
Subject: Re: IDL 6.3 for Mac OS X on Intel Now Available

Posted by [FL](#) on Thu, 13 Jul 2006 20:10:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

FL can be started as 'fl --gui', which will present a very primitive interface, one window for standard input, and another for standard output (this is the default on Windows). However, the event loop is running in this case, and the results for TT3 and JD_TEST are the same.

I use gcc 3.4.x and gcc 4.1.x for developing FL, I think ITTVIS/RSI uses the same compiler (on linux, Intel compiler on Windows). I have written MMX/SSE/SSE2 array arithmetics routines in inline assembly, this can be accounted for some speedup. I have written my own memory allocators (more than one, different allocators for different usage patterns.) But the real difference comes from overall design, from the very first day performance was my main concern. Implementation is very important. Take eg. matrix multiplication in IDL and Matlab: for 1000x1000 matrices, Matlab is 6-7 times faster, although both are using the same BLAS API. The difference is in the implementation: IDL uses a simple C BLAS, while Matlab uses ATLAS (<http://math-atlas.sf.net>).

My plans:

- very short term: some more vacation :-)
- short term: widgets have absolute priority. Some basic functionality already work, usable widgets will be available, let's say, by end of September.
- long term (1-2 years):

DFL (Distributed FL): similar to the IDL-IDL bridge, but slave processes can be invoked on different machines, communicating through TCP/IP (FL's of the world, unite! :-)

Matlab front-end: I guess 90% of the present code can be reused in a Matlab-like interpreter. So, why not create a 2-in-1 monster?

The bad news is that FL will remain a one man show. I used to work in small groups earlier, but I have very bad memories. Countless hours were spent for meetings and communication, instead of productive work.

regards,
lajos

On Thu, 13 Jul 2006, JD Smith wrote:

- > Impressive. How much of FL's performance gain comes from compiler
 - > optimizations? I am suspect of speedups in operations limited by
 - > looping, since FL and GDL don't yet implement widgets or other even
 - > processing which slows IDL's loops down. However, just taking the
 - > square root of a bunch of numbers.... that's a different question.
 - >
 - > What are the plans for FL? Any hope of combining the two efforts
 - > (FL/GDL) into one open source project?
 - >
 - > JD
 - >
-