, 2
times
% Compiled routine: FILEPATH
% Warning: OPENW: extra arguments ignored
% Error: OPENW: can not open file: test.dat
% Error occured at:
%   TIME_TEST2_INTERNAL       491
/usr/local/rsi/idl_6.2/lib/time_test.pro
%   TIME_TEST2                24
/usr/local/rsi/idl_6.2/lib/time_test2.pro
%   $MAIN$
% Execution halted at:
%   TIME_TEST2        24  /usr/local/rsi/idl_6.2/lib/time_test2.pro
%   $MAIN$
FL> device, retain=2
% Error: DEVICE: keyword not allowed: RETAIN
% Execution halted at:
%   TIME_TEST2        24  /usr/local/rsi/idl_6.2/lib/time_test2.pro
%   $MAIN$


########## FL #################
```

FL wrote:

> Hi guys!
>
> The first public snapshot of Fawlty Language, an IDL compatible compiler
> is available. (IDL is a registered trademark of Research Systems Inc.)
>
> FL binaries can be freely used and downloaded from the homepage
>

> http://web.interware.hu/fl
>
>
> Available packages:
>
> fl_0.6-i686-linux.tar.gz (Linux, Pentium II.
> kernel 2.4/2.6, glibc >= 2.2.5)
>
> fl_0.6-amd64-linux.tar.gz (Linux, AMD Athlon64,
> kernel 2.6, glibc >= 2.3.3)
>
> and for the bravehearted:
>
> fl_0.6-i686-mingw.zip (Windows 2000/XP/?, Pentium II)
>
>
> Features:
>
> - all language elements are supported
>
> - multithreaded operators
>
> - array operations use MMX/SSE/SSE2, if available
>
> - module profiling
>
> - line profiling
>
> - about 280 library functions (more or less usable)
>
> - true-color (24 bit) direct graphics devices: X, WIN, PS, PDF, Z
>
> - run-time performance: for many programs, FL is faster than IDL
> (eg. the empty loop is three times faster in FL :-)
>
>
> Enjoy!
>
> fl
> (Földy Lajos)
>
>
> ps: Hungarian names are in reverse order. Wait, no! English names are
> in reverse order :-). So my name is Lajos Foldy in English.
