Subject: Re: IDL -> FORTRAN translator?
Posted by grunes on Wed, 23 Aug 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <41ctj7$oph@news.bu.edu> dlmatt@bu.edu (David Matthews) writes:

> We are writing a lot of IDL code to analyze large data sets.  A colleague at
> another institution doesn't want to use IDL, wants us to provide Fortran
> source to do the same jobs. Has anyone written a program to run on unix or VMS
> that will translate IDL source (*.pro) into Fortran? I recognize that the IDL
> approach is sufficiently different from Fortran that it might be very hard to
> do a complete job automatically, but Fortran equivalents to IDL are sometimes
> given in the IDL manuals.

If you get any useful responses, please share them with the news group.
I, for one would find any translater between IDL or PV-WAVE and any
compiled language, or the reverse, to be quite useful.

The problem is inherently dificult, in fact almost insoluable, since
compiled languages like Fortran require compile-time information about
symbols that is often dificult to obtain in interpreted languages like
IDL.

The obvious way of dealing with that--changing each symbol
reference to a subroutine call, which interprets stack contents, or some
such structure--is not only inefficient, but it would be unreadable.

(In a sense that was once done--the initial implementation of
IDL was written in Fortran.)

Some time ago (13 Jul 1995) I posted the reverse question:

> Do converters exist to translate Fortran or C into IDL or PV-WAVE?
>
> Obviously, a lot of posts have asked the reverse, and the
> answer always has to be that an Interpreter like IDL or PV-WAVE
> allows symbols to change their meanings (one moment it is a scalar integer,
> the next it is a character array, the next it is a procedure...) in such a
> way that it would be very hard to generate efficient code in a
> compiled language like Fortran or C.
>
> But this direction should mostly be possible.  And some of us would
> find it quite useful.  (But I don't want to write it.)  It would be
> especially useful if it had the option of taking some (perhaps user
> labeled) Fortran 77 loops and convert them into array operations.

I got a response from knipp@ipi.uni-hannover.de.  Unfortunately
it was fairly primitive, made some strong assumptions about the
code and the way (e.g., case) in which it was coded, and would

only work on a certain classof quite simple programs.  E.G.,
no dimensioned variables, etc.  But it did some of the work, by
changing "program" to "pro", ** to ^, .eq. to eq, etc.

------------------------------------------------------------ -------------
Mitchell R Grunes.   Opinions are mine alone.